Optimal Routing Tables*

Harry Buhrman
CWI

Jaap-Henk Hoepman
CWI

Paul Vitányi
CWI and University of Amsterdam

Abstract

The optimal space used to represent routing schemes in communication networks is established, both for worst-case static networks and on the average for all static networks. Several factors may influence the cost of representing a routing scheme for a particular network. It is therefore unavoidable that we first describe several reasonable models in which to measure this cost. Failure to do so in the past has obfuscated previous results.

We show that, in most models, for almost all graphs $\Theta(n^2)$ bits are necessary and sufficient for shortest path routing. By 'almost all graphs' we mean the Kolmogorov random graphs which constitute a fraction of $1-1/n^c$ of all graphs on n nodes, where $c \geq 3$ is an arbitrary fixed constant. In contrast, there is a model that rises the average case lower bound to $\Omega(n^2 \log n)$ and another model where the average case upper bound drops to $O(n \log^2 n)$. This clearly exposes the sensitivity of such bounds to the model under consideration. Furthermore, if paths have to be short, but need not be shortest (i.e., if the stretch factor may be larger than 1), our other upper bounds indicate that much less space is needed on average, even in the more demanding models.

For worst-case static networks we prove a $\Omega(n^2 \log n)$ lower bound for shortest path routing, for those models where the nodes in the network are labelled $1, \ldots, n$. This lower bound holds even for all stretch factors < 2.

Throughout, we use the incompressibility method based on Kolmogorov complexity.

1 Introduction

A universal *routing strategy* for static communication networks will, for every network, generate a *routing scheme* for that particular network. Such a routing scheme comprises a *local routing function* for every node in this network. The routing function of node u returns for every destination $v \neq u$ an edge incident to u on a path from u to v. This way, a routing scheme describes a path, called a *route*, between every pair of nodes u, v in the network. The *stretch factor* of a routing scheme equals the maximum ratio between the length of a route it produces, and the shortest path between the endpoints of that route. The stretch factor of a routing strategy equals the maximal stretch factor attained by any of the routing schemes it generates. If the stretch factor of a routing strategy equals 1, it is called a *shortest path routing strategy* because then it generates for every graph a routing scheme that will route a message between arbitrary u and v over a shortest path between u and v.

In a *full information* shortest path routing scheme, the routing function in u must, for each destination v return all edges incident to u on shortest paths from u to v. These schemes allow alternative, shortest, paths to be taken whenever an outgoing link is down.

We consider point to point communication networks on n nodes described by an undirected graph G. The nodes of the graph initially have unique labels taken from $\{1, \ldots, n\}$. Edges incident to a node v with degree $d(v)$ are connected to *ports*, with fixed labels $1, \ldots, d(v)$, by a so called *port assignment*. This coincides with the minimal local knowledge a node needs to route: a) a unique identity to determine whether it is the destination of an incoming message, b) the guarantee that each of its neighbours can be reached over a link connected to exactly one of its ports, and c) that it can distinguish these ports.

The space requirements of a routing scheme is measured as the sum over all nodes of the number of bits needed on each node to encode its routing function. If the nodes are not labelled with $\{1, \ldots, n\}$—the minimal set of labels—we have to add to the space requirement, for each node, the number of bits needed to encode its label. Otherwise, the bits needed to represent the routing function could be appended to the original identity yielding a large label that is not charged for but does contain all necessary information to route.

The cost of representing a routing function at a particular node depends on the amount of (uncharged) information initially there. Moreover, if we are allowed to relabel the graph and change its port assignment before generating a routing scheme for it, the resulting routing functions may be simpler and easier to encode. On a chain, for example, the routing function is much less complicated if we can relabel the graph and number the nodes in increasing order along the chain. We list these assumptions below, and argue that each of them is reasonable for certain systems. We

*All authors partially supported by the European Union through NeuroCOLT ESPRIT Working Group Nr. 8556, and by NWO through NFI Project ALADDIN number NF 62-376. Address: CWI, Kruislaan 413, 1098 SJ Amsterdam, The Netherlands. Email: {buhrman,jhh,paulv}@cwi.nl.

start with the three options for the amount of information initially available at a node.

I Nodes do not initially know the labels of their neighbours, and use ports to distinguish the incident edges. This models the basic system without prior knowledge.

  IA The assignment of ports to edges is fixed and cannot be altered. This assumption is reasonable for systems running several jobs where the optimal port assignment for routing may actually be bad for those other jobs.

  IB The assignment of ports to edges is free and can be altered before computing the routing scheme (as long as neighbouring nodes remain neighbours after re-assignment). Port re-assignment is justifiable as a local action that usually can be performed without informing other nodes.

II Nodes know the labels of their neighbours, and know over which edge to reach them. This information is for free. Or, to put it another way, an incident edge carries the same label as the node it connects to. This model is concerned only with the additional cost of routing messages beyond the immediate neighbours, and applies to systems where the neighbours are already known for various other reasons[1].

Orthogonal to that, the following three options regarding the labels of the nodes are distinguished.

α Nodes cannot be relabelled. For large scale distributed systems relabelling requires global coordination that may be undesirable or simply impossible.

β Nodes may be relabelled before computing the routing scheme, but the range of the labels must remain $1, \ldots, n$. This model allows a bad distributions of labels to be avoided.

γ Nodes may be given arbitrary labels before computing the routing scheme, but the number of bits used to store its label are added to the space requirements of a node. Destinations are given using the new, complex, labels[2]. This model allows us to store additional routing information, e.g. topological information, in the label of a node. This option is justified for centrally designed interconnect networks for multiprocessors and communication networks.

These two orthogonal sets of assumptions IA, IB, or II, and α, β, or γ, define the nine different models we will consider in this paper.

1.1 Summary of our results

We determine the optimum space used to represent shortest path routing schemes on almost all graphs, namely the Kolmogorov random graphs which constitute a fraction of at least $1-1/n^3$ of all graphs. These bounds straightforwardly imply the same bounds for the average case over all graphs. For an overview of the results, refer to Table 1[3].

We prove that for almost all graphs $\Omega(n^2)$ bits are necessary to represent the routing scheme, if relabelling is not allowed and nodes know their neighbours (II $\wedge$ α) or nodes do not know their neighbours (IA $\vee$ IB)[4]. Partially matching this lower bound, we show that $O(n^2)$ bits are sufficient to represent the routing scheme, if the port assignment may be changed or if nodes do know their neighbours (IB $\vee$ II). In contrast, for almost all graphs, the lower bound rises to $\Omega(n^2 \log n)$ bits if both relabelling and changing the port assignment is not allowed (IA $\wedge$ α). And, again for almost all graphs, the upper bound drops to $O(n \log^2 n)$ bits if nodes know the labels of their neighbours and nodes may be arbitrarily relabelled (II $\wedge$ γ).

Full information shortest path routing schemes are shown to require, on almost all graphs, $\Omega(n^3)$ bits to be stored, if relabelling is not allowed (α). This matches the obvious upper bound for all graphs.

For stretch factors larger than 1 we obtain the following results. When nodes know their neighbours (II), for almost all graphs, routing schemes achieving stretch factors s with $1 < s < 2$ can be stored using a total of $O(n \log n)$ bits[5]. Similarly, for almost all graphs in the same models (II), $O(n \log \log n)$ bits are sufficient for routing with stretch factor ≥ 2. Finally, for stretch factors $\geq 6 \log n$ on almost all graphs again in the same model (II), the routing scheme occupies only $O(n)$ bits.

For worst case static networks we prove, by construction of explicit graphs, a $\Omega(n^2 \log n)$ lower bound on the total size of any routing scheme with stretch factor < 2, if nodes may not be relabelled (α). The techniques used throughout are incompressibility arguments based on Kolmogorov complexity, [8].

1.2 Comparison with related work

Previous upper- and lower bounds on the total number of bits necessary and sufficient to store the routing scheme in worst-case static communication networks are due to Peleg and Upfal [9], and Fraigniaud and Gavoille [2]. Our bounds are stronger, because they apply to the average case as well.

In [9] it was shown that for any stretch factor $s \geq 1$, the total number of bits required to store the routing scheme for some n-node graph is at least $\Omega(n^{1+1/(2s+4)})$ and that there exist routing schemes for all n-node graphs, with stretch factor $s = 12k + 3$, using $O(k^3 n^{1+1/k} \log n)$ bits in total. For example, with stretch factor $s = 15$ we have $k = 1$ and their method guarantees $O(n^2 \log n)$ bits to store the routing scheme. The lower bound is shown in the model where nodes may be arbitrarily relabelled and where nodes

[1] We do not consider models that give neighbours for free and, at the same time, allow free port assignment. For, given a labelling of the edges by the nodes they connect to, the actual port assignment doesn't matter at all, and can in fact be used to represent $d(v) \log d(v)$ bits of the routing function. Namely, each assignment of ports corresponds to a permutation of the ranks of the neighbours — the neighbours at port i moves to position i. There are $d(v)!$ such permutations.

[2] In this model it is assumed that a routing function cannot tell valid from invalid labels, and that a routing function always receives a valid destination label as input. Requiring otherwise makes the problem harder.

[3] In this table, arrows indicate that the bound for that particular model follows from the bound found by tracing the arrow. In particular, the average case lower bound for model IA $\wedge$ β is the same as the IB $\wedge$ γ bound found by tracing $\rightarrow$ and $\downarrow$. The reader may have guessed that a ? marks an open question

[4] We write A $\vee$ B to indicate that the results hold under model A or model B. Similarly, we write A $\wedge$ B to indicate the result holds only if the conditions of both model A and model B hold simultaneously. If only one of the two 'dimensions' is mentioned, the other may be taken arbitrary (i.e., IA is a shorthand for (IA $\wedge$ α) $\vee$ (IA $\wedge$ β) $\vee$ (IA $\wedge$ γ)).

[5] For Kolmogorov random graphs which have diameter 2 by Lemma 2 routing schemes with $s = 1.5$ are the only ones possible in this range.

	no relabelling (α)	*permutation* (β)	*free relabelling* (γ)
worst case — *lower bounds*			
port assignment free (IB)	$\rightarrow$	$\Omega(n^2 \log n)$ [3]	?
neighbours known (II)	$\Omega(n^2 \log n)$	$\Omega(n^2)$ [2]	$\Omega(n^{7/6})$ [9]
average case — *upper bounds*			
port assignment fixed (IA)	$O(n^2 \log n)$	$\leftarrow$	$\leftarrow$
port assignment free (IB)	$O(n^2)$	$\leftarrow$	$\leftarrow$
neighbours known (II)	$O(n^2)$	$\leftarrow$	$O(n \log^2 n)$
average case — *lower bounds*			
port assignment fixed (IA)	$\Omega(n^2 \log n)$	$\rightarrow$	$\downarrow$
port assignment free (IB)	$\downarrow$	$\rightarrow$	$\Omega(n^2)$
neighbours known (II)	$\Omega(n^2)$	?	?

Table 1: Size of shortest path routing schemes: overview of results

know their neighbours (II $\wedge$ γ). Free port-assignment in conjunction with a model where the neighbours are known (II) can, however, not be allowed. Otherwise, each node would gain $n \log n$ bits to store the routing function in (see the footnote to model II).

Fraigniaud and Gavoille [2] showed that for stretch factors $s < 2$ there are routing schemes that require a total of $\Omega(n^2)$ bits to be stored in the worst case if nodes may be relabelled by permutation (β).

Kranakis *et al.* in [6, 7, 5] independently use Kolmogorov complexity to obtain results on interval routing, bounded degree graphs, full information routing, shortest path families, and an alternative proof for the result of Fraigniaud and Gavoille [2]. In particular, they show that for each n *there exist* graphs on n nodes (actually about a $1/2^{n/2}$th fraction of all such graphs), which may not be relabelled (α), that require $\Omega(n^3)$ bits to store a *full information* shortest path routing scheme.

Finally, Gavoille and Pérennès [3] recently showed that there are routing schemes that require a total of $\Omega(n^2 \log d)$ bits to be stored in the worst case for some graphs with maximal degree d, if nodes may be relabelled by permutation and the port-assignment may be changed (IB $\wedge$ β).

To the best of our knowledge, Jan van Leeuwen was the first to formulate explicitly the question of what exactly is the optimal size of the routing functions, and he recently drew also our attention to this group of problems. See also [1].

2 Kolmogorov complexity

The Kolmogorov complexity, [4], of x is the length of the *shortest* effective description of x. That is, the *Kolmogorov complexity* $C(x)$ of a finite string x is simply the length of the shortest program, say in FORTRAN (or in Turing machine codes) encoded in binary, which prints x without any input. A similar definition holds conditionally, in the sense that $C(x|y)$ is the length of the shortest binary program which computes x given y as input. It can be shown that the Kolmogorov complexity is absolute in the sense of being independent of the programming language, up to a fixed additional constant term which depends on the programming language but not on x. We now fix one canonical programming language once and for all as reference and thereby $C()$.

For the theory and applications, see [8]. Let $x, y, z \in \mathcal{N}$, where $\mathcal{N}$ denotes the natural numbers. Identify $\mathcal{N}$ and $\{0,1\}^*$ according to the correspondence $(0,\epsilon), (1,0), (2,1), (3,00), (4,01), \ldots$. Hence, the length $|x|$ of x is the number of bits in the binary string x. Let $T_1, T_2, \ldots$ be a standard enumeration of all Turing machines. Let $\langle \cdot, \cdot \rangle$ be a standard invertible effective bijection from $\mathcal{N} \times \mathcal{N}$ to $\mathcal{N}$. This can be iterated to $\langle \langle \cdot, \cdot \rangle, \cdot \rangle$.

Definition 1 Let U be an appropriate universal Turing machine such that $U(\langle \langle i, p \rangle, y \rangle) = T_i(\langle p, y \rangle)$ for all i and $\langle p, y \rangle$. The *Kolmogorov complexity* of x given y (for free) is

$$C(x|y) = \min\{|p| : U(\langle p, y \rangle) = x, p \in \{0,1\}^*\}.$$

3 Kolmogorov random graphs

One way to express irregularity or *randomness* of an individual network topology is by a modern notion of randomness like Kolmogorov complexity. A simple counting argument shows that for each y in the condition and each length n there exists at least one x of length n which is *incompressible* in the sense of $C(x|y) \geq n$, 50% of all x's of length n is incompressible but for 1 bit ($C(x|y) \geq n-1$), 75% of all x's is incompressible but for 2 bits ($C(x|y) \geq n-2$) and in general a fraction of $1 - 1/2^c$ of all strings cannot be compressed by more than c bits, [8].

Definition 2 Each graph $G = (V, E)$ on n nodes $V = \{1, 2, \ldots, n\}$ can be coded by a binary string $E(G)$ of length $n(n-1)/2$. We enumerate the $n(n-1)/2$ possible edges uv in a graph on n nodes in standard lexicographical order without repetitions and set the ith bit in the string to 1 if the i-th edge is present and to 0 otherwise. Conversely, each binary string of length $n(n-1)/2$ encodes a graph on n nodes. Hence we can identify each such graph with its corresponding binary string.

Definition 3 Let δ be a simply described recursive function over the natural numbers, such as $\log n, \log\log n, \sqrt{n}$. That is,

$$C(\delta) := \min\{C(T) : T \text{ is a Turing machine computing } \delta\}$$

is bounded by a small fixed constant. An individual graph G on n nodes is δ-*random* (in contrast to a randomly generated graph) if it satisfies

$$C(E(G)|n) \geq n(n-1)/2 - \delta(n), \tag{1}$$

Elementary counting shows that *a fraction* of at least

$$1 - 1/2^{\delta(n)}$$

of all graphs on n nodes has that high complexity, [8]. We need the notion of self-delimiting binary strings.

DEFINITION 4 We call x a *prefix* of y if there is a z such that $y = xz$. A set $A \subseteq \{0,1\}^*$ is *prefix-free*, if no element in A is the prefix of another element in A. A 1:1 function $E : \{0,1\}^* \to \{0,1\}^*$ (equivalently, $E : \mathcal{N} \to \{0,1\}^*$) defines a *prefix-code* if its range is prefix-free. A simple prefix-code we use throughout is obtained by reserving one symbol, say 0, as a stop sign and encoding

$$\begin{aligned} \bar{x} &= 1^{|x|}0x, \\ |\bar{x}| &= 2|x| + 1. \end{aligned}$$

Sometimes we need the shorter prefix-code x':

$$\begin{aligned} x' &= \overline{|x|}x, \\ |x'| &= |x| + 2\lceil \log(|x|+1) \rceil + 1. \end{aligned}$$

We call $\bar{x}$ or x' a *self-delimiting* version of the binary string x. We can effectively recover both x and y unambiguously from the binary strings $\bar{x}y$ or $x'y$. For example, if $\bar{x}y = 111011011$, then $x = 110$ and $y = 11$. If $\bar{x}\bar{y} = 1110110101$ then $x = 110$ and $y = 1$. The self-delimiting form $x' \ldots y'z$ allows the concatenated binary sub-descriptions to be parsed and unpacked into the individual items $x, \ldots, y, z$; the code x' encodes a separation delimiter for x, using $2\lceil \log(|x|+1) \rceil$ extra bits, and so on, [8].

LEMMA 1 *The degree d of each node of a δ-random graph satisfies*

$$|d - (n-1)/2| = O\left(\sqrt{(\delta(n) + \log n)n}\right).$$

PROOF. Assume that the deviation of the degree d of a node u in G from $(n-1)/2$ is at least k. From the lower bound on $C(E(G)|n)$ corresponding to the assumption that G is random, we can estimate an upper bound on k, as follows.

Describe $G = (V, E)$ given n as follows. We can indicate which edges are incident on node u by giving the index of the interconnection pattern (the characteristic sequence of the set $V_u = \{v \in V - \{u\} : uv \in E\}$ in $n-1$ bits where the v-th bit is 1 if $v \in V_u$ and 0 otherwise) in the ensemble of

$$m = \sum_{|d-(n-1)/2| \geq k} \binom{n-1}{d} \leq 2^n e^{-k^2/(n-1)} \tag{2}$$

possibilities. The last inequality follows from a general estimate of the tail probability of the binomial distribution, with s_n the number of successful outcomes in n experiments with probability of success $0 < p < 1$ and where $q = 1 - p$. Namely, Chernoff's bounds, [8], pp. 127-130, give

$$\Pr(|s_n - np| \geq k) \leq 2e^{-k^2/4npq}. \tag{3}$$

To describe G it then suffices to modify the old code of G by prefixing it with

- A description of this discussion in $O(1)$ bits;
- the identity of node u in $\lceil \log(n+1) \rceil$ bits,
- the value of d in $\lceil \log(n+1) \rceil$ bits, possibly adding non-significant 0's to pad up to this amount,
- the index of the interconnection pattern in $\log m + 2 \log\log m$ bits in self-delimiting form, [6]

followed by the old code for G with the bits in the code denoting the presence or absence of the possible edges which are incident on node u deleted.

Clearly, given n we can reconstruct the graph G from the new description. The total description we have achieved is an effective program of

$$\log m + 2\log\log m + O(\log n) + n(n-1)/2 - (n-1)$$

bits. This must be at least the length of the shortest effective binary program, which is $C(E(G)|n)$ satisfying Eq. (1). Therefore,

$$\log m + 2\log\log m \geq n - 1 - O(\log n) - \delta(n).$$

Since we have estimated in Eq. (2) that

$$\log m \leq n - 1 - (k^2/(n-1)) \log e,$$

it follows that

$$k = O\left(\sqrt{(\delta(n) + \log n)n}\right).$$

□

LEMMA 2 *All $o(n)$-random graphs have diameter 2.*

PROOF. The only graphs with diameter 1 are the complete graphs which can be described in $O(1)$ bits, given n, and hence are not random. It remains to consider $G = (V, E)$ is an $o(n)$-random graph with diameter greater than 2. Let u, v be a pair of nodes with distance greater than 2. Then we can describe G by modifying the old code for G by prefixing it with

- A description of this discussion in $O(1)$ bits;
- The identities of $u < v$ in $2 \log n$ bits,
- The old code $E(G)$ of G with, for each w with $uw \in E$, all bits representing presence or absence of an edge wv between w and v deleted. We know that all the bits representing such edges must be 0 since the existence of any such edge shows that uw, wv is a path of length 2 between u and v contradicting the assumption that u and v have distance > 2. This way we save at least $n/4$ bits, since we save bits for as many edges wv as there are edges uw, that is, the degree of u which is $n/2 \pm o(n)$ by Lemma 1.

Since we know the identities of u and v, and the nodes adjacent to u (they can be obtained from $E(G)$ because $u < v$) we can reconstruct G from this discussion and the new description, given n. Since by Lemma 1 the degree of u is at least $n/4$, the new description of G, given n, requires at most

$$n(n-1)/2 - n/4 + O(\log n)$$

bits, which contradicts Eq. (1) from some n onwards. □

[6] From now on we write simply '$\log n$' for '$\lceil \log(n+1) \rceil$' in cases where the difference clearly doesn't matter.

LEMMA 3 *Let c be a fixed constant. If G is $c\log n$-random then from each node u all other nodes are either directly connected to u or are directly connected to one of the least $(c+3)\log n$ nodes directly adjacent to u.*

PROOF. Given u, let A be the set of the least $(c+3)\log n$ nodes directly adjacent to u. Assume, by way of contradiction, there is a node w of G that is not directly connected to a node in $A\bigcup\{u\}$. We can describe G as follows.

- A description of this discussion in $O(1)$ bits.
- A literal description of u in $\log n$ bits.
- A literal description of the presence or absence of edges between u and the other nodes in $n-1$ bits.
- A literal description of w and the presence or absence of edges between w and the other nodes in $\log n + n - 2 - (c+3)\log n$ bits (by omitting the bits corresponding to the least $(c+3)\log n$ nodes directly adjacent to u).
- The encoding $E(G)$ with the edges incident with nodes u and w deleted, saving at least $2n-2$ bits,

Altogether the resultant description has

$$n(n-1)/2 + 2\log n + 2n - 3 - (c+3)\log n - 2n + 2$$

bits which contradicts the $c\log n$-randomness of G by Eq. (1). The lemma is proven. □

4 Upper bounds

In this section we show how one can route messages over Kolmogorov random graphs with routing schemes that can be stored efficiently. Specifically we show that in general (i.e., on almost all graphs) one can use shortest path routing schemes occupying at most $O(n^2)$ bits. If one can relabel the graph in advance, and if nodes know their neighbours, shortest path routing schemes are shown to occupy only $O(n\log^2 n)$ bits. Allowing stretch factors larger than one reduces the space requirements — even as low as $O(n)$ bits for stretch factors of $O(\log n)$.

THEOREM 1 *For shortest path routing in $O(\log n)$-random graphs, where the port assignment may be changed or nodes know their neighbours (IB $\vee$ II), it suffices to have local routing functions stored in $6n$ bits per node (hence the complete routing scheme is represented by $6n^2$ bits).*

PROOF. We prove the theorem for the model where nodes know their neighbours (II), without resorting to relabelling (i.e., nodes are labelled 1 through n). If, instead, the port assignment may be chosen arbitrarily, we can represent knowledge of the neighbours and the edges over which they are reached as follows. Neighbours are coded using the standard interconnection vector (as in Def. 2) using $n-1$ bits. The port mapping is chosen such that the i-th neighbour is connected to the i-th port. This adds only $n-1$ bits per node to the local routing function to be constructed next. Hence the theorem holds for the model with arbitrary port mapping as well (IB).

Let G be an $O(\log n)$-random graph on n nodes. By Lemma 3 we know that from each node u we can shortest path route to each node v through the first $O(\log n)$ directly adjacent nodes of u. By Lemma 2, G has diameter 2. Once the message reached node v the destination is either node v or a direct neighbour of node v which is known in node v by assumption of our routing model. It follows readily from Lemma 3 that routing functions of size $O(n\log\log n)$ can be used to do shortest-path routing. As we will see we can do better than this. Let $A_0 \subseteq V$ be the set of nodes in G which are not directly connected to u.

CLAIM 1 Let $v_1, \ldots, v_m$ be $O(\log n)$ directly adjacent nodes to u through which we can shortest path route to all nodes in A_0. (For example, the $O(\log n)$ least nodes directly adjacent to node u, Lemma 3.) For $t := 1, 2\ldots, l$ define $A_t := \{w \in A_0 - \bigcup_{s=1}^{t-1} A_s : v_t w \in E\}$. Let $m_0 = |A_0|$ and define $m_{t+1} = m_t - |A_{t+1}|$. Let l be the first t such that $m_t < n/\log\log n$. Then, $|A_t| > 1/3 * m_{t-1}$ for $1 \le t \le l$. This means that v_t is connected by an edge in E to at least 1/3 of the nodes not connected by edges in E to nodes $u, v_1, \ldots, v_{t-1}$.

PROOF. Suppose, by way of contradiction, that there exists a least $t \le l$ such that $||A_t| - m_{t-1}/2| \ge (1/6)m_{t-1}$. Then we can describe G, given n, as follows.

- This discussion in $O(1)$ bits.
- Nodes u, v_t in $2\log n$ bits.
- The presence or absence of edges incident with nodes $u, v_1, \ldots, v_{t-1}$ in $r = n-1+\cdots+n-(t-1)$ bits. This gives us the characteristic sequences of $A_0, \ldots, A_{t-1}$ in V[7].
- A self-delimiting description of the characteristic sequence of A_t in $A_0 - \bigcup_{s=1}^{t-1} A_s$, using Chernoff's bound as cited in Eq. (2), in at most $m_{t-1} - (1/6)^2 m_{t-1}\log e + O(\log m_{t-1})$ bits.
- The description $E(G)$ with all bits corresponding to the presence or absence of edges between v_t and the nodes in $A_0 - \bigcup_{s=1}^{t-1} A_s$ deleted, saving m_{t-1} bits. Furthermore, we delete also all bits corresponding to presence or absence of edges incident with $u, v_1, \ldots, v_{t-1}$ saving a further r bits.

This description of G uses at most

$$n(n-1)/2 + O(\log n) + m_{t-1} - (1/6)^2 m_{t-1}\log e - m_{t-1}$$

bits, which contradicts the $O(\log n)$-randomness of G by Eq. (1), because $m_{t-1} > n/\log\log n$. □

Recall that l is the first time in the construction such that $m_l < n/\log\log n$. By Lemma 3, $l = O(\log n)$. We construct the local routing function $F(u)$ as follows.

- A table of intermediate routing node entries for all the nodes in A_0 in increasing order. For each node w in $\bigcup_{s=1}^{l} A_s$ we enter in the w-th position in the table the unary representation of the least intermediate node v, with $uv, vw \in E$, followed by a 0. For the nodes that are not in $\bigcup_{s=1}^{l} A_s$ we enter a 0 in their position in the table indicating that an entry for this node can be found in the second table. By Claim 1, the size of this table is bounded by:

$$n + \sum_{s=1}^{l} (1/3)(2/3)^{s-1} sn \le n + \sum_{s=1}^{\infty} (1/3)(2/3)^{s-1} sn \le 4n$$

[7] A characteristic sequence of A in V is a string of $|V|$ bits with for each $v \in V$ the v-th bit equals 1 if $v \in A$ and the v-th bit is 0 otherwise.

- A table with explicitly binary coded intermediate nodes on a shortest path for the ordered set of the remaining destination nodes. Those nodes had a 0 entry in the first table and there are at most $m_l < n/\log\log n$ of them, namely the nodes in $A_0 - \bigcup_{s=1}^{l} A_s$. Each entry consists of the code of length $\log\log n + O(1)$ for the position in increasing order of a node out of $v_1, \ldots, v_m$ with $m = O(\log n)$ by Lemma 3. Hence this second table requires at most $2n$ bits.

The routing algorithm is as follows. The direct neighbours of u are known in node u and are routed without routing table. If we route from start node u to target node w which is not directly adjacent to u, then we do the following. If node w has an entry in the first table then route over the edge coded in unary, otherwise find an entry for node w in the second table.

Altogether, we have $|F(u)| \leq 6n$. Adding another $n-1$ in case the port assignment may be chosen arbitrarily, this proves the theorem with $7n$ instead of $6n$. Slightly more precise counting and choosing l such that m_l is the first such quantity $< n/\log n$ shows $|F(u)| \leq 3n$. $\square$

If we allow arbitrary labels for the nodes, then shortest path routing schemes of $O(n \log^2 n)$ bits suffice on Kolmogorov random graphs, as witnessed by the following theorem.

THEOREM 2 *For shortest path routing on* $c\log n$*-random graphs, if nodes know their neighbours and nodes may be arbitrarily relabelled (II $\wedge$ γ), using labels of size* $(1 + (c + 3)\log n)\log n$ *bits results in local routing functions stored in* $O(1)$ *bits per node (hence the complete routing scheme is represented by* $(c+3)n\log^2 n + n\log n + O(n)$ *bits).*

PROOF. Let $G = (V, E)$ be a $c\log n$-random graph on n nodes. By Lemma 3 we know that from each node u we can shortest path route to each node w through the first $(c+3)\log n$ directly adjacent nodes $f(u) = v_1, \ldots, v_m$ of u. By lemma 2, G has diameter 2. Relabel G such that the label of node u equals u followed by the original labels of the first $(c+3)\log n$ directly adjacent nodes $f(u)$. This new label occupies $(1 + (c+3)\log n)\log n$ bits. To route from source u to destination v do the following.

If v is directly adjacent to u we route to v in 1 step in our model (nodes know their neighbours). If v is not directly adjacent to u, we consider the immediate neighbours $f(v)$ contained in the name of v. By Lemma 3 at least one of the neighbours of u must have a label whose original label (stored in the first $\log n$ bits of its new label) corresponds to one of the labels in $f(v)$. Node u routes the message to any such neighbour. This routing function can be stored in $O(1)$ bits. $\square$

Without relabelling routing using less than $O(n^2)$ bits is possible if we allow stretch factors larger than 1. The next three theorems clearly show a trade-off between the stretch factor and the size of the routing scheme.

THEOREM 3 *For routing with any stretch factor* > 1 *in* $c\log n$*-random graphs, where nodes know their neighbours (II), it suffices to have* $n - 1 - (c+3)\log n$ *nodes with local routing functions stored in at most* $\lceil \log(n+1) \rceil$ *bits per node, and* $1 + (c+3)\log n$ *nodes with local routing functions stored in* $6n$ *bits per node (hence the complete routing scheme is represented by less than* $(6c+20)n\log n$ *bits).*

PROOF. Let $G = (V, E)$ be a $c\log n$-random graph on n nodes. By Lemma 3 we know that from each node u we can shortest path route to each node w through the first $(c+3)\log n$ directly adjacent nodes $v_1, \ldots, v_m$ of u. By Lemma 2, G has diameter 2. Consequently, each node in V is directly adjacent to some node in $B = \{u, v_1, \ldots, v_m\}$. Hence, it suffices to select the nodes of B as routing centers and store, in each node $w \in B$, a shortest path routing function $F(w)$ to all other nodes, occupying $6n$ bits (the same routing function as constructed in the proof of Theorem 1 if the neighbours are known). Nodes $v \in V - B$ route any destination unequal to their own label to some fixed directly adjacent node $w \in B$. Then $|F(v)| \leq \lceil \log(n+1) \rceil + O(1)$, and this gives the bit count in the theorem

To route from a originating node v to a target node w the following steps are taken. If w is directly adjacent to v we route to w in 1 step in our model. If w is not directly adjacent to v then we first route in 1 step from v to its directly connected node in B, and then via a shortest path to w. Altogether, this takes either 2 or 3 steps whereas the shortest path has length 2. Hence the stretch factor is at most 1.5 which for graphs of diameter 2 (i.e., all $c\log n$-random graphs by Lemma 2) is the only possibility between stretch factors 1 and 2. This proves the theorem. $\square$

THEOREM 4 *For routing with stretch factor 2 in* $c\log n$*-random graphs, if nodes know their neighbours (II), it suffices to have* $n-1$ *nodes with local routing functions stored in at most* $\log\log n$ *bits per node and* 1 *node with its local routing function stored in* $6n$ *bits (hence the complete routing scheme is represented by* $n\log\log n + 6n$ *bits).*

PROOF. Let G be a $c\log n$-random graph on n nodes. By Lemma 2, G has diameter 2. Therefore the following routing scheme has stretch factor 2. Let node 1 store a shortest path routing function. All other nodes only store a shortest path to node 1. To route from a originating node v to a target node w the following steps are taken. If w is an immediate neighbour of v, we route to w in 1 step in our model. If not, we first route the message to node 1 in at most 2 steps, and then from node 1 through a node v to node w in again 2 steps. Because node 1 stores a shortest path routing function, either $v = w$ or w is a direct neighbour of v.

Node 1 can store a shortest path routing function in at most $6n$ bits using the same construction as used in the proof of Theorem 1 (if the neighbours are known). The immediate neighbours of 1 either route to 1 or directly to the destination of the message. For these nodes, the routing function occupies $O(1)$ bits. For nodes v at distance 2 of node 1 we use Lemma 3, which tells us that we can shortest path route to node 1 through the first $(c+3)\log n$ directly adjacent nodes of v. Hence, to represent this edge takes $\log\log n + \log(c+3)$ bits and hence the local routing function $F(v)$ occupies at most $\log\log n + O(1)$ bits. $\square$

THEOREM 5 *For routing with stretch factor* $(c+3)\log n$ *in* $c\log n$*-random graphs, where nodes know their neighbours (II), it suffices to have local routing functions stored in* $O(1)$ *bits per node (hence the complete routing scheme is represented by* $O(n)$ *bits).*

PROOF. From Lemma 3 we know that from each node u we can shortest path route to each node v through the first $(c+3)\log n$ directly adjacent nodes of u. By Lemma 2, G has diameter 2. So the local routing function — representable in $O(1)$ bits — is to route directly to the target node if it is a directly adjacent node, otherwise to simply traverse the first $(c+3)\log n$ incident edges of the starting node and look in each of the visited nodes whether the target node is a directly adjacent node. If so, the message is forwarded to that node, otherwise it is returned to the starting node for trying the next node. Hence each message for a destination at distance 2 traverses at most $2(c+3)\log n$ edges. □

5 Lower bounds

The first two theorems of this section together show that indeed $\Omega(n^2)$ bits are necessary to route on Kolmogorov random graphs in all models we consider, except for the models where nodes know their neighbours and relabelling is allowed (II $\wedge$ β and II $\wedge$ γ). Hence the upper bound in Theorem 1 is tight.

THEOREM 6 *For shortest path routing in $o(n)$-random graphs where relabelling is not allowed and nodes know their neighbours (II $\wedge$ α), each local routing function must be stored in at least $n/2-o(n)$ bits per node (hence the complete routing scheme requires at least $n^2/2-o(n^2)$ bits to be stored).*

PROOF. Let G be a $o(n)$-random graph. Let $F(u)$ be the local routing function of node u of G, and let $|F(u)|$ be the number of bits used to store $F(u)$. Let $E(G)$ be the standard encoding of G in $n(n-1)/2$ bits as in Def. 2. We now give another way to describe G using some local routing function $F(u)$.

- A description of this discussion in $O(1)$ bits.
- A description of u in $\log n$ bits. (If it is less pad the description with 0's.)
- A description of the presence or absence of edges between u and the other nodes in V in $n-1$ bits.
- A description of $F(u)$ in $|F(u)|+O(\log|F(u)|)$ bits (the logarithmic term to make the description self-delimiting).
- The code $E(G)$ with all bits deleted corresponding to edges $vw \in E$ for each w and v such that $F(u)$ routes messages to w through the least[8] intermediary node v (saving at least $n/2-o(n)$ bits since there are at least $n/2-o(n)$ nodes w such that $uw \notin E$ by Lemma 1 and since the diameter of G is 2 by Lemma 2 there is a shortest path $uv, vw \in E^2$ for some v).[9] Furthermore, also all bits deleted corresponding to the presence or absence of edges between u and the other nodes in V, saving another $n-1$ bits.

From this description we can reconstruct G, given n, by reconstructing the bits corresponding to the deleted edges from u and $F(u)$ and subsequently inserting them in the appropriate positions to reconstruct $E(G)$. We can do so because these positions can be simply reconstructed in increasing order. In total this new description has

$$n(n-1)/2+O(1)+O(\log n)+|F(u)|-n/2+o(n)$$

which must be at least $n(n-1)/2-o(n)$ by Eq. (1). We conclude that $|F(u)|=n/2-o(n)$, which proves the theorem. □

THEOREM 7 *For shortest path routing in $o(n)$-random graphs, if the neighbours are not known (IA $\vee$ IB), the complete routing scheme requires at least $n^2/32-o(n^2)$ bits to be stored.*

PROOF. In the proof of this theorem we need the following combinatorial result.

CLAIM 2 *Let k and n be arbitrary natural numbers such that $1 \le k \le n$. Let x_i, for $1 \le i \le k$, be natural numbers such that $x_i \ge 1$. If $\sum_{i=1}^k x_i = n$, then*

$$\sum_{i=1}^{k}\lceil \log x_i \rceil \le n-k$$

PROOF. By induction on k. If $k=1$, then $x_1=n$ and clearly $\lceil \log n\rceil \le n-1$ if $n\ge 1$. Supposing the claim holds for k and arbitrary n and x_i, we now prove it for $k'=k+1$, n and arbitrary x_i. Let $\sum_{i=1}^{k'} x_i = n$. Then $\sum_{i=1}^{k} x_i = n-x_{k'}$. Now

$$\sum_{i=1}^{k'}\lceil \log x_i\rceil = \sum_{i=1}^{k}\lceil \log x_i\rceil + \lceil \log x_{k'}\rceil$$

By the induction hypothesis the first term on the right-hand side is less than or equal to $\le n-x_{k'}-k$, so

$$\sum_{i=1}^{k'}\lceil \log x_i\rceil \le n-x_{k'}-k+\lceil \log x_{k'}\rceil = n-k'+\lceil \log x_{k'}\rceil+1-x_{k'}$$

Clearly $\lceil \log x_{k'}\rceil+1 \le x_{k'}$ if $x_{k'}\ge 1$, which proves the claim. □

If we cannot enumerate the labels of all nodes in less than $n^2/32$ we are done. So assume we can (this includes models α and β where the labels are not charged for, but can be described using $\log n$ bits). Let G be an $o(n)$-random graph.

CLAIM 3 Given the labels of all nodes, we can describe the interconnection pattern of a node u using the local routing function of node u plus an additional $n/2+o(n)$ bits.

PROOF. Apply the local routing function to each of the labels of the nodes in turn (these are given by assumption). This will return for each edge a list of destinations reached over that edge. To describe the interconnection pattern it remains to encode, for each edge, which of the destinations reached is actually its immediate neighbour. If edge i routes x_i destinations, this will cost $\lceil \log x_i\rceil$ bits. By Lemma 1 the degree of a node in G is at least $n/2-o(n)$. Then in total, $\sum_{i=1}^{n/2-o(n)}\lceil \log x_i\rceil$ bits will be necessary; separations need not be encoded because they can be determined using the knowledge of all x_i's. Using Claim 2 finishes the proof. □

Now we show that there are $n/2$ nodes in G whose local routing function requires at least $n/8-3\log n$ bits to describe (which implies the theorem).

Assume, by way of contradiction, that there are $n/2$ nodes in G whose local routing function requires at most $n/8-3\log n$ bits to describe. Then we can describe G as follows:

[8] $F(u)$ may route from node u to node w by many different paths of length 2 using more than one intermediary node v. Allowing this may have as consequence that there is a more compact way of encoding $F(u)$, and makes our lower bound results stronger.

[9] Note that we cannot save the bits for the presence or absence of edges uv and uw. Namely, the routing function $F(u)$ may use the particular connection pattern of u saying things like 'route to node w using the 30th least node directly connected to u'. Thus, $F(u)$ may route *using* the connection pattern of node u, while that pattern cannot be reconstructed from $F(u)$.

- A description of this discussion in $O(1)$ bits,
- The enumeration of all labels in at most $n^2/32$ (by assumption),
- A description of the $n/2$ nodes in this enumeration in at most n bits,
- The interconnection patterns of these $n/2$ nodes in $n/8 - 3\log n$ plus $n/2 + o(n)$ bits each (by assumption, and using Claim 3). This amounts to $n/2(5n/8 - 3\log n) + o(n^2)$ bits in total, with separations encoded in another $n \log n$ bits,
- The interconnection patterns of the remaining $n/2$ nodes *only among themselves* using the standard encoding, in $1/2(n/2)^2$ bits.

This description altogether uses

$$\begin{aligned} &O(1) + n^2/32 + n + n/2(5n/8 - 3\log n) + \\ &\quad + o(n^2) + n\log n + 1/2(n/2)^2 = \\ &\quad = n^2/2 - n^2/32 + n + o(n^2) - n/2 \log n \end{aligned}$$

bits, contradicting the $o(n)$-randomness of G by Eq. (1). We conclude that on at least $n/2$ nodes a total of $n^2/16 - o(n^2)$ bits are used to store the routing scheme. □

If neither relabelling nor changing the port assignment is allowed, the next theorem implies that for shortest path routing on such 'static' graphs one cannot do better than storing the routing tables literally, in $O(n^2 \log n)$ bits.

THEOREM 8 *For shortest path routing in $o(n)$-random graphs where relabelling and changing the port assignment is not allowed (IA $\wedge$ α), each local routing function must be stored in at least $n/2 \log n/2 - O(n)$ bits per node (hence the complete routing scheme requires at least $n^2/2 \log n/2 - O(n^2)$ bits to be stored).*

PROOF. If the graph cannot be relabelled and the port-assignment cannot be changed, the adversary can set the port-assignment of each node to correspond to a permutation of the neighbours. As each node has $n/2 - o(n)$ neighbours by Lemma 1, such a permutation can have Kolmogorov complexity as high as $n/2 \log n/2 - O(n)$ [8]. Because the neighbours are not known, the local routing function must for each neighbour determine the port to route messages for that neighbour over. Hence the local routing function completely describes the permutation and thus it must also occupy at least $n/2 \log n/2 - O(n)$ bits. This proves the theorem. □

Even if stretch factors between 1 and 2 are allowed, the next theorem shows that $\Omega(n^2 \log n)$ bits are necessary to represent the routing scheme in the worst case.

THEOREM 9 *For routing with stretch factor < 2 in graphs where relabelling is not allowed (α), there exist graphs on n nodes (almost $(n/3)!$ such graphs) where the local routing function must be stored in at least $(n/3)\log n - O(n)$ bits per node at $n/3$ nodes (hence the complete routing scheme requires at least $(n^2/9)\log n - O(n^2)$ bits to be stored).*

PROOF. Consider the graph G_k with $n = 3k$ nodes depicted in Figure 1. Each node v_i in $v_{k+1}, \ldots, v_{2k}$ is connected to v_{i+k} and to each of the nodes $v_1, \ldots, v_k$. Fix a labelling of the nodes $v_1, \ldots, v_{2k}$ with labels from $\{1, \ldots, 2k\}$. Then any labelling of the nodes $v_{2k+1}, \ldots, v_{3k}$ with labels from $\{2k+1, \ldots, 3k\}$ corresponds to a permutation of $\{2k+1, \ldots, 3k\}$ and vice versa.

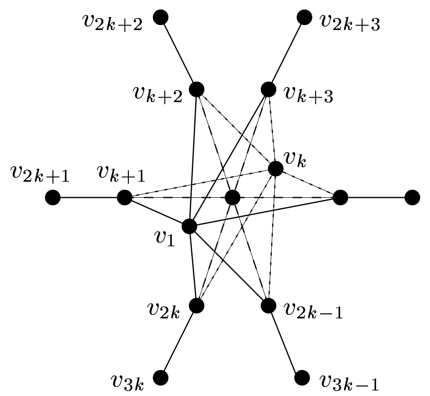

Figure 1: Graph G_k.

Clearly, for any two nodes v_i and v_j with $1 \leq i \leq k$ and $2k+1 \leq j \leq 3k$, the shortest path from v_i to v_j passes through node v_{j-k} and has length 2, whereas any other path from v_i to v_j has length at least 4. Hence any routing function on G_k with stretch factor < 2 routes such v_j from v_i over the edge $v_i v_{j-k}$. Then at each of the k nodes $v_1, \ldots, v_k$ the local routing functions corresponding to any two labellings of the nodes $v_{2k+1}, \ldots, v_{3k}$ are different. Hence each representation of a local routing function at the k nodes v_i, $1 \leq i \leq k$, corresponds one-one to a permutation of $\{2k+1, \ldots, 3k\}$. So given such a local routing function we can reconstruct the permutation (by collecting the response of the local routing function for each of the nodes $k+1, \ldots, 3k$ and grouping all pairs reached over the same edge). The number of such permutations is $k!$. A fraction at least $1 - 1/2^k$ of such permutations π has Kolmogorov complexity $C(\pi) = k \log k - O(k)$ [8]. Because π can be reconstructed given any of the k local routing functions, these k local routing functions each must have Kolmogorov complexity $k \log k - O(k)$ too. This proves the theorem for n a multiple of 3. For $n = 3k - 1$ or $n = 3k - 2$ we can use G_k, dropping v_k and v_{k-1}. □

Our last theorem shows that for full information shortest path routing schemes on Kolmogorov random graphs one cannot do better than the trivial upper bound.

THEOREM 10 *For full-information shortest path routing on $o(n)$-random graphs where relabelling is not allowed (α), the local routing function occupies at least $n^2/4 - o(n^2)$ bits for every node (hence the complete routing scheme requires at least $n^3/4 - o(n^3)$ bits to be stored).*

PROOF. Let G be a graph on nodes $\{1, 2, \ldots, n\}$ satisfying Eq. (1) with $\delta(n) = o(n)$. Then we know that G satisfies Lemmas 1, 2. Let $F(u)$ be the local routing function of node u of G, and let $|F(u)|$ be the number of bits used to encode $F(u)$. Let $E(G)$ be the standard encoding of G in $n(n-1)/2$ bits as in Def. 2. We now give another way to describe G using some local routing function $F(u)$.

- A description of this discussion in $O(1)$ bits.
- A description of u in $\log n$ bits. (If it is less pad the description with 0's.)
- A description of the presence or absence of edges between u and the other nodes in V in $n-1$ bits.

- A description of $F(u)$ in $|F(u)| + O(\log |F(u)|)$ bits (the logarithmic term to make the description self-delimiting).
- The code $E(G)$ with all bits deleted corresponding to the presence or absence of edges between each w and v such that v is a neighbour of u and w is not a neighbour of u. Since there are at least $n/2 - o(n)$ nodes w such that $uw \notin E$ and at least $n/2 - o(n)$ nodes v such that $uv \in E$, by Lemma 1, this saves at least $(n/2 - o(n))^2$ bits.

From this description we can reconstruct G, given n, by reconstructing the bits corresponding to the deleted edges from u and $F(u)$ and subsequently inserting them in the appropriate positions to reconstruct $E(G)$. We can do so because $F(u)$ represents a full information routing scheme implying that $vw \in E$ iff uv is among the edges used to route from u to w. In total this new description has

$$n(n-1)/2 + O(\log n) + |F(u)| - n^2/4 + o(n^2)$$

which must be at least $n(n-1)/2 - o(n)$ by Eq. (1). We conclude that $|F(u)| = n^2/4 - o(n^2)$, which proves the theorem. □

6 Average routing

We now extend our results to the average cost, taken over all labelled graphs of n nodes, of representing a routing scheme for graphs over n nodes.

DEFINITION 5 For a graph G, let $T(G)$ be the number of bits used to store its routing scheme. The *average* total number of bits to store the routing scheme for routing over graphs on n nodes is $\sum T(G)/2^{n(n-1)/2}$ with the sum taken over all graphs G on nodes $\{1, 2, \ldots, n\}$. (That is, the uniform average over all the labelled graphs on n nodes.)

The results on Kolmogorov random graphs above have the following corollaries. Consider the subset of $(3 \log n)$-random graphs within the class of $O(\log n)$-random graphs on n nodes. They constitute a fraction of at least $(1 - 1/n^3)$ of the class of all graphs on n nodes. The trivial upper bound on the minimal total number of bits for all routing functions together is $O(n^2 \log n)$ for shortest path routing on all graphs on n nodes (or $O(n^3)$ for full-information shortest path routing). Simple computation of the average of the total number of bits used to store the routing scheme over all graphs on n nodes shows the following.

COROLLARY 1 *The average total number of bits to store the routing scheme for graphs of n nodes is:*

1. $O(n^2)$ *for shortest path routing in model IB* $\vee$ *II (Theorem 1),*
2. $O(n \log^2 n)$ *for shortest path routing in model II* $\wedge$ γ *(Theorem 2),*
3. $O(n \log n)$ *for routing with any stretch factor* s *for* $1 < s < 2$ *in model II (Theorem 3),*
4. $O(n \log \log n)$ *for routing with stretch factor 2 in model II (Theorem 4),*
5. $O(n)$ *for routing with stretch factor* $6 \log n$ *in model II (Theorem 5 with* $c = 3$*),*
6. $\Omega(n^2)$ *for shortest path routing in model* α*, IA and IB (Theorem 6 and Theorem 7),*
7. $\Omega(n^2 \log n)$ *for shortest path routing in model IA* $\wedge$ α *(Theorem 8),*
8. $\Theta(n^3)$ *for full information shortest path routing in model* α *(Theorem 10).*

Acknowledgements

We thank Jan van Leeuwen, Evangelos Kranakis and Danny Krizanc for helpful discussions.

References

[1] M. Flammini, J. van Leeuwen, and A. Marchetti-Spaccamela. The complexity of interval routing on random graphs. In *20th Int. Symp. on Mathematical Foundations of Computer Science*, Lect. Not. Comp. Sci. (to appear), 1995. Springer.

[2] Pierre Fraigniaud and Cyril Gavoille. Memory requirement for universal routing schemes. In *14th Ann. Symp. on Principles of Distributed Computing*, pages 223–230, Ottawa, Ont., Canada, August 20–23 1995. ACM Press.

[3] Cyril Gavoille and Stéphane Pérennès. Memory requirements for routing in distributed networks. In *15th Ann. Symp. on Principles of Distributed Computing*, Philadelphia, PA, USA, May 23–26 1996. ACM Press. (these proceedings).

[4] A. N. Kolmogorov. Three approaches to the quantitative definition of information. *Problems Inform. Transmission*, 1(1):1–7, 1965.

[5] E. Kranakis and D. Krizanc. Lower bounds for compact routing schemes. Manuscript, 1995.

[6] Evangelos Kranakis, Danny Krizanc, and Jorge Urrutia. Compact routing and shortest path information. In *Proc. 2nd International Colloquium on Structural Information and Communication Complexity.* Carleton University Press, 1995.

[7] Evangelos Kranakis, Danny Krizanc, and Jorge Urrutia. Lower bounds for compact shortest path routing schemes. Manuscript, 1995.

[8] Ming Li and Paul M. B. Vitányi. *An Introduction to Kolmogorov Complexity and its Applications.* Springer-Verlag, New-York, 1993.

[9] David Peleg and Eli Upfal. A trade-off between space and efficiency for routing tables. *Journal of the ACM*, 36(3):510–530, July 1989.

[10] H. S. Wilf. *Algorithms and Complexity.* Prentice-Hall, 1986.