



The compactness of adaptive routing tables

Cyril Gavoille*, Akka Zemmari

LaBRI, Université Bordeaux I, France

Abstract

The compactness of a routing table is a complexity measure of the memory space needed to store the routing table on a network whose nodes have been labelled by a consecutive range of integers. It is defined as the smallest integer k such that, in every node u , every set of labels of destinations having the same output in the table of u can be represented as the union of k intervals of consecutive labels. While many works studied the compactness of deterministic routing tables, few of them tackled the adaptive case when the output of the table, for each entry, must contain a fixed number α of routing directions. We prove that every n -node network supports shortest path routing tables of compactness at most n/α for an adaptiveness parameter α , whereas we show a lower bound of $n/\alpha^{O(1)}$.

© 2003 Elsevier B.V. All rights reserved.

Keywords: Compact routing tables; Adaptive routing; Interval routing

1. Introduction

1.1. Generalities

Given a parallel or distributed system, the interconnection network ensures the communication between the processors, the terminal nodes. Each intermediate node has a router, a dedicated co-processor which forwards the messages between processors through the links of the underlying topology. The routers run a distributed algorithm which specifies the way to go from a node of the network to another. This algorithm is described by a routing function.

Once a router receives a message, it looks at its header and checks the destination of the message, and finds the output port that will be used to forward the message towards to next intermediate node up to its destination. The output port is a number local to each router

* Corresponding author.

E-mail addresses: gavoille@labri.fr (C. Gavoille), zemmari@labri.fr (A. Zemmari).

and associated to each link between routers. A standard way to implement such algorithms is to use a *routing table*. To find the output port, the router consults a table which is kept in its local memory. For each destination, this table returns the output port number through which the message can be forwarded.

A simple method to organize this table is to associate to each destination the output port number which can serve it. This method is simple, but it is very memory expensive. It requires $O(n \log d)$ bits to maintain the routing table in each node of degree d , where n is the number of nodes of the underlying graph representing the network.

For a large or growing network, this method is not feasible. It is interesting to look for another method in order to reduce the size of the data structure stored by the routers, and used for the routing task. In the field of compact routing, several methods and strategies were introduced to reduce the router memory size, as separator-based routing schemes [13, 14], hierarchical routing schemes [2,25], prefix routing [3], Boolean routing [9], and interval routing [27,30]. We focus our work on the latter technique that offers a more compact data structure for routing tables.

1.2. The interval routing schemes

The interval routing was introduced by Santoro and Khatib in [27], and extended in [30] by van Leeuwen and Tan. It has been intensively studied in recent years, and an overview can be found in [16]. This method consists of finding a global labelling of the nodes with integers taken from $\{1, 2, \dots, n\}$, and, given a routing table, to group in the smallest set of intervals the destination labels using the same output port in each node. An interval means a set of consecutive integers, the labels 1 and n being considered as consecutive. If there exists a routing table such that each set of destination labels using a same output port can be grouped with at most k intervals, we deal with a k -interval routing scheme for this network, k -IRS for short. A k -IRS can be implemented with $O(kd \log n)$ bits per node by storing the interval boundaries of the destinations. Actually, this naive coding can be slightly compressed into $O(kd \log(n/k))$ bits [16]. In a sense, interval routing is a compact implementation of routing tables. One can hope to store only $O(k)$ integers per node for bounded degree networks using k -IRS, whereas standard routing tables require $O(n)$ integers. The parameter k is called *compactness* of a routing table.

Many works try to determine routing tables with minimum compactness under several assumptions on the quality of the routing measured in term of length of the routes: shortest path routing [12,17,18,20], stretched routing [4,11,24], routing with bounded dilation [6, 15,22,24,28], etc. (cf. [16]). Nevertheless, these works have studied only the deterministic case: for each source-destination pair, the routing table encodes only one routing path. So, the routing path is completely determined by giving the intervals. On the contrary, *adaptive* routing allows to diversify the routing paths. A destination can belongs to more than one set of intervals. For interval routing schemes, this extension has been partially suggested in [29].

1.3. Adaptive routing tables

More precisely, let us define an α -adaptive routing table as a routing table in which every destination can be founded in each router for exactly α different output ports, for some integer $\alpha \geq 1$. Similarly, an α -adaptive k -interval labelling scheme, or k -ILS $_{\alpha}$ for short, is an α -adaptive routing table for which the set of destination labels using the same output port can be grouped into at most k intervals. An ILS $_{\alpha}$ is termed *valid* if for every source-destination pair u, v , with¹ $u \neq v$, there exists in u (and all the other intermediate nodes) *at least one* output port among the α possible ones that induces a route to v . A valid ILS $_{\alpha}$ is called an α -adaptive interval routing scheme, IRS $_{\alpha}$ for short. Therefore, a k -IRS $_{\alpha}$ is simply an α -adaptive routing table of compactness k .

The definition captures adaptiveness of a routing, since at each step the router can select the next edge of the route among² α . This potentiality provides many routing paths, but not necessarily entirely disjoint paths that would require some strong assumptions on the edge-connectivity of the network. In this model some routes may loop. The router has the guarantee that at least one route connects to the destination. The other paths are called *deflecting paths*. They can be used depending on the load of the network, or on every other parameters, in order to improved the traffic. The case $\alpha = 1$ corresponds to the deterministic one (no deflecting paths).

Of course, in practice, for a complete implementation of a routing protocol, a *selection* function must choose one output port among the valid set. The adaptiveness of a routing table of compactness k implies to store in the router a total of $O(kd \log(n/k)) + |S|$ bits of routing information, where $|S|$ represents the number of bits needed to code the selection function S encoding the policy of the router. For instance, a kind of routing policy may consist to choose at random a permutation of the possible paths returned by the router if several³ messages come in the router at a same time (this occurs, for instance, when the messages cannot be stored locally due to physical constraints of the router). In this case $|S|$ is just the size of a pseudo-random generator. A selection function may also provide some priority ordering between the routing paths. In this case it requires to store extra bits, and $|S|$ might be large. In particular S must differentiate routing paths from deflecting paths. In all the cases, our approach consists in splitting the memory requirements of the router in two parts: one required by the routing tables (the term $O(kd \log(n/k))$), and the other part required by the selection function (the term $|S|$).

In this paper, we are not interested in the coding of the selecting function S , but rather in the parameter k , the compactness. This latter parameter depends on the graph topology only, whereas the coding of the selection function may depend on the strategy to optimize the traffic: the links can be chosen at random, or selected according to

¹ In the framework of compact routing a common assumption is that the destination of a message is never its source. The case $u = v$ can be solved by the local processor (assumed having a relatively high computational level) without any communication with its router. This allows to establish more flexible and deeper results in particular for space memory lower bounds.

² As we will see the degree of the node has to be at least α .

³ No more than α .

some load history tables of the links, or predicted from some other arbitrary policies (deadlock-free, ...). We observe that a space complexity measure that would combine both terms suffers of the general $\Omega(n \log d)$ bit/node lower bound (and an $\Omega(n)$ intervals for the compactness) that applies to shortest path deterministic routing tables [18, 19]. Indeed, as we will see more precisely in Section 4.3, an adaptive routing table and a selection function encode together at least a deterministic routing table. However, such a combination does not allow to measure precisely the contribution of each part (for instance, the $\Omega(n)$ -lower bound on the compactness [18] does not apply for shortest path α -adaptive routing tables, cf. Section 4). So, our approach allows to measure the balance between the information needed for the adaptive routing table and the selection function.

1.4. Related works

Previous works on compact and adaptive routing schemes can be founded in [1,2,9,21] for general schemes, and in [10,11,23,26] for interval routing schemes and its generalizations. However, most of these works try to give a compact representation of *all* the shortest paths. Although these schemes extend the deterministic case, they suffer by the fact that many general lower bounds for deterministic routing established in [12,18–21] apply as well for the adaptive case. Indeed, these lower bounds are based on the uniqueness of the shortest paths between specific subset of nodes in some worst-case graphs. Thus, on these graphs all-shortest-path routing would consist to route along one shortest path as in deterministic routing. In essence, all-shortest-path compact routing schemes are not more compact than deterministic shortest path routing schemes. For instance, the asymptotic $n/4$ -lower bound on the compactness for deterministic shortest path IRS applies also for all-shortest-path IRS [18].

1.5. Our results

As we will see in the following, the situation is better thanks to the definition we propose for α -adaptive routing tables (IRS_α), specially whenever $\alpha > 1$ and becomes larger. All previously cited lower bounds does not apply in that case, and moreover we show that n/α intervals per arc suffice for shortest path IRS_α that is already better than the deterministic case whenever $\alpha \geq 4$.

This paper is organized as follows. Section 2 defines more precisely the model of α -adaptive routing tables. In Section 3 we show that every routing tables can be transformed on an α -adaptive routing table with the same set of routes and the same compactness. In particular we show that n/α intervals per arc suffice, even if shortest paths are required. In Section 4 we specifically study more deeply shortest path routing tables, and we show an existential $n/\alpha^{O(1)}$ -lower bound for the compactness, that is asymptotically optimal for constant α . We conclude in Section 5 by some possible extensions and perspectives of this work.

2. Preliminaries

In this paper, the network is modeled by a connected graph $G = (V, E)$, whose set of nodes V represents the routers, and whose set of arcs E the communication links between the routers. We assume that the links are bi-directional, i.e., if $(u, v) \in E$ then $(v, u) \in E$; G is a symmetric digraph.⁴ For every $u \in V$, we denote by $\deg(u)$ the number of neighbors of u corresponding to the common value of in- and out-degree of u . Finally, $\delta(G)$ denotes the minimum degree of G , that is $\delta(G) = \min\{\deg(u) \mid u \in V\}$.

2.1. Definitions

Formally, an *interval labelling scheme* on an n -node G is a pair $(\mathcal{L}, \mathcal{I})$ of functions where $\mathcal{L}: V \rightarrow \{1, \dots, n\}$ is a one-to-one labelling of the nodes, and $\mathcal{I}: E \rightarrow 2^{\mathcal{L}(V)}$ is a labelling of the arcs such that, for every arc $(u, v) \in E$, $\mathcal{L}(w) \in \mathcal{I}(u, v)$ if and only if the route from u to w uses the arc (u, v) .

Moreover, given an integer $\alpha \geq 1$, the pair $(\mathcal{L}, \mathcal{I})$ is an α -*adaptive interval labelling scheme*, ILS_α for short, if for all $u, w \in V$, $w \neq u$, the set

$$\{(u, v) \in E \mid \mathcal{L}(w) \in \mathcal{I}(u, v)\}$$

is of cardinality α . A *valid* ILS_α is called an IRS_α (α -adaptive interval routing scheme or α -adaptive routing table), if it fulfills the connectivity condition: for all $u, w \in V$, $w \neq u$, there exists a sequence $\rho(u, w) = (v_1, \dots, v_t)$ of nodes such that $v_1 = u$ and $v_t = w$, and for every $i \in \{1, \dots, t-1\}$, $\mathcal{L}(w) \in \mathcal{I}(v_i, v_{i+1})$. The sequence $\rho(u, w)$ is called a *routing path* or *route* from u to w , and may not form a simple path in G .

A *shortest path* IRS_α is an IRS_α for which, for any pair u, w , there exists a routing path $\rho(u, w)$ that is a shortest path in G . This definition easily extends to weighted graphs considering paths of minimum cost. We insist on the fact that between u and w there is at least one routing path $\rho(u, w)$ that is a shortest path, although many routing paths might be represented by the labelling. As said before in Section 1.4, the main interest of this condition is to avoid the $n/4$ -lower bound of [18] on the compactness.

Remark. A consequence of the previous definition is that only the graphs of minimum degree at least α support an ILS_α , and thus an IRS_α . A variant of the previous definition to overcome this problem would consist to impose that

$$|\{(u, v) \in E \mid \mathcal{L}(w) \in \mathcal{I}(u, v)\}| = \min\{\alpha, \deg(u)\}.$$

Although all the results we propose in this paper hold for both definitions, for simplicity, only the former definition is considered in the sequel.

⁴ However, many of the results presented in this paper are still valid for nonsymmetric and strongly connected digraphs.

2.2. Compactness

The *compactness* of an $\text{ILS}_\alpha(\mathcal{L}, \mathcal{I})$ is the smallest integer k such that every set $\mathcal{I}(u, v)$ can be represented as the union of at most k intervals of consecutive integers of $\{1, 2, \dots, n\}$ (1 and n being considered as consecutive). Such ILS_α and IRS_α are denoted respectively by $k\text{-ILS}_\alpha$ and $k\text{-IRS}_\alpha$.

Remark. For $\alpha = 1$, all the definitions match with the standard ILS/IRS introduced by [27, 30]. For simplicity, we denote in the sequel IRS for IRS_1 . The labellings we consider in this paper are supposed to be *strict*, i.e., we impose that $\mathcal{L}(u) \notin \mathcal{I}(u, v)$, for every $(u, v) \in E$.

3. A general labelling scheme

We show in this section that every graph G supports a 1-IRS_α for every $\alpha \leq \delta(G)$, the routing paths being not necessary shortest paths. This result can be seen as a generalization of the labelling scheme of [27] (showing that every graph has a 1-IRS), and will be a tool for the remaining of the paper. We denote by $[1, n]$ the set $\{1, 2, \dots, n\}$.

Theorem 3.1. *Let $(\mathcal{L}, \mathcal{I}_A)$ be any $k\text{-IRS}_\alpha$ on an n -node graph $G = (V, E)$ with $\delta(G) \geq \alpha + 1$, and let $Y \subseteq E$ such that every $x \in V$ has at most one neighbor y so that $(x, y) \in Y$. Then, $(\mathcal{L}, \mathcal{I}_A)$ can be transformed in polynomial time into a $k\text{-IRS}_{\alpha+1}$ on G , $(\mathcal{L}, \mathcal{I}_B)$, such that all the routes represented by $(\mathcal{L}, \mathcal{I}_A)$ are preserved in $(\mathcal{L}, \mathcal{I}_B)$, and such that for every $(x, y) \in Y$, $\mathcal{I}_B(x, y) = [1, n] \setminus \{\mathcal{L}(x)\}$.*

Proof. For every $z \in [1, n]$, let us denote $\text{succ}(z)$ (respectively $\text{pred}(z)$) the successor (respectively predecessor) of z in $[1, n]$ modulo n . Formally, $\text{succ}(z) = (z \bmod n) + 1$, and $\text{pred}(z) = (z + n - 2 \bmod n) + 1$. Let us define the following procedure of inputs $(\mathcal{L}, \mathcal{I}_A)$ and Y , and of output $(\mathcal{L}, \mathcal{I}_B)$ satisfying the statement of Theorem 3.1.

For every node x do (possibly in parallel):

- (1) For every $(x, y) \in E$, set $\mathcal{I}_B(x, y) \leftarrow \mathcal{I}_A(x, y)$.
- (2) Set $R \leftarrow [1, n] \setminus \{\mathcal{L}(x)\}$.
- (3) Let (x, y) be the unique arc of Y (if y does not exist go to 4), set $\mathcal{I}_B(x, y) \leftarrow R$, and update $R \leftarrow \mathcal{I}_A(x, y)$.
- (4) While $R \neq \emptyset$ do:
 - (a) Find y and z such that $(x, y) \in E$, $z \in R \setminus \mathcal{I}_B(x, y)$, and either $\text{pred}(z) \in \mathcal{I}_B(x, y)$ or $\text{succ}(z) \in \mathcal{I}_B(x, y)$.
 - (b) Find y' such that $(x, y') \in E$, and $z \in \mathcal{I}_B(x, y')$. Let $[a, b]$ an interval such that $z \in [a, b] \subseteq \mathcal{I}_A(x, y')$.
 - (c) Update $R \leftarrow R \setminus ([a, b] \setminus \mathcal{I}_B(x, y))$.
 - (d) Update $\mathcal{I}_B(x, y) \leftarrow \mathcal{I}_B(x, y) \cup [a, b]$.

Intuitively, the procedure consists on finding a label $z \in R$ such that its predecessor (or successor) is a boundary of some intervals of $\mathcal{I}_B(x, y)$. Then we append $[a, b]$, an interval

containing z , to $\mathcal{I}_B(x, y)$ solving the problem for z (at least). The procedure iterates on the updated version of R .

Let us consider any node x . Let us show that for every i , at the beginning of the i th run of Instruction 4 (at the test $R \neq \emptyset$), the set R fulfills the following property P_i : R contains at most $n - i$ labels, and if $z \in R$ then z appears in α sets \mathcal{I}_B , and otherwise $z = \mathcal{L}(x)$ or z appears in $\alpha + 1$ sets \mathcal{I}_B . In other words, at each loop, R denotes the set of labels that remains to treat.

By induction on i : the first time in Instruction 4, if no arc $(x, y) \in Y$ exists, R is the set of all the labels (except for $\mathcal{L}(x)$), and \mathcal{I}_B is initialized to \mathcal{I}_A . Hence, if there is no arc $(x, y) \in Y$, P_1 is true. Otherwise, after Instruction 3, all labels remaining in R appear exactly in α sets \mathcal{I}_B (the others appear already in $\alpha + 1$ sets by setting $\mathcal{I}_B(x, y) = [1, n] \setminus \{\mathcal{L}(x)\}$). Hence in any cases P_1 is true.

Now, assume the property holds up to the i th loop. To show that P_{i+1} is true, let us first show that Instruction 4(a) is doable, that is the pair (y, z) can be founded: first, if $i = 1$, then it suffices to choose any y such that $(x, y) \notin Y$ and $\mathcal{I}_A(x, y) \neq R$ (it must exist otherwise every label $\neq \mathcal{L}(x)$ would appear in at least $\delta(G) \geq \alpha + 1$ sets \mathcal{I}_A). Then, we can choose any $z \notin \mathcal{I}_A(x, y)$ (thus $z \in R$) so that $\text{pred}(z) \in \mathcal{I}_A(x, y)$ or $\text{succ}(z) \in \mathcal{I}_A(x, y)$.

For $i > 1$, a pair (y, z) exists otherwise, z and $\text{pred}(z)$ (or $\text{succ}(z)$) would appear in the same number of sets \mathcal{I}_B . By Property P_i , $z \in R$ implies $\text{pred}(z)$ or $\text{succ}(z) \in R$ (otherwise they would not appear in the same number of sets \mathcal{I}_B). This implies that $R = [1, n] \setminus \{\mathcal{L}(x)\}$, which is not possible since $|R| \leq n - i < n - 1$ ($i > 1$). So, Instruction 4(a) is doable. Instruction 4(b) is doable since $z \in R$ and by Property P_i z appears in $\alpha \geq 1$ sets \mathcal{I}_B . Instructions 4(c) and 4(d) are doable as well. We remark, that in Instruction 4(c), $|R|$ decreases by at least one element: $[a, b]$ contains at least z . We check that all labels removed from R appears in exactly $\alpha + 1$ sets \mathcal{I}_B . Therefore, P_{i+1} holds.

So, at the end of the last loop ℓ , R is empty and by Property P_ℓ , all the labels appear in $\alpha + 1$ sets \mathcal{I}_B . Taking a union in Instruction 4(d), we guarantee that $\mathcal{I}_A(x, y) \subseteq \mathcal{I}_B(x, y)$, and thus it preserves the routes. It follows that $(\mathcal{L}, \mathcal{I}_B)$ is a valid $\text{ILS}_{\alpha+1}$. Moreover, in Instruction 4(d), because $\text{pred}(z)$ and z are consecutive modulo n , and because $z \in \mathcal{I}_A(x, y)$ and $\text{pred}(z) \in \mathcal{I}_A(x, y)$, we have that the minimum number of intervals to represent $\mathcal{I}_B(x, y)$ never increase and thus is at most the one of $\mathcal{I}_A(x, y)$. So, $(\mathcal{L}, \mathcal{I}_B)$ has compactness at most k , and by Instruction 3, all the arcs of Y have the interval $[1, n] \setminus \{\mathcal{L}(x)\}$. This completes the proof. \square

Remark. We do not precise the time complexity of the previous algorithm because it may depend on the data structure used to code the input IRS (the one achieving the lowest time complexity is not necessary the most compact one). Anyway, using naive interval coding representation of IRS, this time is less than $O(n^4)$, but can easily be reduced to $O(|E|k\alpha)$ with more efficient data structures.

Using a spanning tree T of G , a DFS-based 1-IRS₁ on T (cf. [27]), and applying inductively on α in Theorem 3.1 we have:

Corollary 3.2. *Every graph G such that $\delta(G) \geq \alpha$, supports a 1-IRS_α .*

Whereas for $\alpha = 1$ every graph has a $k\text{-IRS}$ with $k \leq n/2$, for $\alpha > 1$ we show that $k \leq n/\alpha$. More precisely:

Theorem 3.3. *Let $(\mathcal{L}, \mathcal{I})$ be any $k\text{-IRS}_1$ on an n -node graph G , and let $\alpha \leq \delta(G)$. Then, G supports a $k'\text{-IRS}_\alpha$ such that all the routes of $(\mathcal{L}, \mathcal{I})$ are preserved, and such that $k' \leq \min\{k, (n-1)/\alpha\}$.*

Proof. The statement is obvious for $\alpha = 1$. Assume, $\alpha \geq 2$. We build a set Y composed of the arcs assigned with the largest number of intervals, for each node. After the first application of Theorem 3.1, we obtain a $k\text{-IRS}_2$ for G , with the same set of routes, and where the arc with the largest number of intervals (for each node) is now reduced to one. We can re-apply Theorem 3.1 with a new set Y still composed of the arcs assigned with the largest number of intervals, which is hence at most the second largest one in $(\mathcal{L}, \mathcal{I})$. Finally, after a total of $\alpha - 1$ applications of Theorem 3.1 (this is feasible since $\alpha \leq \delta(G)$), we obtain a $k'\text{-IRS}_\alpha$ with the same set of routes where the maximum number of intervals assigned on an arc, k' , is bounded by k and also by the α th largest number of intervals assigned on an arc in $(\mathcal{L}, \mathcal{I})$.

Let x be any node of G . Let $d = \deg(x)$, and let k_1, \dots, k_d be the number of intervals of the sets $\mathcal{I}(x, y)$ (the $k\text{-IRS}_1$ defined on G) for all neighbors y of x . Moreover assume $k_1 \geq \dots \geq k_d$. We have $\sum_{i=1}^{\alpha} k_i \leq \sum_{i=1}^d k_i \leq n - 1$ ($\alpha \leq d$ and the label of x is not assigned). Thus $k_\alpha \leq (n - 1)/\alpha$. As said before, $k' \leq \min\{k, k_\alpha\}$ completing the proof.

Remark. Theorem 3.1 can be slightly improved to

$$k' \leq \min\{k, (n - 1 - \delta(G))/\alpha + 1\}$$

if all the incident arcs of each node are labelled with non-empty labels (in this case we have $\sum_{i=1}^{\alpha} k_i \leq n - 1 - (\delta(G) - \alpha)$). This assumption occurs, for instance, for shortest paths routing tables.

4. Shortest path labelling

In this section we are interested in IRS_α for which there exists at least one shortest path (represented by the labelling schemes) for all pairs of nodes. Thanks to Theorem 3.1, many graphs can be identified to support shortest path $k\text{-IRS}_\alpha$. For instance, grid, hypercube, complete graph, cycle, trees, outerplanar graphs, interval graphs, etc., have shortest path 1-IRS , and thus also shortest path 1-IRS_α . Families of graphs having shortest path $O(1)\text{-IRS}$ include torus, k -trees with constant k , planar graphs with a constant number of faces, etc. (see [16] for a complete state of the art).

For every graph G , we define

$$\text{IRS}_\alpha(G) = \min\{k \mid G \text{ has a shortest path } k\text{-IRS}_\alpha\}.$$

Note that the computation of $IRS(G)$ (for $\alpha = 1$) already involves several difficult optimizations. The decision problems “is $IRS(G) = 1$?” and “is $IRS(G) = 2$?” are NP-complete [5,7]. Hereafter, the value $IRS_\alpha(G)$ is termed *compactness $_\alpha$* of G .

4.1. Comparison between compactness $_1$ and compactness $_\alpha$

By Theorem 3.1, we have $IRS_{\alpha+1}(G) \leq IRS_\alpha(G) \leq \dots \leq IRS(G)$. It is not a difficult exercise to check that there are graphs that support shortest path 1- IRS_2 , whereas they do not support shortest path 1- IRS_1 (for instance, consider the Petersen graph [17], or the wheel-graph [8]). The next result shows that the difference between the compactness of 1- and α -adaptive routing of a graph can be exponentially large.

Theorem 4.1. For every integer $\delta \geq 0$, there exists a graph G on $2^{\delta+3}$ nodes such that $IRS_1(G) \geq 2^\delta$ and $IRS_2(G) \leq 2\delta + 4$.

Proof. We use the construction given in [17] that shows that $IRS_1(G) \geq n/8$ for some n -node graphs with n a power of two. Here we recall their construction.

For a $p \times q$ Boolean matrix $M = (M_{i,j})$, let $G_M = (V_M, E_M)$ be the graph such that:

- (1) $V_M = \{v_1, \dots, v_p\} \cup \{a_1, \dots, a_q\} \cup \{b_1, \dots, b_q\}$;
- (2) $\{x, y\} \in E_M$ if and only if $(x = a_j$ and $y = b_j)$, $(x = b_j$ and $y = v_i$ and $M_{i,j} = 1)$, or $(x = a_j$ and $y = v_i$ and $M_{i,j} = 0)$.

We have $n = |V_M| = p + 2q$. Roughly speaking, G_M is a two-level graph. The first level consists of edges of type $\{a_j, b_j\}$, and the second one consists of v_i s (a stable) which are connected to a_j or b_j depending on whether $M_{i,j} = 0$ or 1. See Fig. 1 for an example.

For every Boolean matrix M , we denote by \bar{M} the matrix M with every bit complemented. Moreover, if $M = (XY)$, where X and Y are two matrices of same dimensions, we set $\chi(M) = (YX)$, which is the matrix obtained from M by exchanging the columns of X with those of Y . We consider a specific matrix M_δ , $\delta \geq 0$, defined by induction. The

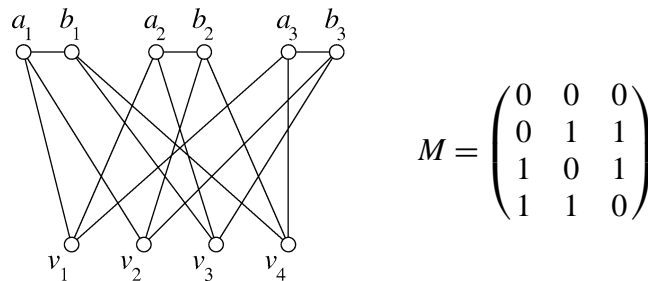


Fig. 1. A graph G_M .

construction of M_δ is summarized by Eq. (4.1).

$$M_0 = \begin{pmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 1 \\ 1 & 0 \end{pmatrix}, \quad \chi(M_0) = \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{pmatrix}, \quad M_{\delta+1} = \begin{pmatrix} M_\delta & M_\delta \\ \chi(M_\delta) & \chi(M_\delta) \end{pmatrix}. \quad (4.1)$$

It is shown in [17] that $\text{IRS}(G_{M_\delta}) \geq 2^\delta$ (roughly speaking, whatever the labelling of the v_i s, the set $\mathcal{I}(a_j, b_j)$ must contain only a particular subset of the v_i s which is made to be hard to represent with intervals). In this case $p = 2^{\delta+2}$ and $q = 2^{\delta+1}$. Thus $n = 2^{\delta+3}$, proving the first part of Theorem 4.1.

Let us show that $\text{IRS}_2(G_{M_\delta}) \leq 2\delta + 4$ for every $\delta \geq 0$. For this purpose, it suffices to define a shortest path IRS_2 on G_{M_δ} , $(\mathcal{L}, \mathcal{I})$, such that for all the arcs $(u, v) \notin \{(a_j, b_j), (b_j, a_j)\}$, $|\mathcal{I}(u, v)| \leq 2\delta + 4$, thus composed of at most $2\delta + 4$ intervals. Indeed, by Theorem 3.1, such a labelling can be transformed into an IRS_2 with the same set of routes such that the edges $\{a_j, b_j\}$ consist of one interval. Therefore it would prove that $\text{IRS}_2(G_{M_\delta}) \leq 2\delta + 4$.

In this proof, we do not optimize the node-labelling, leaving a small space to improve the bound on $\text{IRS}_2(G_{M_\delta})$. Let us choose an arbitrary labelling \mathcal{L} . Since we do not care about the number of intervals on the edges $\{a_j, b_j\}$, let us define B_{M_δ} be the graph G_{M_δ} where the all the edges $\{a_j, b_j\}$ have been removed. First, remark that B_{M_δ} is a $2^{\delta+1}$ -regular bipartite graph. Clearly, B_M is isomorphic to every graph $B_{M'}$, where M' is obtained by complementing some columns of M (this morphism exchanges the roles playing by some a_j s and b_j s), or by permuting some columns (this morphism permutes some edges $\{a_j, b_j\}$). So, for the sake of simplicity, let us set B_δ to be the common graph isomorphic to B_{M_δ} , $B_{\chi(M_\delta)}$, etc. Let $V_1(B_\delta)$ to be the set of the first partition of nodes of B_δ , the a_j s and b_j s, and $V_2(B_\delta)$ as the nodes v_i s of B_δ .

Let us define by induction on δ , $(\mathcal{L}, \mathcal{I})$ on B_δ . $B_{\delta+1}$ consists of two copies of B_δ , B_1 and B_2 , with some extra edges connecting $V_1(B_1)$ to $V_2(B_2)$, and some edges connecting $V_1(B_2)$ to $V_2(B_1)$. Let ψ be the morphism between $V(B_1)$ and $V(B_2)$, and let ϕ_i be the morphism between $V_i(B_1)$ and $V_i(B_2)$ for $i = 1, 2$. For $\delta = 0$, we check that one can find \mathcal{I} such that $|\mathcal{I}(u, v)| \leq 4$ for all arcs (u, v) of B_0 . Let $k = \max |\mathcal{I}(u, v)|$, over all arcs (u, v) of B_δ . Since we do not care about \mathcal{L} , we consider $\mathcal{I}(u, v)$ as a subset of nodes rather than a subset of labels.

We first look at any node $v \in V_2(B_1)$. By induction, assume that $|\mathcal{I}(v, a)| \leq k$ for all $(v, a) \in E(B_1)$. We remark that if $(v, a) \in E(B_1)$, then $(v, \phi_1(a)) \in E(B_{\delta+1}) \setminus E(B_1)$. Setting $\mathcal{I}(v, \phi_1(a)) = \psi(\mathcal{I}(v, a))$ for all $(v, a) \in E(B_1)$, we are able to route from $v \in V_2(B_1)$ to all the nodes of $V(B_{\delta+1}) \setminus \{\phi_2(v)\}$. We add $\phi_2(v)$ to any arc incident of v leading to $\phi_2(v)$ by a shortest path. One can check that the routes are still the shortest, and since the edges (v, a) and $(v, \phi_1(a))$ are distinct, $|\mathcal{I}(v, a)| \leq k + 1$ for all $(v, a) \in E(B_{\delta+1})$.

Then, let us look at any node $a \in V_1(B_1)$. With a similar argument, we can set $\mathcal{I}(a, \phi_2(v)) = \psi(\mathcal{I}(a, v))$ for all $(v, a) \in E(B_1)$. We are able to route from $a \in V_1(B_1)$ to $V(B_{\delta+1}) \setminus \{\phi_1(v), \phi_1(\bar{a})\}$, where \bar{a} is the unique node of $V_1(B_1)$ such that $\{a, \bar{a}\}$ is an edge of $E(G_{M_\delta})$. We add $\phi_1(a)$ and $\phi_1(\bar{a})$ to any arc incident of a allowing shortest route from a . So, $|\mathcal{I}(a, v)| \leq k + 2$ for all $(a, v) \in E(B_{\delta+1})$.

The routing from any $v \in V_2(B_2)$ and any from $V_1(B_2)$ is defined similarly since the graph $B_{\delta+1}$ is the same if B_1 and B_2 are exchanged. In total, for every arc $(u, v) \in E(B_{\delta+1})$, we have $|\mathcal{I}(u, v)| \leq k + 2$, that is at most $2\delta + 4$ for B_δ .

We complete the proof by Theorem 3.1 applied on the edges $\{a_j, b_j\}$. \square

4.2. An upper bound for compactness $_\alpha$

In this section we show that compactness $_\alpha$ of a general n -node graph is not bounded for $\alpha > 1$. Note that for $\alpha = 1$, a tight lower bound exists. It has been shown in [18] that for every graph G , $\text{IRS}(G) \leq n/4 + o(n)$, whereas there exists a worst-case graph G_0 with $\text{IRS}(G_0) \geq n/4 - o(n)$. We first present a general upper bound:

Theorem 4.2. *For every n -node graph G and every $\alpha \leq \delta(G)$,*

$$\text{IRS}_\alpha(G) \leq \frac{1}{\alpha}(n - 1 - \delta(G)) + 1.$$

Proof. It suffices to consider any shortest path k -IRS $_1$ for G (for instance, choosing $k \leq n/4 + o(n)$), and to apply Theorem 3.3 remarking that all the arcs have non-empty labels.

4.3. A lower bound for compactness $_\alpha$

We will show that there exists some worst-case graphs with compactness $_\alpha$ at least $n/\alpha^{O(1)}$. Therefore this shows an asymptotic optimal lower bound for the compactness of shortest path IRS $_\alpha$ with constant α . It is quite complicated to build “by hand” small counter-example G with, for instance, $\text{IRS}_2(G) > 1$. Indeed, we need to argue for such G , that whatever is the node-labelling, whatever are the shortest paths, and mainly, whatever are the deflecting paths, one cannot code the routing table with one interval. The first counter-example with $\text{IRS}_2(G) > 1$ that we are able to build (we will not draw it here) has roughly 10^5 nodes. That is why we present in this paper an existential lower bound only, holding also for unbounded α . We will mainly use the fact that any shortest path α -adaptive routing table combined with a suitable selection function S implements a standard routing table ($\alpha = 1$). So, up to an additive term of $|S|$ one can lower bound the compactness of the α -adaptive routing table thanks to the $\Omega(n \log d)$ bit/node lower bound of [19].

For this purpose, let us present the graph $H_{p,\delta}$ introduced by [19], and defined inductively on p for all integers $p \geq 1$ and $\delta \geq 2$. Let $T_{h,\delta}^i$ be a complete δ -ary tree of height h whose all its leaves are labelled i . For $h = 0$, we set $T_{0,\delta}^i$ as a tree composed of a single node labelled i . For every $m \geq 2$, we define $T_{p,\delta,m}$ as the tree composed of m trees $T_{p-1,\delta}^1, T_{p-1,\delta}^2, \dots, T_{p-1,\delta}^m$, all connected by their root to a single node of degree m . This node is labelled $p + 1$ and forms the root of $T_{p,\delta,m}$. Note that for $m = \delta$, a $T_{p,\delta,\delta}$ tree is isomorphic to a complete δ -ary tree of height p , and thus has δ^p leaves.

$H_{p,\delta}$ has two distinguished subsets of nodes: $A_p = \{1, \dots, p\}$ and $B_p = \{1, \dots, \delta\}^p$ the set of all the words of length p on the alphabet $\{1, \dots, \delta\}$. $H_{1,\delta}$ is isomorphic to $K_{1,\delta}$, where $A_1 = \{1\}$ is reduced to the unique node of degree δ in $K_{1,\delta}$, and where $B_1 = \{1, \dots, \delta\}$ is the set of nodes of degree 1. The $H_{p+1,\delta}$ graph is composed of a copy of $H_{p,\delta}$, a copy of $T_{p,\delta,\delta}$, and of the set of nodes B_{p+1} , connected as follows:

- (1) Every node $u \in B_p$ is connected to the nodes $u i \in B_{p+1}$, for every $i \in \{1, \dots, \delta\}$;
- (2) Every leaf of $T_{p,\delta,\delta}$ labelled i is connected to exactly δ nodes $u i \in B_{p+1}$ such that no two leaves are connected to the same node of B_{p+1} (leaving some freedom in the connections).

The set A_{p+1} is composed of the set A_p of $H_{p,\delta}$, and of the root of $T_{p,\delta,\delta}$. See Fig. 2.

For every integer m such that $2 \leq m \leq \delta$, let us define the $H_{p,\delta,m}$ graph composed of a $H_{p,\delta}$ graph, a $T_{p,\delta,m}$ tree, and a set of $m\delta^p$ nodes, $B_p^m = \{1, \dots, \delta\}^p \times \{1, \dots, m\}$. The connections between B_p , B_p^m , and the leaves of $T_{p,\delta,m}$ are similar to the connections in a $H_{p+1,\delta}$ graph excepted that m may be smaller than δ (every $u \in B_p$ is connected to $u i \in B_p^m$ for every $i \in \{1, \dots, m\}$). The $H_{p,\delta,m}$ graph is an induced subgraph of $H_{p+1,\delta}$. Let us denote by A_p^m the set of nodes composed of A_p and of the root of the $T_{p,\delta,m}$ tree.

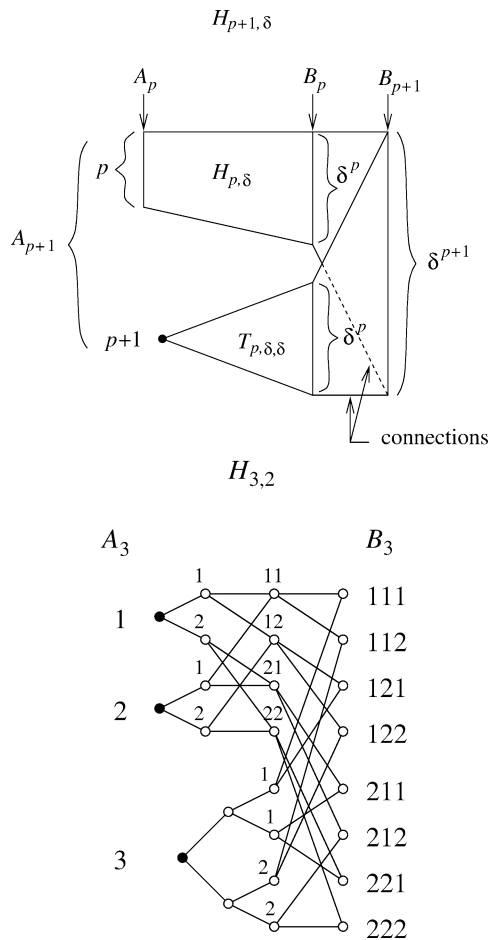


Fig. 2. The recursive construction of the $H_{p+1,\delta}$ graph, and $H_{3,2}$.

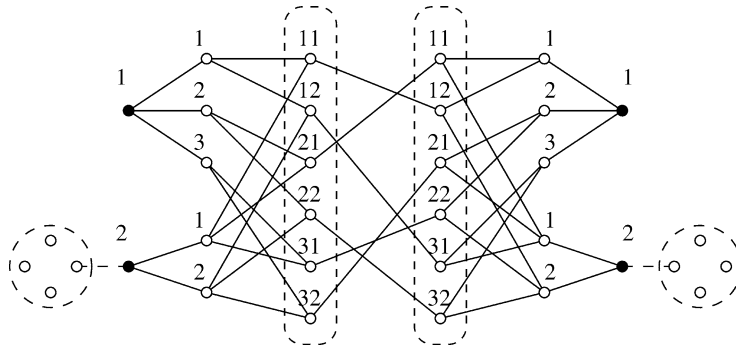


Fig. 3. A graph $G_\pi \in \mathcal{H}_{1,3,2}$, and its modification H_0 : the dashed sets induced a clique in H_0 and are missing from G_π .

Lemma 4.3. For all integers $p \geq 1$, and $\delta, m \geq 2$, $H_{p,\delta,m}$ has at most $2(m+2)\delta^p$ nodes, and $\deg(x) = 2$ if $x \in B_p^m$, $\deg(x) = m$ if x is the root of $T_{p,\delta,m}$, and $\deg(x) \geq \delta$ otherwise.

Given a permutation π of B_p^m , let us denote by G_π the graph composed of two copies of $H_{p,\delta,m}$ whose B_p^m sets are connected by the perfect matching defined by π (see Fig. 3, for an example). Let $\mathcal{H}_{p,\delta,m}$ denote the family composed of all the G_π graphs, for all permutations π of B_p^m . For each $G \in \mathcal{H}_{p,\delta,m}$, we denote by $A(G)$ (respectively $B(G)$) the set of nodes composed of both sets A_p^m (respectively B_p^m) of each copy of $H_{p,\delta,m}$ forming G . The nodes of $A(G)$ are drawn in black on Fig. 3.

In [19], it is shown the following important lemma:

Lemma 4.4 (Gavoille and Perennes [19]). For all integers $p, \delta, m \geq 2$, $m \leq \delta$, and such that $\delta^p \rightarrow +\infty$, there exists a graph $G_0 \in \mathcal{H}_{p,\delta,m}$ such that every shortest path routing table on G_0 has a size of M bits⁵ for a node of $A(G_0)$ such that

$$M \geq \frac{m\delta^p}{2(p+1)} \log(m\delta^p) - O\left(\frac{m\delta^p}{p}\right).$$

This result is based on the uniqueness of the shortest paths between the nodes of $A(G_0)$ and the nodes of $B(G_0)$. In order to prove our result, one transform G_0 into a new graph H_0 such that $\delta(H_0) \geq \delta$, and such that Lemma 4.4 holds for H_0 as well. It consists on connecting all the nodes of B_p^m by a clique in each copy of $H_{p,\delta,m}$ (so making the degree of the nodes of $B(G_0)$ larger than δ in H_0). Then, for the root of both $T_{p,\delta,m}$ trees, we add a clique of $\delta + 1$ nodes and select from them $\delta - m$ nodes that we connect to the root (so making the degree of nodes at least δ , and exactly δ for all the nodes of $A(G_0)$ in H_0). See Fig. 3. In H_0 , the shortest paths between $A(G_0)$ and $B(G_0)$ are not modified, and has $2(\delta + 1)$ more nodes than G_0 .

We are now ready to prove a lower bound on compactness $_\alpha$ of n -node graphs.

⁵ We assume that all logarithms are in base two.

Theorem 4.5. *There exist a constant $c \leq 31$ such that for every n large enough, and for every integer $\alpha \leq (n/18)^{1/(2c)}$, there exists a graph H_0 with at most n nodes such that*

$$\text{IRS}_\alpha(H_0) > \frac{1}{2790} \cdot \frac{n}{\alpha^c}.$$

Proof. From Theorem 4.1, for every n , there exists a graph G with $2^{t+3} \leq n$ nodes and such that $\text{IRS}_1(G) \geq 2^t > n/16$ for $t = \lfloor \log n \rfloor - 3$. Thus the result is true for $\alpha = 1$.

Assume $\alpha \geq 2$. Let us fix $c = 31$, and let $\delta = \alpha^c$. We consider the graph H_0 , the modified graph $G_0 \in \mathcal{H}_{p,\delta,m}$, for some parameters $p, \delta, m \geq 2$ such that $m \leq \delta$ and $\delta^p \rightarrow +\infty$. Let N denote the number of nodes of H_0 . We will fix later the values of p, δ, m as a function of n in order to prove that $N \leq n$. Let $x \in A(G_0)$ be a node of H_0 for which the size of any shortest path routing table is of size at least M (bound given by Lemma 4.4). Note that by construction of H_0 , $\text{deg}(x) = \delta$.

Consider on H_0 any shortest path k - IRS_α , $(\mathcal{L}, \mathcal{I})$. This α -adaptive IRS is an implementation in x of a particular shortest path α -adaptive routing table. This implementation can be done in x with at most $\lceil \delta \log \binom{N}{2k} \rceil$ bits. Indeed, for the δ output ports of x it suffices to store at most k intervals of labels. There is at most $\binom{N}{2k}$ ways to choose k sub-intervals of $[1, N]$. So, a total of $\lceil \delta \log \binom{N}{2k} \rceil$ bits for x , remarking that a sequence of p integers taken from $\{1, \dots, q\}$ can be coded on $\lceil \log(q^p) \rceil$ bits since there are q^p such sequences.

Now, it is easy to transform any shortest path α -adaptive routing tables into a shortest path 1-adaptive routing table, i.e., a standard routing table, adding $\lceil N \log \alpha \rceil$ extra bits per node: for each destination label we specify the output port leading to a shortest path, and there are exactly α possible output ports. Thus, H_0 has a shortest path routing table in x of size at most

$$\left\lceil \delta \log \binom{N}{2k} \right\rceil + \lceil N \log \alpha \rceil$$

using an implementation of the deterministic version of $(\mathcal{L}, \mathcal{I})$. From Lemma 4.4, it turns out for x that:

$$\delta \log \binom{N}{2k} + N \log \alpha + 2 > M. \tag{4.2}$$

We have to prove that H_0 has at most n nodes and that $k \geq n/\alpha^{O(1)}$. Let us fix now p, δ, m , and let us prove that:

$$N \leq n < 4(m + 3 + o(1))\delta^p. \tag{4.3}$$

Let p be the largest integer such that $n \geq 16\delta^p + 2(\delta + 1)$. Clearly, $\delta^p \rightarrow +\infty$ as $n \rightarrow +\infty$. Let $m = \lfloor (n - 2(\delta + 1))/(4\delta^p) \rfloor - 2$. Let us show that p, δ , and m are all greater than 2.

First, $\delta \geq 2$ because $\delta = \alpha^c$, and $\alpha, c \geq 2$. For $p, \alpha \leq (n/18)^{1/(2c)}$ implies $18(\alpha^c)^2 \leq n$, i.e., $n \geq 18\delta^2$. But $18\delta^2 \geq 16\delta^2 + 2(\delta + 1)$ for $\delta \geq 2$, hence the equation $n \geq 16\delta^p + 2(\delta + 1)$ has a solution for $p \geq 2$. For m , since $n \geq 16\delta^p + 2(\delta + 1)$, then $(n - 2(\delta + 1))/(4\delta^p) \geq 4$, and thus $\lfloor (n - 2(\delta + 1))/(4\delta^p) \rfloor - 2 \geq 2$, proving $m \geq 2$.

Let $X = 4(m + 2)\delta^p + 2(\delta + 1)$, and let $Y = n - 2(\delta + 1)$. Note that $m + 2 = \lfloor Y/(4\delta^p) \rfloor$. We have:

$$\begin{aligned} X &= 4\delta^p \left\lfloor \frac{Y}{4\delta^p} \right\rfloor + 2(\delta + 1) \\ &= Y - (Y \bmod 4\delta^p) + 2(\delta + 1) \\ &= n - (Y \bmod 4\delta^p). \end{aligned}$$

Therefore,

$$X \leq n < X + 4\delta^p. \tag{4.4}$$

From Lemma 4.3, the number of nodes of G_0 is at most $2 \cdot 2(m + 2)\delta^p$, and thus the number of nodes of H_0 is $N \leq 4(m + 2)\delta^p + 2(\delta + 1)$, i.e., $N \leq X$. By Eq. (4.4) we have proved that H_0 has at most n nodes, and more precisely that:

$$\begin{aligned} N &\leq n < 4(m + 2)\delta^p + 4\delta^p + 2(\delta + 1) \\ &< 4(m + 3 + o(1))\delta^p \end{aligned}$$

remarking that $2(\delta + 1) = o(\delta^p)$, and proving therefore Eq. (4.3).

In Eq. (4.2), we bound

$$\binom{N}{2k} \leq \binom{n}{2k} \leq \left(\frac{en}{2k}\right)^{2k} \leq t^{en/t},$$

where $t = en/(2k)$. Eq. (4.2) becomes (using $\delta = \alpha^c$, $N \leq n$, and $M \rightarrow +\infty$)

$$\delta \log(t^{en/t}) + n \log \alpha \geq M \tag{4.5}$$

$$\implies \alpha^c en \frac{\log t}{t} + n \log \alpha \geq M \tag{4.6}$$

$$\implies n \left(\frac{\alpha^c e \log t}{t} + \log \alpha \right) \geq M \tag{4.7}$$

$$\implies n\beta \geq M, \tag{4.8}$$

where $\beta = (\alpha^c e \log t)/t + \log \alpha$. From Lemma 4.4 we have a lower bound on M , and plugging in Eq. (4.8) the upper bound on n of Eq. (4.3), we obtain that:

$$\begin{aligned} 4(m + 3 + o(1))\delta^p \beta &> \frac{m\delta^p}{2(p + 1)} \log(m\delta^p) - O\left(\frac{m\delta^p}{p}\right) \\ \implies \beta &> \frac{m \log(m\delta^p)}{8(p + 1)(m + 3 + o(1))} \end{aligned}$$

neglecting the second order term $O(m\delta^p/p)$. We remark that $p, m \geq 2$, thus

$$\begin{aligned} \frac{m \log(m\delta^p)}{8(p + 1)(m + 3 + o(1))} &\geq \frac{2 \log(2\delta^p)}{8(p + 1)(5 + o(1))} > \frac{p \log \delta}{4(p + 1)(5 + o(1))} \\ &> \frac{2 \log \delta}{4 \cdot 3(5 + o(1))} > \frac{\log \delta}{30 + o(1)}. \end{aligned}$$

Replacing β and δ , we obtain:

$$\frac{\alpha^c e \log t}{t} + \log \alpha \geq \frac{c \log \alpha}{30 + o(1)} \quad (4.9)$$

$$\implies \frac{\alpha^c \log t}{t} \geq \frac{1}{e} \left(\frac{c}{30 + o(1)} - 1 \right) \log \alpha \quad (4.10)$$

$$\implies \frac{\alpha^c \log t}{t} \geq \gamma \log \alpha, \quad (4.11)$$

where $\gamma = (c/(30 + o(1)) - 1)/e$. Because $c = 31$, we have $\gamma > 0$ (for n large enough), and since $\log \alpha \geq 1$, it follows that:

$$(4.11) \implies \alpha^c \geq \frac{\gamma t}{\log t} \quad (4.12)$$

$$\implies c \log \alpha \geq \log(\gamma t) - \log \log t. \quad (4.13)$$

Note that $k \geq en/(2t)$. We consider two cases. If $t \leq 2^{35}$, then $k \geq en/2^{36}$. Since $2^{36}/e > 2790\alpha^c$ for $c = 31$ and $\alpha \geq 2$, it follows in this case that:

$$k > \frac{n}{2790\alpha^c}.$$

If $t > 2^{35}$, then we check that (as $\gamma \rightarrow 0.001226\dots$ as $n \rightarrow +\infty$):

$$\log(\gamma t) - \log \log t > \frac{2}{3} \log t$$

thus by Eq. (4.13) $c \log \alpha > \frac{2}{3} \log t$. Bounding $\log t < (3c \log \alpha)/2$, Eq. (4.11) becomes:

$$\begin{aligned} \frac{\alpha^c 3c \log \alpha}{2t} &\geq \gamma \log \alpha \\ \implies t &\leq \frac{3c}{2\gamma} \alpha^c \\ \implies k &\geq \frac{e\gamma}{3c\alpha^c} n = \left(\frac{c/(30 + o(1)) - 1}{3c} \cdot \frac{1}{\alpha^c} \right) n \\ &> \frac{1}{90c\alpha^c} \cdot n \text{ (as } n \rightarrow +\infty) \\ &> \frac{1}{2790} \cdot \frac{n}{\alpha^c} \end{aligned}$$

that completes the proof. \square

5. Conclusion

We showed that α -adaptive routing tables on n -node graphs, that are routing tables mapping each destination on exactly α directions, have compactness at most n/α (i.e., require n/α intervals of destination labels per link), computable in polynomial time. We proved also that, if at least one shortest path must be represented, there are n -node graphs for which every α -adaptive routing table has compactness larger than $n/\alpha^{O(1)}$.

In the other side, it is known that if *all* the shortest paths must be represented, then such routing tables require compactness $n/4$ for some worst-case graphs. Therefore, it would be interesting to study the compactness of β -shortest path α -adaptive routing tables, a natural extension of shortest path α -adaptive routing tables, that map each destination on α directions and whose at least β must be on a shortest path. The present paper concerns $\beta = 1$.

We stress also that our $n/\alpha^{O(1)}$ -lower bound is not a serious obstacle for the study of graphs having small compactness, even for $\alpha = 2$. Indeed, due to some large constants in this existential lower bound, the smallest example of graphs we can prove by Theorem 4.5 to have a compactness greater than 1 must have more than 2^{42} nodes. It suggests that the class of graphs supporting shortest path 2-adaptive routing tables is rather large, and it would be interesting to develop this study to various class of concrete networks.

References

- [1] Y. Afek, E. Gafni, M. Ricklin, Upper and lower bounds for routing schemes in dynamic networks, in: 30th Annual IEEE Symposium on Foundations of Computer Science (FOCS), 1989, pp. 370–375.
- [2] B. Awerbuch, A. Bar-Noy, N. Linial, D. Peleg, Compact distributed data structures for adaptive routing, in: 21st Annual ACM Symposium on Theory of Computing (STOC), 1989, pp. 479–489.
- [3] E.M. Baker, R.B. Tan, J. van Leeuwen, Prefix routing schemes in dynamic networks, Technical Report RUU-CS-90-10, Dept. of Computer Science, Utrecht University, 1990.
- [4] T. Eilam, C. Gavoille, D. Peleg, Compact routing schemes with low stretch factor, in: 17th Annual ACM Symposium on Principles of Distributed Computing (PODC), 1998, pp. 11–20.
- [5] T. Eilam, S. Moran, S. Zaks, The complexity of the characterization of networks supporting shortest-path interval routing, in: 4th International Colloquium on Structural Information & Communication Complexity (SIROCCO), Carleton Scientific, 1997, pp. 99–111.
- [6] T. Eilam, S. Moran, S. Zaks, Lower bounds for linear interval routing, *Networks* 34 (1999) 37–46.
- [7] M. Flammini, On the hardness of devising interval routing schemes, *Parallel Processing Lett.* 7 (1997) 39–47.
- [8] P. Fraigniaud, C. Gavoille, Interval routing schemes, *Algorithmica* 21 (1998) 155–182.
- [9] M. Flammini, G. Gambosi, On devising Boolean routing schemes, *Theoret. Comput. Sci.* 186 (1997) 171–198.
- [10] M. Flammini, G. Gambosi, U. Nanni, R.B. Tan, Characterization results of all shortest paths interval routing schemes, in: 5th International Colloquium on Structural Information & Communication Complexity (SIROCCO), Carleton Scientific, July 1998, pp. 201–213.
- [11] M. Flammini, G. Gambosi, S. Salomone, Interval labeling schemes for chordal rings, in: 1st International Colloquium on Structural Information & Communication Complexity (SIROCCO), Carleton Univ. Press, May 1994, pp. 111–124.
- [12] M. Flammini, J. van Leeuwen, A. Marchetti-Spaccamela, The complexity of interval routing on random graphs, *Comput. J.* 41 (1998) 16–25.
- [13] G.N. Frederickson, R. Janardan, Separator-based strategies for efficient message routing, in: 27th Symposium on Foundations of Computer Science (FOCS), May 1986, pp. 428–437.
- [14] G.N. Frederickson, R. Janardan, Space-efficient message routing in c -decomposable networks, *SIAM J. Comput.* 19 (1990) 164–181.
- [15] C. Gavoille, On the dilation of interval routing, *Comput. J.* 43 (2000) 1–7.
- [16] C. Gavoille, A survey on interval routing, *Theoret. Comput. Sci.* 245 (2000) 217–253.
- [17] C. Gavoille, E. Guévremont, Worst case bounds for shortest path interval routing, *J. Algorithms* 27 (1998) 1–25.
- [18] C. Gavoille, D. Peleg, The compactness of interval routing, *SIAM J. Discrete Math.* 12 (1999) 459–473.

- [19] C. Gavoille, S. Pérennès, Memory requirement for routing in distributed networks, in: 15th Annual ACM Symposium on Principles of Distributed Computing (PODC), 1996, pp. 125–133.
- [20] E. Kranakis, D. Krizanc, Lower bounds for compact routing, in: 13th Annual Symposium on Theoretical Aspects of Computer Science (STACS), in: Lecture Notes in Computer Sci., Vol. 1046, Springer, Berlin, 1996, pp. 529–540.
- [21] E. Kranakis, D. Krizanc, J. Urrutia, Compact routing and shortest path information, in: 2nd International Colloquium on Structural Information & Communication Complexity (SIROCCO), Carleton Univ. Press, 1995, pp. 101–112.
- [22] R. Kráľovič, P. Ružička, D. Štefankovič, The complexity of shortest path and dilation bounded interval routing, *Theoret. Comput. Sci.* 234 (2000) 85–107.
- [23] P. Loh, J. Wenge, Adaptive, fault-tolerant, deadlock-free and livelock-free interval routing in mesh networks, in: 2nd IEEE International Conference on Algorithms & Architectures for Parallel Processing (ICAPP), 1996, pp. 348–355.
- [24] L. Narayanan, J. Opatrny, Compact routing on chordal rings of degree four, in: 4th International Colloquium on Structural Information & Communication Complexity (SIROCCO), Carleton Scientific, 1997, pp. 125–137.
- [25] D. Peleg, E. Upfal, A trade-off between space and efficiency for routing tables, *J. ACM* 36 (1989) 510–530.
- [26] P. Ružička, D. Štefankovič, On the complexity of multi-dimensional interval routing schemes, *Theoret. Comput. Sci.* 245 (1999).
- [27] N. Santoro, R. Khatib, Labelling and implicit routing in networks, *Comput. J.* 28 (1985) 5–8.
- [28] S.S.H. Tse, F.C.M. Lau, An optimal lower bound for interval routing in general networks, in: 4th International Colloquium on Structural Information & Communication Complexity (SIROCCO), Carleton Scientific, 1997, pp. 112–124.
- [29] J. van Leeuwen, R.B. Tan, Computer networks with compact routing tables, in: G. Rozemberg, A. Salomaa (Eds.), *The Book of L*, Springer, Berlin, 1986, pp. 259–273.
- [30] J. van Leeuwen, R.B. Tan, Interval routing, *Comput. J.* 30 (1987) 298–307.