Algorithmica (1998) 21: 155–182

Algorithmica

Interval Routing Schemes[1]

P. Fraigniaud[2] and C. Gavoille[2]

Abstract. Interval routing was introduced to reduce the size of routing tables: a router finds the direction where to forward a message by determining which interval contains the destination address of the message, each interval being associated to one particular direction. This way of implementing a routing function is quite attractive but very little is known about the topological properties that must satisfy a network to support an interval routing function with particular constraints (shortest paths, limited number of intervals associated to each direction, etc.). In this paper we investigate the study of the interval routing functions. In particular, we characterize the set of networks which support a *linear* or a *linear strict* interval routing function with only one interval per direction. We also derive practical tools to measure the efficiency of an interval routing function (number of intervals, length of the paths, etc.), and we describe large classes of networks which support *optimal* (linear) interval routing functions. Finally, we derive the main properties satisfied by the popular networks used to interconnect processors in a distributed memory parallel computer.

Key Words. Routing in distributed networks, Compact routing, Routing function, Interval.

1. Introduction. Given a network of processors (such as the one of a distributed memory parallel computer), the way of routing messages among the processors is characterized, on one hand, by the routing *mode* (store-and-forward, circuit-switched, wormhole, . . .) and, on the other hand, by the routing *function* which determines the paths between the sources and the destinations. This paper focuses on the second parameter.

The routing function is generally implemented locally on the routers. The route of a message from its source to its destination is determined using a header attached to the message, and which contains information that will allow the intermediate routers to know where to forward the message. In this paper we are interested in routing functions which use only the destination address of the message to find the route.

As soon as a router receives a message, it looks at the header to read the destination, and then determines the output port which will be used to forward the message toward its destination. There are mainly two ways of determining the output port from the destination address:

1. Application of an algorithm.
2. Consultation of a routing table.

---

[1] All the results presented in this paper are entirely based on [8]. The first author received the support of the Centre de Recerca Matemàtica, Institut d'Estudis Catalans, Bellaterra, Spain. Both authors are supported by the research programs ANM and PRS of the CNRS.

[2] LIP - CNRS, École Normale Supérieure de Lyon, 46 allée d'Italie, 69364 Lyon Cedex 07, France. {pfraign,gavoille}@ens-lyon.fr.

Received February 7, 1996; revised November 25, 1996. Communicated by F. T. Leighton.

The first case is generally used when the topology of the interconnection network is fixed, and simple. For instance, it is easy to implement locally the XY-routing on a mesh or on a torus, and the e-cube routing on the hypercube: the output port is found by comparing the current address of the router with the address of the destination.

However, if the structure of the interconnection network is fixed but complicated (a pancake graph, an undirected de Bruijn graph, etc.), it could be difficult to derive a "simple algorithmic way" to compute the paths locally (especially if one insists on shortest paths). By a *simple algorithm*, we mean an algorithm whose execution time and space for implementing it on the router are both small. If the interconnection network has no particular structure, it can even be impossible to derive any kind of algorithm. A solution to these problems is obtained by the use of routing tables which are stored locally on each router. The main requirement for these tables is to be as small as possible (for instance, a size of $\Theta(n)$ for a network of n processors is not realistic as soon as the number of processors becomes large).

Compact routing has already been intensively studied (see, for instance, [2], [11], and [12]). There exist many solutions to compress the size of the routing tables. The usual approach consists in grouping the destination addresses which correspond to the same output port, and encoding the group so that it will be is easy to check whether or not a destination address belongs to a given group. A very popular solution of that kind is the use of intervals [26]. Intervals are indeed very simple to code (it is sufficient to store the bounds of each interval), and at most two comparisons are necessary to check whether a destination address belongs to an interval. This kind of routing is used, for instance, on the C104 routing chip [7], [21] of INMOS.

Interval routing is very attractive by its simplicity. Unfortunately, it is not always simple to fix a global labeling of the nodes so that intervals can be easily set for each output port of each router, especially if one insists on shortest paths or other particular properties.

The notion of interval routing was introduced by Santoro and Khatib in [26]. They have mainly shown that any directed acyclic graph can support an interval routing function with shortest paths, and with only one interval per output port. Moreover, if the digraph is not acyclic, they have shown that there exists an interval routing function such that the maximum length of the route between two vertices is at most twice the diameter. Van Leeuwen and Tan [30] have studied the problem for undirected graphs. They have shown that any graph supports an interval routing function with one interval per output port. They have given simple examples of graphs (trees, complete graphs, rings, meshes, . . .) which support an interval routing function with one interval per output port and where all routes are shortest paths. They have also studied the number of intervals per output port that requires a shortest path interval routing function on tori. In [3] Bakker et al. introduced a particular class of interval routing functions, namely *linear* interval routing schemes. They characterized trees which support such a routing scheme with only one interval per output port, and where all the routes are shortest paths. They listed simple examples of graphs which support a linear interval routing function with shortest paths, and graphs which do not support such a scheme. They showed that the hypercube and the d-dimensional meshes support a linear interval routing function with shortest paths. Finally, Bakker et al. also studied linear interval routing schemes, but with constraints on the neighbor-to-neighbor communication costs. This problem was introduced by Frederickson and Janardan in [10], in the field of interval routing.

In this paper we investigate interval routing in depth. In particular, we focus on the topological properties that a graph must satisfy in order to support an interval routing function with some particular given properties. Note that this work has direct practical interest for the construction of interconnection networks for distributed memory computers based on C104 routing chips [7].

In Section 2 we define precisely what an interval routing function is. We focus on some parameters which are directly related to this definition: the number of intervals per output port, the kinds of intervals (linear or cyclic), and the way of checking whether the current address is the destination (strict interval routing function).

In Section 3 we study the minimum number of intervals that must be associated to each output port of a given network to support a (linear) interval routing function. It was known that one interval per output port is enough for cyclic intervals [30], but very little was known about linear intervals. We characterize the networks which support a linear interval routing function with at most one interval per output port. We also characterize the graphs which support a strict linear interval routing function with at most one interval per output port.

In Section 4 we study the length of the routes generated by a (linear) interval routing function. First we study the networks which support a (linear) interval routing function generating shortest paths. Then we study the tradeoff between the maximum length of the paths generated by a routing function and the number of intervals which are used per output port.

Finally, in Section 5, we study properties that satisfy graphs which are interesting either for the practical design of interconnection networks, or for some algorithmic points of view: ring, mesh, hypercube, cube-connected-cycle, shuffle-exchange, ... (see [9] and [20]).

2. Statement of the Problem.

We are interested in parallel distributed memory multicomputers composed of processing elements (PEs) connected to routers in a one-to-one fashion. This last hypothesis is not restrictive since most of the results of this paper can be generalized in the case where more than one PE can be connected to a same router, or where no PE is connected to some routers. As usual, the network is modeled by a graph $G = (V, E)$ whose set of vertices V represents the routers, and whose set of edges E represents the communication links between the routers. We assume that the links are bidirectional (that is, if a router x is able to send messages to one of its neighbors y, then y is also able to send messages to x), and, therefore, we deal with undirected graphs (or symmetric digraphs) only. Of course, we are only interested in connected networks, so all the statements of this paper assume that the graphs are connected (there is a path between any couple of vertices). Also, we always consider finite graphs which are simple (there is at most one edge between two vertices) and loopless. An edge of extremities x and y is therefore denoted by (x, y).

For any vertex $x \in V$, we denote by $\mathrm{out}(x)$ the set of edges of extremity x, that is $\mathrm{out}(x) = \{(x, y) \in E\}$. We get $|\mathrm{out}(x)| = \deg(x)$, the degree of x. Each router, that is each vertex x of G, is connected to the memory of its associated PE by a communication link. We denote this link by $\mathrm{mem}(x)$. We define a routing function as follows:

DEFINITION 1 (Routing Function). A *routing function* R on a graph $G = (V, E)$ is a set of functions

$$R = \{R_x \mid x \in V, R_x\colon V \to \mathrm{out}(x) \cup \mathrm{mem}(x)\}$$

such that, for any couple of vertices $x, y \in V$, there exists a sequence of vertices $x = x_0, \ldots, x_k = y$ such that, for every $i \in \{0, \ldots, k-1\}$, $R_{x_i}(y) = (x_i, x_{i+1})$, and $R_y(y) = \mathrm{mem}(y)$.

This definition understands that we only consider routing functions which are connected. Assume a router x receives a message whose destination address is y. If $y = x$, then the message is sent to the local memory of the PE connected to x via $\mathrm{mem}(x)$. If $y \neq x$, then the message is forwarded on the communication link of $\mathrm{out}(x)$ determined by $R_x(y)$. We focus now on routing functions that are defined using intervals:

DEFINITION 2 (Interval). An *interval* of $\{1, \ldots, n\}$ denoted by $[a, b]$, where $a, b \in \{1, \ldots, n\}$, is the set of integers i satisfying

$$\begin{cases} a \le i \le b & \text{if} \quad a \le b \quad (\textit{linear} \text{ interval}), \\ a \le i \le n \quad \text{or} \quad 1 \le i \le b & \text{if} \quad a > b \quad (\textit{cyclic} \text{ interval}). \end{cases}$$

If $a = b$ we denote by $[a]$ the interval $[a, a]$. We also denote by $]a, b]$ (resp. $[a, b[$) the interval $[a, b] - \{a\}$ (resp. $[a, b] - \{b\}$). $\emptyset$ and [ ] both refer to the empty interval. Informally, an interval routing function on a graph of n vertices is defined as follows. First, label the vertices by integers from 1 to n in a one-to-one manner. Then, for each vertex x, associate intervals to each edge $e \in \mathrm{out}(x)$. The number of intervals associated to $e \in \mathrm{out}(x)$ *on* x is denoted by $k(x, e)$. A message located on x, and of destination y, is routed by y through $e \in \mathrm{out}(x)$ if and only if y belongs to one of the intervals associated to e on x.

For instance, in Figure 1 we have indicated two interval routing functions for the same graph. The labels of the vertices are different, so it is for the intervals. A message sent by A to D will follow the path *ABED* using the interval routing function depicted on the left-hand side of Figure 1, and the path *AD* using the function of the right-hand

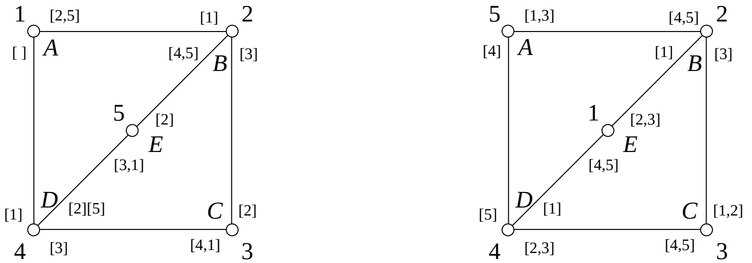

Fig. 1. Two interval routing functions for the same network.

side of Figure 1. These two interval routing functions are very different. The function on the left has many drawbacks: nonshortest paths, two intervals on the same edge for the vertex D, cyclic intervals on vertices C and E, the edge (A, D) is never used from A, and E contains its own label in the interval associated to (E, D). The function on the right offers many good properties: shortest paths, one nonempty interval per edge on each vertex, linear intervals only, and no interval contains the label of the local vertex.

More formally, we define an interval routing function as follows:

DEFINITION 3 (Interval Routing Function). Let $G = (V, E)$ be a graph of n vertices. An *interval routing function* on G is a routing function $R = \{R_x \mid x \in V\}$ on G defined by

1. a one-to-one function $\mathcal{L}$: $V \to \{1, \dots, n\}$ which labels the vertices of G,
2. a set of intervals $\mathcal{I} = \{I^1_{x,e}, \dots, I^{k(x,e)}_{x,e} \mid x \in V, e \in \text{out}(x), k(x, e) \geq 1\}$ such that the sets $I_{x,e} = \bigcup_{i=1}^{k(x,e)} I^i_{x,e}$, $x \in V$, $e \in \text{out}(x)$, satisfy
   - union property: $(\bigcup_{e \in \text{out}(x)} I_{x,e}) \cup \{\mathcal{L}(x)\} = \{1, \dots, n\}$,
   - disjunction property: $\forall e, e' \in \text{out}(x)$, $e \neq e' \Rightarrow I_{x,e} \cap I_{x,e'} = \emptyset$,

and satisfying

$$R_x(y) = \begin{cases} \text{mem}(x) & \text{if} \quad y = x, \\ e \in \text{out}(x) & \text{such that} \quad \mathcal{L}(y) \in I_{x,e} \quad \text{otherwise.} \end{cases}$$

An interval routing function is denoted by a couple $(\mathcal{L}, \mathcal{I})$ which satisfies the conditions of Definition 3. For practical reasons, it might be interesting to restrict the definition, and to allow the use of linear intervals only (see [3]). This notion is particularly useful to derive results on networks built by a cartesian product (as hypercubes and torus).

DEFINITION 4 (Linear Interval Routing Function). A *linear* interval routing function is an interval routing function $R = (\mathcal{L}, \mathcal{I})$ where $\mathcal{I}$ contains linear intervals only.

In Figure 1 the function on the right-hand side is linear and the function on the left-hand side is not linear.

Again, for practical reasons related to the design of the router, it is important to distinguish the case where intervals contain the local address from the case where they do not. Indeed, if an interval contains the local address, then a preprocessing must be implemented to check whether the destination address is the current address before using the intervals. On the contrary, if we know that no interval contains the local address, then we can associate an interval to the memory link, and there is no distinction between this link and the other. Moreover, as we will see later, and as is the case for linear intervals, this notion is particularly interesting for the construction of interval routing functions on networks obtained by a cartesian product.

DEFINITION 5 (Strict Interval Routing Function). An interval routing function $R = (\mathcal{L}, \mathcal{I})$ is *strict* if $\forall x \in V, \forall e \in \text{out}(x)$, $\mathcal{L}(x) \notin I_{x,e}$.

In Figure 1 the function on the right-hand side is strict, but the one on the left-hand side is not strict (see vertex E). Strict interval routing functions have been considered in [10] by Frederickson and Janardan, whereas Bakker et al., in [3], considered nonstrict linear interval routing functions.

3. Compact Interval Routing Scheme. Definition 3 is general in the sense that every routing function on a graph G can be expressed by an interval routing function. Indeed, for any routing function R on G, choose any vertex-labeling $\mathcal{L}$. Then, for every vertex x, associate the set of intervals $I_{x,e} = [x_1] \cup [x_2] \cup \cdots \cup [x_{k(x,e)}]$ to each edge $e \in \mathrm{out}(x)$, where $\{x_1, x_2, \ldots, x_{k(x,e)}\}$ is the list of labels of destinations whose messages use e to leave x applying R. However, such a use of interval routing is close to the use of routing tables. Since interval routing has been introduced to reduce the memory space used on the routers, we are interested in limiting the number of intervals per edge on each vertex. Idealy, one would like to have at most one interval per edge on each vertex yielding a memory space of size $O(d \log n)$ bits per router of degree d. More formally, we define *compactness* as follows:

DEFINITION 6 (Compactness). Let $R = (\mathcal{L}, \mathcal{I})$ be an interval routing function on a graph G. The *compactness* of R is defined by $\max_{x \in V} \max_{e \in \mathrm{out}(x)} k(x, e)$.

In other words, the compactness of R is the maximum taken over all the vertices x of the maximum taken over all the edges e of extremity x of the number of intervals necessary to list the destinations for which e will be used from x. Note that one might object that $\max_{x \in V} \sum_{e \in \mathrm{out}(x)} k(x, e)$ would be a more appropriated parameter for measuring the size of the local tables. However, it is much more tricky to deal with this parameter, and the reader will be convinced soon that compactness, as described in Definition 6, allows us to derive strong results with many practical applications.

NOTATIONS. For any integer $k \geq 1$, we denote by k-IRS the class of graphs supporting an interval routing function of compactness at most k. Similarly we denote by k-LIRS the class of graphs supporting a linear interval routing function of compactness at most k. Also k-(L)IRS *strict* denotes the class of graphs which support a strict (linear) interval routing function of compactness at most k.

3.1. *A Previous Result.* First, very good news due to van Leeuwen and Tan [30]:

THEOREM 1 [30]. *All graphs belong to* 1-IRS *strict.*

We briefly recall their proof.

PROOF. Let r be any vertex of $V(G)$, and consider a spanning tree T of G rooted at r. We define $\mathcal{L}$ by a depth-first labeling from the root r with $\mathcal{L}(r) = 1$, and performing by increasing order. For any vertex x of T, let $l_x = \max \mathcal{L}(y)$ over all the vertices y which belong to the subtree of T of root x.

We assign the empty interval $\emptyset$ to both extremities of all the edges of G which do not belong to T. For each edge $e = (x, y)$, where y is a child of x in T (if it exists), we assign the interval $[\mathcal{L}(y), l_y]$ to e on x. For each edge $e = (x, y)$ of T where y is the father of x (if it exists), we assign the interval $]l_x, \mathcal{L}(x)[$ to e in x.

Clearly, such an interval routing function is strict. $\square$

Note that any path constructed by the routing function defined as in the proof of Theorem 1 is embedded in a tree. Hence, the length of this path is necessary less than twice the depth of the tree. However, it might be too far to be a shortest path between the source and the destination. This strongly moderates the good news!

We now characterize the graphs that belong to 1-LIRS. First, we give the following:

COROLLARY 1. *All graphs belong to* 2-LIRS *strict*.

PROOF. Every (strict) interval routing function of compactness k on a graph G can be transformed in a (strict) linear interval routing function on G of compactness at most $k + 1$ by splitting each cyclic interval into two linear intervals. Therefore, for every integer $k \geq 1$, k-IRS $\subset$ $(k+1)$-LIRS, and k-IRS strict $\subset$ $(k+1)$-LIRS strict. $\square$

3.2. *Characterization of* 1-*LIRS*. Clearly, there exist graphs which do not belong to 1-LIRS. For instance, consider the graph of Figure 2 (that we term the Y-graph), and assume that there exists a linear interval routing function R with compactness 1 on this graph. Then there exists a branch such that neither the vertex x in the middle of the branch, nor the vertex y at the extremity of the same branch is labeled 1 or 7. We call e the edge between x and the center z. Necessarily, the corresponding interval $I_{x,e}$ must contain 1 and 7, thus $I_{x,e} = [1, 7]$, and y is not reachable from x: a contradiction. Below, we characterize the graphs that belong to 1-LIRS.

Recall that an edge e of a graph $G = (V, E)$ is a bridge if and only if $G' = (V, E - \{e\})$ is a disconnected graph.

DEFINITION 7. A *lithium-graph* is a graph with three bridges that connect a same connected component (the *kernel*) with three other distinct connected components (the *electrons*) of at least two vertices.

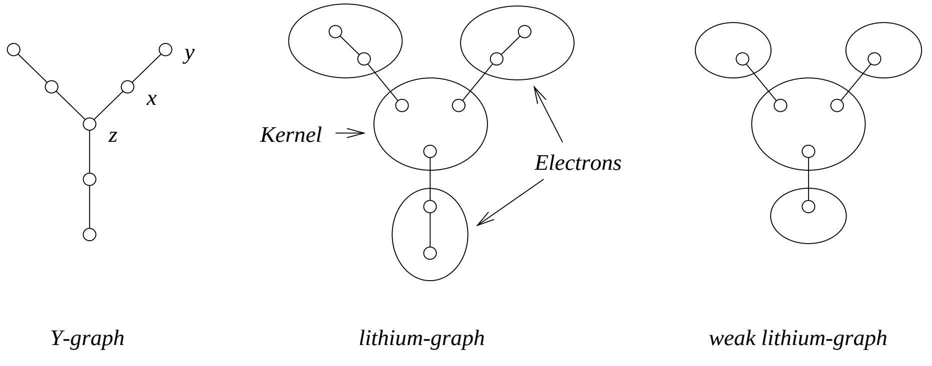

Fig. 2. The Y-graph, a lithium-graph, and a weak lithium-graph.

Figure 2 shows the general form of a lithium-graph. The Y-graph is a typical example of a lithium-graph, the smallest one actually. We can get the following lemma easily by the same arguments which show that the Y-graph $\notin$ 1-LIRS:

LEMMA 1. *If G is a lithium-graph, then $G \notin$* 1-LIRS.

In fact, we get:

THEOREM 2. *$G \in$ 1-LIRS $\Leftrightarrow$ G is not a lithium-graph.*

We have to show that any graph which is not a lithium-graph belongs to 1-LIRS. To do that, we need some preliminary results. In the following, we assume that G has at least three vertices, since otherwise $G \in$ 1-LIRS strict, and G is not a lithium-graph.

LEMMA 2. *Every* 2*-edge-connected graph $G = (V, E)$ belongs to* 1-LIRS *strict. Moreover, for any two vertices x and y of G, there exists a linear strict interval routing function $R = (\mathcal{L}, \mathcal{I})$ of compactness* 1 *satisfying*:

(i) $\mathcal{L}(x) = 1$;
(ii) $\forall z \in V, \mathcal{L}(z) < \mathcal{L}(y), \forall (z, u) \in E$: $\mathcal{L}(y) \in I_{z,(z,u)} \Rightarrow |V| \in I_{z,(z,u)}$;
(iii) *let z be the vertex such that $\mathcal{L}(z) = |V|, \exists (z, u) \in E$, $\mathcal{L}(u) \le \mathcal{L}(y)$, and $I_{z,(z,u)} = \emptyset$.*

PROOF. We proceed iteratively: at each step we consider a subgraph $H = (V_H, E_H)$ of G containing x and y, and a linear strict interval routing function on H satisfying properties (i)–(iii). We successively update this construction, keeping the good properties and adding one or more vertices to H until $|V_H| = |V|$. We detail below the initialization of our construction, and then the way to update our construction.

Initialization. If $x \neq y$, then, from Menger's theorem, let P_1 and P_2 be two edge-disjoint paths from x to y. It is actually possible to find such paths in G such that if they have a certain number of common vertices $u_1, \ldots, u_{r-1}$ distinct from x and y, then these vertices are encountered in the same order going from x to y on P_1, and on P_2. Thus, let $u_0 = x$, $u_r = y$, and, for $i \in \{0, \ldots, r-1\}$, let $\mathcal{C}_i$ be the cycle composed of the path P_1 from u_i to u_{i+1}, and the path P_2 from u_{i+1} to u_i.

If $x = y$, then let $\mathcal{C}_0$ be a cycle of G (of at least three vertices) going through x. Set $u_0 = x = y = u_1$.

Let $H = (V_H, E_H)$ be the subgraph of G obtained by union of the $\mathcal{C}_i$'s. We label the vertices of H as follows (see Figure 3(a)–(c)): set $\mathcal{L}(x) = 1$, and label clockwise the vertices of $\mathcal{C}_0$ in increasing order. If there is more than one cycle, then start from u_1, and label clockwise the vertices of $\mathcal{C}_1$ in increasing order. Repeat this operation considering successively the cycles $\mathcal{C}_i$, $i = 2, \ldots, r-1$, until all the vertices of the $\mathcal{C}_i$'s are labeled.

Now, we set the intervals as follows (see Figure 3(d)). Let n_H be the number of vertices of H. Let n_i be the number of vertices of the cycle $\mathcal{C}_i$, for $i \in \{0, \ldots, r-1\}$. Consider any cycle $\mathcal{C}_i$, for $i \in \{0, \ldots, r-1\}$. Let z be any vertex of $\mathcal{C}_i$. Let e^+ (resp.

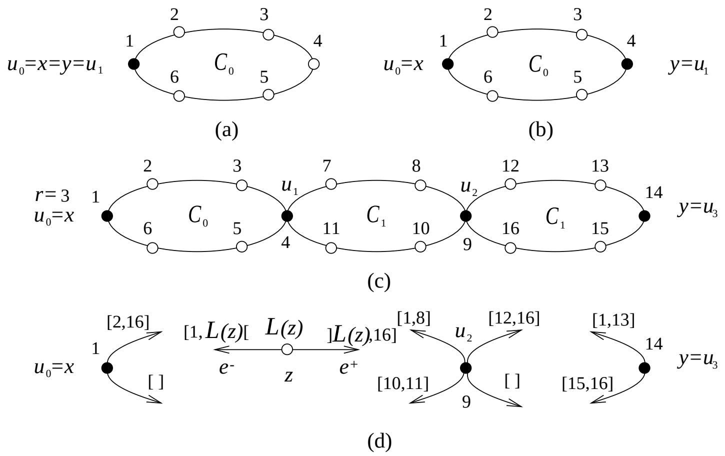

Fig. 3. Construction of the proof of Lemma 2.

e^-) be the clockwise (resp. counterclockwise) edge of $\mathrm{out}(z)$ on $\mathcal{C}_i$. We set

$$I_{z,e^+} = \begin{cases}]\mathcal{L}(z), n_H] & \text{if} \quad z \neq u_{i+1} \quad \text{or} \quad u_{i+1} = y, \\ ]\mathcal{L}(z), \sum_{j=0}^{i} n_j - i] & \text{if} \quad z = u_{i+1} \quad \text{and} \quad u_{i+1} \neq y, \end{cases}$$

and

$$I_{z,e^-} = \begin{cases} [1, \mathcal{L}(z)[& \text{if} \quad z \neq u_i, \\ \emptyset & \text{if} \quad z = u_i. \end{cases}$$

That this labeling and the setting of these intervals yield a strict linear interval routing function on H can be easily checked. Concerning the three properties, (i) is satisfied ($\mathcal{L}(x) = 1$). Now, if $\mathcal{L}(z) < \mathcal{L}(y)$, then $\mathcal{L}(y)$ can belong to I_{z,e^+} only. If $I_{z,e^+} =]\mathcal{L}(z), n_H]$, then (ii) is satisfied; if $I_{z,e^+} =]\mathcal{L}(z), \sum_{j=0}^{i} n_j - i]$, then $\mathcal{L}(y)$ cannot belong to I_{z,e^+}. Thus (ii) is satisfied. Finally, by definition, if z is the vertex labeled n_H, $I_{z,e^+} =]n_H, \sum_{j=0}^{r-1} n_j - r + 1] =]n_H, n_H] = \emptyset$, and thus (iii) is also satisfied.

Updating. Let H be a subgraph of G with $n_H < |V|$ vertices, and containing vertices x and y. Let $R = (\mathcal{L}, \mathcal{I})$ be a linear strict interval routing function on H satisfying conditions (i)–(iii). There exists a path $P = (v_0, v_1, \ldots, v_k, v_{k+1})$, $k \geq 1$, in G such that $v_0 \in V_H$, $v_{k+1} \in V_H$, and $v_i \notin V_H$, for every $i \in \{1, \ldots, k\}$. Consider $H' = H \cup P$, and assume $\mathcal{L}(v_0) < \mathcal{L}(v_{k+1})$. Consider the following labeling $\mathcal{L}'$ of the vertices of H': for $v \in V_H$, if $\mathcal{L}(v) \leq \mathcal{L}(v_0)$, then $\mathcal{L}'(v) = \mathcal{L}(v)$, otherwise $\mathcal{L}'(v) = \mathcal{L}(v) + k$. For $i \in \{1, \ldots, k\}$, $\mathcal{L}'(v_i) = \mathcal{L}(v_0) + i$.

We update the intervals of $\mathcal{I}$ as follows: let $v \in V_H$, and let $e \in E_H$ be an edge incident to v. Assume $I_{v,e} = [a, b]$, we set

$$I'_{v,e} = \begin{cases} [a, b] & \text{if} \quad b < \mathcal{L}(v_0), \\ [a, b + k] & \text{if} \quad a \leq \mathcal{L}(v_0) \leq b, \\ [a + k, b + k] & \text{if} \quad \mathcal{L}(v_0) < a. \end{cases}$$

Now we have to fix the intervals on the path P. Let e be the edge of P of extremity v_0, we set $I'_{v_0,e} = [\mathcal{L}'(v_1), \mathcal{L}'(v_k)]$. Let e be the edge of P of extremity v_{k+1}, we set $I'_{v_{k+1},e} = \emptyset$. Let $e = (v_i, v_{i+1})$, $i \in \{1, \dots, k\}$, we set $I'_{v_i,e} = [\mathcal{L}'(v_{i+1}), n_H + k]$. Let $e = (v_{i-1}, v_i)$, $i \in \{1, \dots, k\}$, we set $I'_{v_i,e} = [1, \mathcal{L}'(v_{i-1})]$. It is easy to check that $R' = (\mathcal{L}', \mathcal{I}')$ is a linear strict interval routing function on H'. Properties (i) and (iii) are of course still satisfied. Assume $v \in V_H$, $e \in E_H$, and $I_{v,e} = [a, b]$. If $\mathcal{L}'(y) \in I'_{v,e}$, then $b \geq \mathcal{L}(v_0)$ because otherwise $\mathcal{L}'(y)$ would be strictly less than $\mathcal{L}(v_0)$, thus $\mathcal{L}'(y) = \mathcal{L}(y)$ and $b = |V|$ which is impossible if $b < \mathcal{L}(v_0)$. There are two cases: either $\mathcal{L}(y) \leq \mathcal{L}(v_0)$, or $\mathcal{L}(y) > \mathcal{L}(v_0)$. In both cases, if $\mathcal{L}'(y) \in I'_{v,e}$, then $\mathcal{L}(y) \in I_{v,e}$ and $b = n_H$, that is $n_H + k \in I'_{v,e}$. It is also easy to check that property (ii) also holds for the intervals on P. Thus R' satisfies property (ii) on H'.

Let $H = H'$, and repeat this process until $n_H = |V|$. □

LEMMA 3. *Let $G \in$ 1-LIRS strict. Let H be the graph obtained from G by adding a set of independent vertices S of G such that, for all $x \in S$, x is connected to only one vertex of G. Then $H \in$ 1-LIRS.*

PROOF. Let $R = (\mathcal{L}, \mathcal{I})$ be a strict linear interval routing function on G. For any vertex x in G, we denote by $\nu(x)$ the number of vertices of S which are connected to x in $H = (V', E')$, $V' = V \cup S$. Then we define the labeling $\mathcal{L}'$ of the vertices of H as follows. For any $x \in V$, $\mathcal{L}'(x) = \mathcal{L}(x) + \sum_{y \in V | \mathcal{L}(y) < \mathcal{L}(x)} \nu(y)$. If x is connected to p vertices of S in H, namely $s_1, \dots, s_p$, then $\mathcal{L}'(s_i) = \mathcal{L}'(x) + i$ for all $i \in \{1, \dots, p\}$. An interval $[\mathcal{L}(u), \mathcal{L}(v)] \in \mathcal{I}$ is transformed in an interval $[\mathcal{L}'(u), \mathcal{L}'(v) + \nu(v)] \in \mathcal{I}'$. Finally, we set new intervals in $\mathcal{I}'$: $I_{x,(x,s_i)} = [\mathcal{L}'(s_i)]$ and $I_{s_i,(s_i,x)} = [1, |V'|]$. □

Now, we can state our proof:

PROOF OF THEOREM 2. Let $G = (V, E)$ be a graph which is not a lithium-graph. We say that an edge of E is a *strong* bridge if it is a bridge (that is, a cut-edge) that splits G into two connected components, each of at least two vertices. We decompose G, by deletion of all the strong bridges of E, in the maximum number of connected components $G_0, \dots, G_k$, where $k \geq 0$ is the total number of strong bridges of G. Note that some components may contain only one vertex.

Since G is not a lithium-graph, each component G_i is connected to at most two other components. Therefore, we can assume that G is a "path" of G_i's of the form $G_0 - G_1 - \cdots - G_k$. Let $x_i \in V(G_i)$ and $y_{i-1} \in V(G_{i-1})$ be the vertices such that the strong bridges are the edges (y_{i-1}, x_i), $i \in \{1, \dots, k\}$. We also define $x_0 = y_0$ and $y_k = x_k$.

For every $i \in \{0, \dots, k\}$, let G'_i be the graph obtained from G_i by removing all the vertices of degree 1 in G, and let $G' = \bigcup_{i=0}^{k} G'_i$. From Lemma 2, each $G'_i \in$ 1-LIRS strict and, more precisely, a strict linear interval routing function $R_i = (\mathcal{L}_i, \mathcal{I}_i)$ on G'_i can be found such that

(i) $\mathcal{L}_i(x_i) = 1$,
(ii) $\forall z \in V(G'_i), \mathcal{L}_i(z) < \mathcal{L}_i(y_i), \forall (z, u) \in E(G'_i)$: $\mathcal{L}_i(y_i) \in I_{z,(z,u)} \in \mathcal{I}_i \Rightarrow |V(G'_i)| \in I_{z,(z,u)}$,

(iii) let z be the vertex such that $\mathcal{L}_i(z) = |V(G_i')|$, $\exists(z,u) \in E(G_i')$, $\mathcal{L}_i(u) \leq \mathcal{L}_i(y_i)$, and $I_{z,(z,u)} = \emptyset \in \mathcal{I}_i$.

From the routing functions $R_i = (\mathcal{L}_i, \mathcal{I}_i)$ on the G_i''s, $i \in \{1,\ldots,k\}$, a strict linear interval routing function $R = (\mathcal{L}, \mathcal{I})$ on G' can be defined as follows. Let z be any vertex of G', and let i be such that z is a vertex of G_i'. We set $\mathcal{L}(z)$ by a simple shift of $\mathcal{L}_i(z)$: $\mathcal{L}(z) = \mathcal{L}_i(z) + \sum_{j=0}^{i-1} |V(G_j')|$. The intervals of $\mathcal{I}_i$ are shifted similarly by adding $\sum_{j=0}^{i-1} |V(G_j')|$ to both extremities excepted in the following cases: if one of the two extremities is 1, this extremity is unchanged; if one of the two extremities is $|V(G_i')|$, we set this extremity to $|V(G')|$. We also replace the empty intervals $I_{z,(z,u)}$ defined by property (iii) by $I_{z,(z,u)} =]\mathcal{L}(z), |V(G')|]$. Finally, we set $I_{x_i,(x_i,y_{i-1})} = [1, \mathcal{L}(x_i)[$ and $I_{y_{i-1},(y_{i-1},x_i)} =]\sum_{j=0}^{i-1} |V(G_j')|, |V(G')|]$, for every $i \in \{1,\ldots,k\}$.

With these labeling and intervals, property (i) ensures that the route from a vertex in G_i' to a vertex of G_j', $j < i$, goes through x_i, leaves G_i' by the strong bridge (x_i, y_{i-1}), then goes to x_{i-1}, leaves G_{i-1}' by the strong bridge (x_{i-1}, y_{i-2}), etc. Property (ii) ensures that the route from a vertex z in G_i', with $\mathcal{L}(z) \leq \mathcal{L}(y_i)$, to a vertex of G_j', $j > i$, goes through y_i and leaves G_i' by the strong bridge (y_i, x_{i+1}). Property (iii) ensures that the route from a vertex z in G_i', with $\mathcal{L}(z) > \mathcal{L}(y_i)$, to a vertex of G_j', $j > i$, goes through the vertex of the highest label in G_i', then reaches a vertex of G_i' with a label smaller than y_i, then goes through y_i and leaves G_i' by the strong bridge (y_i, x_{i+1}), then goes to y_{i+1} and leaves G_{i+1}' by the strong bridge (y_{i+1}, x_{i+2}), etc. We get $G' \in$ 1-LIRS strict.

We conclude the proof by adding the vertices of degree 1 and applying Lemma 3. □

The characterization of Theorem 2 gives an easy way to determine whether a graph supports a linear interval routing function of compactness 1. From a time complexity point of view, checking whether a graph is a lithium-graph is polynomial. Moreover, the reader can check that Theorem 2 gives an $O(n^2)$ algorithm to derive an interval routing function $(\mathcal{L}, \mathcal{I})$ on a graph which belongs to 1-LIRS. Therefore, it is quite easy to know if a graph belongs to 1-LIRS or not. For instance:

COROLLARY 2. *Every interval graph belongs to* 1-LIRS.

Recall that an interval graph [15] is a graph in which each vertex is an interval of $\mathbb{R}$, and where edges are pairs of intervals which intersect.

PROOF. Let a $\tilde{\mathrm{Y}}$-graph be a particular case of lithium-graphs for which there exist three bridges which connect a same connected component (the kernel) with three distinct connected components (the electrons) of *exactly* two vertices. Note that the vertices of the kernel which connect each electron with the kernel can be distinct or not. It is easy to check that any $\tilde{\mathrm{Y}}$-graph is not an interval graph (see [15] and [16]). Any lithium-graph has a $\tilde{\mathrm{Y}}$-graph as an induced subgraph, and any induced subgraph of an interval graph is an interval graph. Therefore a lithium-graph is not an interval graph, that is equivalent to saying that any interval graph belongs to 1-LIRS (from Theorem 2). □

Note that the cycle of n vertices, C_n, for $n \geq 4$, is not an interval graph [16]. However, $C_n \in$ 1-LIRS by a trivial application of Theorem 2. Therefore, the class 1-LIRS is not reduced to interval graphs. In fact, this class contains most of the usual networks considered for interconnecting PEs of a distributed memory computer. Therefore, the result of Theorem 2 is quite good news: it means that the use of cyclic intervals is not necessary to build an interval routing function on usual networks.

Now we characterize the graphs which belong to 1-LIRS strict.

3.3. *Characterization of* 1-*LIRS Strict*

DEFINITION 8. A *weak lithium-graph* is a graph with a least three bridges which connect the same connected component (the kernel) with three other distinct connected components (the electrons).

Any lithium-graph is a weak lithium-graph. A lithium-graph is indeed a weak lithium-graph where each of the electrons has at least two vertices (see Figure 2).

THEOREM 3. *$G \in$ 1-LIRS strict $\Leftrightarrow$ G is not a weak lithium-graph.*

PROOF. ($\Rightarrow$) Assume $G \in$ 1-LIRS strict, and suppose that G is a weak lithium-graph. Consider the vertices of the electrons which connect the electrons to the kernel. Necessarily, one of these vertices is not labeled 1 nor n (where n is the number of vertices of G). We call this vertex x, and let e be the bridge between x and the kernel. The interval $I_{x,e}$ is equal to $[1, n]$ otherwise the routing function would be not connected. However, since $1 < \mathcal{L}(x) < n$, we get that $I_{x,e}$ contains $\mathcal{L}(x)$, a contradiction.

($\Leftarrow$) If G is not a weak lithium-graph, then it is not a lithium-graph and we can decompose G as in the proof of Theorem 2 to obtain a "path" $G_0 - G_1 - \cdots - G_k$, where the G_i's are connected by strong bridges. Since G is not a weak lithium-graph, we get $\deg(x) \neq 1$ for every vertex x of G_i, $i \in \{1, \ldots, k-1\}$. Now, there is at most one vertex of degree 1 in G_0 and G_k if $k \geq 1$, and there are at most two such vertices if $k = 0$. This means that we can decompose G in a "path" $\{x\} - G'_0 - G_1 - \cdots - G'_k - \{y\}$ where x (resp. y) is the vertex of degree 1 of G_0 (resp. G_k) if it exists, and G'_0 (resp. G'_k) is obtained from G_0 (resp. G_k) by removing x (resp. y). We can apply the same construction as in the proof of Theorem 2 on the "path" $\{x\} - G'_0 - G_1 - \cdots - G'_k - \{y\}$. Since each component has edge-connectivity 2, the constructed routing function uses only strict linear intervals (from Lemma 2). $\square$

Theorem 3 has a direct simple consequence on the cartesian product. Recall that the cartesian product of a graph G by a graph H is the graph denoted by $G \times H$, whose vertices are elements of the cartesian product $V(G) \times V(H)$, and whose edges are the pairs $\{(x, y), (x', y')\}$ such that either $(x, x') \in E(G)$ or $(y, y') \in E(H)$.

COROLLARY 3. *Let G and H be two graphs of at least two vertices, then $G \times H \in$ 1-LIRS strict.*

Theorem 8 later takes advantage of Theorem 3 in a more significant way.

3.4. *Interval Routing With Extended Labels.* The memory space complexity of a router of degree d implementing an interval routing function of compactness k is bounded by $O(kd \log n)$ bits. One can argue that classes of graphs supporting interval routing functions with certain properties like compactness, strictness, shortest paths, etc., might be limited by the space labeling of the vertices. We define an interval routing function with *extended labels* on an n-vertex graph as an interval routing function whose labels are taken in $\{1, \ldots, n^c\}$, for some constant $c \geq 1$. Standard interval routing schemes correspond to $c = 1$. Clearly, any router implementing an interval routing function with extended labels still has a memory space of $O(kd \log n)$ bits because all the integers can be stored in at most $\lceil c \log_2 n \rceil = O(\log n)$ bits. Hence, it is natural to ask whether a larger labeling space allows us to increase the power of the standard interval routing scheme in term of routing ability. We answer this question negatively by the following theorem:

THEOREM 4. *Let G be a graph of order n, and let m be an integer such that $m \geq n$. Let $R = (\mathcal{L}, \mathcal{I})$ be an interval routing function on G with labels taken in* $\{1, \ldots, m\}$. *There exists an interval routing function R' on G such that*:

- *the set of routing paths induced by R' is the same as the one induced by R*;
- *the compactness of R' is at most the compactness of R*;
- *the linear and strictness properties of the intervals of R are preserved for the intervals of R'.*

PROOF. We define $R' = (\mathcal{L}, \mathcal{I})$ on $G = (V, E)$ as follows: for every $x \in V$, $\mathcal{L}'(x)$ is the rank of $\mathcal{L}(x)$ in the ordered set $\mathcal{L}(V)$ of all the labels of vertices of V. For every interval $I = [a, b]$ of $\mathcal{I}$, we construct an interval I' of $\mathcal{I}'$ as follows. First, we define a' and b' (if they exist) as the boundaries of the largest interval $[a', b']$ such that $[a', b'] \subset I$, and $[a', b'] \subset \mathcal{L}(V)$. For every $i \in \mathcal{L}(V)$, let $\mathcal{L}^{-1}(i)$ denote the unique vertex x such that $\mathcal{L}(x) = i$, and let $\psi(i) = \mathcal{L}'(\mathcal{L}^{-1}(i))$. We set

$$I' = \begin{cases} \emptyset & \text{if } a' \text{ and } b' \text{ do not exist,} \\ [\psi(a'), \psi(b')] & \text{otherwise.} \end{cases}$$

By construction, ψ is a strict increasing function on the set $\mathcal{L}(V)$. Thus for every interval $[\alpha, \beta] \subset \mathcal{L}(V)$, $\psi([\alpha, \beta]) = \{\psi(i) \mid i \in [\alpha, \beta]\} = [\psi(\alpha), \psi(\beta)]$. For every $x \in V$, $\mathcal{L}(x) \in I \Leftrightarrow \mathcal{L}(x) \in [a', b'] \Leftrightarrow \psi(\mathcal{L}(x)) \in \psi([a', b']) \Leftrightarrow \mathcal{L}'(x) \in [\psi(a'), \psi(b')] = I'$. Therefore, for every $x, y \in V$, $R_x(y) = R'_x(y)$ (we get of course $x = y \Leftrightarrow \mathcal{L}'(x) = \mathcal{L}'(y)$), that is, the set of induced routing paths is the same for both routing functions R and R'. Since, for every $x \in V$, $\mathcal{L}(x) \in I \Leftrightarrow \mathcal{L}'(x) \in I'$, the union and disjunction properties are clearly satisfied, as the linearity and the strictness of each interval. Finally, I' is composed of at most one interval (that can be removed if $I' = \emptyset$). Thus the compactness of R' is not greater than that of R. □

3.5. *Summary.* Figure 4 summarizes the results we obtained in this section. In the following section we study the length of the paths induced by an interval routing function.

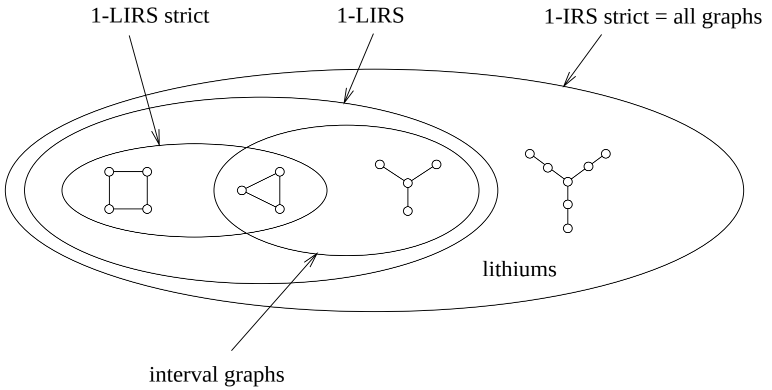

Fig. 4. Classes 1-IRS strict, 1-LIRS and 1-LIRS strict.

4. Efficiency of an Interval Routing Scheme. We have characterized graphs that support a (linear) interval routing function of compactness 1. Both constructions of Theorems 1 and 2 do not necessarily produce shortest paths between the sources and the destinations. In this section we study the tradeoff between the compactness of the routing and the length of the induced paths.

4.1. *Optimal Interval Routing Schemes*

DEFINITION 9 (Optimality). Let R be a routing function on a graph G. R is *optimal* if and only if the route built by R between any pair source–destination is of minimum length.

NOTATION. For every integer $k \geq 1$, we denote by k-IRS* the class of graphs which support an optimal interval routing function of compactness at most k. Similarly we denote by k-LIRS* the class of graphs which support an optimal linear interval routing function of compactness at most k.

DEFINITION 10 (Compactness of a Graph). The *compactness* of a graph G is the minimum taken over all the optimal interval routing functions R on G of the compactness of R. Similarly, the *linear-compactness* of G is the minimum taken over all the optimal linear interval routing functions R on G of the compactness of R.

In other words, the (linear-)compactness of a graph G is the smallest integer k such that $G \in k$-(L)IRS*. The compactness of G is an important parameter, since it measures the efficiency in term of memory complexity, of interval representations of optimal routing functions on G.

Frederickson and Janardan have showed in [10] that outer-planar graphs, that is, a subclass of planar graphs including trees, belong to 1-IRS* strict. In the following we present two large classes of graphs of compactness 1.

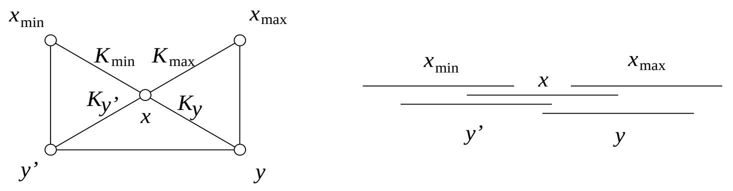

Fig. 5. Setting the intervals of a routing function on an interval graph.

4.1.1. *Families of Graphs Belonging to* 1-(*L*)*IRS**

THEOREM 5. *Every unit interval graph belongs to* 1-LIRS* *strict*.

A unit interval graph [16] is an interval graph in which all the intervals representing the vertices have the same length.

PROOF. Let G be any unit interval graph of order $n > 1$. Each vertex x of G is represented by a unit interval $J_x \subset \mathbb{R}$. For any vertex x let $\alpha(x)$ and $\beta(x)$ be such that $J_x = [\alpha(x), \beta(x)]$. We construct an optimal strict linear interval routing function $R = (\mathcal{L}, \mathcal{I})$ as follows.

We label the vertices from left to right based on the ranks of the $\alpha(x)$. Let x be any vertex of G, then let $x_{\min}$ (resp. $x_{\max}$) be the vertex satisfying $\alpha(x_{\min}) = \min_{y \neq x,\, J_y \cap J_x \neq \emptyset} \alpha(y)$ (resp. $\alpha(x_{\max}) = \max_{y \neq x,\, J_y \cap J_x \neq \emptyset} \alpha(y)$).

Assume first that $x_{\min} \neq x_{\max}$. Then let $K_{\min} = [1, \mathcal{L}(x_{\min})]$ and $K_{\max} = [\mathcal{L}(x_{\max}), n]$. Moreover, for any neighbor y of x distinct from $x_{\min}$ and $x_{\max}$, set $K_y = [\mathcal{L}(y)]$. Then set $I_{x,(x,x_{\min})} = K_{\min}$, $I_{x,(x,x_{\max})} = K_{\max}$, and $I_{x,(x,y)} = K_y$ (see Figure 5). We get that, for any y (if it exists), $K_{\min} \cap K_y = K_{\max} \cap K_y = K_{\min} \cap K_{\max} = \emptyset$. That is, the disjunction property is satisfied. Moreover, $K_{\min} \cup K_{\max} \cup (\bigcup_y K_y) = [1, n]$. That is, the union property is also satisfied. Finally the intervals $K_{\min}$, $K_{\max}$, and K_y do not contain $\mathcal{L}(x)$.

If $x_{\min} = x_{\max}$, then either $\mathcal{L}(x) = 1$ or $\mathcal{L}(x) = n$. In the former case (resp. latter case), the interval of the unique edge of extremity x is $]1, n]$ (resp. $[1, n[$).

Now, we prove the property $\mathcal{P}_k$: for every k, the routing function defined above builds a shortest path between any two vertices at distance at most k. $\mathcal{P}_1$ is true. Assume $\mathcal{P}_{k'}$ is true for all $k' \in \{1, \ldots, k-1\}$, and let x and y be two vertices at distance $k > 1$. Since all the intervals are of the same length, if $\mathcal{L}(x) < \mathcal{L}(y)$, then $x_{\max}$ is on a shortest path between x and y, and $y \in K_{\max}$; otherwise $\mathcal{L}(x) > \mathcal{L}(y)$, and $x_{\min}$ is on a shortest path between x and y, and $y \in K_{\min}$. Thus $\mathcal{P}_k$ is true since $\mathcal{P}_{k-1}$ is true. $\square$

Note that $C_4 \in$ 1-LIRS* strict but C_4 is not a unit interval graph. Similarly, the complete bipartite graph $K_{1,3} \in$ 1-LIRS* but is not a unit interval graph. Note also that there are interval graphs which do not belong to 1-LIRS* as, for instance, the graph on Figure 6(a). Indeed, assume it belongs to 1-LIRS*. Then let x and y be two vertices, both different from 1 and 7. $I_{x,(x,z)}$ must contain 1 and 7, thus $I_{x,(x,z)} = [1, 7]$, and the route from x to y is not a shortest path: a contradiction. Of course, it is easy to generalize the class of graphs on which such an argument can be applied. However, at the present time, no characterization of 1-LIRS* and 1-IRS* is known.

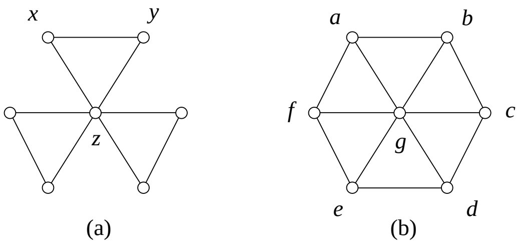

Fig. 6. (a) An interval graph which does not belong to 1-LIRS*, and (b) a circular-arc graph which does not belong to 1-IRS*.

Recently, Narayanan and Shende have shown [22] that all interval graphs belong to 1-IRS* strict. We give below another class of graphs belonging to 1-IRS* strict.

The reader can easily check that $C_5 \notin$ 1-LIRS* (this result is proved in [3] and is proved again later in this paper for any $C_n, n \geq 5$). On the other hand, one can also check that $C_n \in$ 1-IRS* for every n. This is a consequence of a more general result. Recall that G is a circular-arc graph if there exists a circle $\mathcal{C}$ such that each vertex x of G can be represented by an arc c_x of $\mathcal{C}$, and two vertices x and y of G are adjacent if and only if $c_x \cap c_y \neq \emptyset$. A unit circular-arc graph [16] is a circular-arc graph such that all arcs representing the vertices have the same length.

THEOREM 6. *Every unit circular-arc graph belongs to* 1-IRS* *strict.*

PROOF. Let G be a unit circular-arc graph. If G is a unit interval graph, then $G \in$ 1-LIRS* strict from Theorem 5. Assume G is not a unit interval graph. Consider a circular representation of G on the trigonometric circle. Once this representation is fixed, we can set an angle $\theta \in [0, 2\pi[$ as a measure of the length of the arcs representing the vertices. Each vertex x can also be represented by an angle $\theta_x \in [0, 2\pi[$ (for instance, the angle between the horizontal axis and the line joining the center of the circle to the middle of the arc representing x), and two vertices x and y are adjacent if and only if $(\theta_x - \theta_y) \bmod 2\pi$ is either in $[0, \theta] \cap [0, \pi]$ or in $[2\pi - \theta, 2\pi] \cap [\pi, 2\pi]$.

We label each vertex x by its rank in the set of all the angles $\{\theta_y \mid y \in V(G)\}$.

We set the intervals as follows. Let $x \in V(G)$, and let z be the vertex at distance the eccentricity of x (that is, the maximum distance between x and any other vertex of G) counterclockwise which maximizes the angle $\theta_z - \theta_x$. Then let x^- (resp. x^+) be the neighbor of x maximizing $\theta_y - \theta_x$ (resp. $\theta_x - \theta_y$) among the neighbors y of x. Since G is not an interval graph, $x^- \neq x^+$. We set $I_{x,(x,x^+)} =]\mathcal{L}(z), \mathcal{L}(x^+)]$, $I_{x,(x,x^-)} = [\mathcal{L}(x^-), \mathcal{L}(z)]$, and, for every neighbor y of x distinct from x^+ and x^-, set $I_{x,(x,y)} = [\mathcal{L}(y)]$. The union and the disjunction properties are clearly satisfied, and $\mathcal{L}(x)$ does not belong to $I_{x,e}, \forall e \in \mathrm{out}(x)$.

As in the proof of Theorem 5, that the paths built by the routing function are shortest paths can be verified by induction on the length of the paths. □

Note that $K_{1,3} \in$ 1-IRS* strict but $K_{1,3}$ is not a unit circular-arc graph. Note also that there are graphs in 1-IRS* strict which are not circular-arc graphs: for instance, the

Y-graph. Moreover, there are circular-arc graphs which do not belong to 1-IRS*. For instance, consider the graph of Figure 6(b). Assume it belongs to 1-IRS*, and consider the interval $I_{a,(a,g)}$. This interval must contain $\mathcal{L}(g)$ and $\mathcal{L}(d)$, but neither $\mathcal{L}(b)$ nor $\mathcal{L}(f)$. Similarly, $I_{c,(c,g)}$ must contain $\mathcal{L}(g)$ and $\mathcal{L}(f)$, but neither $\mathcal{L}(b)$ nor $\mathcal{L}(d)$. Finally, $I_{e,(e,g)}$ must contain $\mathcal{L}(g)$ and $\mathcal{L}(b)$, but neither $\mathcal{L}(d)$ nor $\mathcal{L}(f)$. All these conditions cannot hold simultaneously.

4.1.2. *Tools to Recognize Optimal Routing Schemes*. The two following results are useful for knowing whether a graph belongs to k-(L)IRS* for a fixed k. We recall the definition of a subgraph of shortest paths:

DEFINITION 11 (Subgraphs of Shortest Paths). A graph G' is a *subgraph of shortest paths* of a graph G if and only if G' is a partial subgraph of G, and all the shortest paths of G between any pair of vertices of G' are contained in G'.

THEOREM 7. *For every integer $k \geq 1$, and for every class of graphs $\mathcal{G}$ equal to k-LIRS* strict, k-LIRS*, k-IRS* strict, or k-IRS*,*

$$G \in \mathcal{G} \quad \Rightarrow \quad \forall G' \textit{ subgraph of shortest paths of } G,\ G' \in \mathcal{G}.$$

Theorem 7 says that the compactness of a graph is always at least the compactness of any of its subgraphs of shortest paths. Note that since G is a subgraph of shortest paths of itself, the converse property of Theorem 7 is of course satisfied.

PROOF. Let G' be a subgraph of shortest paths of $G \in \mathcal{G}$. Let $m = |V(G)|$ and $n = |V(G')|$. Let R be an optimal interval routing function on G. We show that G' supports an optimal interval routing function R' which makes $G' \in \mathcal{G}$. Since all the shortest paths between any pair of vertices of G' are wholly contained in G', R is an optimal interval routing on G' with extended labels taken in $\{1, \ldots, m\}$, $m \geq n$. From Theorem 4, there exists an interval routing function R' on G' which has the same induced routing paths, the same compactness, and the same linear and strictness properties as R. Therefore G' belongs to the same class as G. □

Theorem 7 is used to prove the next theorem that states results concerning cartesian products. Such structures are particularly interesting for the design of networks of processors (mesh, torus, hypercube, ...). In Section 5 we will see many applications of Theorem 7.

THEOREM 8. *For every integer $k \geq 1$,*

(i) $G \in k$-LIRS* *strict and* $H \in k$-LIRS* *strict* $\Rightarrow$ $G \times H \in k$-LIRS* *strict*;
(ii) $G \in k$-LIRS* *and* $H \in k$-LIRS* *strict* $\Rightarrow$ $G \times H \in k$-LIRS*;
(iii) $G \in k$-LIRS* *strict and* $H \in k$-IRS* *strict* $\Rightarrow$ $G \times H \in k$-IRS* *strict*;
(iv) $G \in k$-LIRS* *and* $H \in k$-IRS* *strict* $\Rightarrow$ $G \times H \in k$-IRS*;
(v) $G \notin k$-LIRS* $\Rightarrow$ $G \times H \notin k$-LIRS* *for any graph* H;
(vi) $G \notin k$-IRS* $\Rightarrow$ $G \times H \notin k$-IRS* *for any graph* H.

Note that we were not able to derive similar conservative properties for the product of two graphs belonging to k-IRS*, or for the product of two graphs belonging to k-IRS* strict. The linear property, and the strictness of the intervals are two relevant characteristics for cartesian products. To apply Theorem 8, one of the two graphs must support a strict interval routing function, and the other graph must support a linear interval routing function.

PROOF. The two last results (v) and (vi) are direct consequences of Theorem 7 since the graph $G \times H$ contains G as a subgraph of shortest paths.

Let $n_G = |V(G)|$ and $n_H = |V(H)|$.

To prove (i), let $R_G = (\mathcal{L}_G, \mathcal{I}_G)$ and $R_H = (\mathcal{L}_H, \mathcal{I}_H)$ be the interval routing functions defined on the graphs G and H, respectively, that makes G and $H \in k$-LIRS* strict. Let f be the one-to-one function $f\colon [1, n_G]\times[1, n_H] \mapsto [1, n_G \cdot n_H]$ defined by $f(a, b) = a + (b-1) \cdot n_G$. We define the labeling $\mathcal{L}$ of the vertices of $G\times H$ by $\mathcal{L}(z) = f(\mathcal{L}_G(x), \mathcal{L}_H(y))$, for every vertex $z = (x, y)$ of $G \times H$. Let $z = (x, y)$ and $z' = (x', y')$ be two adjacent vertices of $G \times H$, and let $e = (z, z')$. We set:

$$I_{z,e} = \begin{cases} [\,f(a, \mathcal{L}_H(y)),\ f(b, \mathcal{L}_H(y))] & \text{if} \quad y = y' \quad \text{where} \quad I_{x,(x,x')} = [a, b] \in \mathcal{I}_G \\ [\,f(1, c),\ f(n_G, d)] & \text{if} \quad x = x' \quad \text{where} \quad I_{y,(y,y')} = [c, d] \in \mathcal{I}_H \end{cases}$$

Clearly, if $[a, b] \cap [a', b'] = \emptyset$, then $[\,f(a, \mathcal{L}_H(y)),\ f(b, \mathcal{L}_H(y))] \cap [\,f(a', \mathcal{L}_H(y)),\ f(b', \mathcal{L}_H(y))] = \emptyset$. Similarly, if $[c, d] \cap [c', d'] = \emptyset$, then $[\,f(1, c),\ f(n_G, d)] \cap [\,f(1, c'),\ f(n_G, d')] = \emptyset$. Since $H \in k$-LIRS* strict, $\mathcal{L}_H(y) \notin [c, d]$, and thus $[\,f(a, \mathcal{L}_H(y)),\ f(b, \mathcal{L}_H(y))] \cap [\,f(1, c),\ f(n_G, d)] = \emptyset$. Therefore, the disjunction property is satisfied. It is trivial to check that the union property is also satisfied. The routing function $R = (\mathcal{L}, \mathcal{I})$ built on $G \times H$ as above routes messages as follows. The path from a vertex (x, y) to a vertex (x', y') goes first inside the xth copy of H toward the vertex (x, y'). Then it goes inside the y'th copy of G toward (x', y'). These two parts of the route use shortest paths, therefore $G \times H \in k$-LIRS*. Since $\mathcal{L}_G(x) \notin [a, b]$, $G \times H \in k$-LIRS* strict.

If $\mathcal{L}_G$ is not strict, we only get $G \times H \in k$-LIRS*, and property (ii) holds. If $\mathcal{I}_H$ contains a cyclic interval $[c, d]$, then the interval $[\,f(1, c),\ f(n_G, d)]$ is also cyclic and $G\times H \in k$-IRS*, that is property (iv) holds. Finally, if $\mathcal{L}_G$ is strict, we get $G \times H \in k$-IRS* strict, and property (iii) holds. □

Points (i) and (ii) of the previous theorem have been proved independently by Kranakis et al. in [19].

Theorems 7 and 8 are basic tools for computing the compactness of graphs. A third tool is described in Section 5. The last result of this section concerns the class of graphs which possess an optimal interval routing function. It shows that, up to a small increase of the compactness, an optimal strict interval routing function can always be designed.

PROPOSITION 1. $\forall k \geq 1$,

(i) $G \in k$-LIRS$^* \Rightarrow G \in (k+1)$-LIRS* *strict*;
(ii) $G \in k$-IRS$^* \Rightarrow G \in (k+1)$-IRS* *strict*.

PROOF. Let G be a graph which supports an optimal nonstrict interval routing function $R = (\mathcal{L}, \mathcal{I})$. Then, for any vertex x, there exists at most one edge $e \in \mathrm{out}(x)$ such that an interval $I_{x,e}$ contains the label $\mathcal{L}(x)$. We transform the routing function R in a strict interval routing function $R' = (\mathcal{L}, \mathcal{I}')$ in splitting all intervals $I_{x,e} = [a, b]$ containing the label of their local vertex x into two intervals $[a, \mathcal{L}(x)[$ and $]\mathcal{L}(x), b]$. The routes built by R and R' are the same. □

4.2. *Fast and Compact Interval Routing Schemes*. From a practical point of view, the designer of a routing system must balance the hardware constraints with the efficiency of the communications. Very fast communications may be required even up to a deficiency in hardware constraints (mainly increasing the surface of the routers), or this surface may be required to be as small as possible even up to degradation of the efficiency of communication.

4.2.1. *Minimum Time for a Fixed Compactness*

DEFINITION 12 (Time). Let R be a routing function on a graph G.

- The *time* of R is the maximum length of the paths built by R between any couple of vertices.
- The k-(*linear*)-*time* of a graph G is the minimum taken over all the (linear) interval routing functions R on G of compactness at most k, of the *time* of R.

The following result is a direct consequence of the proof of Theorem 1. It shows that interval routing schemes are relatively efficient. Indeed, although graphs are not always optimal as far as interval routing is concerned, it is possible to derive interval routing functions which build paths of small maximum length in any graph.

COROLLARY 4. *For every graph G of radius r, the* 1-time *of G is at most* $2r$.

PROOF. In the proof of Theorem 1, construct the spanning tree T as a shortest paths spanning tree of root any vertex of eccentricity r. □

This result is particularly interesting for a graph whose diameter is $2r$ as, for instance, the graphs of Figure 6. Note however that, in the routing scheme of Corollary 4, many messages go through a single vertex (the root of the tree). Therefore, many contentions might occur. This moderates the practical use of Corollary 4. Nevertheless, this upper bound is the best that we can hope for in term of radius because, for every r, there exists a graph of radius r and 1-time $2r$. For example, a path of $2r + 1$ vertices. Corollary 4 implies that the 1-time of a graph of diameter D is at most $2D$. However, we do not know if this upper bound is tight. For $k = 1$ and $k = 2$, we summarize below the most recent results about k-time as a function of the diameter.

THEOREM 9. *For every integer D,*

- *there exists a graph of diameter D of* 1-time *at least* $7D/4 - 1$ [29];
- *there exists a graph of diameter D of* 2-time *at least* $5D/4 - 1$ [28].

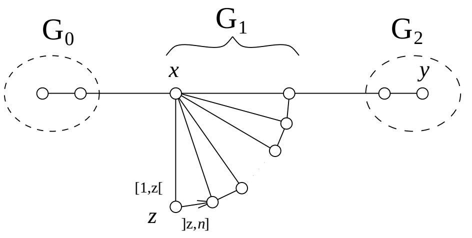

Fig. 7. A graph for which the constructive proof of Theorem 2 gives rise to a very inefficient routing function.

Many experimental results can be found in [17] where the author studies the 1-time of some graphs of compactness at least 2, like the torus. Still in [17], general results on the 3-time of arbitrary graphs are also presented.

Concerning linear interval routing schemes, the proof of Theorem 2 does not allow us to derive a tight upper bound on the 1-linear-time of a graph. Furthermore, Figure 7 shows a graph of order n for which the constructive proof of Theorem 2 gives rise to a very inefficient routing function: its time is $n-4$ (the route from z to y does not pass through x), whereas its diameter is only 5. Clearly, it would have been possible to design a much more efficient linear interval routing function for this graph (for example, using a shortest paths spanning tree rooted in x).

We give below a lower bound on the maximum 1-linear-time of a graph which is not a lithium-graph.

PROPOSITION 2. *For every integer D, there exists a nonlithium-graph of diameter D, and of* 1-linear-time *at least* $2D-1$.

The proof of this proposition is a direct consequence of the following lemma.

LEMMA 4. $\forall n \geq 3$, *the* 1-linear-time *of* C_n *is* $n-2$.

PROOF. If $n=3$ (resp. 4), then there exists an interval routing function which has a time of 1 (resp. 2), and is the best that can be done (1 and 2 are the diameters of C_3 and C_4, respectively).

We show that, for every $n \geq 5$, the 1-linear-time τ of C_n is $\geq n-2$. Assume $\tau < n-2$, and let $R = (\mathcal{L}, \mathcal{I})$ be a linear interval routing function of time τ. Let P be the path y'-x'-t-x-y in C_n where $\mathcal{L}(t) = 1$. Let z be the vertex of P which has the greatest label among all the vertices of P.

If $z = x$, then the interval $I_{x',(x',t)}$ must contain $\mathcal{L}(t)$ and $\mathcal{L}(z)$ because, otherwise, the time to reach t or z from x' would be $\geq n-2$ via the edge (x', y'). For the same reason $I_{x',(x',t)}$ cannot contain $\mathcal{L}(y')$. However, $\mathcal{L}(y') \in [\mathcal{L}(t), \mathcal{L}(z)] \subset I_{x',(x',t)}$: a contradiction.

If $z = y$, then the interval $I_{t,(t,x)}$ must contain $\mathcal{L}(x)$ and $\mathcal{L}(z)$, but not $\mathcal{L}(x')$. If $\mathcal{L}(x') > \mathcal{L}(x)$, then $\mathcal{L}(x') \in [\mathcal{L}(x), \mathcal{L}(z)]$: a contradiction. If $\mathcal{L}(x') < \mathcal{L}(x) < \mathcal{L}(y')$, then we get another contradiction with the fact that $I_{t,(t,x')}$ must contain $\mathcal{L}(x')$ and $\mathcal{L}(y')$, but not $\mathcal{L}(x)$. Finally, if $\mathcal{L}(x') < \mathcal{L}(x)$, and $\mathcal{L}(y') < \mathcal{L}(x)$, then again we get a contradiction because $I_{x',(x',t)}$ must contain $\mathcal{L}(t)$ and $\mathcal{L}(x)$, but not $\mathcal{L}(y')$.

If $z = x'$, or $z = y'$, we obtain similar contradictions by reversing the roles of vertices x, and y on one hand, and x' and y' on the other hand. Thus $\tau \geq n - 2$. Actually there exists a strict linear interval routing function on C_n of 1-linear-time $n - 2$: label the vertices clockwise from 1 to n, set $I_{i,(i,i+1)} = [i + 1, n]$ for $1 \leq i < n$, and $I_{n,(n,1)} = [1]$, set $I_{i,(i,i-1)} = [1, i - 1]$ for $1 < i \leq n$, and set $I_{1,(1,n)} = [n]$. □

Lemma 4 combined with Theorem 7 yields another proof of a result in [3]:

COROLLARY 5. $C_n \in$ 1-LIRS* $\Leftrightarrow n < 5 \Leftrightarrow C_n \in$ 1-LIRS* *strict*.

Recent works in [4] have showed that the 1-linear-time of some graphs of diameter D can be as bad as $\Omega(D^2)$. However, finding tight bounds on the worst-case 1-linear-time of a graph remains open. The following result allows, under some conditions, the derivation of the time of a cartesian product of graphs.

THEOREM 10. *Let k be any integer, let G be a graph of k*-linear-time t_G, *and let H be a graph which supports an interval routing function R of compactness k and* time t_H. *Then,*

- *if R is a strict interval routing function, then $G \times H$ has a k*-time *at most $t_G + t_H$;*
- *if R is a strict linear interval routing function, then $G \times H$ has a k*-linear-time *at most $t_G + t_H$.*

PROOF. Consider the routing function constructed in the proof of Theorem 8. The two conditions for the construction are satisfied: $G \in k$-LIRS*, and the routing function R is strict. Clearly the time of this routing function is at most $t_G + t_H$. □

4.2.2. *Minimum Compactness for an Optimal Time.* For any graph G, let IRS(G) denote the compactness of G, and let LIRS(G) denote its linear-compactness. The following theorem improves a result stated in [30]:

THEOREM 11. *For every graph G of order $n \geq 2$,* LIRS$(G) \leq n/2$, *and* IRS$(G) \leq (n - 1)/2$.

PROOF. It is always possible to design a routing table which correspond to a shortest paths routing function. For any vertex x, there are at most $n - 1$ destinations associated to an edge of out(x). These destinations can be encoded by at most $\lfloor n/2 \rfloor$ linear intervals, and by at most $\lfloor (n - 1)/2 \rfloor$ cyclic intervals (recall that intervals are not necessarily strict). □

As for the time, it is possible to derive the value of the compactness of a cartesian product of graphs.

THEOREM 12. *For any graphs G and H:*

- $\max\{\text{LIRS}(G), \text{LIRS}(H)\} \leq \text{LIRS}(G \times H) \leq \max\{\text{LIRS}(G), \text{LIRS}(H) + 1\}$;
- $\max\{\text{IRS}(G), \text{IRS}(H)\} \leq \text{IRS}(G \times H) \leq \max\{\text{LIRS}(G), \text{IRS}(H) + 1\}$.

PROOF. Since G and H are both subgraphs of shortest paths of $G \times H$, it is a direct consequence of Theorem 7 that $\mathrm{LIRS}(G \times H) \geq \max\{\mathrm{LIRS}(G), \mathrm{LIRS}(H)\}$, and that $\mathrm{IRS}(G \times H) \geq \max\{\mathrm{IRS}(G), \mathrm{IRS}(H)\}$. Let $k_L = \mathrm{LIRS}(H)$, and let $k_I = \mathrm{IRS}(H)$. From Proposition 1, $H \in (k_L + 1)$-LIRS* strict, and $H \in (k_I + 1)$-IRS* strict. Let $k = \max\{\mathrm{LIRS}(G), k_L + 1\}$, and let $k' = \max\{\mathrm{LIRS}(G), k_I + 1\}$. We have, on the one hand, $G \in k$-LIRS* and $H \in k$-LIRS* strict, and, on the other hand, $G \in k'$-LIRS* and $H \in k'$-IRS* strict. By application of Theorem 8, we get that $G \times H \in k$-LIRS* and $G \times H \in k'$-IRS*. □

The following result show that, unfortunately, there does not exist any constant upper bound of the value of the compactness of a graph.

THEOREM 13. *$\forall k \geq 1, \exists G$ such that $G \notin k$-LIRS*.*

PROOF. The graph drawn in Figure 6(a) provides an example for $k = 1$.

Let $k \geq 2$, and consider the graph G composed of three isomorphic components G_1, G_2, and G_3, all connected to a single vertex u (see Figure 8). Each G_i has three "levels." The first level is composed of $2k - 1$ independent vertices $x_1, \ldots, x_{2k-1}$, each x_i being connected to u. The second level is composed of $\binom{2k-1}{k-1}$ independent vertices denoted $(i_1, \ldots, i_{k-1})$, for $1 \leq i_1 < \cdots < i_{k-1} \leq 2k - 1$. The vertex $(i_1, \ldots, i_{k-1})$ is connected to the $k - 1$ vertices $x_{i_1}, \ldots, x_{i_{k-1}}$. The third level is a complete graph of the same number of vertices as in the second level. There is a one-to-one connection between the second and the third level. We denote by n the order of G. Assume $G \in k$-LIRS*, and let $R = (\mathcal{L}, \mathcal{I})$ be the corresponding routing function. Consider the subgraph G_i such that none of its vertices is labeled 1 or n (say the graph G_1 in Figure 8). Since the x_i's play the same role, assume that $\mathcal{L}(x_1) < \cdots < \mathcal{L}(x_{2k-1})$. Then consider the edge e connecting the vertex $z = (2, 4, 6, \ldots, 2k-2)$ of the second level with its corresponding vertex of the third level, denoted by y. From the structure of G, we get that 1, n, and $\mathcal{L}(x_{2i})$ belong to $I_{y,e}$, for every $i \in \{1, \ldots, k - 1\}$. Similarly, $\mathcal{L}(x_{2i-1}) \notin I_{y,e}$, for every

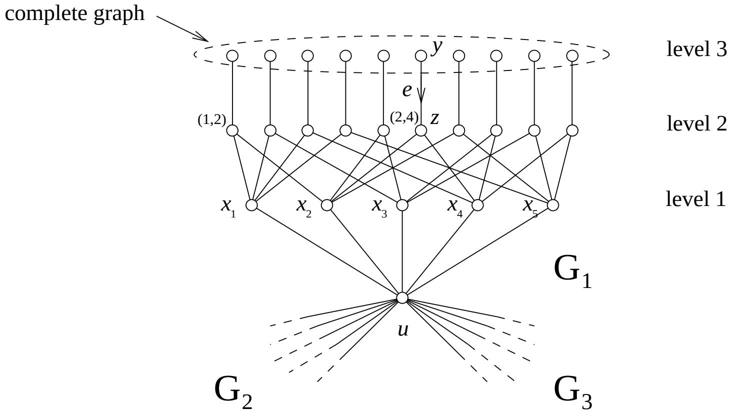

Fig. 8. A graph which does not belong to 3-LIRS*.

$i \in \{1, \ldots, k\}$. Thus $I_{y,e}$ is composed of at least $k+1$ linear intervals: a contradiction. Thus $G \notin k$-LIRS*. □

COROLLARY 6. *$\forall k \geq 1$, $\exists G$ such that $G \notin k$-*IRS*.

PROOF. $\forall k \geq 2$, $(k-1)$-IRS $\subset k$-LIRS. We complete the proof by applying Theorem 13. □

REMARK. The graph used in the proof of Theorem 13 has $n = 3(2k - 1 + 2\binom{2k-1}{k-1}) + 1$ vertices, that is $\Theta(4^k/\sqrt{k})$. Its compactness is thus at least $\Omega(\log n)$. Its diameter is 6, and its maximum degree $\binom{2k-1}{k-1} - 1 = \Theta(n)$. Many works deal with the asymptotic behavior of the maximum compactness of a graph of order n [6], [13], [14], [18], [19]. To our knowledge, the best results are a tight bound of $\Theta(n)$ for cubic graphs, and a lower bound of $\Omega(\sqrt{n})$ for cubic planar graphs [14].

We are now ready to study the properties satisfied by the usual graphs considered as candidates for interconnecting processors of parallel distributed memory computers.

5. Usual Networks.

In this section we describe constructions of (linear) interval routing functions designed for many usual network as meshes, hypercubes, and shuffle-exchange.

5.1. *Paths, Cycles, and Complete Graphs.* All these networks have already been considered in this paper. We refer to Lemma 4 and Corollary 5. C_n is a unit circular-arc graph. The path with n vertices P_n and the complete graph K_n are both unit interval graphs. According to Theorems 6 and 5, $C_n \in$ 1-IRS* strict, whereas P_n and $K_n \in$ 1-LIRS* strict.

5.2. *Meshes.* In [3] it is proved that the n-dimensional mesh belongs to 1-LIRS*. The following proposition simplifies their proof:

PROPOSITION 3. *The n-dimensional mesh $P_{d_1} \times P_{d_2} \times \cdots \times P_{d_n} \in$ 1-LIRS* strict.*

PROOF. For every m, P_m belongs to 1-LIRS* strict. Thus, we can apply $n-1$ times Theorem 8. □

5.3. *Generalized Hypercubes.* In [3] it is proved that the n-dimensional binary hypercube belongs to 1-LIRS*. The following proposition generalizes this result:

PROPOSITION 4. *The generalized hypercube H_n^d, with n dimensions on an alphabet of $d \geq 2$ letters, belongs to* 1-LIRS* *strict.*

PROOF. The generalized hypercube is recursively defined by $H_1^d = K_d$ and $H_n^d = H_{n-1}^d \times K_d$. That is, $H_n^d = K_d \times \cdots \times K_d$, n times. Now, $K_d \in$ 1-LIRS* strict, thus we can apply Theorem 8 $n-1$ times. □

5.4. *Torus*. In [3] it is shown that the n-dimensional torus $T_n = C_{d_1} \times C_{d_2} \times \cdots \times C_{d_n} \in$ 1-LIRS* if and only if $d_i < 5$ for every $i \in \{1, \ldots, n\}$. The next theorem generalizes this result.

THEOREM 14. *Let $T_n = C_{d_1} \times C_{d_2} \times \cdots \times C_{d_n}$ be an n-dimensional torus such that $d_1 \leq \cdots \leq d_n$. We have*:

(i) $T_n \in$ 2-LIRS* *strict*;
(ii) $T_n \in$ 1-LIRS* $\Leftrightarrow$ $d_n < 5$ $\Leftrightarrow$ $T_n \in$ 1-LIRS* *strict*;
(iii) $T_n \in$ 1-IRS* $\Leftrightarrow$ $d_{n-1} < 5$ $\Leftrightarrow$ $T_n \in$ 1-IRS* *strict*.

PROOF. For any n, $C_n \in$ 1-IRS* strict. Therefore $C_n \in$ 2-LIRS* strict, and thus, from Theorem 8, we obtain (i). Result (ii) is obtained by application of Corollary 5 and Theorem 8. If $d_{n-1} < 5$, then $T_{n-1} = C_{d_1} \times C_{d_2} \times \cdots \times C_{d_{n-1}} \in$ 1-LIRS* strict (from (ii)). Since $C_{d_n} \in$ 1-IRS* strict, we can apply Theorem 8. The reciprocal of (iii) is stated in [24], i.e., $C_{d_1} \times C_{d_2} \notin$ 1-IRS* for $d_1 \geq 5$. □

5.5. *A List of Usual Networks Which Do Not Belong to* 1-*LIRS**. In [30] van Leeuwen and Tan asked the question whether there is an optimal interval routing function for any arbitrary graph. Ružička already answered this question in a negative way in [25] by showing a graph of 1-time at least three-half its diameter. In this section we present a list of graphs which are not in 1-LIRS*. In Section 5.5.1 we use Theorem 7, and in Section 5.5.2 we present a new tool for checking the optimality of an interval routing function.

5.5.1. *Using a Subgraph of Shortest Paths*

PROPOSITION 5. *The following graphs do not belong to* 1-LIRS*:

- *the* Shuffle-Exchange [20], [27], SE_n, $\forall n \geq 3$;
- *the* Cube-Connected-Cycle [23], CCC_n, $\forall n \geq 2$;
- *the* Star-Graph [1], S_n, $\forall n \geq 1$.

PROOF. All these graphs contain a cycle of at least five vertices as a subgraph of shortest paths. More precisely:

- For every $n \geq 5$, SE_n contains C_5 as a subgraph of shortest paths. Indeed, with the standard binary representation of the vertices, $C_5 = \{01x10, 01x11, 1x110, x1101, x1100\}$ where $x = 1^{n-4}$. It is easy to check that it is a subgraph of shortest paths. SE_4 contains C_7 as a subgraph of shortest paths, and $SE_3 \notin$ 1-LIRS* as we see later in Proposition 6.
- For every $n \geq 2$, CCC_n contains C_8 as a subgraph of shortest paths.
- For every $n \geq 1$, S_n contains C_6 as a subgraph of shortest paths.

Hence we get the result by applying Theorem 7 together with Lemma 4. □

5.5.2. *A New Tool for Checking the Optimality of an Interval Routing Function.* Let $R = (\mathcal{L}, \mathcal{I})$ be an interval routing function on a graph G. We denote by $T_{x,e}$ the set of labels $\mathcal{L}(y)$ such that all the shortest paths between vertices x and y traverse the edge $e \in \text{out}(x)$. For any subset A of $\{1, \ldots, n\}$, we denote by $[A]$ the smallest linear interval which contains all the elements of A, i.e., $[A] = [\min_{x \in A}(x), \max_{x \in A}(x)]$. Of course, if $G \in$ 1-LIRS*, then $\forall x \in V(G)$, $\forall e \neq e' \in \text{out}(x)$, $T_{x,e} \cap T_{x,e'} = \emptyset$ for any routing function which makes G in 1-LIRS*. In fact we can get a stronger result:

LEMMA 5. *$G \in$ 1-LIRS* $\Rightarrow$ $\forall x \in V(G)$, $\forall e \neq e' \in \text{out}(x)$, $[T_{x,e}] \cap [T_{x,e'}] = \emptyset$ for any routing function which makes G in* 1-LIRS*.

PROOF. Let $R = (\mathcal{L}, \mathcal{I})$ be a linear interval routing function on G such that $G \in$ 1-LIRS*. Clearly, for any interval $I_{x,e} \in \mathcal{I}$, $[T_{x,e}] \subset I_{x,e}$. We conclude by applying the disjoint property. □

For any two subsets A and B of $\{1, \ldots, n\}$, we say that A and B are *separable* if and only if $[A] \cap [B] = \emptyset$ (that is, either $\forall (a, b) \in A \times B$, $a < b$, or $\forall (a, b) \in A \times B$, $a > b$). We denote by $A \mid B$ the property "A and B are separable." We get a method to prove that a given graph G does not belong to 1-LIRS*. It is sufficient to find a subset V' of vertices of $V(G)$ such that the system of equations $T_{x,e_i} \mid T_{x,e_j}$, $\forall x \in V'$ and $\forall e_i \neq e_j \in \text{out}(x)$, leads to a contradiction whatever the labeling. Such a system of equations is said to be *induced* by V'. It can be expressed independently of the labeling. The goal is to show that there is no labeling compatible with the system. For instance:

PROPOSITION 6. *The following graphs do not belong to* 1-LIRS*:

- *the* Shuffle-Exchange SE_3;
- *the* 6-*directional* Mesh (*see Figure* 9(*b*));
- *the* 8-*directional* Mesh (*see Figure* 9(*c*));
- *the* Butterfly [20], BF_n, $\forall n \geq 2$ (*see Figure* 9(*d*));
- *all the other graphs drawn in Figure* 9.

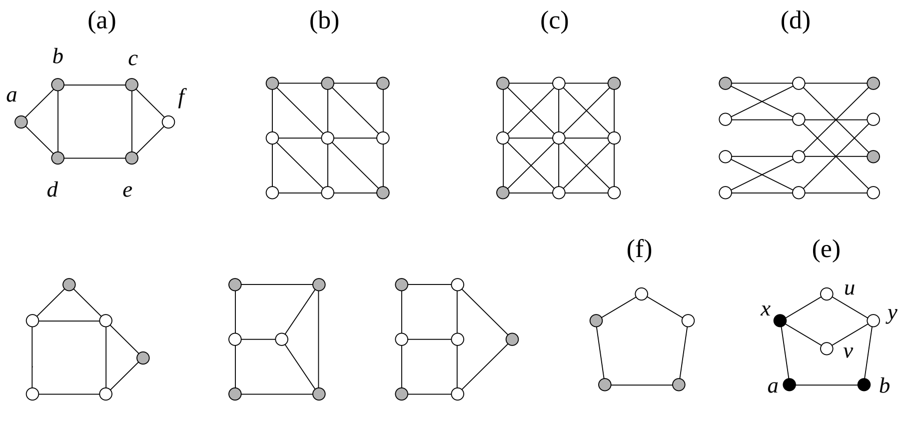

Fig. 9. Some graphs which do not belong to 1-LIRS*.

PROOF. For the graph SE_3, it is sufficient to prove that the graph G drawn in Figure 9(a) is not in 1-LIRS*. Indeed, this graph is a subgraph of the shortest paths of SE_3. Assume that the vertices of G have been arbitrary labeled $a, b, \ldots, f$ by an optimal linear interval routing function of compactness 1. We then consider the system induced by the vertices in gray in Figure 9(a):

$$\begin{array}{ll} \text{in } a: & bc|de, \\ \text{in } b: & a|cf, \\ \text{in } c: & e|ab, \\ \text{in } d: & a|ef, \\ \text{in } e: & c|ad. \end{array}$$

Assume $b < e : b < e \overset{(a)}{\Longrightarrow} c < d \overset{(e)}{\Longrightarrow} c < a \overset{(b)}{\Longrightarrow} f < a \overset{(d)}{\Longrightarrow} e < a \overset{(c)}{\Longrightarrow} e < b :$ contradiction (the assumption $b > e$ would also lead to a contradiction by symmetry). Thus, from Theorem 7, $SE_3 \notin$ 1-LIRS*.

For every $n \geq 3$, BF_n contains BF_2 as a subgraph of shortest paths. Since $BF_2 \notin$ 1-LIRS*, by looking at the system induced by the three vertices in gray in Figure 9(d), we get $BF_n \notin$ 1-LIRS*. For all the other graphs, the reader can check that the result holds by looking at the system induced by the vertices in gray in Figure 9. □

REMARK. An easy way to find a solution of a system $\{(A_i \mid B_i)_i\}$ obtained from graph G is to consider a digraph R_G associated to the system such that

$$V(R_G) = \bigcup_i (A_i \cup B_i) \quad \text{and} \quad E(R_G) = \bigcup_i \{(a, b) \in A_i \times B_i \mid a < b\}.$$

We consider the transitive closure $\overline{R_G}$ of R_G. If $\overline{R_G}$ possesses a cycle, then $G \notin$ 1-LIRS*. If $\overline{R_G}$ is acyclic, then it does not prove that $G \in$ 1-LIRS*. However, sorting the vertices of $\overline{R_G}$ by outer degree can give indications about a possible labeling: label the vertex of the smallest outer degree with 1, the vertex with the second smallest outer degree with 2, and so on. With this method we found several labelings for the graphs in 1-LIRS* drawn in Figure 10 (once the labeling is given, the interval setting is easy by using a greedy algorithm). Using this approach, we found some counterexamples to the affirmation stated in [3] that any combination of more than one square with one triangle sharing a common face cannot be in 1-LIRS*.

Even if Lemma 5 seems useful in general to determine whether a graph is in 1-LIRS* or not, one cannot hope to characterize the class 1-LIRS* completely with this method. For instance, one cannot prove with this method whether the graph draw on Figure 9(e)

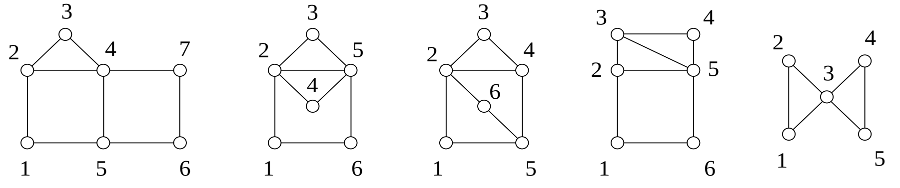

Fig. 10. Some graphs which belong to 1-LIRS*.

does or does not belong to 1-LIRS*. Indeed, the reader can check that any equation system has a solution even if this graph does not belong to 1-LIRS*. Indeed, from x, there are two possible equations: $ab \,|\, uy$ or $ab \,|\, vy$, depending on the choice of the shortest path of the routing function from x to y. Assume the shortest path from x to y is fixed, and consider the equation system induced by the black vertices of the graph of Figure 9(e) for both choices of the shortest path from x to y. Whatever this choice is, the corresponding system will be the same as the system induced be the gray vertices of C_5 (drawn in Figure 9(f)), which has no solution. Although Flammini recently proved in [5] that answering whether a graph G does or does not belong to 2-LIRS* strict is NP-complete, the full characterization of the class 1-LIRS* remains an open problem.

Note that the graph in Figure 9(b) contains the graph of Figure 6(b) as subgraph of shortest paths. The graph of Figure 6(b) does not belong to 1-IRS*, therefore 6-directional meshes do not belong to 1-LIRS* neither to 1-IRS*.

Acknowledgments. The authors are grateful to Eric Fleury, Jean-Claude König, and Claudine Peyrat for many helpful remarks.

References

[1] S. B. Akers, D. Harel, and B. Krishnamurthy. The star graph: an attractive alternative to the n-cube. In *Proceedings of the International Conference on Parallel Processing* (*ICPP*), pages 393–400, St. Charles, Illinois, August 1987.

[2] B. Awerbuch, A. Bar-Noy, N. Linial, and D. Peleg. Improved routing strategies with succint tables. *Journal of Algorithms*, 11:307–341, February 1990.

[3] E. M. Bakker, J. van Leeuwen, and R. B. Tan. Linear interval routing. *Algorithms Review*, 2:45–61, 1991.

[4] T. Eilam, S. Moran, and S. Zaks. A lower bound for linear interval routing. In O. Babaoglu and K. Marzullo, editors, *Proceedings of the* 10*th International Workshop on Distributed Algorithms* (*WDAG*), pages 191–205. Lecture Notes in Computer Science, volume 1151. Springer-Verlag, Berlin, October 1996.

[5] M. Flammini. On the hardness of devising interval routing schemes. *Parallel Processing Letters*, 1997. To appear.

[6] M. Flammini, J. van Leeuwen, and A. Marchetti-Spaccamela. The complexity of interval routing on random graphs. In J. Wiederman and P. Hájek, editors, *Proceedings of the* 20*th International Symposium on Mathematical Foundations of Computer Sciences* (*MFCS*), pages 37–49. Lecture Notes in Computer Science, volume 969. Springer-Verlag, Berlin, August 1995.

[7] E. Fleury and M. Picquendar. Massivelly parallel machine based on T9000 and C104. Research Report 94-14, LIP, École Normale Supérieure de Lyon, 69364 Lyon Cedex 07, May 1994. (Short version in Transputers '94.)

[8] P. Fraigniaud and C. Gavoille. Interval routing schemes. Research Report 94-04, LIP, École Normale Supérieure de Lyon, 69364 Lyon Cedex 07, January 1994.

[9] P. Fraigniaud and E. Lazard. Methods and problems of communication in usual networks. *Discrete Applied Mathematics*, 53:79–133, 1994.

[10] G. N. Frederickson and R. Janardan. Designing networks with compact routing tables. *Algorithmica*, 3:171–190, 1988.

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

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

[13] C. Gavoille and E. Guévremont. On the compactness of bounded degree graphs for shortest path interval routing. In L. M. Kirousis and E. Kranakis, editors, *Proceedings of the* 2*nd Colloquium on Structural Information & Communication Complexity* (*SIROCCO*), pages 113–121. Carleton University Press, Carleton, June 1995.

[14] C. Gavoille and S. Pérennès. Lower bounds for interval routing on 3-regular networks. In *Proceedings of the* 3*rd Colloquium on Structural Information & Communication Complexity* (*SIROCCO*), pages 88–103, June 1996.

[15] P. C. Gilmore and A. J. Hoffman. A characterization of comparability graphs and of interval graphs. *Canadian Journal of Mathematics*, 16:539–548, 1964.

[16] M. C. Golumbic. *Algorithmic Graph Theory and Perfect Graphs*. Academic Press, New York, 1980.

[17] M Juganaru. Routage avec table de routage compacte. Stage de DEA, ENSIMAG, Grenoble, June 1993.

[18] E. Kranakis and D. Krizanc. Lower bounds for compact routing. In *Proceedings of the* 13*th Annual Symposium on Theoretical Aspects of Computer Science* (*STACS*), pages 529–540, February 1996.

[19] E. Kranakis, D. Krizanc, and S. S. Ravi. On multi-label linear interval routing schemes. In *Proceedings of the* 19*th International Workshop on Graph-Theoretic Concepts in Computer Science—Distributed Algorithms* (*WG*), pages 338–349. Lecture Notes in Computer Science, volume 790. Springer-Verlag, Berlin, June 1993.

[20] F. T. Leighton. *Introduction to Parallel Algorithms and Architectures*: *Arrays - Trees - Hypercubes*. Morgan Kaufmann, Los Altos, CA, 1992.

[21] M. D. May, P. W. Thompson, and P. H. Welch. Networks, routers and transputers: function, perfomance, and applications. Technical report, Inmos, SGS-THOMSON, 1993.

[22] L. Narayanan and S. Shende. Characterizations of networks supporting shortest-path interval labeling schemes. In *Proceedings of the* 3*rd Colloquium on Structural Information & Communication Complexity* (*SIROCCO*), pages 73–87, June 1996.

[23] F. Preparata and J. Vuillemin. The cube connected cycles: a versatile network for parallel computation. *Communications of the ACM*, 24(5):300–309, 1981.

[24] P. Ružička. On efficient interval routing algorithms. In M. P. Chytil, L. Janiga, and V. Koubek, editors, *Mathematical Foundations of Computer Science* (*MFCS*), pages 492–500. Lectures Notes in Computer Science, volume 324. Springer-Verlag, Berlin, 1988.

[25] P. Ružička. A note on the efficiency of an interval routing algorithm. *The Computer Journal*, 34:475–476, 1991.

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

[27] H. S. Stone. Parallel processing with the perfect shuffle. *IEEE Transactions on Computers*, 20(2):153–161, 1971.

[28] S. S. H. Tse and F. C. M. Lau. Lower bounds for multi-label interval routing. In L. M. Kirousis and E. Kranakis, editors, *Proceedings of the* 2*nd Colloquium on Structural Information & Communication Complexity* (*SIROCCO*), pages 123–134. Carleton University Press, Carleton, June 1995.

[29] S. S. H. Tse and F. C. M. Lau. A lower bound for interval routing in general networks. *Networks*, 29(1):49–53, January 1997.

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