Laboratoire de l'Informatique du Parallélisme

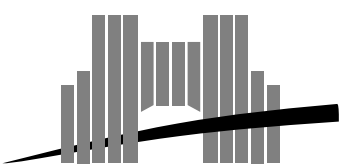
Ecole Normale Supérieure de Lyon
Unité de recherche associée au CNRS n°1398

Interval Routing Schemes

Pierre Fraigniaud
Cyril Gavoille

January 1994

Research Report N^{o} 94-04



Ecole Normale Supérieure de Lyon
46, Allée d'Italie, 69364 Lyon Cedex 07, France,
Téléphone : + 33 72 72 80 00; Télécopieur : + 33 72 72 80 80;
Adresse électronique : lip@lip.ens−lyon.fr

Interval Routing Schemes

Pierre Fraigniaud
Cyril Gavoille

January 1994

Abstract

Interval routing was introduced to reduce the size of the routing tables: a router finds the direction to forward a message by determining the interval that contains the destination address of the message, each interval being associated to one particular direction. This way of implementing routing function is quite attractive but very few was known about the topological properties that should satisfy a network to admit an interval routing function satisfying particular constraints (shortest paths, limited number of intervals associated to each direction, etc). In this paper, we deeply investigate the study of the interval routing functions. In particular we characterize the networks that admit 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 that admit an *optimal* (linear) interval routing function. Finally, we derive the main properties that satisfy the popular networks used to interconnect the processors of a distributed memory parallel computer.

Keywords: routing function, intervals, interconnected networks, distributed memory parallel computer

Résumé

Le routage par intervalles a été introduit pour réduire la taille des tables de routages : un routeur trouve la direction pour communiquer un message en déterminant l'intervalle contenant l'adresse de destination du message, chaque intervalle étant associé à une direction particulière. Cette méthode d'implémentation des fonctions de routage est vraiment attractive, mais très peu de choses sont connues des propriétés topologiques que doit satisfaire un réseau pour admettre une fonction de routage par intervalles réalisant des contreintes particulières (les plus courts chemins, limitation du nombre d'intervalles pour chaque direction, etc). Dans ce papier, nous étudions en profondeur les fonctions de routages par intervalles. En particulier nous caractérisons les réseaux qui admettent une fonction de routage par intervalles *linéaires* ou *strictement linéaires*, avec seulement un intervalle par direction. Nous donnons également des critères pratiques pour mesurer l'efficacité d'une fonction de routage par intervalles (nombre d'intervalles, longueur des chemins, etc), et nous décrivons une grande classe de graphes admettant une fonction de routage *optimale* par intervalles (linéaires). Finalement, nous décrivons les principales propriétés que réalisent les réseaux usuels utilisés pour la connexion des processeurs dans la plupart des ordinateurs parallèles à mémoires distribuées.

Mots-clés: fonctions de routage, intervalles, réseaux d'interconnexion, ordinateurs parallèles à mémoires distribuées

Interval Routing Schemes

Pierre Fraigniaud* and Cyril Gavoille †

LIP - CNRS
Ecole Normale Supérieure de Lyon
69364 Lyon Cedex 07, France

{pfraign,gavoille}@lip.ens-lyon.fr

January 15, 1994

Abstract

Interval routing was introduced to reduce the size of the routing tables: a router finds the direction to forward a message by determining the interval that contains the destination address of the message, each interval being associated to one particular direction. This way of implementing routing function is quite attractive but very few was known about the topological properties that should satisfy a network to admit an interval routing function satisfying particular constraints (shortest paths, limited number of intervals associated to each direction, etc). In this paper, we deeply investigate the study of the interval routing functions. In particular we characterize the networks that admit 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 that admit an *optimal* (linear) interval routing function. Finally, we derive the main properties that satisfy the popular networks used to interconnect the processors of a distributed memory parallel computer.

*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 C3.

1 Introduction

Given a network of processors (it could be the one of a distributed memory machine), the way of routing messages among these processors is characterized by the routing *mode* (store-and-forward, circuit-switched, wormhole,...) and the routing *function* that determines the paths between the sources and the destinations. This paper focuses on this second parameter.

The routing function is generally implemented locally on the routers. The route of a message among the processors from its source to its destination is determined using its header that contains several informations that will help the routers to find the route. In this paper, we are interested in routing functions that use only the destination address of the message to find the route.

Once a router receives a message, it looks at its header to know the destination of the message and finds the output port that will be used to forward the message towards 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.

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 a torus, or the e-cube routing on the hypercube: the output port is found by comparing the current address of the router to 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 find locally the path to the destination (especially if one insists on shortest paths for instance). By a simple algorithm, we mean an algorithm whose both execution time and space for implementing it on the router are small. If the interconnection network has no particular structure, it can be even impossible to derive any kind of algorithm. A solution to these problems is obtained by the use of routing tables that are stored locally on each router. Of course, 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 is larger that some tens).

Compact routing has already been intensively studied (see for instance [2, 7, 8]). In particular, there exist many solutions to compress the size of the routing tables. The general idea is to group in some manner the destination addresses that correspond to the same output port, and to encode the group so that it is easy to check if a destination address belongs to a given group. A very popular solution of that kind is the use of intervals [17]. They are indeed very simple to code (only store the bounds of each interval) and at most two comparisons are enough to check if a destination address belongs to an interval. This kind of routing is used for instance on the C104 routing chip [13, 4] of Inmos.

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

The notion of interval routing has been introduced by Santoro and Khatib in [17]. They mainly show that any directed acyclic graph can support an interval routing function with shortest paths

and only one interval per output port. Moreover, if the digraph is not acyclic, they show that there exists an interval routing function such that the maximum length of the route between two vertices is at most two times twice the diameter. Then van Leeuwen and Tan [19] studied the problem for undirected graphs. They showed that any graph supports an interval routing function with one interval per output port. They gave simple examples of graphs (trees, complete graphs, rings, meshes,...) that support such an interval routing function and where all the routes are shortest paths. They also studied the number of intervals per output port that needs an interval routing function to build routes that are shortest paths on a torus. In [3], Bakker, van Leeuwen and Tan introduced a particular class of interval routing functions, namely the *linear* interval routing schemes. They characterized the trees that 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 that can support or not a linear interval routing function with shortest paths. They showed that the hypercube and the d-dimensional meshes support a linear interval routing function with shortest paths. Bakker, van Leeuwen and Tan also studied the linear interval routing schemes with constraints on the communication costs between neighbors. This problem had been introduced by Frederickson and Janardan in [6].

In this paper, we deeply investigate the study of interval routing functions. In particular, we focus on the topological properties that a graph should satisfy in order to support an interval routing function with some particular given properties. Note that this work has a direct practical interest for the construction of the interconnection network of a distributed memory computer based on the C104 routing chip [14].

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

In Section 3, we study the minimum number of intervals associated to each output port that are necessary for a given network to admit a (linear) interval routing function. It was already known that one interval per output port is enough for cyclic intervals [19], but very few was known about linear intervals. We characterize the networks that admit a linear interval routing function with at most one interval per output port. We also characterize the graphs that admit a strict linear interval routing function with at most one interval per output port.

Then, in Section 4, we study the length of the routes generated by a (linear) interval routing function. In a first approach, we study the networks that admit a (linear) interval routing function that generates always shortest paths. Then we study a tradeoff between the maximum length of the paths generated by a routing function and the number of intervals that are used per output port.

Finally, in Section 5, we study the properties that satisfy graphs that are interesting either for practical design of interconnection networks, or for some algorithmic points of view: ring, meshes, hypercube, cube-connected-cycle, shuffle-exchange,...(see [5, 12]).

2 Statement of the problem

We are interested in parallel distributed memory multicomputers that are composed of processing elements (PE's), each connected to a router. As usual, the network composed by the routers 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. Therefore, we deal with undirected graphs (or symmetric digraphs). 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 that are simple (there is at most one edge between two neighbors) and loopless. An edge of extremities x and y is therefore denoted (x, y). The degree of any vertex x is denoted $\delta(x)$. The distance between any two vertices x and y is denoted $d(x, y)$.

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

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

$$R = \{R_x, x \in V, R_x : V \rightarrow out(x) \cup mem(x)\}$$

satisfying that for any couple of vertices $x, y \in V$, *there exists a sequence of vertices* $x = x_0, x_1, \ldots, x_k = y$ *such that* $\forall i, 0 \leq i < k, R_{x_i}(y) = (x_i, x_{i+1})$ *(we consider only routing functions that are* connected*), and* $R_y(y) = mem(y)$.

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 $\text{mem}(x)$. If $y \neq x$, then the message is forwarded by the communication link of $\text{out}(x)$ determined by $R_x(y)$. We will focus now on routing functions that are defined using intervals:

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

$$\begin{cases} a \leq i \leq b & \text{if } a \leq b \text{ (linear interval)} \\ a \leq i \leq n \text{ or } 1 \leq i \leq b & \text{if } a > b \text{ (cyclic interval)} \end{cases}$$

If $a = b$ we note the interval $[a]$ instead of $[a, a]$, we also note $]a, b]$ the interval $[a, b] - \{a\}$, and we note $\emptyset$ or $[]$ the empty interval. Unformally, an interval routing function on a graph of n vertices is defined as follows. First, label the vertices from 1 to n. Then, for each vertex x, associate one or many intervals to each edge $e \in \text{out}(x)$. The number of intervals associated to e on x is denoted $k(x, e)$. A message of destination y is routed through $e \in \text{out}(x)$ by 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. They are very different: the left one has all the drawbacks (non shortest paths, two intervals on the same edge for the vertex **D**, cyclic intervals on vertices

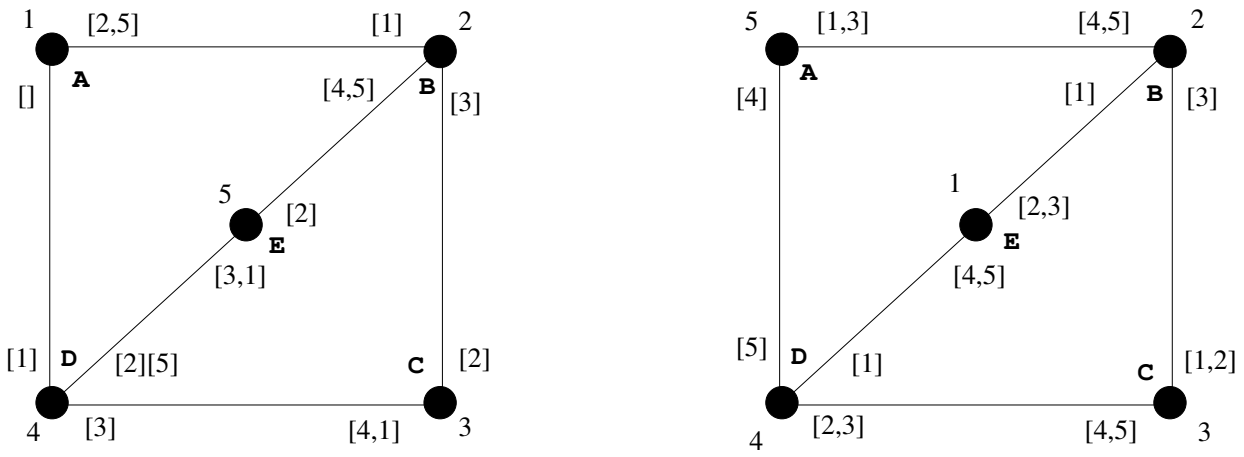


Figure 1: Two interval routing functions for the same network.

C and **E**, the edge (**A**,**D**) is never used from **A**, and **E** contains its label in the interval on (**E**,**D**)) and the right one has good properties (shortest paths, one non empty interval per edge on each vertex, and only linear intervals without label of the local vertex in its intervals).

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 *is a routing function $R = \{R_x, x \in V\}$ that is defined by*

1. *a one-to-one function $\mathcal{L} : V \rightarrow \{1, \ldots, n\}$ that labels the vertices of G*
2. *a set of intervals $\mathcal{I} = \{I^1_{x,e}, \ldots, I^{k(x,e)}_{x,e},\ x \in V, e \in out(x), k(x,e) \in \mathbb{N}^*\}$ such that the sets $I_{x,e} = \bigcup_{i=1}^{k(x,e)} I^i_{x,e},\ x \in V, e \in out(x)$ satisfy*
   - $(\bigcup_{e \in out(x)} I_{x,e}) \cup \{\mathcal{L}(x)\} = \{1, \ldots, n\}$ *(Union property)*
   - $\forall e_1, e_2 \in out(x),\ e_1 \neq e_2 \Rightarrow I_{x,e_1} \bigcap I_{x,e_2} = \emptyset$ *(Disjunction property)*

and satisfies

$$R_x(y) = e \Leftrightarrow \mathcal{L}(y) \in I_{x,e}$$

An interval routing function can be denoted by a couple $(\mathcal{L}, \mathcal{I})$ that satisfies the condition of Definition 3. It might be interesting for practical reasons to restrict the definition in allowing only the use of linear intervals (see [3]). This notion is particularly useful to derive results on networks built by cartesian products (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 only linear intervals.*

In Figure 1, the function on the right is linear and the one on the left is not (see vertices **C** and **E**).

Again, for practical reasons related to the conception of the router, we will deal with intervals that contain the local address or not. Indeed, if some intervals contain the local address then a preprocessing must be implemented to check if 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 open a new entry in the table associated to the memory link. Moreover, as we will see later, this notion is particularly interesting for the construction of interval routing functions on networks obtained by cartesian product.

Definition 5 (Strict interval routing function)
A (linear) interval routing function $R = (\mathcal{L},\mathcal{I})$ *is* strict *if* $\forall x \in V, \forall e \in out(x), \mathcal{L}(x) \notin I_{x,e}$.

In Figure 1, the function on the right is strict and the one on the left is not (see vertex **E**).

3 Compact interval routing scheme

Since the interval routing mode has been introduced to reduce the space used on the router, 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 to obtain a space of size $O(\delta(x))$ on each router x. More formally, we define the 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* $\max_{x\in V} \max_{e\in out(x)} k(x,e)$. *The* compactness *of* G *is the* minimum *over all the interval routing functions* R *on* G *of the compactness of* R.

In other words, the compactness of R is the maximum over all the vertices x of the maximum over all the edges of extremity x of the number of intervals that list destinations for which this edge 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 it and the reader will be convinced soon that the compactness as described in definition 6 allow to derive much stronger results with many practical applications.

Notation. For any positive integer k, we denote k-IRS the class of graphs of compactness $\leq k$. Similarly we denote k-LIRS the class of graphs of compactness $\leq k$ where the compactness is computed by only considering *linear* interval routing functions. Also k-(L)IRS *strict* denotes the graphs that admit a strict (linear) interval routing function of compactness at most k.

First, a very good news due to van Leeuwen and Tan [19]:

Theorem 1 *All graphs belong to* 1*-IRS strict.*

Let us give a short proof of their result.

Proof. Let r be any vertex of $V(G)$ and consider a spanning tree T 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 that belong to the subtree of T of root x.

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

Clearly, such an interval routing function is strict. □

Note that any path constructed by the routing function defined as in the proof of Theorem 1 is embedded in a tree. Hence, on one hand the length of this path is necessary less than 2 times the depth of the tree, but on the other hand, it will certainly not be a shortest path between the source and the destination. This strongly moderates the good news!

Let us now characterize the graphs that belong to 1-LIRS.

Lemma 1 $\forall k \geq 1, k\text{-}IRS \subset (k+1)\text{-}LIRS$

Proof. Any interval routing function R of compactness k on a graph G can be transformed in a linear interval routing function of compactness $k + 1$ by splitting each cyclic interval in two linear intervals (there is at most one cyclic interval per edge on any vertex). □

Remark: the paths constructed by these two routing functions (the one that insures that $G \in k$-IRS and the one constructed by splitting the cyclic intervals) are the same.

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

Now, there exists graphs that do not belong to 1-LIRS. For instance, consider the graph of Figure 2 that we call the Y-graph, and assume there exists a linear interval routing function R on the Y-graph with compactness 1. Then there exists a branch such that neither the vertex x staying in the middle of the branch or the vertex y at the extremity of the same branch is labeled 1 or 7. Let us 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 is a bridge if and only if $G' = (V, E - \{e\})$ is a disconnected graph.

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

Figure 2 presents the general form of a lithium-graph (the Y-graph is also a lithium-graph). We get easily the following lemma by the same arguments that show that the Y-graph $\notin$ 1-LIRS:

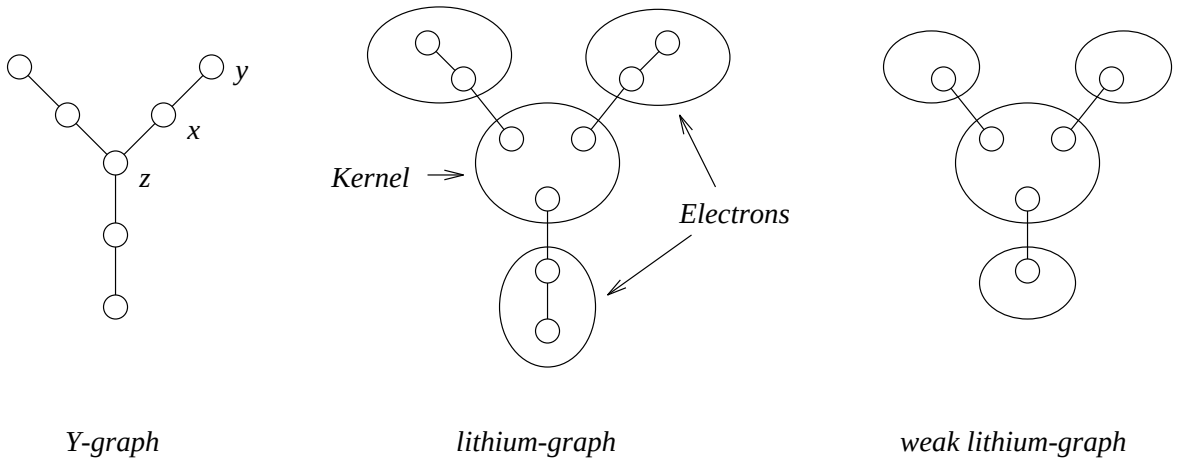


Figure 2: The Y-graph, a lithium-graph and a weak lithium-graph.

Lemma 2 *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.

Lemma 3 *Any 2-edge-connected graph G belongs to 1-LIRS strict. Moreover, for any two vertices x and y of G, there exists a linear strict interval routing function $R = (\mathcal{L}, \mathcal{I})$ satisfying:*

i. $\mathcal{L}(x) = 1$

ii. $\forall z \in V(G), \mathcal{L}(z) < \mathcal{L}(y), \forall e \in out(z)$: $\mathcal{L}(y) \in I_{z,e} \Rightarrow |V(G)| \in I_{z,e}$

iii. *let z be the vertex such that $\mathcal{L}(z) = |V(G)|$, $\exists e \in out(z), e = (z, u), \mathcal{L}(u) \leq \mathcal{L}(y)$ and $I_{z,e} = \emptyset$*

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

Initialization.

Assume G has at least three vertices.

If $x \neq y$, then from Menger's theorem, let P_1 and P_2 be two edge-disjoint paths from x to y. Moreover, it is possible to find such paths in G such that if they have a certain number of common vertices $u_1, u_2, \ldots, u_{r-1}$ distinct from x and y, then these vertices are encountered in the same order

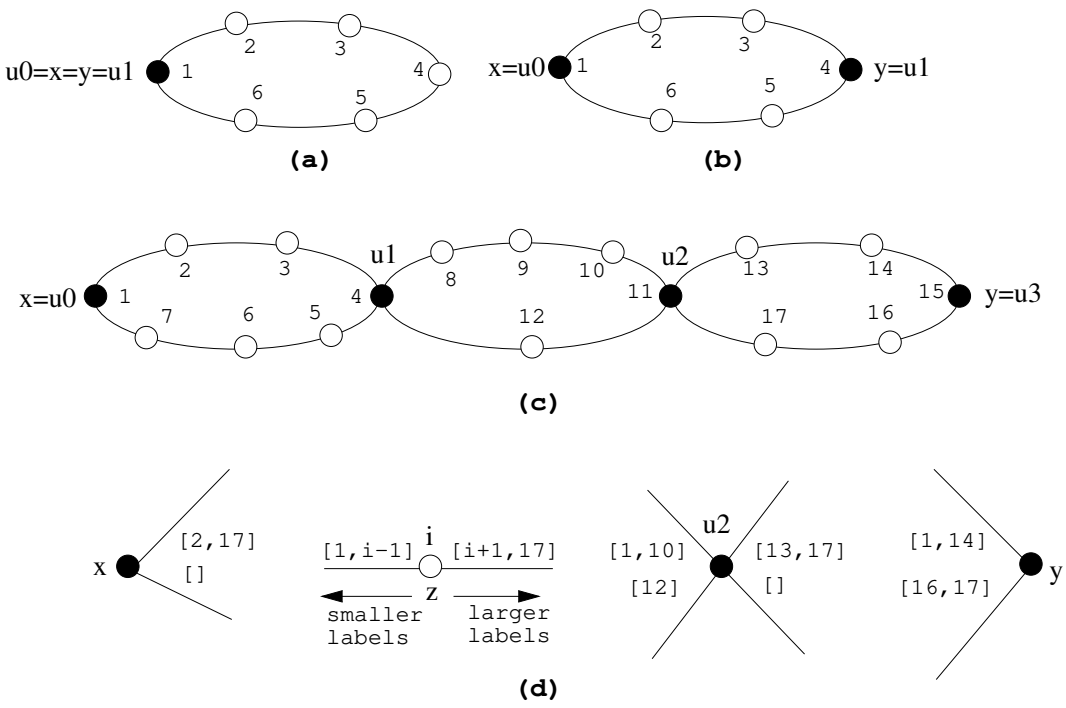


Figure 3: Labeling and intervals for the proof of Lemma 3

going from x to y on P_1 or on P_2. Thus let $u_0 = x$, $u_r = y$ and for $i \in [0, r[$, 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 3 vertices) going through x. Set $u_0 = x = y = u_1$.

Let H be the subgraph of G composed 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 an increasing order. If there is more than one cycle, then start from u_1 and label clockwise the vertices of $\mathcal{C}_1$ in an increasing order. Repeat this operation considering successively the cycles $\mathcal{C}_i, i = 2, \ldots, r-1$ until all the vertices 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 $\mathcal{C}_i, 0 \leq i < r$. Consider any cycle $\mathcal{C}_i, 0 \leq i < r$. Let z be any vertex of $\mathcal{C}_i$. Let e^+ (resp. e^-) be the clockwise (resp. counterclockwise) edge of out(z) on $\mathcal{C}_i$. We set:

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

and

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

One can easily check that this labeling and the setting of these intervals build a linear strict interval routing function on H. Concerning the three properties, (i) is satisfies ($\mathcal{L}(x) = 1$). Now, if $\mathcal{L}(z) < \mathcal{L}(y)$ then $\mathcal{L}(y)$ may belong only to I_{z,e^+}. 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, from the 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 satisfied.

Updating.

Let H be a subgraph of G of $n_H < n$ 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), (ii) and (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 $v_i \notin V(H)$, $1 \leq i \leq k$. 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 $\mathcal{L}(v) \leq L(v_0)$ then $\mathcal{L}'(v) = \mathcal{L}(v)$, otherwise $\mathcal{L}'(v) = \mathcal{L}(v) + k$. For $i = 1, \ldots, k$, $\mathcal{L}'(v_i) = \mathcal{L}(v_0) + i$.

We update the intervals as follows: let $v \in V(H)$, and $e \in E(H)$ an edge incident to v. Assume $I_{v,e} = [a, b]$, we set

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

We now have to set 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 $e = (v_i, v_{i+1})$, $1 \leq i \leq k$, we set $I'_{v_i,e} = [\mathcal{L}'(v_{i+1}), n_H + k]$. Let $e = (v_{i-1}, v_i)$, $1 \leq i \leq k$, 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 $b = n$ 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 properties (ii) also holds for the intervals on P. Thus R' satisfies property (ii) on H'.

Let $H := H'$ and repeat this process until $n_H = |V(G)|$. □

Lemma 4 *Let $G \in$ 1-LIRS strict. Let H be the graph obtained from G by adding to G a set of vertices S such that for all $x \in S$, x is connected to only one vertex of G. Then $H \in$ 1-LIRS.*

Proof. Let $R = (\mathcal{L}, \mathcal{I})$ that makes $G \in$ 1-LIRS strict. For any vertex x in G, we denote $\nu(x)$ the number of vertices of S that are connected to x in H. Then we define the labeling $\mathcal{L}'$ of the vertices of H as follows. For all $x \in V(G)$, $\mathcal{L}'(x) = \mathcal{L}(x) + \sum_{y \in V(G) | \mathcal{L}(y) < \mathcal{L}(x)} \nu(y)$. If x is connected in H to p vertices of S, namely $s_1, s_2, \ldots, s_p$, then $\mathcal{L}'(s_i) = \mathcal{L}'(x) + i$ for all $i, 1 \leq i \leq p$.

An interval $[\mathcal{L}(u), \mathcal{L}(v)] \in \mathcal{I}$ is transformed in $[\mathcal{L}'(u), \mathcal{L}'(v) + \nu(v)] \in \mathcal{I}'$ and we set new intervals in $\mathcal{I}'$: $I_{x,(x,s_i)} = [\mathcal{L}'(s_i)]$ and $I_{s_i,(s_i,x)} = [1, |V(H)|]$. □

Now, we can state our proof:

Proof of Theorem 2.

Let $G = (V, E)$ be a connected graph that is not a lithium-graph, and let $n = |V|$. We say that an edge of E is a *strong* bridge, if it is a bridge (that is a cut-edge) that splits G in two connected

components, each of at least 2 vertices. Let us decompose G, by deleting all the strong bridges from E, in the maximum number of connected components $G_0, \ldots, G_k$, 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 $G_0 - G_1 - \ldots - G_k$. 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), $i = 1, \ldots, k$. Let us also define $x_0 = y_0$ and $y_k = x_k$.

For any $i, 0 \leq i \leq k$, let G'_i be the graph obtained from G_i by removing all the vertices of degree 1 in G, and let $G' = \cup_{i=0}^{k} G'_i$. From Lemma 3, each $G'_i \in$ 1-LIRS strict and, more precisely, one can find a strict linear interval routing function $R_i = (\mathcal{L}_i, \mathcal{I}_i)$ such that

i. $\mathcal{L}_i(x_i) = 1$

ii. $\forall z \in V(G'_i), \mathcal{L}_i(z) < \mathcal{L}_i(y_i), \forall e \in \text{out}(z)$: $\mathcal{L}_i(y_i) \in I_{z,e} \in \mathcal{I}_i \Rightarrow |V(G'_i)| \in I_{z,e}$

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

From the routing functions $R_i = (\mathcal{L}_i, \mathcal{I}_i)$ of the G'_i's, $i = 1, \ldots, k$, one can define a strict interval routing function $R = (\mathcal{L}, \mathcal{I})$ of G' as follows. Let z be any vertex of G' and let i such that z is a vertex of G'_i. We set $\mathcal{L}(z)$ by simple shift of $\mathcal{L}_i(z)$: $\mathcal{L}(z) = \sum_{j=0}^{i-1} |V(G'_j)| + \mathcal{L}_i(z)$. 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 extremities is 1, this extremity is unchanged; if one of the extremities is $|V(G'_i)|$, we set this extremity to $|V(G')|$. We also replace the empty intervals $I_{z,e}$ defined by property (iii) by $I_{z,e} =]\mathcal{L}(z), |V(G')|]$. Finally, we set $I_{x_i,(x_i,y_{i-1})} = [1, \mathcal{L}(x_i)[$ and $I_{y_i,(y_i,x_{i+1})} =]\sum_{j=0}^{i} |V(G'_j)|, |V(G')|]$.

With these labeling and intervals, property (i) insures that the route from a vertex in G'_i to a vertex of $G'_j, j < i$, goes through x_i and leaves G'_i by the strong bridge (x_i, y_{i-1}), then goes to x_{i-1} and leaves G'_{i-1} by the strong bridge (x_{i-1}, y_{i-2}), etc. Property (ii) insures 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) insures 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 it 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 applying Lemma 4. □

Therefore, it is quite easy to know if a graph belongs to 1-LIRS or not. For instance:

Corollary 2 *Any interval graph [10] belongs to* 1*-LIRS.*

Proof. Let a $\tilde{Y}$-graph be a particular case of lithium-graphs: there exist three bridges that 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 that connect each electron with the kernel can be distinct of not. It is easy to check that any $\tilde{Y}$-graph is not an interval

graph (see [11, 10]). Now, any lithium-graph has a $\tilde{Y}$-graph as 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 say that any interval graph belongs to 1-LIRS (from Theorem 2). □

Note that $C_n, n > 3$ (the cycle of n vertices) is not an interval graph [11]. However $C_n \in$ 1-LIRS ($\mathcal{L}$ is a cyclic labeling and the intervals are $I_{x,(x,y)} = [1, n]$ if $\mathcal{L}(y) = 1$ or $\mathcal{L}(y) > \mathcal{L}(x)$, and $I_{x,(x,y)} = \emptyset$ otherwise). Therefore, the class 1-LIRS is not reduced to the interval graphs. In fact, this class contains most of the usual graphs considered for interconnecting PE's of a distributed memory computer. Therefore, the news of Theorem 2 is quite as good as the one of Theorem 1. Moreover, it means that the cyclic intervals facility are not necessary to build an interval routing function on usual networks. However, as the news of Theorem 1, the one of Theorem 2 must be moderated because the routes built by the linear interval routing function are not shortest paths.

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

Definition 8 *A* weak lithium-graph *is a graph with a least three bridges that connect a same connected component (the kernel) with three other distinct connected components (the electrons) – the cardinality of the electrons does not matter (see figure 2).*

Note that 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).

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

Proof.
$\Rightarrow$ Assume $G \in$ 1-LIRS strict and G is a weak lithium-graph. Consider the vertices of electrons that 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). Let us call x this vertex 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, from the hypothesis $1 < \mathcal{L}(x) < n$, and 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 the Theorem 2 to obtain a "path" $G_0 - G_1 - \ldots - G_k$, where the G_i's are connected by strong bridges. Since G is not a weak lithium-graph, we get $\delta(x) > 1$ for any vertex x of $G_i, 0 < i < k$. Now, in G_0 and G_k there is at most one vertex of degree 1 if $k > 0$, and there are at most two vertices of degree 1 if $k = 0$. That means that we can decompose G in a "path" $\{x\} - G'_0 - G_1 - \ldots - G'_k - \{y\}$ where x (resp. y) is the vertex of degree 1 of G_0 (resp. G_k) if exists, and G'_0 (resp. G'_k) is obtain from G_0 (resp. G_k) by removing x (resp. y). Then we can apply the same construction as in the proof of theorem 2 on the "path" $\{x\} - G'_0 - G_1 - \ldots - G'_k - \{y\}$. Since each component has edge-connectivity 2, the constructed routing function uses only strict intervals (from Lemma 3). □

Theorem 3 has a direct simple consequence:

Corollary 3 *Let G and H be two graphs of at least 2 vertices, then $G \times H \in$ 1-LIRS strict (where $G \times H$ is the cartesian product of G and H).*

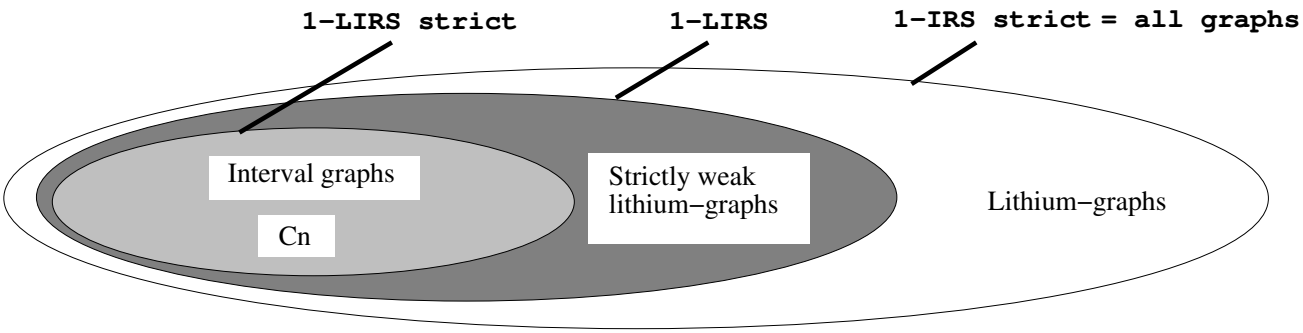


Figure 4: Classes 1-IRS strict, 1-LIRS and 1-LIRS strict.

Figure 4 summerizes the results we obtained in this section (we call a strictly weak lithium-graph a weak lithium-graph that is not a lithium-graph). In the following section, we study the length of the paths induced by an interval routing function.

4 Efficiency of an interval routing scheme

We have previously characterized graphs that admit or not a (linear) interval routing function of compactness 1. We have seen that constructions of Theorem 1 and Theorem 2 do not necessarily produce short paths between the sources and the destinations. In this section we study the relation between the compactness 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 couple of vertices x and y is of length $d(x,y)$, that is of minimum length.*

Notation. For any positive integer k, we denote k-IRS* the class of graphs that possess an optimal interval routing function of compactness $\leq k$. Similarly we denote k-LIRS* the class of graphs that possess an optimal linear interval routing function of compactness $\leq k$.

4.1.1 Families of graphs belonging to 1-(L)IRS*.

Theorem 4 *Any* unit *interval graph [11] belongs to* 1*-LIRS** *strict.*

Proof. Let G be any unit interval graph of order n: each vertex x of G can be represented by an interval J_x of $\mathbb{R}$, and all the intervals have the same length. For any vertex x let $\alpha(x)$ and $\beta(x)$ be such that $J_x = [\alpha(x), \beta(x)]$. We construct an optimal linear interval routing function $R = (\mathcal{L}, \mathcal{I})$ as follows.

We label the vertices from left to right based on $\alpha(x)$.

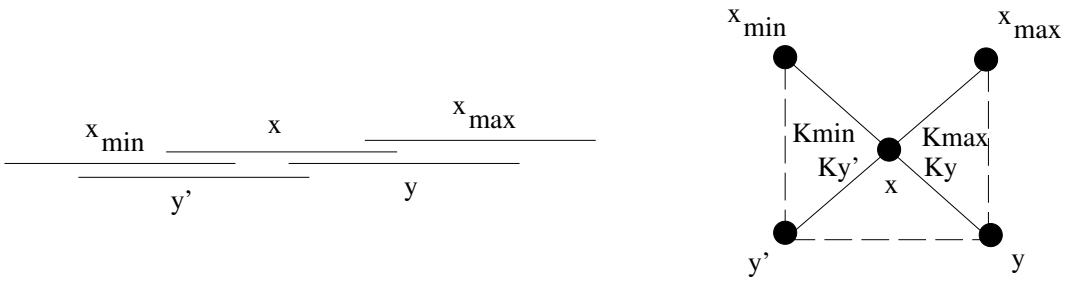


Figure 5: Setting the intervals of a routing function on an interval graph.

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 $K_{\min} = [1, \mathcal{L}(x_{\min})]$, $K_{\max} = [\mathcal{L}(x_{\max}), n]$ where n is the number of vertices of G, and for any neighbor y of x distinct from $x_{\min}$ and $x_{\max}$, let $K_y = [y]$. Then set $I_{x,(x,x_{\min})} = K_{\min}$, $I_{x,(x,x_{\max})} = K_{\max}$, $I_{x,(x,y)} = K_y$ (see Figure 5). We get for any y (if any), $K_{\min} \cap K_y = K_{\max} \cap K_y = K_{\min} \cap K_{\max} = \emptyset$, 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 does not contains $\mathcal{L}(x)$.

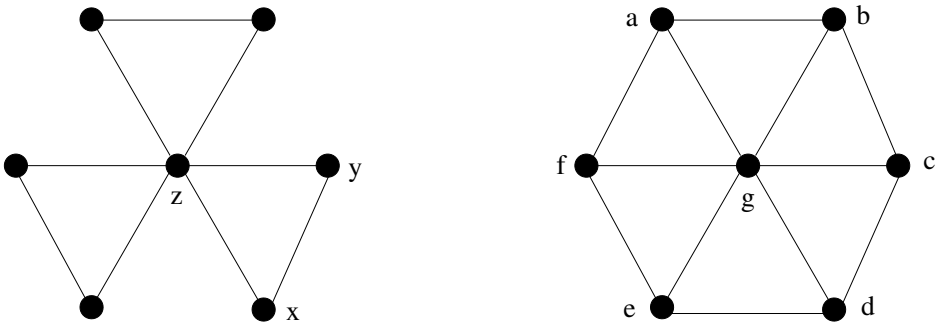
If $x_{\min} = x_{\max}$ then $\mathcal{L}(x) = 1$ or $\mathcal{L}(x) = n$. In the first case (resp. second case), the interval of the unique edge of extremity x is set as $]1, n]$ (resp. $[1, n[$).

Now we prove the property $\mathcal{P}_k$: for any k, the routing function builds a shortest path between any two vertices at distance less or equal to k. $\mathcal{P}_1$ is true. Assume $\mathcal{P}_{k'}$ is true for all $k', 1 \leq k' < k$, and let x and y be two vertices at distance k. 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 if $\mathcal{L}(x) > \mathcal{L}(y)$ then $x_{\min}$ 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. □

Note that the converse of this theorem does not hold since the C_4 graph belongs to 1-LIRS* strict and is not a unit interval graph. Also the complete bipartite graph $K_{1,3} \in$ 1-LIRS* and is not a unit interval graph. Note also that there are interval graphs that does not belong to 1-LIRS*. For instance, consider the graph in figure 6(a). Assume it belongs to 1-LIRS*. Then let x and y both different from 1 and 7, then $I_{x,(x,z)}$ must contain 1 and 7, thus $I_{x,(x,z)} = [1, 7]$ and the route from x to y is not the shortest, a contradiction.

The reader can easily verify that $C_5 \notin$ 1-LIRS* [3] (this result will be proved later for any $C_n, n \geq 5$). Now, one can prove by hand that C_n belongs to 1-IRS* for any n, but 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 is a circular-arc graph such that all arcs (representing the vertices) have the same length.

Theorem 5 *Any* unit *circular-arc graph [11] belongs to* 1-*IRS** *strict.*



(a) **(b)**

Figure 6: An interval graph that does not belong to 1-LIRS*, and a circular-arc graph that does not belong to 1-IRS*.

Proof. Let G be a unit circular-arc graph. If G is a unit interval graph then $G \in$ 1-LIRS* strict and the theorem holds. 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 and the middle of the arc representing x), and two vertices x and y are adjacent if and only if $\theta_x - \theta_y \leq \theta$ or $\theta_y - \theta_x \leq \theta$.

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

We set the intervals as follows. Let x be any vertex of G. Let z be the vertex at distance the eccentricity of x (that is the maximum distance between x and any other vertex of G) in the counterclockwise and maximizing the angle $\theta_z - \theta_x$. Then let x^- (resp. x^+) be the neighbor of x maximizing $\theta_y - \theta_x \in [0, 2\pi[$ (resp. $\theta_x - \theta_y \in [0, 2\pi[$) among the neighbors y of x satisfying $0 \leq \theta_y - \theta_x \leq \theta$ (resp. $0 \leq \theta_x - \theta_y \leq \theta$). Since G is not an interval graph, $x^- \neq x^+$. Let $K^- =]\mathcal{L}(z), \mathcal{L}(x^-)]$, $K^+ = [\mathcal{L}(x^+), \mathcal{L}(z)]$, and for any neighbor y of x distinct from x^+ and x^-, let $K_y = [y]$. Then set $I_{x,(x,x^-)} = K^-$, $I_{x,(x,x^+)} = K^+$, $I_{x,(x,y)} = K_y$. The union and the disjunction properties are clearly satisfied, and $\mathcal{L}(x)$ does not belong to $I_{x,e}$, $\forall e \in out(x)$.

As in the proof of Theorem 4, one can verify by induction on the length of the paths that the paths built by the routing function are shortest paths. □

Again the converse of this theorem is not true since $K_{1,3} \in$ 1-IRS* strict and is not an unit circular-arc graph. Note also that there are circular-arc graphs that does not belong to 1-IRS*. For instance, consider the graph in figure 6(b). Assume it belongs to 1-IRS*. Consider $I_{a,(a,g)}$, this interval must contain $\mathcal{L}(g)$ and $\mathcal{L}(d)$, and not $\mathcal{L}(b)$ or $\mathcal{L}(f)$. Similarly $I_{c,(c,g)}$ must contain $\mathcal{L}(g)$ and $\mathcal{L}(f)$, and not $\mathcal{L}(b)$ or $\mathcal{L}(d)$. Finally, $I_{e,(e,g)}$ must contain $\mathcal{L}(g)$ and $\mathcal{L}(b)$, and not $\mathcal{L}(d)$ or $\mathcal{L}(f)$, which is impossible.

4.1.2 Tools to recognize optimal routing schemes

The two following results are useful to know if a graph belongs to k-(L)IRS* for a fixed k. We recall the definition of a subgraph of shortest paths:

Definition 10 (Subgraphs of shortest paths)
A partial subgraph G' of a graph $G = (V, E)$ is a subgraph of shortest paths *of G if and only if $V(G') \subseteq V$ and $E(G')$ is composed of all the edges of all the shortest paths in G between all the couples of vertices of $V(G')$.*

The next theorem gives a necessary condition that must satisfy a graph to belong to k-(L)IRS*.

Theorem 6 *For any integer $k > 0$,*

i. *$G \in k$-LIRS$^* \Rightarrow \forall G'$ subgraph of shortest paths of G, $G' \in k$-LIRS**

ii. *$G \in k$-IRS$^* \Rightarrow \forall G'$ subgraph of shortest paths of G, $G' \in k$-IRS**

Note that since G is a subgraph of shortest paths of itself, the converse of the two properties of Theorem 6 are crudeness.

Proof. Let $G' = (V', E')$ be a subgraph of shortest paths of $G = (V, E)$, with $n = |V|$. Let $R = (\mathcal{L}, \mathcal{I})$ be an optimal interval routing function on the graph G. We define the interval routing function $R' = (\mathcal{L}', \mathcal{I}')$ of G' as follows:

$\mathcal{L}'(x)$ is the rank of $x \in V'$ where V' is sorted by increasing order according to the labeling $\mathcal{L}$.

Let f and g be the two functions from $[1, n]$ to V' such that $f(a)$ returns the vertex of V' that has the smallest label $\geq a$, and $g(b)$ returns the vertex of V' that has the largest label $\leq b$. Let $e = (x, y) \in E'$, if $I_{x,e} = [a, b]$ then we set $I'_{x,e} = [\mathcal{L}' \circ f(a), \mathcal{L}' \circ g(b)]$. Since $\mathcal{L}'^{-1}([\mathcal{L}' \circ f(a), \mathcal{L}' \circ g(b)]) = \mathcal{L}^{-1}([a, b]) \cap V'$, the disjunction and union properties are clearly satisfied.

Let x and y be two vertices of V'. We have

$$\mathcal{L}'(y) \in I'_{x,e} = [\mathcal{L}' \circ f(a), \mathcal{L}' \circ g(b)] \Leftrightarrow \mathcal{L}(y) \in I_{x,e} = [a, b].$$

Thus $R'_x(y) = e \Leftrightarrow R_x(y) = e$ and therefore the routes built by R' are shortest paths. □

In Section 5, we will see many applications of Theorem 6. This result is also used to prove the next theorem that state results concerning cartesian products. These structures are particularly interesting for constructing networks of processors (torus, hypercube,...).

Theorem 7 *For any integer $k > 0$*

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. $G \in k\text{-}LIRS^*$ *strict and* $H \in k\text{-}IRS^*$ *strict* $\Rightarrow G \times H \in k\text{-}IRS^*$ *strict;*

iv. $G \in k\text{-}LIRS^*$ *and* $H \in k\text{-}IRS^*$ *strict* $\Rightarrow G \times H \in k\text{-}IRS^*$*;*

v. $G \notin k\text{-}LIRS^* \Rightarrow G \times H \notin k\text{-}LIRS^*$ *for any graph* H*;*

vi. $G \notin k\text{-}IRS^* \Rightarrow G \times H \notin k\text{-}IRS^*$ *for any graph* H*.*

Proof. The two last results (v) and (vi) are direct consequences of Theorem 6 since the graph $G \times H$ contains G as 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 $G \in k$-LIRS* strict and $H \in k$-LIRS* strict. Let $f : [1, n_G] \times [1, n_H] \mapsto [1, n_G n_H]$ be the one-to-one function satisfying $f(a, b) = a + (b-1)n_G$. We define the labeling $\mathcal{L}$ of the vertices of $G \times H$ by $\mathcal{L}(x, y) = f(\mathcal{L}_G(x), \mathcal{L}_H(y))$. Let $z = (x, y)$ and $z' = (x', y')$ be two neighboring vertices of $G \times H$, and let $e = (z, z')$. We set

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

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 $[f(a, \mathcal{L}_H(y)), f(b, \mathcal{L}_H(y))] \cap [f(1, c), f(n_G, d)] = \emptyset$. Thus the disjunction property is satisfied. It is easy 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 as follows. The path from a vertex (x, y) to a vertex (x', y') goes first inside the x-th 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 routes 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. □

The two previous results are basic tools. A third tool will be described in Section 5. The last result of this section concerns the class of graphs that possess a strict optimal interval routing function. It shows that up to a small increase of the compactness, one can alway design a strict optimal interval routing function from any optimal interval routing function.

Theorem 8 $\forall k > 0$

i. $G \in k\text{-}LIRS^* \Rightarrow G \in (k+1)\text{-}LIRS^*$ *strict*

ii. $G \in k\text{-}IRS^* \Rightarrow G \in (k+1)\text{-}IRS^*$ *strict*

Proof. Assume a graph G possesses a non strict optimal interval routing function $R = (\mathcal{L}, \mathcal{I})$. Then for any vertex x, there exists at most an edge $e \in \text{out}(x)$ such that an interval $I_{x,e}$ contains the label $\mathcal{L}(x)$. We transform the optimal routing function $R = (\mathcal{L}, \mathcal{I})$ in a function $R' = (\mathcal{L}, \mathcal{I}')$ in splitting all these intervals $I_{x,e} = [a, b]$ in two intervals $[a, \mathcal{L}(x)[$ and $]\mathcal{L}(x), b]$. All the paths have the same length. □

4.2 Fast and compact interval routing schemes

From a practical point of view, the designer of a routing system must find a tradeoff between the hardware constraints and the efficiency of the communications. One may impose to have very fast communications and accept a lack in the hardware constraint (mainly an increasing of the surface of the routers), or impose this surface to be as small as possible even if it is up to a certain degradation of the efficiency of the communications.

4.2.1 Minimum time for a fixed compactness

Definition 11 (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 all pairs of vertices. The* k-time *of a graph G, is the minimum over all the interval routing function R on G of compactness at most k of the time of R. The* k-linear-time *of a graph G is defined similarly by considering only linear interval routing functions.*

The following result is obtained from the proof of Theorem 1. It shows that interval routing schemes are relatively efficient since, although all graphs are not optimal, it is always possible to derive routing function that build paths of small maximum length.

Corollary 4 *Let G be any graph of radius r, then the 1-time of G is $\leq 2r$.*

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

Note that one should moderate the practical use of this corollary by noting that, in such a routing scheme, many messages will go through a single vertex (the root of the tree) and therefore, many contention might happen. We do not know if this bound is tight or not (that is if there exists a graph of 1-time equal to twice its radius). In [16], the author pretends that there exists a graph whose 1-time is at least three half of its diameter. However, the proof contains so many mistakes that it lowers our confidence in this result.

Concerning linear interval routing schemes, the proof of Theorem 2 does not allow to derive a tight upper bound on the 1-linear-time of a graph that is not a lithium-graph. Furthermore, Figure 7 shows a graph on which the constructive proof of Theorem 2 gives a very inefficient routing function (1-linear-time of order the number of vertices) where it would have been possible to derive a much more efficient linear interval routing function for this graph. On the other hand we give below the best possible upper bound for the 1-linear-time of a graph which is not a lithium-graph.

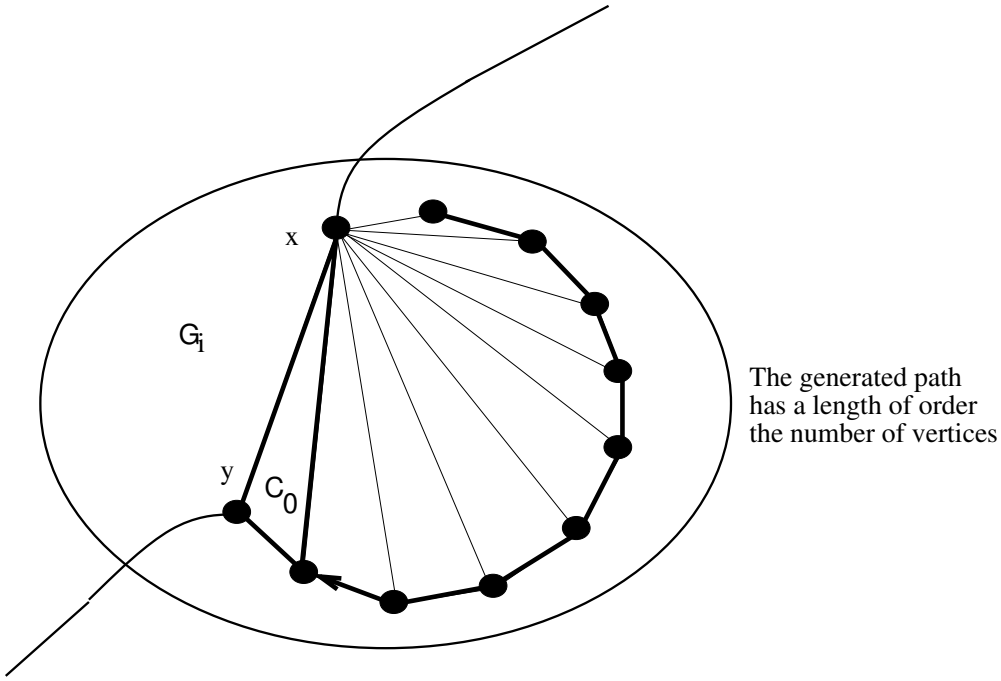


Figure 7: The constructive proof of Theorem 2 can built an inefficient routing function.

Theorem 9 *There exists a graph G of diameter D and 1-linear-time $\geq 2D-1$*

The proof of this theorem is a simple consequence of the following proposition.

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

Proof. If $n = 3$ or 4, then there exists an interval routing function that has a time of 1 or 2 respectively.

Let us show that, for $n > 4$, the 1-linear-time τ of C_n is $\geq n-2$. Assume $\tau < n-2$, and let $R = (\mathcal{L}, \mathcal{I})$ be the corresponding routing function. 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 that 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 $\mathcal{L}(y') \in [\mathcal{L}(t), \mathcal{L}(z)] \subset I_{x',(x',t)}$ yields to 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 $\mathcal{L}(x') \in [\mathcal{L}(x), \mathcal{L}(z)]$ gives rise to a contradiction. If $\mathcal{L}(x') < \mathcal{L}(x) < \mathcal{L}(y')$ then since $I_{t,(t,x')}$ must contain $\mathcal{L}(x')$ and $\mathcal{L}(y')$ but not $\mathcal{L}(x)$, one obtains another contradiction. Finally, if $\mathcal{L}(x') < \mathcal{L}(x)$ and $\mathcal{L}(y') < \mathcal{L}(x)$ then since $I_{x',(x',t)}$ must contain $\mathcal{L}(t)$ and $\mathcal{L}(x)$ but not $\mathcal{L}(y')$, one obtains the final contradiction.

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

Theorem 6 gives another proof of a result in [3]:

Corollary 5 *$\forall n \leq 4$, $C_n \in$ 1-LIRS* strict, and $\forall n \geq 5$, $C_n \notin$ 1-LIRS*.*

The following result allows, under some conditions, to derive the (linear)-time of cartesian product.

Theorem 10 *Let k be any integer, and let G be a graph of k-linear-time t_G,*

i. *if H is a graph that supports a strict interval routing function of compactness k and of k-time t_H then $G \times H$ has a k-time $\leq t_G + t_H$;*

ii. *if H is a graph that supports a strict linear interval routing function of compactness k and of k-linear-time t_H, then $G \times H$ has a k-linear-time $\leq t_G + t_H$.*

Proof. We consider the routing function constructed in the proof of Theorem 7. The two conditions for the construction are satisfied: $G \in k$-LIRS and the routing function on H is strict. Clearly the time of this routing function is $\leq t_G + t_H$. □

4.2.2 Minimum compactness for an optimal time

Definition 12 (Space)
The space *of a graph G, is the minimum k such that $G \in k$-IRS*. Similarly the* linear-space *of a graph G, is the minimum k such that $G \in k$-LIRS*.*

The following theorem improve the result in [19]:

Theorem 11 *$\forall G$ of order $n \geq 2$, the linear-space of G is $\leq \lfloor \frac{n}{2} \rfloor$ and the space of G is $\leq \lfloor \frac{n-1}{2} \rfloor$.*

Proof. It is always possible to derive a general routing table that insures that the routing function builds routes that are shortest paths. For any vertex x, there are at most $n-1$ destinations associated to each edge of out(x). These destinations can be encoded by at most $\lfloor \frac{n}{2} \rfloor$ linear intervals and at most $\lfloor \frac{n-1}{2} \rfloor$ cyclic intervals. □

As for the parameter *time*, it is possible to derive under some conditions the value of the *space* of a cartesian product:

Theorem 12 *Let G be a graph of linear-space k_G*

i. *if $H \in k_H$-LIRS* strict then $G \times H$ has a linear-space $\leq \max(k_G, k_H)$;*

ii. *if $H \in k_H$-IRS* strict then $G \times H$ has a space $\leq \max(k_G, k_H)$.*

Proof. If $k = \max(k_G, k_H)$, then G belongs to k-LIRS* and H belongs to k-(L)IRS* strict. Theorem 7 builts for $G \times H$ an interval routing function of compactness k with shortest paths. Therefore the (linear-)space of $G \times H$ is $\leq k$. □

The following results show that, unfortunately, there does not exist any bound on the maximum value of the space of a graph.

Proposition 2 *The Petersen graph does not belong to* 1-*IRS**

Proof. Assume the Petersen graph $\mathcal{P} \in$ 1-IRS* and let $R = (\mathcal{L}, \mathcal{I})$ be a corresponding interval routing function. Consider the vertex u on Figure 8, since $\mathcal{P}$ is vertex-transitive we can assume that $\mathcal{L}(u) = 1$. Consider now the vertices x, y and z drawn on Figure 8. Still by vertex-transitivity, we can assume without loss of generality that their labels are all distinct from 2 and 10. One necessarily have $\mathcal{L}(u)$, $\mathcal{L}(x)$ and $\mathcal{L}(z)$ belong to $I_{y,(y,x)}$. This implies $1 \in I_{y,(y,x)}$, $2 \notin I_{y,(y,x)}$, $10 \notin I_{y,(y,x)}$, and therefore $I_{y,(y,x)} = [1]$, a contradiction. □

Theorem 13 $\forall k > 0$, $\exists G$, *such that* $G \notin k$-*LIRS**.

Proof. For $k = 1$ we can use the result of proposition 2.

Let $k > 1$ 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 $x_1, ..., x_{2k-1}$, each x_i being connected to u. The second level is composed of $\binom{2k-1}{k-1}$ independent vertices that are labeled $(i_1, ..., i_{k-1})$ for $1 \leq i_1 < ... < i_{k-1} \leq 2k-1$. The vertex $(i_1, ..., i_{k-1})$ is connected to the $k-1$ vertices $x_{i_1}, ..., x_{i_{k-1}}$. 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.

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 his vertices is labeled 1 or n, the order of G (the graph G_1 in Figure 8). Since the x_i's play the same role, assume that $\mathcal{L}(x_1) < ... < \mathcal{L}(x_{2k-1})$. Then consider the edge e between the vertex $z = (2, 4, 6, ..., 2k-2)$ and its corresponding vertex of the third level, denoted by y. From the structure of G, we get that $1, n$ and $\mathcal{L}(x_{2i}) \in I_{y,e}$, $\forall i, 1 \leq i \leq k-i$, and $\mathcal{L}(x_{2i-1}) \notin I_{y,e}$, $\forall i, 1 \leq i \leq k$. Thus $I_{y,e}$ is composed of at least $k+1$ intervals: a contradiction. Thus $G \notin k$-LIRS*.

Note that for $k = 2$, we can reduce the graph G by grouping the first and the second level. □

Corollary 6 $\forall k > 0$, $\exists G$ *such that* $G \notin k$-*IRS**

Proof. From the remark of Lemma 1 and Theorem 13. □

Note that the precedent graph G has, as a function of k, a bounded diameter 6 but an unbounded degree $\binom{2k-1}{k-1} - 1$.

Now we are ready to study the properties that satisfy the usual graphs considered as possible interconnection networks for parallel distributed memory computers.

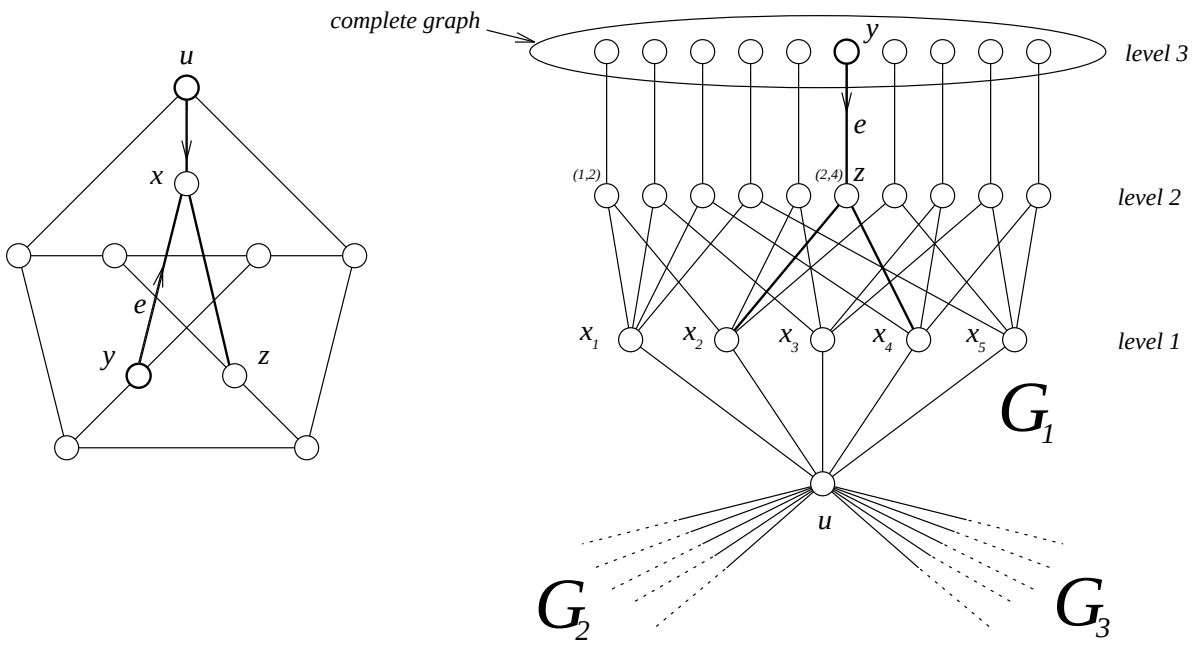


Figure 8: A Petersen graph, and a graph that does not belong to 3-LIRS*

5 Usual networks

In this section, we describe constructions of (linear) interval routing functions for many usual networks (meshes, hypercubes,...). We also indicate many particular properties that satisfy these networks.

5.1 Paths and cycles

These networks have been studied all along the paper, we refer to Proposition 1 and Corollary 5. Note also that C_n (resp. P_n, the path with n vertices) is a unit circular-arc graph (resp. unit interval graph). Thus $C_n \in$ 1-IRS* strict and $P_n \in$ 1-LIRS* strict (see theorems 5 and 4).

5.2 Meshes

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

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

Proof. P_m belongs to 1-LIRS* strict, thus we can apply $n-1$ times Theorem 7. □

5.3 Generalized hypercubes

In [3], the authors 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. $H_1^d = K_d$, and $H_n^d = H_{n-1}^d \times K_d$ that is $H_n^d = K_d \times \ldots \times K_d$, n times. Now, K_d belongs to 1-LIRS* strict, thus we can apply $n-1$ times Theorem 7. □

5.4 Torus

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

Theorem 14 *Let $T_n = C_{d_1} \times C_{d_2} \times ... \times C_{d_n}$ be an n dimensional torus, and assume $d_1 \leq d_2 \leq ... \leq d_n$. We get:*

i. *$T_n \in$ 2-LIRS* strict;*

ii. *$T_n \in$ 1-LIRS* if and only if $d_n < 5$ (moreover if $T_n \in$ 1-LIRS*, in fact $T_n \in$ 1-LIRS* strict);*

iii. *if $d_{n-1} < 5$, then $T_n \in$ 1-IRS*;*

iv. *if $d_1 > 2$, then the 1-time τ of T_n satisfies $\tau \leq \sum_{i=1}^{n-1} d_i + \lfloor \frac{d_n}{2} \rfloor - 2(n-1)$.*

Proof. For all n, $C_n \in$ 1-IRS* strict, therefore $C_n \in$ 2-LIRS* strict (by the remark of Lemma 1). From Theorem 7, we obtain (i). The result (ii) is obtained by application of Corollary 5 and Theorem 7. If $d_{n-1} < 5$ then $C_{d_1} \times C_{d_2} \times ... \times C_{d_n - 1} \in$ 1-LIRS* strict (from (ii)). Since $C_{d_n} \in$ 1-IRS* strict, we can apply Theorem 7. Note that the reciprocal of (iii) is stated in [16], but we are not confident in this result. For (iv), one have $C_{d_n} \in$ 1-IRS* strict, and one can apply Proposition 1, Theorem 10 and Theorem 7. □

5.5 A list of usual networks that do not belong to 1-LIRS*

5.5.1 Using the subgraph of shortest paths

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

- *The Shuffle-Exchange graph [12, 18], SE_n, $\forall n \geq 3$*
- *The Cube-Connected-Cycle [15] , CCC_n, $\forall n \geq 2$*

- *The Star-Graph [1], S_n, $\forall n \geq 1$*

Proof. All these graphs contain a cycle of 5 vertices at least as subgraph of shortest paths.

For any $n \geq 5$, SE_n contains C_5 as 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 subgraph of shortest paths, and $SE_3 \notin$ 1-LIRS* as we will see later in Proposition 6.

For any $n \geq 2$, CCC_n contains C_8 as subgraph of shortest paths.

For any $n \geq 1$, S_n contains C_6 as subgraph of shortest paths. □

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 $T_{x,e}$ the set of labels $\mathcal{L}(y)$ such that all the shortest paths between the vertices x and y, traverse the edge $e \in \text{out}(x)$. For any subset A of $[1,n]$, we denote $[A] = \{x \in [1,n] | \min A \leq x \leq \max A\}$, that is the smallest interval that contains all the integers of A. Of course, if $G \in$ 1-LIRS*, then $\forall x \in V(G)$, $\forall e_1 \neq e_2 \in out(x), T_{x,e_1} \cap T_{x,e_2} = \emptyset$; in fact we can get a little stronger result:

Lemma 5 *$G \in$ 1-LIRS* $\Rightarrow \forall x \in V(G), \forall e_1 \neq e_2 \in out(x), [T_{x,e_1}] \cap [T_{x,e_2}] = \emptyset$.*

Proof. Let $R = (\mathcal{L}, \mathcal{I})$ be a linear interval routing function on G such that $G \in$ 1-LIRS*. Clearly for all intervals $I_{x,e} \in \mathcal{I}$, $[T_{x,e}] \subset I_{x,e}$. □

For any two subsets A and B of $[1,n]$ 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 $A|B$ the property "A and B are separable". We get an easy method to prove that a given graph $G \notin$ 1-LIRS*. It is sufficient to find a subset V' of vertices of $V(G)$ such that the system of equations $T_{x,e_i} | T_{x,e_j}$, $\forall x \in V'$ and $\forall e_i \neq e_j \in out(x)$, leads to a contradiction. Such a system of equations is said *induced* by V'. For instance:

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

- *The graph SE_3 (see Figure 9(a))*
- *The 6-directional Mesh (see Figure 9(b))*
- *The 8-directional Mesh (see Figure 9(c))*
- *The Butterfly [12], BF_n, $\forall n \geq 2$ (see Figure 9(d))*
- *All the other graphs drawn in the Figure 9*

Proof. For the graph SE_3, we consider the system induced by the vertices in grey on Figure 9(a):

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

Assume $b < e$: $b < e \overset{(a)}{\Rightarrow} c < d \overset{(e)}{\Rightarrow} c < a \overset{(b)}{\Rightarrow} f < a \overset{(d)}{\Rightarrow} e < a \overset{(c)}{\Rightarrow} e < b$, a contradiction. Thus $SE_3 \notin$ 1-LIRS*.

The BF_n graph, with $n \geq 3$, always contains BF_2 as subgraph of shortest paths, and $BF_2 \notin$ 1-LIRS* by looking at the system induced by the 3 vertices in grey in Figure 9(d). For all the other graphs, look also at the system induced by the vertices in grey in Figure 9. □

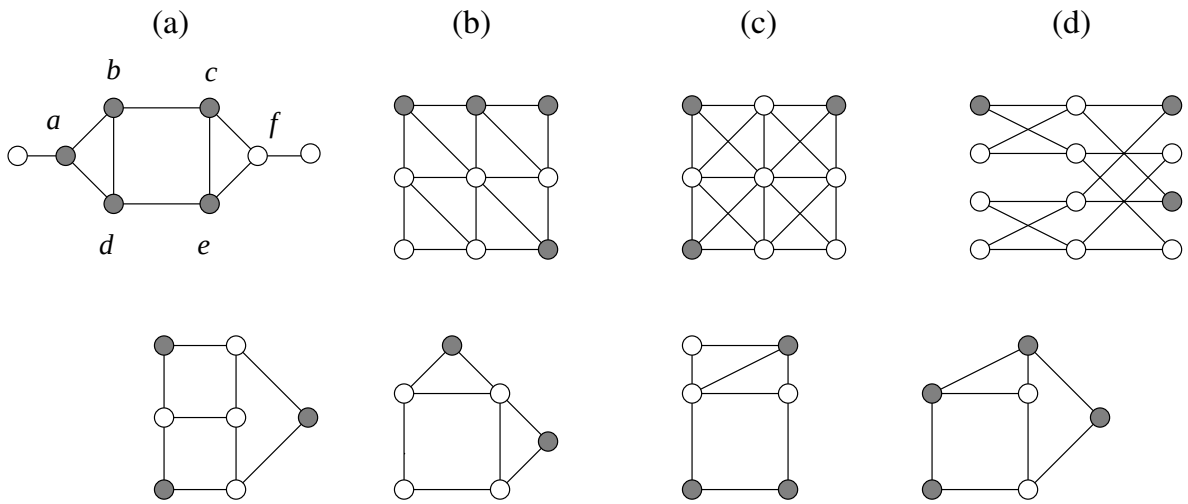


Figure 9: Some graphs that do not belong to 1-LIRS*

Remark. An easy way to find a solution of a system $\{A_i|B_i\}_i$ is to consider the associated directed graph R_G of G such that

$$V(R_G) = \{x \in V(G), \exists i, x \in A_i \text{ or } x \in B_i\}$$

and

$$(a,b) \in E(R_G) \Leftrightarrow (a,b) \in A_i \times B_i \text{ with } a < b$$

We consider the transitive closure $\overline{R_G}$ of R_G. If $\overline{R_G}$ possess a cycle, then $G \notin$ 1-LIRS*. If $\overline{R_G}$ is acyclic, then it does not prove that $G \in$ 1-LIRS*, but sorting the vertices by outer degree can give indications about a possible labeling. With this method we found labeling for the graphs $\in$ 1-LIRS* drawn in Figure 10. Note that it is stated in [3] that the left most graph in Figure 10 does not belong to 1-LIRS* which appears to be not true, and thus also the conclusion of the authors that all combinations of more than one square (i.e. K_4) with one triangle (i.e. K_3) sharing a common face $\notin$ 1-LIRS*.

The notion of induced equations can be generalized to the non-linear interval routing. For instance:

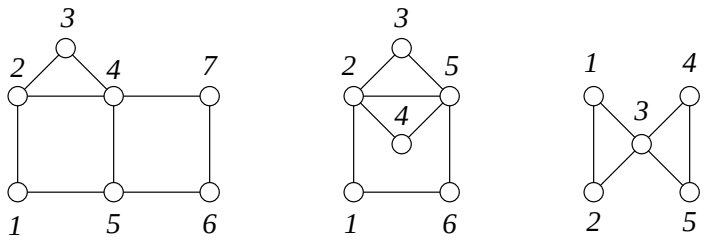


Figure 10: Some graphs that belong to 1-LIRS*

Proposition 7 $\forall n \geq 3$, $CCC_n \notin$ 1-*IRS**.

Proof. Let $R = (\mathcal{L}, \mathcal{I})$ be an interval routing function on the graph CCC_n with $n \geq 3$. Let x, y, z the labels of the vertices of a cycle such that $d(x,y) < d(x,z)+d(z,y)$ and $d(z,y) < d(z,x)+d(x,y)$. It is always possible if $n \geq 5$ or $n = 3$. We will consider later the case $n = 4$. A, B and C are respectively the set of labels of cycles A, B and C connected to x, y and z (see Figure 11). We have $A \cup \{x\} \subset T_{y,(y,x)}$, $B \cup \{y\} \subset T_{x,(x,y)}$ and $C \cup \{z\} \subset T_{y,(y,z)}$, thus $Ax|By$, $Ax|Cz$ and $By|Cz$. Now let us consider a and a' the two labels of two vertices of A (see Figure 11). Necessarily an interval $[x, y, z]$ must contains either the interval $[A]$, or $[B]$ or $[C]$ (for instance consider the interval $I_{a,(a,x)}$). Without loss of generality, assume that $[A] \subset [x, y, z]$. Clearly the set $T_{a,(a,x)}$ must contain the labels x, y and z, but not a'. Therefore, $[T_{a,(a,x)}]$ contains $[A]$ and not a', a contradiction. In the particular case $n = 4$ we have a fourth cycle D connected to a vertex of label t on the first cycle considered. Similarly we can derive an interval $[x, y, z, t]$ that must contain the set either A or B or C or D, leading to a contradiction. □

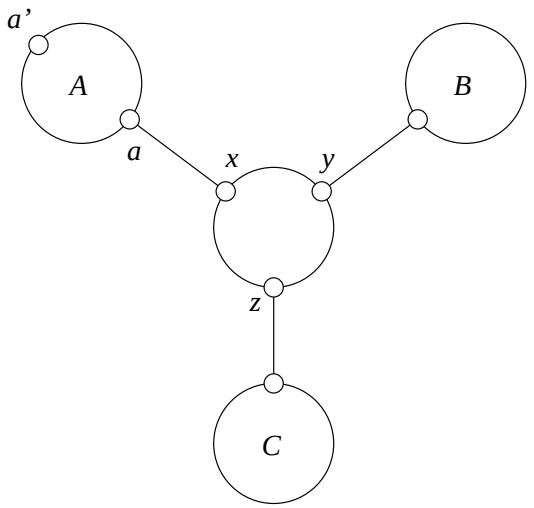


Figure 11: $CCC_n \notin$ 1-IRS*

References

[1] Sheldon B. Akers, Dov Harel, and Balakrishnan Krishnamurthy. The star graph: An attractive

alternative to the n-cube. In *International Conference on Parallel Processing (ICPP-87)*, pages 393–400, St. Charles, Illinois, August 1987.

[2] Baruch Awerbuch, Amotz Bar-Noy, Nathan Linial, and David Peleg. Improved routing strategies with succint tables. *Journal of Algorithms*, 11:307–341, February 1990.

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

[4] Frédéric Desprez, Eric Fleury, and Michel Loi. T9000 et c104 : La nouvelle génération de transputers. Technical Report 93-01, LIP-ENS Lyon, February 1993.

[5] Pierre Fraigniaud and E. Lazard. Methods and Problems of Communication in Usual Networks. *Discrete Applied Maths (special issue on broadcasting)*, (to appear).

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

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

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

[9] Cyril Gavoille. Routage par intervalles. Master's thesis, Ecole Normale Supérieure de Lyon, June 1993.

[10] Paul C. Gilmore and Alan J. Hoffman. A characterization of comparability graphs and of interval graphs. *Canad. J. Math.*, 16:539–548, 1964.

[11] Martin Charles Golumbic. *Algorithmic Graph Theory and Perfect Graphs.* A Subsidiary of Harcourt Brace Jovanovich, academic press edition, 1980.

[12] T. Leighton. *Introduction to Parallel Algorithms and architectures: Arrays - Trees - Hypercubes.* Morgan Kaufmann, 1992.

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

[14] M. Picquendar. Communication Performances of Several Interconnection Topologies for T9000-C104 Multicomputers. Technical report, Laboratoire de l'Informatique du Parallélisme, 1993.

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

[16] Peter Rušička. On efficient of interval routing algorithms. In M.P. Chytil, L. Janiga, and V. Koubek, editors, *in Mathematical Foundations of Computer Science*, volume 324 of Lectures Notes in Computer Science, pages 492–500, 1988.

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

[18] H. S. Stone. Parallel processing with the perfect shuffle. *IEEE TC*, c-20(2):153–161, 1971.

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

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