A Characterization of Networks Supporting Linear Interval Routing

Pierre Fraigniaud* and Cyril Gavoille+

LIP - CNRS
Ecole Normale Supérieure de Lyon
69364 Lyon Cedex 07, France
{pfraign,gavoille}@lip.ens-lyon.fr

Abstract

Compact routing tables are useful to implement routing algorithms on a distributed memory parallel computer. Interval routing is a popular way of building such compact tables. It was already known that any network can support an interval routing function with only one interval per output port as soon as one allows intervals to be "cyclic" [13]. However, it might be interesting for practical reasons to allow only the use of "linear" intervals (see [2]). This notion is particularly useful to derive results on networks built by cartesian products (as hypercubes and torus) [4]. In this paper, we characterize the networks that admit a *linear* interval routing function with at most one interval per output port. We also characterize the networks that admit a *strict* linear interval routing function with at most one interval per output port.

1 Introduction

If the structure of the interconnection network of a distributed memory parallel computer is fixed but complicated (a pancake graph, an undirected de Bruijn graph, etc) or if the interconnection network has no particular structure, it could be difficult to derive a "simple algorithmic" way to find locally the path between two nodes.

By a simple algorithm, we mean an algorithm whose both execution time and space for implementing it on the router are small. A solution to that problem 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 [1, 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 [12]. 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 [11, 3] of Inmos.

The notion of interval routing has been introduced by Santoro and Khatib in [12]. They mainly show that any directed acyclic network 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 the diameter. Then van Leeuwen and Tan [13] studied the problem for undirected networks. They showed that any network supports an interval routing func-

* *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.*

PODC 94 - 8/94 Los Angeles CA USA

tion with one interval per output port. They gave simple examples of networks (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 [2], 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 networks 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 investigate the study of linear interval routing functions. In particular, we focus on the topological properties that a network must satisfy in order to support a linear interval routing function with only one interval per output port. In Section 2, we recall what is an interval routing function. We overlight 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 characterize the networks that admit a linear interval routing function with at most one interval per output port. Indeed, it was already known that one interval per output port is enough for cyclic intervals [13], but very few was known about linear intervals. We also characterize in Section 4 the networks that admit a strict linear interval routing function with at most one interval per output port. Finally Section 5 summarizes our results.

We refer to [4] for results concerning the length of the routes built by an interval routing function and results on topologies usually chosen for interconnecting the processors of a distributed memory multicomputer (see [5]).

2 Definitions

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 $\text{out}(x) = \{(x, y) \in E, y \in V\}$ (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* $\forall i, 0 \leqslant i < k, R_{x_i}(y) = (x_i, x_{i+1})$ *(we consider only routing function is 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:*

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

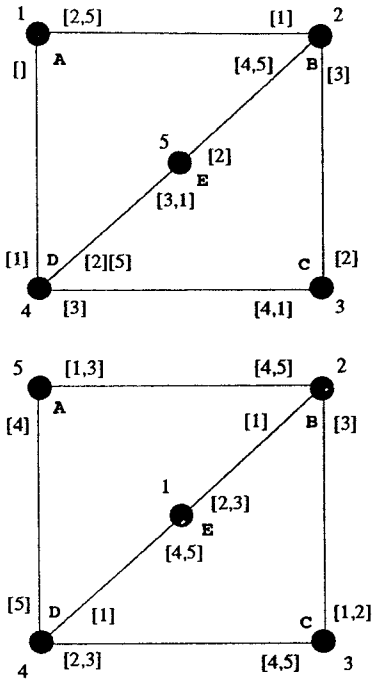

Figure 1: Two interval routing functions for the same network.

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 $e \in \text{out}(x)$. 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 top one has all the drawbacks (non shortest 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 label in the interval on (**E**,**D**)) and the bottom 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).

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. The *compactness* of a routing function 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.

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 *(of compactness 1) 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 $\mathcal{I} = \{I_{x,e}, x \in V, e \in out(x)\}$ that 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 [2]). This notion is particularly useful to derive results on networks built by cartesian products as hypercubes and torus (see [4]).

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 bottom is linear and the one on the top is not (see vertices **C** and **E**).

Notation. We denote 1-IRS the class of graphs of compactness 1. Similarly we denote 1-LIRS the class of graphs of compactness 1 where the compactness is computed by only considering *linear* interval routing functions. We refer to [4] for descriptions of the classes k-(L)IRS.

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, this notion is particularly interesting for the construction of interval routing functions on networks obtained by cartesian product (see [4]).

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 bottom is strict and the one on the top is not (see vertex **E**).

3 Characterization of classes 1-IRS and 1-LIRS

First, a very good news due to van Leeuwen and Tan [13] (see also [12]):

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.

If, for any positive integer k, we denote k-(L)IRS the class of graphs of supporting a (linear) interval routing function with less that k intervals per output port (that is of compactness $\leqslant k$), then we get:

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

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). □

Note that 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.

Let us now 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 6 *A* lithium-graph *is a connected graph with four connected component* E_1, E_2, E_3 *and* K *such that*
(i) each component $E_i, i = 1, 2, 3$ *has at least 2 vertices;*
(ii) there is no edge connecting E_i *with* E_j *for* $i, j = 1, 2, 3$ *and* $i \neq j$*;*
(iii) each component $E_i, i = 1, 2, 3$ *is connected with* K *by one and only one bridge.*

In other words, 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

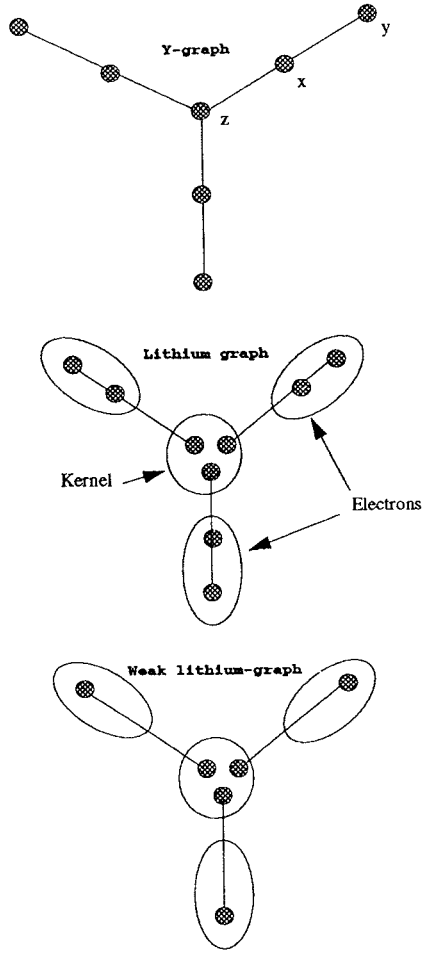

Figure 2: The Y-graph, a lithium-graph and a weak lithium-graph.

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:

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 $\mathcal{L}(z) = |V(G)|$, $\exists e \in out(z), e = (z,u), \mathcal{L}(u) < \mathcal{L}(y)$ 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 going from x to y on P_1 or on P_2. Thus let $u_0 = x$, $u_r = y$ and for $i = 0$ to $r-1$, 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 the subgraph of G that is a cycle of at least 3 vertices and 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. counter clockwise) edge of out(z) on $\mathcal{C}_i$. We set:

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

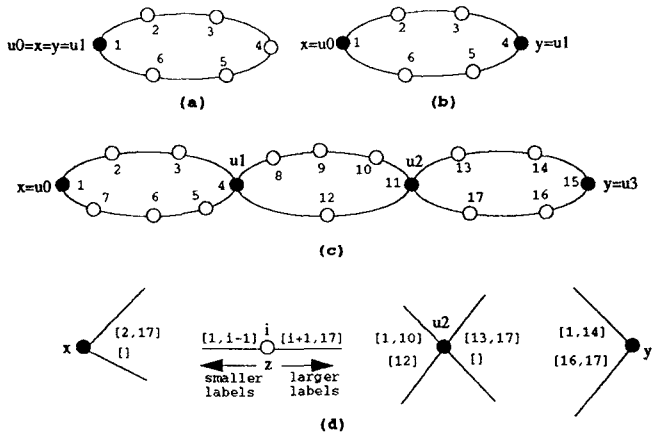

Figure 3: Labeling and intervals for the proof of Lemma 3

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 in the maximum number of connected components $G_0, \ldots, G_k$, these components being linked by all the strong bridges. In this decomposition, k is the total number of strong bridges of G and note that some component may contain only one vertex.

Since G is not a lithium-graph, each component G_i is connect 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 $\mathcal{L}_i(z) = |V(G'_i)|$, $\exists e \in \text{out}(z), e = (z, u) \in E(G'_i), \mathcal{L}_i(u) < \mathcal{L}_i(y_i)$ 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 heighest 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 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 [10, 9]). 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 $C_n, n \geqslant 3$ (the cycle of n vertices) is not an interval graph [10]. 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.

4 Characterization of the graphs that belong to 1-LIRS strict

Definition 7 *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 three 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 three vertices of the three 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.*

5 Conclusion

Figure 4 summerizes the results we obtained in this paper (we call a strictly weak lithium-graph a weak lithium-graph that is not a lithium-graph).

We refer to [4] for many other results concerning interval routing. In particular discussions about a trade-off between the length of the routes built by an interval routing function and the compactness. Also, the reader will find in [4] results concerning usual networks as meshes, hypercube, CCC, Butterfly,...

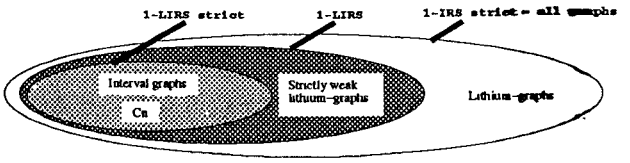

Figure 4: Classes 1-IRS strict, 1-LIRS and 1-LIRS strict.

References

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

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

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

[4] P. Fraigniaud and C. Gavoille. Interval Routing Schemes. Research Report 94-04, Laboratoire de l'Informatique du Parallélisme, ENS-Lyon, France, 1994. Submitted to the Journal of the ACM.

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

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

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

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

[13] 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.