Optimal Interval Routing

Pierre Fraigniaud[1] and Cyril Gavoille[1]

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

Abstract. Interval routing was introduced to reduce the size of the routing tables. This way of implementing routing functions is quite attractive but very few is known on 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 investigate the study of *optimal* interval routing functions, that is routing functions that construct shortest paths. In particular we derive practical tools that allow to determine if a network supports or not an optimal interval routing function. We describe large classes of networks that admit optimal interval routing functions. We also study the case of the usual networks that interconnect the processors of a distributed memory parallel computer.

1 Introduction

Compact routing has already been intensively studied (see for instance [2, 7, 8]). There exist many techniques to compress the size of routing tables. The general idea is to group 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 [14]. This kind of routing is used for instance on the C104 routing chip [10, 4] of Inmos.

The notion of interval routing has been introduced by Santoro and Khatib in [14]. 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 the diameter. Then van Leeuwen and Tan [16] 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 investigate the study of interval routing functions with shortest paths. We call them *optimal* interval routing functions. 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 [11]. In Section 2, we recall the definition of an interval routing function. We distinguish two different kinds of intervals (linear or cyclic), and the way of checking if the current address is the destination (strict interval routing function). Then, in Section 3, we describe several manners to check if a network admits or not an optimal interval routing function by considering either subgraphs of shortest paths or a system of "induced constraints". We also consider the particular case when the network is defined by a cartesian product. In section 4, we give families of networks that support an optimal interval routing function. In Section 5, we study the usual networks that connect the processors of a distributed memory parallel computer (hypercube, meshes, CCC, Shuffle-Exchange, ...). Finally, Section 6 contains some concluding remarks.

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. For any vertex $x \in V$, we denote $\mathrm{out}(x)$ the set of edges of extremity x, that is $\mathrm{out}(x) = \{(x, y) \in E, y \in V\}$. 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 $\mathrm{mem}(x)$ this link.

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

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

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 $\forall i, 0 \le i < k, R_{x_i}(y) = (x_i, x_{i+1})$ (we consider only routing function is that are *connected*), and $R_y(y) = \mathrm{mem}(y)$.

We focus on routing functions that are defined using intervals. 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:

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

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 $e \in \mathrm{out}(x)$. A message of destination y is routed through $e \in \mathrm{out}(x)$ 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 **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).

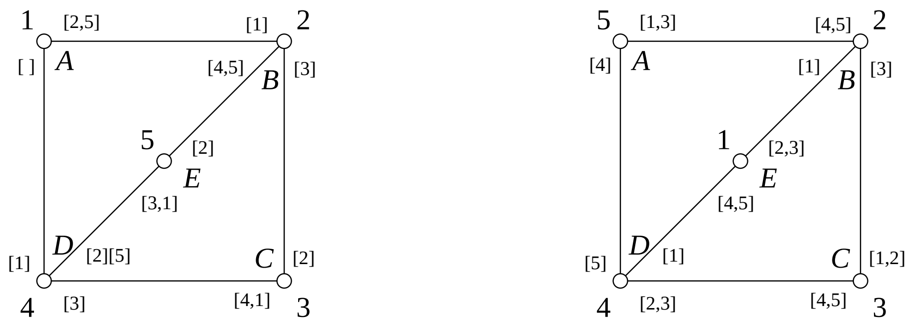

Fig. 1. Two interval routing functions for the same network.

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

Definition 1. (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 $\mathcal{L} : V \to \{1, \ldots, n\}$ that labels the vertices of G*
2. *a set of intervals $\mathcal{I} = \{\mathcal{I}_{\S,\rceil}, \S \in \mathcal{V}, \rceil \in \mathrm{out}(\S)\}$ such that the sets $I_{x,e}$ satisfy*
   – $(\bigcup_{e \in \mathrm{out}(x)} I_{x,e}) \cup \{\mathcal{L}(x)\} = \{1, \ldots, n\}$ *(Union property)*
   – $\forall e_1, e_2 \in \mathrm{out}(x),\ e_1 \neq e_2 \Rightarrow I_{x,e_1} \bigcap I_{x,e_2} = \emptyset$ *(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 1. It might be interesting for practical reasons to restrict the definition: a *linear* interval routing function is an interval routing function $R = (\mathcal{L}, \mathcal{I})$ where $\mathcal{I}$ contains only linear intervals. (see [3]). This notion is particularly useful to derive results on networks built by cartesian products (as hypercubes and torus). 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. A (linear) interval routing function $R = (\mathcal{L}, \mathcal{I})$ is *strict* if $\forall x \in V, \forall e \in \text{out}(x), \mathcal{L}(x) \notin I_{x,e}$. As we will see later, this notion is particularly interesting for the construction of interval routing functions on networks obtained by cartesian product. In Figure 1, the function on the right is strict and the one on the left is not (see vertex **E**).

Since the interval routing mode has been introduced to reduce the space used on the router, we defined the routing function using a single interval per edge on each vertex to obtain a space of size $O(\delta(x))$ on each router x. However the definition can easily be generalized to consider more than one interval per edge in each direction. The *compactness* of an interval routing function is the maximum over all the edges of the number of intervals associated to each edge in each direction. In Figure 1, the routing function on the right has a compactness 1 and the one on the left has a compactness 2 (see vertex **D**). The *compactness* of G is the *minimum* over all the interval routing functions R on G of the compactness of R.

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.

It has been shown by van Leeuwen and Tan [16] that all graphs belong to 1-IRS strict. Therefore all graphs belong to 2-LIRS strict because $\forall k \geq 1$, k-IRS $\subset$ $(k+1)$-LIRS. Now, there exists graphs that do not belong to 1-LIRS. The graphs that belong to 1-LIRS and 1-LIRS strict have been characterized in [5]. However, the main question to solve is: what about the length of the routes induced by the routing functions?

3 Optimal interval routing schemes

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 $\leq k$.

The following results are useful to know if a graph belongs to k-(L)IRS* for a fixed k.

Subgraph 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 1. *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 1 are tautologies.

Proof. Let $G' = (V', E')$ be a subgraph of shortest paths of $G = (V, E)$ of order n. 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.

For instance, it is easy to see that the cycle $C_n \notin 1-$LIRS* for $n \geq 5$. Therefore, any graph that contains a cycle of more than 5 vertices as subgraph of shortest paths cannot belong to 1-LIRS*.

Cartesian products. Cartesian products are particularly interesting for constructing networks of processors (torus, hypercube,...).

Theorem 2. *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) $G \in k$-LIRS* strict and $H \in k$-IRS* strict $\Rightarrow G \times H \in k$-IRS* strict;
(iv) $G \in k$-LIRS* and $H \in k$-IRS* strict $\Rightarrow G \times H \in k$-IRS*;
(v) $G \notin k$-LIRS* $\Rightarrow G \times H \notin k$-LIRS* for any graph H;
(vi) $G \notin k$-IRS* $\Rightarrow G \times H \notin k$-IRS* for any graph H.

Proof. We prove the two last results (v) and (vi) by Theorem 1 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 $R_G = (\mathcal{L}_G, \mathcal{I}_\mathcal{G})$ and $R_H = (\mathcal{L}_H, \mathcal{I}_\mathcal{H})$ 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

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

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.

It is easy to see that the path $P_n \in 1-$LIRS* strict. Thus, any d-dimensional mesh belongs to 1-LIRS* strict.

System of induced constraints. 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 $[A] = \{x \in [1, n] | \min A \le x \le \max A\}$, that is the smallest interval that contains all the integers of A. Of course, $\forall x \in V(G)$, $\forall e_1 \neq e_2 \in out(x), T_{x,e_1} \cap T_{x,e_2} = \emptyset$; in fact we can get a little stronger result:

Lemma 1. *$G \in 1$-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 the graph shuffle exchange SE_3 do not belong to 1-LIRS*. Indeed, consider the system induced by the vertices in grey on Figure 5(a), we get:

$$\text{in } a\text{: } bc|de \text{ ; in } b\text{: } a|cf \text{ ; in } c\text{: } e|ab \text{ ; in } d\text{: } a|ef \text{ ; in } e\text{: } c|ad$$

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*. Similarly, the butterfly graph BF_n do not belong to 1-LIRS*. Indeed, BF_n 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 5(d).

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 2. Note that it is stated in [3] that the left most graph in Figure 2 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*.

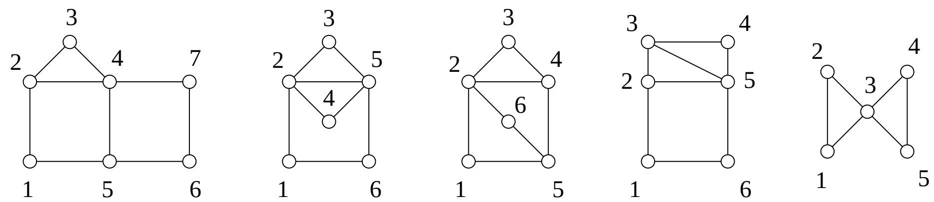

Fig. 2. Some graphs that belong to 1-LIRS*

We will see later that the notion of induced equations can be generalized to the non-linear interval routing (see the proof of Proposition 5).

4 Families of graphs that belong to 1-LIRS*

Theorem 3. *Any* unitary *interval graph belongs to* 1*-LIRS* strict.*

Proof. Let G be any unitary interval graph of order n: each vertex x of G can be represented by an interval J_x of the real axis, 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 $\alpha(x)$.

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 3). We get for any y (if any), $K_{\min} \cap K_{\max} \cap K_y = \emptyset$, 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 $\mathcal{L}(x)$.

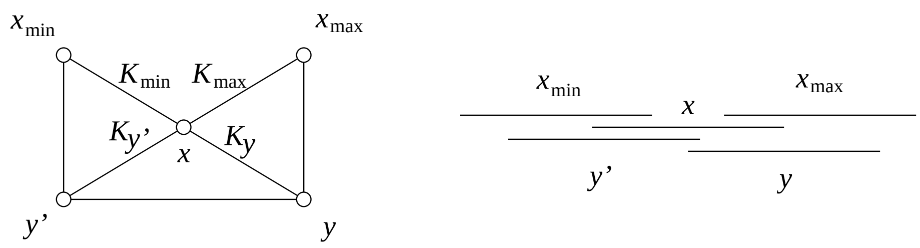

Fig. 3. Setting the intervals of a routing function on an interval graph.

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 $\mathcal{P}_{\|}$: for any k, the routing function builds a shortest path between any two vertices at distance less or equal to k. $\mathcal{P}_\infty$ is true. Assume $\mathcal{P}_{\|'}$ 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 $\mathcal{P}_{\|}$ is true since $\mathcal{P}_{\|-\infty}$ is true.

Note that the converse of this theorem does not hold since the complete bipartite graph $K_{1,3} \in$ 1-LIRS* and is not a unitary interval graph. Note also that there are interval graphs that does not belong to 1-LIRS*. For instance, consider the graph in figure 4(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.

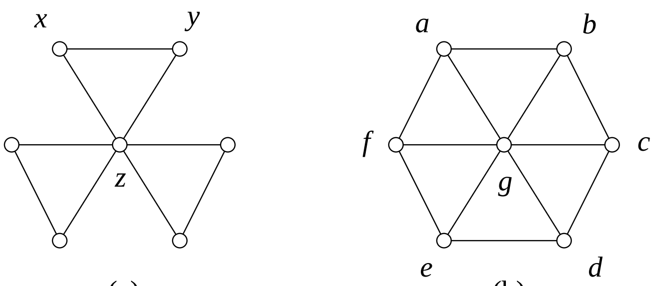

Fig. 4. An interval graph that does not belong to 1-LIRS*, and a circular graph that does not belong to 1-IRS*.

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 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 unitary circular graph is a circular graph such that all arcs (representing the vertices) have the same length.

Theorem 4. [5] *Any* unitary *circular graph belongs to* 1-*IRS** *strict.*

Again the converse of this theorem is not true since $K_{1,3} \in$ 1-IRS* and is not an unitary circular graph. Note also that there are circular graphs that does not belong to 1-IRS*. For instance, consider the graph in figure 4(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 $\mathcal{L}(d)$. Finaly, $I_{e,(e,g)}$ must contain $\mathcal{L}(g)$ and $\mathcal{L}(b)$, and not $\mathcal{L}(d)$ or $\mathcal{L}(f)$, which is impossible.

5 Usual networks

In this section, we consider many usual networks (meshes, hypercubes,...). These graphs are interesting to interconnect the processors of a distributed memory multicomputer.

Paths and cycles. C_n (resp. P_n, the path with n vertices) is a unitary circular graph (resp. unitary interval graph). Thus $C_n \in$ 1-IRS* strict and $P_n \in$ 1-LIRS* strict (see theorems 4 and 3).

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

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

Proof. P_m belongs to 1-LIRS* strict, thus we can apply $n-1$ times Theorem 2.

Generalized hypercubes. In [3], the authors proved that the n-dimensional binary hypercube belongs to 1-LIRS*. The following proposition generalizes this result:

Proposition 2. *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 $H_n^d = K_d \times \cdots \times K_d$, n times. Now, K_d belongs to 1-LIRS* strict, thus we can apply $n-1$ times Theorem 2.

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

Theorem 5. *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* strict if and only if $d_n < 5$, and (iii) if $d_{n-1} < 5$, then $T_n \in$ 1-IRS*;*

Proof. For all n, $C_n \in$ 1-IRS* strict, therefore $C_n \in$ 2-LIRS* strict. From Theorem 2, we obtain (i). The result (ii) is obtained by application of Theorem 2. If $d_{n-1} < 5$ then $C_{d_1} \times C_{d_2} \times \cdots \times C_{d_{n-1}} \in$ 1-LIRS* strict (from (ii)). Since $C_{d_n} \in$ 1-IRS*, we can apply Theorem 2. Note that the reciprocal of (iii) is stated in [13], but we are not confident in this result.

A list of usual networks that do not belong to 1-LIRS.*
– Using the subgraph of shortest paths.

Proposition 3. *The following graphs do not belong to 1-LIRS*: (i) the Shuffle-Exchange graph [9, 15], SE_n, $\forall n \geq 3$, (ii) the Cube-Connected-Cycle [12], CCC_n, $\forall n \geq 2$, (iii) 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 we have seen that $SE_3 \notin$ 1-LIRS*.

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.

– Using the system of induced constraints.

Proposition 4. *The following graphs do not belong to 1-LIRS*: (i) the graph SE_3 (see Figure 5(a)), (ii) the 6-directional Mesh (see Figure 5(b)), (iii) the 8-directional Mesh (see Figure 5(c)), (iv) the Butterfly [9], BF_n, $\forall n \geq 2$ (see Figure 5(d)), (v) all the other graphs drawn in the Figure 5*

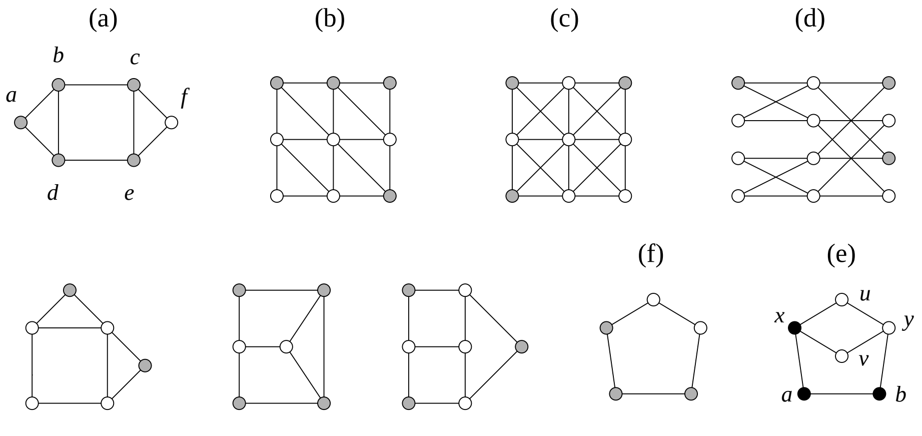

Fig. 5. Some graphs that do not belong to 1-LIRS*

Proof. Look at the system induced by the vertices in grey in Figure 5.

As said before, the notion of induced equations can be generalized to the non-linear interval routing. For instance:

Proposition 5. [5] $\forall n \geq 3$, $CCC_n \notin 1\text{-}IRS^*$.

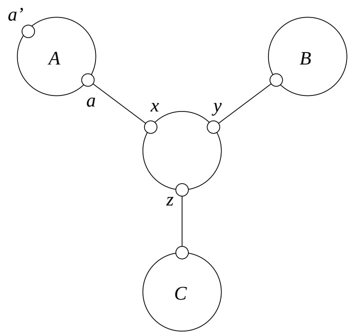

Fig. 6. $CCC_n \notin$ 1-IRS*

6 Conclusion

In this paper, we derived many tools to study the efficiency of a (linear) interval routing function. This tools are very practical. There are still many problems that can be valuable to solve, in particular is it possible to derive a practical characterization of the classes 1-IRS* and 1-LIRS*? and what is the complexity of deciding if a graph belongs to a class 1-IRS* or 1-LIRS*?

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)*, pages 393–400, St. Charles, Illinois, August 1987.
2. Baruch Awerbuch, Amotz Bar-Noy, Nathan Linial, and David Peleg. Improved routing strategies with succinct tables. *Journal of Algorithms*, 11(3):307–341, September 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. Research Report 93-01, LIP, École Normale Supérieure de Lyon, 69364 Lyon Cedex 07, France, February 1993.
5. Pierre Fraigniaud and Cyril Gavoille. Interval routing schemes. Research Report 94-04, LIP, École Normale Supérieure de Lyon, 69364 Lyon Cedex 07, France, January 1994.
6. Greg N. Frederickson and Ravi Janardan. Designing networks with compact routing tables. *Algorithmica*, 3:171–190, August 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. Frank Thomson Leighton. *Introduction to Parallel Algorithms and Architectures: Arrays - Trees - Hypercubes.* Morgan Kaufmann, 1992.
10. M.D. May, P.W. Thompson, and P.H. Welch. Networks, routers and transputers: Function, perfomance, and applications. Technical report, Inmos, SGS-THOMSON, 1993.
11. Marc Picquendar. Communication Performances of Several Interconnection Topologies for T9000-C104 Multicomputers. Research Report, Laboratoire de l'Informatique du Parallélisme, 1993.
12. F. Preparata and J. Vuillemin. The cube connected cycles: a versatile network for parallel computation. *Communications of the ACM*, 24(5):300–309, 1981.
13. Peter Ružička. On efficiency of interval routing algorithms. In M.P. Chytil, L. Janiga, and V. Koubek, editors, 13^{rd} *International Symposium on Mathematical Foundations of Computer Science (MFCS)*, volume 324 of Lectures Notes in Computer Science, pages 492–500. Springer-Verlag, 1988.
14. Nicola Santoro and Ramez Khatib. Labelling and implicit routing in networks. *The Computer Journal*, 28(1):5–8, February 1985.
15. H. S. Stone. Parallel processing with the perfect shuffle. *IEEE Transactions on Computers*, C-20(2):153–161, 1971.
16. 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.