Characterization of Networks Supporting Multi-dimensional Linear Interval Routing Schemes*

Yashar Ganjali†

yganjali@stanford.edu

MohammadTaghi Hajiaghayi‡

hajiagha@mit.edu

April 2004

Abstract

An *Interval Routing Scheme (IRS)* is a well-known, space efficient routing strategy for routing messages in a distributed network. In this scheme, each node of the network is assigned an integer label and each link at each node is labeled with an interval. The interval assigned to a link e at a node v indicates the set of destination addresses of the messages which should be forwarded through e at v. A *Multi-dimensional Interval Routing Scheme (MIRS)* is a generalization of IRS in which each node is assigned a multi-dimensional label (which is a list of $d \geq 1$ integers for the d-dimensional case). The labels assigned to the links of the network are also multi-dimensional (a list of d 1-dimensional intervals). The class of networks supporting linear IRS (in which the intervals are not cyclic) is already known for the 1-dimensional case [FG94]. In this paper, we generalize this result and completely characterize the class of networks supporting linear MIRS (or MLIRS) for a given number of dimensions d. We show that by increasing d, the class of networks supporting MLIRS is strictly expanded. We also give a characterization of the class of networks supporting strict MLIRS that is an MLIRS in which the intervals assigned to the links incident to a node v does not contain the label of v.

*This work was completed while both authors were graduate students in the Department of Computer Science, University of Waterloo. The preliminary version of this paper has appeared in the proceedings of SIROCCO'01, Barcelona, Spain, July 2001.

†Department of Electrical Engineering, Stanford University

‡Computer Science and Artificial Intelligence Laboratory, MIT

Key words: Computer networks, compact routing, interval routing schemes, multi-dimensional, graph theory, characterization.

1 Introduction

One of the most fundamental tasks in any network of computers is routing messages between pairs of nodes. The classical method used for routing messages in a network is to store a *routing table* at each node of the network. A routing table has one entry for each destination address that indicates which of the adjacent links should be used to forward the message towards that destination.

Each routing table requires $\Omega(n)$ entries in an n-node network, which is not efficient (and even feasible) for large networks of computers. The methods to reduce the amount of space needed at each node have been intensively studied and there are many techniques to compress the size of routing tables [FJ88, FJ89, ABNLP90, TvL95]. The general idea is to group the destination addresses that correspond to the same outgoing link (at a node), and to encode the group so that it is easy to verify whether a given destination address is in the group or not. A well-known solution is to use intervals as groups of destination addresses.

In an *Interval Routing Scheme (IRS)*, which was implicitly introduced by Santoro and Khatib [SK85] and made well-known by van Leeuwen and Tan [vLT87], each node of the network is assigned an integer label taken from $\{1, 2, ..., n\}$ and each link of the network at each node is assigned an interval which can be cyclic. Routing messages is completed in a distributed way. At each intermediate node v, if the label of the node equals the destination address, *dest*, the routing process ends. Otherwise, the message is forwarded through a link labeled by an interval I such that $dest \in I$. Clearly, this method requires $O(l)$ entries at each node (l is the number of links at the node), which is an efficient memory allocation.

A *Linear Interval Routing Scheme (LIRS)* is an IRS in which the intervals are not cyclic. The concept of LIRS was first introduced by Bakker et al. [BvLT91]. They mentioned practical reasons for which we allow only the use of *linear* intervals instead of *cyclic* ones. This notion is especially useful to derive results on networks built by Cartesian products (as hypercubes and torus) [FG98]. Also, a *Strict Interval Routing Scheme (SIRS)* is an IRS in which the interval assigned to a link e at a node v does not contain the label of v. A *Strict and Linear Interval Routing Scheme (SLIRS)* is an IRS which is both linear and strict. If we assign k intervals to each link of the network we will have a k-IRS (respectively, k-LIRS, k-SIRS, and k-SLIRS). Gavoille has done a survey of results concerning this method [Gav00].

It has been proved that any network supports an SIRS and therefore an IRS [SK85, vLT87]. The class of networks which support LIRS and SLIRS has also been characterized by Fraigniaud and Gavoille [FG94]. They defined a class of graphs called *lithium graphs* and showed that a network supports an LIRS if and only if its underlying graph is not a lithium graph. They also showed that a network supports an SLIRS if and only if its underlying graph is not a *weak lithium graph*.

A very interesting extension of IRS is a *Multi-dimensional Interval Routing Scheme (MIRS)* in which the labels assigned to the nodes are elements from $\mathbb{N}^d$ for some $d \geq 1$ (in the d-dimensional case) and each link is labeled with a d-tuple $([a_1, b_1], [a_2, b_2], \ldots, [a_d, b_d])$ of intervals, $a_i, b_i \in \mathbb{N}$, for $1 \leqslant i \leqslant d$ [FGNT98]. The routing process in an MIRS is quite similar to the routing process in 1-dimensional IRS. [1]

A network is said *to be in* $\langle k, d\rangle$*-MIRS* or *support* $\langle k, d\rangle$*-MIRS* if there is a d-dimensional MIRS with k intervals in each link such that for any pair of nodes s and t, the message originating from s eventually reaches t (through a path of arbitrary length and not necessarily through a shortest path). The classes $\langle k, d\rangle$-MLIRS and $\langle k, d\rangle$-MSLIRS are defined similarly. The only known classes of networks which support different variations of MIRS are specific interconnection networks such as rings, grids, tori, hypercubes and chordal rings. In this paper, we will investigate the problem of characterizing classes of networks supporting MIRS. Using a new inductive labeling scheme, we give a complete characterization of the class of networks supporting $\langle 1, d\rangle$-MLIRS and $\langle 1, d\rangle$-MSLIRS. This characterization is based on a new technique for labeling vertices and assigning intervals to the links of the network. We show that the class of networks supporting $\langle 1, d\rangle$-MLIRS ($\langle 1, d\rangle$-MSLIRS) is a proper subset of the class of networks supporting $\langle 1, d+1\rangle$-MLIRS ($\langle 1, d+1\rangle$-MSLIRS). Therefore, by increasing the number of dimensions in an MLIRS (MSLIRS) we can apply this routing scheme to a larger set of networks, thus increasing the flexibility of this scheme. This is specially important because the class of networks supporting LIRS (or SLIRS) is relatively restricted [FG94].

The rest of this paper is organized as follows: first, we will introduce some definitions and preliminaries in Section 2. In Section 3, we will characterize the class of graphs supporting $\langle 1, d\rangle$-MLIRS. Then, in Section 4, based on the arguments of the previous section, we will give a characterization for graphs supporting $\langle 1, d\rangle$-MSLIRS. Finally, in Section 5 we will conclude and give a list of open problems.

2 Preliminaries

Throughout this paper, a network is modeled by an undirected, connected, and nontrivial graph $G = (V, E)$. The set V of vertices of the graph represents the nodes in the network and the set E of edges represents the links between the nodes in the network. We assume

[1] Throughout the paper, we assume a multi-dimensional label belongs to a multi-dimensional interval if inclusion is satisfied for each dimension.

that the graph does not have any self-loop and any multi-edge. For any edge $(u, v) \in E$, each end will be assigned a label. We refer the reader to standard texts for basic graph theoretic definitions [BM76, Wes96].

A graph G is said to be *connected* if for any pair of vertices, s and t, there is a path connecting s and t. In this paper, we always assume that the network is connected. If removing an edge e disconnects a graph G, e is called a *bridge*. If a graph does not have a bridge, it is said to be *edge-biconnected*. *Edge-biconnected components of a graph* G are maximal subgraphs (i.e., further vertices and edges can not be added) of G which are edge-biconnected.

Observation 1. *If G_1 and G_2 are two distinct edge-biconnected components in a graph G, then any path P connecting G_1 and G_2 goes through a unique bridge connected to G_1.*

3 Characterization of networks supporting $\langle 1, d\rangle$-MLIRS ($d \geq 1$)

In this section we first give some examples of graphs which do not support $\langle 1, d\rangle$-MLIRS. Using the idea behind these examples, we introduce a class of graphs which do not support $\langle 1, d\rangle$-MLIRS. Finally, we show that for any graph that is not in this class, one can always construct a $\langle 1, d\rangle$-MLIRS.

Bakker et al. [BvLT91] have proved that the graph shown in Figure 1 (a) (known as the Y graph) does not have an LIRS (which is a $\langle 1, 1\rangle$-MLIRS). Here, we prove a similar result in the d-dimensional case. First, let us start by generalizing the definition of a Y graph.

Definition 1. The Y_k *graph* is a graph having $2k + 1$ vertices $u_1, u_2, ..., u_k$, $v_1, v_2, ...,$ v_k and z. There is an edge connecting u_i to v_i, for every i, $1 \leqslant i \leqslant k$, and another edge connecting each v_i to z, $1 \leqslant i \leqslant k$ (Figure 1 (b)). We call the subgraph consisting of u_i and v_i the ith *wing* of the graph.

The Y graph of Figure 1 (a) is a Y_3 graph by our definition. To prove that the Y_3 graph does not have an LIRS let us assume it has an LIRS and the vertices of the graph are assigned integer labels taken from $\{1, 2, ..., 7\}$. Since we have three wings, there is a wing, say the ith wing, which does not contain 1 nor 7 (the minimum or the maximum label). Now, the interval assigned to the edge (v_i, z) at v_i must contain both 1 and 7. Therefore, this interval contains the label of u_i which is not possible.

We can prove a similar result for d-dimensional LIRS and for the Y_{2d+1} graph. In fact, we can immediately observe that if each wing of the Y_{2d+1} graph had more than just two

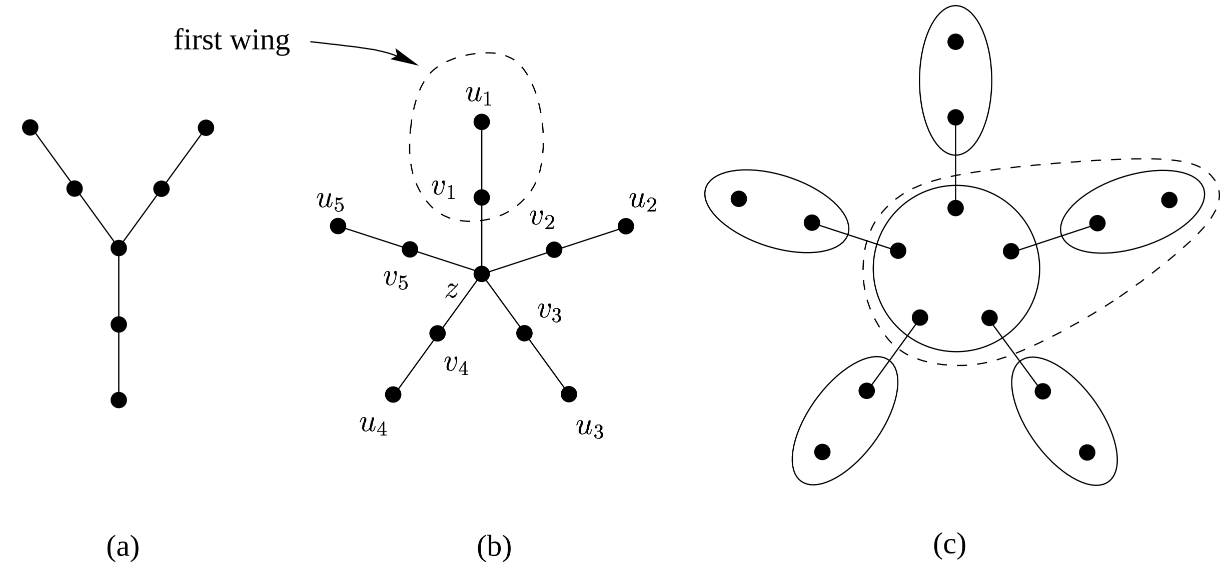

Figure 1: (a) The Y graph (b) The Y_5 graph (c) A 5-windmill graph.

vertices, as long as those extra vertices are not directly connected to the vertex z or to the vertices in other wings, the graph cannot support a d-dimensional MLIRS. In order to prove this more general statement, we define a k-windmill graph as follows.

Definition 2. A k-*windmill* graph is a connected graph with $k+1$ connected components $A_1, A_2, ..., A_k$ (*arms* of the k-windmill graph) and a connected component R (*center* of the k-windmill graph) such that:

(i) each component $A_i, 1 \leqslant i \leqslant k$, has at least two vertices;

(ii) there is no edge connecting A_i to A_j for $1 \leqslant i, j \leqslant k$ and $i \neq j$; and

(iii) each component $A_i, 1 \leqslant i \leqslant k$, is connected with R by exactly one bridge.

Figure 1 (c) illustrates a 5-windmill graph. Obviously, by this definition, a Y_k graph is also a k-windmill graph. Also, as Figure 1 (c) indicates, a k-windmill graph is an i-windmill graph for any i, $1 \leqslant i \leqslant k-1$. This can easily be shown by expanding R to include $A_{i+1}, ..., A_k$.

Lemma 1. Any $(2d+1)$-windmill graph $\notin \langle 1, d\rangle$-MLIRS.

Before proving this lemma, let us give a new definition, which will be used in the proof. We consider a set of points P in d-dimensional space. For any dimension i, $1 \leqslant i \leqslant d$, if the ith coordinate of a point b in P is less than or equal to the ith coordinate of every other point in P, b is called a *minimum point for the ith dimension.* A maximum point is defined in the opposite way. A *boundary set B of P* is a minimal set of points in P containing a minimum and a maximum point for each dimension $i, 1 \leqslant i \leqslant d$, where one point can be both the minimum and the maximum point for the same or different dimensions.

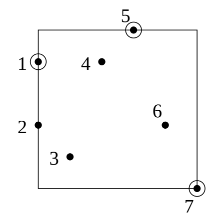

Figure 2: An example of a boundary set in 2-dimensional space.

Figure 2 illustrates an example of a boundary set in 2-dimensional space. Here, $P = \{1, \ldots, 7\}$ and $\{1, 5, 7\}$ is a boundary set of P. The set $\{2, 5, 7\}$ is also a boundary set of P. We note that point 7 is the maximum point for one dimension and the minimum point for another dimension.

For any set of points in the d-dimensional space, the number of points in any boundary set is at most $2d$. We can easily show that if a multi-dimensional interval contains the points in the boundary set B of a set of points P, it contains all points in P. Now we can easily prove Lemma 1. In this proof, we consider the d-dimensional labels of vertices as points in the d-dimensional space.

Proof. (Lemma 1) Let us assume, by way of contradiction, that there is a $\langle 1, d\rangle$-MLIRS for a given $(2d+1)$-windmill graph $(d \geqslant 1)$ and consider a boundary set B (not necessarily unique) of the vertices of the graph. We have at most $2d$ vertices in the boundary set B. Since a $(2d+1)$-windmill graph has $2d+1$ arms, there is an arm, say the jth arm, that does not contain any vertex in the boundary set B. Every d-dimensional interval containing all of the vertices in B contains all vertices of the $(2d+1)$-windmill graph as well. Thus, the interval assigned to the bridge connecting the jth arm to the center of the $(2d+1)$-windmill graph, say (u, v) (u is in the jth arm and v is a vertex in the center of the graph) contains all vertices in the $(2d+1)$-windmill graph. The jth wing has at least another vertex other than u, say u'. Hence, the interval assigned to the edge (u, v) includes u'. Obviously, there is no path going through (u, v) to reach u', which is a contradiction. ■

Lemma 1 introduces a class of graphs which do not support $\langle 1, d\rangle$-MLIRS. In other words, it states a necessary condition for a graph to support a $\langle 1, d\rangle$-MLIRS. In the rest of this section we will show that this is also a sufficient condition.

Fraigniaud and Gavoille have proved that a graph supports LIRS if and only if it is not a lithium graph [FG94] (which is exactly the 3-windmill graph). We will use this result as the basis for an inductive construction of a $\langle 1, d\rangle$-MLIRS for a given graph G. We start with some new definitions.

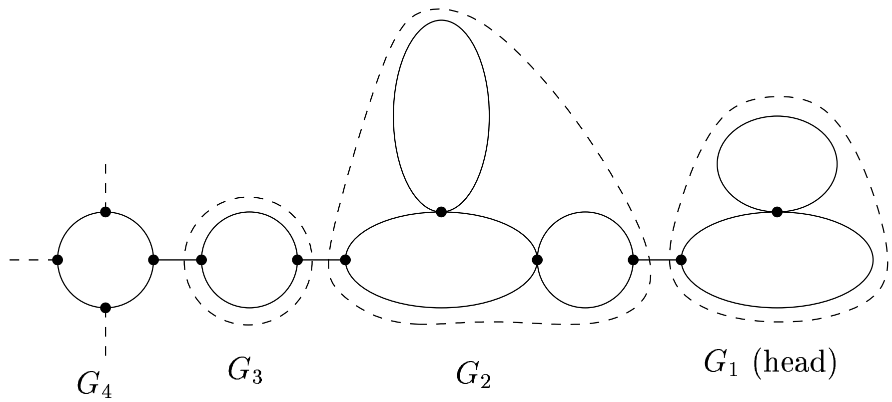

Figure 3: The dashed curves indicate edge-biconnected components in this figure. The edge-biconnected components G_1 and G_2 form a chain. The edge-biconnected components G_1, G_2 and G_3 (and not G_4) form a perfect chain.

Definition 3. In a graph G, a *chain of edge-biconnected components*[2], or a *chain* for short, is a set of edge-biconnected components of G with a special ordering of these edge-biconnected components, say $G_1, G_2, ..., G_k$ for some $k > 1$, such that:

(i) for each $i, 1 \leqslant i \leqslant k-1$, there is a bridge connecting G_i to G_{i+1};

(ii) G_1 is connected to exactly one bridge in G;

(iii) each edge-biconnected component G_i, $2 \leqslant i \leqslant k-1$, is connected to exactly two bridges in G; and

(iv) The edge-biconnected component G_k is connected to exactly one or two bridges (in which case the second bridge is toward a node of G not belonging to the chain).

We call G_1 the *head* and G_k the *tail* of the chain. Trivially if $k = 1$ then G_1 is both the head and the tail of the chain. A chain is said to be *perfect* if the tail of the chain is connected to an edge-biconnected component which is connected to more than two bridges. If the nodes in a chain include all the nodes of a graph G, then G is said to be a *chain graph.* Whenever there is no ambiguity we will use the term chain instead of chain graph.

The next observation follows directly from the definition of a chain.

[2]We note that a single vertex can be an edge-biconnected component of a graph when it is maximal, *i.e.* when it is not included in a larger edge-biconnected component of the graph.

Observation 2. *A perfect chain (if exists) in a graph G is a proper induced subgraph of G, and the tail of a perfect chain (which is an edge-biconnected component) is connected to the rest of the graph by a bridge.*

The edge-biconnected components G_1, G_2 and G_3 in the graph depicted in Figure 3 and the bridges connecting them form a chain. G_1 and G_3 are the head and the tail of this chain, respectively. This is also a perfect chain since G_3 (tail) is connected to an edge-biconnected component (G_4) which is connected to more than two bridges. As mentioned in Observation 2, G_3 (which is the tail of the perfect chain) is connected to the rest of the graph by a bridge. Since G_3 is connected to exactly two bridges, the edge-biconnected components G_1 and G_2 does not form a perfect chain.

3.1 Properties of chains and k-windmill graphs

In this section we review some of the properties of chains and k-windmill graphs.

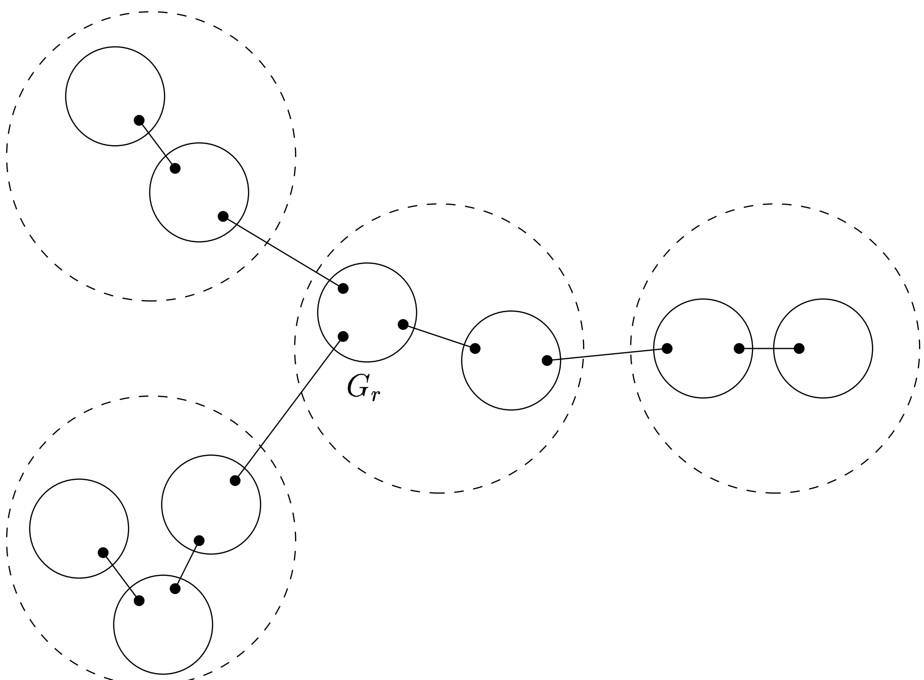

Figure 4: Edge-biconnected components in a 3-windmill graph.

Lemma 2. If a graph G is a k-windmill graph for $k \geqslant 3$ then it is not a chain.

Proof. We consider each edge-biconnected component of G as a super-node and we have an edge between two super-nodes if and only if they are connected by a bridge. Clearly, the resulting graph is a tree of super-nodes (otherwise, we have a cycle which contains some bridges, a contradiction). Since G is a k-windmill graph ($k \geqslant 3$), there is a super-node v in this tree such that the degree of v is at least 3 (the super-node G_r in Figure 4). In any chain, each edge-biconnected component is connected to at most 2 other edge-biconnected components. Therefore, G is not a chain. ■

The proof of the following lemma is simple and similar to the proof of Lemma 2 and hence omitted.

Lemma 3. Any non-trivial (having at least one vertex) graph G which is not a chain contains a perfect chain as a proper induced subgraph.

For example, we have a perfect chain in the graph depicted in Figure 3 if we consider the chain starting from the super-node G_1 and going to G_3 (which is connected to G_4 that is of degree four).

In constructing a $\langle 1, d \rangle$-MLIRS, we will use this lemma in the induction step to reduce the size of the graph. This reduction has a very nice property that is the heart of the main proof, which is stated in the following lemma.

Lemma 4. If a graph G is not a chain and is not a k-windmill graph ($k > 3$), we can remove any perfect chain from G and the resulting graph is not a $(k-1)$-windmill graph.

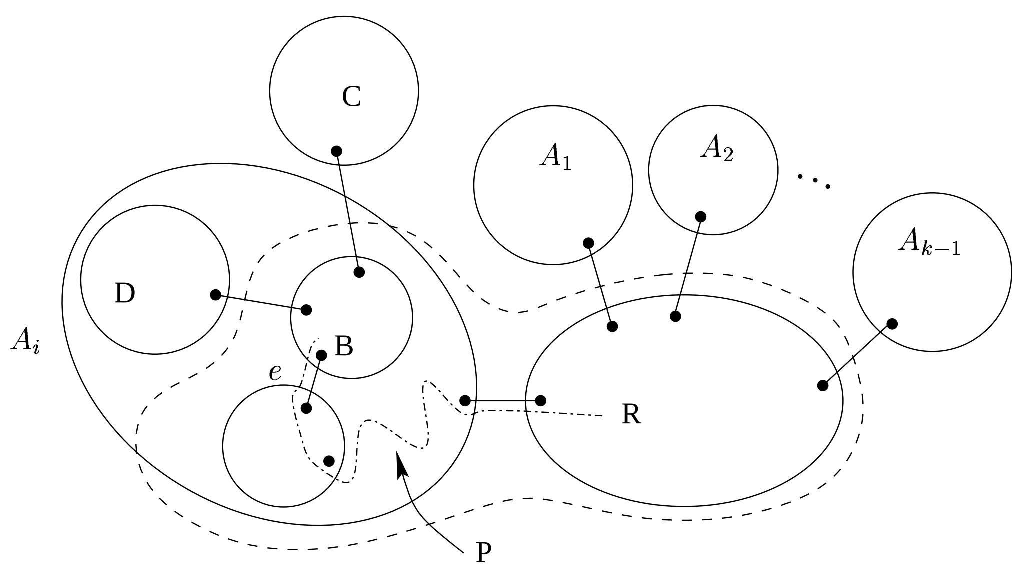

Figure 5: C and D will become arms in the k-windmill graph.

Proof. Since G is not a chain, by Lemma 3, there is a perfect chain C which is a proper induced subgraph of G. Let G' denote the graph $G - C$ and assume by contradiction that G' is a $(k-1)$-windmill graph. By the definition of $(k-1)$-windmill graph, G' has k disjoint sets of vertices $A_1, A_2, ..., A_{k-1}$ and R. Since C is a perfect chain, by Observation 2 its tail is connected to G' by a bridge. C cannot be connected to R, otherwise G must be a k-windmill graph. Let us assume that C is connected to an edge-biconnected component, B, which is in the arm A_i for some i, $1 \leqslant i \leqslant k-1$ (Figure 5).

By the definition of perfect chain, the edge-biconnected component B is connected to at least three bridges, one connecting B to C and at least two other bridges connecting B to some other edge-biconnected components in G'. By Observation 1, all the paths connecting B and R go through one of the bridges connected to B, say e. This bridge e cannot connect B and C because C cannot connect to R directly without B. Then we need two bridges of B for R and C. However, B has three bridges, and it should be connected to another edge-biconnected component D by its last bridge.

Now, we expand R to contain B and all the edge-biconnected components in the arm A_i except C and D. Since G is a $(k-1)$-windmill graph it has $k-2$ arms other than A_i. We can also consider C and D as two new arms. Hence, G has k arms and is a k-windmill graph: a contradiction. ■

3.2 Characterization

In this section we will prove the main result of this paper. First, we need to show how to convert a d-dimensional IRS into a $(d+1)$-dimensional IRS.

If a graph G supports a $\langle 1, d\rangle$-MLIRS ($\langle 1, d\rangle$-MSLIRS), we can convert the d-dimensional to a $(d+1)$-dimensional one by adding a new coordinate to the vertex labels. The label of this coordinate is set to one for all vertices. We also set the newly added coordinate of each interval to be $[1, 1]$. It is a trivial task to verify that this IRS routes the messages exactly like the d-dimensional IRS. In other words, we can expand a d-dimensional IRS to a $(d+1)$-dimensional IRS.

Lemma 5. If a graph G supports a $\langle 1, d\rangle$-MLIRS ($\langle 1, d\rangle$-MSLIRS) it also supports a $\langle 1, d+1\rangle$-MLIRS ($\langle 1, d+1\rangle$-MSLIRS).

Now, we have all the tools we need to prove the main theorem of this section.

Theorem 1. *A graph G has a $\langle 1, d\rangle$-MLIRS if and only if it is not a $(2d+1)$-windmill graph.*

Proof. First, we show that if a graph is not in the class of the $(2d+1)$-windmill graphs, then it has a $\langle 1, d\rangle$-MLIRS. We use induction on d, the number of dimensions. Fraigniaud and Gavoille [FG94] have proved that if a graph G is not a lithium graph, which is exactly a 3-windmill graph, then there is a 1-LIRS for G (a $\langle 1, 1\rangle$-MLIRS). This is the basis of the induction.

Let us suppose that for each $i \leqslant d-1$, if a graph is not a $(2i+1)$-windmill graph, it has a $\langle 1, i\rangle$-MLIRS. Now, we want to show that if a graph G is not a $(2d+1)$-windmill graph, $d > 1$, then it has a $\langle 1, d\rangle$-MLIRS. We first show how to label the vertices of G. Then, we describe how we can update intervals in each step of the induction. Finally, we prove the correctness of such vertex and link labeling.

Labeling vertices:
Although G is not a $(2d+1)$-windmill graph it can be a $(2d-1)$-windmill graph. If G is not a $(2d-1)$-windmill graph, by the induction hypothesis it has a $\langle 1, d-1\rangle$-MLIRS and by Lemma 5 we know that G also has a $\langle 1, d\rangle$-MLIRS, completing the proof. Hence, we can assume that G is a $(2d-1)$-windmill graph and by Lemma 2 we can assume that G is not a chain. Therefore, by Lemma 3 we know that G has a perfect chain, say C_1, as a proper induced subgraph. Since G is not a $(2d+1)$-windmill graph and $d > 1$, by Lemma 4 we can remove C_1 and the resulting graph will not be a $2d$-windmill graph. Since $2d > 3$, we can repeat these steps and remove another perfect chain, C_2, so that the resulting graph, G', is not a $(2d-1)$-windmill graph.

By the induction hypothesis, G' has a $\langle 1, d-1\rangle$-MLIRS. We just need to expand this labeling to a $\langle 1, d\rangle$-MLIRS for G.

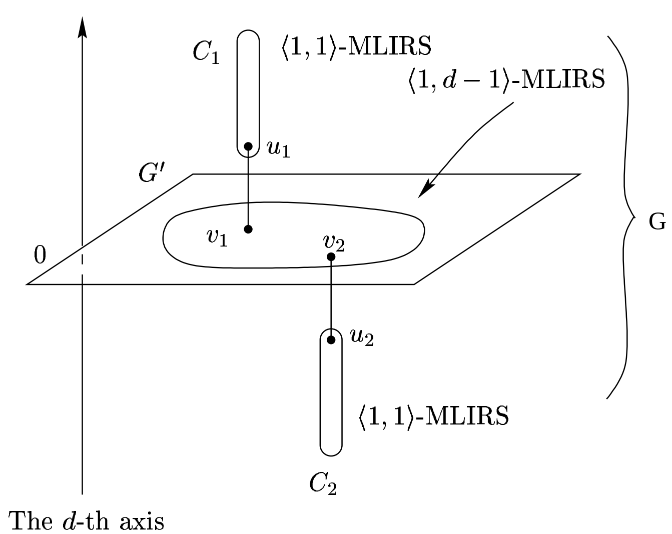

Figure 6: Expanding the labels of vertices in G' to labels for vertices in G.

C_1 and C_2 are chains and thus, by Lemma 2, they are not 3-windmill graphs. Therefore, by the induction hypothesis, there is a $\langle 1, 1\rangle$-MLIRS for each of them. In fact, in [FG94], it has been proved that if a given graph is not a 3-windmill (lithium) graph, we can specify any vertex of the graph and find a labeling for the vertices such that the label of the specified vertex is 1. We find such a $\langle 1, 1\rangle$-MLIRS for C_1 (C_2) such that the label for the vertex in C_1 (C_2) joining C_1 (C_2) to the rest of the graph G, say u_1 (u_2), is 1 (Figure 6).

To construct the new labeling for G, each vertex in G' is assigned a d-dimensional label in which the first $d-1$ coordinates are the same as the labels in the linear $\langle 1, d-1\rangle$-MIRS corresponding to G' and the dth coordinate is 0. Let us for the moment relax the constraint that all the labels assigned to the vertices must be positive integers. The value of the dth coordinate can be any integer value (including 0 and negative integers). We can shift all the labels such that the dth coordinate of all labels becomes positive later. Figure 6 illustrates an example in which $d = 3$. In this example, the third coordinates of the labels assigned to the vertices of G' are all 0's, so G' lies in the plane passing through the first and the second axes.

Let (v_1, u_1) and (v_2, u_2) denote the bridges connecting G' to C_1 and C_2, respectively, where u_1 is a vertex of C_1, u_2 is a vertex of C_2 and v_1 and v_2 are of G'. We will set the first $d-1$ coordinates of each vertex in C_1 to be equal to the first $d-1$ coordinates of v_1. The dth coordinates of vertex labels in C_1 are the labels assigned to vertices in the previously mentioned $\langle 1, 1\rangle$-MLIRS. In Figure 6 the vertices in C_1 all lie on the line passing through v_1 and parallel to the dth axis.

For the vertices in C_2, we will similarly set the first $d-1$ coordinates of each vertex equal to the first $d-1$ coordinates of v_2. If the label of a vertex v in the previously mentioned $\langle 1, 1\rangle$-MLIRS is $l(v)$, we assign $-l(v)$ as the dth coordinate of the new labeling (Figure 6). Now as mentioned before, we can shift the dth coordinate of all the labels such that the dth coordinate of the vertex with minimum value becomes 1. We let s denote the minimum amount of shifting after which all the labels in the dth coordinate become positive. We also let M denote the maximum value in the dth coordinate of all the new labels (*i.e.* after the shift).

Updating Intervals:

We update intervals as follows: the first $d-1$ coordinates of each interval assigned to a link in G' is the same as the $(d-1)$-dimensional interval associated with that edge in the $\langle 1, d-1\rangle$-MLIRS defined on G'. The dth coordinate of all intervals is set to be $[1, M]$. Any $(d-1)$-dimensional interval in G' that does not contain v_1 or v_2 will still contain the same set of vertices and any interval containing v_1 (respectively v_2) will also contain all the vertices in C_1 (C_2). For example the two-dimensional interval I shown in Figure 7 (a) contains v_1,

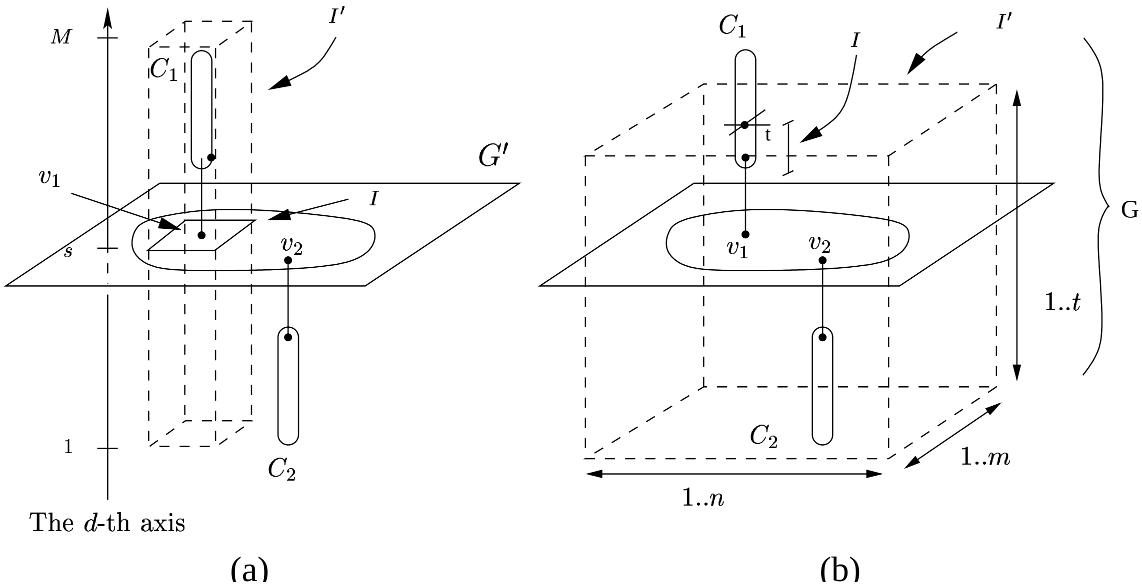

Figure 7: (a) Updating an interval in G' (b) Updating an interval, which includes u_1, in C_1 (I is the old interval, I' is the new one in both (a) and (b))

so the new three-dimensional interval I' contains all the vertices in C_1. Since I does not contain v_2, I' does not contain any of the vertices in C_2.

For the intervals associated with the links in C_1 or C_2, the first $d-1$ coordinates are set to $[1,n]$ (We note that no nodel label contains a coordinate exceeding n). To set the dth coordinate of each interval we will use the previously mentioned $\langle 1,1\rangle$-MLIRS. Let us assume that in the $\langle 1,1\rangle$-MLIRS defined on C_1 the interval assigned to a link e is $I_e=[a,b]$. If I_e does not contain u_1, the dth coordinate of the newly assigned d-dimensional interval will be $[a+s,b+s]$ (we shift the dth coordinate by s units because we have already shifted the vertices in this dimension). If I_e contains u_1, i.e. $I_e=[1,b]$ for some b, the dth coordinate of the newly assigned interval will be $I_e=[1,b+s]$. This means that any 1-dimensional interval defined in C_1 will be transformed into a d-dimensional interval containing the same set of vertices in C_1 and if it contains u_1, it will also contain all the vertices in G' and C_2. The interval I depicted in Figure 7 (b) contains u_1, so the new interval I' contains the set of vertices in C_1 that where in I and also all the vertices in C_2 and G'. We will analogously assign intervals to the links in C_2.

The only remaining labels to update are labels of the links $(v_1,u_1),(u_1,v_1),(v_2,u_2)$ and (u_2,v_2). The first $d-1$ coordinates of intervals associated with $(v_1,u_1),(u_1,v_1),(v_2,u_2)$ and (u_2,v_2) are set to $[1,n]$ and the dth coordinates will respectively be $[s+1,n],[1,s],[1,s-1]$ and $[s,n]$.

Correctness:

Now, let us consider a message originating from vertex w_s and with destination w_t. If both w_s and w_t are in C_1 (similarly C_2 or G') one can easily check that the newly defined $\langle 1, d\rangle$-MLIRS will route the messages on the same path as the $\langle 1, 1\rangle$-MLIRS defined on C_1 (C_2 or the $\langle 1, d-1\rangle$-MLIRS defined on G'). To verify this, note that if we just consider the set of vertices in C_1 (C_2 or G') each interval assigned to a link contains the same set of vertices as it contained before expanding the labels to d dimensions.

If w_s is in C_1 and w_t in C_2, the message must go through the link (u_1, v_1) since this is the only link connecting C_1 to G'. The intervals in C_1 which contain w_t are exactly the intervals containing u_1. Therefore, this message will be forwarded on the same path as the one taken by a message destined to u_1. When the message reaches u_1, the bridge (u_1, v_1) forwards the message to v_1, because the interval assigned to (u_1, v_1) contains all the vertices in G' and C_2. Similarly we can show that the message will traverse G' to reach v_2, and go through the link (v_2, u_2). The rest of the routing will be the same as the route $\langle 1, 1\rangle$-MLIRS defined on C_2. This also covers the case in which w_s is in C_1 and w_t is in G'. Proving that any message generated in a vertex in C_2 and destined to a node in C_1 (or G') is similar. Hence, any message originating at any vertex and going to an arbitrary destination will eventually reach the destination, and the $\langle 1, d\rangle$-MLIRS routes messages on G properly.

We have thus shown that a graph not belonging to the class of the $(2d+1)$-windmill graphs has a $\langle 1, d\rangle$-MLIRS. Lemma 1 shows that no graph in this class supports a $\langle 1, d\rangle$-MLIRS. Combining these two results completes the proof of the theorem. ■

Since for each $d \geq 1$ we have a $(2d+1)$-windmill graph which is not a $(2d+3)$-windmill graph (for example the Y_{2d+1} graph), we can state the following corollary:

Corollary 1. *The class of graphs supporting $\langle 1, d\rangle$-MLIRS is a proper subset of the class of graphs supporting $\langle 1, d+1\rangle$-MLIRS.*

In other words, increasing the number of dimensions increases the power of the routing scheme.

4 Characterization of networks supporting $\langle 1, d\rangle$-MSLIRS

In this section we will give a characterization of the class of graphs supporting $\langle 1, d\rangle$-MSLIRS. We will give some new definitions and show that with slight changes in some steps of the proofs we can use the same ideas used to characterize the class of graphs supporting $\langle 1, d\rangle$-MLIRS.

In proving Lemma 1, we needed to have at least two vertices in each arm of a $(2d+1)$-windmill graph. Otherwise, if the arm which did not have any vertex in the boundary set, say A_i, had just one vertex, say x, the interval assigned to the edge connecting A_i to R could contain x and this was not a contradiction. On the other hand, if the intervals assigned to the links are supposed to be strict, we could prove a similar lemma, even if we had an arm having just one vertex. This is the main difference between the proofs of this section and the previous one. More formally, let us start with a new definition. We define a *weak k-windmill* graph to be a k-windmill graph whose arms can have only one vertex (see Figure 8).

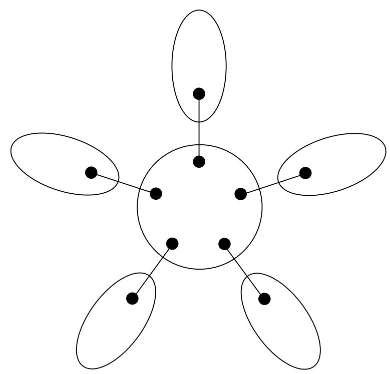

Figure 8: A weak 5-windmill graph.

As mentioned above, if the IRS is strict, then with even one vertex in each arm the proof of Lemma 1 will still be valid, because the label of a vertex which is not in the boundary set should be contained in the interval of an incident edge. Therefore, any weak $(2d+1)$-windmill graph does not have a $\langle 1, d \rangle$-MSLIRS. Using the same argument of Lemma 4, it is possible to check that removing any perfect chain from a graph G which is not a weak k-windmill graph will produce a graph which is not a weak $(k-1)$-windmill graph.

The only remaining step is to show that the induction basis and step are also valid in constructing a $\langle 1, d \rangle$-MSLIRS for any graph that is not a weak $(2d+1)$-windmill graph. We already know that any graph which is not a weak 3-windmill graph (a weak lithium graph as defined in [FG94]) has a $\langle 1, 1 \rangle$-MSLIRS, so the induction basis is true. Since we have lemmas similar to Lemmas 3 and 4 one can verify that a similar induction step still works here. This gives us the complete characterization of graph supporting $\langle 1, d \rangle$-MSLIRS as follows:

Theorem 2. *A graph G has a $\langle 1, d \rangle$-MSLIRS if and only if it is not a weak $(2d+1)$-windmill graph.*

Corollary 2. *The class of graphs supporting $\langle 1, d \rangle$-MSLIRS is a proper subset of the class of graphs supporting $\langle 1, d+1 \rangle$-MSLIRS.*

5 Conclusions and open problems

In this paper we completely characterized the classes of networks admitting $\langle 1, d\rangle$-MLIRS and $\langle 1, d\rangle$-MSLIRS. We showed that increasing the number of dimensions makes the routing scheme more powerful. One natural extension to this problem is to characterize the networks having a $\langle 1, d\rangle$-MLIRS or $\langle 1, d\rangle$-MSLIRS when the network has weighted links with dynamic costs. If messages are supposed to route shortest paths and we can relabel the edges after each change in the cost of links, there is a complete characterization for $\langle 1, d\rangle$-MSLIRS [Gan01, Gan03]. If the intervals are the same for all the costs of the links, the characterization problem is open except for the 1-dimensional case [BvLT91]. There is a partial characterization for the class of networks supporting optimum LIRS in 1-dimension [NS98]. One can also consider the problem of finding bounds on the length of routing paths for each of these classes. Finally, an interesting problem which remains open is to characterize the class of networks which support a $\langle k, d\rangle$-MLIRS ($\langle k, d\rangle$-MSLIRS) for an arbitrary $k \geqslant 1$, that is the case with k intervals assigned to each link of the network.

6 Acknowledgment

We would like to express our sincere gratitude to Professor Naomi Nishimura for her thoughtful comments, guidance and support. We would also like to thank Hamideh Emrani and the anonymous referees whose comments and suggestions improved the presentation of this paper.

References

[ABNLP90] Baruch Awerbuch, Amotz Bar-Noy, Nathan Linial, and David Peleg. Improved routing strategies with succinct tables. *J. Algorithms*, 11(3):307–341, 1990.

[BM76] John A. Bondy and U. S. R. Murty. *Graph theory with applications.* American Elsevier Publishing Co., Inc., New York, 1976.

[BvLT91] Erwin M. Bakker, Jan van Leeuwen, and Richard Tan. Linear interval routing. *ALCOM: Algorithms Review, Newsletter of the ESPRIT II Basic Research Actions Program Project no. 3075 (ALCOM)*, 2, 1991.

[FG94] Pierre Fraigniaud and Cyril Gavoille. A characterization of networks supporting linear interval routing. In 13^{th} *Annual ACM Symposium on Principles of Distributed Computing (PODC)*, pages 216–224. ACM PRESS, August 1994.

[FG98] Pierre Fraigniaud and Cyril Gavoille. Interval routing schemes. *Algorithmica*, 21(2):155–182, 1998.

[FGNT98] Michele Flammini, Giorgio Gambosi, Umberto Nanni, and Richard B. Tan. Multidimensional interval routing schemes. *Theoret. Comput. Sci.*, 205(1-2):115–133, 1998.

[FJ88] Greg N. Frederickson and Ravi Janardan. Designing networks with compact routing tables. *Algorithmica*, 3(1):171–190, 1988.

[FJ89] Greg N. Frederickson and Ravi Janardan. Efficient message routing in planar networks. *SIAM J. Comput.*, 18(4):843–857, 1989.

[Gan01] Yashar Ganjali. Multi-dimensional interval routing schemes. Master's thesis, Department of Computer Science, University of Waterloo, 2001.

[Gan03] Yashar Ganjali. Optimum multi-dimensional interval routing schemes on networks with dynamic cost links. *Computing and Informatics*, 22(1):1–18, 2003.

[Gav00] Cyril Gavoille. A survey on interval routing. *Theoret. Comput. Sci.*, 245(2):217–253, 2000. Algorithms for future technologies (Saarbrücken, 1997).

[NS98] Lata Narayanan and Sunil Shende. Partial characterizations of networks supporting shortest path interval labeling schemes. *Networks*, 32(2):103–113, 1998.

[SK85] Nicola Santoro and Ramez Khatib. Labelling and implicit routing in networks. *The Comput. J.*, 28(1):5–8, 1985.

[TvL95] Richard B. Tan and Jan van Leeuwen. Compact routing methods: A survey. In *Proceedings of Colloquium on Structural Information and Communication Complexity (SICC'94), SCS, Carleton University, Ottawa*, pages 99–109, 1995.

[vLT87] Jan van Leeuwen and Richard B. Tan. Interval routing. *The Comput. J.*, 30(4):298–307, 1987.

[Wes96] Douglas B. West. *Introduction to graph theory*. Prentice Hall Inc., Upper Saddle River, NJ, 1996.