On Compact and Efficient Routing in Certain Graph Classes

Feodor F. Dragan[1] and Irina Lomonosov[2]

[1] Department of Computer Science, Kent State University, Kent, Ohio, USA
dragan@cs.kent.edu
[2] Department of Computer Science, Hiram College, Hiram, Ohio, USA
lomonosovi@hiram.edu

Abstract. In this paper we extend the notion of tree-decomposition by introducing acyclic (R, D)-clustering, where clusters are subsets of vertices of a graph and R and D are the maximum radius and the maximum diameter of these subsets. We design a routing scheme for graphs admitting induced acyclic (R, D)-clustering where the induced radius and the induced diameter of each cluster are at most 2. We show that, by constructing a family of special spanning trees, one can achieve a routing scheme of deviation $\Delta \leq 2R$ with labels of size $O(\log^3 n / \log\log n)$ bits per vertex and $O(1)$ routing protocol for these graphs. We investigate also some special graph classes admitting induced acyclic (R, D)-clustering with induced radius and diameter less than or equal to 2, namely, chordal bipartite, homogeneously orderable, and interval graphs. We achieve the deviation $\Delta = 1$ for interval graphs and $\Delta = 2$ for chordal bipartite and homogeneously orderable graphs.

Keywords: Message routing, localized distributed algorithms, tree-decomposition, acyclic clustering, chordal bipartite graphs, homogeneously orderable graphs.

1 Introduction

Routing is one of the basic tasks that a distributed network of processors must be able to perform. A *routing scheme* is a mechanism that can deliver packets of information from any node of the network to any other node. More specifically, a routing scheme is a distributed algorithm. Each processor in the network has a routing daemon (known also as a *message passing algorithm* or a *forwarding protocol*) running on it. This daemon receives packets of information and has to decide whether these packets have already reached their destination, and if not, how to forward them towards their destination.

A network can be viewed as a graph, with the vertices representing processors and the edges representing direct connections between processors. It is naturally desirable to route messages along paths that are as short as possible. Routing scheme design is a well-studied subject. For a general overview we refer the reader to [24].

Most routing schemes are *labeling schemes* that assign two kind of labels to every vertex of a graph. The first label is the *address* of the vertex, the second is a data structure called *local routing table*. These labels are assigned in such a way that at every source vertex x its routing daemon can quickly decide, based on the two labels stored locally in x and the address of any destination node y, whether the packet has reached its destination, and if not, to which neighbor of x to forward the packet.

A straightforward approach to routing is to store a *complete routing table* at each vertex of the graph, specifying for each destination y the first edge (or identifier of that edge, indicating the output port) along some shortest path from x to y. While this approach guarantees optimal (shortest path) routing, it is too expensive for large systems since it requires total $O(n^2\ log\ \delta)$ memory bits for an n-vertex graph with maximum degree δ. Thus, for large scale communication networks, it is important to design routing schemes that produce short enough routes and have sufficiently low *memory requirements.*

Unfortunately, for every shortest path routing strategy and for all δ, there is a graph of degree bounded by δ for which $\Omega(n \log \delta)$ bit routing tables are required simultaneously on $\Theta(n)$ vertices [19]. This matches the memory requirements of complete routing tables. To obtain routing schemes for general graphs that use $o(n)$ of memory at each vertex, one has to abandon the requirement that packets are always delivered via shortest paths, and settle instead for the requirement that packets are routed on paths that are relatively close to shortest. The efficiency of a routing scheme is measured in terms of its additive stretch, called *deviation* (or multiplicative stretch, called *delay*), namely, the maximum surplus (or ratio) between the length of a route, produced by the scheme for a pair of vertices, and the shortest route. There is a tradeoff between the memory requirements of a routing scheme and the worst case stretch factor it guarantees. Any multiplicative t-stretched routing scheme must use $\Omega(\sqrt{n})$ bits for some vertices in some graphs for $t < 5$ [28], $\Omega(n)$ bits for $t < 3$ [17] (see also [13]), and $\Omega(n \log n)$ bits for $t < 1.4$ [19]. These lower bounds show that it is not possible to lower memory requirements of a routing scheme for an arbitrary network if it is desirable to route messages along paths close to optimal. Therefore it is interesting, both from a theoretical and a practical view point, to look for specific routing strategies on graph families with certain topological properties.

One way of implementing such routing schemes, called *interval routing*, has been introduced in [26] and later generalized in [21]. In this special routing method, the complete routing tables are compressed by grouping the destination addresses which correspond to the same output port. Then each group is encoded as an interval, so that it is easy to check weather a destination address belongs to the group. This approach requires $O(\delta \log n)$ bit labels and $O(\log \delta)$ forwarding protocol, where δ is the maximum degree of a vertex of the graph. A graph must satisfy some topological properties in order to support interval routing, especially if one insists on paths close to optimal. Routing schemes for many graph classes were obtained by using interval routing techniques. The classical and most recent results in this field are presented in [15, 16].

New routing schemes for interval graphs, circular-arc graphs and permutation graphs were presented in [11]. The design of these simple schemes uses properties of intersection models. Although this approach gives some improvement over existing earlier routing schemes, the local memory requirements increase with the degree of the vertex as in interval routing.

Graphs with regular topologies, as hypercubes, tori, rings, complete graphs, etc., have specific routing schemes using $O(\log n)$-bit labels [22]. It is interesting to investigate which other classes of graphs admit routing schemes with labels not depending on vertex degrees, that route messages along near-optimal path.

A shortest path routing scheme for trees of arbitrary degree and diameter is described in [14]. It assigns each vertex of an n-vertex tree a $O(\log^2 n/ \log\log n)$-bit label. Given the label of a source vertex and the label of a destination vertex it is possible to determine in constant time the neighbor of the source vertex that leads towards the destination. A similar result was independently obtained in [27]. These routing schemes for trees serve as a base for designing routing strategies for more general graphs. Indeed, if there is a family of spanning trees such that for each pair of vertices of a graph, there is a tree in the family containing a low-stretch path between them, then the tree routing scheme can be applied within that tree.

This approach was used in [10] to obtain a routing scheme of deviation 2 with labels of size $O(\log^3 n/ \log\log n)$ bits per vertex and $O(1)$ routing protocol for chordal graphs. The scheme uses the notion of tree-decomposition introduced in [25]. There, a family of spanning trees is a collection of Breadth-First-Search trees associated with each node of the tree-decomposition. It is shown that, despite the fact that the size of the family can be $O(n)$, it is enough for each vertex to keep routing labels of only $O(\log n)$ trees and, nevertheless, for each pair of vertices, a tree containing a low-stretch path between them can be determined in constant time.

In this paper we extend the notion of tree-decomposition by introducing acyclic (R, D)-clustering, where clusters are subsets of vertices of a graph and R and D are the maximum radius and the maximum diameter, respectively, of these subsets. We develop a routing scheme for graphs admitting induced acyclic (R, D)-clustering where the induced radius and the induced diameter of each cluster are at most 2. We show that, by constructing a family of special spanning trees, one can produce a routing scheme of deviation $\Delta \leq 2R$ with labels of size $O(\log^3 n/\log\log n)$ bits per vertex and $O(1)$ routing protocol for these graphs. Our routing strategy is inspired by and based on the work of Dourisboure and Gavoille [10]. Recently we learned that [9], too, generalizes the approach taken in [10] and obtains a routing scheme of deviation $\Delta \leq 2D$ with labels of size $O(D\log^3 n)$ bits per vertex and $O(log(Dlogn))$ routing protocol for the so-called tree-length D graphs [9] (which turns out to be equivalent to the class of graphs admitting acyclic (D, D)-clustering).

We investigate some special graph classes admitting induced acyclic (R, D)-clustering with induced radius and diameter less than or equal to 2, namely, chordal bipartite, homogeneously orderable, and interval graphs. We achieve the deviation $\Delta = 1$ for interval graphs and $\Delta = 2$ for chordal bipartite and homogeneously orderable graphs, while the routing schemes of [9, 10] produce $\Delta = 2$ for interval graphs and $\Delta = 4$ for chordal bipartite graphs. To the best of our knowledge this is the first routing scheme that is presented for homogeneously orderable graphs. Note that they include such well known families of graphs as distance-hereditary graphs, strongly chordal graphs, dually chordal graphs as well as homogeneous graphs (see [4]). Additionally, we achieve a constant time routing protocol and slightly lower memory requirements for chordal bipartite graphs (from [9] one could infer for chordal bipartite graphs a scheme with labels of size $O(\log^3 n)$ bits per vertex and $O(loglogn)$ routing protocol).

2 Preliminaries

All graphs occurring in this paper are connected, finite, undirected, loopless, and without multiple edges. For a subset $S \subseteq V$ of vertices of G, let $G(S)$ be a subgraph of G induced by S. By $n = |V|$ we denote the number of vertices in G.

The *distance* $dist_G(u, v)$ between vertices u and v of a graph $G = (V, E)$ is the smallest number of edges in a path connecting u and v. The distance between a vertex $u \in V$ and a set S is $dist_G(u, S) = min_{v\in S}\{dist_G(u, v)\}$. The *radius* of a set S in G is $rad_G(S) = min_{v\in S}\{max_{u\in S}\{dist_G(v, u)\}\}$ and the *diameter* is $diam_G(S) = max_{v,u\in S}\{dist_G(v, u)\}$. The *induced radius* of a set S is $rad(S) = min_{v\in S}\{max_{u\in S}\ \{dist_{G(S)}(v, u)\}\}$ and the *induced diameter* is $diam(S) = max_{v,u\in S}\{dist_{G(S)}(v, u)\}$. A vertex $v \in S$ such that $dist_{G(S)}(u, v) \leq rad(S)$ for any $u \in S$, is called a *central vertex* of S. Also, we denote by $N_G(v) = \{u \in V : uv \in E\}$ the *neighborhood* of a vertex v in G and by $N_G[v] = N_G(v) \cup \{v\}$ the *closed neighborhood* of v in G. The *kth neighborhood* $N^k(v)$ of a vertex v of G is the set of all vertices of distance k to v: $N^k_G(v) = \{u \in V : dist_G(u, v) = k\}$.

We extend the notion of tree-decomposition introduced by Robertson and Seymour [25] in the following way.

Definition 1. *A graph $G = (V, E)$ admits an acyclic (R, D)-clustering if there exists a tree T whose nodes $\mathcal{C} = \{C_1, C_2, ..., C_\kappa\}$ are subsets of V, called clusters, such that the following holds:*

1. *$\bigcup_{C\in\mathcal{C}} C = V$;*
2. *For any edge $uv \in E$, there exists $C \in \mathcal{C}$ such that $u, v \in C$;*
3. *For all $X, Y, Z \in \mathcal{C}$, if Y is on the path from X to Z in T then $X \cap Z \subseteq Y$;*

4. $max_{C\in\mathcal{C}}\{rad_G(C)\} \leq R$ *and* $max_{C\in\mathcal{C}}\{diam_G(C)\} \leq D$*, where* R *and* D *are non-negative integers.*

T is called a *tree-decomposition* of G. The value $\kappa = |\mathcal{C}|$ is called the *size of the clustering*, R and D are called the *radius of clustering* and the *diameter of clustering,* respectively. We assume that acyclic clustering is *reduced*, meaning that no cluster is contained in any other cluster (clearly any acyclic clustering can be reduced).

We say that a graph $G = (V, E)$ admits an *induced acyclic* (R, D)*-clustering* if $max_{C\in\mathcal{C}}$ $\{rad(C)\} \leq R$ and $max_{C\in\mathcal{C}}\{diam(C)\} \leq D$, where R and D are non-negative integers called the *induced radius of clustering* and the *induced diameter of clustering,* respectively. An example of a graph admitting an induced acyclic (1,2)-clustering is given in Fig. 1.

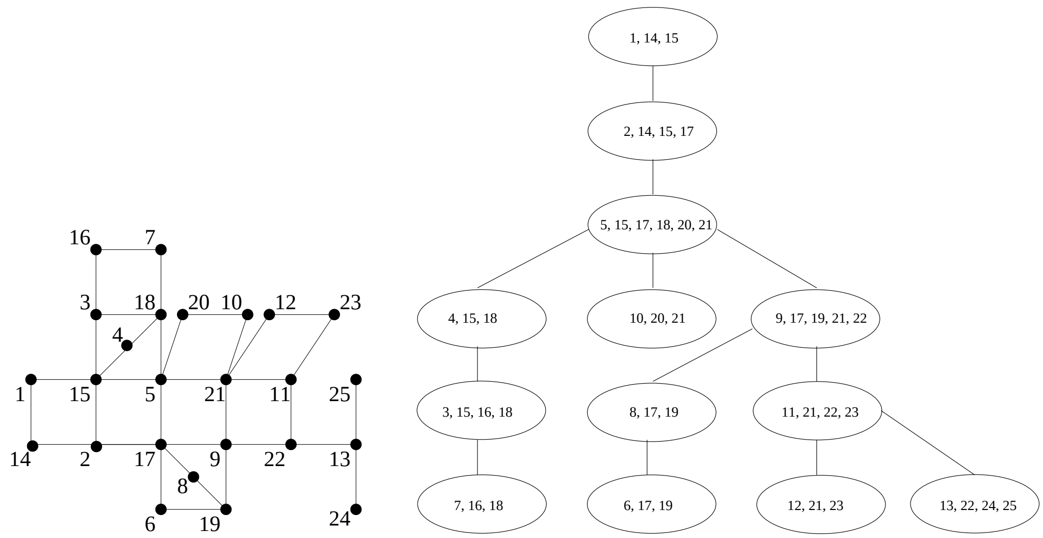

Fig. 1. A graph admitting an induced acyclic (1,2)-clustering and its tree-decomposition.

We will use the following two properties of acyclic clustering.

Lemma 1. *Let* G *be a graph admitting an acyclic* (R, D)*-clustering and* $\mathcal{C}$ *be its set of clusters. For any clusters* X*,* Y *and* Z *from* $\mathcal{C}$ *there exists a cluster* $C \in \mathcal{C}$ *such that* $(X \cap Y) \cup (X \cap Z) \cup (Z \cap Y) \subseteq C$*.*

Proof. Let Q_{XY}, Q_{XZ}, and Q_{ZY} be the paths of T between X and Y, X and Z, and Z and Y, respectively. By condition 3 of Definition 1, any node of Q_{XY} contains $X \cap Y$, any node of Q_{XZ} contains $X \cap Z$, and any node of Q_{ZY} contains $Z \cap Y$. Since T is a tree, $Q_{XY} \cap Q_{XZ} \cap Q_{YZ} \neq \emptyset$. Therefore, a cluster C such that $C \in Q_{XY} \cap Q_{XZ} \cap Q_{YZ}$ exists and $C \supseteq (X\cap Y) \cup (X \cap Z) \cup (Z \cap Y)$ holds. □

Corollary 1. *Let* G *be a graph admitting an acyclic* (R, D)*-clustering and* $\mathcal{C}$ *be its set of clusters. Let* $\{C_0, C_1, \ldots, C_p\} \subseteq \mathcal{C}$ *be a set of clusters such that* $C_0 \cap C_i \neq \emptyset$ *for* $1 \leq i \leq p$ *and there exists a vertex* x *of* G *such that* $x \in \cap\{C_i : 1 \leq i \leq p\}$*. Then there exists a cluster* $C \in \mathcal{C}$ *such that* $\bigcup_{i=1}^{p}\{C_i \cap C_0\} \bigcup\{x\} \subseteq C$*.*

Proof. First, by Lemma 1 applied to C_0, C_1, and C_2, there exits a cluster C' containing $(C_0 \cap C_1) \bigcup (C_0 \cap C_2) \bigcup \{x\}$. Now assume, by induction, that there exists a cluster C'' such that $\bigcup_{i=1}^{k}\{C_i \cap C_0\} \bigcup \{x\} \subseteq C''$, where $k < p$. By Lemma 1 applied to C_0, C'', and C_{k+1}, there exists a cluster C^* such that $\bigcup_{i=1}^{k+1}\{C_i \cap C_0\} \bigcup \{x\} \subseteq C^*$. □

We will need also the following well-known characterization of chordal graphs [5, 7, 8]. Recall that a graph is chordal if it does not contain any induced cycles of length greater than 3. A vertex $v \in V$ is simplicial in G if $N_G(v)$ is a clique in G.

Theorem 1. *[5, 7, 8] Let G be a graph. The following statements are equivalent:*

1. *G is a chordal graph;*
2. *There exists a tree-decomposition T of G where nodes of T are the maximal cliques of G.*
3. *G has a perfect elimination ordering, i.e. an ordering $v_1, v_2, \dots, v_n$ of vertices of G such that for any i, $i \in \{1, 2, \dots, n\}$, vertex v_i is simplicial in the graph $G(v_i, \dots, v_n)$, a subgraph of G induced by vertices $v_i, \dots, v_n$.*

Lemma 2. *The following statements are equivalent.*

1. *A graph $G = (V, E)$ admits an acyclic (R, D)-clustering.*
2. *For a graph $G = (V, E)$ there exists a graph $G^+ = (V, E^+)$ such that $E \subseteq E^+$, G^+ is chordal, and for any maximal clique X of G^+, $diam_G(X) \le D$ and $rad_G(X) \le R$.*

Proof. (1) ⇒ (2) Assume that $G = (V, E)$ admits an acyclic (R, D)-clustering with the cluster set $\mathcal{C}$. Consider a graph $G^+ = (V, E^+)$, where $E^+ = \{uv : u, v \in V$ and there exists a cluster $C \in \mathcal{C}$ such that $u, v \in C\}$. By condition 2 of Definition 1, $E \subseteq E^+$. By construction of G^+, any cluster $C \in \mathcal{C}$ is a clique in G^+. We will show that C is a maximal clique of G^+ as follows. Assume that $C = \{c_1, c_2, \dots, c_p\}$ and, by contradiction, there exists a vertex $x \in V$ such that $x \notin C$ and $C \subseteq N_{G^+}(x)$. By construction of G^+, there must be a cluster $C_1 \in \mathcal{C}$ such that $x, c_1 \in C_1$. Assume, by induction hypothesis, that there exists a cluster $C_k \in \mathcal{C}$ such that $x, c_1, c_2, \dots, c_k \in C_k$, $k < p$. Since $xc_{k+1} \in E^+$, there exists a cluster C_{k+1} such that $x, c_{k+1} \in C_{k+1}$. Applying Lemma 1 to C, C_k and C_{k+1}, we obtain that there exists a cluster C' such that $(C \cap C_k) \cup (C \cap C_{k+1}) \cup (C_k \cap C_{k+1}) = \{x, c_1, \dots, c_k, c_{k+1}\} \subseteq C'$. Thus, there must exist a cluster C_p such that $x, c_1, \dots, c_p \in C_p$ and $C \subset C_p$, This contradiction with acyclic clustering being reduced establishes that any node of T is a maximal clique in G^+. It is easy to see that T is a tree-decomposition for G^+.

Let now $X = \{x_1, x_2, \dots, x_p\}$ be a maximal clique of G^+. We will show that there exists a cluster $C \in \mathcal{C}$ such that $X = C$. By construction of G^+, there exists a cluster containing x_1 and x_2. Assume, by induction, that there exists a cluster $C' \in \mathcal{C}$ such that $x_1, x_2, \dots x_k \in C'$, $k < p$. Since there are edges $x_{k+1}x_i$ in G^+ $(1 \le i \le k)$, there exist clusters C_i such that $x_{k+1}, x_i \in C_i$. Notice that $x_i \in C' \cap C_i \ne \emptyset$, $1 \le i \le k$, and $x_{k+1} \in C_i \cap C_j$ for $1 \le i \ne j \le k$. By Corollary 1, there exists a cluster C^* such that $\{x_1, x_2, \dots, x_k, x_{k+1}\} \subseteq C^*$. Thus, we proved that for any maximal clique X of G^+ there exists a cluster C such that $X \subseteq C$. Taking into account that any cluster C of T is a maximal clique in G^+, we immediately have $X = C$. Thus, by Theorem 1, G^+ is a chordal graph.

Since the radius and the diameter of clusters in G are R and D, correspondingly, we immediately have that for any maximal clique X of G^+, $diam_G(X) \le D$ and $rad_G(X) \le R$.

(2) ⇒ (1) Since G^+ is chordal, by Theorem 1, there exists a tree-decomposition T for G^+, where clusters are the maximal cliques of G^+. Since G and G^+ have the same vertex set and $E \subseteq E^+$, it is easy to see that T is a tree-decomposition for G. Since for any maximal clique X of G^+, $diam_G(X) \le D$ and $rad_G(X) \le R$, we immediately conclude that G admits an acyclic (R, D)-clustering. □

Since a chordal graph can have at most n maximal cliques [20], from Lemma 2 we obtain that any acyclic (R, D)-clustering has at most n clusters, i.e., $\kappa \le n$.

3 Routing Scheme

Let G be a graph that admits an acyclic (R, D)-clustering and T be a tree-decomposition associated with it. We assume that T is rooted (say, at C_1). In a rooted tree T, $nca_T(X, Y)$ denotes the nearest common ancestor of nodes X and Y of T.

Definition 2. *For every vertex u of G, the ball of u, denoted by $B(u)$, is a node Z of T with minimum depth such that $u \in Z$.*

It is well known that any tree T with κ nodes has a node C, called a *centroid* and computable in $O(\kappa)$ time, such that any maximal by inclusion subtree of T, not containing C, (i.e., any connected component of $T \setminus C$) has at most $\kappa/2$ nodes. For the tree T of acyclic clustering we build a hierarchical tree H recursively as follows. All nodes of T are nodes in H. The root of H is C, a centroid of T, and the children are the roots of the hierarchical trees of the connected components of $T \setminus C$. Note that the height of H is $O(\log \kappa)$. The hierarchical tree for the graph in Fig. 1 is given in Fig. 2.

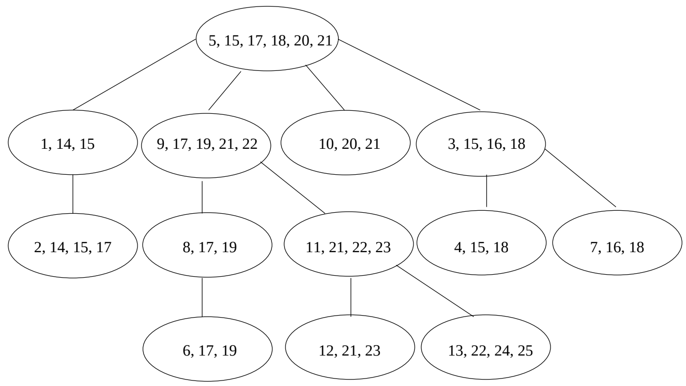

Fig. 2. A hierarchical tree H for the graph in Fig. 1

Let $G = (V, E)$ be a graph that admits an induced acyclic (R, D)-clustering with $R \leq 2$ and $D \leq 2$. Let X be a node of H and u be a vertex of G such that $u \notin X$ and $B(u)$ is a descendant of X in H. Let $P = \{u = z_0, z_1, z_2, \ldots, z_k = x^*\}$, $x^* \in X$, be a shortest path of G from u to X.

Let $C_0 = B(u)$ and C_i be the cluster closest to C_{i-1} in T such that $z_{i-1}, z_i \in C_i$, $1 \leq i \leq k$. Note that such clusters exist by condition 2 of Definition 1. Let Q_i be the shortest path in T between C_{i-1} and C_i, $1 \leq i \leq k$. Let Q_{k+1} be the shortest path in T between C_k and X. Observe that, by condition 3 of Definition 1, $z_{i-1} \in Y$ for all $Y \in Q_i$. Let $Q(P) = \bigcup_{i=1}^{k+1} Q_i$ be a path between $B(u)$ and X and Q' be the shortest path between $B(u)$ and X in T. Note that, in general, $Q(P)$ is not a simple path, and $Q' \subseteq Q(P)$ for any path P between u and X.

For any two clusters Y and Z such that $Y, Z \in Q'$, we say that Y precedes Z, denoted by $Y \prec Z$, if Y is closer to $B(u)$ in T than Z. We use a notation $Y \preceq Z$ if $Y \prec Z$ or $Y = Z$.

Lemma 3. *There exists a shortest path $P = \{u = z_0, z_1, z_2, \ldots, z_k = x^*\}$ between u and X such that $Q(P) = Q'$.*

Proof. Obviously, $C_0 = B(u) \in Q'$ for any P. Assume, by induction, that there exists a path $P = \{z_0, z_1, z_2, \ldots, z_k\}$ between u and X such that $C_l \in Q'$ for $0 \leq l \leq i - 1 < k$ and

$C_0 \preceq C_1 \preceq \ldots \preceq C_{i-1}$. We will show that there exists a path P' between u and X such that $C'_l \in Q'$ for $0 \leq l \leq i$ and $C'_0 \preceq C'_1 \preceq \ldots \preceq C'_{i-1} \preceq C'_i$ as follows (see Fig. 3 for an illustration).

Let C be a cluster closest to C_i in T such that $C \in Q'$. Since $X \in Q'$, there exists an integer p such that $i < p \leq k+1$ and $C \in Q_p$. Let $j > 0$ be the smallest number such that $C \in Q_{i+j}$. Notice that $z_{i+j-1} \in C$. Since $z_{i-1} \in C$ and C has diameter 2, we immediately obtain that $1 \leq j \leq 2$. Note that $z_i \notin C$, otherwise $C = C_i \in Q'$, a contradiction with $C_i \notin Q'$. Thus, $j = 2$, and C contains z_{i+1}.

We claim that $C_{i-1} \prec C$. Otherwise, C_{i-1} would contain either z_i, meaning $C_i = C_{i-1} \in Q'$, a contradiction, or z_q, $q > i$, which is not possible, since C_{i-1} has diameter 2 and P is a shortest path.

Let C^* be the cluster closest to C_{i-1} in T such that $z_{i+1} \in C^*$. Since $z_{i+1} \notin C_{i-1}$, we have $C_{i-1} \prec C^* \preceq C$. Since C^* is on the path in T between C_{i-1} and C_i, by condition 3 of Definition 1, $z_{i-1} \in C^*$. Recall that P is a shortest path and, therefore, z_{i-1} and z_{i+1} are not adjacent in G. Since C^* has induced diameter 2, there exists a vertex $z^* \in C^*$ such that z^* is adjacent to both z_{i-1} and z_{i+1}.

We replace z_i with z^* in P and obtain a new shortest path

$$P' = \{z_0, z_1, \ldots, z_{i-1}, z^*, z_{i+1}, \ldots, z_k\}.$$

Clearly, the paths $Q(P')$ and $Q(P)$ have a common prefix $\bigcup_{i=1}^{i-1} Q_i$.

Let C'_i be the cluster closest to C_{i-1} such that $z_{i-1}, z^* \in C'_i$. We will prove that $C'_i \in Q'$ and $C_{i-1} \preceq C'_i$ as follows. Assume, by contradiction, that $C'_i \notin Q'$. Let C''_i be the cluster closest to C'_i in T such that $C''_i \in Q'$. Since C''_i is on the path in T between C'_i and C^*, $z^* \in C'_i$ and $z^* \in C^*$, by condition 3 of Definition 1, $z^* \in C''_i$. Similarly, since C''_i is on the path in T between C'_i and C_{i-1}, $z_{i-1} \in C'_i$ and $z_{i-1} \in C_{i-1}$, by condition 3 of Definition 1, $z_{i-1} \in C''_i$. Obviously, C''_i is closer to C_{i-1} than C'_i. Since $z^*, z_{i-1} \in C''_i$, we obtain a contradiction, which proves $C'_i \in Q'$.

It remains to prove that $C_{i-1} \preceq C'_i$. Consider other possibilities. If $C_{i-2} \prec C'_i \prec C_{i-1}$, then, by condition 3 of Definition 1, $z_{i-2} \in C'_i$. In this case, C'_i is the cluster containing z_{i-2} and z_{i-1} and closer to C_{i-2} than C_{i-1}, a contradiction. If $C'_i \preceq C_{i-2}$, then C'_i contains a vertex z_{i-j}, $j > 2$, which is not possible since $z^* \in C'_i$, C'_i has diameter 2, and P is a shortest path. Thus, $C_{i-1} \preceq C'_i$. □

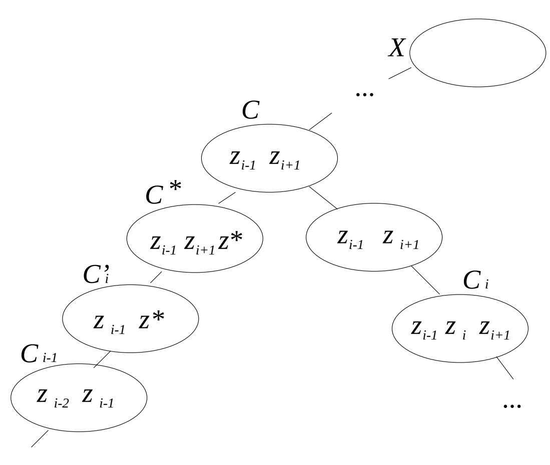

Fig. 3. Arrangements of clusters C_{i-1}, C_i, and C'_i in T.

Such path P is called *Q-simple* and can be constructed in $O(n^2)$ time by Algorithm *Q-simple path* presented in Fig. 4. The correctness of the algorithm follows from the proof of Lemma 3.

Input: $G = (V, E)$;
$P = \{u = z_0, z_1, \ldots, z_k\}$, a shortest path from u to X;
$Q' = \{B(u) = Y_0, Y_1, \ldots, Y_m = X\}$, the shortest path in T from $B(u)$ to X.
Output: $P' = \{u = z'_0, z'_1, \ldots, z'_k = z_k\}$, a Q-simple path from u to X.

```
set P' = ∅;
set j = 0;
for (i = 1; i < k; i++)
 set found = 0;
 while (!found)
    if (z_i ∈ Y_j) then P' ← {z_i}; found = 1;
    else if (z_{i+1} ∈ Y_j) then
           choose z* from {z ∈ Y_i : z* z_{i-1}, z* z_{i+1} ∈ E};
           P' ← {z*};
           found = 1;
       else j++;
return P'.
```

Fig. 4. *Q-simple path algorithm*

Lemma 4. *Let P be a Q-simple shortest path between u and X. Let $W = \{w \in P : B(w) \notin Q' = Q(P)\}$. Then $|W| \leq 3$.*

Proof. By Lemma 3, for each $w \in P$ there exists a cluster $C_w \in Q'$ such that $w \in C_w$. By Definition 2, for $w \in W$, $B(w) \in Q^*$ holds, where Q^* is the path between C_w and the root of T.

Clearly, $B(w) \notin Q' \cap Q^*$, otherwise $B(w) \in Q'$. Thus, $B(w)$ is on the path between $nca_T(B(u), X)$ and the root of T. Since $w \in C_w$, $w \in B(w)$, and $nca_T(B(u), X)$ is on the path between C_w and $B(w)$, by condition 3 of Definition 1, $w \in nca_T(B(u), X)$ for all $w \in W$. Since the diameter of clusters is 2, and P is a shortest path, $|P \cap nca_T(B(u), X)| \leq 3$. Thus, $|W| \leq 3$. □

Corollary 2. *Let P be a Q-simple shortest path from u to X in G. There are no more than 3 vertices z of P such that $B(z)$ is not a descendant of X in H.*

Proof. Let z be a vertex of P. Assume that $B(z) \in Q' = Q(P)$ and consider possible arrangements of nodes X, $B(u)$, and $B(z)$ in H, taking into account that X is an ancestor of $B(u)$ in H. First, note that $B(z)$ cannot be an ancestor of X in H, otherwise during the construction of hierarchical subtree rooted at $B(z)$, $B(u)$ and X would belong to different connected components of $T \setminus \{B(z)\}$, and, therefore, X could not be an ancestor of $B(u)$ in H. Second, if there exists a node Y such that X and $B(z)$ are descendants of Y in H, then $Y \in Q'$, and, again, during the construction of hierarchical subtree rooted at Y, $B(u)$ and X would belong to different connected components of $T \setminus \{Y\}$, and, therefore, X could not be an ancestor of $B(u)$ in H. Thus, if $B(z) \in Q'$, then the only possible arrangement is that $B(z)$ is a descendant of X in H. If $B(z) \notin Q'$, then, by Lemma 4, the number of such vertices z is bounded by 3. □

Lemma 5. *Let u and v be two vertices of G and $X = nca_H(B(u), B(v))$, then X is a separator between u and v in G.*

Proof. Let $P = \{u = z_0, z_1, z_2, \ldots, z_k = v\}$ be a path from u to v and $C_0 = B(u)$. Let C_i be the cluster closest to C_{i-1} in T such that $z_{i-1}, z_i \in C_i$, $1 \le i \le k$. Note that such clusters exist by condition 2 of Definition 1. Let Q_i be the shortest path in T between C_{i-1} and C_i, $1 \le i \le k$. Let Q_{k+1} be the shortest path in T between C_k and $B(v)$. Let $Q_{uv}(P) = \bigcup_{i=1}^{k+1} Q_i$ be a path between $B(u)$ and $B(v)$ and Q'_{uv} be the shortest path between $B(u)$ and $B(v)$ in T. Note that $Q'_{uv} \subseteq Q_{uv}(P)$ for any path P between u and v. By condition 3 of Definition 1, $z_{i-1} \in Y$ for all $Y \in Q_i$, $1 \le i \le k+1$. Thus, any node of $Q_{uv}(P)$ and, hence, any node of Q'_{uv} contains a vertex of any path P between u and v. By construction of H, $X \in Q'_{uv}$ and, therefore, X is a separator between u and v. □

For any node X of H, we construct a tree in G in the following way. Let U be a set of vertices of G such that $U \subseteq \{V \setminus X\}$ and $B(u)$ is a descendant of X in H for $u \in U$. First, for each $u \in U$, we construct a Q-simple shortest path $P(u)$ from u to X. Second, we construct a tree $t(X)$ spanning X such that its diameter $diam_{t(X)} = max_{x_1,x_2 \in X}\{dist_t(x_1, x_2)\}$ is minimal. Clearly, $diam_{t(X)} \le 2R$. Finally, we build a graph $G_X = \bigcup_{u \in U} P(u) \bigcup t(X)$ and construct in a Breadth-First-Search manner starting from $t(X)$ a special spanning tree $\mathcal{T}$ of G_X.

Lemma 6. *A spanning tree $\mathcal{T}$ of G_X can be constructed in such a way that for any $u \in U$, the path of $\mathcal{T}$ from u to X contains at most 3 vertices z such that $B(z)$ is not a descendent of X in H.*

Proof. Let $P(u)$ be a path in G_X from $u \in U$ to X and $W(P(u)) = \{z \in P(u) : B(z)$ is not a descendant of X in $H\}$. Let $L_i = \{v \in V(G_X) : dist_{G_X}(v, X) = i\}$, $i \ge 0$, be the BFS-layers of G_X with respect to X. A spanning tree $\mathcal{T}$ of G_X can be constructed starting from $t(X)$ in the following way. For all $u \in L_1$, the *parent*(u) is a vertex $x \in X$ such that $|W(P(u))|$ is minimum, where $P(u)$ is the path $\{u, x\}$ of G_X. For all $u \in L_i$, $i > 1$, *parent*(u) is a neighbor $v \in L_{i-1}$ of u in G_X such that $|W(P(u))|$ is minimum, where $P(u) = \{u, P(v)\}$. The above construction guarantees that u is connected to X in $\mathcal{T}$ via a path $P(u)$ with minimum possible $|W(P(u))|$. Since there is a path in G_X between $u \in U$ and X that is Q-simple, by Corollary 2, $|W(P(u))| \le 3$ for any $u \in U$. □

Lemma 7. *Let u and v be two vertices of G, $X = nca_H(B(u), B(v))$, $\mathcal{T}$ be the tree associated with X, and $P_{\mathcal{T}}$ be a path from u to v in $\mathcal{T}$. Then there are no more than 7 vertices z, such that $z \in P_{\mathcal{T}}$ and $B(z)$ is not a descendant of X in H.*

Proof. By Lemma 6, there are at most 3 such vertices on the path between u and X and there are at most 3 more such vertices on the path between v and X. Since X has induced diameter 2, there is at most 1 other such vertex of X that is on the path between u and v in $\mathcal{T}$. □

Lemma 8. *Let u and v be two vertices of G, $X = nca_H(B(u), B(v))$, and $\mathcal{T}$ be the tree associated with X, then $dist_{\mathcal{T}}(u, v) \le dist_G(u, v) + \Delta$, where $\Delta \le diam_{t(X)}$.*

Proof. By Lemma 5, X is a separator between u and v. Let P_G be a shortest path from u to v in G. Let $u' \in P_G$ be the vertex closest to u such that $u' \in X$ and let $v' \in P_G$ be the vertex closest to v such that $v' \in X$. Clearly,

$$dist_G(u, v) = dist_G(u, u') + dist_G(u', v') + dist_G(v', v). \tag{1}$$

Similarly, let $P_{\mathcal{T}}$ be the path from u to v in $\mathcal{T}$. Let $u'' \in P_{\mathcal{T}}$ be the vertex closest to u such that $u'' \in X$ and let $v'' \in P_{\mathcal{T}}$ be the vertex closest to v such that $v'' \in X$. Clearly,

$$dist_{\mathcal{T}}(u,v) = dist_{\mathcal{T}}(u,u'') + dist_{\mathcal{T}}(u'',v'') + dist_{\mathcal{T}}(v'',v). \qquad (2)$$

From (1) and (2), we have

$$dist_{\mathcal{T}}(u,v) = dist_G(u,v) + [dist_{\mathcal{T}}(u,u'') - dist_G(u,u')] + [dist_{\mathcal{T}}(u'',v'') - dist_G(u',v')] + [dist_{\mathcal{T}}(v'',v) - dist_G(v',v)]. \qquad (3)$$

We observe that, by construction of $\mathcal{T}$, $dist_{\mathcal{T}}(v'',u'') \leq diam_{t(X)}$, $dist_{\mathcal{T}}(u,u'') \leq dist_G(u,u')$, and $dist_{\mathcal{T}}(v'',v) \leq dist_G(v',v)$. Thus, from (3), we immediately have that $dist_{\mathcal{T}}(u,v) \leq dist_G(u,v) + \Delta$ where $\Delta \leq diam_{t(X)}$. □

Theorem 2. *If a graph G admits an induced acyclic (R,D)-clustering with $R \leq 2$ and $D \leq 2$, then G has a loop-free routing scheme of deviation $\Delta \leq 2R$ with addresses and routing labels of size $O(\log^3 n/\log\log n)$ bits per vertex and $O(1)$ routing protocol.*

Proof. We associate a tree $\mathcal{T}(X)$, constructed as described above, with each node X of the hierarchical tree H. Each vertex u of G only stores routing information for trees $\mathcal{T}(X)$ such that $B(u)$ is a descendant of X. Since the height of H is at most $\log n$, there are at most $\log n$ such trees. For every pair of vertices u and v we can find $X = nca_H(B(u), B(v))$. This can be done in constant time by introducing a binary label of $O(\log n)$ bits in the address of each vertex [18]. By Lemma 8, we have $dist_{\mathcal{T}}(u,v) \leq dist_G(u,v) + \Delta$, where $\Delta \leq diam_{t(X)} \leq 2R$.

To implement the routing in the tree $\mathcal{T}(X)$ we use the scheme presented in [14]. This scheme uses addresses and labels of length $O(\log^2 n/\log\log n)$ bits and runs in constant time.

Along the route between u and v in $\mathcal{T}(X)$, there might be vertices w such that $B(w)$ is not a descendant of X in H and therefore w does not have the routing label for the tree $\mathcal{T}(X)$. By Lemma 7, the number of such vertices is constant. We store in advance port numbers for such vertices in routing labels, which requires each vertex u to have an additional $O(\log n)$-bit label for each of $\log n$ trees. □

We distinguish a special case of induced acyclic clustering with radius $R = 1$.

Corollary 3. *If G admits an induced acyclic (R,D)-clustering with $R = 1$ and $D \leq 2$, then G has a loop-free routing scheme of deviation 2 with addresses and routing labels of size $O(\log^3 n/\log\log n)$ bits per vertex and $O(1)$ routing protocol.*

4 Chordal-Bipartite Graphs

A bipartite graph is *chordal bipartite* if it does not contain any induced cycles of length greater than 4 [20].

Let $G = (X \cup Y, E)$ be a chordal bipartite graph. We construct a graph $G^+ = (X \cup Y, E^+)$ by adding edges between any two vertices $x_1, x_2 \in X$ for which there exists a vertex $y \in Y$ such that $x_1y,\ x_2y \in E$.

Lemma 9. *The graph G^+ is chordal.*

Proof. First notice that any $y \in Y$ is simplicial in G^+ by construction of G^+. Assume now, by contradiction, that there is an induced cycle C_p of length p, $p > 3$, in G^+. Necessarily, all vertices of C_p are from part X of G, since C_p is induced and all vertices from Y are simplicial in G^+. Let $C_p = \{x_1, x_2, \ldots, x_p, x_1\}$. For any edge x_ix_{i+1} of C_p (including edge x_px_1), since it is not an edge of G, there must be a vertex $y_i \in Y$ such that both x_i and x_{i+1} are adjacent to y_i

in G. Also, since C_p is induced in G^+, y_i is not adjacent to any other vertex of C_p. Therefore, a cycle $\{x_1, y_1, x_2, y_2, \ldots, x_p, y_p, x_1\}$ of G must be induced. But, since its length is $2p \geq 8$, a contradiction with G being a chordal bipartite graph arises. □

Lemma 10. *For any maximal clique C of G^+ there exists a vertex $y \in Y$ such that $N_G[y] = C$.*

Proof. Let $|C| = p$. First note that, by construction of G^+, the clique C can contain at most one vertex from Y. If C contains a vertex from Y (say $y \in C \cap Y$) then for all $v \in C \setminus \{y\}$, vy is an edge of G, and therefore $C \subseteq N_G[y]$ must hold. Let now $C \cap Y = \emptyset$. By induction on p, we will show that there exists a vertex $y \in Y$ such that $C \subset N_G[y]$. Since G is connected, any vertex $x \in C \subseteq X$ has a neighbor in Y. Also, by construction of G^+, for any edge uv of G^+ with $u, v \in X$ there must exists a vertex $y \in Y$ adjacent to both u and v. Assume now, by induction, that each $p-1$ vertices of C have a common neighbor $y \in Y$. Consider three different vertices a, b, and c in C and three corresponding vertices a', b', and c' in Y such that $C \setminus \{a\} \subset N_G[a']$, $C \setminus \{b\} \subset N_G[b']$, and $C \setminus \{c\} \subset N_G[c']$. Since the graph G cannot have any induced cycles of length 6, the cycle $\{a, b', c, a', b, c', a\}$ of G cannot be induced. Without loss of generality, assume that a is adjacent to a' in G. But then, all p vertices of C are contained in $N_G[a']$. □

From Lemmas 2, 9, and 10 we immediately establish the following result.

Lemma 11. *Any chordal-bipartite graph G admits an induced acyclic (R, D)-clustering with $R = 1$ and $D = 2$. Moreover, $\mathcal{C} = \{C_1, C_2, ..., C_{|Y|}\}$, where $C_i = N_G[y_i]$, $y_i \in Y$.*

Recalling Corollary 3, we conclude.

Theorem 3. *Any chordal-bipartite graph G admits a loop-free routing scheme of deviation $\Delta = 2$ with addresses and routing labels of size $O(log^3 n/\log\log n)$ bits per vertex and $O(1)$ routing protocol.*

5 Homogeneously Orderable Graphs

A nonempty set $U \subseteq V$ is homogeneous in $G = (V, E)$ if all vertices of U have the same neighborhood in $V \setminus U$. The *disk* of radius k centered at v is the set of vertices of distance at most k from v: $D(v, k) = \{u \in V : dist_G(u, v) \leq k\} = \bigcup_{i=0}^{k} N^i(v)$. For $U \subseteq V$ we define $D(U, k) = \bigcup_{u \in U} D(u, k)$. The *kth power* G^k of a graph $G = (V, E)$ is the graph with vertex set V and edges between vertices u and v with distance $dist_G(u, v) \leq k$. A subset U of V is a *k-set* of G if U induces a clique in G^k.

A vertex v of G with $|V| > 1$ is h-extremal if there is a proper subset $H \subset D(v, 2)$ which is homogeneous in G and for which $D(v, 2) \subseteq D(H, 1)$ holds. A vertex ordering $v_1, .., v_n$ is a homogeneous elimination ordering of vertices of G if for every i, v_i is h-extremal in the induced subgraph $G_i = G(v_i ... v_n)$. G is *homogeneously orderable* if it has a homogeneous elimination ordering. As it was shown in [3], homogeneously orderable graphs include such well known classes of graphs as distance-hereditary graphs, strongly chordal graphs, dually chordal graphs, and homogeneous graphs (for the definitions see [4]).

Let U_1, U_2 be disjoint sets in V. If every vertex of U_1 is adjacent to every vertex of U_2 then U_1 and U_2 form a *join*, denoted by $U_1 \bowtie U_2$. A set $U \subseteq V$ is *join-split* if U is the join of two non-empty sets, i.e. $U = U_1 \bowtie U_2$.

The following theorem represents a well-known characterization of homogeneously orderable graphs.

Theorem 4. *[3] G is homogeneously orderable if and only if G^2 is chordal and every maximal 2-set of G is join-split.*

Taking into account Lemma 2 and considering $G^+ = G^2$, we immediately conclude.

Corollary 4. *Any homogeneously orderable graph G admits an induced acyclic clustering with $R = 2$ and $D = 2$. The cluster set $\mathcal{C}$ is the collection of all maximal 2-sets of G.*

This corollary and Theorem 2 already imply for homogeneously orderable graphs existence of a loop-free routing scheme of deviation $\Delta = 4$ with addresses and routing labels of size $O(log^3 n/\log\log n)$ bits per vertex and $O(1)$ routing protocol. In what follows, we will show that, in fact, the scheme described in Section 3 gives for homogeneously orderable graphs a routing scheme of deviation $\Delta = 2$.

For a vertex function $r : V \to \mathcal{N}$ (note that zero is assumed to be a natural number), a clique $C \subseteq V$ *r-dominates* a subset $S \subseteq V$ if for each vertex $u \in S \setminus C$ there is a vertex $x \in C$ such that $dist_G(u, x) \le r(u)$ holds. We will use the following known result.

Theorem 5. *[12] For any homogeneously orderable graph G with vertex function $r : V \to \mathcal{N}$ and for any subset S of V, S is r-dominated by some clique C if and only if $dist_G(x, y) \le r(x) + r(y) + 1$ for all $x, y \in S$.*

Lemma 12. *Let $U = U_1 \bowtie U_2$ be a maximal 2-set of a homogeneously orderable graph G, z be a vertex of G such that $z \notin U$, and $d = dist_G(z, U)$ be the distance between z and the set U. Then, for any two vertices $v, w \in U$ such that $dist_G(z, v) = dist_G(z, w) = d$, either $v, w \in U_1$ or $v, w \in U_2$ holds.*

Proof. Assume, by way of contradiction, that $v \in U_1$ and $w \in U_2$. Since U is join-split, $dist_G(v, w) = 1$. Let $S = \{z, v, w\}$ and a vertex function r be defined as $r(z) = d - 1$, $r(v) = r(w) = 0$ and $r(u) = diam(G) := diam_G(V)$ for any $u \in V \setminus S$. Note that the inequality $dist_G(x, y) \le r(x) + r(y) + 1$ holds for all $x, y \in S$. According to Lemma 5, S is r-dominated by a clique $C \subseteq V$. Thus, there must be a vertex $z' \in C$ such that $dist_G(z', v) = dist_G(z', w) = 1$ and $dist_G(z, z') = d - 1$. It is easy to see that $dist_G(z', u) \le 2$ for any $u \in U$ and, since U is a maximal 2-set, $z' \in U$ must hold. Since $dist_G(z, z') = d - 1$, we obtain $dist_G(z, U) = d - 1$, which is a contradiction. □

Let T be a tree decomposition of a homogeneously orderable graph $G = (V, E)$ and H be its hierarchical tree. With each node $X = U_1 \bowtie U_2$ of H we associate a spanning tree $\mathcal{T}$ of G_X as described in Section 3, where a spanning tree $t(X)$ of X is constructed as follows. Beginning at an arbitrary vertex $s_1 \in U_1$, we visit all vertices in U_2, then continuing from any vertex $s_2 \in U_2$, we visit all vertices in $U_1 \setminus \{s_1\}$. Clearly, $diam_{t(X)} = 3$.

Lemma 13. *Let u and v be two vertices of a homogeneously orderable graph G, $X = nca_H(B(u), B(v))$, and $\mathcal{T}$ be a tree associated with node X. Then $dist_{\mathcal{T}}(u, v) \le dist_G(u, v) + \Delta$ with $\Delta = 2$.*

Proof. From Lemma 8, we have

$$\begin{aligned} dist_{\mathcal{T}}(u, v) = dist_G(u, v) &+ [dist_{\mathcal{T}}(u, u'') - dist_G(u, u')] + \\ &[dist_{\mathcal{T}}(u'', v'') - dist_G(u', v')] + [dist_{\mathcal{T}}(v'', v) - dist_G(v', v)], \end{aligned} \tag{4}$$

where u' and u'' are vertices closest to u such that $u' \in P_G$, $u'' \in P_{\mathcal{T}}$, $u', u'' \in X$, v' and v'' are vertices closest to v such that $v' \in P_G$, $v'' \in P_{\mathcal{T}}$, $v', v'' \in X$, and P_G and $P_{\mathcal{T}}$ are shortest paths from u to v in G and in $\mathcal{T}$, respectively.

We notice that, by construction of $\mathcal{T}$, $dist_{\mathcal{T}}(v'', u'') \leq 3$, $dist_{\mathcal{T}}(u, u'') \leq dist_G(u, u')$, and $dist_{\mathcal{T}}(v'', v) \leq dist_G(v', v)$. Hence, $dist_{\mathcal{T}}(u, v) \leq dist_G(u, v) + 2$, if at least one of the following holds: $dist_{\mathcal{T}}(u, u'') < dist_G(u, u')$, $dist_{\mathcal{T}}(v'', v) < dist_G(v', v)$, or $dist_{\mathcal{T}}(u'', v'') < 3$.

Therefore, assume that $dist_{\mathcal{T}}(u, u'') = dist_G(u, u')$, $dist_{\mathcal{T}}(v'', v) = dist_G(v', v)$, and $dist_{\mathcal{T}}(u'', v'') = 3$. Since $dist_{\mathcal{T}}(u'', v'') = 3$, u'' and v'' belong to different parts of $U = U_1 \bowtie U_2$. Without loss of generality, assume that $u'' \in U_1$ and $v'' \in U_2$. Since $dist_{\mathcal{T}}(u, u'') = dist_G(u, u'') = dist_G(u, u')$ and $dist_{\mathcal{T}}(v'', v) = dist_G(v'', v) = dist_G(v', v)$, by Lemma 12, we have $u' \in U_1$ and $v' \in U_2$. Thus, $u' \neq v'$, i.e., $dist_G(u', v') > 0$, and from (4) we immediately have $dist_{\mathcal{T}}(u, v) \leq dist_G(u, v) + \Delta$, where $\Delta = 2$. □

Taking into account Theorem 2, we obtain the following.

Theorem 6. *Any homogeneously orderable graph G admits a loop-free routing scheme of deviation $\Delta = 2$ with addresses and routing labels of size $O(log^3 n/\log\log n)$ bits per vertex and $O(1)$ routing protocol.*

6 Interval Graphs

The intersection graph of a family of n sets is the graph where the vertices are the sets, and the edges are the pairs of sets that intersect. A graph G is an *interval graph* if it is the intersection graph of a finite set of intervals (line segments) on a line.

Let $G = (V, E)$ be an interval graph. Since for a given interval graph the interval model of G (i.e., a corresponding set of intervals) can be constructed in linear $O(|V| + |E|)$ time (see, e.g., [2, 6, 23]), in what follows, we will assume that an interval model of G is given.

It is well known [20] that interval graphs form a proper subclass of chordal graphs. Hence, by Theorem 1, we have

Lemma 14. *Any interval graph G admits an induced acyclic (R, D)-clustering with $R = D = 1$, where clusters are the maximal cliques of G.*

This lemma and Theorem 2 already imply for interval graphs existence of a loop-free routing scheme of deviation $\Delta = 2$ with addresses and routing labels of size $O(log^3 n/\log\log n)$ bits per vertex and $O(1)$ routing protocol. In what follows, we will show that even a deviation $\Delta = 1$ can be achived.

Lemma 15. *For any maximal clique X of an interval graph $G = (V, E)$ there exist two vertices x_l and x_r such that $dist_G(v, X) = dist_G(v, x_l)$ or $dist_G(v, X) = dist_G(v, x_r)$ for any vertex $v \in V \setminus X$.*

Proof. We consider an interval model of G and consequently assign numbers from 1 to $2n$ to the endpoints of the line segments, from left to right. Each vertex v of the graph is represented by a distinct pair of integers $l(v), r(v)$, where $l(v)$ and $r(v)$ are the numbers of the left and right endpoints of the segment representing vertex v. For any maximal clique X, let x_l be a vertex of X such that $l(x_l)$ is smallest and x_r be a vertex of X such that $r(x_r)$ is largest. See Fig. 5 for an illustration.

Let x be a vertex of X such that $P = \{v, z_1, \ldots, z_k, x\}$ is a shortest path from v to X. By definition of x_l and x_r, z_k must be adjacent to x_l or to x_r. Thus, either $P' = \{v, z_1, \ldots, z_k, x_l\}$ or $P'' = \{v, z_1, \ldots, z_k, x_r\}$ is a shortest path from v to X. □

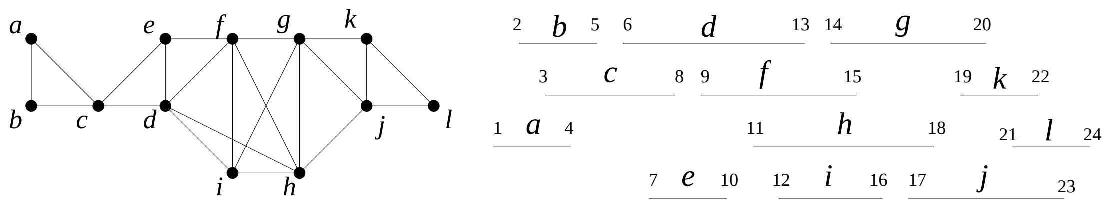

Fig. 5. An interval graph and its interval model. For $X = \{d, f, h, i\}$, $x_r = h$ and $x_l = d$.

Let H be a hierarchical tree for G. For any node X of H, we construct a spanning tree $\mathcal{T}$ of G_X in the following way. Let U be a set of vertices of G such that $U \subseteq \{V \setminus X\}$ and $B(u)$ is a descendant of X in H for any $u \in U$. For each $u \in U$, we construct a Q-simple shortest path $P(u) = \{u, z_1, \ldots, z_k, x\}$ from u to X such that x is either x_l or x_r. Since X is a clique, a spanning tree $t(X)$ is a star with center at x_l. Finally, we build a graph $G_X = \bigcup_{u \in U} P(u) \bigcup t(X)$ and construct in a Breadth-First-Search manner starting from $t(X)$ a special spanning tree $\mathcal{T}$ of G_X (see Lemma 6).

Lemma 16. *Let u and v be two vertices of an interval graph G, $X = nca_H(B(u), B(v))$, and $\mathcal{T}$ be a tree associated with X. Then, $dist_{\mathcal{T}}(u, v) \leq dist_G(u, v) + \Delta$ with $\Delta = 1$.*

Proof. From Lemma 8, have

$$dist_{\mathcal{T}}(u, v) = dist_G(u, v) + [dist_{\mathcal{T}}(u, u'') - dist_G(u, u')] + [dist_{\mathcal{T}}(u'', v'') - dist_G(u', v')] + [dist_{\mathcal{T}}(v'', v) - dist_G(v', v)], \quad (5)$$

where u' and u'' are vertices closest to u such that $u' \in P_G$, $u'' \in P_{\mathcal{T}}$, $u', u'' \in X$, v' and v'' are vertices closest to v such that $v' \in P_G$, $v'' \in P_{\mathcal{T}}$, $v', v'' \in X$, and P_G and $P_{\mathcal{T}}$ are shortest paths from u to v in G and in $\mathcal{T}$, respectively.

We notice that, by construction of $\mathcal{T}$, u'' and v'' are x_l or x_r. Taking into account that $dist_{\mathcal{T}}(v'', u'') \leq 1$, $dist_{\mathcal{T}}(u, u'') = dist_G(u, u')$ and $dist_{\mathcal{T}}(v'', v) = dist_G(v', v)$, from (5), we immediately obtain $dist_{\mathcal{T}}(u, v) \leq dist_G(u, v) + 1$. □

Thus, from Theorem 2, we have the following.

Theorem 7. *Any interval graph G admits a loop-free routing scheme of deviation $\Delta = 1$ with addresses and routing labels of size $O(log^3 n / \log\log n)$ bits per vertex and $O(1)$ routing protocol.*

Acknowledgment The authors would like to thank Cyril Gavoille for careful reading the preliminary draft of this paper and pointing out to an error in that version. As he has shown, old method - without considering special (Q-simple) paths - could result in a scheme with labels of size $O(log^3 n)$ bits per vertex and $O(loglogn)$ routing protocol even on chordal bipartite graphs (although with deviation 2, not 4 as in [9]). We are grateful to him also for acquaintance with the results of [9].

References

1. G. Ausiello, A. D'Arti, and M. Moscarini, Chordality properties on graphs and minimal conceptual connections in semantic data models, *J. Comput. System Sci.*, 33 (1986), 179–202.
2. K.S. Booth and G.S. Lueker, Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms. *J. Comput. System Sci.* 13 (1976), 335–379.

3. A. Brandstädt, F.F. Dragan, and F. Nicolai, Homogeneously orderable graphs, *Theoretical Computer Science*, 172 (1997) 209–232.
4. A. Brandstädt, V.B. Le, J. Spinrad, Graph Classes: A Survey, *SIAM Monographs on Discrete Math. Appl.,* (SIAM, Philadelphia, 1999)
5. P. Buneman, A characterization of rigit circuit graphs, *Discerete Mathematics*, 9 (1974), 205–212.
6. D.G. Corneil, S. Olariu, and L. Stewart, The ultimate interval graph recognition algorithm? (extended abstract). In *Proceedings of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 175-180, San Francisco, California, 25-27 January 1998.
7. R. Diestel, *Graph Theory*, 2nd eddition, volume 173 of Graduate Text in Mathematics, Springer, 2000.
8. G. A. Dirac, On rigit circuit graphs, *Abh. Math. Sem. Univ. Hamburg*, 25 (1961), 71-76.
9. Y. Dourisboure, Routage compact et longueur arborescente, December 2003, *PhD Thesis*, LaBRI, University of Bordeaux I.
10. Y. Dourisboure and C. Gavoille, Improved Compact Routing Scheme for Chordal Graphs, In prceedings to the *16th International Conference on Distributed Computing (DISC 2002)*, Toulose, France, October 2002, *Lecture Notes in Computer Science* 2508, Springer, 252–264.
11. F.F. Dragan and I. Lomonosov, New Routing Schemes for Interval Graphs, Circular-Arc Graphs, and Permutation Graphs, in *Proceeding of the 14th IASTED International Conference on Parallel and Distributed Computing and Systems*, Cambridge, USA, 2003, 78–83.
12. F.F. Dragan and F. Nicolai, r-Domination Problems on Homogeneously Orderable Graphs, *Networks*, 30 (1997), 121-131.
13. P. Fraigniaud and C. Gavoille, Memory requirements for univesal routing schemes, In *Proceedings of the 14th Annual ACM Symposium on Principles of Distributed Computing*, Ontario, Canada, 1995, pp. 223–230.
14. P. Fraigniaud and C. Gavoille, Routing in trees, *28th International Colloquim on Atomata, Languages and Programming (ICALP)*, volume 2076 of Lecture Notes in Computer Science, 757–772, 2001.
15. C. Gavoille, A survey on interval routing schemes, *Theoretical Computer Science*, 245 (1999), 217–253.
16. C. Gavoille, Routing in distributed networks: Overview and open problems, *ACM SIGACT News - Distributed Computing Column,* 32 (2001).
17. C. Gavoille and M. Gengler, Space-efficiency of routing schemes of stretch factor three, *Journal of Parallel and Distributed Computing*, 61 (2001), 679–687.
18. C. Gavoille, M. Katz, N. Katz, C. Paul, and D. Peleg, Approximate distance labeling schemes, *Research Report RR-1250-00*, LaBRI, University of Bordeaux, December 2000.
19. C. Gavoille and S. Pérennès, Memory requirements for routing in distributed networks, In *Proceedings of the 15th Annual ACM Symposium on Principles of Distributed Computing,* Philadelphia, Pennsylvania, 1996, pp. 125–133.
20. M.C. Gollumbic, *Algorithmic Graph Theory and Perfect Graphs*, Academic Press, New York, 1980.
21. J. van Leeuwen and R.B. Tan, Interval routing, *The Computer Journal,* 30 (1987), 298–307.
22. F. Leighton, *Introduction to Parallel Algorithms and Architechtures: Arrays - Trees - Hypercubes*, Morgan Kaufmann, 1992.
23. R.M. McConnell, Linear-time recognition of circular-arc graphs, In *Proceedings of the 42nd Annual IEEE Symposium on Foundations of Computer Science (FOCS01)*, Las Vegas (2001), 386-394.
24. D. Peleg, *Distributed computing – A locality-sensitive approach*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2000.
25. N. Robertson and P.D. Seymour. Graph minors. Algorithmic aspects of tree-width. *Journal of Algorithms*, 7 (1986), 309–322.
26. N. Santoro and R. Khatib, Labeling and implicit routing in networks, *The Computer Journal,* 28 (1985), 5–8.
27. M. Thorup and U. Zwick, Compact routing schemes, In 13^{th} *Ann. ACM Symp. on Par. Alg. and Arch.*, July 2001, pp. 1–10.
28. M. Thorup and U. Zwick, Approximate distance oracles, In 33^{rd} *Ann. ACM Symp. on Theory of Computing (STOC)*, July 2001, pp. 183–192.