INFORMATION AND COMPUTATION **139**, 92–109 (1997)
ARTICLE NO. IC972669

On Interval Routing Schemes and Treewidth*

Hans L. Bodlaender and Jan van Leeuwen

Department of Computer Science, Utrecht University, P.O. Box 80.089, 3508 TB Utrecht, The Netherlands
E-mail: hansb,jan@cs.ruu.nl

Richard Tan

Department of Computer Science, Utrecht University, P.O. Box 80.089, 3508 TB Utrecht, The Netherlands and Department of Computer Science, University of Sciences and Arts of Oklahoma, Chickasha, Oklahoma 73018
E-mail: rbtan@cs.ruu.nl

and

Dimitrios M. Thilikos

Department of Computer Science, Utrecht University, P.O. Box 80.089, 3508 TB Utrecht, The Netherlands
E-mail: sedthilk@cs.ruu.nl

In this paper, we investigate which processor networks allow k-label Interval Routing Schemes, under the assumption that costs of edges may vary. We show that for each fixed $k \geqslant 1$, the class of graphs allowing such routing schemes is closed under minor-taking in the domain of connected graphs, and hence has a linear time recognition algorithm. This result connects the theory of compact routing with the theory of graph minors and treewidth. We show that every graph that does not contain $K_{2,r}$ as a minor has treewidth at most $2r-2$. As a consequence, graphs that allow k-label Interval Routing Schemes under dynamic cost edges have treewidth at most $4k$. Similar results are shown for other types of Interval Routing Schemes. © 1997 Academic Press

* This research was partially supported by the ESPRIT Basic Actions Program of the EC under Contract 8141 (Project ALCOM II). The research of the second author was also partially supported by the Netherlands Organisation for Scientific Research (NWO) under Contract NF 62-376 (NFI Project ALADDIN: *Algorithmic Aspects of Parallel and Distributed Systems*). The research of the last author was supported by the Training and Mobility of Researchers (TMR) Program (EU Contract ERBFMBICT950198). Correspondence on this paper should be addressed to the first author.

0890-5401/97 $25.00
Copyright © 1997 by Academic Press
All rights of reproduction in any form reserved.

1. INTRODUCTION

A common problem in processor networks is that messages that are sent from one processor to another processor must be routed through the network. The classical solution is to give each processor a routing table, with an entry for each possible destination specifying over which link the message must be forwarded. A disadvantage of this method is that these tables grow with network size, and may become too large for larger processor networks.

Several different routing methods have been proposed that do not have this disadvantage. One such method is the interval routing method, together with its generalisation k-label interval routing and variants of these. An overview of these and other compact routing methods can be found in [19].

Interval routing was introduced by Santoro and Khatib [24] and van Leeuwen and Tan [18]. Several well-known classes of networks allow interval routing schemes that are optimal, in the sense that messages always follow the shortest path to their destination. The method was applied in the C104 Router Chip, used in the INMOS T9000 Transputer design [16].

Frederickson and Janardan [15] considered interval routing in the setting of dynamic cost links (i.e., in the case that the cost of edges is variable). Actually, they considered a variant of interval routing, called *strict* interval routing. For this, they gave a precise characterisation of the graphs with dynamic cost links which allow optimum strict interval routing schemes: these are exactly the *outerplanar* graphs. Another restriction of interval routing was introduced by Bakker, van Leeuwen, and Tan in [2]: linear interval routing. It has also been applied in concrete networks. Here, also a precise characterisation exists of the graphs which allow optimum linear interval routing schemes with dynamic cost links.

All of the interval routing schemes assume that each link has one unique label, which is a (possibly cyclic) interval of processor names. All can be generalised to multi-label schemes, where each link has a number of labels. We consider the k-label schemes: each link has at most k labels. The issue we study in this paper is: *which graphs allow* k*-label interval routing schemes in the setting of dynamic cost links.*

Surprisingly, new and deep graph theoretical results on graph minors due to Robertson and Seymour (see Section 2.1) can be used for the analysis of this problem. With the help of these results, we show *non-constructively* the existence of finite characterisations of which graphs allow certain routing schemes. Also, we give a non-constructive proof of the *existence* of linear time algorithms that check whether a desired routing scheme exists for a given graph. These algorithms heavily depend on the use of tree-decompositions. We show that graphs, allowing a k-label interval routing scheme (in the setting of dynamic cost links) have treewidth at most $4k$. This not only gives a partial characterisation of the graphs which have such routing schemes, but also, as the hidden constant factor of these algorithms is exponential in the treewidth of the tree-decomposition, it helps to decrease the running time of algorithms that would test the property.

As a main lemma, we show that every graph either contains $K_{2,r}$ as a minor, or has treewidth at most $2r-2$. This can be seen as a special case of a result of

Robertson and Seymour [21]: every planar graph $H=(V(H), E(H))$ has an associated constant c_H, such that any graph G either contains H as a minor or has treewidth at most c_H. The best general bound for c_H known is $20^{2(2|V(H)|+4|E(H)|)^5}$ [23]. Our result gives a much better bound in the case of graphs of the form $K_{2,r}$. Also, this result is constructive, and can be turned into an $O(rn)$ time algorithm, that either outputs that the input graph G has $K_{2,r}$ as a minor, or that outputs a tree-decomposition of G of treewidth at most $2r-2$. Similar results for other specific graphs can be found in [3] (trees), [14] (cycles and subgraphs of cycles), [6] (disjoint copies of K_3), and [5] (graphs that are minor of a circus graph and $(2\times k)$-grid). The result of this main lemma can be seen as an additional result, fitting into this framework. Applied to the routing problem, it gives the first graph-theoretic complexity bound on the graphs that admit optimal k-label interval routing schemes. Another consequence we discuss is that 'most' random graphs (even "sparse random graphs") do not allow k-label interval routing schemes under the dynamic cost edges assumption, for small values of k.

This paper is organised as follows. In Section 2, we give most necessary definitions and some preliminary results. In Section 3, we establish minor-closedness of the considered classes of graphs, each class containing those networks allowing certain types of k-label interval routing schemes. As a consequence, we obtain a non-existential proof of the existence of linear time membership algorithms for these classes. Also, slower, but constructive algorithms for these problems are given. In Section 4, we give the result on the treewidth of graphs, avoiding $K_{2,r}$ as a minor (as discussed above). We also mention a similar result for planar graphs with a better bound. Some open problems are mentioned in Section 5.

2. DEFINITIONS AND PRELIMINARY RESULTS

In this section, we introduce the most important definitions and mention some known results. In Section 2.1, we introduce graph-theoretic notions and results, and in Section 2.2, concepts and results from interval routing and its variants.

2.1. *Graph Theoretic Definitions and Preliminary Results*

All graphs in this paper will be assumed to be undirected, simple and finite. Given a graph G we denote as $V(G)$ and $E(G)$ the set of its vertices and edges respectively. The number of vertices of a graph $G=(V, E)$ will be denoted by $n=|V(G)|$. The notion of treewidth was introduced by Robertson and Seymour [21].

DEFINITION. A tree-decomposition of a graph $G=(V, E)$ is a pair $D=(X, T)$ with $T=(I, F)$ a tree and $X=\{X_i \mid i\in I\}$ a family of subsets of V, one for each node of T, such that

- $\bigcup_{i\in I} X_i = V$.
- For all edges $\{v, w\}\in E$, there exists an $i\in I$ with $v\in X_i$ and $w\in X_i$.
- For all $i, j, k\in I$: if j is on the path from i to k in T, then $X_i\cap X_k\subseteq X_j$.

The treewidth of a tree-decomposition $(\{X_i \mid i \in I\}, T = (I, F))$ is $\max_{i \in I} |X_i| - 1$. The treewidth of a graph G is the minimum treewidth over all possible tree-decompositions of G.

There are several well-known equivalent characterisations of the notion of treewidth; for instance, a graph has treewidth at most k, if and only if it is a partial k-tree, or a subgraph of a chordal graph with maximum clique size at most $k+1$ (see [17]).

A graph $G = (V, E)$ is said to be a *minor* of a graph $H = (W, F)$ if G can be obtained from H by a series of vertex deletions, edge deletions, and edge contractions, where an edge contraction is the operation that takes two adjacent vertices v and w and replaces it by a new vertex adjacent to all vertices that were adjacent to v or w. A class of graphs $\mathcal{G}$ is said to be *closed under taking of minors* if for every $G \in \mathcal{G}$, every minor H of G belongs to $\mathcal{G}$. For classes of graphs $\mathcal{G}$, $\mathcal{H}$, we say that $\mathcal{G}$ is *closed under taking of minors in the domain* $\mathcal{H}$ if for every graph $G \in \mathcal{G} \cap \mathcal{H}$, every minor H of G with $H \in \mathcal{H}$ belongs to $\mathcal{G}$.

In a long series of papers, Robertson and Seymour proved their famous graph minor theorem (formerly "Wagner's conjecture"):

THEOREM 1 (See [20]). *For every class of graphs $\mathcal{G}$ that is closed under taking of minors, there exists a finite set of graphs, called the obstruction set of $\mathcal{G}$,* $\mathrm{ob}(\mathcal{G})$, *such that for all graphs H, $H \in \mathcal{G}$ if and only if there is no graph G in the obstruction set of $\mathcal{G}$ that is a minor of H.*

Fellows and Langston [13] derived the following consequence and variant of this result.

THEOREM 2. *Let $\mathcal{G}$ be a class of graphs closed under taking of minors in the domain $\mathcal{H}$, with $\mathcal{G} \subseteq \mathcal{H}$. There exists a finite set of graphs, the obstruction set of $\mathcal{G}$ in $\mathcal{H}$,* $\mathrm{ob}_{\mathcal{H}}(\mathcal{G})$, *such that for all graphs $H \in \mathcal{H}$, $H \in \mathcal{G}$ if and only if there is no graph $G \in \mathrm{ob}_{\mathcal{H}}(\mathcal{G})$ that is a minor of H.*

It should be noted that the proofs of these results are (inherently) non-constructive. As for every fixed graph H, there exists an $O(n^3)$ time algorithm that tests whether H is a minor of a given graph G with n vertices [22], it follows that every minor-closed class of graphs has a cubic recognition algorithm, and every minor-closed class of graphs in a domain $\mathcal{H}$ has a cubic algorithm that tests whether graphs from $\mathcal{H}$ belong to $\mathcal{G}$. However, as the proof of Theorem 1 is non-constructive, we know that the algorithm exists, but we do not have the algorithm itself.

In several cases, faster algorithms exist.

THEOREM 3 [21]. *For every planar graph H, there exists a constant c_H, such that for every graph G, either H is a minor of G, or the treewidth of G is at most c_H.*

Moreover, for every fixed integer k and graph H, there exists a linear time algorithm, such that when given a graph $G = (V, E)$ with a tree-decomposition of treewidth at most k, the algorithm decides whether H is a minor of G, using standard methods for graphs with bounded treewidth (see, e.g., [1].) As such tree-decompositions can be found in linear time [4], when they exist, the following result holds:

THEOREM 4. *Let $\mathscr{G}$ be a class of graphs that is closed under taking of minors and that does not contain all planar graphs. Then there exists a linear time algorithm that tests whether a given graph G belongs to $\mathscr{G}$.*

Proof. This proof is basically taken from [13], but we now use the algorithm of [4] for finding tree-decomposition of small treewidth. Suppose G is a planar graph that does not belong to $\mathscr{G}$. First test whether the treewidth of input graph G is at most c_H. If not, we can safely conclude that $G \notin \mathscr{G}$. Otherwise, find a tree-decomposition of G of treewidth at most c_H with the algorithm of [4], and use this tree-decomposition to test whether a graph in ob($\mathscr{G}$) is a minor of G. ∎

THEOREM 5. *Let $\mathscr{G}$ be a class of graphs that is closed under taking of minors in the domain $\mathscr{H}$, $\mathscr{G} \subseteq \mathscr{H}$. Suppose there is at least one planar graph that belongs to $\mathscr{H}$ but not to $\mathscr{G}$. Then there exists a linear time algorithm that tests whether a given graph* G $\in \mathscr{H}$ *belongs to* G.

Proof. We again use an only slightly modified variant of a proof from [13].) Suppose H is a planar graph with $H \in \mathscr{H}$, $H \notin \mathscr{G}$. If $G \in \mathscr{G}$, then G does not contain H as a minor, hence has treewidth at most c_H. So, again we can first test whether the treewidth of G is at most k. If not, we are done. Otherwise, we compute a tree-decomposition of G with treewidth at most c_H and then use this tree-decomposition to test in linear time whether G contains a graph in $\text{ob}_{\mathscr{H}}(\mathscr{G})$ as a minor. ∎

The constant factor of the linear time algorithms mentioned above is exponential in the treewidth of the tree-decomposition used, i.e., in c_H, H a planar graph not in $\mathscr{G}$ (but in $\mathscr{H}$). The constant factor in the original result of Robertson and Seymour was "astronomically large." In a later paper, Robertson et al. [23] improved this result, and obtained a constant factor of $20^{2(2|V(H)|+4|E(H)|)^5}$.

Still, in most, if not all, practical cases, this constant factor is much too large and makes the algorithm practically infeasible. This is why we looked for much smaller values of c_H for graphs of the form $K_{2,r}$, as these graphs are planar, are connected, and can be shown to be "outside" the considered classes of graphs.

2.2. *Definitions and Preliminary Results on Interval Routing*

Unless stated otherwise, intervals will be assumed to be "cyclic" in the set $\{0, 1, ..., n-1\}$, $(n = |V|)$; thus if $a > b$ then the interval $[a, b)$ denotes the set $\{a, a+1, ..., n-1, 0, ..., b-1\}$.

The shortest distance from vertex $u \in V$ to a vertex $v \in V$ in a graph $G = (V, E)$ when edges have costs given by the edge cost function $c: E \to \mathbf{R}$ is denoted by $d_{G,c}(u, v)$. When G and/or c are clear from the context, we drop them from the subscript. The cost of a path p under edge cost function c is denoted by $c(p)$.

A node labelling of a graph $G = (V, E)$ is a bijective mapping $nb: V \to \{0, 1, ..., n-1\}$. An interval labelling scheme (ILS) of a graph $G = (V, E)$ is a node labelling nb of G, together with a labelling l mapping each link to an interval

$[a, b)$, $a, b \in \{0, 1, ..., n-1\}$, such that for every vertex v, the set of all labels of links outgoing from v partitions the set $\{0, 1, ..., n-1\}$.

Given an ILS, routing is done as follows. Each message contains, amongst others, the node label $nb(w)$ of its destination node w. When a node x receives a message with destination-label *dest*, it first looks whether $nb(x) = dest$. If so, the message has reached its destination, and is not routed any further. Otherwise, the message is transferred over the link with label $[a, b)$ such that $dest \in [a, b)$. An ILS is *valid* if for all nodes v, w, messages sent from v to w eventually reach w by this procedure. An *interval routing scheme* (IRS) is a valid ILS.

The notion of strict interval labelling schemes is obtained in a similar way: modify the definition of ILS in the sense that all labels of links associated with nodes v must partition the set $\{0, 1, ..., n-1\} - \{nb(v)\}$, i.e., the label of v may not appear in the labels of any of its outgoing links. A *linear interval labelling scheme* is an ILS where no interval label "wraps" around; i.e., for all interval labels $[a, b)$ $a < b$. Strict linear interval labelling schemes, strict interval routing schemes (SIRS), linear interval routing schemes (LIRS), and strict linear interval routing schemes (SLIRS) are defined in the obvious way.

For each of these notions, we also define a k-label variant. Here, each link is labelled with at most k (cyclic) intervals. All (cyclic) intervals associated with links of a node v must together partition $\{0, 1, ..., n-1\}$ (or $\{0, 1, ..., n-1\} - \{nb(v)\}$, in the case of strict labellings.) Again, a message is transferred over the link e for which one of its labels is an interval that contains the destination-number. k-label interval routing schemes, k-label linear interval routing schemes, etc., are defined as can be expected, and abbreviated as k-IRS, k-LIRS, etc. Note that an IRS is a 1-IRS, etc.

A routing scheme is *optimal* for a graph $G = (V, E)$, together with an assignment of a non-negative costs to each edge $e \in E$, if, whenever a message is sent from node v to node w, the path taken by this message is a minimum cost path from v to w.

Costs of edges denote the time needed to send a message over the edge. However, in many practical cases, this time may vary. This situation is modelled by the dynamic cost links setting.

We say that graph $G = (V, E)$ with dynamic cost links has an optimum k-IRS if there exists a node labelling nb of G such that for all assignments of non-negative costs to edges of E, there exists an IRS (nb, l) that is optimal for this cost assignment.

The class of graphs $k - \mathscr{IRS}$ is defined as the set of all graphs G that have an optimum k-IRS with dynamic cost links. In the same way, we define classes $k - \mathscr{LIRS}$, $k - \mathscr{SIRS}$, $k - \mathscr{SLIRS}$. See [19] for an overview of several results on these classes. We have the following relationships.

THEOREM 6. (i) $k - \mathscr{IRS} \subset (k+1) - \mathscr{IRS}$.

(ii) (Frederickson and Janardan [15]) $k - \mathscr{SIRS} \subset (k+1) - \mathscr{SIRS}$.

(iii) (Bakker *et al.* [2]) $k - \mathscr{LIRS} \subset (k+1) - \mathscr{LIRS}$.

(iv) (Bakker *et al.* [2]) $k - \mathscr{IRS} \subset (k+1) - \mathscr{LIRS}$.

(v) $k - \mathscr{SIRS} \subseteq k - \mathscr{IRS} \subseteq (k+1) - \mathscr{SIRS}$.

(vi) $k - \mathscr{SLIRS} \subseteq k - \mathscr{LIRS} \subseteq k - \mathscr{IRS}$.

The proof of (i) in the above theorem is very similar to that of (ii) in [15]. (v) and (vi) are easy.

3. CLOSEDNESS UNDER MINOR TAKING

In this section we prove that for each fixed integer $k \geqslant 1$, each of the classes $k - \mathcal{IRS}$, $k - \mathcal{LIRS}$, $k - \mathcal{SIRS}$, and $k - \mathcal{SLIRS}$ is closed under taking of minors in the domain of connected graphs. The reason this result is interesting is that it enables us to apply results from the theory of graph minors and of graphs of bounded treewidth to the theory of interval routing. We first prove a lemma which will be used later.

LEMMA 7. *Let $G = (V, E)$ be a graph with edge costs $c: E \rightarrow \mathbf{R}^+ \cup \{0\}$. There exists an edge cost function $c': E \rightarrow \mathbf{Z}^+$ such that for all $u, v \in V$: each shortest path p from u to v in G under edge costs c' is also a shortest path from u to v in G under edge costs c.*

Proof. Let $\mathcal{P}$ be the set of all simple paths in G. Define

$$\varepsilon = \min\{|c(p) - c(p')| \mid p, p' \in \mathcal{P}, c(p) \neq c(p')\}.$$

Note that $\varepsilon > 0$. Define $c': E \rightarrow \mathbf{Z}^+$ by taking for all $e \in E$

$$c'(e) = \left\lfloor \frac{|V| c(e)}{\varepsilon} \right\rfloor + 1.$$

Suppose p is a shortest path from u to v under edge costs c', but not under edge costs c. Let p' be another path from u to v with $c(p') < c(p)$. By definition of ε, we have $c(p') \leqslant c(p) - \varepsilon$. Let $n = |V|$. Now

$$\begin{aligned} c'(p') &= \sum_{e \in p'} \left(\left\lfloor \frac{n \cdot c(e)}{\varepsilon} \right\rfloor + 1 \right) \\ &\leqslant n - 1 + \frac{n}{\varepsilon} c(p') < n + \frac{n}{\varepsilon} (c(p) - \varepsilon) \\ &= \frac{n}{\varepsilon} c(p) = \sum_{e \in p} \frac{n \cdot c(e)}{\varepsilon} \\ &\leqslant \sum_{e \in p} \left(\left\lfloor \frac{n \cdot c(e)}{\varepsilon} \right\rfloor + 1 \right) = c'(p). \end{aligned}$$

So, $c'(p') < c'(p)$, hence p was not a shortest path from u to v under edge cost c', contradiction. ∎

THEOREM 8. *Let $k \in \mathbf{N}$ be a fixed constant.*

(i) *$k - \mathcal{IRS}$ is closed under minor taking in the domain of connected graphs.*

(ii) *$k - \mathcal{LIRS}$ is closed under minor taking in the domain of connected graphs.*

(iii) $k-\mathcal{SIRS}$ *is closed under minor taking in the domain of connected graphs.*

(iv) $k-\mathcal{SLIRS}$ *is closed under minor taking in the domain of connected graphs.*

Proof. (i) It is sufficient to prove, that if a connected graph $G=(V', E')$ is obtained from a graph $H=(V, E) \in k-\mathcal{IRS}$ by one of the following operations: removal of a vertex, removal of an edge, contraction of an edge, then $G \in k-\mathcal{IRS}$. Suppose $H \in k-\mathcal{IRS}$; let nb be a vertex labelling, such that for any cost assignment, there exists a k-label interval routing scheme (nb, l) for H.

First, suppose that G is obtained from H by removing an edge e_0. Use the same numbering nb for G. For any cost assignment c: $E_G \to \mathbf{R}^+ \cup \{0\}$, consider the cost assignment c': $E_H \to \mathbf{R}^+ \cup \{0\}$, where for all $e \in E_G$: $c'(e)=c(e)$, and take $c(e_0)=1+\sum_{e \in E_G} c(e)$; i.e., the cost of e_0 is chosen so large that no minimum cost path will ever use the edge e_0. Hence, any k-label interval routing scheme (nb, l) for H with costs c' will also be a k-label interval routing scheme for G with costs c.

Next, suppose that G is obtained from H by removing a vertex $v \in V$ and all of its adjacent edges. By first removing all edges adjacent to v but one, as in the previous case, it follows that we may assume v has degree 1. Now, no shortest path between two vertices w and x, $x \neq v$, $x \neq w$ uses v. Label the vertices in V' as follows: if $nb(w) < nb(v)$, then take $nb'(w)=nb(w)$, and if $nb(w) > nb(v)$, then $nb'(w)=nb(w)-1$. For any edge cost function c on G, we can make an IRS as follows: consider the same edge cost function c on H, giving the unique edge from v some arbitrary cost, and find an IRS (nb, l) for this function on H. Applying the same relabelling (decrease all labels larger than $nb(v)$ by one) on labelling l, we obtain a labelling l' such that (nb', l') is an IRS for G with edge costs c.

Finally, suppose G is obtained from H by contracting the edge $(v, w)=e_0 \in E_H$ to a vertex, say v'. Let nb': $V \to \{0, 1, ..., |V(H)|-1\}$ be the function, obtained by taking for all $x \in V(G)-\{v\}$, $nb'(x)=nb(x)$. Actually, there is a "gap" in nb': there is no vertex x with number $nb'(x)=nb(v)$. This is resolved by decreasing all labels larger than $nb(v)$ by one, as in the case of removing a vertex.

Let $c: E_G \to \mathbf{R}^+ \cup \{0\}$ be a cost assignment for G. By Lemma 7, there exists a cost assignment c': $E_G \to \mathbf{Z}^+$, such that all shortest paths under cost assignment c' are shortest paths under cost assignment c. Let $\alpha=1+\sum_{e \in E_G} c'(e)$, a forbidding weight. Now let c'': $E_H \to \mathbf{R}^+$ be defined as follows: for all edges $\{x, y\} \in E_H$ with $x, y \notin \{v, w\}$, let $c''(\{x, y\})=c'(\{x, y\})$. For $y \neq w$, if $\{v, y\} \in E(H)$, take $c''(\{v, y\})=c'(\{v', y\})$. For $y \neq v$, if $\{w, y\} \in E_H$, then if $\{v, y\} \in E_H$, then let $c''(\{w, y\})=\alpha$, otherwise let $c''(\{w, y\})=c'(\{v', y\})+1/4$. Finally, we let $c''(\{v, w\})=1/8$.

Let (nb, l) be a k-IRS for H with cost c''. We can use l to build a k-IRS (nb, l) for G with cost c'. First note that H without the edges of cost α is still connected. So, no shortest path takes an edge of cost α, and all links corresponding to these edges have an empty label. For every link $(x, \{x, y\})$ with $x \notin \{v, w\}$, take in l' the same labels as in l. For a link $(v', \{v', y\})$, take in l' the union of the labels of links $(v, \{v, y\})$ and $(w, \{w, y\})$. Note that one of these links is either non-existing or empty, so this label will not consist of more than k intervals. Also, note that for every node x, the shortest path from v to x does not use w, if and only if the shortest path from w to x uses v. The same holds with roles of v and w reversed. It follows that no vertex label will appear in more than one label of a link outgoing from v'. We now have shown that l' is a k-ILS.

It remains to be shown that l' gives shortest paths in G. Consider nodes x and y in $V(G)$. Let p be a shortest path in H between nodes x and y following links as directed by l'. If $x = v'$, then take $x = v$ in H. Similar, if $y = v'$. Note that if both v and w appear in p, then they must occur as consecutive nodes on this path, as all edges except $\{v, w\}$ have cost at least 1. Let p' be the path in G, obtained from p by replacing a possible occurrence of v, w or both by one occurrence of v'. Observe that l' will direct a message from x to y via path p'. Finally, observe that p' is a shortest path from x to y in G with costs c', hence also with costs c.

(ii) (iii) (iv) Similar. ▮

THEOREM 9. *$K_{2,2k+1} \notin k-\mathcal{IRS}$ and hence $K_{2,2k+1} \notin k-\mathcal{SIRS}$, $K_{2,2k+1} \notin k-\mathcal{LIRS}$, $K_{2,2k+1} \notin k-\mathcal{SLIRS}$.*

Proof. Frederickson and Janardan [15] prove that $K_{2,2k+1} \notin k-\mathcal{SIRS}$. Very similarly one can prove that $K_{2,2k+1} \notin k-\mathcal{IRS}$. ▮

It follows now from Theorem 2 that for each fixed $k \geqslant 1$, the classes $k-\mathcal{IRS}$, $k-\mathcal{LIRS}$, $k-\mathcal{SIRS}$, and $k-\mathcal{SLIRS}$ have finite characterisation in terms of obstruction sets. Combining Theorem 9, Theorem 8 and Theorem 5 gives the following result.

COROLLARY 10. *For each fixed $k \in \mathbf{N}$, there exists a linear time algorithm that decide whether given a graph $G = (V, E)$ belongs to the class $k-\mathcal{IRS}$ (or: $k-\mathcal{SIRS}$, $k-\mathcal{LIRS}$, $k-\mathcal{SLIRS}$).*

It should be noted that this result is *non-constructive*: we know the algorithm exists, but to write down the algorithm, we must know the corresponding finite obstruction set, which we do not know. Unfortunately, we only know of much slower constructive versions of these results. For establishing these constructive version, we first need the following lemma.

LEMMA 11. *Let $G = (V, E)$ be a connected graph, and let nb be a node labelling of G. The following statements are equivalent:*

1. *For every cost assignment $c: E \to \mathbf{R}^+ \cup \{0\}$, there exists an optimal k-SIRS.*
2. *For every vertex v, and for every edge $\{v, w\} \in E$, there does not exist vertices $a_1, ..., a_{k+1}, b_1, ..., b_{k+1} \in V$, and a spanning tree $T = (V, F)$ of G such that*

- *$nb(a_1) < nb(b_1) < nb(a_2) < nb(b_2) < \cdots < nb(a_k) < nb(b_k) < nb(a_{k+1}) < nb(b_{k+1})$ or $nb(b_1) < nb(a_1) < nb(b_2) < nb(a_2) < \cdots < nb(b_k) < nb(a_k) < nb(b_{k+1}) < nb(a_{k+1})$.*
- *For each i, $1 \leqslant i \leqslant k+1$, the path in T from v to a_i uses the edge $\{v, w\}$.*
- *For each i, $1 \leqslant i \leqslant k+1$, the path in T from v to b_i does not use the edge $\{v, w\}$.*
- *$nb(w) \in \{a_1, ..., a_{k+1}\}$.*

Proof. $2 \rightarrow 1$. Suppose that v, $\{v, w\}$, $a_1, ..., a_{k+1}, b_1, ..., b_{k+1}$ and T are as stated. Now, let c be the cost assignment that assigns cost 1 to every edge in T, and $|V|+1$ to every other edge, i.e., all shortest paths follow T. Now, each $nb(a_i)$ must be in a different interval for the link $(v, \{v, w\})$, as when $nb(a_i)$ and $nb(a_{i+1})$ would be in the same interval, then $nb(b_i)$ or $nb(b_{i+1})$ also would belong to the interval, and messages to this node b_i or b_{i+1} would be routed in the wrong direction.

$1 \rightarrow 2$. Suppose for cost assignment c, there is no optimal k-IRS. Note that we may assume that between every two pairs of nodes, there is a unique shortest path. (If not, then we can change the weights of some edges with very small amounts, such that there some non-unique shortest paths disappear, but no new shortest path routes are created.) Now, there are a vertex $v \in V$ and an adjacent edge $\{v, w\} \in E$ such that at least $k+1$ intervals, say $[c_1, d_1], ..., [c_r, d_r]$, $r \geqslant k+1$, are necessary to give the set of numbers of nodes whose shortest paths from v use the edge $\{v, w\}$. For each interval $[c_i, d_i]$, $1 \leqslant i \leqslant k$, choose a vertex a_i with $nb(a_i) \in [c_i, d_i]$, and choose a vertex a_{k+1} with $nb(a_{k+1}) \in [c_{k+1}, d_{k+1}] \cup \cdots [c_r, d_r]$, such that $w \in \{a_1, ..., a_{k+1}\}$. Next, choose $b_1, ..., b_{k+1}$, such that no $nb(b_i)$ belongs to an interval $[c_j, d_j]$ $(1 \leqslant i \leqslant k+1$, and $1 \leqslant j \leqslant r)$, and that $nb(a_1) < nb(b_1) < nb(a_2) < nb(b_2) < \cdots < nb(a_k) < nb(b_k) < nb(a_{k+1}) < nb(b_{k+1})$ or $nb(b_1) < nb(a_1) < nb(b_2) < nb(a_2) < \cdots < nb(b_k) < nb(a_k) < nb(b_{k+1}) < nb(a_{k+1})$. (It is easy to see that this can be done: in general, pick vertices whose number is between d_j and c_{j+1}.)

Let T be the shortest paths tree containing shortest paths from v to all other vertices. Such a tree exist and is unique as we assumed that between every two pairs of nodes, there is a unique shortest path (see, e.g., [11, Chap. 25]). The paths in T from v to a vertex a_i, $1 \leqslant i \leqslant k+1$ must use the edge $\{v, w\}$, while the paths in T from v to a vertex b_i do not use this edge. ∎

Similar results can be shown for k-IRS, k-LIRS, and k-SLIRS: in the case of non-strict versions, additionally we require that $v \notin \{a_1, ..., a_{k+1}, b_1, ..., b_{k+1}\}$, and in case of linear versions, b_{k+1} is not used, and the condition on the numbers of vertices a_i, b_i becomes: $nb(a_1) < nb(b_1) < nb(a_2) < nb(b_2) < \cdots < nb(a_k) < nb(b_k) < nb(a_{k+1})$.

THEOREM 12. *For any fixed $k \geqslant 1$, one can construct algorithms that test whether for a given graph $G = (V, E)$ with a node labelling nb and for all costs assignments $c: E \rightarrow \mathbf{R}^+ \cup \{0\}$, there exists an optimal k-IRS (or: k-SIRS, k-LIRS, k-SLIRS)(nb, l) for G with costs c, in $O(n^{2k+3})(O(n^{2k+3}), O(n^{2k+2}), O(n^{2k+2}))$ time.*

Proof. We consider the algorithm for checking existence of an optimal k-SIRS. First, we use the algorithm from [4] to check in linear time whether the treewidth of G is at most $4k$, and if so, to build a tree-decomposition of G of treewidth at most $4k$. If the treewidth of G is more than $4k$, then by Corollary 22, $G \notin k - \mathcal{SIRS}$, so also for the node labelling nb, there exists a cost assignment which requires at least $k+1$ intervals for some link: we can output "no" and stop.

So, now suppose we have a tree-decomposition of G of treewidth at most $4k$. It is well known that $|E| \leqslant 4k|V|$. Now, for every vertex $v \in V$, and for every $(v, w) \in E$, and for all vertices $a_1, ..., a_{k+1}, b_1, ..., b_{k+1} \in V$, with $nb(a_1) < nb(b_1) < nb(a_2) < nb(b_2) < \cdots < nb(a_k) < nb(b_k) < nb(a_{k+1}) < nb(b_{k+1})$ or $nb(b_1) < nb(a_1) < nb(b_2) < nb(a_2) < \cdots < nb(b_k) < nb(a_k) < nb(b_{k+1}) < nb(a_{k+1})$ and $w \in \{a_1, ..., a_{k+1}\}$, we check whether there exists a spanning tree $T = (V, F)$ of G such that

- For each i, $1 \leqslant i \leqslant k+1$, the path in T from v to a_i uses the edge $\{v, w\}$.
- For each i, $1 \leqslant i \leqslant k+1$, the path in T from v to b_i does not use the edge $\{v, w\}$.

If one of these checks is true, we know by Lemma 11 that there is a cost assignment for which no k-SIRS (nb, l) exists; otherwise we know that for all cost assignments such a k-SIRS does exist.

Each check can be done in linear time, with the help of the tree-decomposition: notice, that for fixed $v, w, a_1, ..., a_{k+1}, b_1, ..., b_{k+1}$, the existence of T fulfilling the given properties can be formulated in *Monadic Second Order Logic*, and hence be decided (with an algorithm that can be constructed) in linear time for graphs of bounded treewidth (see [1, 10, 12]). As we must make in total less than $|E| \cdot n^{2k+1} \cdot k = O(n^{2k+2})$ checks, the time bound follows.

The algorithms for the cases of k-IRS, k-LIRS, and k-SLIRS are similar: because b_{k+1} is not used, the time bounds for k-LIRS and k-SLIRS are a linear factor smaller. ∎

Corollary 13. *One can construct an algorithm that tests whether for a given integer* $k \in \mathbf{N}$, *and graph* $G = (V, E)$, $G \in k - \mathcal{IRS}$ (*or*: $G \in k - \mathcal{SIRS}$, $G \in k - \mathcal{LIRS}$, $G \in k - \mathcal{SLIRS}$).

Proof. Use the algorithm of Theorem 12 for each permutation (numbering) of the vertices of G. ∎

4. THE TREEWIDTH OF GRAPHS WITH *k*-LABEL INTERVAL ROUTING SCHEMES

The main object of this section is to prove the following result.

Theorem 14. *Every graph* $G = (V, E)$ *contains* $K_{2,r}$ *as a minor or has treewidth at most* $2r - 2$.

A variant of these results with a sharper bound for the case that G is planar is discussed at the end of this section.

Given a graph $G = (V, E)$ and a set $S \subseteq V$, let $\partial S = \{v \in V - S \mid \exists u \in S, \{u, v\} \in E\}$ (i.e., the neighbours of vertices in S that do not belong to S).

Definition. A set $S \subseteq V$ is an *s-t-separator* in $G = (V, E)$ $(s, t \in V)$, if s and t belong to different connected components of $G[V - S]$. S is a *minimal s-t-separator*, if it does not contain another s-t-separator as a proper subgraph. S is a *minimal separator* if there exist vertices $s, t \in V$ for which S is a minimal s-t-separator.

Note that minimal separators can contain other minimal separators as proper subgraphs. We will use in fact a different property of minimal separators, as given in the following lemma, which is easy to prove.

LEMMA 15. *A non-empty set S is a minimal separator in G if and only if there are at least two connected components G_1, G_2 of $G[V-S]$ such that $S \subseteq \partial V(G_i)$, $i=1, 2$ (i.e. each vertex in S has a neighbour in both G_1 and G_2). We call two such components* separated components.

LEMMA 16. *If G contains a minimal separator S, with $|S| \geqslant r$, then $K_{2,r}$ is a minor of G.*

Proof. Let S be a minimal separator and consider two separated components G_A and G_B of $G[V-S]$. Remove any vertex from any other component and $|S|-r$ vertices from S. If we now contract all edges in G_A and G_B that are not incident with a vertex in S, we obtain $K_{2,r}$. ▮

DEFINITION. Let $G=(V, E)$ be a graph and $\mathscr{S}$ a collection of subsets of $V(G)$. Denote by $\mathrm{CL}(G, \mathscr{S})$ the graph obtained from G by making every set $S_i \in \mathscr{S}$ into a clique, i.e., $\mathrm{CL}(G, \mathscr{S}) = (V, E \cup \{\{v, w\} \mid v \neq w, \exists S_i \in \mathscr{S}: v, w \in S_i\})$.

DEFINITION. For a given $r \geqslant 1$, let $\mathscr{D}_r$ be the class of all graphs $G=(V_0 \cup V_1 \cup V_2 \cup V_3, E)$, such that

- V_0, V_1, V_2, V_3 are disjoint sets.
- $V_0 = \{v_0\}$. v_0 is adjacent to all vertices in V_1 and no vertices in $V_2 \cup V_3$.
- $|V_1| < r$. Every vertex in V_1 is adjacent to at least one vertex in V_2 and to no vertex in V_3.
- Every vertex in V_2 is adjacent to at least one vertex in V_1.
- Every vertex in V_3 is adjacent to less than r vertices in V_2, and is not adjacent to vertices in $V_0 \cup V_1 \cup V_3$.

Finally, if $R \in \mathscr{D}_r$, we define $\mathrm{CL}(R) = \mathrm{CL}(R, \{\partial\{v\}: v \in V_0(R) \cup V_3(R)\})$. Also, we define $\mathrm{CL}(\mathscr{D}_r) = \{\mathrm{CL}(R) : R \in \mathscr{D}\}$.

Let $V_3 = \{v_1^3, ..., v_m^3\}$.

LEMMA 17 (See [5]). *For any graph $G=(V, E)$, either $K_{1,r}$ is a minor of G or treewidth$(G) \leqslant r-1$.*

Proof. W.l.o.g., suppose that G is connected. Take an arbitrary depth first search tree T of G. For any vertex v, let Y_v be the set of ancestors of v in T that are adjacent to v or to a descendant of v, and let $X_v = \{v\} \cup Y_v$. One can show that if $|Y_v| \geqslant r$, then G contains $K_{1,r}$ as a minor (contract v with all its descendants, and then remove all vertices not in X_v.) For all v, if $|Y_v| \leqslant r-1$ then $(\{X_v \mid \in V\}, T)$ is a tree-decomposition of G of treewidth at most $r-1$. ▮

Lemma 18. [See, e.g., [8]). *Let* $(\{X_i, i \in I\}, T)$ *be a tree-decomposition of graph* $G = (V, E)$. *For any clique* K *of* G, *there exists an* $i \in I$ *with* $V(K) \subseteq X_i$.

Lemma 19. *For any graph* $G \in \mathscr{D}_r$, *either* $K_{2,r}$ *is a minor of* G *or* G *has a tree-decomposition of treewidth* $\leqslant 2r-2$ *which is also a tree-decomposition of* $\mathrm{CL}(G)$.

Proof. Let $G_{\text{clique}} = \mathrm{CL}(G)$, $V(K_{1,r}) = \{w_0, w_1, ..., w_r\}$, and $E(K_{1,r}) = \{w_0, w_1\}, ..., w_0, w_r\}\}$. From Lemma 17, either $K_{1,r}$ is a minor of $G_{\text{clique}}[V_2]$ or $\mathrm{treewidth}(G_{\text{clique}}[V_2]) \leqslant r-1$. We consider these cases separately.

Case 1. $K_{1,r}$ is a minor of $G_{\text{clique}}[V_2]$. Let $i = 0, ..., r$ and S_{w_i} be the set of vertices in $G_{\text{clique}}[V_2]$ that were identified to w_i when creating $K_{1,r}$ as a minor. Notice that any set S_{w_i} induces a connected subgraph in $G_{\text{clique}}[V_2]$. Denote by R the set of vertices in V_3 that are adjacent to vertices in S_{w_0}. Finally let w_i be a vertex in S_{w_i} that is adjacent to a vertex in S_{w_0}. (Note that these vertices w_i exist, by the construction of $K_{1,r}$ as a minor.) We observe that $G_{\text{clique}}[R \cup S_{w_0}]$ is connected.

Claim I. $G[R \cup S_{w_0}]$ is connected.

Suppose not. As $E(G[R \cup S_{w_0}]) \subseteq E(G_{\text{clique}}[R \cup S_{w_0}])$, we can add edges in $E(G_{\text{clique}}[R \cup S_{w_0}]) - E(G[R \cup S_{w_0}])$ to $G[R \cup S_{w_0}]$ until an edge, say $\{x_1, x_2\}$ makes the graph connected. As $\{x_1, x_2\}$ belongs to $E(G_{\text{clique}}[R \cup S_{w_0}])$, but not to $E(G[R \cup S_{w_0}])$, the edge is in one of the added cliques; i.e., there must be a vertex $x_3 \in V_3$ that is adjacent to both x_1 and x_2. Now we have a contradiction, as $x_3 \in R \subseteq R \cup S_{w_0}$.

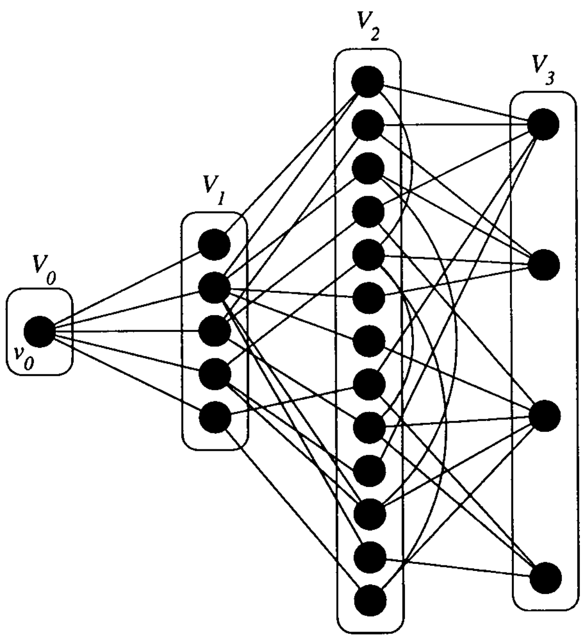

FIG. 1. Example of a graph in D_6.

Claim II. For all i, w_i is adjacent to a vertex in $R \cup S_{w_0}$ in G.

For all i, there exists a vertex $x_i \in S_{w_0}$ that is adjacent to w_i in G_{clique}. If $\{w_i, x_i\} \in E(G)$, then we are done. If $\{u_i, x_i\} \notin E(G)$, then there is a vertex $x_i^3 \in V_3$ with $\{x_i, x_i^3\}$, $\{w_i, x_i^3\} \in E(G)$, and the claim is true, as w_i is adjacent to $x_i^3 \in R$.

We can now show that $K_{2,r}$ is a minor of G. First contract all vertices in $R \cup S_{w_0}$ to a single vertex z_0. Next for each i, contract all vertices in S_{w_i} to a single vertex, say z_i. Then contract all vertices in $V_0 \cup V_1$ to a single vertex z_{r+1}. (We can make all of these contractions, as each of these sets induces a connected subgraph of G.) By claim II, z_0 is adjacent to each vertex in $\{z_1, ..., z_r\}$. Also for each i, as $S_{w_i} \subseteq V_2$ and each vertex in V_2 is adjacent to at least one vertex in V_1, z_{r+1} is adjacent to each vertex z_i, $1 \leqslant i \leqslant r$. We now have a $K_{2,r}$ minor.

Case 2. Treewidth$(G_{\text{clique}}[V_2]) \leqslant r-1$: We now show that treewidth$(G_{\text{clique}}) \leqslant 2r-2$. Take a tree-decomposition $(\{X_i \mid i \in I\}, T=(I, F))$ of $G_{\text{clique}}[V_2]$ with treewidth $\leqslant r-1$. Observe that $(\{X_i \cup V_1 \mid i \in I\}, T)$ is a tree-decomposition of $G_{\text{clique}}[V_1 \cup V_2]$ with treewidth at most $r-1+|V_1| \leqslant 2r-2$. Using this tree-decomposition, we can build a tree-decomposition of G_{clique} of treewidth $\leqslant 2r-2$, as follows. Add nodes $j_0, j_1^3, ..., j_m^3$ to I, with $X_{j_0} = \{v_0\} \cup \partial\{v_0\}$ and $X_{j_i^3} = \{v_i^3\} \cup \partial\{v_i^3\}$. By Lemma 18 there exists for each i a node $j'_i \in I$, with $\partial\{v_i^3\} \subseteq X_{j'_i}$. We make j_i^3 adjacent to this node j'_i. Finally, make j_0 adjacent to an arbitrary node $j'_0 \in I$. We now have a tree-decomposition of G of treewidth at most $2r-2$. ∎

Definition. A *terminal graph* is a triple $G=(V, E, S)$ where (V, E) is a graph and $S \subseteq V$ is an ordered subset of its vertices. We call S the *terminal set* of G.

Definition. Consider two terminal graphs $G_i = (V_i, E_i, S_i)$, $i=1, 2$ such that $|S_1| = |S_2|$. Define $G_1 \oplus G_2$ as the graph obtained by taking the disjoint union of G_1 and G_2 and then identifying the corresponding terminal vertices in S_1 and S_2.

Lemma 20. *Consider two terminal graphs* $G_i = (V_i, E_i, S_i)$, $i=1, 2$ *such that* $|S_1| = |S_2|$. *Suppose that for* $i=1, 2$, $G_i[S_i]$ *is a clique. If treewidth*$(G_i) \leqslant k_i$, $i=1, 2$, *then there is a tree-decomposition with* $G_1 \oplus G_2$ *of treewidth at most* $\max\{k_1, k_2\}$.

Proof. Take tree-decompositions $(\{X_j^i \mid j \in I^i\}, T^i = (I^i, F^i))$ of G^i of treewidth at most k_i, $i=1, 2$. By Lemma 18, there are $j_0^i \in I^i$ with $S_i \subseteq X_{j_0^i}^i$, $i=1, 2$. Taking the disjoint union of the two tree-decompositions and connecting nodes j_0^1 and j_0^2 yields the desired tree-decomposition: one easily verifies that $(\{X_j^1 \mid j \in I^1\} \cup \{X_j^2 \mid j \in I_2\}$, $T = (I^1 \cup I^2, F^1 \cup F^2 \cup \{\{j_0^1, j_0^2\}\}))$ is a tree-decomposition with $G_1 \oplus G_2$ with treewidth at most $\max\{k_1, k_2\}$. ∎

Definition. Let G be a graph and $\mathscr{S}$ a collection of subsets of $V(G)$. Denote by $\text{EX}(G, \mathscr{S})$ the graph obtained from G by adding to every set $S_i \in \mathscr{S}$ a new vertex

$v_{\mathrm{new},\,i}$ which is adjacent to all vertices in S_i. (In case $|\mathscr{S}|=1$, we denote the "new" vertex as v_{new}.)

We are now ready to prove Theorem 14. In fact, we prove the following, slightly stronger result.

THEOREM 21. *Let $G=(V,E)$ be a graph that is not a clique. Then, for any $r \geqslant 1$, either $K_{2,r}$ is a minor of G or for any minimal separator S where $|S|<r$, G has a tree-decomposition with treewidth $\leqslant 2r-2$ that is also a tree-decomposition of* $\mathrm{CL}(G, \{S\})$.

Proof. We use induction on $|V|$. The theorem clearly holds for $|V|=3$. Assume that the theorem holds for any graph with less than n vertices. Let $G=(V,E)$ be a graph with n vertices and let S be a minimal separator with $|S|<r$ (in the case where $|S| \geqslant r$, we have by Lemma 16 that $K_{2,r}$ is a minor of G). Let $G_i=(V_i,E_i)$, $i=1,\ldots,m$, be the connected components of $G[V-S]$ and $\bar{G}_i=\mathrm{EX}(G[V_i \cup \partial V_i], \{\partial V_i\})$. We denote the corresponding "new" nodes as v^i_{new}. Notice that each graph $\bar{G}_i$ has $\partial\{v^i_{\mathrm{new}}\}$ as a minimal separator and $|\partial\{v^i_{\mathrm{new}}\}|<r$. We consider two cases.

Case 1. $|V(\bar{G}_i)|<n$ for all $i \leqslant i \leqslant m$. From the induction hypothesis it follows that either $K_{2,r}$ is a minor of $\bar{G}_i$ for some i or for all i, $\bar{G}_i$ has a tree-decomposition of treewidth $\leqslant 2r-2$ that is also a tree-decomposition of $\mathrm{CL}(\bar{G}_i, \{\partial\{v^i_{\mathrm{new}}\}\})$. In the first case, as $\bar{G}_i$ is a minor of G, $K_{2,r}$ is also a minor of G.

So suppose that for all i, $\bar{G}_i$ has a tree-decomposition of treewidth $\leqslant 2r-2$ which is also a tree-decomposition of $\mathrm{CL}(\bar{G}_i, \{\partial\{v^i_{\mathrm{new}}\}\})$. We now construct a tree-decomposition of $\mathrm{CL}(G, \{S\})$ of treewidth $\leqslant 2r-2$. Let H_i be the graph obtained from $\mathrm{CL}(\bar{G}_i, \{\partial\{v^i_{\mathrm{new}}\}\})$ by removing the "new" vertex v^i_{new}. Clearly any graph H_i, $i=1,\ldots,m$ has a tree-decomposition of treewidth $\leqslant 2r-2$ and is a subgraph of $\mathrm{CL}(G, \{S\})$. Consider now the graphs H_1 and H_2. We have that $S_{1,2}=V(H_1) \cap V(H_2) \subseteq S$ and thus $S_{1,2}$ induces a clique in H_1 and H_2. Make H_1 and H_2 into terminal graphs with terminal set $S_{1,2}$. From Lemma 20 the graph $H_{1,2}=H_1 \oplus H_2$ has also a tree-decomposition of treewidth $\leqslant 2r-2$. Notice that $H_{1,2}$ is a subgraph of $\mathrm{CL}(G, \{S\})$ and $S_{1,2,3}=V(H_{1,2}) \cap V(H_3) \subseteq S$, thus $S_{1,2,3}$ induces a clique in $H_{1,2}$ and H_3. Now make $H_{1,2}$ and H_3 terminals with terminal set $S_{1,2,3}$ and apply again Lemma 20 to obtain a tree-decomposition of $H_{1,2,3}=H_{1,2} \oplus H_3$ with treewidth $\leqslant 2r-2$. In this manner, by repeatedly applying Lemma 20, we can merge all the tree decompositions of the graphs H_i, $i=1,\ldots,m$ and thus construct a tree-decomposition of $\mathrm{CL}(G, \{S\})$ that has treewidth $\leqslant 2r-2$.

Case 2. We now examine the remaining case, where there are only two connected components in $G[V-S]$ and at least one of them contains only one vertex v_0. Consider the set $D=\partial S-\{v_0\}$, and assume that it does not contain a minimal separator of cardinality $\geqslant r$ (if it does, then by Lemma 16, $K_{2,r}$ is a minor of G). Let G_i be the connected components of $G[V-D-S-\{v_0\}]$ and $N_i=\partial V(G_j) \subseteq D$, $i=1,\ldots,m$. Notice that, as D does not contain minimal separators of cardinality $\geqslant r$, $|N_i|<r$. Let $\bar{G}_i=\mathrm{EX}(G_i, \{N_i\})$, $i=1,\ldots,m$, and denote the corresponding

"new" vertices as v^i_{new}. As each $\bar{G}_i$ has less than n vertices, by the induction hypothesis, either $K_{2,r}$ is a minor of $\bar{G}_i$ for some i or $\bar{G}_i$ has a tree-decomposition of treewidth $\leqslant 2r-2$ which is also tree-decomposition of $\mathrm{CL}(\bar{G}_i, \{N_i\})$ for each i. In the first case, as before, $\bar{G}_i$ is a minor of G and thus $K_{2,r}$ is a minor of G. In the second case, observe that $F = \mathrm{EX}(G[D \cup S], \{S, N_1, ..., N_m\})$ is a member of D_r. From Lemma 19, either $K_{2,r}$ is a minor of F (which implies that $K_{2,r}$ is a minor of G, as F is a minor of G) or F has a tree-decomposition of treewidth $\leqslant 2r-2$ which is also a tree-decomposition of $\mathrm{CL}(F, \{S, N_1, ..., N_m\})$. We now construct a tree-decomposition of $\mathrm{CL}(G, \{S, N_1, ..., N_m\})$ with treewidth $\leqslant 2r-2$. For each i let H_i be the graph obtained from $\mathrm{CL}(\bar{G}_i, \{N_i\})$ if we eliminate the "new" vertex v^i_{new}. Also let F_0 be the graph obtained from F by eliminating the "new" vertices $v_{\text{new},i}$ corresponding to the sets $N_1, ..., N_m$. Clearly each H_i and F_0 have tree decompositions of treewidth $\leqslant 2r-2$. We observe that for F_0 and H_1, $V(F_0) \cap V(H_i) = N_1$ induces a clique in F_0 and H_1. Make F_0 and H_1 into terminal graphs with terminal set N_1. By Lemma 20 the graph $F_1 = F_0 \oplus H_1$ has also a tree-decomposition of treewidth $\leqslant 2r-2$. Now N_2 induces a clique on F_1 and H_2, so we can make them terminals with terminal set N_2, and apply Lemma 20 again to obtain a tree decomposition of $F_2 = F_1 \oplus H_2$ of treewidth $\leqslant 2r-2$. Continuing in this fashion we can merge all the tree decompositions of the graphs H_i, $i = 1, ..., r$ to the tree-decomposition of F_0, and thus construct a tree-decomposition of $\mathrm{CL}(G, \{S, N_1, ..., N_m\})$ of treewidth $\leqslant 2r-2$. As $\mathrm{CL}(G, \{S\})$ is a subgraph of $\mathrm{CL}(G, \{S, N_1, ..., N_m\})$, this completes the proof of the theorem. ∎

COROLLARY 22. *Every graph in* $k-\mathcal{IRS}$ *(and hence in* $k-\mathcal{SIRS}$, $k-\mathcal{LIRS}$, *and* $k-\mathcal{SLIRS}$*) has treewidth at most* $4k$.

Proof. If $G \in k-\mathcal{IRS}$, then $K_{2,2k+1}$ is not a minor of G; hence G has treewidth at most $2(2k+1)-2 = 4k$. ∎

Our results can be seen as partial characterisations of graphs which allow k-label interval routing schemes (with dynamic edge costs). The result also indicates a limitation of the interval routing method: as "most graphs have large treewidth" (see e.g., [17], Chap. 5), the set of graphs in $k-\mathcal{IRS}$ only covers a small part of all graphs (or even of all sparse graphs, see [17]).

Interestingly, the proof of Theorem 14 can be made constructive, and can be used to build an algorithm, that either outputs that input graph G has $K_{2,r}$ as a minor, or that outputs a tree-decomposition of G of treewidth at most $2r-2$, and that uses $O(rn)$ time. Combined with the results of Lemma 12 this can lead to a practical algorithm that checks whether for a given node labelling, an k-IRS (or k-SIRS, k-LIRS, k-SLIRS) exists for this labelling and all possible cost assignments, especially when additional optimisations are used, and k is small (e.g., $k=2$, or $k=3$.)

In some cases, more precise bounds are known. As $1-\mathcal{SIRS}$ equals the class of connected outerplanar graphs [15], and outerplanar graphs have treewidth at most 2, every graph in $1-\mathcal{SIRS}$ has treewidth at most 2. Similarly, the characterisation of $1-\mathcal{LIRS}$ in [2] shows that every graph in $1-\mathcal{LIRS}$ has treewidth (and even pathwidth) at most 2.

The results also have consequences for *random graphs*. We mention some results, obtained by Kloks [17]. Let $G_{n,m}$ denote a random graph with n vertices and m edges. For a precise meaning of the term "almost every" we refer to [17] or [9].

THEOREM 23 [Kloks [17]). (i) *Let* $\delta > 1.18$. *Then almost every graph* $G_{n,m}$ *with* $m \geqslant \delta n$ *has treewidth* $\Theta(n)$.

(ii) *For all* $\delta > 1$ *and* $0 < \varepsilon < (\delta - 1)/(\delta + 1)$, *almost every graph* $G_{n,m}$ *with* $m \geqslant \delta n$ *has treewidth at least* n^{ε}.

COROLLARY 24. (i) *Let* $\delta > 1.18$. *Then for almost every graph* $G_{n,m}$ *with* $m \geqslant \delta n$, *the smallest* k *for which* $G_{n,m} \in k - \mathscr{I}\mathscr{R}\mathscr{S}$ *is of size* $\Theta(n)$.

(ii) *Let* $\delta > 1$ *and* $0 < \varepsilon < (\delta - 1)/(\delta + 1)$. *Then for almost every graph* $G_{n,m}$ *with* $m \geqslant \delta n$, *the smallest* k *for which* $G_{n,m} \in k - \mathscr{I}\mathscr{R}\mathscr{S}$ *fulfills* $k \geqslant n^{\varepsilon}/4$.

We end this section by mentioning some results, similar to those shown above, for the case in which the graph G is planar.

THEOREM 25. *If* G *is planar*, *then either* G *contains* $K_{2,r}$ *as a minor or the treewidth of* G *is at most* $r + 2$.

For the lengthy proof, see [7].

COROLLARY 26. *Every planar graph in* $k - \mathscr{I}\mathscr{R}\mathscr{S}$ (*and hence in* $k - \mathscr{S}\mathscr{I}\mathscr{R}\mathscr{S}$, $k - \mathscr{L}\mathscr{I}\mathscr{R}\mathscr{S}$, *and* $k - \mathscr{S}\mathscr{L}\mathscr{I}\mathscr{R}\mathscr{S}$) *has treewidth at most* $2k + 3$.

5. CONCLUSIONS

In this paper, we made a perhaps somewhat surprising and interesting connection between the theory of compact routing schemes and the theory of graph minors and treewidth of graphs. Several angles of this connection are still left unexplored.

As main open problems, we like to mention several issues that deal with constructivity. Is it possible to *construct* linear time algorithms that test whether a given graph belongs to k-$\mathscr{I}\mathscr{R}\mathscr{S}$ or one of its variants, for a fixed k? In several other cases, a non-constructive proof of a linear or small degree polynomial time bound was only the first step towards a fully constructive solution (e.g., [4]). Will our Corollary 10 also be such a first step? But even if we know that a graph belongs to k-$\mathscr{I}\mathscr{R}\mathscr{S}$ (or a related class), we do not have a corresponding node labelling. How much time does it cost to construct such a node labelling? And, given a node labelling, how much time does it cost to verify that it has a k-label IRS (or variant) for every edge cost assignment? More related open problems are mentioned, e.g., in [19].

Received October 2, 1996; final manuscript received July 17, 1997

REFERENCES

1. Arnborg, S., Lagergren, J., and Seese, D. (1991), Easy problems for tree-decomposable graphs, *J. Algorithms* **12**, 308–340.
2. Bakker, E. M., van Leeuwen, J. and Tan, R. B. (1991), Linear interval routing schemes, *Algorithms Rev.* **2**, 45–61.
3. Bienstock, D., Robertson, N., Seymour, P. D., and Thomas, R. (1991), Quickly excluding a forest, *J. Combin. Theory Ser. B* **52**, 274–283.
4. Bodlaender, H. L. (1996), A linear time algorithm for finding tree-decompositions of small treewidth, *SIAM J. Comput.* **25**, 1305–1317.
5. Bodlaender, H. L. ((1993), On linear time minor tests with depth first search, *J. Algorithms* **14**, 1–23.
6. Bodlaender, H.L. (1994), On disjoint cycles, *Int. J. Found. Comput. Sci.* **5** (1), 59–68.
7. Bodlaender, Hans L., van Leeuwen, Jan, Tan, Richard, and Thilikos, Dimitrios (1996), (1996), "On Interval Routing Schemes and Treewidth," Technical Report UU-CS-1996-41, Department of Computer Science, Utrecht University, Utrecht. Available ftp://ftp.cs.ruu.nl/pub/RUU/CS/techreps/CS-1996/1996-41.ps.gz
8. Bodlaender, H. L., and Möhring, R. H. (1993), The pathwidth and treewidth of cographs, *SIAM J. Discrete Method* **6**, 181–188.
9. Bollobas, B. (1985), "Random Graphs," Academic Press, London,.
10. Borie, R. B., Parker, R. G., and Tovey, C. A. (1992), Automatic generation of linear-time algorithms from predicate calculus descriptions of problems on recursively constructed graph families, *Algorithmica* **7**, 555–581.
11. Cormen, T. H., Leiserson, C. C., and Rivest, R. L. (1989), "Introduction to Algorithms," MIT Press, Cambridge, MA.
12. Courcelle, B. (1990), The monadic second-order logic of graphs. I. Recognizable sets of finite graphs, *Inform. and Comput.* **85**, 12–75.
13. Fellows, M. R., and Langston, M. A. (1988), Nonconstructive tools for proving polynomial-time decidability, *J. Assoc. Comput. Mach.* **35** 727–739.
14. Fellows, M. R., and Langston, M. A. (1994), On search, decision and the efficiency of polynomial-time algorithms, *J. Comput. System. Sci.* **49**, 769–779.
15. Frederickson, G. N., and Janardan, R. (1988), Designing networks with compact routing tables, *Algorithmica* **3**, 171–190, 1988.
16. Inmos. (1991), "The T9000 Transputer Products Overview Manual."
17. Kloks, T. (1994), "Treewidth. Computations and Approximations," Lecture Notes in Computer Science," Vol. 842, Springer-Verlag, Berlin/New York.
18. van Leeuwen, J., and Tan, R. B. (1986), Computer networks with compact routing tables, *in* "The Book of *L*" (G. Rozenberg and A. Salomaa, Ed.), pp. 298–307, Springer-Verlag, Berlin.
19. van Leeuwen, J., and Tan, R. B. Compact routing methods: A survey, *in* "Proceedings, Colloquium on Structural Information and Communication Complexity (SIROCCO'94), School of Computer Science, Carleton University, Ottawa, 99–109.
20. Robertson, N., and Seymour, P. D. (1985), Graph minors—A survey, *in* "Surveys in Combinatorics" (I. Anderson, Ed.), pp. 153–171, Cambridge Univ. Press, Cambridge, UK.
21. Robertson, N., and Seymour, P. D. (1986), Graph minors. V. Excluding a planar graph, *J. Combin. Theory Ser. B* **41**, 92–114.
22. Robertson, N., and Seymour, P. D. (1995), Graph minors. XIII. The disjoint paths problem, *J. Combin. Theory Ser. B* **63**, 65–110.
23. Robertson, N., Seymour, P. D., and Thomas, R. (1994), Quickly excluding a planar graph, *J. Combin. Theory Ser. B* **62**, 323–348.
24. Santoro, N., and Khatib, R. (1985), Labelling and implicit routing in networks, *Comput. J.* **28**, 5–8.

Printed in Belgium