ELSEVIER
Theoretical Computer Science 290 (2003) 1503–1540
Theoretical Computer Science
www.elsevier.com/locate/tcs

Interval routing in some planar networks ☆

Victor Chepoi, Alexis Rollin

CNRS UMR 6166 Laboratoire d'Informatique Fondamentale de Marseille, Faculté des Sciences de Luminy, Université de la Méditerranée, 163, Avenue de Luminy, F-13288 Marseille Cedex 9, France

Received 31 July 2000; received in revised form 29 November 2001; accepted 21 December 2001
Communicated by G. Ausiello

Abstract

In this article, we design optimal or near optimal interval routing schemes (IRS, for short) with small compactness for several classes of plane quadrangulations and triangulations (by optimality or near optimality we mean that messages are routed via shortest or almost shortest paths). We show that the subgraphs of the rectilinear grid bounded by simple circuits allow optimal IRS with at most two circular intervals per edge (2-IRS). We extend this result to all plane quadrangulations in which all inner vertices have degrees $\geqslant 4$. Namely, we establish that every such graph has an optimal IRS with at most seven linear intervals per edge (7-LIRS). This leads to a 7-LIRS with the stretch factor 2 for all plane triangulations in which all inner vertices have degrees $\geqslant 6$. All routing schemes can be implemented in linear time. © 2002 Elsevier Science B.V. All rights reserved.

Keywords: Communication networks; Internal routing; Shortest path; Plane quadrangulation

1. Introduction

The routing of messages between pairs of nodes in a network of processors is a fundamental problem in distributed computation. A network can be viewed as a symmetric directed graph, with the vertices representing processors and the directed edges representing direct connections between processors. A *routing scheme* is a strategy that assigns to every source–destination pair the path that a message from the source to the destination should take. Since the cost of sending a message is roughly proportional to the number of edges the message has to traverse, it is desirable to route messages

☆ An extended abstract of this paper appeared in the Proceedings of the 8th International Colloquium on Structural Information and Communication Complexity—SIROCCO'2001.

E-mail address: chepoi@lim.univ-mrs.fr (V. Chepoi).

PII: S0304-3975(02)00067-1

along paths as short as possible. One possible approach is to store a complete routing table in each of the n vertices of the network, specifying for each destination the next edge in some shortest path over which the message must be forwarded. While this solution guarantees optimal (shortest path) routing, it requires a total of $O(n^2)$ node names. If the network is dense, then one would not expect to be able to do better than using complete routing tables, although in [1,22] it is shown that routing table space can be reduced at the expense of increasing the distance traversed by the message (see [14] for similar but better results in the particular case of planar networks).

A different way of implementing routing schemes, called *interval routing*, has been presented in [23], and later in [17,18]. In this method, each node is assigned a distinct label from the set $\{1,\ldots,n\}$. Arcs are bi-directional and are labelled with one or several *subintervals* of the (linear or circular) interval $[1\ldots n]$ so that for any node v the intervals associated with outgoing edges from v are pairwise disjoint and their union covers $[1\ldots n]$ (a precise definition is given in the next section). When a message with destination v arrives at node $u\neq v$, the message is forwarded on the unique outgoing edge labelled with an interval containing the label of v. In most cases, $[1\ldots n]$ is the cyclic interval, i.e., all subintervals are understood to wrap around. Such a scheme is called a *circular interval routing scheme* (IRS for short). Variants of the scheme include *linear interval routing schemes* (LIRS), in which $[1\ldots n]$ is viewed as a linear interval; *k-interval routing schemes*, in which edges can be labelled with at most k intervals (k-IRS or k-LIRS and their variants), *strict interval routing schemes* (k-SIRS or k-SLIRS) if any label of an outgoing edge from u cannot include the label for u. The efficiency of an interval routing scheme is measured in terms of its *stretch factor*—the maximum ratio between the length of the path traversed by a message and that of the shortest path between its source and destination, and its *compactness*—the maximum number of intervals constituting the label of an edge. An interval routing scheme for which all messages are routed along shortest paths is called an *optimal scheme*.

Many highly regular networks such as complete graphs, grids (alias meshes), hypercubes, complete bipartite graphs, unit interval graphs admit optimal 1-LIRS [2,11,17]. Other networks such as trees, rings, tori, unit circular-arc graphs, outerplanar graphs, and interval graphs have optimal 1-IRS or 1-SIRS [11,13,18,21,23]. Finally, in [20] it is shown that the 2-trees allow optimal 3-IRS. It is known that all graphs have nonoptimal 1-SIRS [17,23]: it is sufficient to route along paths of an arbitrary spanning tree. On the other hand, the problem of determining whether a given graph has a k-SIRS (or its variants) with stretch factor s is NP-complete for every integer $k\geqslant 1$ and for every $1\leqslant s<3/2$ (see [9,10]). If one considers the class of planar graphs, then [14] establishes that they allow optimal SIRS with compactness $\leqslant 3p/2$, where p is the smallest number of disjoint faces that cover all the nodes. On the negative side, [16] shows that for every integer n large enough there exists a planar graph with n vertices (a plane triangulation of bounded degree, in fact) of compactness $\Omega(\sqrt{n})$ (for other lower bounds concerning interval routing in planar graphs see [24]). It is conjectured in [15] that every n-vertex plane graph has compactness $O(\sqrt{n})$. For a complete list of results and concepts from the domain of interval routing see the recent survey [15].

In this note, we design optimal or near-optimal interval routing schemes with small compactness for several natural classes of planar networks. First we consider the case of rectilinear cells (subgraphs of the rectilinear two-dimensional grid bounded by simple circuits of this grid) and show that they allow optimal 2-SIRS. Next we show that every plane graph in which all inner faces are quadrangles and all inner vertices have degrees larger than 3 supports optimal 7-SLIRS. As a consequence of these results, we establish that two further classes of plane graphs admit compact routing schemes with a small stretch factor, in particular, that the plane graphs in which all inner faces are triangles and all inner vertices with degrees larger than 5 allow 7-SIRS with the stretch factor 2.

Notice that the quadrangulations G arising in this paper can be viewed as special subgraphs of most popular network topologies (hypercubes, meshes, tori). Even more, each such G can be represented as a subgraph of a respective host graph H such that the distances in G and in H between any pair of vertices of G coincide (i.e. G is a distance-preserving subgraph of H). Although finding interval routing schemes with small compactness for large and natural classes of graphs seems to be an interesting problem in its own rights, this remark shows that the routing schemes in our quadrangulations may be used for efficient routing in multi-user multi-processor systems. Many multi-processor systems (e.g., Intel*i*PSC860, Intel *Paragon*) may be configured as multi-user systems to better utilize the computational power. For this, processors are allocated to users so that no processor is simultaneously used by more than one user. As a result, the respective hypercube or mesh topology is divided into subhypercubes or submeshes specifying a restricted access to a portion of the network for particular users. In general, the set of processors allocated to a specific user may induce an arbitrary subgraph of the network. This implies that messages of a user may pass via processors allocated to another user. In order to avoid this phenomenon, one can force the subgraphs to be connected subhypercubes or submeshes [6,12]. Since many communication procedures are based on distance criteria, one can further force the respective subgraphs to be distance-preserving. There is a price to pay for this: even if a large number of algorithms for basic communication and routing problems have been developed for all common network architectures, it can be difficult (and different) to solve the corresponding problems when the subgraph allocated to a specific user is arbitrary, connected, or distance-preserving. From this perspective, our note contributes to the routing problem in a multi-user systems with a hypercube, torus or mesh topology in which the subgraph allocated to each user is a quadrangulation defined above. Notice that, motivated by this application, the broadcasting problem in submeshes and, in particular, in rectilinear cells has been considered in [12].

2. Preliminaries

2.1. Interval routing schemes

Let $G=(V,E)$ be a connected undirected graph with n vertices which represents a network. In order to model bidirectionality of the links in the network, we will treat an undirected edge uv between the vertices u and v as a pair (u,v) and (v,u) of complementary directed edges. Note that in all other respects the graph is considered to be undirected. The *distance* $d_G(u,v)$ between two vertices u and v of G is the length of a shortest path between u and v.

A (*sub*)*interval* $[a \dots b]$ of the discrete cyclic interval $[1 \dots n]$ is a contiguous subcollection of integers between a and b, where 1 follows n in cyclic order, i.e., $[a \dots b] = \{i: a \leqslant i \leqslant b\}$ if $a \leqslant b$ and $[a \dots b] = [1 \dots n] - [b+1 \dots a-1]$ otherwise.

A *k-interval routing scheme* (k-IRS for short) of G is a pair $\mathscr{R} = (\mathscr{L}, \mathscr{I})$ verifying the following conditions:

(i) $\mathscr{L}$ is a bijection between V and $\{1,\dots,n\}$, where $\mathscr{L}(v)$ is called the *label* of the vertex $v \in V$;

(ii) $\mathscr{I}: E \rightarrow 2^{\mathscr{L}(V)}$ assigns to each directed edge (u,v) an edge label $\mathscr{I}(u,v)$ which is a set containing k or fewer disjoint subintervals of the cyclic (or linear) interval $[1 \dots n]$, such that for each $v \in V$, the intervals associated with the outgoing edges form a partition of $[1 \dots n]$ (possibly excluding $\mathscr{L}(v)$);

(iii) for each distinct $u, v \in V$, there exists a path $R(u,v) = (u = x_0, x_1, \dots, x_{t-1}, x_t = v)$ such that $\mathscr{L}(v) \in \mathscr{I}(x_{i-1}, x_i)$ for every $i = 1,\dots,t$.

The path $R(u,v)$ in (iii) is called a *routing path* and its length is denoted by $d_R(u,v)$. If all routing paths are shortest paths, then $\mathscr{R}$ is called an *optimal* k-IRS, otherwise $\max\{d_R(u,v)/d_G(u,v):\ u,v \in V\}$ is the *stretch factor* of the routing scheme $\mathscr{R}$. For a subset of vertices $X \subseteq V$ set $\mathscr{L}(X) = \bigcup\{\mathscr{L}(x):\ x \in X\}$.

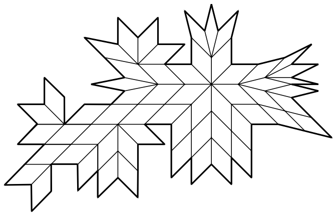
To give a simple but instructive example consider the classical routing scheme for a tree T. In this case, the label of a vertex v is its number in a depth first search (DFS) traversal of the vertices of T. Removing an arbitrary edge uv of T we obtain two subtrees T_u and T_v, where $u \in T_u$ and $v \in T_v$. The characteristic feature of DFS is that the labels of the vertices from T_u and T_v constitute two complementary subintervals of the cyclic interval $[1 \dots n]$. Therefore, if one set $\mathscr{I}(u,v) = \mathscr{L}(T_v)$ and $\mathscr{I}(v,u) = \mathscr{L}(T_u)$, we obtain an optimal 1-SIRS.

2.2. Plane graphs

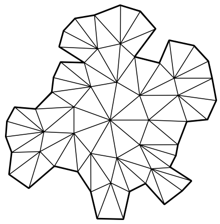
In this subsection we briefly introduce some classes of plane graphs and recall their basic properties which will be used in the sequel. All graphs occurring here are finite, without loops or multiple edges.

By a *plane quadrangulation* (resp., *triangulation*) we mean a plane graph in which all inner faces are quadrangles (resp., triangles). One would not expect that finding a routing scheme in such graphs is easier than in general plane graphs, however, imposing some constraints on degrees of inner vertices, one can separate (enough large) classes of triangulations and quadrangulations allowing compact routing schemes.

Denote by $\mathscr{Q}_4$ the class of plane quadrangulations in which all inner vertices have degree $\geqslant 4$. Let $\mathscr{T}_6$ denote the class of plane triangulations in which all inner vertices have degree $\geqslant 6$. $\mathscr{Q}_4$ and $\mathscr{T}_6$ occur in [3–5] as the 2-dimensional instances of more general classes of graphs (for example, $\mathscr{Q}_4$ is a subclass of median graphs). Examples of graphs from both classes are presented in Fig. 1. Using the specific properties of graphs from $\mathscr{Q}_4$, one can show that every such quadrangulation G has a plane drawing

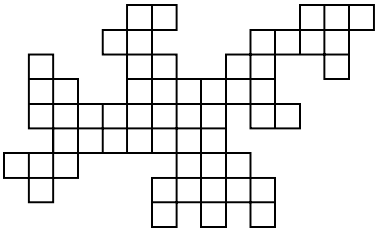


Quadrangulation from $\mathcal{Q}_4$

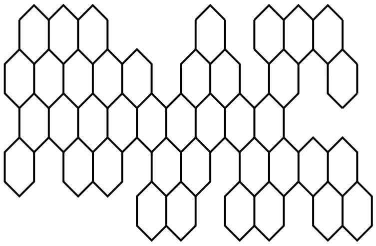


Triangulation from $\mathcal{T}_6$

Fig. 1.



Rectilinear Cell



Hexagonal system

Fig. 2.

in which every inner face is a parallelogram. In the sequel, we will draw all examples of $G \in \mathcal{Q}_4$ in this way.

A *region* R of a (not necessarily finite) plane graph is a plane subgraph induced by a finite connected set of inner faces. The (*topological*) *boundary* ∂R of R is the set of edges which occur in precisely one inner face of R. (In a similar way, one can define the *boundary* ∂G, alias the *outer face*, of a finite plane graph G.) A region R is *simply connected* if its complement $\mathbb{R}^2 \backslash R$ is connected and if its boundary edges can be ordered to form a single closed path.

An important particular case of quadrangulations in $\mathcal{Q}_4$ is constituted by the simply connected regions R of the square lattice $\mathbb{Z}^2$ (for an illustration see Fig. 2). We will call such a graph R a *rectilinear cell.* All the inner vertices of a rectilinear cell R have degree 4, while all vertices of ∂R have degree $\leqslant 4$. In a similar way, a *hexagonal* (*alias benzenoid*) *system* is a finite, simply connected region of the hexagonal lattice (a tiling of the plane into regular hexagons).

We continue with some additional notions and properties of plane graphs introduced above. An induced subgraph H of a graph G is called *gated* if for every vertex x outside H there exists a vertex x' (the *gate* of x) in H such that each vertex y of H is connected with x by a shortest path passing through the gate x'. An induced subgraph (or a subset of vertices) H is called *convex* if H includes every shortest path of G between two vertices u,v from H. Every gated subgraph is convex. The converse is not true in general. However, this is so for median graphs (and therefore, for quadrangulations from $\mathscr{Q}_4$); see, for example, [25].

Let $G=(V,E)$ be a plane graph. A *cut* $\{A,B\}$ of G is a partition of V into two sets A and B. Let $E(A,B)$ be the set of edges with one end in A and another one in B. Evidently, removing the edges of $E(A,B)$ from G we obtain a graph with at least two connected components, i.e. $E(A,B)$ is a cutset of edges. If both halves occurring in the cut $\{A,B\}$ are convex, we call such a cut *convex*. In this case the sets A and B are called *halfplanes*. If $G\in\mathscr{Q}_4$, the amount of convex cuts of G is rich enough to encode its metric. Moreover, every convex cut $\{A,B\}$ of G can be obtained by cutting G along a polygonal line. Namely, as G is bipartite, each of its edge ab induces a bipartition of V into $V=V(a,b)\cup V(b,a)$, where

$$V(a,b)=\{v\in V:\ d(v,a)<d(v,b)\},$$

$$V(b,a)=\{v\in V:\ d(v,b)<d(v,a)\}.$$

The sets $V(a,b)$ and $V(b,a)$ are convex for every median graph (actually this holds exactly for graphs isometrically embeddable into hypercubes; see [7] for details), in particular for $G\in\mathscr{Q}_4$. The convexity of all the sets $V(a,b),V(b,a)$ is equivalent to the fact that the following relation Θ is an equivalence relation [7,8]: define for any edges uv and xy of G,

$$uv\ \Theta\ xy \Leftrightarrow \text{either } x\in V(u,v) \text{ and } y\in V(v,u),$$

$$\text{or } y\in V(u,v) \text{ and } x\in V(v,u).$$

We may compare Θ to the following relation Ψ^*. First say that two edges uv and xy are in relation Ψ if they either are equal or constitute opposite edges on some inner face of G. Then let Ψ^* be the transitive closure of Ψ on the edge set E. From [3] we know that $\Theta=\Psi^*$. Let $E_1,E_2,\dots,E_m$ be the equivalence classes of Θ and let $\mathscr{L}=\{\{A_1,B_1\},\{A_2,B_2\},\dots,\{A_m,B_m\}\}$ be the collection of underlying convex cuts of G. The equality $\Theta=\Psi^*$ implies a straightforward linear time algorithm for listing the equivalence classes of G. Let Z_i be the family of inner faces of G which are crossed by $\{A_i,B_i\}$. We call Z_i the *zone* of the cut $\{A_i,B_i\}$ (having a look at figures from Section 4, one can equally view Z_i as a *train track*). Finally, let C_i be a polygonal line defined in the following way: the vertices of C_i are middles of the edges of E_i and two such vertices are adjacent if and only if they are hosting edges in relation Ψ. If we cut the plane along C_i, then once entering a face of Z_i we must exit this face through a parallel edge and we will never visit this face again. In particular, the line along which we cut has no self-intersections. Furthermore, from more general results from [3,5] it is known that C_i and the sets $bd(A_i)=A_i\cap Z_i$ and $bd(B_i)=B_i\cap Z_i$ are paths. We call $bd(A_i)$ and $bd(B_i)$ the *border lines* of the convex cut $\{A_i,B_i\}$ and C_i the *pseudoline* of this cut. Every two pseudolines C_i and C_j intersect in at most one point (resp., every two zones Z_i and Z_j share at most one common inner face of G). Finally, notice that there is at least one cut $\{A_i,B_i\}$ such that one of A_i,B_i is a path

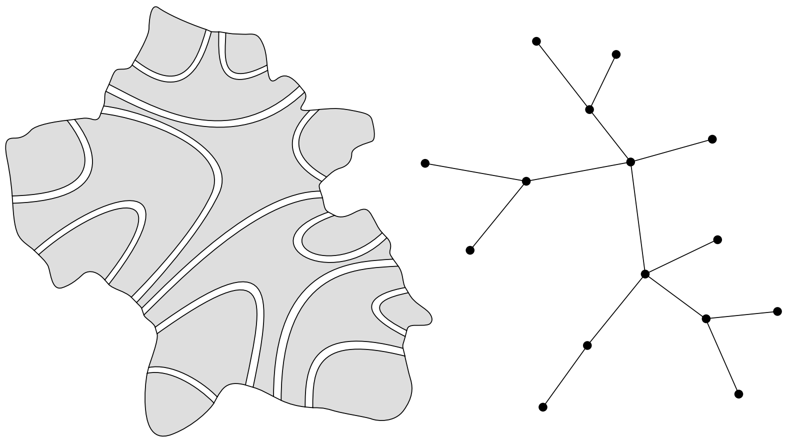


Fig. 3.

(using this property and induction, one can show that G has a plane drawing in which all inner faces are parallelograms).

We call two cuts $\{A_i, B_i\}$ and $\{A_j, B_j\}$ *transversal* if all four intersections $A_i \cap A_j, A_i \cap B_j, B_i \cap A_j, B_i \cap B_j$ are non-empty (i.e., if the pseudolines C_i and C_j cross each other), and *laminar* otherwise. A subfamily $\mathscr{C} \subseteq \mathscr{Z}$ of cuts is called *laminar* if every two cuts of $\mathscr{C}$ are laminar. The importance of laminar collections of cuts resides in the underlying tree structure defined by these cuts. Namely, let $\mathscr{C} = \{\{A_{i_1}, B_{i_1}\}, \ldots, \{A_{i_k}, B_{i_k}\}\}$ be a collection of laminar cuts, i.e., they pseudolines are pairwise disjoint. Cutting G along the disjoint pseudolines $C_{i_1}, \ldots, C_{i_k}$, we will obtain a partition of G into $k+1$ regions $R_0, R_1, \ldots, R_k$, each containing one connected component of the graph $(V, E \backslash (E_{i_1} \cup \ldots E_{i_k}))$ (see Fig. 3 for an illustration). To make our treatment more intuitive, further we identify the connected components of this graph with their hosting regions. Define a graph $T(\mathscr{C})$ having $R_0, R_1, \ldots, R_{k+1}$ as a vertex-set and two vertices R_s and R_t are adjacent if and only if there is a cut of $\mathscr{C}$ whose two border lines belong one to R_s and another to R_t. Since $\mathscr{C}$ is laminar, $T(\mathscr{C})$ is a tree.

This simple observation is crucial in designing the routing algorithm for G. First, we construct a collection $\mathscr{C}$ of k laminar cuts, such that the resulting regions $R_0, R_1, \ldots, R_k$ are rectilinear or pseudorectilinear cells (roughly speaking, a pseudorectilinear cell is a quadrangulation of $\mathscr{Q}_4$ in which all inner vertices have degree 4 except a certain amount of vertices of degree 5 located in a rather specific way). To assign contiguous intervals of labels to each of these regions, we perform a DFS traversal of the tree $T(\mathscr{C})$. To label the vertices inside each cell, we exploit the specific structure of rectilinear and pseudorectilinear cells. Routing inside rectilinear cells can be done using two circular or three linear intervals per edge (this is the result of Section 3), while routing inside pseudorectilinear cells needs at most five linear intervals per edge. Since every cell R_j is convex, the routing paths between two vertices of R_j are shortest paths in the global graph G. To route messages between vertices in different cells, we exploit the treelike structure of laminar cuts and the fact that the regions R_j are not only convex but also gated. This needs two extra intervals per edge, yielding an optimal 7-SLIRS.

2.3. Properties of quadrangulations from $\mathcal{Q}_4$

Here we give a list of properties of graphs $G\in\mathcal{Q}_4$. As we already noticed, the graphs from $\mathcal{Q}_4$ are median, and this implies that Θ is an equivalence relation which coincides with Ψ^*, and that the halves A_i, B_i and the border lines $bd(A_i), bd(B_i)$ of every convex cut are gated. For the proofs of these and other results about median graphs see [3,5,25].

Let $\mathscr{C}$ be a collection of k laminar convex cuts of G which partition the graph G into the regions $R_0, R_1, \ldots, R_k$. For two regions R_i and R_j let $R_{i_0} := R_i, R_{i_1}, \ldots, R_{i_s} := R_j$ be the path connecting the nodes R_i and R_j in the tree $T(\mathscr{C})$. Pick two arbitrary vertices $x\in R_i$ and $y\in R_j$. Denote by x_1 the gate of x in R_{i_1} and by y_{s-1} the gate of y in $R_{i_{s-1}}$. Suppose that the vertices $x_1\in R_{i_1}, \ldots, x_l\in R_{i_l}$ and $y_{s-1}\in R_{i_{s-1}}, \ldots, y_{s-l}\in R_{i_{s-i}}$ have been recursively defined. If $l<s$, then let $x_{i_{l+1}}$ be the gate of x_l in $R_{i_{l+1}}$ and $y_{i_{s-l-1}}$ be the gate of y_{s-l} in $R_{i_{s-l-1}}$.

Property 1. *The vertices $x_1, x_2, \ldots, x_s$ lie on a common shortest path between x and y. Similarly, the vertices $y_{s-1}, y_{s-2}, \ldots, y_0$ lie on a common shortest path between y and x.*

Proof. It suffices to establish that x_1 belongs to a shortest path between x and y, the general statement is obtained by induction on s. Consider the cut $\{A_i, B_i\}$ of $\mathscr{C}$ such that $bd(A_i)\subseteq R_{i_0}$ and $bd(B_i)\subseteq R_{i_1}$. Then A_i contains the region R_{i_0}, while the halfplane B_i contains the regions $R_{i_1}, \ldots, R_{i_s}$. In particular, $x\in A_i$ and $y\in B_i$. The gate of x in B_i is a vertex of $bd(B_i)$. Since $bd(B_i)\subset R_{i_1}$, this gate coincides with the gate of x in R_{i_1}, i.e., with x_1. Hence $d_G(x,y) = d_G(x,x_1) + d_G(x_1,y)$. □

By a *corner* of a plane quadrangulation we will mean a vertex of degree 2.

Property 2. *Every quadrangulation $G\in\mathcal{Q}_4$ contains at least four corners.*

Proof. Let f denote the number of inner faces of G, e the number of edges, n the number of vertices, b the number of vertices incident with the outer face, and c the number of corners. Then $f-e+n=1$ and $4f+b=2e$ hold according to Euler's formula and the hypothesis that all inner faces have 4 edges. Eliminating f yields $4n-b-4=2e$. On the other hand, from the condition on the vertex degrees of inner vertices, we obtain the inequality $2e\geqslant 4(n-b)+3(b-c)+2c=4n-b-c$, whence $c\geqslant 4$, as required. □

We conclude with a (trivial) hereditary property of the class $\mathcal{Q}_4$.

Property 3. *If $G\in\mathcal{Q}_4$ and R is a simply connected region of G, then $R\in\mathcal{Q}_4$.*

3. Routing in rectilinear cells

Throughout this section, $G=(V,E)$ is a rectilinear cell. The edges of G are divided into *horizontal* and *vertical edges*. Removing all vertical edges of G, we will obtain a graph consisting of horizontal edges of G grouped into horizontal paths (*h-paths*, for short) $hp_0,\dots,hp_s$; for illustration see Fig. 4. Each h-path is a connected component of this graph. Define the following tree $T^h := T^h(G)$: its nodes are the h-paths and two nodes hp_i and hp_j are adjacent in T^h if and only if there exists a vertical edge with one end in hp_i and another in hp_j (the definition of a rectilinear cell implies that T^h is indeed a tree). If two horizontal paths hp_i and hp_j are adjacent in T^h, the edges with one end in hp_i and another in hp_j will constitute an equivalence class of the relation Θ. Removing these edges from G, we will obtain a graph with two connected components P_{ij} and P_{ji}, where $hp_i \subseteq P_{ij}$ and $hp_j \subseteq P_{ji}$. We call P_{ji} a *pocket* of G with respect to hp_i. Here is another way to view P_{ij} and P_{ji}: removing the edge between hp_i, hp_j from T^h, we will obtain two subtrees T^h_{ij} and T^h_{ji} such that $hp_i \in T^h_{ij}$ and $hp_j \in T^h_{ji}$. Then P_{ij} contains precisely the vertices of G which lie on h-paths from T^h_{ij} and P_{ji} contains the vertices of G on h-paths from T^h_{ji}.

The subpath s_j of hp_i which is a border line of the cut defined by P_{ij} and P_{ji} is called the *support* of the pocket P_{ji}. Given a vertex $x \in hp_i$ and a pocket P_{ji} of hp_i, we say that P_{ji} is located *left* (resp., *right*) from x if the support of P_{ji} is left (resp., right) from x. Finally, we say that a vertex v of G is *left* from $x \in hp_i$ if either v belongs to a pocket left from x or $v \in hp_i$ and v is left from x (similarly define the vertices right from x). Let $\mathrm{Left}_i(x)$ and $\mathrm{Right}_i(y)$ denote the set of vertices left from x and the set of vertices right from x. If y is the vertex of hp_i immediately left (resp., right) from x, then $\mathrm{Left}_i(x)$ (resp., $\mathrm{Right}_i(x)$) consists of the vertices z such that every shortest path

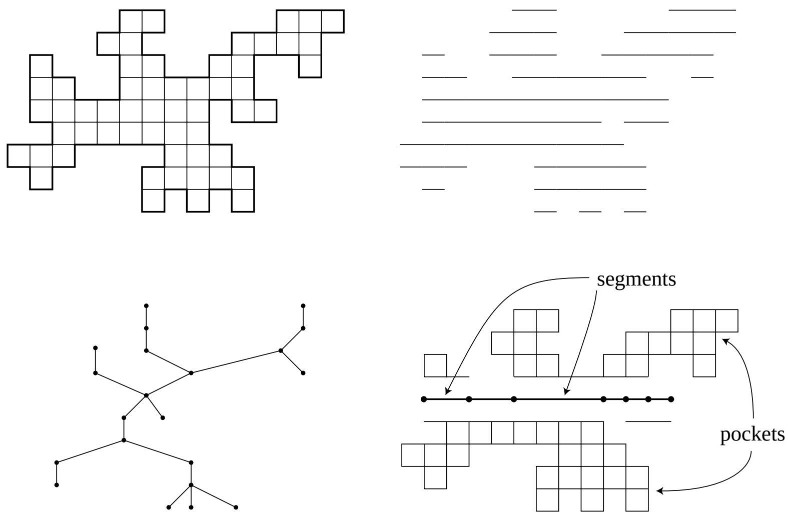


Fig. 4.

from z to x passes via y. Notice that $\mathrm{Left}_i(x) \cup \mathrm{Right}_i(x) \cup \{x\}$ contains all vertices of G except the vertices located in the pockets whose supports contain the vertex x.

3.1. Vertex labelling

Hereafter T^h is assumed to be rooted at hp_0. The labelling algorithm is recursive, starting with the root-path hp_0. At each stage we traverse some h-path, say hp_i and successively number its vertices and reserve contiguous intervals of available labels to all pockets of hp_i, except the pocket containing the root-path. Then we recursively continue the distribution of labels within each pocket of hp_i. Suppose P_{ji} is a pocket of hp_i not containing hp_0. To specify the labelling in P_{ji}, we first label the h-path hp_j of P_{ji}. At the previous stage, we assigned to P_{ji} a contiguous interval $[a_j \dots b_j]$ of numbers. Now, we label the vertices of hp_j with numbers from $[a_j \dots b_j]$ and reserve contiguous subintervals of $[a_j \dots b_j]$ to each pocket of hp_j not containing hp_0. The labelling algorithm finishes when all h-paths of G are labelled. At each stage we have the following global picture: one part of vertices of G is labelled, and the remaining vertices are grouped into pockets with common or distinct supports, and to each such pocket a subinterval of $[1 \dots n]$ is assigned.

Now we describe the recursion step in details. Suppose as above that the current h-path is hp_j, its father in T^h is the path hp_i, and the circular interval assigned to P_{ji} is $I = [a_j \dots b_j]$. To facilitate the exposition, we assume, without loss of generality, that I is a linear interval (if not, this can be done by a suitable rotation of the circular interval $[1 \dots n]$). Notice that P_{ij} is the pocket of h_j containing the root hp_0. Denote the pockets of hp_j by $P_{i_1 j}, \dots, P_{i_k j}$. Every edge and vertex of hp_j may belong to 0,1, or 2 supports of pockets. We partition the path hp_j from left to right into maximal by inclusion subpaths $c_1, c_2, \dots, c_q$ (called *segments*) each consisting of vertices which belong to supports of the same pockets of hp_j. The support of each pocket is a union of one or several consecutive segments.

We start by labelling the rightmost segment c_t in the support of P_{ij} with $|c_t|$ smallest labels of $[a_j, b_j]$ and update I by setting $I = [a_j + |c_t| - 1 \dots b_j]$. To label the remaining segments and pockets of hp_j, we traverse the segments of hp_j first from c_t to the left and then from the rightmost segment of hp_j until c_t. Notice that if c_t belongs to the support of yet another pocket, the algorithm will end up by labelling this pocket. Let c_l be the current segment with respect to a chosen direction of labelling. We have three possibilities:

(i) c_l does not belong to any support of a pocket. Then label c_l with the interval consisting of first $|c_l|$ numbers from I and update I (case of I_{13} from Fig. 5);

(ii) c_l belongs to the support of exactly one pocket P_{tj}. If P_{tj} was not labelled, then assign to this pocket an interval consisting of $|P_{tj}|$ smallest labels of I, then label c_l with the next portion of $|c_l|$ available labels, and update I (for an illustration, see the segment I_2 in Fig. 5). Otherwise proceed as in case (i);

(iii) c_l belongs to the supports of two pockets $P_{t_1 j}$ and $P_{t_2 j}$. If one of these pockets, say $P_{t_1 j}$ has been already labelled, then first we label c_l with $|c_l|$ smallest labels of I, then assign to $P_{t_2 j}$ the interval consisting of the next portion of $|P_{t_2 j}|$ consecutive labels of I, and update I (case of I_7 in Fig. 5). Otherwise, if neither of two pockets is

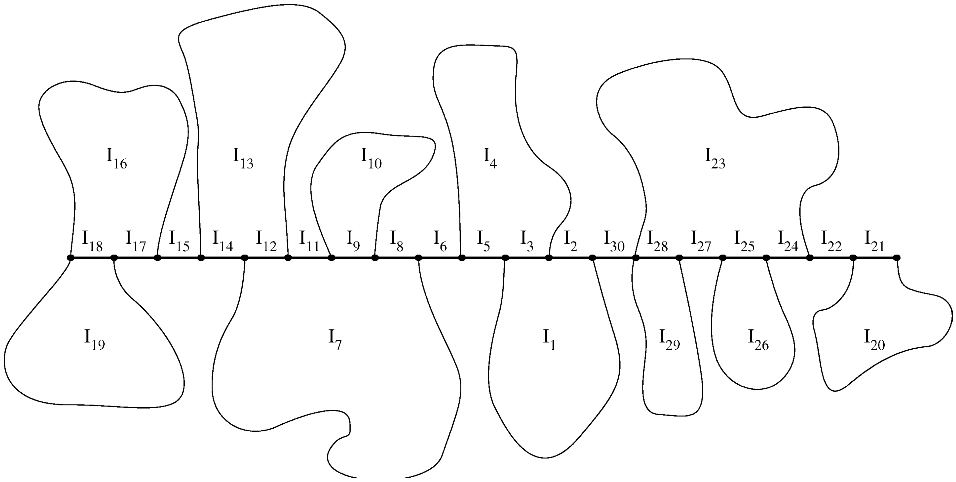


Fig. 5. Reserving labels for pockets and segments.

labelled (in this case, all three paths c_l, s_{t_1}, and s_{t_2} share a common end-vertex), then first label the pocket with the largest support, next label the segment c_l, then label the second pocket, and, finally, update the interval I (case of I_{14} in Fig. 5).

Notice that case (i) may occur only if G is not two-connected. After the distribution of labels to the segments and the pockets of hp_j, the vertices in each segment are labelled in a left-to-right order. For a vertex v of G, let $\mathscr{L}(v)$ be the label of v given by our algorithm. We conclude with the following property of the labelling scheme.

Lemma 1. *Let hp_j be an arbitrary h-path of G. Then the vertices in one pocket of hp_j form a single interval in the labelling.*

Proof. The property follows from the labelling procedure: while treating the current h-path, its pockets are labelled with single circular intervals, and the pocket containing the root-path has been labelled at some previous stage. On the other hand, the labels of pockets occurring on previous iterations do not change. □

3.2. Edge labelling

First, pick a vertical edge uv of G, with $u \in hp_i$ and $v \in hp_j$. Assign to $\mathscr{I}(u,v)$ the interval of labels of P_{ji}, similarly let $\mathscr{I}(v,u)$ be the interval of labels of P_{ij}.

Now, assume that uv is a horizontal edge of the h-path hp_j, say u is left from v. Assign to $\mathscr{I}(u,v)$ the set of labels of vertices from $\text{Right}_j(u)$ and to $\mathscr{I}(v,u)$ the set of labels of vertices from $\text{Left}_j(v)$. We assert that $\mathscr{I}(u,v)$ and $\mathscr{I}(v,u)$ occupy one or two circular intervals. If the edge uv does not lie in the support of some pocket, then $\text{Left}_j(v) \cup \text{Right}_j(u) = V$ and each $\mathscr{I}(u,v)$ and $\mathscr{I}(v,u)$ is a single circular interval. To

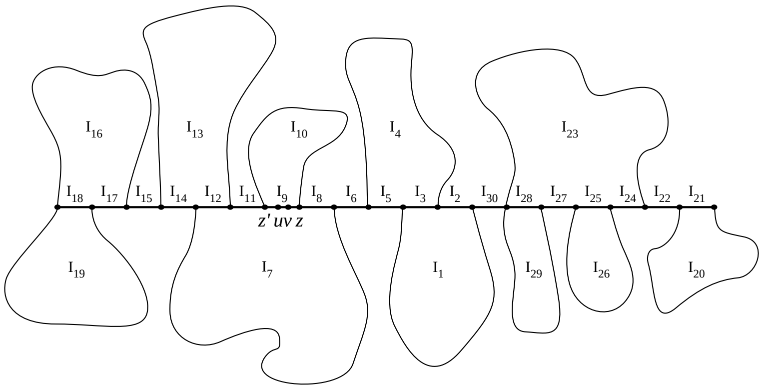


Fig. 6.

consider the remaining cases, suppose that u and v are left from c_t (the case when u and v are right from c_t is analogous by exchanging the roles of u and v). Two cases may occur.

Case 1. The edge uv belongs to the supports of two pockets P_{t_1j} and P_{t_2j} of hp_j.

Then u and v lie on a common segment c_l, therefore their labels are consecutive numbers. According to the labelling algorithm, one of the pockets, say P_{t_1j}, was labelled before c_l and the second pocket was labelled immediately after c_l. Hence $\mathscr{I}(v,u)$ occupies two intervals: the vertices of c_l which are left from v form one interval and the remaining vertices of $\text{Left}_j(v)$ form a single contiguous interval in the labelling. The left endpoint of this interval is the next label after the right endpoint of the label of the pocket P_{t_2j}, and the right endpoint is the largest label of a vertex in a segment or a pocket of hp_i which is left from c_t. In order to show that $\mathscr{I}(u,v)$ consists of two intervals, consider the segment c_k which was labelled immediately after the pocket P_{t_1j}. The segment c_k is either the leftmost segment which belongs to the support of P_{t_1j} or the second leftmost segment (this case occurs when the leftmost segment belongs to the support of a previously labelled pocket). Let z be the rightmost end-vertex of c_k. Denote by P the subpath of hp_i comprised between v and z (see Fig. 6). Clearly P belongs to the support of P_{i_1j}. Consider the subset X of $\text{Right}_j(u)$ consisting of P and all pockets having their support in P. Due to the labelling algorithm, the labels of vertices from X form a single contiguous interval $[\mathscr{L}(z)\ldots\mathscr{L}(v)]$. Indeed, after labelling P_{t_1j}, the algorithm labels c_k first, next it labels the pocket having the segment c_k as support (if such a pocket exists), and then it considers the segment immediately left from c_k. It treats this segment in the same way as c_k. The labelling of X will finish when the algorithm arrives at the segment c_l. Thus the label of X is indeed $[\mathscr{L}(z)\ldots\mathscr{L}(v)]$. Using similar arguments, one can show that $\text{Right}_j(u) - X =: \text{Right}_j(z)$ constitutes a single circular interval.

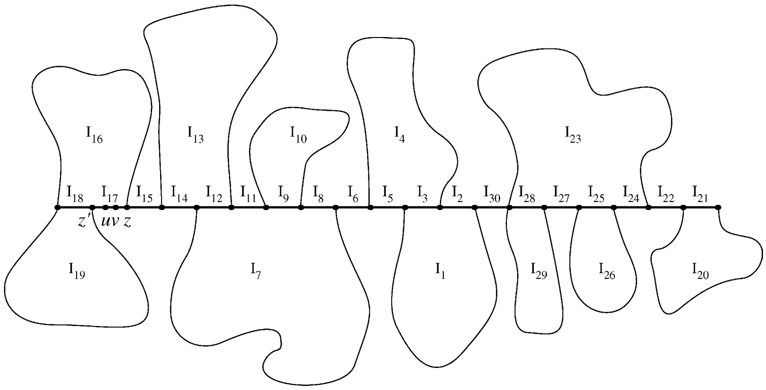


Fig. 7.

Case 2. The edge uv belongs to the support of a single pocket P_{t_1j}.

First consider $\mathscr{I}(v,u)$. If u does not belong to another support of a pocket, then the labels of all vertices from $\mathrm{Left}_j(v)$ constitute a single interval. Otherwise, proceed as in Case 1 to establish that $\mathscr{I}(v,u)$ occupies two intervals.

It remains to prove the same thing about $\mathscr{I}(u,v)$. Let c_l be the segment containing v. First suppose that v does not belong to another support of a pocket. In this case the pocket P_{t_1j} was labelled before the segment c_l. As in Case 1, consider the segment c_k which was labelled immediately after P_{t_1j}. Using similar arguments as in Case 1, one can show that $\mathscr{I}(u,v)$ consists of two circular intervals: one is generated by the labels of all vertices of hp_i between v and the rightmost vertex z of c_k and all pockets with supports in this path, and the second interval is formed by all vertices of G which are right from z (see Fig. 7).

Now, suppose that v belongs to the support s_{t_2} of a pocket P_{t_2j}. Clearly, $u \notin s_{t_2}$ from the initial assumption, whence v is the leftmost vertex of s_{t_2} and c_l. If P_{t_2j} was labelled before c_l, then according to the algorithm, the pocket P_{t_1j} was labelled after $c_l = s_{j_1} \cap s_{j_2}$. In this case, $\mathscr{I}(u,v)$ forms a single circular interval. Otherwise, if P_{t_2j} was labelled after $c_l = s_{j_2}$, then P_{t_1j} was numbered before c_l. Using arguments as in Case 1, one can show that $\mathscr{I}(u,v)$ consists of two circular intervals. Concluding, we have established the following result:

Lemma 2. *For every edge uv of G each of the labels $\mathscr{I}(u,v)$ and $\mathscr{I}(v,u)$ occupies at most two contiguous circular intervals.*

3.3. Routing

We continue by showing that for distinct vertices u,v of G the messages from u to v are rooted via a shortest path, i.e., that there is a shortest (u,v)-path $R(u,v) =$

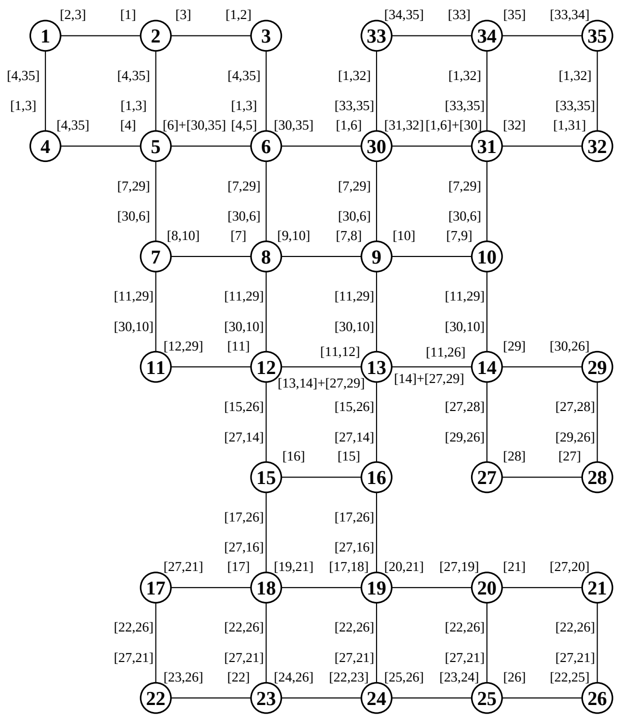


Fig. 8. A routing scheme in a rectilinear cell.

$(u = x_0, x_1, \ldots, x_{k-1}, x_k = v)$ such that $\mathscr{L}(v) \in \mathscr{I}(x_{i-1}, x_i)$ for every $i = 1, \ldots, k$. We proceed by induction on $k = d_G(u, v)$. If u and v are adjacent, this is obviously true. Now, let $k \geqslant 2$. Consider the h-path hp_i passing via u. Let w be the vertex of hp_i immediately right from u. As we noticed already (and this easily follows from the fact that h-paths are gated), w lies on a shortest path between u and any vertex located right from u. Therefore, if $v \in \mathrm{Right}_i(u)$, then $\mathscr{L}(v)$ belongs to the label of the directed edge (u, w). Since $d_G(u, v) = d_G(w, v) + 1$, applying the induction assumption to the couple w, v, we obtain the shortest path $R(u, v) = \{u\} \cup R(w, v)$ along which messages from the source u are routed to the destination v. Now, suppose that v belongs to a pocket P_{ji} whose support s_j contains the vertex u. Let z be the neighbour of u in P_{ji}. According to the algorithm, the label of the directed edge (u, z) is precisely the label of the pocket P_{ji}. In particular, $\mathscr{L}(v) \in \mathscr{I}(u, z)$. Since z is the gate of u in P_{ji}, we have $d_G(u, v) = d_G(z, v) + 1$.

Again, applying the induction hypothesis to the pair z, v we obtain the shortest path $R(z, v)$. Adjoining to this path the edge uz to derive the desired shortest path $R(u, v)$. Notice that the shortest paths used in the routing "prefer" vertical edges provided the current vertex and the destination v are in a common pocket and use horizontal edges only to move towards the current pocket hosting v (for illustration see Figs. 8 and 9). Summarizing, here is the main result of this section:

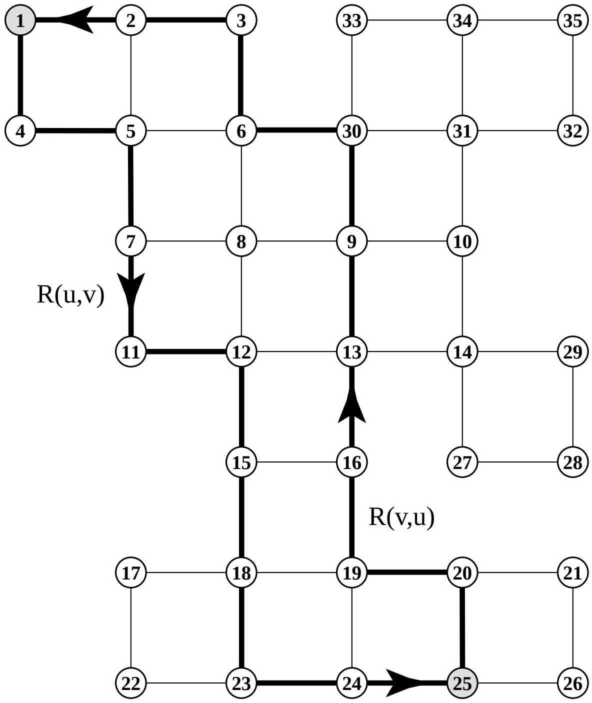


Fig. 9.

Theorem 1. *For rectilinear cells the described routing scheme* $\mathscr{R} = (\mathscr{L}, \mathscr{I})$ *is an optimal* 2-*SIRS and an optimal* 3-*SLIRS*

Notice that the routing scheme can be constructed in total linear time. For this, we construct the tree T^h and assign to each node hp_i the number of vertices in the h-path hp_i. Using this, one can compute for each edge of T^h the weight of the two subtrees defined by this edge. This can be done recursively by taking the reverse ordering defined by a DFS numbering of the nodes of T^h. With this information at hand, the labelling of vertices and pockets is performed in total $O(n)$ time. Having assigned intervals of admissible labels to all pockets, we immediately can write down the intervals affected to vertical oriented edges. Traversing each horizontal path from left to right and from right to left, we derive the labels of all horizontal arcs of this path, establishing our assertion.

Open question. *We conjecture that there exists a rectilinear cell not admitting an optimal* 1-*SIRS.*

The results of this section can be extended in a straightforward way to two-connected quadrangulations $G \in \mathscr{Q}_4$ in which all inner vertices have degree 4 and all boundary

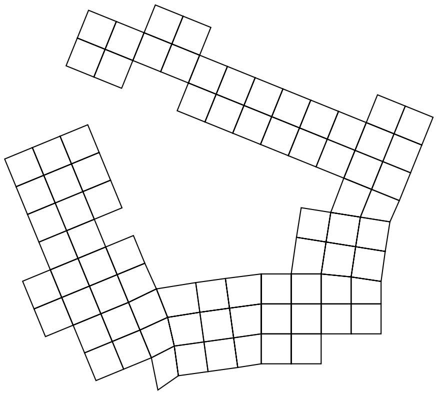


Fig. 10.

vertices have degree $\leqslant 4$. We will use the same name *rectilinear cells* for such graphs as well. Notice that not every such graph can be represented as a subgraph of the rectilinear grid: one simple example is given in Fig. 10 below. To adapt the routing scheme to such graphs G, we have to find the analogous of h-paths. For this, pick a convex cut of G, say $\{A_1, B_1\}$, and take its border lines $bd(A_1)$ and $bd(B_1)$. Extend these paths to maximal by inclusion convex (alias, gated) paths P', P'' of G. Let us explain how to extend the path $P := bd(A_1)$. Pick the end-vertices x, y of P and denote by x', y' their neighbours in P. If x has an adjacent inner vertex v such that x', x, v do not belong to a common inner face, then set $P := P \cup \{v\}$. Otherwise, if x has a neighbour v on the boundary of G and x', x, v do not lie on a common inner face, then again set $P := P \cup \{v\}$. If such a vertex v does not exist, then we stop augmenting P from this side. Perform the same operation on the other side of P. The resulting path P' is locally convex, because no three consecutive vertices of P' lie on a common 4-face, therefore it is convex and gated. Now, take P' and P'' as the first two h-paths and remove the edges in between. Next, pick all convex cuts of G which have one border line contained in P', take their opposite border lines and augment them as we did with $bd(A_1)$ and $bd(B_1)$. The resulting convex paths are the next h-paths. Delete the edges from the equivalence classes defining all such cuts. After this operation, P' will become a connected component in the current graph (because of degree constraints). Perform the same operation with P'', and then with each h-path found after P' and P''. Continuing this, we will arrive at a graph in which all connected components are convex paths containing one or several border lines of some cuts of G. With this structure at hand, we can further define the tree T^h, the pockets and their supports, which altogether allow to construct the required routing scheme.

Finally, notice that the results can be immediately extended to quadrangulations from $\mathcal{Q}_4$ in which all maximal two-connected components are rectilinear cells.

4. Routing in quadrangulations from $\mathcal{Q}_4$

In this section, we present the main result of this note: an optimal 7-SLIRS for graphs $G=(V,E)$ from $\mathcal{Q}_4$. For this, we construct a collection $\mathscr{C} \subset \mathscr{L}$ of laminar cuts which partitions G into rectilinear or pseudorectilinear cells. Each pseudorectilinear cell is further subdivided into rectilinear cells using a new family of laminar cuts (but with respect to this cell only). Applying the algorithm from Section 3, the fact that the regions in the resulting partition are gated, and the treelike structure of the family $\mathscr{C}$, we can establish the desired routing scheme.

4.1. Construction of $\mathscr{C}$

The collection $\mathscr{C}$ is constructed step by step, by adding each time a new convex cut which is laminar to previously defined cuts. We take an arbitrary convex cut as the first cut of $\mathscr{C}$ and place its halves into a queue $\mathbf{Q}$. Now, suppose we have defined a collection $\mathscr{C}$ of i laminar cuts. Their pseudolines partition G into $i+1$ regions. Several of these regions are in the current queue. Pick the region R at the front of $\mathbf{Q}$. We search R for a new laminar convex cut. If such a cut is not found, we delete R from $\mathbf{Q}$. Now suppose a new laminar cut $\{A,B\}$ has been found. Its pseudoline C partitions R into two regions R' and R'' (notice that C intersects the boundary of R in two edges of ∂G). One of these regions, say R'', is a halfplane of G (hence it can be further treated as the halves of the first cut). We add $\{A,B\}$ to $\mathscr{C}$, replace R by R' at the front of $\mathbf{Q}$ and add the region R'' at the back of the queue. The algorithm stops when $\mathbf{Q}$ is empty.

Notice that the region $R_0:=R$ which just arrived in front of $\mathbf{Q}$ is always a halfplane. Let R^+ be the first region removed from the front of the queue after the arrival of R in head. Obviously, $R^+\subseteq R$ is a region in the final subdivision of G. We call the intermediate steps between handling R and removing R^+ from $\mathbf{Q}$ a *phase* of the algorithm. During a phase we cut off disjoint halfplanes from R and add their defining cuts to $\mathscr{C}$. The updated region will be also denoted by R. To prove the correctness of the algorithm, it suffices to precise the evolution of the region R in front of $\mathbf{Q}$ during the phase and to establish the structure of the final region R^+.

Denote by L_0 the border line of the cut of G whose halfplane is the initial region R_0. Obviously, L_0 will belong to boundaries of all regions R occurring in the phase. For this reason, we call L_0 a *basis line*. We need two other terms, inspired by computational geometry. The complement of L_0 in the boundary of the current region R is called the *beach line* of R and is denoted by $\beta(R)$. This path is an alternating sequence of subpaths of the boundary of G and subpaths of border lines of laminar cuts found within the phase but before handling the current region R (in pictures, border lines look like arcs of parabolas or hyperbolas, whence the name). Finally, the *sweep front* $\alpha(R)$ consists of one or several paths (called *sweep lines*) with both end-vertices on the beach line. Together with the basis line and some subpaths in the beach line, the sweep front bounds a subregion R^+ of R. The vertices in R^+ are precisely those vertices of R which have been already swept. For each vertex $v\in\alpha(R)$, we denote by $\deg^-(v)$ the number of *external neighbours* of v, i.e., neighbours located in R but

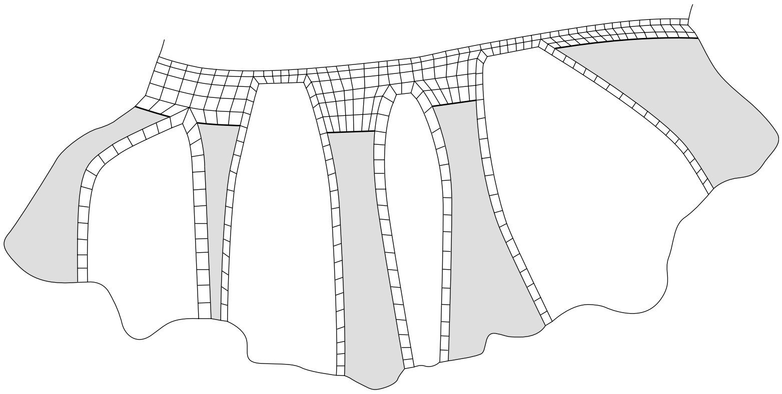


Fig. 11. A partially swept region.

outside $R^+ \cup \alpha(R)$. In one iteration of the phase, the vertices of current sweep line L are considered one after other. If a current vertex v of L has $\deg^-(v) \geqslant 2$, and there exists a cut passing in the neighbourhood of v such that one of its halfplanes is disjoint from R^+, then we add this cut to $\mathscr{C}$, update the region R, its beach line, and R^+. Finally, replace in the sweep front the path containing v by two its subpaths, and start a new iteration of the same phase. Otherwise, if all vertices in a path of the sweep front have been considered without finding a new cut, we advance this path, add the swept line to R^+, update the sweep front, and start a new iteration. In Fig. 11 we present an example of a partially swept region R together with its current sweep front.

Each path L of $\alpha(R)$ contains both its end-vertices on the beach line. These vertices divide the boundary of R into two chains C_L, C'_L. One of these chains, say C_L is disjoint from L_0. In analogy with rectilinear cells, call the region R_L bounded by L and C_L a *pocket* of L. Clearly $R_L \cap R^+ = L$. At each iteration of the phase, we take care to preserve the following structural invariant: *for each path L of the sweep front $\alpha(R)$, the chain C_L either consists of a single subpath of ∂G, or of a subpath of ∂G and a subpath P of a border line of some cut of $\mathscr{C}$, or of two subpaths P', P'' of border lines of two cuts of $\mathscr{C}$ and a subpath of ∂G in between*. In the first case, the pocket R_L is a halfplane. In the second and third cases, we call R_L a *bigon* and a *trigon*, respectively. The trigons and bigons can be viewed as variants of open triangles and strictly asymptotic open triangles in hyperbolic geometry; cf. Chapter 8 of [19]. For an illustration of these notions see Fig. 12.

At the beginning of the phase, R is a halfplane. The sweep front $\alpha(R)$ consists solely of the basis line L_0. The region R^+ is empty and the pocket of L_0 is the whole region R. Now, let a current region R be given. Suppose its beach line $\beta(R)$ and the sweep front $\alpha(R)$ are defined, and a current line $L \in \alpha(R)$ has to be considered ($L = L_0$ at the beginning). Denote by v'_0, v''_0 the end-vertices of L. We traverse the vertices of the line L from one end to another (to fix an orientation, say from left to right), and stop at the first vertex (if it exists) with at least two external neighbours. In dependence of its

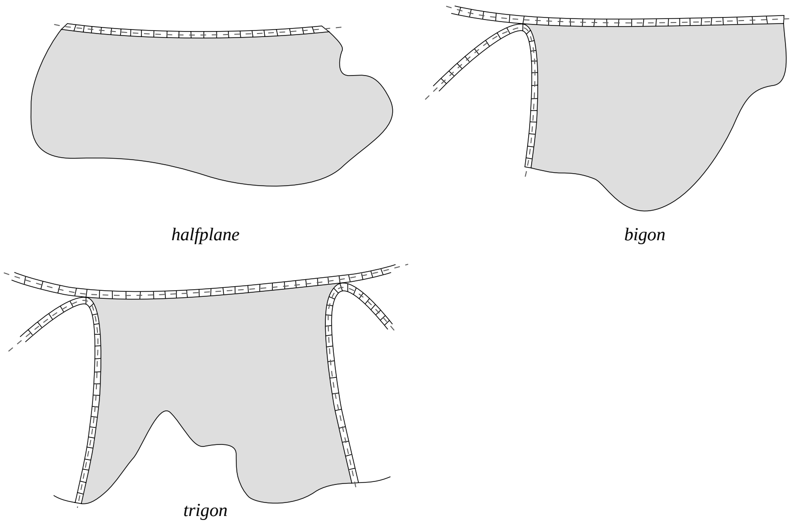


Fig. 12. Halfplane 1.

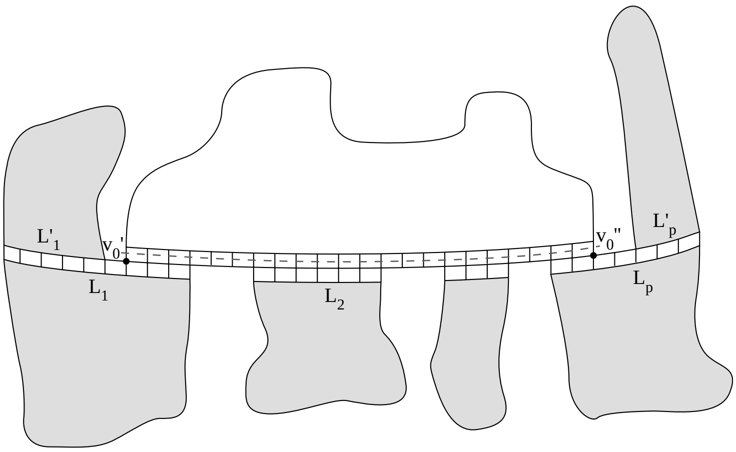


Fig. 13. Halfplane 2.

existence, location, and the form of the pocket R_L, three main cases may occur: R_L is a halfplane, a bigon, or a trigon.

Case Halfplane 1: For each vertex $v \in L$, we have $\deg^{-}(v) \leqslant 1$. The external neighbours of vertices of L induce one or several paths $L_1, \ldots, L_p$ as in Fig. 13 (since the border lines are convex, the external neighbours of two vertices from L must be distinct). Every path $L_2, \ldots, L_{p-1}$ is a border line of some cut of G, while L_1 and L_p are subpaths of border lines of two cuts $\{A_i, B_i\}$ and $\{A_l, B_l\}$, say $L_1 \subseteq bd(A_i)$ and $L_p \subseteq A_l$. Denote by L'_1 the subpath of $bd(B_i)$ induced by all its vertices left from v'_0

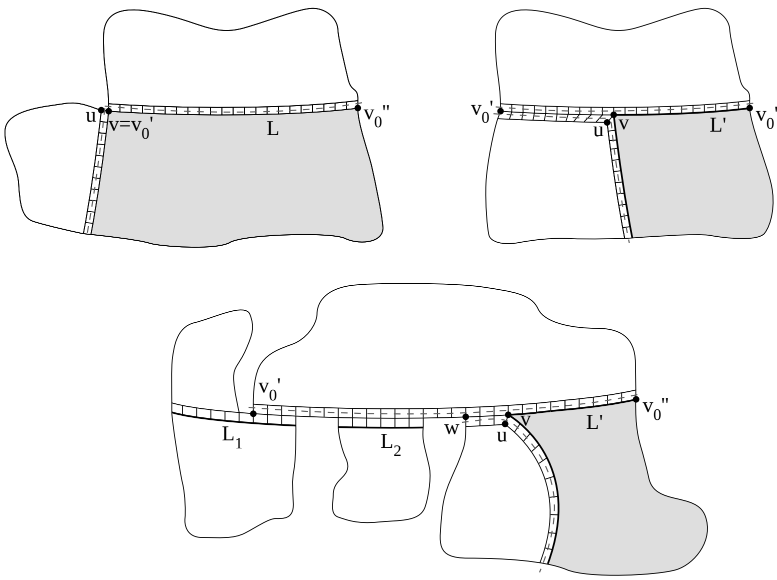


Fig. 14. Halfplane 2.

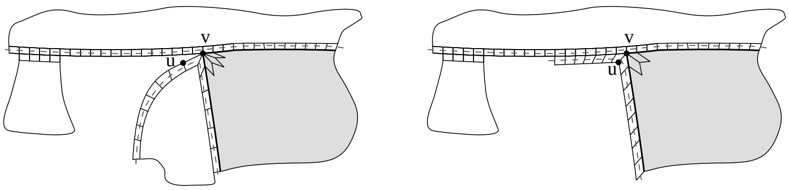


Fig. 15. Halfplane 2.

and denote by L'_p the subpath of $bd(B_l)$ induced by all its vertices right from v''_0. We remove the path L from $\alpha(R)$ and add it to R^+. Then add to the sweep front the paths $L_1,\dots,L_p,L'_1,L''_p$. The pockets of all these paths are also halfplanes.

Case Halfplane 2: We have found the leftmost vertex $v\in L$ with at least two external neighbours. Let u be the leftmost external neighbour of v. Suppose that the edge uv belongs to the jth equivalence class of Θ, where $v\in A_j$ and $u\in B_j$. Add the cut $\{A_j,B_j\}$ to $\mathscr{C}$ (that this and subsequent additions to $\mathscr{C}$ are feasible will be established latter). Update R by letting $R=R\backslash B_j$. Let L' be the subpath of L induced by v and all vertices right from v (if $v=v'_0$, then $L'=L$). First assume that $v'_0\in bd(A_j)$. Then either $v=v'_0$ or v'_0 must be the left end-vertex of the path $bd(A_j)$. In this case, replace in the sweep line the path L by L' and add $L\backslash L'$ to R^+. The pocket of L' is a bigon (see Figs. 14 and 15). Now assume that the left end-vertex of $bd(A_j)$ is the unique external neighbour of a vertex $w\in L$.

As in the previous case, consider the paths $L_1,\dots,L_p$ defined by the external neighbours of vertices of L comprised between v'_0 and w. Every path $L_2,\dots,L_p$ is a border

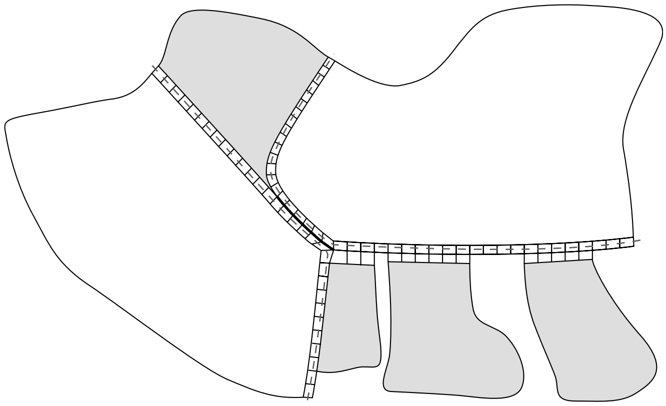


Fig. 16. Bigon 1.

line of some cut of G, while L_1 is a subpath of the border line $bd(A_i)$ of some cut $\{A_i, B_i\}$ $(i \neq j)$. Denote by L_1' the subpath of $bd(B_i)$ induced by all its vertices left from v_0'. We replace in $\alpha(R)$ the path L by the paths $L_1, \ldots, L_p, L_1', L'$ and add the vertices of L left from v to the region R^+. The pockets of all these paths except L' are halfplanes and the pocket of L' is a bigon.

Now we show how to handle bigons. Let R be a bigon bounded by a sweep line L, a subpath P in the border line of some cut of $\mathscr{C}$, and a subpath of ∂G. Let v_0' be the common vertex of L and P (see Figs. 16–18). We say that R is a bigon with sides L and P and vertex v_0'. Let u_0 be the neighbour of v_0' in P. We sweep the vertices of L from left to right, until a vertex with at least two external neighbours is found (if it exists).

Case Bigon 1: For each vertex $v \in L$, we have $\deg^-(v) \leqslant 1$. Replace in $\alpha(R)$ the path L by the paths induced by the external neighbours of the vertices from L. The pockets of all such paths are halfplanes, except the pocket of the path beginning at v_0', which maybe a bigon having u_0 as vertex and this path and $P - \{v_0'\}$ as sides (see Fig. 16).

Case Bigon 2: The vertex v_0' has a second external neighbour $u \notin P$, where v_0', u_0, u lie on a common inner face. Let uv be in the jth equivalence class of Θ, where $v \in A_j$ and $u \in B_j$. As in the previous case, add the cut $\{A_j, B_j\}$ to $\mathscr{C}$ and set $R := R \backslash B_j$. Let y be the furthest from v_0' common vertex of the paths P and $bd(A_j)$. Analogously, let x be the furthest from v_0' common vertex of the paths L and $bd(A_j)$ (it may happen that v_0' coincides with one or both vertices x, y). Update the beach line in the following way: remove the subpath of P between v_0', y and add the path $bd(A_j)$. Replace in the sweep front the path L by its subpath L' induced by x and all vertices of L to its right. Additionally, add to $\alpha(R)$ the subpath P' of P starting with y and avoiding v_0' (see Fig. 17). Notice that the pocket of L' is the bigon with x as vertex and having L' and a subpath of $bd(A_j)$ as sides. Similarly, the pocket of P' is the bigon having y as the vertex, and P' and a subpath of $bd(A_j)$ as sides. In both cases, no new type of pocket occurs.

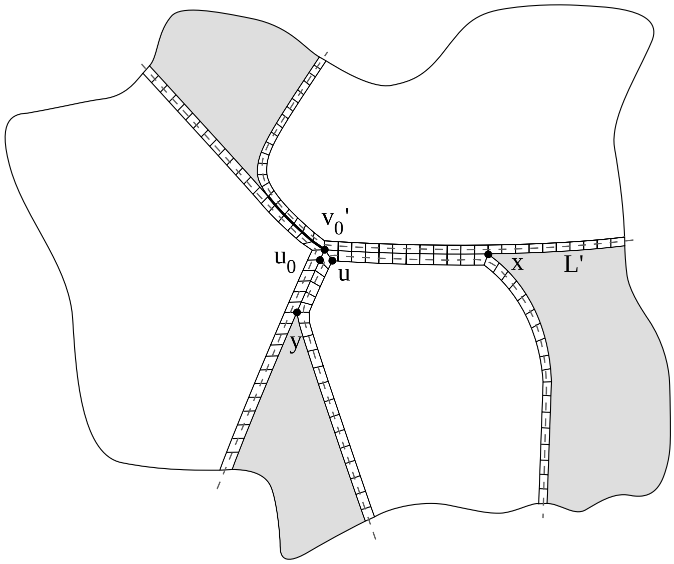


Fig. 17. Bigon 2.

So assume that u_0 is the unique external neighbour of v_0'.

Case Bigon 3: We have found the leftmost vertex v of L with $\deg^-(v) \geqslant 2$. Clearly, $v \neq v_0'$. Let w, u be the first and second leftmost external neighbours of v. Assume that the edge uv belongs to the jth equivalence class of the relation Θ with $v \in A_j$ and $u \in B_j$. Notice also that the edges vw and $v_0'u_0$ belong to the same equivalence class, say $vw, v_0'u_0 \in E_i$ with $v_0', v \in A_i$ and $u_0, w \in B_i$. Add the cut $\{A_j, B_j\}$ to $\mathscr{C}$ and set $R := R \setminus B_j$. Let x be the furthest from v common vertex of the paths L and $bd(A_j)$ (maybe x coincides with v). First add the path $bd(A_j)$ to the beach line. Next, replace in the sweep front the path L by the subpath L'' of $bd(B_i)$ between u_0 and w and the subpath L' of L induced by x and all vertices of L located right from x (see Fig. 18). Add the subpath of L comprised between v_0' and x to the region R^+. The pocket of L'' is a trigon bounded by the paths P, L'' and a subpath of $bd(A_j)$ consisting of all its vertices left from v. The pocket of L' is a bigon with x as a vertex and bounded by L' and the subpath of $bd(A_j)$ consisting of all its vertices right from x (notice that if x is the right end-vertex of L, then this pocket and the path L' are empty). So, in this case a new type of pocket appears.

Finally, we describe how to handle trigons. Let R be a trigon bounded by a line L of the sweep front, two subpaths P', P'' in the beach line (each of them is a subpath of a border line of a cut from $\mathscr{C}$) and a subpath of ∂G. Suppose P' bounds R from left and P'' bounds R from right. Let $P' \cap L = \{v_0'\}$ and $P'' \cap L = \{v_0''\}$. Denote by u_0' and u_0'' the neighbours of v_0' and v_0'' in P' and P'', respectively (see Figs. 20–27). Notice that in this case the line L is not a longer subpath of the border line of a cut from $\mathscr{C}$. Namely, either the path L belongs to the border line of a single cut or $L = L_1 \cup L_2$, where the paths L_1 and L_2 belong to border lines of two transversal cuts.

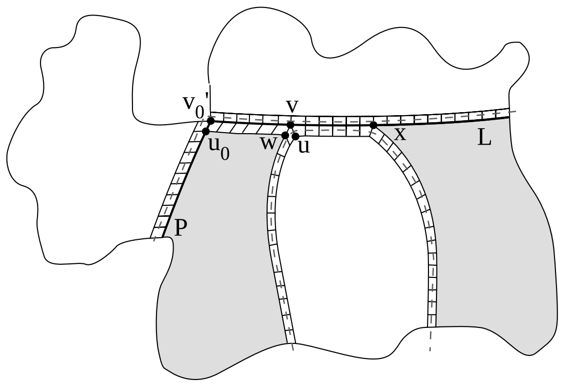


Fig. 18. Bigon 3.

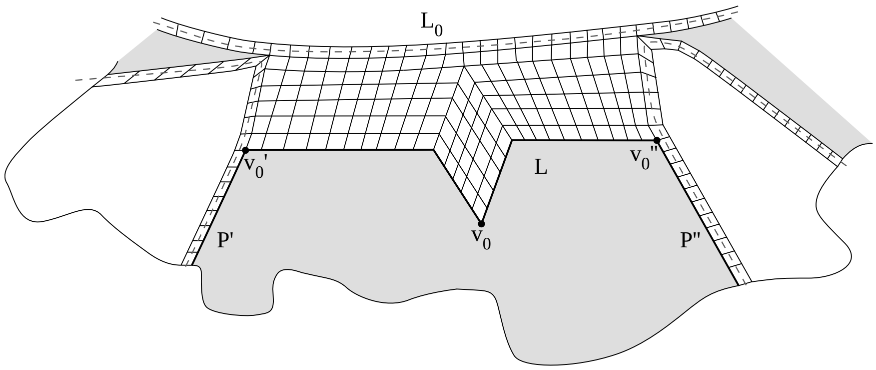


Fig. 19. A 2-line.

In this case, denote by v_0 the common vertex of L_1 and L_2 and call it the *peak* of L (for an illustration see Fig. 19). In the first case we say that L is an 1-line, in the second case L is called a 2-line.

While sweeping a trigon, the evolution of the sweep line is roughly the following: at the beginning it is a 1-line and it preserve this form (or it is replaced by several 1-lines) until we will come to a 1-line which contains exactly one vertex with at least two external neighbours and this vertex is an inner vertex of degree 5. Then the new sweep line becomes a 2-line. It can remain a 2-line during several iterations, until on this line we will find a vertex of degree at least 5. Then it is replaced by two 1-lines or by an 1-line and a 2-line. The region R^+ obtained at the end of the phase is not always a rectilinear cell, it may contain a certain number of inner vertices of degree 5. These vertices are kept in the list $D(R^+)$.

Case Trigon 1: Each vertex of $L\backslash\{v_0\}$ has at most one external neighbour and v_0 has precisely two such neighbours. Replace in $\alpha(R)$ the path L by the paths induced by all external neighbours of vertices of L. If there is only one such path, then its pocket

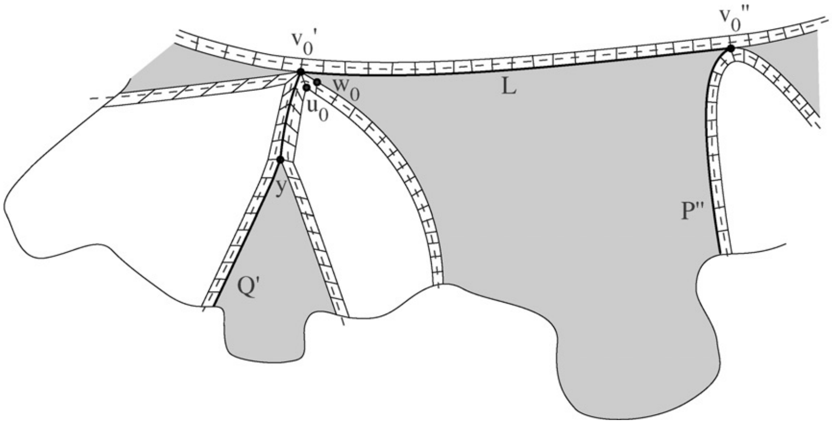


Fig. 20. Trigon 2.

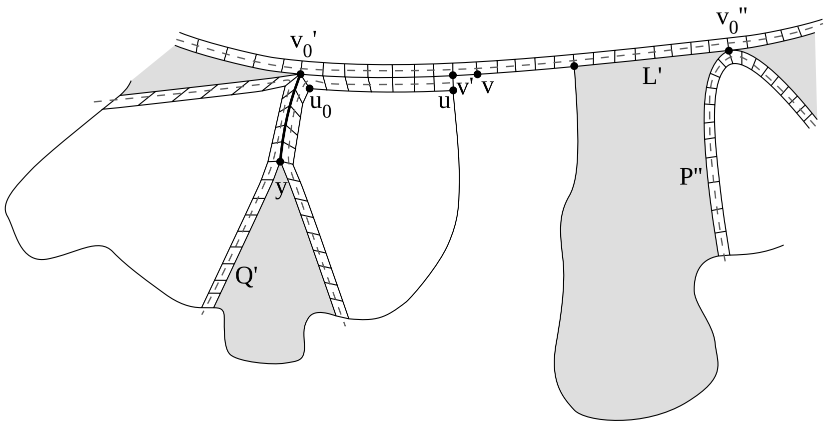


Fig. 21. Trigon 2.

will be also a trigon. Otherwise, the pockets of all these paths are halfplanes, except the pockets of the leftmost and the rightmost paths, which maybe bigons having u_0' and u_0'' as vertices and the paths $P' - \{v_0'\}$ and $P'' - \{v_0''\}$ as sides.

Now, assume that either $L\backslash\{v_0\}$ has a vertex with at least two external neighbours, or v_0 has at least three external neighbours.

Case Trigon 2: The vertex v_0' has a second external neighbour $u_0 \notin P$, where v_0', u_0', u_0 lie on a common inner face. Suppose the edge $v_0'u_0$ belongs to the equivalence class E_j, where $v_0' \in A_j$ and $u_0 \in B_j$. Let y be the furthest from v_0' common vertex of the paths P' and $bd(A_j)$.

First assume that v_0' has yet another external neighbour w_0 (see Fig. 20). This vertex may be chosen so that v_0', u_0, w_0 lie on a common inner face of G. Add the cut $\{A_j, B_j\}$

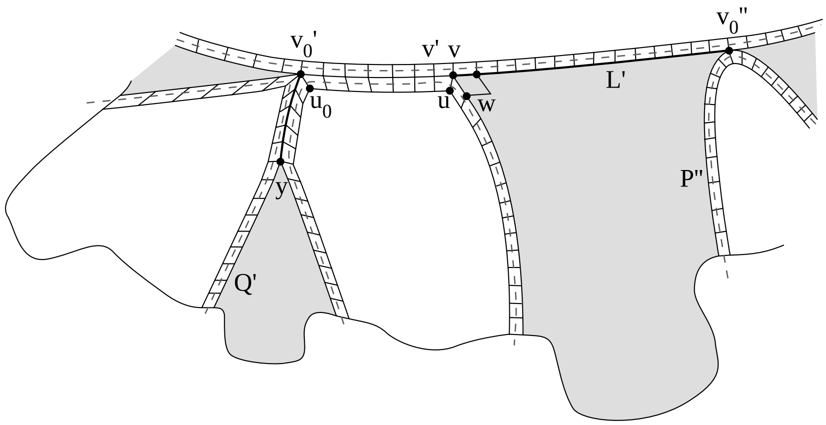


Fig. 22. Trigon 2.

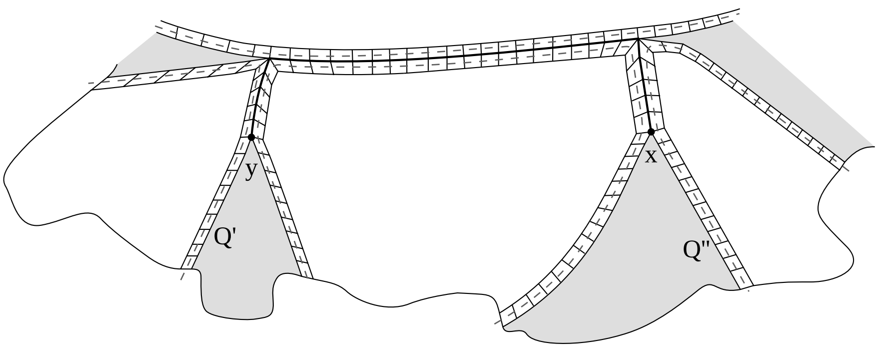


Fig. 23. Trigon 2.

to $\mathscr{C}$, set $R := R \backslash B_j$, and add the path $bd(A_j)$ to the beach line. Next, add to the sweep front the subpath Q' of P' comprised between y and the vertex of P' on the outer face of G. Finally, add the subpath of P' comprised between v_0' and y to the region R^+. The new pocket of L is a trigon bounded by a subpath of $bd(A_j)$ and the paths L, P''. The pocket of Q' is a bigon which is bounded by Q' and the subpath of $bd(A_j)$ comprised between y and the vertex of $bd(A_j)$ on the outer face of G (if these two vertices coincide, this pocket and the path Q' are empty).

Now assume that v_0' has only two external neighbours u_0' and u_0. Sweep L from left to right until we find the first vertex $v \in L$ (if it exists) with zero or at least two external neighbours. Suppose we have found a vertex v without external neighbours. Let v' be the left neighbour of v in L. Denote by u' the unique external neighbour of v'. Obviously, the edge $v'u'$ lies on the outer face of G. Moreover, this edge belongs to the equivalence class E_j. Now, proceed as in previous subcase: add the jth cut to $\mathscr{C}$ and update the beach line, the sweeping front and the region R^+ with one exception:

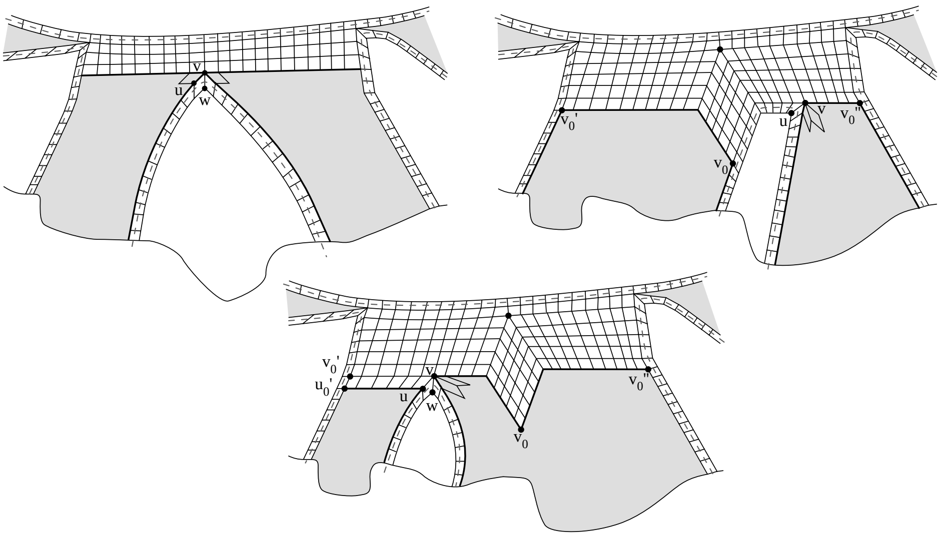


Fig. 24. Trigon 3i.

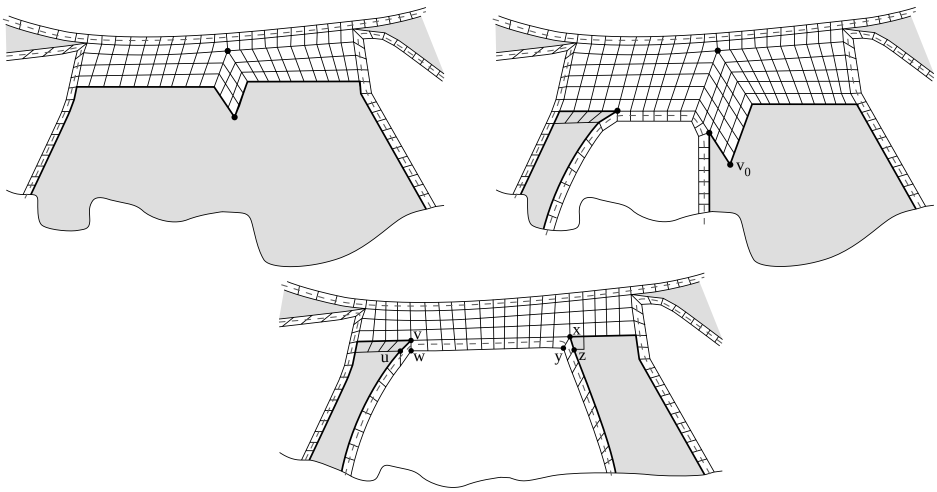


Fig. 25. Trigon 3ii.

in the sweep front replace L by its subpath L' comprised between v_0'' and the first vertex right from v having at least one external neighbour. Also add the subpath of L complementary to L' to the region R^+. In this case, the pocket of L' is not longer a trigon but a bigon (see Fig. 21).

Now, suppose that the vertex v has at least two external neighbours. Denote by u, w the first and the second leftmost external neighbours of v. Obviously, the edge

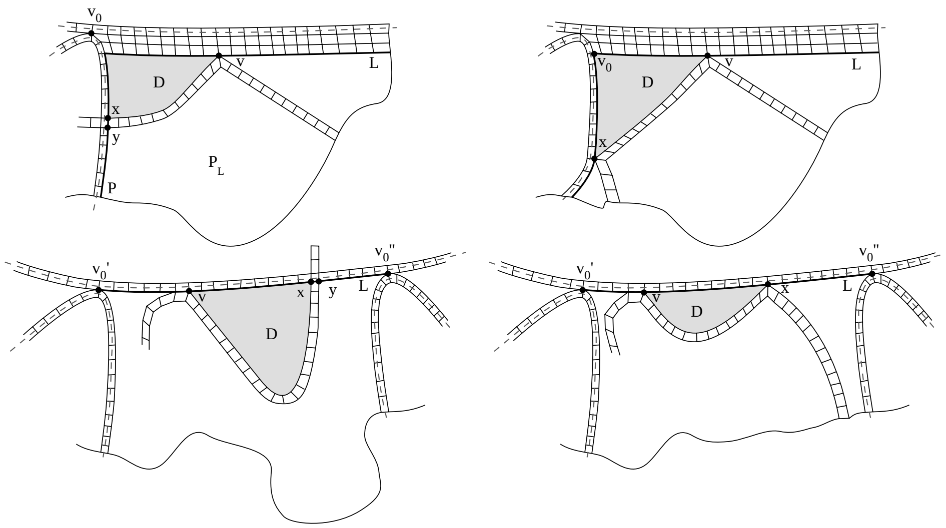


Fig. 26.

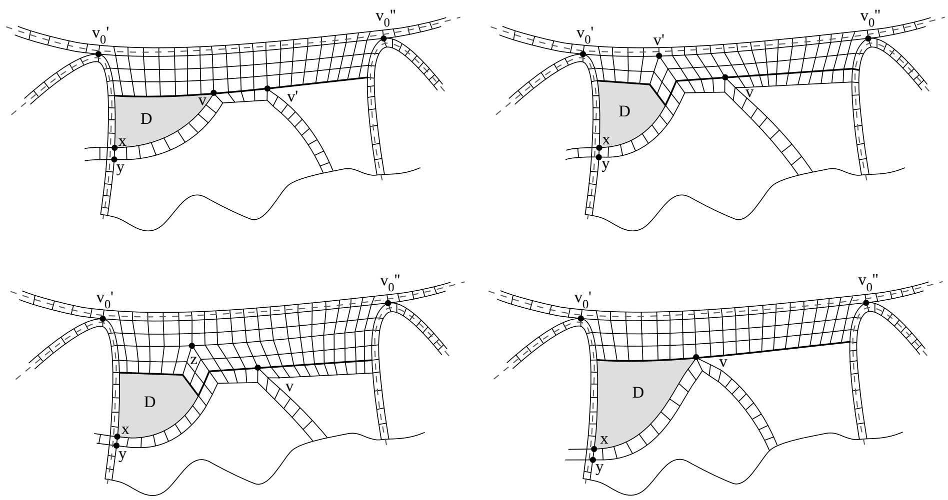


Fig. 27.

uv belongs to the same equivalence class E_j as $v_0'u_0$ (see Fig. 22). Again proceed as in two previous subcases: add the jth cut to $\mathscr{C}$ and update correspondingly the beach line, the sweeping front and the region R^+ with the following exception. First, in the sweep front we replace L by its subpath L' comprised between v and v_0''. If $v \neq v_0''$, then L' is nonempty, and its pocket is again a trigon.

Finally, if $v = v_0''$, let x be the furthest from v_0'' common vertex of the paths P'' and $bd(A_j)$ (see Fig. 23). In this case, we remove L from the sweep front and add this path to the region R^+. Additionally, add to the sweep front the subpath Q'' of P'' comprised between x and the vertex of P'' on the outer face of G, and add the subpath of P'' between v and x (except x) to R^+. The pockets of new sweep lines Q' and Q'' are two bigons having y and x as their vertices.

Finally suppose that every vertex of $L\backslash\{v_0'\}$ has exactly one external neighbour. Delete the path L from the sweep front and add it to the region R^+. Add to the sweep front the path L' induced by u_0', w_0, u_0 and the external neighbours of the vertices from $L\backslash\{u_0'\}$. The pocket of L' is also a trigon. Notice also that the resulting path L' is a 2-line.

Case Trigon 3: The unique external neighbour of v_0' is the vertex $u_0' \in P$. Let v be the leftmost vertex of L which has degree larger than or equal to 5.

Subcase (i): v has degree larger than or equal to 6.

Let u, w be the two leftmost external neighbours of v (see Fig. 24). If L is an 1-line or a 2-line with $v \in L_1$, then add to $\mathscr{C}$ the cut $\{A_j, B_j\}$ defined by the edge vw. Otherwise, if L is a 2-line and $v \in L_2$, then add to $\mathscr{C}$ the cut $\{A_j, B_j\}$ defined by the edge vu. In both cases, suppose that $v \in A_j$. Update R by letting $R := R\backslash B_j$ and add $bd(A_j)$ to the beach line. In the first case, replace in $\alpha(R)$ the path L by two paths L' and L'', where L' is the subpath of $bd(B_i)$ comprised between u_0' and u (here uv $\in E_i$ with $u \in B_i$) and L'' is the subpath of L comprised between v and v_0''. The pocket of both L' and L'' are trigons. Notice that if L is a 1-line, then both L' and L'' are 1-lines, else L' is a 1-line and L'' is a 2-line. Then add the subpath of L between u_0' and u to R^+. In the second case, replace L by a path L' induced by (unique) external neighbours of all vertices of L_1 and a subpath L'' of L_2 comprised between v and v_0''. Add L_1 and the subpath of L_2 between v_0 and v to the region R^+. In this case, each of the paths L' and L'' is a 1-path and their pockets are trigons. The occurring situations are illustrated in Fig. 24.

Subcase (ii): v has degree 5.

Let u, w be the two external neighbours of v and let $vw \in E_j$. If L is a 2-path, then proceed exactly as in Subcase (i) with the unique difference that if $v \in L_1$, then L'' is the subpath of L comprised between v_0'' and the rightmost common vertex z of L and $bd(A_j)$. In this case, we additionally add the subpath between v and x to R^+. So, further assume that L is an 1-path (see the first graph from Fig. 25). Then the analysis is similar to that from case Trigon 2.

Namely, we sweep L from v to right to find the next vertex $x \in L$ (if it exists) with zero or at least two external neighbours. If x does not have external neighbours, then we just follow the case Trigon 2. Now, suppose the vertex x has at least two external neighbours. Denote by y, z the first and the second leftmost external neighbours of x. Obviously, the edge xy belongs to the equivalence class E_j (see Fig. 25). Again proceed as before: add the jth cut to $\mathscr{C}$ and update correspondingly the beach line, the sweeping front and the region R^+.

Finally suppose that every vertex of $L\backslash\{v\}$ has exactly one external neighbour. Then delete the path L from the sweep front and add it to the region R^+. Add to the sweep front the 2-path L' induced by the external neighbours of the vertices from L (see Fig. 25). The pocket of L' is also a trigon. Finally add v to $D(R^+)$.

4.2. $\mathscr{C}$ is a collection of laminar cuts

We continue by establishing that the final collection $\mathscr{C}$ consists of laminar cuts only. Additionally, we must show that the pocket of each line in the sweep front is a halfplane, a bigon, or a trigon. This also will show the correctness of presented figures. The basic tools in this proof are Properties 2 and 3 of quadrangulations $G \in \mathscr{Q}_4$. In all occurring cases, we have a current line L in the sweep front and its pocket R_L which is either a halfplane, a bigon, or a trigon. Additionally, we have found a vertex $v \in L$ so that a cut $\{A_j, B_j\}$, intersecting an edge incident to v, is added to the collection $\mathscr{C}$. In all cases, we assumed that $v \in bd(A_j)$. The subpath of ∂R_L induced by the vertices which do not belong to the boundary of G is called the *inner boundary* of R_L and is denoted by $\partial' R_L$ (if R_L is a bigon, then $\partial' R_L = P \cup L$, otherwise, if R_L is a trigon, then $\partial' R_L = P' \cup L \cup P''$, else, if R_L is a halfplane, then $\partial' R_L = L$).

In order to prove that $\{A_j, B_j\}$ is laminar with all previous cuts from $\mathscr{C}$, it suffices to establish that E_j does not share common edges with the inner boundary of the pocket R_L. To prove the second assertion, additionally we must show that if the path $bd(A_i)$ touches $\partial' R_L$ in a vertex $x \neq v$, then the whole subpath of $bd(A_j)$ comprised between v and x belongs to $\partial' R_L$. Suppose by way of contradiction that either E_j intersects $\partial' R_L$ in some edge xy or that $bd(A_j)$ touches the inner boundary of R_L in some vertex x so that the subpaths of $bd(A_j)$ and $\partial' R_L$ comprised between v and x intersect only in v and x. In both cases, denote by C' and C'' the subpaths of $bd(A_j)$ and $\partial' R_L$ comprised between the vertices v and x. Denote by D the subgraph of G induced by all vertices located on the simple circuit $C' \cup C''$ or inside the region of the plane bounded by this circuit. From Property 3 we infer that D is a quadrangulation from $\mathscr{Q}_4$, therefore, by Property 2 it must contain at least four corners. However, in all occurring cases D contains either two or three corners: the vertex x, the furthest from v common vertex of $bd(A_i)$ and $\partial' R_L$, and one of the end-vertices v_0', v_0'' of the sweep line L (see Figs. 26 and 27 for generic cases).

4.3. Dealing with pseudorectilinear cells

Assume that the resulting collection $\mathscr{C}$ consists of k laminar cuts $\{A_{i_1}, B_{i_1}\}, \ldots, \{A_{i_k}, B_{i_k}\}$, whose pseudolines cut G into $k+1$ regions $R_0^+, R_1^+, \ldots, R_k^+$ numbered in order of their creation. Since every R_j^+ is the intersection of halfplanes containing this region, R_j^+ is a convex, and therefore gated, subgraph of G.

As we noticed already, the two-connected components of the regions $R_0^+, R_1^+, \ldots, R_k^+$ are not always rectilinear cells, they may contain a certain number of inner or boundary vertices of degree 5. These vertices are kept in the lists $D(R_j^+)$, $j = 0, \ldots, k$. To partition every two-connected component of a pseudorectilinear cell R_j^+ into rectilinear cells, we construct a collection $\mathscr{C}(R_j^+)$ of laminar convex cuts of R_j^+ such that for each vertex v of degree 5 there is a cut from $\mathscr{C}(R_j^+)$ which includes an edge incident to v. Notice that every cut of $\mathscr{C}(R_j^+)$ extends in a unique way to a cut from $\mathscr{L}$.

From the sweeping algorithm we know that when the inner vertex v is added to the list $D(R_j^+)$, the current sweep line L does not contain other vertices of degree $\geqslant 5$.

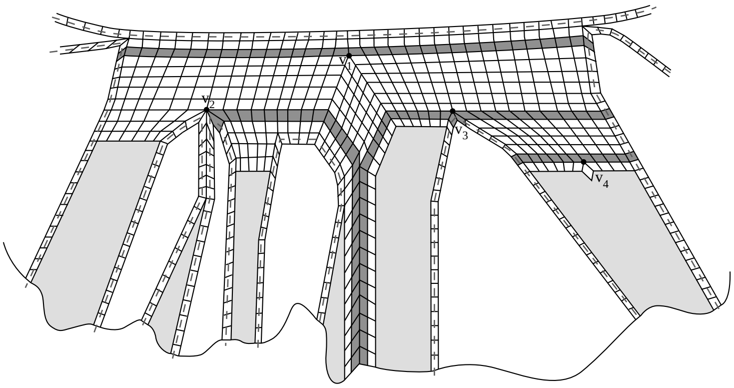


Fig. 28.

Denote by L' the sweep line considered before L and such that every vertex of L has a unique neighbour in L'. Let v' be the neighbour of v in L' and suppose that $v'v \in E_{i_l}$. Add the convex cut $\{A_{i_l} \cap R_j^+, B_{i_l} \cap R_j^+\}$ of R_j^+ to $\mathscr{C}(R_j^+)$; for an illustration see the vertices v_1 and v_4 from Fig. 28. Set $A_{jl} := A_{i_l} \cap R_j^+, B_{jl} := B_{i_l} \cap R_j^+$. Now, suppose that $v \in D(R_j^+)$ lies on the boundary of R_j^+. In this case, we consider one or two edges incident to v whose other end-vertices lie on forthcoming sweep lines (see the vertices v_2 and v_3 from Fig. 28 for two occurring cases). Then add the cut(s) defined by this edge (or these edges) to $\mathscr{C}(R_j^+)$. Pick another vertex w of degree 5 and let $\{A_{jl}, B_{jl}\}$ be the cut defined by the edge $w'w$, where w' is defined in the same way as v'. These two cuts are necessarily laminar in R_j^+, otherwise we obtain a subregion of R_j^+ with at most three corners, contrary to Properties 2 and 3. Thus $\mathscr{C}(R_j^+) = \{\{A_{j1}, B_{j1}\}, \ldots, \{A_{jk_j}, B_{jk_j}\}\}$ is indeed a collection of laminar cuts whose pseudolines partition the cell R_j^+ into the rectilinear cells $SR_{j0}^+, \ldots, SR_{jk_j}^+$. Denote by $T(\mathscr{C}(R^+))$ the tree whose nodes are these rectilinear cells and two nodes are adjacent iff each of the corresponding cells contains a border line of the same cut from $\mathscr{C}(R_j^+)$. Removing the lth edge from this tree, we will get two subtrees whose nodes are the cells contained in A_{jl} and B_{jl}, respectively.

Each of the rectilinear cells $SR_{j0}^+, \ldots, SR_{jk_j}^+$ can be represented as the intersection of R_j^+ with some halfplanes. Since R_j^+ and the halfplanes are gated, every SR_{jl}^+ $(l=0, \ldots, k_j)$ is a gated subgraph as well.

4.4. Vertex labelling

First we reserve contiguous intervals of labels to each of the pseudorectilinear cells $R_0^+, R_1^+, \ldots, R_k^+$. Due to the implementation of the queue $\mathbf{Q}$, the numbering of these regions corresponds to a Breadth First Search traversal of the tree $T(\mathscr{C})$. In particular, R_0^+ is the root of $T(\mathscr{C})$. According to the algorithm from Section 4.1, R_0^+ is obtained from a halfplane of G, say A_{i_1}, by cutting off from A_{i_1} some disjoint halfplanes $B_{i_2}, \ldots, B_{i_s}$.

Notice that each of these halfplanes contains the cells which correspond to nodes lying in the same subtree of $T(\mathscr{C})$ obtained after removing the root R_0^+. To R_0^+ we reserve the labels from the interval $[1\dots|R_0^+|]$. Now, we traverse the boundary of R_0^+ in counterclockwise order starting from B_{i_1} and reserve contiguous cyclic subintervals of the interval $[|R_0^+|+1\dots n]$ to each of the halfplanes B_{i_j}, $j=1,\dots,s$. Assume without loss of generality that $B_{i_1}, B_{i_2}, \dots, B_{i_s}$ is the order in which we meet the border lines of the complements of these halfplanes. To B_{j_1} we assign the interval $[|R_0^+|+1\dots|R_0^+|+|B_{i_1}|]$ and to B_{i_j} $(j=2,\dots,s)$ we assign the interval $[|R_0^+|+\sum_{l=1}^{j-1}|B_{i_l}|+1\dots|R_0^+|+\sum_{l=1}^{j}|B_{i_l}|]$. The cyclic interval assigned to each of the halfplanes $B_{i_j}, j=1,\dots,s$, is further subdivided into smaller contiguous intervals using the same procedure: the interval with smallest available labels is reserved to the cell R_j^+ obtained at the end of the phase on which the halfplane B_{i_j} is sweeped, while the remaining labels are distributed in intervals among the halfplanes which are cut off from B_{i_j} during that phase (the order of affecting intervals is the same as for the root cell, i.e., by traversing in counterclockwise way the beach line of R_j^+). Notice the following elementary but basic property of the resulting assignment.

Lemma 3. *Let x, y be two arbitrary vertices of ∂R_j^+. Let P be one of the subpaths of ∂R_j^+ comprised between x and y, and consider all halfplanes disjoint from ∂R_j^+ such that the border lines of their complements are completely contained in P. Then the union of labels of these halfplanes occupies one or two linear subintervals of* $[1\dots n]$.

At the end of this first part of the labelling algorithm, to each cell R_j^+ $(j=0,\dots,k)$ is assigned one circular interval $\mathscr{L}(R_j^+)$, such that $\mathscr{L}(R_0^+),\dots,\mathscr{L}(R_k^+)$ form a partition of $[1\dots n]$. Moreover, the unions of labels of cells in each of two subtrees obtained by removing an arbitrary edge of $T(\mathscr{C})$ constitute two complementary subintervals of the cyclic interval $[1\dots n]$. If R_j^+ is a rectilinear cell (i.e., every inner vertex has degree 4 and every boundary vertex has degree at most 4), then applying the algorithm from Section 3.1 we label the vertices v of R_j^+ with numbers $\mathscr{L}(v)$ from the interval $\mathscr{L}(R_j^+)$.

Now, suppose that R_j^+ is a pseudorectilinear cell. In order to label its vertices, first we assign disjoint subintervals of $\mathscr{L}(R_j^+)$ to the rectilinear cells $SR_{j0}^+,\dots,SR_{jk_j}^+$ into which R_j^+ is partitioned by the pseudolines of the cuts from $\mathscr{C}(R_j^+)$. This can be done by using the usual procedure: we reserve a sufficient amount of least available labels to the first subcell SR_{j0}^+, and the rest of labels is distributed in intervals among the halfplanes of cuts from $\mathscr{C}(R_j^+)$ which are disjoint from SR_{j0}^+ and such that the border lines of their complements are contained in $\partial(SR_{j0}^+)$. Again the order of labelling of these halfplanes follows the counterclockwise traversal of $\partial(SR_{j0}^+)$. These intervals are further subdivided if new regions bounded by cuts from $\mathscr{C}(R_j^+)$ are found. As a result, we assign to each SR_{jl}^+ $(l=0,\dots,k_j)$ a single subinterval $\mathscr{L}(SR_{jl}^+)$, which altogether partition the interval $\mathscr{L}(R_j^+)$. Since each SR_{jl}^+ is a rectilinear cell, the final labelling of the vertices from SR_{jl}^+ is done by the algorithm from Section 3.1.

One can note that the labelling of the vertices of a quadrangulation $G\in\mathscr{Q}_4$ is performed via a three-level treelike structure. The tree $T(\mathscr{C})$ whose nodes are the cells

$R_0^+, \ldots, R_k^+$ is the main tree. For each node R_j^+ we construct the second-level tree $T(\mathscr{C}(R_j^+))$ whose nodes are the rectilinear cells SR_{jl}^+, $l = 0, \ldots, k_j$. The vertices of each SR_{jl} are further grouped into horizontal paths, and the incidence relation between these paths defines the tree $T^h(SR_{jl}^+)$, the third-level tree.

4.5. Edge labelling and routing

In this final part of Section 4, we explain how to label the oriented edges of G. From this algorithm we will immediately establish that the resulting routing scheme is optimal, i.e., that the messages from a source u to a destination x are rooted via a shortest (u,x)-path of G. Actually, we will see that our routing scheme obeys the following equivalent condition: if v is the neighbour of u such that $\mathscr{L}(x) \in \mathscr{I}(u,v)$, then v lies on a shortest path between u and x.

Pick an arbitrary vertex u of G, say $u \in R_j^+$. Assume additionally that u belongs to the rectilinear cell SR_{jl}^+. Next we will show how to label the outgoing edges (u,v). For this, divide the label $\mathscr{I}(u,v)$ into three groups

$$\mathscr{I}(u,v) = \mathscr{I}_1(u,v) \cup \mathscr{I}_2(u,v) \cup \mathscr{I}_3(u,v)$$

(as we will see immediately, one or two groups maybe empty). If both u and v belong to a common rectilinear cell SR_{jl}^+, then $\mathscr{I}_1(u,v)$ consists of the labels of vertices $x \in SR_{jl}^+$ such that the messages from u to x will be routed via v, otherwise $\mathscr{I}_1(u,v)$ is empty. If $v \in R_j^+$, then $\mathscr{I}_2(u,v)$ is the union of labels of all vertices $x \in R_j^+ \setminus SR_{jl}^+$ such that the messages from u to x will be routed via v ($\mathscr{I}_2(u,v)$ is empty if $v \notin R_j^+$). Finally, $\mathscr{I}_3(u,v)$ consists of the labels of all vertices of $x \in G \setminus R_j^+$ such that the messages from u to x are routed via v.

The label $\mathscr{I}_1(u,v)$ is computed using the algorithm from Section 3 (in fact, the algorithm provides a labelling of all vertices and oriented edges of the rectilinear cell SR_{jl}^+). Hence each $\mathscr{I}_1(u,v)$ occupies one or two cyclic subintervals of the interval $\mathscr{L}(SR_{jl}^+)$. In the global routing scheme we are allowed to use only subintervals of $[1 \ldots n]$, thus we transform two cyclic subintervals of $\mathscr{L}(SR_{jl}^+)$ into at most three linear subintervals of $[1 \ldots n]$. Since SR_{jl}^+ is convex and routing in each SR_{jl}^+ is done as in the rectilinear cells, we immediately deduce that the labels from the first group $\mathscr{I}_1$ assure the optimal routing inside each rectilinear cell R_{jl}^+.

Now assume that $uv \in E_{i_l}$ for a cut $\{A_{i_l}, B_{i_l}\} \in \mathscr{C}$. Suppose without loss of generality that $u \in R_{i_j}^+ \subseteq A_{i_l}$ and $v \in B_{i_l}$. Set $\mathscr{I}_3(u,v) = \mathscr{L}(B_{i_l})$ and $\mathscr{I}_3(v,u) = \mathscr{L}(A_{i_l})$. From Section 4.4 we know that the labels assigned to B_{i_l} and A_{i_l} constitute two complementary circular subintervals of $[1 \ldots n]$. Since v is closer than u from every vertex $x \in B_{i_l}$, we obtain the required property of routing.

It remains to deal with outgoing edges (u,v) which belong to the pseudorectilinear cell R_j^+, more precisely, to a two-connected component of this cell (one can assume without loss of generality that R_j^+ itself is two-connected). Assume that R_j^+ was constructed in the phase during which the halfplane A_{i_j} was swept. Suppose that at this phase the cuts $\mathscr{C}_j = \{\{A_{i_l}, B_{i_l}\}, \{A_{i_{l+1}}, B_{i_{l+1}}\}, \ldots, \{A_{i_{l+s}}, B_{i_{l+s}}\}\}$ have been added to $\mathscr{C}$, where $B_{i_l}, B_{i_{l+1}}, \ldots, B_{i_{l+s}}$ are disjoint from R_j^+. This means that $R_j^+ = A_{i_j} \setminus (B_{i_l} \cup B_{i_{l+1}} \cdots \cup B_{i_{l+s}})$. The numbering of the halfplanes $B_{i_j}, B_{i_l}, B_{i_{l+1}}, \ldots, B_{i_{l+s}}$ is the same as in Section 4.4, i.e., following the counterclockwise traversal of ∂R_j^+. From the sweeping algorithm of Section 4.1 we know that $\{A_{i_j}, B_{i_j}\} \in \mathscr{C}$, hence $\mathscr{L}(A_{i_j})$ and $\mathscr{L}(B_{i_j})$ form two complementary subintervals of the cyclic interval $[1 \ldots n]$.

The vertex u has at most five neighbours in R_j^+, say v_1, v_2, v_3, v_4, v_5. Consider the equivalence classes E_p of Θ defined by the edges uv_p, $p \leqslant 5$. Since the region R_j^+ is convex, every such class E_p shares precisely two edges with ∂R_j^+, the leftmost edge le_p and the rightmost edge re_p. These are either two edges of ∂G, or one edge of ∂G and one edge of a border line of a cut from $\mathscr{C}_j$, or two edges from the border lines of two distinct cuts of $\mathscr{C}_j$. Hence $\{A_p, B_p\}$ is laminar with all cuts of $\mathscr{C}_j$ except one or two cuts. If u is an inner vertex of degree 4, the four cuts defined by the edges incident to u divide R_j^+ into eight bigons. Similarly, if u has degree 5, we will obtain ten bigons (the case $u \in \partial R_j^+$ is similar, even easier). Consequently, the boundary of the region R_j^+ will be partitioned into eight or ten paths. Since the cuts defined by two edges uv_p and uv_q lying in a common 4-face are transversal and $\mathscr{X}$ does not contain three pairwise transversal cuts, E_p and E_q cannot share common edges with the border line of the same halfplane. To $\mathscr{I}_3(u, v_1)$ we assign the labels of all halfplanes $B_{i_{l+q}}$ such that $bd(A_{i_{l+q}})$ is contained entirely in the subpath of ∂R_j^+ comprised between the edges le_1 and re_1. If $le_1 \in \partial G$, to $\mathscr{I}_3(u, v_2)$ we assign the labels of all halfplanes $B_{i_{l+p}}$ such that $bd(A_{i_{l+p}})$ is contained entirely in the subpath of ∂R_j^+ comprised between le_1 and re_2. Otherwise, if $le_1 \in bd(A_{i_{l+q}})$, then assign to $\mathscr{I}_3(u, v_2)$ the label of the halfplane $B_{i_{l+q}}$ plus the labels of all halfplanes $B_{i_{l+p}}$ such that $bd(A_{i_{l+p}})$ is contained entirely in the subpath of ∂R_j^+ comprised between le_1 and re_2. Using the same method, we distribute labels to the rest of outgoing arcs (see Fig. 29). From Lemma 3, we conclude that each $\mathscr{I}_3(u, v_p)$ occupies one or two linear subintervals of $[1 \ldots n]$. Notice also that if $\mathscr{L}(x) \in \mathscr{I}_3(u, v_p)$, then v_p is closer to x than u, because $x \in V(v_p, u) = B_p$.

In a similar way, we specify the group $\mathscr{I}_2$. Suppose that R_j^+ was partitioned into the rectilinear cells $SR_{j0}^+, \ldots, SR_{jk_j}^+$ using the collection of laminar cuts $\mathscr{C}(R_j^+)$. Let $\{A_{j1} B_{j1}\}, \ldots, \{A_{jk_j}, B_{jr_j}\}$ be the cuts from $\mathscr{C}(R_j^+)$ such that $bd(A_{j1}) \cup \ldots bd(A_{jk_j}) \subseteq \partial(SR_{jl}^+)$. If uv participates in one of these cuts, then set $\mathscr{I}_2(u, v) = \mathscr{L}(B_{jl})$ and $\mathscr{I}_2(v, u) = \mathscr{L}(A_{jl})$. Each of these labels occupies a single cyclic subinterval of $\mathscr{L}(R_j^+)$, therefore one or two linear subintervals of $[1 \ldots n]$. It remains to define $\mathscr{I}_2(u, v_p)$ for neighbours v_p of u inside the rectilinear cell SR_{jl}^+. If u is an inner vertex of SR_{jl}^+ then it has four neighbours v_1, v_2, v_3, v_4 and the labelling is performed similarly to that from Fig. 29. The case when $u \in \partial(SR_{jl}^+)$ is completely similar. Each $\mathscr{I}_2(u, v_p)$ occupies a single circular subinterval of $\mathscr{L}(R_j^+)$, therefore two linear subintervals of $[1 \ldots n]$. Since R_j^+ is convex, this shows that routing messages between two vertices in different rectilinear subcells of R_j^+ can be done using at most two intervals per edge (from the group $\mathscr{I}_2$).

Note that in order to route the message from u to the destination outside the pseudorectilinear cell R_j^+ or outside the rectilinear cell SR_{jl}^+, we send the message to a neighbour of u which has the same gate as u in the halfplane of G or of R_j^+. Consequently, the message from u will be sent to its gate in the respective halfplane. This establishes the optimality of the routing scheme.

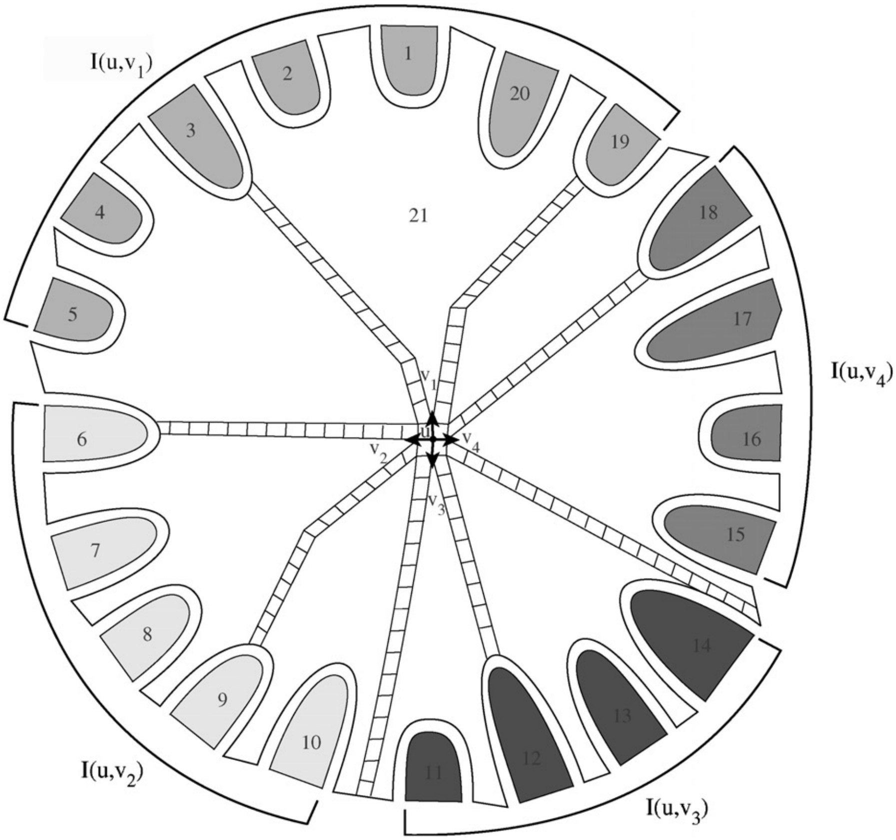


Fig. 29.

Summarizing, here is the main result of this paper.

Theorem 2. *For a quadrangulation* $G \in \mathcal{Q}_4$, *the described routing scheme* $\mathcal{R} = (\mathcal{L}, \mathcal{I})$ *is an optimal* 7-*SLIRS*.

Notice that only a few edges of G will be labelled with seven linear intervals, the most edges will have labels consisting of five or less intervals.

5. Some further results

In this section, we adjust the routing schemes from Sections 3 and 4 to plane triangulations and hexagonal systems.

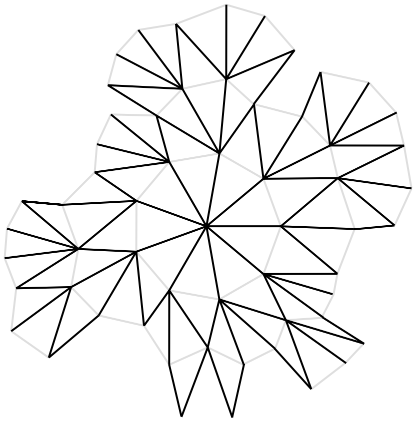


Fig. 30.

5.1. Triangulations from $\mathscr{T}_6$

Metric properties of triangulations $G=(V,E)$ from $\mathscr{T}_6$ have been investigated in several papers, see for example [4] and the papers cited therein. We will need two simple properties of such graphs established before. Let b be a fixed vertex of G. We call an edge uv *horizontal* if $d_G(b,u)=d_G(b,v)$ and *vertical* if $d_G(b,u)\neq d_G(b,v)$.

Property 4. *If uv is a horizontal edge of G, then there exists precisely one common neighbour x of u and v such that $d_G(b,x)<d_G(b,u)=d_G(b,v)$.*

Property 5. *G does not contain three pairwise adjacent vertices which are equidistant to b.*

From this property we immediately conclude that every inner face of G contains exactly one horizontal edge. Denote by E_0 the set of vertical edges of G.

Lemma 4. *The partial subgraph $G_0=(V,E_0)$ of G is a quadrangulation from $\mathscr{Q}_4$. Moreover, G_0 is a 2-spanner of G, i.e. $d_{G_0}(u,v)\leqslant 2d_G(u,v)$ holds for arbitrary vertices u and v.*

Proof. Suppose that the plane embedding of G_0 is obtained from that of G by removing all horizontal edges. Pick a horizontal edge uv of G. Then uv belongs either to one or to two triangles (inner faces) of G. By Property 4, two other edges of those triangles are vertical. This immediately implies that G_0 is a quadrangulation. Now, we will show that every inner vertex w of G_0 has at least four neighbours in G_0. Assume by way of contradiction that w has only three neighbours u_1,u_3,u_3 in G_0. Let v_1,v_2, and v_3 be the second common neighbours in G_0 of the pairs $\{u_1,u_2\},\{u_2,u_3\}$, and $\{u_3,u_1\}$, respectively. Therefore, (w,u_1,v_1,u_2), (w,u_2,v_2,u_3), and (w,u_3,v_3,u_1) are the inner faces of G_0 sharing the vertex w. Since w belongs to the region R bounded by the 6-cycle $C=(u_1,v_1,u_2,v_2,u_3,v_3)$ of G, all neighbours of w in G are located inside or on the boundary of R. But inside R we do not have other vertices, because R is the union of all 4-faces incident to w. Consequently, since w has degree $\geqslant 6$ in G, it must be adjacent in G to all vertices of the cycle C. Thus, wv_1, wv_2, and wv_3 are horizontal edges of G. Let $k=d_G(b,v_1)=d_G(b,v_2)=d_G(b,v_3)=d_G(b,w)$. Applying Property 4 to each of the horizontal edges wv_1, wv_2, and wv_3 we conclude that at least two of the vertices u_1,u_2,u_3, say the first two, are at distance $k-1$ to b. This, however, contradicts the uniqueness of the vertex from Property 4: for the edge wv_1, both vertices u_1 and u_2 are closer to b. This contradiction establishes that indeed $G_0 \in \mathcal{Q}_4$.

It remains to show that G_0 is a 2-spanner of G. Indeed, pick two arbitrary vertices u,v and a shortest path P of G connecting these vertices. We will transform P into a (u,v)-path P_0 whose length is at most twice the length of P. Every vertical edge of P belongs also to P_0. Now, if xy is a horizontal edge of P, then by Property 4 there is a common neighbour z of x and y which is closer to the base-point b. Hence xz and yz are vertical edges of G, thus we may replace xy by xz and zy. This concludes the proof. □

From this lemma and the results of Section 4 we obtain the following consequence.

Corollary 1. *For a triangulation $G \in \mathcal{T}_6$, the routing scheme $\mathcal{R}=(\mathcal{L},\mathcal{I})$ for G_0 is a 7-SLIRS with stretch factor 2 for G.*

One can easily construct examples of such triangulations in which for certain pairs of vertices there is a unique shortest path and this path consists of horizontal edges only. Therefore, the stretch factor 2 is tight here.

5.2. Hexagonal systems

Hexagonal systems are quite similar with rectilinear cells. If for rectilinear cells we have two types of edges (horizontal and vertical), hexagonal systems have three types of parallel edges, one of them is the class of vertical edges. Notice that if one contracts all vertical edges of a hexagonal system H, we will get a rotated rectilinear cell. If, instead, we remove the vertical edges from H, again the edges from two remaining classes will be grouped into paths (to some extent, we call them also h-paths). The incidence relation between these paths give raise to a tree $T^h(H)$. Furthermore, one can also define the notions of pockets and segments. The routing scheme $\mathcal{R}=(\mathcal{L},\mathcal{I})$ for H is quite similar to that for rectilinear cells. The bijection $\mathcal{L}: V \to [1\ldots n]$ and the labels of vertical edges are the same. The labelling of an edge uv on a h-path hp_i is slightly different. Suppose u is left from v. The edge uv belongs to the supports of one or two pockets P' and P'', say P' is above hp_i and P'' is below hp_i (one of these pockets maybe absent). Up to symmetry one can assume that u is adjacent to a vertex of P' and v is adjacent to a vertex of P''. Now, in addition to the labels of (u,v) and (v,u) assigned as for rectilinear cells, add to $\mathcal{I}(u,v)$ the labels of the pocket P'' and to $\mathcal{I}(v,u)$ add the labels of the pocket P'. One can easily check that $\mathcal{I}(u,v)$ and

$\mathscr{I}(v,u)$ occupy two circular intervals each. On the other hand, the routing is no longer optimal.

Up to symmetry, it suffices to consider the following case. Suppose that the message sent from a source x arrives at some h-path hp_i and its destination y is a vertex in a pocket P' below hp_i. It will be further sent along the h-path to the closest vertex u of the support of P'. Inside the pocket P', the vertical moves and the oblique moves to the right will alternate until we will come to a vertex v in a new h-path hp_j such that the support of the pocket P'' (with respect to hp_j) containing y will not contain the node v. In this case the message will be sent along hp_j to the closest vertex w of the support of P'', and so one, until it arrives at the destination y. In order to find the stretch factor of this routing scheme $\mathscr{R}$, it suffices to compare $d_R(u,w)$ and $d_G(u,w)$. Indeed, the vertices u and w lie on every shortest (x,y)-path. The worst case is when P'' is located left from v and from the closest to u vertex v' from hp_j. In this case, the shortest way from u to w is to go via v'. The shortest path between u and v' is an alternating path of vertical edges and oblique edges oriented to the left. One can easily show that $d_G(u,v)=d_G(v,v')=d_G(u,v')$, i.e. the region comprised between the $(u,v)-,(v,v')-$, and $(u,v')-$shortest paths is a kind of equilateral triangle formed of regular hexagons. Since $d_R(u,w)=d_G(u,v)+d_G(v,v')+d_G(v',w)$ and $d_G(u,w)=d_G(u,v')+d_G(v',w)$, we conclude that $d_R(u,w)/d_G(u,w)\leqslant 2$. As a consequence, we obtain the following result.

Corollary 2. *For a hexagonal system $G\in\mathscr{H}$ the routing scheme presented above is a 2-SIRS with the stretch factor 2.*

Acknowledgements

While preparing this paper, the authors were supported by travel grants from France Telecom R&D and Bayerisch-Franzoesisches Hochschulzentrum.

References

[1] B. Awerbuch, A. Bar-Noy, N. Linial, D. Peleg, Improved routing strategies with succinct tables, J. Algorithms 11 (1990) 307–341.
[2] E.M. Bakker, J. van Leeuwen, R.B. Tan, Linear interval routing, Algorithms Rev. 2 (1991) 45–61.
[3] H.-J. Bandelt, V. Chepoi, Cellular bipartite graphs, European J. Combin. 17 (1996) 121–134.
[4] H.-J. Bandelt, V. Chepoi, Decomposition and l_1-embedding of weakly median graphs, European J. Combin. 21 (2000) 701–714.
[5] V. Chepoi, Graphs of some CAT(0) complexes, Adv. Appl. Math. 24 (2000) 125–179.
[6] D. Das Sharma, D. Pradhan, Submesh allocation in mesh multicomputers using busy-lists: a best-fit approach with complete recognition capability, J. Parallel and Distributed Comput. 36 (1996) 106–118.
[7] M. Deza, M. Laurent, Geometry of Cuts and Metrics, Springer, Berlin, 1997.
[8] D.Ž. Djoković, Distance-preserving subgraphs of hypercubes, J. Combin. Theory Ser. B 14 (1973) 263–267.
[9] T. Eilam, S. Moran, S. Zaks, The complexity of characterization of networks supporting shortest-path interval routing, in: 4th Colloq. on Structural Information and Communication Complexity (SIROCCO), Carleton University Press, Waterloo, Ontario, 1997, pp. 99–111.

[10] M. Flammini, Deadlock-free interval routing schemes, in: 14th Annu. Symp. on Theoretical Aspects of Computer Science (STACS), Lecture Notes in Computer Science, Vol. 1200, Springer, Berlin, 1997, pp. 351–362.
[11] P. Fraigniaud, C. Gavoille, Interval routing schemes, Algorithmica 21 (1998) 155–182.
[12] P. Fraigniaud, S. Vial, One-to-all and all-to-all communications in partial meshes, Parallel Process. Lett. 9 (1999) 9–20.
[13] G.N. Frederickson, R. Janardan, Designing networks with compact routing tables, Algorithmica 3 (1988) 171–190.
[14] G.N. Frederickson, R. Janardan, Efficient message routing in planar networks, SIAM J. Comput. 19 (1990) 164–181.
[15] C. Gavoille, A survey on interval routing, Theoret. Comput. Sci. 245 (2000) 217–253.
[16] C. Gavoille, S. Pérennès, Lower bounds for interval routing on 3-regular networks, in: N. Santoro, P. Spirakis (Eds.), 3rd Internat. Colloq. on Structural Information and Communication Complexity (SIROCCO) Carleton University Press, Waterloo, Ontario, 1996, pp. 88–103.
[17] J. van Leeuwen, R.B. Tan, Computer networks with compact routing tables, in: The Book of L, Springer, Berlin, 1986, pp. 298–307.
[18] J. van Leeuwen, R.B. Tan, Interval routing, Comput. J. 30 (1987) 298–307.
[19] R.S. Millman, G.D. Parker, Geometry. A metric approach with models, Springer, Berlin, 1991.
[20] L. Narayanan, N. Nishimura, Interval routing on k-trees, J. Algorithms 26 (1998) 325–369.
[21] L. Narayanan, S. Shende, Partial characterizations of networks supporting shortest path interval labelling schemes, Networks 32 (1998) 103–113.
[22] D. Peleg, E. Upfal, A trade-off between space and efficiency for routing tables, J. ACM 36 (1989) 510–530.
[23] N. Santoro, R. Khatib, Labeling and implicit routing in networks, Comput. J. 30 (1987) 298–307.
[24] S.S.H. Tse, F.C.M. Lau, Some lower-bound results on interval routing in planar graphs, Technical Report TR-97-05, Department of Computer Science, The University of Hong Kong, April 1997.
[25] M.L.J. van de Vel, Theory of Convex Structures, North-Holland, Amsterdam, 1993.