Discrete Applied Mathematics 40 (1992) 363-378
North-Holland

Algorithms for routing around a rectangle

András Frank

Computer Science Department, Eötvös University, Budapest, Hungary

Takao Nishizeki*, Nobuji Saito* and Hitoshi Suzuki*

Department of Electrical Communications, Faculty of Engineering, Tohoku University, Sendai 980, Japan

Éva Tardos**

School of Operations Research, Cornell University, Ithaca, NY, USA

Received 15 December 1989
Revised 1 January 1991

Abstract

Frank, A., T. Nishizeki, N. Saito, H. Suzuki and É. Tardos, Algorithms for routing around a rectangle, Discrete Applied Mathematics 40 (1992) 363-378.

Simple efficient algorithms are given for three routing problems around a rectangle. The algorithms find routing in two or three layers for two-terminal nets specified on the sides of a rectangle. All algorithms run in linear time.

One of the three routing problems is the minimum area routing previously considered by LaPaugh and Gonzalez and Lee. The algorithms they developed run in time $O(n^3)$ and $O(n)$ respectively. Our simple linear time algorithm is based on a theorem of Okamura and Seymour and on a data structure developed by Suzuki, Ishiguro and Nishizeki.

1. Introduction

In this paper we give efficient algorithms for three routing problems around a rectangle. The minimum area routing problem in Section 5 has been considered previously by LaPaugh [6] and Gonzalez and Lee [4,5]. Our simple algorithm for this problem can serve as useful building block in practical algorithms when routing

Correspondence to: Professor É. Tardos, School of Operations Research, Cornell University, Ithaca, NY 14853, USA.

in the assigned area is not possible, and one has to minimally enlarge the area so as to make the routing possible.

The routing region of our problem is the part of the plane grid between a connected boundary O and a rectangular hole I inside O. The pairs of terminals to be connected are on the boundary of the hole I, every node on I is the terminal of at most one pair, and none of the four corners are terminals. In our routing the paths connecting the terminals are pairwise edge-disjoint and are wired in two or three layers.

The paper is organized as follows. In Section 2 we consider an edge-disjoint path problem on a capacitated cycle. This will be the basis of all three routing algorithms.

Next we present an algorithm that finds two layer routing using the knock-knee free model in the case when O is also a rectangle. The algorithm runs in time linear in the number of terminals if the order in which they appear around I is known in advance.

The second routing algorithm finds a routing using three layers in a given region with a nonrectangular outside boundary O. The time complexity is linear in the perimeter of O.

In Section 5 we give a linear time algorithm to find a minimum area rectangle O surrounding I such that there is a routing using two layers in the grid graph defined by I and O. The algorithm is based on ideas from Section 3. This problem has been studied previously by LaPaugh [6] and Gonzalez and Lee [4,5]. Our algorithm is considerably simpler than either of these algorithms, and runs as well as one in [5], in time linear in the number of terminals if their order is given around I. The area of the minimum outer rectangle is expressed explicitly in terms of the "density of terminals".

Finally we consider the problem of finding all of the feasible positions of the rectangle I inside a given rectangle O. We give a linear time algorithm that finds all such positions.

Basic ingredients of our algorithms are a theorem of Okamura and Seymour [7] and the data structure variable-priority queue developed in [8]. Preliminary versions of this paper appeared independently as extended abstracts [2] and [9].

2. Routing around a cycle

In this section we define and solve the *edge-disjoint path problem in cycle networks* $N=(G,P)$, where $G=(V,E)$ is a cycle, with nonnegative integer capacities $c(e)$ on its edges, the arcs of the cycle, and $P=\{(s_i,t_i)\colon i=1,\dots,k\}$ is a set of k pairs of terminals. The problem is to find paths in G connecting the pairs of terminals in P such that the number of paths using an edge e is at most its capacity $c(e)$.

Okamura and Seymour [7] considered the edge-disjoint path problem for a more general class of graphs. A *planar network* consists of a planar graph G given with a planar embedding and pairs of terminals P such that every terminal is on the outer face. A *node* $v \in V$ is *even* if the sum of the capacities of the edges adjacent to v plus the number of pairs in P with v as one terminal is even, and *odd* otherwise. A *network* N is *even* if every node is even. A *cut* $A \subseteq E$ is a minimal set of edges disconnecting G. The *demand* $d(A)$ of a cut is the number of terminal pairs separated by the cut. The *capacity* $c(A)$ of the cut is the sum of the capacities of the edges in the cut. The following condition is trivially necessary for the required paths to exist:

Cut criterion. $d(A) \leq c(A)$ for every cut A.

Okamura and Seymour [7] proved the following theorem.

Theorem 2.1. *For an even planar network N the required paths exist if and only if the cut criterion is satisfied.*

In this section we consider cycle networks. A cycle is a particular planar graph where all nodes are on the outer boundary. However, we shall not assume that the network is even. In [1] a more general theorem is developed characterizing the existence of edge-disjoint paths if evenness is only required for nodes *not* on the outer face. A polynomial time algorithm that finds the required paths in a cycle network follows from the results in [1]. However, the special case when the graph is a cycle is considerably easier. Here we give an $O(n)$ algorithm for the special case.

First note that the cut criterion is not sufficient. To show this consider the cycle network, consisting of a cycle of length 4 with all edges having capacity 1. Let the two pairs of terminals be the opposite pairs of nodes on the cycle. This network satisfies the cut criterion, but the required paths do not exist.

We shall introduce another necessary criterion, the *parity criterion*. A cut is *tight* if the inequality in the cut criterion is satisfied as equality. The edges in a tight cut must be used up to their capacity in every solution. A cut A is *odd* if $c(A) - d(A)$ is odd, and *even* otherwise. A simple parity argument shows that in any solution and in every odd cut A there must be at least one edge e such that the number of paths through e is strictly less than $c(e)$. Note that cuts of the cycle consist of pairs of edges. The above observations imply the necessity of the parity criterion.

Parity criterion. If each of the edges e and f is in some tight cut, then the cut $\{e, f\}$ is even.

Theorem 2.2. *The edge-disjoint path problem in a cycle network has a solution if and only if both the cut and the parity criterion are satisfied.*

Proof. The necessity has been indicated above. Here we prove the sufficiency. If the cycle network is even the statement follows from the Okamura–Seymour theorem. Otherwise let T denote the set of odd nodes. Observe that the number of odd nodes is even. Let $\{e_1, e_2\}$ denote a tight cut separating some of the nodes in T, if there is any. We shall refer to the nodes in T as $\{v_1, \ldots, v_{2l}\}$ in the order as they appear clockwise around the cycle with v_1 being the first after e_1 if there is a tight cut. Add the following additional pairs of terminals to the problem: (v_1, v_2), $(v_3, v_4), \ldots, (v_{2l-1}, v_{2l})$. The resulting network augmented is even by definition, therefore the Okamura–Seymour theorem applies. We claim that it satisfies the cut criterion and hence the required paths exist.

The augmented network is even. In an even network $c(A) - d(A)$ is even for every cut A. On the other hand the difference $c(A) - d(A)$ has decreased by at most 2 by augmenting the network with the new pairs of terminals. This implies the cut criterion in the augmented network provided that in the original network no tight cuts separate the nodes of T.

Next assume that there is a tight cut $\{e_1, e_2\}$. We proceed by contradiction. Let $\{f_1, f_2\}$ denote a cut violating the cut criterion in the augmented network. The difference $c(A) - d(A)$ is even, therefore it is at most -2. This implies that $\{f_1, f_2\}$ is a tight cut in the original network, and it must be crossed by two of the new pairs of terminals. In this case all of the four cuts of the form $\{e_i, f_j\}$ for $i, j = 1, 2$ are odd, contradicting the parity criterion. □

The above proof is based on the Okamura–Seymour theorem. Using the proof of that theorem this proof can be converted into an $O(n^2)$ time algorithm. In [8] the variable-priority queue data structure is used to find a solution in $O(n+k)$ time.

3. Routing on two layers

A *grid graph* is a subgraph of the integer grid in the plane. For easy reference we shall assume that the grid lines are horizontal and vertical, and the grid nodes are

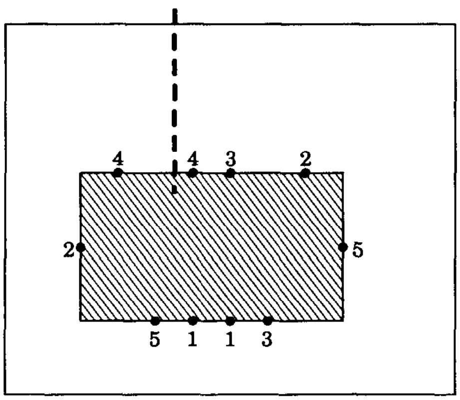

Fig. 1. The semicut associated with an edge.

the integer points on the plane. In this section we shall consider routing problems on grid graphs consisting of the part of the grid between a rectangular boundary O and a rectangular hole I inside O. The network $N=(G,P)$ consists of such a grid graph G and k pairs of terminals P on the inside rectangle I. We assume that every node on I is the terminal of at most one pair and the four corners of I are not terminals.

Routing of a network N is defined as follows. A *conducting layer* is a graph isomorphic to the grid graph G. Assume that $l \geq 2$ layers $L_1, L_2, \ldots, L_l$ are available. These layers are stacked on top of each other. A *routing* of a network $N=(G,P)$ is a set of edge-disjoint paths $\{R_1, R_2, \ldots, R_k\}$ in G connecting the pairs of terminals, and the assignment of each edge of the paths R_i for $1 \leq i \leq k$ to a layer. The paths are node-disjoint on each layer. If R_i changes from layer L_g to layer L_h at a node v, then a via is established between layers L_g and L_h at node v. If a via connects layers L_g and L_h with $(g<h)$ at a node v, then no other path can use node v on any of the layers L_j for $g \leq j \leq h$.

A set of edge-disjoint paths in a grid graph is *knock-knee free* if whenever two paths share a node v one of them passes through v vertically and the other one horizontally, i.e., no two paths bend at the same node. Note that knock-knee free edge-disjoint paths in a grid graph can be turned into two layer routing by assigning the vertical and horizontal edges on the paths to the two layers respectively.

In this section we solve the two layer routing problem defined above by finding knock-knee free edge-disjoint paths in the grid graph. The problem is reduced to an edge-disjoint path problem in a cycle network.

Given a routing problem with inner rectangle I and outer rectangle O define an associated cycle network $N_{\mathrm{cy}} = (G_{\mathrm{cy}}, P)$. The nodes of G_{cy} are the grid points on the inner boundary of I, and the edges of G_{cy} are the grid edges between nodes along I, denoted by $E(I)$. For an edge $e \in E(I)$ define the associated *semicut* as the set of edges intersecting the line segment perpendicular to e that connects e and some edge on O and does not intersect any other edge in I (see Fig. 1). Let the capacity of an edge e in N_{cy} be the number of edges in the associated semicut. Note that G_{cy} can be obtained from the grid graph by first contracting all edges that do not participate in semicuts, and then replacing an edge with l parallel copies by a single edge with capacity l.

The following theorem, due to Gallai [3], will be useful in converting solutions for the cycle network problem to knock-knee free edge-disjoint paths in the grid graph G.

Theorem 3.1 [3]. *Given a set of closed intervals $\mathscr{F} = \{(a_i, b_i): i=1, \ldots, s\}$ on a line, $\mathscr{F}$ can be partitioned into m sets of disjoint intervals $\mathscr{F}_i$ if and only if no $m+1$ intervals share a point.*

Proof. The theorem can be proved by constructing the partition by a greedy algorithm. Order the intervals according to the left end points. Consider the intervals in this order and add each of them to one of the sets $\mathcal{F}_i$ containing no interval intersecting it. This procedure will produce m sets of disjoint intervals unless for some interval (a_i, b_i) none of the sets constructed so far is disjoint from (a_i, b_i). By the order in which intervals are considered this implies that all of the m sets have an interval containing the point a_i, contradicting to the assumption that no point is contained in more than m intervals. □

This procedure can easily be implemented in O(s) time if the order in which the end points of the intervals appear on the line is known in advance.

Theorem 3.2 *There exists knock-knee free edge-disjoint paths connecting terminal pairs around a rectangle I in a grid graph defined by I and a surrounding rectangle O, if and only if there are edge-disjoint paths connecting the terminals in the associated cycle network N_{cy}.*

Proof. Given a set of edge-disjoint paths in the grid graph a solution to the cycle network problem can be obtained by contracting all edges that do not participate in semicuts. This proves the *only if* direction.

Next we prove the *if* direction. Consider a solution of the edge-disjoint cycle problem. Assume, without loss of generality, that the paths are simple. Let I_j for $j = 1, \ldots, 4$ denote the four sides of the rectangle I (two horizontal and two vertical) with the corners included in both sides. Let $E(I_j)$ denote the set of edges in the cycle network connecting nodes in I_j. Break each path in the solution of the cycle network problem into intervals parallel to the four sides (see Fig. 2). A set of paths defines four sets of intervals. The end points of these intervals are the terminals and the four corners of the rectangle I.

For a side I_j of the rectangle I, let h_j denote the number of grid lines parallel to I_j on the corresponding side of I, or equivalently the number of edges in the

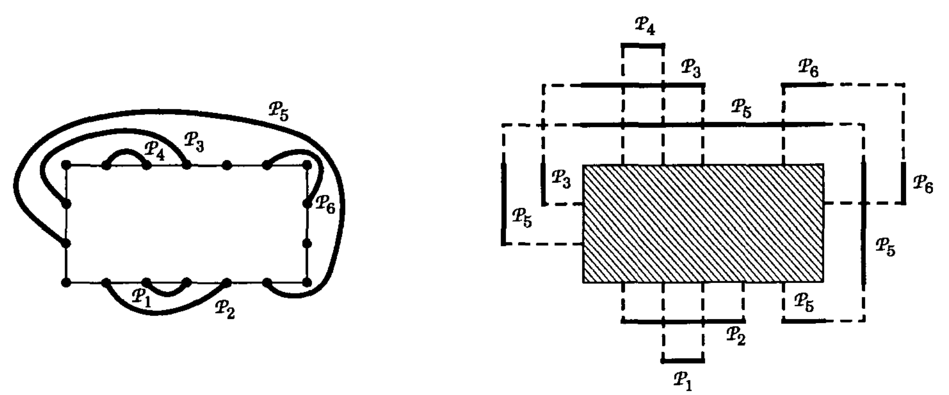

Fig. 2. Paths connecting corresponding terminals in the cycle network, the associated sets of intervals with the knock-knee free routing.

semicuts associated with an edge in $E(I_j)$. An edge e participates in exactly as many intervals as the number of paths through e. Hence the interval system associated with side I_j will not contain any edge in more than h_j intervals.

Use the previous theorem to partition the set of intervals associated with side I_j into h_j sets of edge-disjoint intervals for every j. Assign these sets to the grid lines parallel to side I_j of the rectangle. The intervals corresponding to different segments of the same path can be connected by extending both intervals until they meet. The beginning of each path can be connected to the corresponding terminals by a line orthogonal to the corresponding side of the rectangle I. See Fig. 2. □

4. Routing on three layers

We extend some of the results from the previous section to problems where the outside boundary O is connected, but not necessarily a rectangle. First we find edge-disjoint paths connecting the paired terminals in the grid graph. The paths will *not* be knock-knee free. Therefore this path system does not correspond to a two layer routing. Next we show that the proof can be extended to yield a routing using three layers.

We define the associated cycle network $N_{\mathrm{cy}} = (G_{\mathrm{cy}}, P)$ in a way similar to the definition in the previous section. The cycle is the boundary of the rectangle I, as before. The difference is in the capacity of an edge e. For a corner v of I let e_v denote the next edge after v going clockwise around I as shown in Fig. 4. The semicut associated with an edge in $E(I)$ is defined as before. For an edge e, that is not one of the edges e_v, the capacity is defined to be the number of edges in the associated semicut.

Consider a corner v of I. We define G_v, the corner subgraph of the grid graph G, as follows (see also Fig. 3). Consider the horizontal and vertical lines through v and delete every node that is on the same side of one of these lines as the rectangle I. Let $l(v)$ denote the distance of v from the outside boundary O in G_v. The capacity of the edge e_v is defined to be the minimum of $l(v)+1$ and the number of edges in the associated semicut.

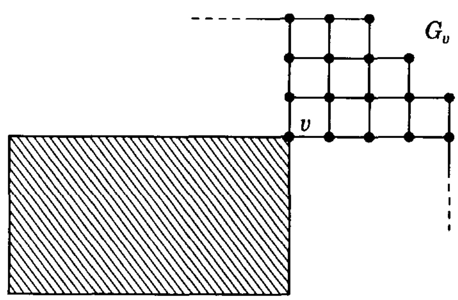

Fig. 3. The corner subgraph G_v.

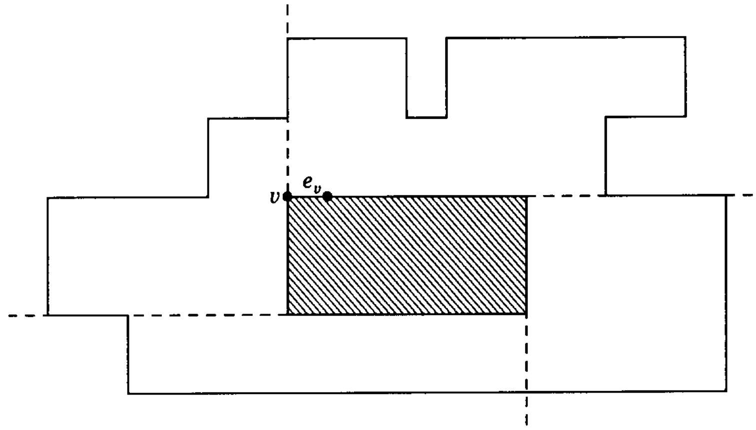

Fig. 4. Dividing a three layer routing problem into four subproblems.

Analogously to the algorithm in the previous section we shall divide the problem into four subproblems (parallel to the four sides of I) and solve these problems separately. However one has to be somewhat more careful in the way the subproblems are defined (see Fig. 4).

The following analog of Gallai's theorem (Theorem 3.1) will be used to solve the four subproblems. Let $N=(G,P)$ be a grid graph defined by the grid points inside a boundary B, such that B contains the horizontal line segment connecting the origin to point $(x,0)$, is between the two vertical lines passing through the origin and the point $(x,0)$, and is in the nonnegative orthant. Assume that pairs of terminals are given such that the terminals are on the horizontal and vertical lines through the origin, the terminals on the vertical side are the nodes $(0,i)$ for $i=0,1,\ldots,m$ for some m, every node is the terminal of at most one pair, and a pair has at most one terminal on the vertical part of the boundary B.

Analogously to the cycle network associated with routing around a rectangle we define the *line network* $N_{\text{li}}=(G_{\text{li}},P)$ associated with the above problem $N=(G,P)$. The nodes of the line network are the integers on the line segment from the origin to $(x,0)$. The network is obtained by contracting all vertical edges, and then replacing parallel copies of an edge by a single one with the appropriate capacity.

Lemma 4.1. *There exist edge-disjoint paths connecting terminal pairs in the network problem defined above if and only if there are paths connecting the terminals in the associated line network* N_{li}.

Proof. The *only if* part is trivial. Given a solution to the problem in the grid graph a solution in the line network can be constructed by contracting all vertical edges.

The *if* part can be shown by a simple greedy construction. We assume that the paths in the line graph are simple. The paths in the grid graph can be constructed

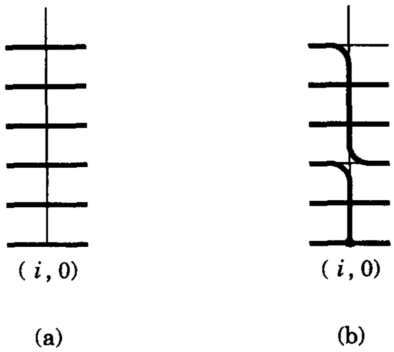

Fig. 5. The paths above $(i, 0)$: (a) $(i, 0)$ is not a terminal; (b) $(i, 0)$ is a terminal.

from left to right, so that if r paths use an edge e in the line graph, then the edge-disjoint paths in the grid graph will use the bottom r parallel copies of e. If a point $(i, 0)$, $0 \le i \le x$ is not a terminal, then no path will bend above this point; if it is a terminal, then at most one path, in addition to the one ending at $(i, 0)$, will bend above $(i, 0)$ (see Fig. 5). □

Theorem 4.2. *There exist edge-disjoint paths connecting terminal pairs around a rectangle I in a grid graph defined by I and a connected boundary O surrounding I if and only if there are edge-disjoint paths connecting the terminals in the associated cycle network* N_{cy}.

Proof. The necessity is trivial. To prove the sufficiency assume that we have a solution for the problem in the cycle network $N_{\text{cy}} = (G_{\text{cy}}, P)$. One may assume without loss of generality that the paths in N_{cy} are simple. For a corner v of I let e_v denote the edge along I in the clockwise direction after v (as was used in the definition of N_{cy}). Assign the paths using this edge to the parallel copies of e_v in the grid graph closest to e_v. We will show the existence of edge-disjoint paths in the grid graph where a path using the edge e_v in N_{cy} uses the assigned parallel copy of e_v in the grid graph (see Fig. 4). Deleting the parallel copies of the edges e_v for every corner v divides the problem into four problems of the form considered by the previous lemma (with the considered region turned). Using the lemma four times gives the edge-disjoint paths in N. □

The resulting edge-disjoint paths are not knock-knee free. Therefore these paths may not give rise to a two layer routing. Next we use the proof of Lemma 4.1 to construct a three layer routing. We need a version of the lemma that constructs routings rather than edge-disjoint paths. In fact, we shall need two different routings, one will be used on the two horizontal subproblems, the other one on the two vertical subproblems.

Lemma 4.3. *Consider the network problem defined before Lemma* 4.1. *If there are edge-disjoint paths connecting the terminals in the associated line network* N_{li}, *then there exists a three layer routing satisfying either one of the following two sets of properties.*

- *The horizontal edges between nodes* $(0,i)$ *and* $(1,i)$ *(for any* i*) are assigned to the second layer, and every vertical edge is either assigned to the first or the third layer.*
- *Any assignment of the edges between nodes* $(0,i)$ *and* $(1,i)$ *to the first and third layer can be extended to a three layer routing where the edges between nodes* $(j,1)$ *and* $(j,0)$ *used by a path to enter terminal* $(j,0)$ *are assigned to the second layer.*

Proof. Both routings can be constructed greedily along the same line as the proof of the previous lemma. We leave some of the details to the reader.

For constructing the first kind of routing one can use the same edge-disjoint paths as used for the previous lemma with maintaining the additional property that for every edge e of the line graph all but at most one parallel copies of e are assigned to the second layer.

The second routing can also be constructed along the same lines as the proof of the previous lemma. Here the resulting edge-disjoint paths are going to be slightly different than in that lemma. We maintain the additional property, that all horizontal edges are assigned to the first and third layer, and for every node v in the line graph all but at most one of the vertical edges on the vertical line through v are assigned to the second layer. For this routing one has to slightly modify the natural construction for the previous lemma. If the path ending at terminal $(j,0)$ reaches the vertical line through $(j,0)$ at (j,l) then the path at $(j,l+1)$ might also have to be bent at this point and continued horizontally from (j,l). This construction also maintains that for every edge e the paths use the lowest parallel copies of e. □

Theorem 4.4. *A set of terminal pairs on a rectangle* I *can be connected by edge-disjoint paths routed in three layers in a grid graph defined by* I *and a connected boundary* O *surrounding* I *if and only if there are edge-disjoint paths connecting the terminals in the associated cycle network* N_{cy}.

Proof. Let us assume that we are given a solution to the problem in N_{cy}. We use the construction in the proof of Theorem 4.2 to break the problem into four subproblems. Use the first kind of routing established in Lemma 4.3 on the two subproblems along the two horizontal boundaries of I. Then use the second kind of routing established by the lemma (turned around by 90 degrees) along the two vertical sides. □

One can easily verify that the procedure used for proving the theorem can be implemented in $O(k+n_O)$ time where n_O is the size of the boundary O.

5. Minimum area routing

The *minimum area routing problem* is given by a rectangle I and k pairs of terminals P in I, so that every node on I is the terminal of at most one pair and the corners of I are not terminals. A rectangle O surrounding I is called *feasible* if the grid network defined by I and O and the set of pairs P has a two layer routing. The results in Section 3 give an easy way to check whether a given rectangle O is feasible. Here we shall use some of the ideas from that section to present an algorithm for finding a minimum area feasible rectangle O.

Let I_1 denote the top horizontal side of the rectangle I, and analogously let I_i for $i=2,3,4$ denote the ith side, where the sides are numbered clockwise around I. The corners are considered as part of both adjacent sides. Indices will be meant modulo 4. Let $E(I_i)$ denote the set of edges in the cycle network N_{cy} connecting nodes in I_i. A rectangle O can be defined by four numbers h_i for $i=1,\ldots,4$, where h_i is the number of edges in a semicut associated with an edge $e \in E(I_i)$. If a rectangle I has height h_I and length l_I the area of the rectangle O is

$$\text{Area}(O) = (h_I + h_1 + h_3 - 2)(l_I + h_2 + h_4 - 2). \tag{1}$$

We will show that there is a feasible rectangle that minimizes both $h_1 + h_3$ and $h_2 + h_4$ simultaneously.

Recall the notion of the demand. For two edges e and e' of the cycle $d(e,e')$ denotes the number of terminal pairs separated by the cut $\{e,e'\}$. For i, j such that $1 \le i, j \le 4$ let

$$d_{i,j} = \max\{d(e,e')\colon\ e \in E(I_i),\ e' \in E(I_j)\}. \tag{2}$$

Combining Theorems 2.2 and 3.2 we get that four integers (h_1, h_2, h_3, h_4) define a rectangle O such that two layer routing is feasible if and only if the corresponding cycle network N_{cy} satisfies the cut criterion and the parity criterion. The cut criterion in the cycle network is equivalent to the following inequalities

$$h_i + h_j \ge d_{i,j} \quad \text{for every } 1 \le i, j \le 4. \tag{3}$$

The two inequalities for $i=1$, $j=3$ and $i=2$, $j=4$ imply lower bounds on the size of the rectangle O. However, there might not exist a feasible rectangle O matching these lower bounds. A tighter lower bound can be established using parity arguments. For every i and j we consider the set of edges in $E(I_i)$ on which the maximum defining $d_{i,j}$ is attained. These are the edges that participate in tight cuts if $h_i + h_j = d_{i,j}$.

$$D_{i,j} = \{e \in E(I_i)\colon\ \exists e' \in E(I_j) \text{ such that } d(e,e') = d_{i,j}\}. \tag{4}$$

The set $D_{i,j}$ is defined to be *even* if each connected component of $I - D_{i,j}$ contains an even number of terminals. Otherwise it is *odd*. An easy parity argument shows that $D_{1,3}$ is even if and only if $D_{3,1}$ is, and an analogous statement holds for $D_{2,4}$ and $D_{4,2}$. Edges in tight cuts must be used by as many paths as their capacity, whereas every odd cut must contain at least one edge that is not used up to its capacity. This implies the following improved lower bound.

$$h_1+h_3 \geq \begin{cases} d_{1,3}, & \text{if } D_{1,3} \text{ is even,} \\ d_{1,3}+1, & \text{if } D_{1,3} \text{ is odd;} \end{cases} \tag{5}$$

$$h_2+h_4 \geq \begin{cases} d_{2,4}, & \text{if } D_{2,4} \text{ is even,} \\ d_{2,4}+1, & \text{if } D_{2,4} \text{ is odd.} \end{cases} \tag{6}$$

Theorem 5.1. *There exists a feasible rectangle O whose sizes h_1, h_2, h_3 and h_4 match the lower bounds in the inequalities* (5) *and* (6) *with equality.*

The proof of the theorem will be given at the end of this section. In the next section we are going to present a simple algorithm that for every x and y finds all the feasible rectangles with sizes $h_2+h_4=x$ and $h_1+h_3=y$. This algorithm and the above theorem can be used to find a minimum area feasible rectangle. However, the proof of the theorem is also algorithmic. Define h_1^*, h_2^*, h_3^* and h_4^* as follows

$$h_1^* = \frac{d_{1,2}+d_{1,4}-(d_{2,3}+d_{3,4})+2d_{1,3}}{4},$$

$$h_2^* = \frac{d_{1,2}+d_{2,3}-(d_{1,4}+d_{3,4})+2d_{2,4}}{4},$$

$$h_3^* = \frac{d_{2,3}+d_{3,4}-(d_{1,2}+d_{1,4})+2d_{1,3}}{4},$$

$$h_4^* = \frac{d_{1,4}+d_{3,4}-(d_{1,2}+d_{2,3})+2d_{2,4}}{4}. \tag{7}$$

Note that $h_1^*+h_3^*=d_{1,3}$ and $h_2^*+h_4^*=d_{2,4}$, but h_i^* is not necessarily an integer. The following theorem establishes that h_i^* can be used as a fairly close approximation of the optimal h_i.

Theorem 5.2. *There exists a minimum area feasible rectangle O whose sizes h_1, h_2, h_3 and h_4 satisfy $|h_i-h_i^*| \leq \frac{3}{2}$ for every $1 \leq i \leq 4$.*

The following lemma will be useful in establishing the inequalities in (3).

Lemma 5.3. *For any i $(1 \leq i \leq 4)$,*

$$d_{i,i}+d_{i+1,i-1} \leq d_{i,i+1}+d_{i,i-1},$$

$$d_{i,i+1}+d_{i+2,i+3} \leq d_{i,i+2}+d_{i+1,i+3}. \tag{8}$$

Proof. It is easy to check that for four edges $e_1, \dots, e_4 \in E(I)$ that appear in this order around I,

$$d(e_1, e_3) + d(e_2, e_4) \geq d(e_1, e_2) + d(e_3, e_4).$$

To establish the second inequality choose $e_j \in E(I_j)$ such that $d(e_j, e_{j+1}) = d_{j,j+1}$ for $j = i$ and $j = i+2$. We have the following chain of inequalities: $d_{i,i+1} + d_{i+2,i+3} = d(e_i, e_{i+1}) + d(e_{i+2}, e_{i+3}) \leq d(e_i, e_{i+2}) + d(e_{i+1}, e_{i+3}) \leq d_{i,i+2} + d_{i+1,i+3}$. The first inequality can be established similarly. $\square$

Corollary 5.4. *$h_1^* + h_3^* = d_{1,3}$, $h_2^* + h_4^* = d_{2,4}$ and the numbers h_1^*, h_2^*, h_3^*, h_4^* satisfy the inequalities* (3).

Let us recall some ideas from the proof of Theorem 2.2. Independent of the choice of $h_1, \dots, h_4$ all terminals will be odd nodes in the associated cycle network N_{cy}. This suggests adding some pairs of these as additional pairs of terminals to the cycle network. For a set of nodes $V' = \{v_1, \dots, v_{2l}\}$ occurring in this order around the rectangle I, we consider the cycle network with the additional pairs of terminals (v_{2i-1}, v_{2i}) for $i = 1, \dots, l$. Denote by N' the cycle network defined in this way, and let $d'(e,f)$ denote its demand function, and let $d'_{i,j}$ and h'_i (for $1 \leq i, j \leq 4$) be defined by equalities (2) and (7) with N' in place of N_{cy}.

Lemma 5.5. *There exists a set $V' = \{v_1, v_2, \dots\}$ numbered around the rectangle I, such that V' consists of all the terminals and a subset V_c of the four corners of I, $d'_{i,j} \leq d_{i,j} + 2$ for every $1 \leq i, j \leq 4$ and for $i = 1$ and 2,*

$$d'_{i,i+2} = \begin{cases} d_{i,i+2}, & \textit{if } D_{i,i+2} \textit{ is even,} \\ d_{i,i+2} + 1, & \textit{otherwise.} \end{cases}$$

Proof. The inequalities $d'_{i,j} \leq d_{i,j} + 2$ are satisfied for any choice of V'. The appropriate set V_c can be defined depending on the parities of $D_{1,3}$, $D_{2,4}$, $d_{1,3}$ and $d_{2,4}$. For example, consider the case when both $D_{1,3}$ and $D_{2,4}$ are odd. Let c_i denote the corner of I between sides I_i an I_{i-1}. The set V_c can be defined as

$$V_c = \begin{cases} \{c_1, c_3\}, & \text{if } d_{1,3} \text{ and } d_{2,4} \text{ are both even,} \\ \{c_1, c_2\}, & \text{if } d_{1,3} \text{ is even and } d_{2,4} \text{ is odd,} \\ \{c_2, c_3\}, & \text{if } d_{1,3} \text{ is odd and } d_{2,4} \text{ is even,} \\ \emptyset, & \text{if } d_{1,3} \text{ and } d_{2,4} \text{ are both odd.} \end{cases} \tag{9}$$

The resulting set V' satisfies the requirements of the lemma if its nodes are ordered around I. To prove this first notice that the only nodes in the cycle network N' that can have an odd number of terminals are the corners. Therefore the parity of $d'(e, e')$ for $e \in E(I_i)$ and $e' \in E(I_j)$ depends only on i and j. Now consider a pair of edges $e \in E(I_i)$ and $e' \in E(I_{i+2})$ such that $d_{i,i+2} = d(e, e')$. Both components of

$I-\{e,e'\}$ contain an odd number of terminals in V'. Therefore $d'(e,e')=d(e,e')+1$. This implies that $d'_{i,i+2}=d_{i,i+2}+1$.

The cases when one or both of $D_{1,3}$ and $D_{2,4}$ are even can be dealt with similarly. This is left to the reader. □

Let N' denote the cycle network defined by the additional set of terminals as given by the above lemma.

Corollary 5.6. *$h_i^*-1\le h_i'\le h_i^*+\frac{3}{2}$ and $h_i^*+h_{i+1}^*-1\le h_i'+h_{i+1}'\le h_i^*+h_{i+1}^*+2$ for every $1\le i\le 4$.*

Lemma 5.7. *The two sides of the inequalities (8) with $d_{i,j}$ replaced by $d'_{i,j}$ have the same parity. The corner c_i between sides I_{i-1} and I_i is in V_c if and only if $d'_{i-1,i}$ is odd.*

Proof. The only nodes that can have an odd number of terminals in N' are the four corners of I. This implies that, for $e\in E(I_i)$ and $e'\in E(I_j)$ the parity of $d'(e,e')$ depends only on i and j. Therefore the parity of $d'_{i-1,i}$ is the same as the parity of $d'(e_i,e_i')$ if e_i and e_i' are the two edges incident to the corner c_i. This latter number is odd if and only if c_i is a terminal. This proves the second statement. To prove the first statement observe that for every four edges $e_1,\dots,e_4$ the two numbers $d(e_1,e_3)+d(e_2,e_4)$ and $d(e_1,e_2)+d(e_3,e_4)$ have the same parity. □

Lemma 5.8. *The numbers $h_1',\dots,h_4'$ are either all integers or are all odd multiples of $\frac{1}{2}$.*

Proof. By Lemma 5.7 all of $d'_{1,2}+d'_{3,4}$, $d'_{1,3}+d'_{2,4}$ and $d'_{1,4}+d'_{2,3}$ have the same parity. This implies the lemma. □

Proof of Theorems 5.1 and 5.2. First note that $d'_{i,i}$ is even for every i. Next consider the two numbers $h_i'+h_{i+1}'-d'_{i,i+1}$ and $h_i'+h_{i-1}'-d'_{i-1,i}$. By the previous lemma both are integers. By Corollary 5.4 and Lemma 5.7 their sum has the same parity as $2h_i'-d'_{i,i}$. Consequently, the two numbers have the same parity if and only if h_i' is an integer.

For any quadruple $h=(h_1,\dots,h_4)$ let N_h denote the cycle network corresponding to the rectangle O with these sizes, and the enlarged set of terminals. For any h all nodes of the cycle network are even except some of the four corners. We will define numbers h_i for $i=1,\dots,4$ such that $h_1+h_3=h_1'+h_3'$, $h_2+h_4=h_2'+h_4'$, $|h_i-h_i^*|\le\frac{3}{2}$ and the cycle network N_h is even and satisfies the cut criterion. This will establish both theorems.

Case 1: *All of h_1', h_2', h_3', h_4' are integers.* This implies that $h_i'+h_{i+1}'-d'_{i,i+1}$ have the same parity for every $1\le i\le 4$. If these numbers are even, then every corner is even in $N_{h'}$, so we can choose $h_i=h_i'$ for $i=1,\dots,4$.

Otherwise we have that all corners are odd in $N_{h'}$ and $h_i'+h_{i+1}'-d_{i,i+1}'\geq 1$ for every $1\leq i\leq 4$. This implies that

$$\begin{aligned}2h_i' &= (h_i'+h_{i+1}')+(h_i'+h_{i-1}')-(h_{i+1}'+h_{i-1}')\\ &\geq d_{i,i+1}'+1+d_{i,i-1}'+1-d_{i+1,i-1}'\geq d_{i,i}'+2.\end{aligned}$$

The quadruple h is defined by decreasing h_i' by 1 and increasing h_{i+2}' by 1 for some i. The resulting network N_h satisfies the cut criterion and is even. By Corollary 5.6 one of the four possible choices also satisfies $|h_j-h_j^*|\leq\frac{3}{2}$ for every j.

Case 2: *All of* h_1', h_2', h_3', h_4' *are odd multiples of* $\frac{1}{2}$. In this case $2h_i'-d_{i,i}'$ is odd, and hence at least 1 for every $1\leq i\leq 4$. Furthermore, the parities of $h_i'+h_{i+1}'-d_{i,i+1}'$ and $h_i'+h_{i-1}'-d_{i,i-1}'$ are different. Either both $h_1'+h_2'-d_{1,2}'$ and $h_3'+h_4'-d_{3,4}'$ or both $h_2'+h_3'-d_{2,3}'$ and $h_1'+h_4'-d_{1,4}'$ are positive odd integers. Consider the first case (the second case can be treated analogously). We can decrease h_1' and h_2' and at the same time increase h_3' and h_4' by $\frac{1}{2}$ (or increase h_1' and h_2' and decrease h_3' and h_4') without violating the cut criterion. Both of the resulting networks N_h are even. By Corollary 5.6 one of the two possible choices also satisfies $|h_i-h_i^*|\leq\frac{3}{2}$. □

6. Finding all feasible rectangles of a given size

Finally, we consider the problem when both I and the size of the outside rectangle O are given and we are interested in finding the feasible relative positions of these two rectangles. More formally, given integers x and y, and a rectangle I and k pairs of terminals P in I, so that every node in I is the terminal of at most one pair and the corners of I are not terminals. Find all feasible rectangles whose size satisfies $h_2+h_4=x$ and $h_1+h_3=y$. We call a pair of integers (h_1,h_2) *feasible* if $h_1,h_2,h_3=y-h_1$ and $h_4=x-h_2$ define a feasible rectangle O.

Combining Theorems 2.2 and 3.2 we see that integers $h_1,h_2,h_3=y-h_1$ and $h_4=x-h_2$ define a feasible rectangle O if and only if the inequalities (3) are satisfied and certain parity conditions hold. Inequalities (3) define a two-dimensional polytope Q so that all feasible pairs (h_1,h_2) are in Q. We shall call two pairs of integers (h_1,h_2) and (h_1',h_2') in Q *equivalent* if they are on the same face of Q (i.e., they satisfy the same set of inequalities from (3) with equation).

Theorem 6.1. *Let* (h_1,h_2), (h_1',h_2') *be equivalent pairs in* Q. *If one is feasible and the other one is not then* h_1+h_2 *and* $h_1'+h_2'$ *have different parity.*

Proof. Consider the two rectangles O and O' and the two corresponding cycle networks $N_{\rm cy}$ and $N_{\rm cy}'$. By Theorem 3.2 we can concentrate on the corresponding problems in the cycle network. The two pairs are equivalent. Therefore the same cuts are tight in the two cycle networks. This implies that one must satisfy the parity criterion and the other one must not. Hence there must be a cut (e,e') that is even in one problem and odd in the other one. Consequently, the parity of $h_i + h_j$ and $h'_i + h'_j$ must be different for some i and j, which is only possible if the parities of $h_1 + h_2$ and $h'_1 + h'_2$ are different. $\square$

This theorem can be used to find all feasible vectors in linear time by checking two points on each of the at most eight facets of Q and the vertices. Note that all of the interior points in Q are feasible.

References

[1] A. Frank, Edge-disjoint paths in planar graphs, J. Combin. Theory Ser. B 39 (1985) 164–178.

[2] A. Frank and E. Tardos, On routing around a rectangle, in: Proceedings of the ETAN Network Theory Symposium, Yugoslavia (1984).

[3] T. Gallai, Graphen mit triangulierbaren ungeraden Vierecken, Magyar Tud. Akad. Mat. Kut. Int. Kozl. 7 (1962) 381–395.

[4] T.F. Gonzalez and S.L. Lee, An algorithm for optimal routing around a rectangle, in: Proceedings of the 20th Allerton Conference on Communication, Control, and Computing (1982) 636–645.

[5] T.F. Gonzalez and S.L. Lee, A linear time algorithm for optimal routing around a rectangle, J. ACM 35 (1988) 810–831.

[6] A.S. LaPaugh, A polynomial time algorithm for optimal routing around a rectangle, in: Proceedings of the 21st Symposium on Foundation of Computer Science (1980) 280–302.

[7] H. Okamura and P.D. Seymour, Multicommodity flows in planar graphs, J. Combin. Theory Ser. B 31 (1981) 75–81.

[8] H. Suzuki, A. Ishiguro and T. Nishizeki, Variable priority queue and doughnut routing, J. Algorithms, to appear.

[9] H. Suzuki, T. Nishizeki and N. Saito, Algorithms for routing around a rectangle (extended abstract), in: D.S. Johnson et al., eds., Discrete Algorithms and Complexity, Proceedings of the Japan-US Joint Seminar, 1986, Kyoto (Academic Press, Boston, MA, 1987) 99–105.