ALGORITHMS FOR ROUTING*
IN PLANAR GRAPHS

A 84/07

M. Becker K. Mehlhorn
Fachbereich 1o
Universität des Saarlandes
D-66oo Saarbrücken
West-Germany

Abstract:

We present algorithms for solving routing problems for two-terminal nets in planar graphs. Our algorithms run in time $O(n^2)$ for general planar graphs and in time $O(bn)$ for grid graphs where n is the number of vertices and b is the number of vertices on the boundary of the infinite face.

* This research was supported by the Deutsche Forschungsgemeinschaft, SFB 124, VLSI-Entwurfsmethoden und Parallelität. Part of it was done while the second author was visiting SIEMENS A.G., ZTI VENUS.

I. Introduction

A planar routing problem is given by

a) a planar undirected graph $G = (V,E)$ with a fixed embedding into the plane

b) a set Ne of two-terminal nets where a net $N \in Ne$ is an unordered pair of vertices on the boundary of the infinite face.

A solution to a planar routing problem (V,E,Ne) is given by a set of pairwise edge-disjoint paths P_N, $N \in Ne$, such that P_N connects the two terminals of net N.

For sets $X,Y \subseteq V$ we define

$$
\begin{aligned}
cap(X,Y) &= |\{(x,y) \in E;\ x \in X,\ y \in Y\}| \\
cap(X) &= cap(X,V-X) \\
d(X,Y) &= |\{(x,t) \in Ne;\ s \in X,\ t \in Y\}| \\
d(X) &= d(X,V-X) \\
fcap(X) &= cap(X) - d(X)
\end{aligned}
$$

We call $fcap(X)$ the free capacity of cut X. If x is a vertex we write $fcap(x)$, $cap(x)$, $d(x)$ instead of $fcap(\{x\})$, $cap(\{x\})$ and $d(\{x\})$. Note that $cap(x)$ is the degree of vertex x. The following theorem was shown by OKamura/Segmour.

Theorem (OKamura/Segmour): Let (V,E,Ne) be a planar routing problem. If $fcap(X)$ is non-negative and even for every $X \subseteq V$ then the routing problem has a solution. □

In this paper we will present algorithms for solving routing problems in planar graphs. We use the following notation.

Definition: Let $P = (V,E,Ne)$ be a planar routing problem, $n = |V|$. Let b be the number of vertices on the boundary of the infinite face. The problem P is solvable if it has a solution. It is even if fcap(x) is even for all $x \in V$. It is half-even if fcap(x) = cap(x) is even for all nodes x which do not lie on the boundary of the infinite face. □

We prove in this paper the following

Theorem: Let $P = (V,E,Ne)$ be a half-even routing problem.

a) It can be tested in time $O(bn)$ whether P is solvable.

b) A solution can be constructed in time $O(n^2)$.

c) If (V,E) is a grid graph, i.e. a subgraph of the planar integer grid, then a solution can be found in time $O(bn)$. If P is even then time $O(n^{3/2})$ suffices.

A proof of parts a), b) and c) for even problems can be found in section 2. The extension to half-even problems is made in section 3. In section 4, a weak generalization of the theorems above to multi-terminal nets is made. If $2d(X) < cap(X)$ for all cuts X then a solution exists and can be found in time $O(n^2)$.

We close the introduction with a comparison to other work. Hassim and Matsumoto/Nishizeki/Saito consider the problem of multi-commodity flows in planar networks, i.e. edges have real capacities and nets have real demands. The goal is to construct flow functions which realize the demands and obey the capacity constraints. Our problem differs from the multi-commodity flow problem in two respects

a) it is simpler since all capacities and demands are equal to one in our case.

b) it is harder since we insist that a solution consists of edge-disjoint paths. Figure 1 shows a problem which has a solution as a flow problem (all edges carry flow 1/2 for both commodities) but not as a routing problem.

Hassim and Matsumoto/Nishizeki/Saito describe algorithms which solve multi-commodity flow problems in planar graphs in time $O(n^4)$ and time $O(kn + n^2\sqrt{\log n})$ respectively where $k = |Ne|$. If all capacities and demands are integral and the problem is even then the flow functions constructed by their algorithms are integral. In particular, even routing problems can be solved in time $O(n^2\sqrt{\log n})$ by their algorithms; note that $k \leq |E| = O(n)$ in planar routing problems. Matsumoto/Nishizeki/Saito also show how to test solvability in time $O(\min(n^2 \log^* n, bn(\log n)^{1/2}))$. Our improvement to their results for even problems is trivial; we only show how to drop the various factors involving log n. However, our algorithm for even problems differs considerably from theirs. It has the advantage that it can be made to run in time $O(n^{3/2})$ on grid graphs while their algorithm will always run in time $O(n^2)$. The other major contribution of the present paper is the extension of the theory to half-even problems.

Grid graphs are particularly relevant for VLSI routing problems. Special cases of grid graph routing problems were considered previously by Rivest/Baratz/Miller, Preparata/Lipski, Frank, Mehlhorn/Preparata, Kaufmann/Mehlhorn. They show how to solve problems for channels, rectangles and grid graphs without holes respectively.

II. Even Routing Problems

Throughout this section $P = (V,E,Ne)$ is an even routing problem. We will show how to construct a solution (if there is one) in time $O(n^2)$. For grid graphs time $O(n^{3/2})$ suffices. Our algorithm is an almost direct implementation of Okamura/Seymour's constructive proof of their theorem and hence correctness can be shown using their methods.

Theorem 1: Let $P = (V,E,Ne)$ be an even routing problem. Then it can be tested in time $O(bn)$ whether P has a solution. If P has a solution then a solution can be constructed in time $O(n^2)$.

Proof: We will first show how to construct a solution in time $O(n^2)$. The test for solvability will be a corollary.

We need one additional definition. A cut X is simple if there are at most two edges e_1, e_2 on the boundary of the infinite face having exactly one endpoint in X. We are now ready for the algorithm.

(1) while $E \neq \emptyset$
(2) do let $e = (a,b)$ be an edge on the boundary of the infinite face;
(3) let $e_0, e_1, \ldots, e_{m-1}$ be the edges on the boundary of the infinite face in clockwise order with $e = e_0$;
(4) if there is a simple cut X with $a \notin X$, $b \in X$ with $fcap(X) < 0$
then halt and declare the problem unsolvable
fi;

(5) <u>if</u> there is a simple cut X with $a \notin X$, $b \in X$ and
$fcap(X) = 0$

(6) <u>then</u> let X be such a simple cut which contains as few vertices of the boundary as possible;
let $N = (s,t)$ be a net with $s \notin X$, $t \in X$ and s as close to a as possible, i.e. there is no net (s',t') with $s' \notin X$, $t' \in X$ and s' lying on that piece of the boundary between s (exclusive) and a (inclusive) not containing b (cf. Figure 3);
delete edge e, reserve edge e for net N and replace net N by nets $N_1 = (s,a)$ and $N_2 = (b,t)$.
<u>else</u> delete edge e and add an additional net $N = (a,b)$.
<u>fi</u>
<u>od</u>

The correctness of this algorithm is almost immediate from the work of Okamura/Seymour. We give a short correctness proof to make the paper self-contained.

Let as assume that P is solvable. Then clearly $fcap(X) \in 2\,\mathbb{N}_0$ for every simple cut X initially. It suffices to show that $fcap(X) \in 2\,\mathbb{N}_0$ for every simple cut X is an invariant of the algorithm because then line (4) is never executed and the algorithm will construct a solution.

Consider an execution of the while-loop which removes edge e from the current graph G_c. Note first fcap(Y) is even for all cuts Y iff fcap(v) is even for every vertex v. This can be seen as follows. Let $Y \subseteq V$ be arbitrary. Then

$$\begin{aligned} fcap(Y) &= cap(Y) - d(Y) \\ &= \sum_{v \in Y} cap(v) - 2cap(Y,Y) - \sum_{v \in Y} d(v) + 2d(Y,Y) \end{aligned}$$

and hence fcap(Y) is even if fcap(v) is even for all $v \in V$. It is now easy to see that all problems constructed by the algorithm are even.

Let Y be a simple cut in $G_c - e$. Then Y is also a simple cut in G_c and $\overline{\text{fcap}}(Y) \geq \text{fcap}(Y) - 2$ where $\overline{\text{fcap}}$ is the free capacity in the modified problem. If $\text{fcap}(Y) \geq 2$ or $\overline{\text{fcap}}(Y) = \text{fcap}(Y)$ then clearly $\overline{\text{fcap}}(Y) \in 2\,\mathbb{N}_0$. Recall that we have already shown that the modified problem is even. The only case to consider is therefore that $\text{fcap}(Y) = 0$ and $\overline{\text{fcap}}(Y) = \text{fcap}(Y) - 2$. We may assume w.l.o.g. that $t \notin Y$; consider $V - Y$ otherwise. If $\overline{\text{fcap}}(Y) = \text{fcap}(Y) - 2$ then either (case A) $|Y \cap \{a,b\}| = 1$ and $s \notin Y$ or (case B) $a,b \in Y$ and $s \notin Y$. We can restrict case A further. If $a \notin Y$, $b \in Y$ then Y contains fewer boundary nodes than X, a contradiction to the choice of X. We may therefore assume $a \in Y$, $b \notin Y$ in case A. The following lemma was shown in Okamura/Seymour.

<u>Lemma 1:</u> For all cuts X and Y

$$d(X) + d(Y) = d(X \cup Y) + d(X \cap Y) + 2d(X - Y, Y - X)$$
$$\text{cap}(X) + \text{cap}(Y) = \text{cap}(X \cup Y) + \text{cap}(X \cap Y) + 2\text{cap}(X - Y, Y - X).$$

□

We will apply Lemma 1 in both cases. The following observation is also used in both cases. The boundary nodes in $Y - X$ lie between s (exclusive) and a (inclusive) and hence $d(X - Y, Y - X) = 0$ by choice of net $N = (s,t)$. We consider cuts $X \cup Y$ and $X \cap Y$. We have

$$\begin{aligned} d(X \cup Y) + d(X \cap Y) &= d(X) + d(Y) && \text{, by lemma 1 and the observation above} \\ &= \text{cap}(X) + \text{cap}(Y) && \text{, since X and Y are saturated} \end{aligned}$$

$= \mathrm{cap}(X \cup Y) + \mathrm{cap}(X \cap Y) + 2\mathrm{cap}(X - Y, Y - X)$

, by lemma 1

Case A: In case A we have $a \in Y - X$, $b \in X - Y$ and hence $\mathrm{cap}(X - Y, Y - X) > 0$ and thus $d(X \cup Y) + d(X \cap Y) > \mathrm{cap}(X \cup Y) + \mathrm{cap}(X \cap Y$. Thus one of the simple cuts $X \cup Y$ or $X \cap Y$ is oversaturated, a contradiction.

Case B: In case B we conclude $d(X \cup Y) + d(X \cap Y) \geq \mathrm{cap}(X \cup Y) + \mathrm{cap}(X \cap Y)$. Thus $\mathrm{fcap}(X \cup Y) = 0 = \mathrm{fcap}(X \cup Y)$. Next note that $b \in X \cap Y$ and that simple cut $X \cap Y$ contains fewer boundary nodes than X. This contradicts the choice of X.

This completes the correctness proof. Note that the correctness proof also shows that P is solvable iff $\mathrm{fcap}(X) \geq 0$ for every simple cut in the initial problem. This observation will lead to the efficient test for solvability.

We turn to the implementation next. The main task is to determine the existence of a saturated cut through edge $e = (a,b)$. As in Hassim and Matsumoto/Nishizeki/Saito we solve this task by means of the multiple source dual graph (cf. Figure 4). In the dual graph there is a dual edge for every edge of the original graph. The dual edge connects vertices which are located in the faces separated by the edge. In every face (exept the infinite face) we position one dual vertex but in the infinite face we have a dual vertex for every edge on the boundary of the infinite face.

Let $e_o,\ldots,e_{m-1}$ be the edges on the boundary of the infinite face in clockwise order and let $e = e_o$. Let

$$\mathrm{cap}(e,e_i) = \min\{\mathrm{cap}(X);\ X \text{ is a simple cut which cuts boundary edges } e \text{ and } e_i\}$$

$$d(e,e_i) = \{d(X);\ X \text{ is a simple cut which cuts boundary edges } e \text{ and } e_i\}\ , \text{ and}$$

$$\mathrm{fcap}(e,e_i) = \mathrm{cap}(e,e_i)$$

Let v_i be the dual vertex in the infinite face corresponding to edge e_i. Then $\mathrm{cap}(e,e_i)$ is equal to the length of a shortest path from v_o to e_i in the dual graph and hence $\mathrm{cap}(e,e_i)$, $1 \le i \le m - 1$, can be computed in time $O(n)$ by breadth first search. It is also easy to see that $d(e,e_i)$, $1 \le i \le m - 1$, can be computed in time $O(n)$ by a simple walk around the boundary of the infinite face. We summarize in

Lemma 2: $\mathrm{fcap}(e,e_i)$, $1 \le i \le m - 1$, can be computed in time $O(n)$. □

The remainder of the loop body can clearly also be done in time $O(n)$. Thus a single execution of the loop body takes time $O(n)$ and hence total running time is $O(n^2)$.

For the test of solvability we only have to compute $\mathrm{fcap}(e_i,e_j)$ for all i and j. For every fixed i this take time $O(n)$. Thus total running time is $O(bn)$ where b is the number of edges on the boundary of the infinite face in the initial graph. □

We will next show how to improve upon theorem 1 for planar graphs with small C-connected edge separators.

Definition: a) Let $G = (V,E)$ be a planar graph. A set $E' \subseteq E$ of edges is a C-connected edge separator if

1) removal of E' splits G into subgraphs (V_1,E_1) and (V_2,E_2) with $|V_1| \leq 2/3\ |V|$ and $|V_2| \leq 2/3\ |V|$

2) E' can be ordered, say $E' = \{e_1,\ldots,e_k\}$ such that edge e_i is on the boundary of the infinite face of graph $G - \{e_1,\ldots,e_{i-1}\}$.

b) A family F of planar graphs has C-connected edge separators of size f(u) if every $G = (V,E) \in F$ has a C-connected edge separator E' of size $|E'| \leq f(|V|)$ and if the subgraphs (V_1,E_1) and (V_2,E_2) obtained by removing E' again belong to F. □

C-connected edge separators are a very restrictive notion of separator which is particularly suitable for our application. However, not all planar graphs have small C-connected edge separators as Figure 5 shows. The graph shown in Figure 5 consists of n nested triangles. All C-connected edge separators have size $\Omega(n)$. Fortunately, there is a family of graphs having small C-connected separators, namely grid graphs.

Definition: A grid graph is a subgraph of the two-dimensional integer grid. □

Figure 1o shows a grid graph.

Lemma 3: Let G be a grid graph with n vertices. Then G has a C-connected edge separator of size $\leq 4\sqrt{n} + 2$. Moreover, the separator can be found in time O(n).

Proof (adapted from Lengauer/Mehlhorn, theorem 1): Let integers a and b be minimal such that a rectangle of side lengths a and b encloses the grid graph. Assume w.l.o.g. that $a \leq b$ and that the side of length a is horizontal. Let L_i, $-1 \leq i \leq b$, be the number of edges intersected by a horizontal line which runs in distance $i + 1/2$ from the bottom side of the rectangle. Then $L_{-1} = L_b = 0$ and $\sum_i L_i \le n$. Let i_o be minimal such that at least $n/2$ nodes lie below line L_{i_o}, i.e. $\le n/2$ nodes lie below line L_{i_o-1} and $\le n/2$ nodes lie above line L_{i_o}. Let $i_1 < i_o$ and $i_2 \ge i_o$ be such that (cf. figure 7)

1) $i_1 \ge i_o - 1 - \sqrt{n}$ and $L_{i_1} \le \sqrt{n}$

2) $i_2 \le i_o + \sqrt{n}$ and $L_{i_2} \le \sqrt{n}$

Note that i_1 exists because $\sum_{i_o-1-\sqrt{n}\le i\le i_o-1} L_i \le n$; i_2 exists for a similar reason. Let A be the set of vertices above line L_{i_2}, B be the set of vertices below line L_{i_1} and C be the set of vertices between the two lines. Let L be a vertical line with maybe one horizontal segment of length one running between L_{i_1} and L_{i_2} and dividing C into two parts C_1 and C_2 of size $\lfloor |C|/2 \rfloor$ and $\lceil |C|/2 \rceil$ respectively. Then $|A| \le \lceil n/2 \rceil$, $|B| \le \lceil n/2 \rceil$, $|C_1| \le \lceil n/2 \rceil$ and $|C_2| \le \lceil n/2 \rceil$ and hence the set of edges intersected by lines L_{i_1}, L_{i_2} and L forms a C-connected edge separator. This set has size $\le 4\sqrt{n} + 2$. □

<u>Theorem 2:</u> Let $P = (V,E,Ne)$ be an even routing problem with (V,E) a grid graph. Then a solution can be constructed (if there is one) in time $O(n^{3/2})$.

<u>Proof:</u> Let $E' \subseteq E$ be a C-connected edge separator of size $|E'| = O(\sqrt{n})$; E' exists by Lemma 3. Let $E' = \{e_1,\dots,e_k\}$ where e_i is on the boundary of the infinite face of graph $G - \{e_1,\dots,e_{i-1}\}$. We use the algorithm described in the proof of theorem 1 and let e run through edges $e_1,e_2,\dots,e_k$

in the first $k = O(n)$ iterations of the loop. This takes time $O(n^{3/2})$ and splits the graph into subgraphs with n_1 and n_2 vertices respectively, where $\max(n_1,n_2) \le 2n/3$. We therefore have the following recurrence for the running time

$$T(n) \le O(n^{3/2}) + \max\{T(n_1) + T(n_2);\ \begin{array}{l} n_1 + n_2 = n \\ \max(n_1,n_2) \le 2n/3\} \end{array}$$

Thus $T(n) = O(n^{3/2})$ as claimed (cf. Mehlhorn, Vol. 2, page 121 where a similar recurrence is solved). □

III. Half-even Problems

A routing problem $P = (V,E,Ne)$ is half-even if $fcap(v) = cap(v) = deg(v)$ is even for all nodes not on the boundary of the infinite face. Figure 1 shows a half-even routing problem. In this section we extend our results to half-even routing problems. We proceed in three steps. In the first step (Lemma 4a) we show that a solvable half-even problem $P = (V,E,Ne)$ always has a solvable even extension $P = (V,E,Ne \cup Ne')$, in the second step (Lemma 4b), we develop one particular strategy for computing such an extension and in the third step (Theorem 3) we show how to make the strategy run in time $O(bn)$.

Our reduction is based on the concepts of U-minimal cut and canonical extension. Let $U = \{v;\ fcap(v) \text{ is odd}\}$ be the vertices with odd free capacity. Then U has even cardinality and all vertices in U lie on the boundary of the infinite face (recall that we deal with half-even problems). Let X be a saturated cut. Then $X \cap U$ has even cardinality (this can be seen as follows: $0 = cap(X) - d(X) = \sum_{x \in X} (cap(x) - d(x)) - 2cap(X,X) + 2d(X,X)$. Thus the number of vertices in X with odd free capacity is even). Let $u_1,\ldots,u_{2k}$ be the vertices in $X \cap U$ in clockwise order on the boundary of the infinite face. Cut X is U-minimal if $X \cap U \neq \emptyset$ and there is no simple saturated cut Y with $Y \cap U = \{u_i,\ldots,u_j\}$ where $1 < i < j < 2k$. The canonical extension of $P = (V,E,Ne)$ with respect to X is given by $P' = (V,E,Ne \cup \{(u_{2i-1},u_{2i});\ 1 \le i \le k\})$. Note that all vertices of X have even free capacity in P'.

Lemma 4: Let $P = (V,E,Ne)$ be a solvable half-even routing problem.

a) There is a solvable even extension $(V,E,Ne \cup Ne')$ where Ne' is a pairing of $U = \{v;\ fcap(v) \text{ is odd}\}$

b) If X is a U-minimal cut then the canonical extension of P with respect to X is a half-even solvable routing problem.

Proof: a) Let P_N, $N \in Ne$, be a solution for P. Consider graph $G' = (V,E')$ with $E' = E - \{P_N;\ N \in Ne\}$ which is obtained from $G = (V,E)$ by removing all edges which are used in the solution paths. We have

a) if v is a vertex on the boundary of G then v has odd degree in G' iff $v \in U$

b) if v is a vertex in the interior of G then v has even degree in G'.

We conclude that G' decomposes into paths connecting vertices in U and cycles. The paths connecting vertices in U induce the desired pairing Ne'. This proves a).

b) Let $P_o = (V,E,Ne \cup Pa)$ be a solvable even extension of the half-even problem $P = (V,E,Ne)$ and let X be a U-minimal cut. We have argued above that $U \cap X$ is even. Let

$$Pa' = \{(s,t) \in Pa;\ s \notin X,\ t \notin X\}$$

Let $P' = (V,E,Ne \cup Pa')$. Then P' is a half-even solvable problem with $U \cap X$ as its set of vertices of odd free capacity. Let P" be the canonical extension of P' with respect to X. Then P" is an even problem which extends the canonical extension of P with respect to X. It therefore suffices to show that P" is solvable.

Let Y be a simple cut. Then $d''(Y) \geq d'(Y) - 2$ where $d''(Y)$ ($d'(Y)$) denotes the density of Y with respect to problem P"(P'). Also $d'(Y) \leq cap'(Y)$ since P' is solvable. Assume first that $d''(Y) \geq d'(Y) - 1$. Then $cap''(Y) - d''(Y) \geq cap'(Y) - d'(Y) - 1 \geq -1$. Since P" is even conclude $cap''(Y) - d''(Y) \geq 0$. This leaves the case that $d''(Y) - d'(Y) - 2$. Then $Y \cap U = \{u_i,\dots,u_j\}$ for some i,j with $1 < i < j < 2k$ where $u_1,\dots,u_{2k}$ is the clockwise ordering of the vertices in $U \cap X$. Thus Y is not saturated in P since X is a U-minimal cut. Also the nets in Pa' pair only vertices outside Y. Thus Y is not saturated in problem P' and therefore $cap''(Y) - d''(Y) \geq cap'(Y) - d'(Y) - 2 \geq 1 - 2 = -1$. Since P" is even this implies $cap''(Y) - d''(Y) \geq 0$.

We conclude that there are no oversaturated simple cuts in P". Hence P" is solvable. □

Lemma 4b leads to the following algorithm for turning half-even problems into even problems.

```
(1)  U ← {v; fcap(v) is odd}
(2)  while U ≠ ∅
(3)  do if there is an oversaturated cut
(4)     then terminate and declare that the problem has no
             integer-valued solution
(5)     fi;
(6)     let X be a U-minimal cut (X = V is possible);
(7)     construct the canonical extension;
(8)     U ← U - X
(9)  od
```

The correctness of this algorithm is immediate from lemma 4b. Note that $U \cap X$ has always even dardinality and hence line (7) is always executable. Figure 8 illustrates the extension algorithm.

It remains to discuss an efficient implementation. The most difficult design decision is how to handle U-minimum cuts. We start with the observation that if X and X' are cuts with $U \cap X = U \cap X'$ and $fcap(X) \leq fcap(X')$ then only cut X has to be considered because of the following two trivial facts.:

1) if net (s,t) is added in line (7) then $fcap(X)$ goes down by one iff $fcap(X')$ does.

2) if X' is U-minimal then X is U-minimal.

The observation above suggests the following strategy for finding U-minimal cuts. Let the vertices on the boundary of the exterior face be labelled $0,1,\ldots,\ell-1$ in clockwise order and let $U = \{u_0,u_1,\ldots,u_{2k-1}\}$. For $0 \leq p,q \leq 2k-1$ let $X(p,q)$ be a simple cut with

1) $X(p,q) \cap U = \{u_p, u_{p+1}, \ldots, u_q\}$

2) $X(p,q)$ has the smallest free capacity among all cuts satisfying 1). Ties are broken arbitrarily.

Note that we may assume w.l.o.g. that $X(p,q) = X(q+1,p-1)$. We represent cut $X(p,q)$ by the triple (p,q,d) where $d = \text{dens}(X(p,q))$. We store the cuts $X(p,q)$ as follows.

For every p we have the linked list of (representatives of) cuts $X(p,q)$ in clockwise order of q.

In addition, we have a pointer min(p) pointing to the first q such that $X(p,q)$ is saturated. Furthermore, we link the two occurrences of a cut (namely as $X(p,q)$ and as $X(q+1,p-1)$). We have

Lemma 5: a) The family $X(p,q)$ of cuts can be constructed in time $O(bn)$

b) A U-minimum cut X can be found in time $O(|U|)$ given the data structure described above

c) The data structure can be updated in time $O(|U|)$ after adding net (u_i, u_{i+1}) for some i.

Proof: a) Immediate from the proof of Theorem 1.

b) If there is no p such that min(p) is defined then $X = V$ is the only saturated cut and hence the only U-minimum cut. Assume now that min(p) is defined for some p, say $p = p_0$. Then the following algorithm find a U-minimal cut in time $O(|U|)$.

```
p ← p_o + 1                    -- indices are mod 2K
while p lies between p_o and min(p_o)
do if min(p) is defined and min(p) lies between p and
      min(p) in the clockwise ordering
   then p_o ← min(p)
   fi;
   p ← p + 1
od
```

c) Suppose that we add net (u_i, u_{i+1}) for some i and hence delete vertices u_i and u_{i+1} from U. Three actions are required.

1) Reduce fcap(i+1,q) and fcap(q,i) by one for all q. Update the min-pointers min(q) for all q. This takes time $O(|U|)$.

2) Among the cuts X(i,q), X(i+1,q) and X(i+1,q) select the one with smallest free capacity for all q. This defines the new linked list for vertex u_{i+2} and takes time $O(|U|)$.

3) For every q keep only one of the cuts X(q,i-1), X(q,i), and X(q,i+1). This reduces the length of the list for vertex u_q by 2 and takes time $O(|U|)$.

Theorem 3: Let P = (V,E,Ne) be a half-even routing problem. Then in time O(bn) one can

a) decide whether P is solvable and
b) extend P to a solvable even routing problem P' if P is solvable.

Proof: By the algorithm and lemma 5 above. Note that $|U| \le b$ and that the while-loop is executed at most b times. □

IV. Multi-terminal Nets

Let $G = (V,E)$ be a planar graph. A multi-terminal net N is a subset (of size ≥ 2) of the vertices of the boundary of the infinite face. A routing for a multi-terminal net is a tree $T(N) \subseteq E$ connecting the points in N. A routing problem for multi-terminal nets is given by a planar graph $G = (V,E)$ and a set $Ne = \{N_1,\ldots,N_M\}$ of multi-terminal nets. A solution is given by routings $T(N_1),\ldots,T(N_M)$ such that $T(N_i)$ is a routing for N_i and $T(N_i)$ and $T(N_j)$ are edge-disjoint for $i \neq j$.

The density of a cut X is defined in complete analogy to the case of two-terminal nets, i.e. $d(X)$ is the number of nets having one but not all terminals in X

$$d(X) = |\{N \in Ne;\ \emptyset \neq X \cap N \neq N\}|$$

As before, we call a problem (V,E,Ne) half-even if $\mathrm{cap}(v) = |\{w \in V;, (v,w) \in E\}|$ is even for all interior nodes v. We have:

Theorem 4: Let $P = (V,E,Ne)$ be a half-even routing problem with multi-terminal nets. If

$$2\ \mathrm{dens}(X) < \mathrm{cap}(X)$$

for every cut X then P has a solution. Moreover a solution can be found in time $O(n^2)$. For grid graphs time $O(n^{3/2})$ suffices.

Proof: Let $Ne = \{N_1,\ldots,N_M\}$. Consider a net N_i. Let $v_1,v_2,\ldots,v_k$ be the terminals of N_i as they appear in clockwise order on the boundary of the infinite face. Replace N_i

by the set of k-1 two-terminal nets $N_i^j = (v_j, v_{j+1})$; $1 \leq j < k$. Let Ne' be the new set of nets and let dens'(X) be the density of a cut X with respect to family Ne' of nets. Then dens'(X) $\leq$ 2 dens(X) $<$ cap(X). The problem (V,E,Ne') is half-even and has no saturated non-trivial cut. Therefore, X = V is the only U-minimal cut and hence (V,E,Ne') can be turend into an even problem in time O(n) by the algorithm of section III. The algorithms of section II can then be used to solve the even problem. □

Figure 9 illustrates the proof of theorem 4.

V. Conclusion

We presented efficient algorithms for routing problems in planar graphs. The algorithms are guaranteed to find a solution if there is one. A weak generalization to multi-terminal nets was made. Multi-terminal nets deserve further investigation.

Acknoledgement: We thank M. Kaufmann and F. Preparata for many inspiring discussions about routing problems.

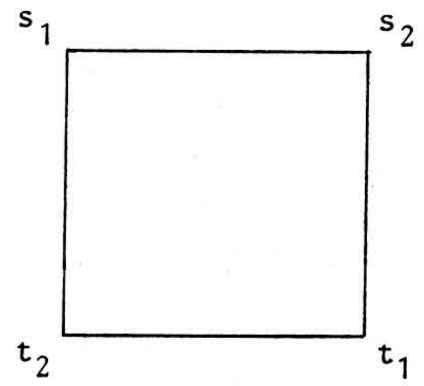


Figure 1: A problem which is solvable as a flow problem but not as a routing problem; the two nets are (s_1,t_1) and (s_2,t_2).

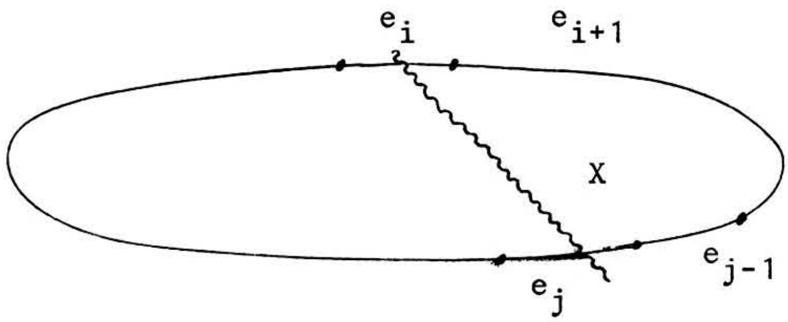


Figure 2: A cut X through edges e_i and e_j

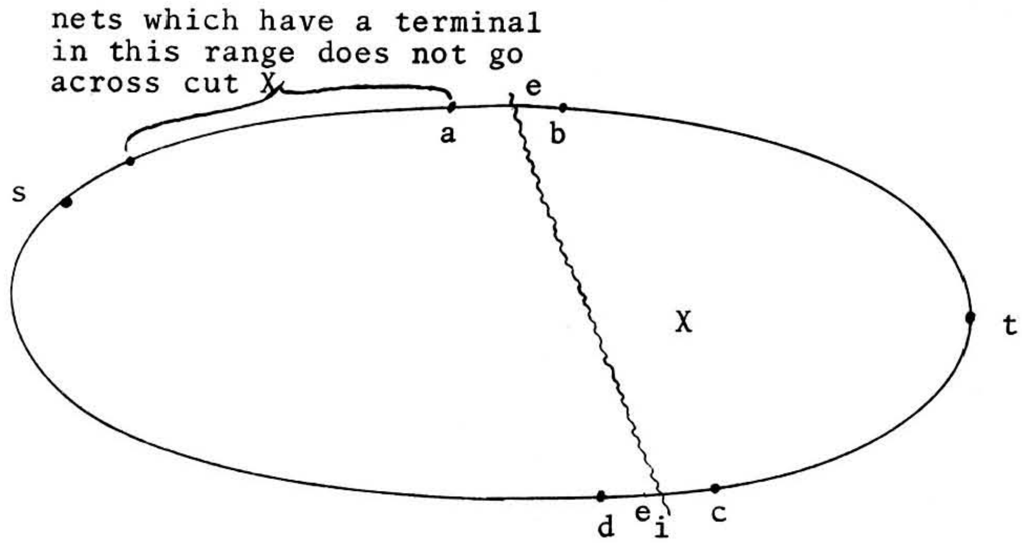


Figure 3: $fcap(e,e_i) = fcap(X) = 0$

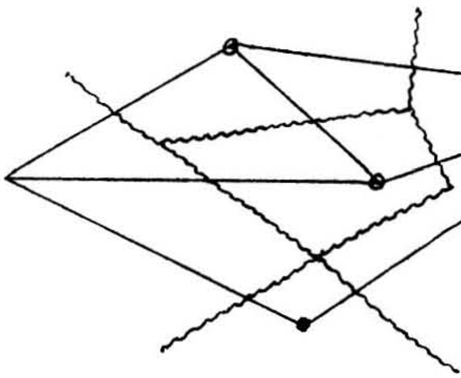


Figure 4: A planar graph G and its multiple-source dual. Dual edges are shown as wiggled lines.

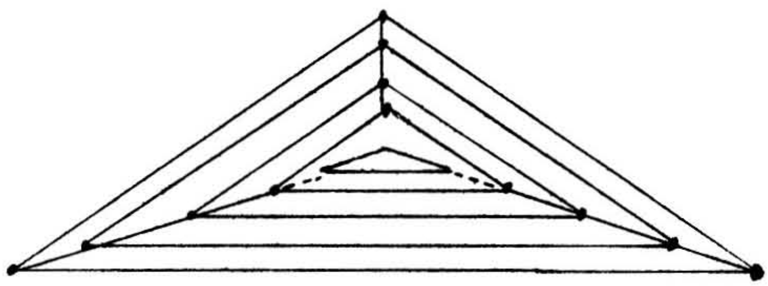


Figure 5: A planar graph all of whose C-connectes edge separators have size $\Omega(n)$.

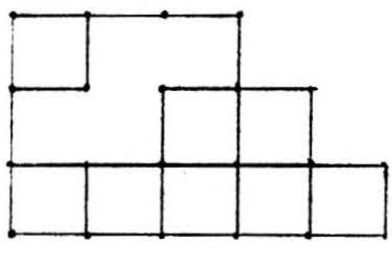


Figure 6: A grid graph

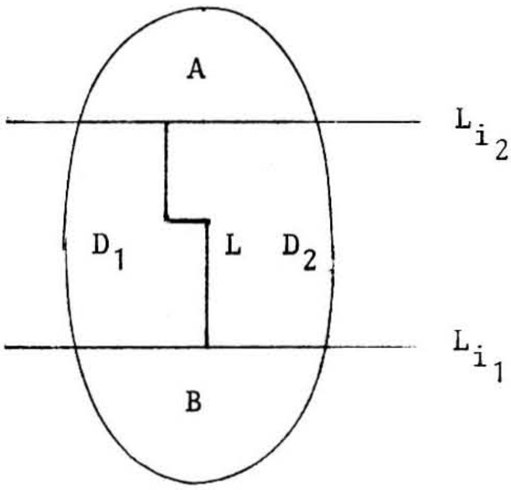
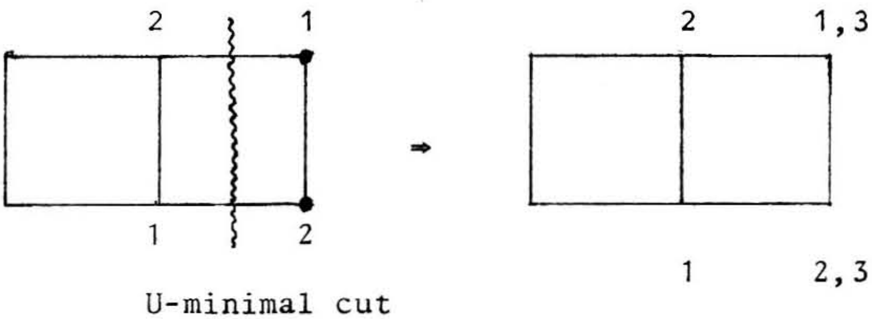
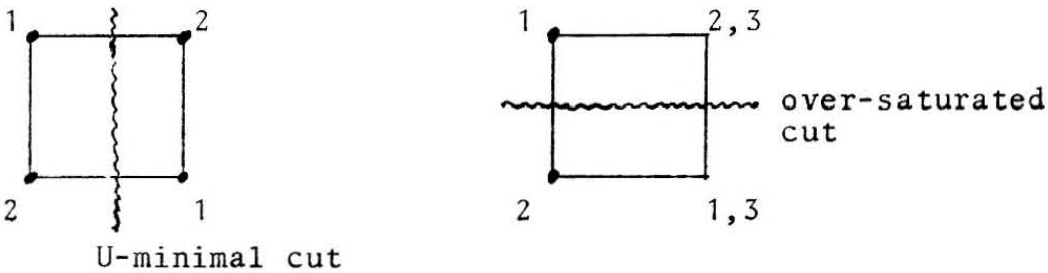


Figure 7: Cuts L_{i_1}, L_{i_2} and L



a) a solvable problem



b) an unsolvable problem

Figure 8: The extension algorithm; odd nodes are shown solid.

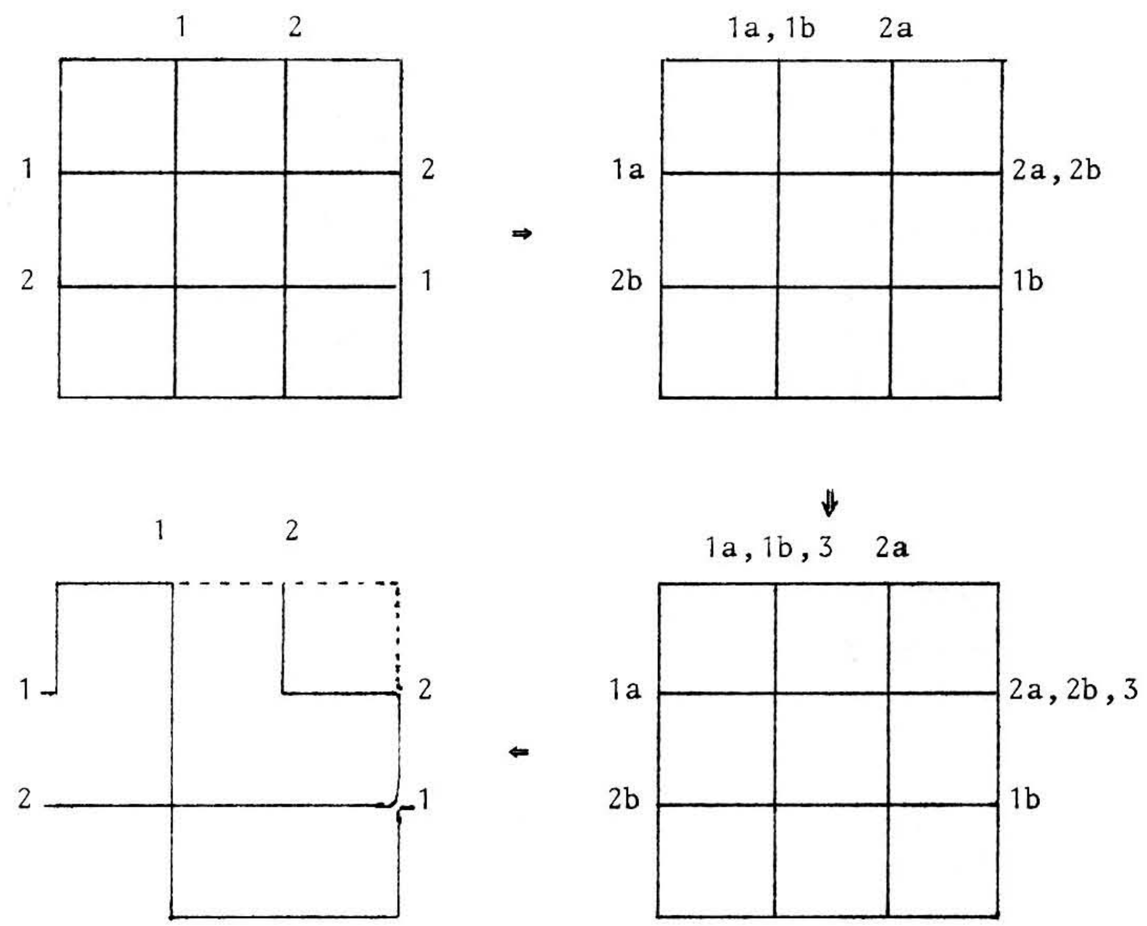


Figure 9: A routing problem with multi-terminal nets, a related problem with two-terminal nets, the even problem obtained from theorem 6, and the solution obtained by theorem 4 (the path for the artificial net 3 is shown dashed).

References

[1] Frank, A., Disjoint paths in a rectilinear grid, Combinatorica 2, 4.

[2] Hassin, R., On Multicommodity Flows in Planar Graphs, Networks, Vol 14 (1984), 225-235.

[3] Kaufmann, M./Mehlhorn, K., Generalized Switchbox Routing, Techn. Report, FB 1o, Universität des Saarlandes, 1984

[4] Lengauer, Th./Mehlhorn, K., On the Complexity of VLSI Computations, in CMU Conference on VLSI Systems and Computations, Computer Science Press, 1981, 89-99.

[5] Matsumoto, K./Nishizeki, T./N. Saito, N., An efficient algorithm for finding multi-commodity flows in planar graphs, SICOMP, to appear

[6] Mehlhorn, K., Data Structures and Efficient Algorithms Vol 2: Graph Algorithms and NP-completeness, Springer Verlag, EATCS Monographs in Theoretical Computer Science

[7] Mehlhorn, K./Preparata, F.P., Routing through a rectangle, Techn. Report, ACT-42, Coordinated Science Lab., University of Illinois, Urbana; (Oct. 1983), submitted for publication.

[8] Okamura, H./Seymour, P.D., Multicommodity flows in planar graphs, Journal of Combinatorial Theory, B31 (1983) 75-81.

[9] Preparata, F.P./Lipski, Jr., W., Optimal three-layer channel routing, IEEE Trans. on Computers, C-33, 5 (May 1984) 427-437.

[1o] Rivest, R.L./Baratz, A.E./Miller, G., Provably good channel routing algorithms, Proceedings CMU Conference on VLSI Systems and Computations, (Oct. 1981) 151-159.