

# Routing through a Generalized Switchbox

MICHAEL KAUFMANN AND KURT MEHLHORN

*Fachbereich 10, Universität des Saarlandes, 6600 Saarbrücken, West Germany*

Received November 15, 1984

We present an algorithm for the routing problem for two-terminal nets in generalized switchboxes. A generalized switchbox is any subset  $R$  of the planar rectangular grid with no nontrivial holes, i.e., every finite face has exactly four incident vertices. A net is a pair of nodes of nonmaximal degree on the boundary of  $R$ . A solution is a set of edge-disjoint paths, one for each net. Our algorithm solves standard generalized switchbox routing problems in time  $O(n(\log n)^2)$  where  $n$  is the number of vertices of  $R$ , i.e., it either finds a solution or indicates that there is none. A problem is standard if  $\deg(v) + \text{ter}(v)$  is even for all vertices  $v$  where  $\deg(v)$  is the degree of  $v$  and  $\text{ter}(v)$  is the number of nets which have  $v$  as a terminal. For nonstandard problems we can find a solution in time  $O(n(\log n)^2 + |U|^2)$  where  $U$  is the set of vertices  $v$  with  $\deg(v) + \text{ter}(v)$  is odd © 1986 Academic Press, Inc.

## 1. INTRODUCTION

In this paper we solve the routing problem for two-terminal nets in generalized switchboxes. A *generalized switchbox* is any subset  $R$  of the planar rectangular grid without holes, i.e., all finite faces of  $R$  have exactly four incident edges (cf. Fig. 1). Let

$$B(R) := \{v; v \text{ node of } R \text{ and } v \text{ has degree } \leq 3\}$$

be the nodes of  $R$  which do not have maximal degree. Note that all nodes of  $B(R)$  are incident to the infinite face. A *two-terminal net* is an unordered pair of points in  $B(R)$ . A *generalized switchbox routing problem* (GSRP) is given by a generalized switchbox  $R$  and a set  $N = \{(s_i, t_i); 1 \leq i \leq m\}$  of nets. A solution to the problem is a set  $P = \{p_i, 1 \leq i \leq m\}$  of paths such that

- (1)  $p_i$  connects  $s_i$  and  $t_i$  for  $1 \leq i \leq m$ ,
- (2)  $p_i$  and  $p_j$  are edge-disjoint for  $i \neq j$ .

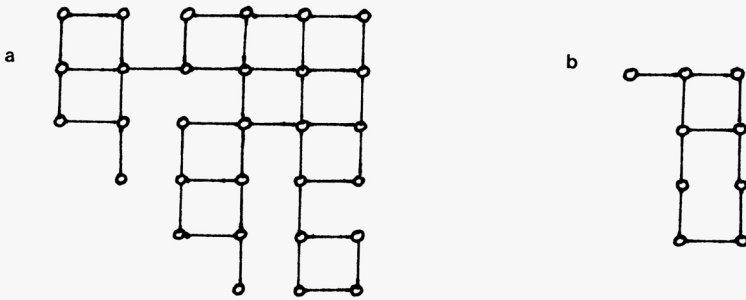


FIG. 1. (a) A generalized switchbox, (b) not a generalized switchbox.

In this paper we present an algorithm which solves standard generalized switchbox routing problems in time  $O(n(\log n)^2)$  where  $n$  is the number of vertices of the routing region  $R$ . A routing problem is *standard* if  $\text{deg}(v) + \text{ter}(v)$  is even for all nodes  $v$  where  $\text{deg}(v)$  is the degree of node  $v$  and  $\text{ter}(v)$  is the number of nets which have  $v$  as a terminal. We call  $\text{deg}(v) + \text{ter}(v)$  the extended degree of node  $v$ . For nonstandard GSRPs we do slightly worse. We show how to find a solution in time  $O(n \log^2 n + |U|^2)$ , where  $U$  is the set of vertices with odd extended degree.

A solution to a routing problem in the sense described above is usually called a solution in *knock-knee mode*. Note that a vertex  $v$  of  $R$  is used by either one wire or two wires which either go straight through  $v$  or bend in  $v$  (cf. Fig. 2). Previous work on routing problems in knock-knee mode can be found in Preparata–Lipski, Frank, Mehlhorn–Preparata, Becker–Mehlhorn, Kramer–van Leeuwen, and Brady–Brown. Preparata–Lipski solve the channel routing problem, Frank and Mehlhorn–Preparata solve the switchbox routing problem. A switchbox is a rectangular subset of the plane grid. The running time of their algorithm is  $O(n \log n)$  and  $O(u \log u)$ , respectively, where  $u$  is the circumference of the rectangle. Becker–Mehlhorn consider a more general problem than the one considered here. They consider arbitrary subsets of the planar grid (holes are allowed!!) and solve the routing problem in time  $O(n^{3/2})$ . Finally Brady–Brown consider

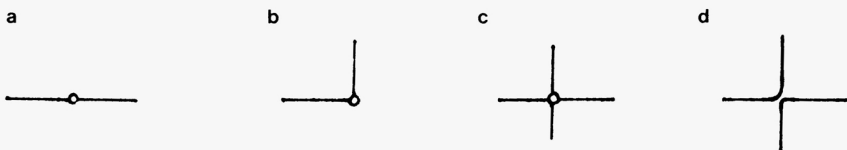


FIG. 2. (a) straight, (b) knee, (c) crossing, (d) knock-knee.

the problem of layer assignment. They show that any layout in knock-knee mode can be wired using four conducting layers.

All papers mentioned above (except Brady–Brown) and also the present paper are based on a theorem of Okamura–Seymour on multi-commodity flow in planar graphs. We review their theorem in Section 2. In Section 3 we refine their theorem to the special case of standard generalized switchboxes. In Section 4 we derive an algorithm for standard GSRPs and analyse its running time. In Section 5 we deal with nonstandard GSRPs.

## 2. THE THEOREM OF OKAMURA–SEYMOUR

Let  $G = (V, E)$  be a graph and let  $N$  be a set of unordered pairs of vertices of  $G$ ;  $N = \{(s_i, t_i); 1 \leq i \leq m\}$ . A cut is a subset  $X \subseteq V$  of the vertices of  $G$ . The capacity of a cut  $X$  is the number of edges in  $E$  with exactly one end in  $X$  and the density of a cut  $X$  is the number of nets  $(s, t) \in N$  with exactly one terminal in  $X$ , i.e.

$$\text{cap}(X) = |\{e \in E; e = (a, b) \text{ and } a \in X, b \notin X\}|,$$

$$\text{dens}(X) = |\{(s, t) \in N; s \in X, t \notin X\}|.$$

We will also use

$$\text{CAP}(X) = \{e \in E; e = (a, b) \text{ and } a \in X, b \notin X\}$$

and

$$\text{dens}(X_1, X_2) = \{(s, t) \in N; s \in X_1, t \in X_2\}$$

$$\text{for } X_1, X_2 \subseteq V, X_1 \cap X_2 = \emptyset.$$

**THEOREM (Okamura–Seymour).** *If  $G$  is planar and can be drawn in the plane such that  $s_1, \dots, s_m, t_1, \dots, t_m$  are all on the boundary of the infinite region and  $\text{cap}(X) - \text{dens}(X)$  is nonnegative and even for all cuts  $X \subseteq V$  then there are pairwise edge-disjoint paths  $p_1, \dots, p_m$  such that  $p_i$  connects  $s_i$  and  $t_i$ ,  $1 \leq i \leq m$ .*

Okamura–Seymour give a constructive proof of their theorem; their proof leads to the following algorithm which can be made to run in time  $O(n^2)$  as shown by Becker–Mehlhorn.

Let  $\hat{G}$  be an embedding of  $G$  with  $s_1, \dots, s_m, t_1, \dots, t_m$  on the boundary of the infinite face. We may assume w.l.o.g. that  $\hat{G}$  is 2-connected. Then the boundary of the infinite face consists of a circuit  $C$  which we regard as a

subgraph of  $\hat{G}$ . We say that a cut  $X$  is *critical* if  $X$  is connected, *saturated*, i.e.,  $\text{cap}(X) = \text{dens}(X)$ , and  $\text{CAP}(X)$  contains exactly two edges of  $C$ . Thus if  $X$  is critical then  $C|(\mathcal{V}(C) \cap X)$  and  $C|(\mathcal{V}(C) - X)$  are both paths.

We can now describe the algorithm.

let  $e = (v, w)$  be an arbitrary edge on the boundary  $C$  of the infinite face of  $C$ ;

**if** there is a critical cut  $X$  with  $v \in X, w \notin X$

**then** let  $X$  be such a critical cut with

$|\mathcal{V}(C) \cap X|$  minimal;

let  $(s, t) \in N$  be a net with  $s \in X, t \notin X$  such that the subpath of  $C$  from  $w$  to  $t$  not using  $v$  has minimal length; (cf. Fig. 3)

remove edge  $e$  from  $G$ ;

replace net  $(s, t)$  by the pair  $(s, v)$  and  $(w, t)$  of nets;

construct a solution for the reduced graph using the algorithm recursively and obtain the path for net  $(s, t)$  by connecting the paths for nets  $(s, v)$  and  $(w, t)$  by edge  $e$ .

**else** remove edge  $e$  from  $G$  and add net  $(v, w)$  to the set of nets;

construct a solution for the reduced graph and throw away the path for net  $(v, w)$

**fi**

The correctness of this algorithm can be deduced from the paper of Okamura–Seymour; a proof can be found in Becker–Mehlhorn.

We close this section with a collection of simple observations. For vertex  $v \in \mathcal{V}$  let  $\text{deg}(v)$  be the degree of  $v$  and let  $\text{ter}(v)$  be the number of nets in  $N$  which have  $v$  as a terminal. We call a routing problem (given as a planar

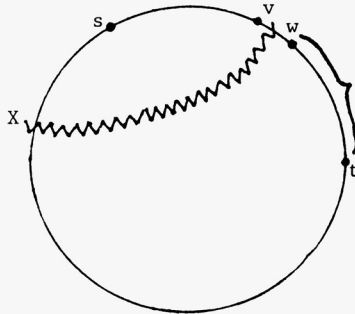


FIG. 3. Choice of  $(s, t)$ . The area between  $w$  and  $t$ : no net having a terminal in the interval goes across cut.



graph and a set of nets) *standard* if the extended degree  $\deg(v) + \text{ter}(v)$  is even for all  $v \in V$ . We call it *solvable* if it has a solution.

LEMMA 1. *Let  $G = (V, E)$  be a planar graph and let  $N$  be a set of nets having their terminals on the boundary of the infinite face.*

(a) *The routing problem  $(G, N)$  is standard iff  $\text{cap}(X) - \text{dens}(X)$  is even for every cut  $X$ .*

(b) *A standard routing problem  $(G, N)$  is solvable iff no cut  $X$  is oversaturated, i.e., there is no cut with  $\text{cap}(X) < \text{dens}(X)$ .*

*Proof.* (a) Let  $X \subseteq V$  be arbitrary. We have

$$\text{dens}(X) = \sum_{v \in X} \text{ter}(v) - 2|\{(s, t); (s, t) \in N \text{ and } s, t \in X\}|$$

and

$$\text{cap}(X) = \sum_{v \in X} \text{deg}(v) - 2|\{(a, b); (a, b) \in E \text{ and } a, b \in X\}|.$$

This proves (a).

(b) If  $(G, N)$  is solvable then there is clearly no oversaturated cut. Conversely, if  $\text{dens}(X) \leq \text{cap}(X)$  for every cut  $X$  then  $\text{cap}(X) - \text{dens}(X)$  is nonnegative and even by part (a). Hence  $(G, N)$  is solvable by Okamura–Seymour.  $\square$

### 3. CRITICAL CUTS IN STANDARD GENERALIZED SWITCHBOXES

Let  $R$  be a generalized switchbox. We use  $C(R)$  to denote its boundary, i.e., the boundary of the infinite face. Let  $B(R) = \{v \in C(R); \text{deg}(v) \leq 3\}$  and let  $N \subseteq B(R) \times B(R)$  be a set of nets. We assume throughout this section that  $(R, N)$  is a standard problem, i.e.,  $\text{deg}(v) - \text{ter}(v)$  is even for all  $v \in V$ .

Our first goal is to show that nodes  $v \in C(R)$  with  $\text{ter}(v) = \text{deg}(v)$  are easily handled.

LEMMA 2. *Let  $v \in B(R)$  be a node with  $\text{deg}(v) = \text{ter}(v)$ . Let  $(v, t_i)$ ,  $1 \leq i \leq \text{ter}(v)$ , be the sets which have  $v$  as a terminal and let  $b_i$ ,  $1 \leq i \leq \text{ter}(v)$ , be the neighbors of  $v$ . The following transformations turn a solvable problem into a solvable problem.*

- (1) *If  $\text{ter}(v) = 1$  then delete  $v$  and replace net  $(v, t_1)$  by net  $(b_1, t_1)$*
- (2) *If  $\text{ter}(v) = 2$  then let  $b_1, t_1, t_2, b_2$  be the order in which these four points lie on circuit  $C$ ; consecutive points may be equal. Delete node  $v$  and replace nets  $(v, t_1), (v, t_2)$  by  $(b_1, t_1)$  and  $(b_2, t_2)$ .*

(3) If  $\text{ter}(v) = 3$  then let  $b_1, b_3$  be neighbors of  $v$  on circuit  $C$ . Let  $b_1, t_1, t_2, t_3, b_3$  be the order in which these five points lie on circuit  $C$ ; consecutive points may be equal. Delete node  $v$  and replace net  $(v, t_i)$  by  $(b_i, t_i)$ ,  $1 \leq i \leq 3$ .

*Proof.* We prove part (3) the other two cases being simpler. Consider a solution  $p_1, \dots, p_m$  for our routing problem. Assume w.l.o.g. that  $p_i$  is the path for net  $(v, t)$ ,  $1 \leq i \leq 3$ . We may assume w.l.o.g. that paths  $p_1, p_2, p_3$  do not cross. Hence path  $p_i$  passes through vertex  $b_i$  for  $1 \leq i \leq 3$ . Thus the modified problem is solvable.  $\square$

Lemma 1 allows us to simplify routing problems. In a simplified standard generalized switchbox routing problem (SSGS) there are no nodes  $v$  with  $\text{deg}(v) = 1$ , and all nodes  $v$  with  $\text{deg}(v) = 2$  ( $\text{deg}(v) = 3$ ) satisfy  $\text{ter}(v) = 0$  ( $\text{ter}(v) = 1$ ). Also nodes with  $\text{deg}(v) = 4$  satisfy  $\text{ter}(v) = 0$ . We will next characterize the form of critical cuts in SSGSs.

Let  $(R, N)$  be a solvable SSGS. We may index the vertices of  $R$  by integer coordinates. Let  $v$  be a left upper corner (i.e.,  $\text{deg}(v) = 2$  and the left and top neighbor of  $v$  do not exist) of  $R$  with maximal  $y$ -coordinate. Let  $w$  be the bottom neighbor of  $v$  (cf. Fig. 4). Note that no vertex of  $R$  has  $y$ -coordinate larger than  $v$ .

We consider critical cuts  $X$  with  $v \in X$  and  $w \notin X$  if there are any. Among these cuts we select one with  $|V(C) \cap X|$  minimal and among these cuts we select one with  $|X|$  minimal. We denote this cut by  $X_0$ . One main goal of this section is to show that  $X_0$  has a very simple form. Its boundary consists of at most two line segments (see Lemma 4 for a precise statement). We start with several simple observations.

(1)  $R|X_0$  is connected. Otherwise we could take as  $X_0$  the connected component of  $R|X_0$  containing  $v$ , a contradiction to the choice of  $X_0$ .

(2)  $R|X$  is a generalized switchbox. Assume otherwise. Let  $X' \supseteq X_0$  be obtained from  $X_0$  by filling the holes. Then  $v \in X'$ ,  $w \notin X'$ ,  $\text{dens}(X') = \text{dens}(X_0)$  and  $\text{cap}(X') < \text{cap}(X_0)$ . Thus  $X'$  is over-saturated and our routing problem is not solvable.

(3) Let  $v$  and  $v'$  be the endpoints of the path  $V(C) \cap X_0$ . Then every node  $x \in X_0 - \{v, v'\}$  has degree  $\geq 2$  in  $R|X_0$ . Assume otherwise. Con-

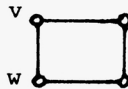


FIG. 4. Points  $v$  and  $w$ .

sider cut  $X' = X_0 - \{x\}$ . Then  $R|X'$  is still connected,  $V(C) \cap X'$  is still a path and hence  $x \notin V(C)$ ,  $\text{dens}(X') \geq \text{dens}(X_0)$  and  $\text{cap}(X') \leq \text{cap}(X_0)$ , a contradiction to the choice of  $X_0$ .

Consider the edges in  $\text{CAP}(X_0)$ , i.e., the edges with exactly one endpoint in  $X_0$ . We can view the “cut”  $X_0$  as a polygonal line  $S$  intersecting exactly the edges in  $\text{CAP}(X_0)$ . Line  $S$  consists of straight line segments  $s_1, s_2, \dots, s_k$  where  $s_1$  intersects the edge  $(v, w)$ .

LEMMA 3. *Each segment  $s_i$  intersects an edge  $e = (x, y) \in \text{CAP}(X_0)$  such that either  $x$  or  $y$  lies on the boundary  $C(R)$  of  $R$ .*

*Proof.* The claim is certainly true for segments  $s_1$  and  $s_k$ . Assume now that there is a segment  $s_i$ ,  $1 < i < k$ , which cuts no edge incident to a node on the boundary. Assume w.l.o.g. that  $s_i$  is vertical and that  $X_0$  is to the right of  $s_i$ . Then  $s_{i-1}$  and  $s_{i+1}$  are horizontal.

*Case 1.* Either  $s_{i-1}$  or  $s_{i+1}$  extends to the right of  $s_i$ . Then we can move  $s_i$  one unit to the right and obtain a cut  $X'$  with  $\text{dens}(X') = \text{dens}(X_0)$ ,  $|X'| < |X_0|$ ,  $\text{cap}(X') \leq \text{cap}(X_0)$  and  $|V(C) \cap X'| = |V(C) \cap X_0|$ , a contradiction to the minimality of  $X_0$ .

*Case 2.*  $s_{i-1}$  and  $s_{i+1}$  extend to the left of  $s_i$ . Then we can move  $s_i$  one unit to the left and obtain a cut  $X'$  with  $\text{dens}(X') = \text{dens}(X_0)$  and  $\text{cap}(X') = \text{cap}(X) - 2$ . Thus  $X'$  is oversaturated, a contradiction to our global assumption that we deal with a solvable problem.  $\square$

LEMMA 4. *Line  $S$  consists of at most two segments. In addition, if there are two segments then the angle  $\sphericalangle(s_1, s_2)$  is concave relative to  $X_0$ .*

*Proof.* Assume first that angle  $\sphericalangle(s_1, s_2)$  is convex relative to  $X_0$  (cf. Fig. 5). Then  $k = 2$ . Since  $\text{ter}(v) = 0$  and  $\text{ter}(x) = 1$  for all other nodes  $x \in X_0$  cut  $X_0$  cannot be saturated, a contradiction.

This shows that  $k \geq 2$  implies that  $\sphericalangle(s_1, s_2)$  is concave relative to  $X_0$ . It remains to show that  $k \geq 2$ . Assume otherwise, i.e.  $k \leq 3$ . We have to distinguish two cases

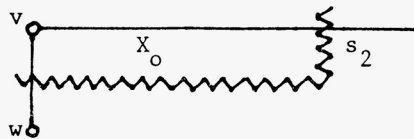


FIG. 5. Angle  $(s_1, s_2)$  is convex and  $k = 2$ .

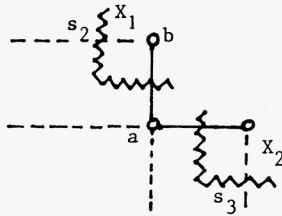


FIG. 6. The boundary  $C(R)$  is shown solid.

Case 1.  $\sphericalangle(s_2, s_3)$  is convex relative to  $X_0$ . We know from the proof of Lemma 3 that there are points in  $C(R)$  immediately to the right of  $s_2$ . Let  $a$  be the lowest such boundary point above  $s_3$ . Then either the point above or below  $a$  is also a boundary point.

Case 1.1. The point immediately below  $a$  is not a boundary point. Then the point above  $a$  is a boundary point; call it  $b$  (cf. Fig. 6).

We consider the two cuts as shown in Fig. 6. Note that cut  $X_2$  exists since vertex  $a$  was chosen as the lowest boundary point to the right of  $s_2$ . We have

$$\text{cap}(X) = \text{cap}(X_1) + \text{cap}(X_2)$$

and

$$\text{dens}(X) = \text{dens}(X_1) + \text{dens}(X_2) - 2 \text{dens}(X_1, X_2)$$

since vertex  $a$  has degree 4 (if  $a$  had only degree 3 or less then  $R$  would not be biconnected) and hence  $\text{ter}(a) = 0$ ,  $\text{dens}(X) \leq \text{dens}(X_1) + \text{dens}(X_2)$ .

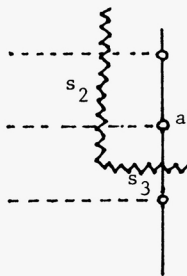


FIGURE 7

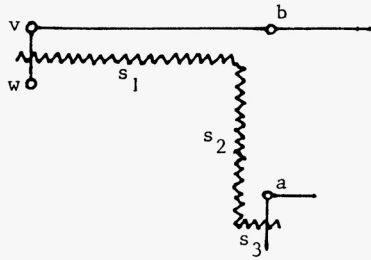


FIGURE 8

From  $\text{cap}(X) = \text{dens}(X)$  and  $\text{cap}(X_i) \geq \text{dens}(X_i)$  for  $i = 1, 2$  we conclude  $\text{cap}(X_i) = \text{dens}(X_i)$  for  $i = 1, 2$ . In particular,  $X_1$  is saturated. This contradicts the minimality of  $X_0$ .

*Case 1.2.* The point below  $a$  is a boundary point and hence  $s_3$  cuts only one edge (cf. Fig. 7). If the point above  $a$  is also a boundary point then we can certainly shorten cut  $X_0$  and still have a saturated cut, a contradiction. So let us assume that the point above  $a$  is not a boundary point. Let  $b$  be the boundary point which lies above  $a$  and is closest to  $a$ . Then the boundary  $C(R)$  either goes straight through  $b$  or bends in  $b$ .

*Case 1.2.1.* The boundary  $C(R)$  goes straight through  $b$  (cf. Fig. 8). Then  $b$  must lie in the top row of  $R$ . We consider the cut  $X'$  obtained by moving  $s_2$  one unit to the right (cf. Fig. 9). We have  $\text{dens}(X') = \text{dens}(X_0)$  since  $\text{ter}(a) = 0$  (note that  $\text{deg}(a) = 4$ ). Also  $\text{cap}(X') = \text{cap}(X_0)$  and hence  $X'$  is saturated. This contradicts the minimality of cut  $X_0$ .

*Case 1.2.2.* The boundary  $C(R)$  bends in vertex  $b$  (cf. Fig. 10). Consider cuts  $X_1$  and  $X_2$  as indicated in Fig. 11. We have  $\text{cap}(X_0) = \text{cap}(X_1) + \text{cap}(X_2)$  and  $\text{dens}(X_0) = \text{dens}(X_1) + \text{dens}(X_2) -$

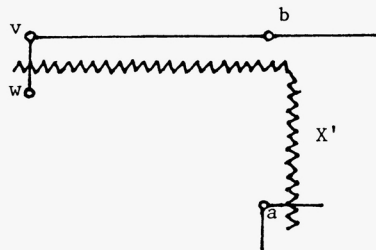


FIGURE 9

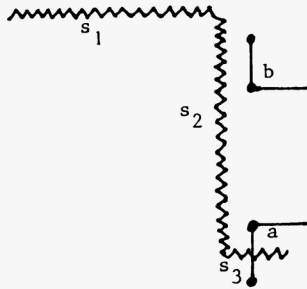


FIGURE 10

$2 \text{ dens}(X_1, X_2) \leq \text{dens}(X_1) + \text{dens}(X_2)$ . Thus  $\text{cap}(X_1) = \text{dens}(X_1)$ , a contradiction to the minimality of  $X_0$ .

*Case 2.*  $\prec(s_2, s_3)$  is concave relative to  $X_0$ . The proof of Lemma 3 implies that there is a boundary point immediately to the left of segment  $s_2$ .

*Case 2.1.*  $s_3$  cuts at least two edges. Then the boundary points to the left of  $s_2$  lie in  $h (\geq 1)$  segments as shown in Fig. 12. Let  $\ell_i$  be the number of vertices in the segment between  $a_i$  and  $b_i$  inclusive,  $1 \leq i \leq h$ . Note that  $\text{deg}(a_i) = \text{deg}(b_i) = 4$  and hence  $\text{ter}(a_i) = \text{ter}(b_i) = 0$ . We consider cuts  $X_1, \dots, X_{h+1}$  as shown in Fig. 13. We have

$$\text{cap}(X_0) = \text{cap}(X_1) + \dots + \text{cap}(X_{h+1}) + \sum_{i=1}^h (\ell_i - 2) + 2$$

since  $\ell_i - 2$  horizontal edges are not cut anymore in the  $i$ th segment and two vertical edges are not cut anymore. These edges are indicated as dashed

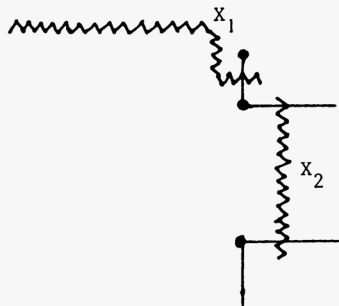


FIGURE 11

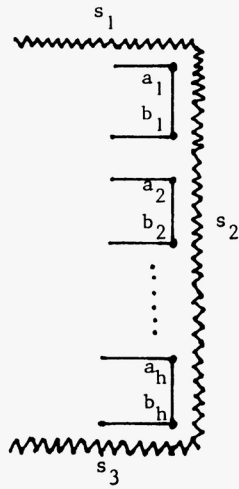


FIGURE 12

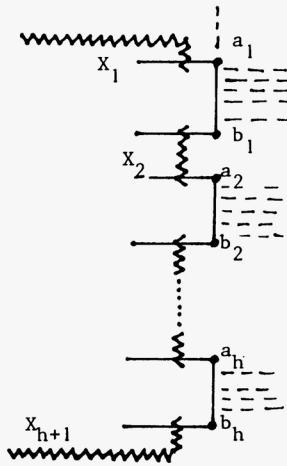


FIGURE 13

lines in Fig. 13. Also

$$\text{dens}(X_0) \leq \text{dens}(X_1) + \dots + \text{dens}(X_{h+1}) + \sum_{i=1}^h (\ell_i - 2)$$

since every net which goes across cut  $X$  also goes across one of the cuts  $X_i$  or has a terminal in one of the vertical segments between  $a_i$  and  $b_i$ . Since  $\text{cap}(X_0) = \text{dens}(X_0)$  and  $\text{cap}(X_i) \geq \text{dens}(X_i)$ ,  $1 \leq i \leq h + 1$ , we conclude

$$\begin{aligned} \text{cap}(X_1) + \dots + \text{cap}(X_{h+1}) + \sum_{i=1}^h (\ell_i - 2) + 2 \\ &= \text{cap}(X_0) \\ &= \text{dens}(X_0) \\ &\leq \text{dens}(X_1) + \dots + \text{dens}(X_{h+1}) + \sum_{i=1}^h (\ell_i - 2) \\ &\leq \text{cap}(X_1) + \dots + \text{cap}(X_{h+1}) + \sum_{i=1}^h (\ell_i - 2), \end{aligned}$$

a contradiction.

*Case 2.2.*  $S_3$  cuts exactly one edge. Then the situation is as shown in Fig. 14. Let  $\ell_i$  be the number of vertices between and including  $a_i$  and  $b_i$ ,  $1 \leq i \leq h - 1$ , let  $\ell_h$  be the number of vertices below and including  $a_h$  and

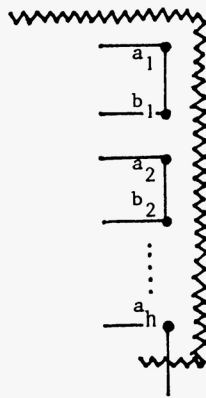


FIGURE 14



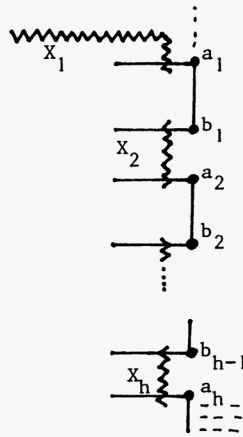


FIGURE 15

above  $s_3$ . Consider cuts  $X_1, \dots, X_h$  as shown in Fig. 15. We have

$$\text{cap}(X_0) = \text{cap}(X_1) + \dots + \text{cap}(X_h) + \sum_{i=1}^{h-1} (\ell_i - 2) + (\ell_h - 1) + 2$$

and

$$\text{dens}(X_0) \leq \text{dens}(X_1) + \dots + \text{dens}(X_h) + \sum_{i=1}^{h-1} (\ell_i - 2) + \ell_h - 1.$$

As in Case 2.1 we can now derive a contradiction. This finishes the case analysis and proves Lemma 4.  $\square$

Lemma 4 is very crucial for the efficiency of our algorithm. It completely characterizes the form of the minimal critical cuts  $X_0$  through edge  $(v, w)$ .

#### 4. THE ALGORITHM

Let  $R$  be a generalized switchbox with  $n$  vertices and let  $N$  be a set of nets. Throughout this section we assume that  $(R, N)$  is a *standard* problem. The goal of this section to describe an algorithm which solves any standard generalized switchbox routing problem in time  $O(n(\log n)^2)$ .

The algorithm is a special case of the general multicommodity flow algorithm outlined in Section 2. It derives its speed from the clever use of the characterization of minimal critical cuts derived in Section 3. The algorithm processes the routing region  $R$  row by row starting at the top

row. In every step it considers a left upper corner in the top row, say  $v$ , and eliminates the vertical edge  $(v, w)$  incident to  $v$  as described in Section 2. There are two main tasks which we have to solve (efficiently).

(1) Find the minimal critical cut  $X_0$  through edge  $(v, w)$ , if there is any, and

(2) choose the appropriate net to route across cut  $X_0$ .

We use two data structures to solve these tasks efficiently. The first data structure is a range tree for the set of nets and is global to the algorithm. The second data structure is a priority queue for the free capacities of the cuts through edge  $(v, w)$  and is local to each row of the routing region. We assume that the vertices on the boundary  $C(R)$  of the routing region are numbered in clock-wise order by the integers in range  $[1 \dots M]$ .

As the algorithm proceeds vertices in  $C(R)$  are deleted (always a left-upper corner) and new nodes become boundary nodes. The new boundary vertices inherit the number from deleted vertices as shown in Fig. 16. In this way the numbering of the boundary vertices remains in increasing clockwise order. However, adjacent boundary vertices are not necessarily numbered by consecutive integers. From now on we identify nodes in  $C(R)$  with their number.

A net is represented as a pair of integers, namely by the pair of numbers associated with its terminals. The set  $N = \{(s_i, t_i); s_i \leq t_i, 1 \leq i \leq m\}$  of nets is stored in a range tree. Range trees were introduced by Lueker and Willard; see also Mehlhorn, Section VII.2.2. We briefly review range trees. Range trees consist of a primary tree and a set of secondary trees, one for each node of the primary tree.

In our case the primary tree is a *static* search tree for integers  $1, \dots, M$  of depth  $O(\log M) = O(\log n)$ . Let  $v$  be a node of the primary tree and let  $NL(v) = \{(s, t) \in N; \text{the leaf labeled } s \text{ is a descendant of } v\}$ . The secondary tree  $ST(v)$  associated with node  $v$  is a balanced tree (*AVL*-tree, *BB* $[\alpha]$ -tree, or ...) for the ordered multiset  $\{t; (s, t) \in NL(v)\}$ . In every node of  $w$  of a secondary tree we store two auxiliary fields: the first field contains the number of leaf descendants of  $w$  and the second field contains



FIG. 16. How numbers are inherited, (b)  $y$  inherits  $v$ 's number.

the maximal  $s$  such that  $\text{net}(s, t) \in N$  is stored in that secondary tree and the leaf  $t$  is a descendant of  $w$ . It is clear that a range tree requires space  $O(m \log M) = O(n \log n)$  since every net belongs to  $O(\log M)$  node lists. It supports the following operations in time  $O(\log n)^2$ .

(1) Insert a net into  $N$  or delete a net from  $N$ .

(2) Given  $a, b, c, d$  finds nets  $(s, t) \in N$  and  $(s', t') \in N$  with  $a \leq s, s' \leq b, c \leq t, t' \leq d$  and  $t$  maximal or  $s'$  maximal, respectively. These nets can be found as follows: Consider the search paths for  $a$  and  $b$  in the primary tree and let  $C_{\max}$  to be the roots of the maximal subtrees of the primary tree between these paths. Then every net  $(s, t) \in N$  with  $a \leq s \leq b$  belongs to  $NL(v)$  for exactly one node  $v \in C_{\max}$ . Also  $|C_{\max}| = O(\log M)$ . For every node  $v \in C_{\max}$  we search for  $C$  in the secondary tree  $\text{ST}(v)$  and find the maximal  $t(v)$  and  $s'(v)$  such that  $c \leq t(v) \leq d, c \leq t'(v) \leq d$  and  $(s(v), t(v)) \in NL(v)$  and  $(s'(v), t(v)) \in NL(v)$ . To find  $t(v)$  we only have to inspect the leaf immediately to the left of the search path for  $d$  and in order to find  $s'(v)$  we have to inspect the auxiliary fields of the nodes between the search paths to  $c$  and  $d$ . Finally comparing  $(s(v), t(v))$  and  $(s'(v), t'(v))$  for all  $v \in C_{\max}$  allows us to find the desired nets  $(s, t)$  and  $(s', t')$ .

(3) Given  $a < b$  find the number of nets  $(s, t) \in N$  with either  $a \leq s \leq b < t$  or  $s < a \leq t \leq b$ . Let  $n_1 = |\{(s, t); a \leq s \leq b < t\}|$  and  $n_2 = |\{(s, t); s < a \leq t \leq b\}|$ . We can determine  $n_1$  as follows;  $n_2$  is determined similarly. Define  $C_{\max}$  as above. For every node  $v \in C_{\max}$  compute  $|\{(s, t) \in NL(v); b < t\}|$  in time  $O(\log n)$  by a search in  $\text{ST}(v)$  using the auxiliary information associated with the nodes.

The local data structures for the rows will be described below. We give the algorithm first.

- (1) initialize the range tree for the set  $N$  of nets
- (2) **while** routing region nonempty
- (3) **do** consider a top row of the routing region;
- (4) initialize the local data structure for the current row;
- (5) **while** row nonempty
- (6) **do** let  $v$  be the left corner of the row, let  $w$  be its bottom neighbor and let  $x$  be its right neighbor;
- (7) **if**  $\text{ter}(v) = \text{deg}(v)$
- (8) **then** route as given by Lemma 2 and delete  $v$ ;
- (9) update data structures
- (10) **else** find minimal critical cut  $X$  through edge  $(v, w)$ ;
- (11) **if** this cut does not exist
- (12) **then** delete node  $v$  add net  $(x, w)$
- (13) and update data structures

```

(14)         else find net (s, t) to be routed across
(15)         cut  $X_0$ ,  $s \in X_0$ ,  $t \notin X_0$ ; delete vertex  $v$ ;
(16)         delete (s, t) from the set of nets and add nets
              (x, s) and (w, t);
(17)         update data structures
(18)       fi
(19)     fi;
(20)   split routing region if it is not biconnected anymore;
(21) od
(22) od
    
```

We will next describe the local data structure for each row. Let  $L$  be the length of the top row. We consider cuts consisting of one horizontal segment and one vertical segment or of only a horizontal segment. Let  $X_i$  be the cut where the horizontal segment intersects exactly  $i$  edges of the routing region (cf. fig. 17). For every  $i$  let

$$fcap(X_i) = cap(X_i) - dens(X_i)$$

be the free capacity of cut  $X_i$ . We have to execute the following operations on  $fcap(X_i)$ ,  $1 \leq i \leq L$ .

- (1) compute  $fcap(X_i)$ ,  $1 \leq i \leq L$  to initialize the local data structure in line (4)
- (2) find the maximal  $i$  with  $fcap(X_i) = 0$  to find the minimal critical cut  $X_0$  through edge  $(v, w)$  in line (10).
- (3) decrease  $fcap(X_i)$  by two for  $a \leq i \leq b$  to update the local data structure in lines (9), (13), and (18).

We show first how to do the first task in time  $O(L(\log n)^2)$ . Consider cut

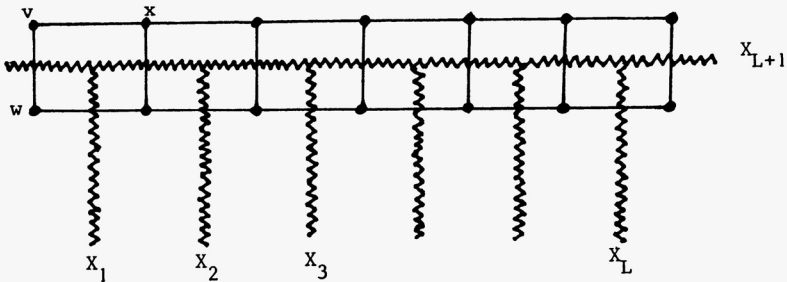


FIG. 17. The cuts  $X_i$ ,  $1 \leq i \leq L$ .

$X_i$ . We know that  $X_i \cap V(C)$  is a path and hence the numbers of vertices  $X_i \cap V(C)$  form an interval  $[h, j]$  with  $h < j$  or two intervals  $[h, M], [1, j]$  with  $h > j$ . Note that  $h$  is the number of vertex  $v$ . The integer  $j$  is easily found by computing in a preprocessing step for every vertex  $U$  of  $R$  the highest vertex below  $u$  which lies on  $C(R)$ . The capacity of cut  $X_i$  is now readily computed in time  $O(1)$  by adding the lengths of its constituting segments. The density of cut  $X_i$  is computed in time  $O((\log n)^2)$  using the third property of range trees derived above.

It remains to show how to solve the other two tasks. We use priority queues with updates as described in Galil-Naamad; see also Mehlhorn, Section IV.9.1. They allow us to perform these tasks in time  $O(\log n)$  each.

We will next discuss the lines of our algorithm in more detail. Lines (1) and (4) were already described. In line (8) we route as given by Lemma 2. Let  $(v, t_1)$  and  $(v, t_2)$  be the two nets having  $v$  as a terminal with  $v, t_1, t_2$  in clockwise order on  $C(R)$ . Let  $i_0$  be maximal such that  $t_1$  and  $t_2$  both belong to  $X_i$  for  $i \leq i_0$ . Then  $\text{fcap}(X_i)$  decreases by two for  $i \leq i_0$ . We also have to delete two nets from  $N$  and add two other nets. Thus the cost of line (8) is  $O(\log n + (\log n)^2) = O((\log n)^2)$ . Line (10) takes time  $O(\log n)$  by property (2) of the local data structure. In line (12) we have to add one net to  $N$  and to reduce  $\text{fcap}(X_i)$  by two for all  $i$ .

In line (13) we first have to find the net  $(s, t)$  which has to be routed across and cut  $X_0$ , i.e.,  $s \in X_0, t \notin X_0$  and  $t$  is as close to  $w$  as possible. Since  $X_0 \cap V(C)$  is a path the boundary nodes in  $X_0$  form an interval  $[h, j]$  with  $h < j$  or two intervals  $[h, M]$  and  $[1, j]$  with  $h > j$ . In the former case net  $(s, t)$  is either the net  $(s', t')$  with  $s' < h' \leq t' \leq j$  and  $s'$  maximal or the net  $(s'', t'')$  with  $h \leq s'' \leq j \leq t''$  and  $t''$  maximal. In the latter case the net  $(s, t)$  is either the net  $(s', t')$  with  $s' < h \leq t' \leq M$  and  $s'$  maximal or the net  $(s'', t'')$  with  $1 \leq s'' \leq h < t'' < j$  and  $t''$  maximal.

In line (16) we have to delete one net from  $N$  and add two other nets for a cost of  $O((\log n)^2)$ . In line (17) we have to change  $\text{fcap}(X_i)$  for some cuts  $X_i$ . Let  $(s, t)$  be the net to be routed across  $X_0$ . Let  $i_0$  and  $i_1$  be such that  $s, t \notin X_i$  for  $i \leq i_0$  and  $s, t \in X_i$  for  $i \geq i_1$ . Then  $\text{fcap}(X_i)$  decreases by two for  $i \leq i_0$  and  $i \geq i_1$ . This change requires time  $O(\log n)$ .

We finally have to discuss line (20). Let  $y$  be the "diagonal" neighbor of vertex  $v$  (cf. Fig. 17). Then  $y$  (and only  $y$ ) may become an articulation point by the removal of  $v$ . Vertex  $y$  becomes an articulation point if  $y$  belongs to  $C(R)$  before the removal of  $v$ , i.e., if  $y$  was numbered prior to the removal of  $v$ . Thus it is easy to test whether the routing region has to be split.

We split the routing region by finding the nets  $(s_i, t_i)$  which have to go through  $y$  using property (3) of the range tree and by replacing them in nets  $(s_i, y), (y, t_i)$ .

We then apply the algorithm separately to both parts. It is important to observe that we can use the *same* global data structure for both parts and that we can continue to process the current row, using the current local data structure.

This concludes the description of the algorithm and its data structures. The analysis of the running time is also easily completed at this point. All lines except line (4) takes time  $O((\log n)^2)$  and eliminate one vertex. Line (4) takes time  $O(L(\log n)^2)$  where  $L$  is the length of the current row; i.e., time  $O((\log n)^2)$  per vertex. Thus total running time is  $O(n(\log n)^2)$ . We summarize in

**THEOREM 1.** *Let  $(R, N)$  be a standard generalized switchbox routing problem with a routing region of  $n$  vertices. Then a solution (if there is one) can be constructed in time  $O(n(\log n)^2)$ .*

## 5. NONSTANDARD ROUTING PROBLEMS

This section is devoted to nonstandard routing problems. We show how to find efficiently a solution for a nonstandard GSRP if there is one.

We review the next two basic lemmas from the paper of Becker-Mehlhorn; the proofs can be found there.

**LEMMA 5.** *Let  $(R, N)$  be a nonstandard GSRP which has a solution. Then there is a solvable standard GSRP  $(R, N')$  where  $N' = N \cup P$  and  $P$  is a pairing of  $U = \{v; v \text{ has odd extended degree}\}$ .*

We call  $(R, N')$  a standard extension of  $(R, N)$ . Our extension is based on the concepts of *U-minimal cut* and *canonical extension*.

Let  $X$  be a saturated cut and let  $u_1, u_2, \dots, u_{2k}$  be the clockwise ordering of  $X \cap U$ . The cut  $X$  is *U-minimal* if  $X \cap U \neq \emptyset$  and there is no simple saturated cut  $Y$  with  $Y \cap U = \{u_i, u_{i+1}, \dots, u_j\}$  with  $1 < i < j < 2k$ . The canonical extension of  $(R, N)$  with respect to  $X$  is obtained by adding nets  $(u_{2i-1}, u_{2i})$ ,  $1 \leq i \leq k$ . Note that adding these nets will make the extended degrees of all vertices in  $X$  even.

**LEMMA 6.** *Given a solvable nonstandard GSRP. An iterative application of canonical extension with respect to U-minimal cuts leads to a solvable standard GSRP.*

Lemma 6 leads to the following algorithm for turning a nonstandard problem into a standard problem.

- (1)  $U \leftarrow \{v; \text{extended degree of } v \text{ is odd}\}$
- (2)  $U \leftarrow U_0$
- (3) **while**  $U \neq \emptyset$  **do**

- (4)       **if** there is an oversaturated cut  
 (5)           **then** terminate and declare that the problem has no  
                   solution  
 (6)       **fi**  
 (7)       let  $X$  be a  $U$ -minimal cut ( $X = V$  is possible)  
 (8)       construct the canonical extension  
 (9)        $U \leftarrow U - X$   
 (10) **od**

Becker–Mehlhorn showed how to implement this algorithm in time  $O(bn + |U_0|^2) = O(bn)$  where  $b$  is the number of vertices on the boundary of the infinite face. Their algorithm works for arbitrary planar graphs where every interior node has even degree. The algorithm consists of two phases.

(1) In phase one the free capacity of all cuts  $X$  is determined which can conceivably become  $U$ -minimal during the extension of the algorithm. This phase takes  $O(bn)$  and builds up data structure of size  $O(|U_0|^2)$  to be used in the second phase.

(2) In phase two, the algorithm above is used to construct the standard extension. Phase two takes time  $O(|U_0|^2)$ .

We will show how to execute phase one in time  $O(n(\log n)^2)$  in our case. This will give an  $O(n(\log n)^2 + |U_0|^2)$  algorithm for solving nonstandard problems.

The main idea for the improved running time is the following: We may assume w.l.o.g. that  $U$ -minimal cuts have a very restricted form. Let  $X_0$  be a  $U$ -minimal cut. As in Section 3 we can view  $X_0$  as a polygonal line  $S$  intersecting exactly the edges in  $\text{CAP}(X_0)$ . Line  $S$  consists of several straight line segments. We claim that two suffice.

**LEMMA 7.** *Let  $(R, N)$  be a solvable generalized switchbox routing problem with  $U$  as its set of vertices of odd extended degree. Then there is a  $U$ -minimal cut  $X_0$  consisting of at most two straight-line segments.*

*Proof.* If  $V$  is a  $U$ -minimal cut then the claim is certainly true. Assume otherwise. Choose a  $U$ -minimal cut  $X_0$  consisting of straight line segments  $s_1, \dots, s_k$  with  $k$  minimal. Note that  $\emptyset \neq X_0 \cap U \neq U$  since  $V$  is not  $U$ -minimal. If  $k \leq 2$  then we are done.

So let us assume finally that  $k \geq 3$ . We may assume w.l.o.g. that  $s_1$  is horizontal and the left end of  $s_1$  intersects the boundary of  $R$ . Then  $s_2$  intersects an edge of  $R$  whose left endpoint lies of the boundary of  $R$ . As in the proof of Lemma 4 we distinguish two cases.

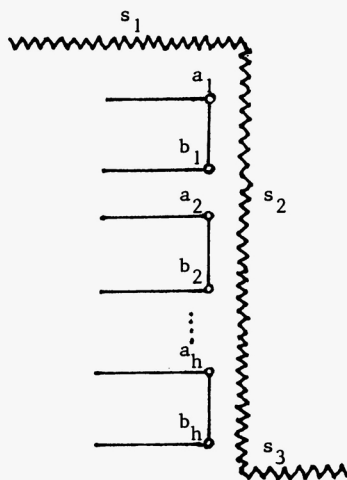


FIGURE 18

Case A.  $s_3$  runs to the right as seen from the lower endpoint of  $s_2$ .

Case Aa. extending  $s_3$  for one segment of the left does not intersect a boundary edge (cf. Fig. 18). Then the boundary points to the left of  $s_2$  lie in  $h$  ( $\geq 1$ ) segments as shown in Fig. 18.

Let  $\ell_i$  be the number of vertices in the segment between  $a_i$  and  $b_i$  inclusive,  $1 \leq i \leq h$ . Note that  $\deg(a_i) = \deg(b_i) = 4$  and hence  $\text{ter}(a_i) = \text{ter}(b_i) = 0$ . We consider cuts  $X_1, \dots, X_{h+1}$  as shown in Fig. 19. Let  $o_i$  be the number of vertices of odd extended degree in the segment between  $a_i$  and  $b_i$ . We have

$$\text{cap}(X_0) = \text{cap}(X_1) + \dots + \text{cap}(X_{h+1}) + \sum_{i=1}^h (\ell_i - 2)$$

and

$$\text{dens}(X_0) \leq \text{dens}(X_1) + \dots + \text{dens}(X_{h+1}) + \sum_{i=1}^h (\ell_i - o_i - 2).$$

Since  $\text{cap}(X_0) = \text{dens}(X_0)$  and  $\text{cap}(X_i) \geq \text{dens}(X_i)$  for all  $i$  (we deal with a solvable problem) we conclude that  $o_i = 0$  for all  $i$ ,  $1 \leq i \leq h$ , and  $\text{cap}(X_i) = \text{dens}(X_i)$  for  $1 \leq i \leq h + 1$ . Since  $o_i = 0$  for all  $i$  we conclude further that  $U \cap X_0 = (U \cap X_1) \cup \dots \cup (U \cap X_{h+1})$  and hence one of the cuts  $X_i$  is  $U$ -minimal. This contradicts the choice of cut  $X_0$ .

Cases Ab ( $\equiv$  not case Aa) and B (not Case A) are similar and are left to the reader.  $\square$



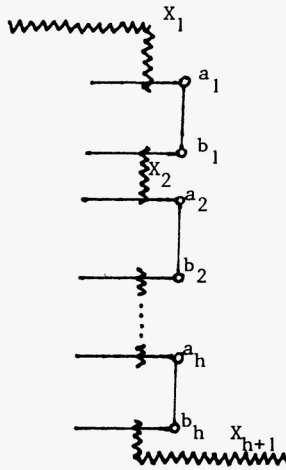


FIGURE 19

Lemma 7 tells us that we need only consider cuts with at most one bend when searching for  $U$ -minimal cuts. Let  $e = (x, y)$  be an edge on the boundary of  $R$  and let  $\ell(e)$  be the length of a cut through  $e$  which consists of a single straight line segment.

Clearly  $\sum \ell(e) \leq O(n)$  where the sum is over all edges on the boundary of  $R$ . Also there are only  $2\ell(e)$  cuts through  $e$  with exactly one bend. Hence only  $O(n)$  cuts must be considered as candidates for  $U$ -minimal cuts. For every single cut we can compute its free capacity in time  $O(\log n)^2$  as shown in Section 4. Thus time  $O(n(\log n)^2)$  suffices to compute the information required for the second stage of the algorithm in Becker-Mehlhorn). We summarize in

**THEOREM 2.** *Nonstandard routing problems with  $n$  vertices and  $U$  vertices of odd extended degree can be solved in time  $O(n(\log n)^2 + |U|^2)$ .*

*Proof.* By the discussion above one can extend the nonstandard problem to a standard problem in time  $O(|U|^2 + n(\log n)^2)$ . The standard extension can be solved in time  $O(n(\log n)^2)$  by Theorem 1.

### CONCLUSION

We exhibit a routing algorithm for two-terminal nets in generalized switchboxes. The algorithm runs in time  $O(n(\log n)^2)$  and finds a solution—if there is one—in the case of standard problems. Several open questions remain.

- (1) Can the running time be improved?
- (2) Can we also solve non-standard problems optimally in time  $O(n(\log n)^2)$ ?
- (3) Can one extend the result to more general routing regions and/or multiterminal nets?

## REFERENCES

1. M. BECKER AND K. MEHLHORN, Routing and edge-disjoint paths in planar graphs, *Acta Informatica*, to appear.
2. M. BRADY AND D. BROWN, "VLSI Routing: Four Layers Suffice," MIT VLSI Conference, 1984.
3. A. FRANK, Disjoint paths in rectilinear grids, *Combinatorica* **2**, (4) (1982), 361–371.
4. M. R. KRAMER AND J. VAN LEEUWEN, "Wire Routing is NP-Complete," Technical Report RUU-CS-82-4, Utrecht, 1982.
5. K. MEHLHORN AND F. PREPARATA, Routing through a rectangle, *JACM*, **33**, (1986), 60–85.
6. H. OKAMURA AND P. D. SEYMOUR, Multicommodity flows in planar graphs, *J. Combin. Theory Ser. B* **31** (1981), 75–81.
7. F. PREPARATA AND W. LIPSKI, "Three Layers are Enough," 23rd FOCS, pp. 350–357, 1982.