Journal of Circuits, Systems, and Computers, Vol. 6, No. 1 (1996) 1–13

AN ALGORITHM FOR STEINER TREES IN GRID GRAPH AND ITS APPLICATION TO HOMOTOPIC ROUTING*

MICHAEL KAUFMANN
Wilhelm-Schickard-Institut, Universität Tübingen, Sand 13
72076 Tübingen, Germany

SHAODI GAO
Department of ECE, Concordia University
Montreal, Quebec, Canada H3G 1M8

K. THULASIRAMAN
School of Computer Science, University of Oklahoma
Norman, Oklahoma 7301-90631, USA

Received 15 December 1993
Revised 21 November 1994
Accepted 25 November 1994

In this paper we present an algorithm for Steiner minimal trees in grid graphs with a terminals located on the boundary of the graph. The algorithm runs in $O(\min\{k^4, k^2n\}$ time, where k and n are the numbers of terminals and vertices of the graph, respectivel It can handle non-convex boundaries and is the fastest known for this case. We also con sider the homotopic routing problem and apply our Steiner tree algorithm to construc minimum-length wires for multi-terminal nets.

1. Introduction

Given a set K of vertices, called *terminals*, in a graph $G(V, E)$, the Steiner problem is to find a minimum-length tree whose vertex set includes all term in K. The minimum-length tree is called a *Steiner minimal tree*, while ve with degree ≥ 3 in the tree are called *Steiner vertices*. This problem has extensively studied for many years because of its wide variety of applications, as communication networks and VLSI layout design. In general, the proble known to be NP-complete.[7] Dreyfus and Wagner[5] gave a dynamic program algorithm for the problem with time complexity $O(n3^k + n^2 2^k + n^3)$, where $|G|$ and $k = |K|$. By specializing this general approach, Provan[15] showed the Steiner tree problem can be solved in polynomial time if G is a planar g and all terminals are located on the boundary of one face of G. His algor runs in time $O(n^2k^2)$. Erickson, Monma and Veinott[6] reduced the complexi $O(nk^3 + (n \log n)k^2)$.

*This paper was recommended by M. Marek-Sadowska.

In this paper we consider a special case of the Steiner tree problem in · (1) G is a grid graph with no holes, i.e., every finite face has exactly four inc vertices, and (2) all terminals are located on the boundary P of G, i.e., the bou of the infinite face (cf. Fig. 1). A grid graph without holes is also called a *gener switchbox*, which is used to formulate many VLSI routing problems.[11] Withou of generality, we assume that (1) G is biconnected, i.e., P is a simple polygon (2) every boundary corner with the inner angle equal to 90° (*convex* corner) terminal on it. The case in which G is not biconnected can be solved by partiti G and then solving several biconnected-graph instances. For the second assum non-terminal convex corners can be removed from G because they are not nece for constructing a Steiner minimal tree. Then P has at most $2k$ corners be there are no more non-convex corners than convex corners. We allow P t *non-convex*, i.e., it may contain two or more consecutive non-convex corners.

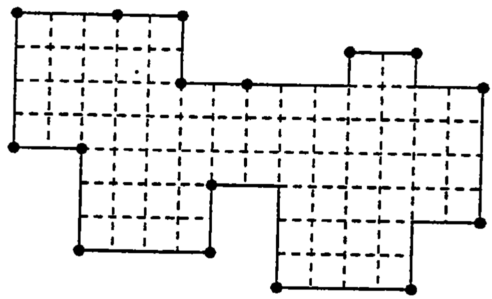


Fig. 1. An example of the Steiner tree problem in a grid graph with terminals (solid dots) on the boundary.

In the following we present an algorithm for this special case. In Sec. 3 we a general description of the dynamic programming approach proposed by Dre and Wagner.[5] In Sec. 4 we introduce a restricted type of subtrees which ca constructed more efficiently. Section 5 gives more details about the complexit the algorithm. Our algorithm runs in $O(\min\{k^4, k^2n\})$ time and $O(\min\{k^4, k^2$ space. Richards and Salowe[16] developed an $O(k\nu^4)$-time algorithm, where ν is number of the boundary sides of the graph. However, their algorithm can handle grid graphs with convex boundaries.

In Sec. 6 we extend our result to construct a collection of Steiner minimal t in a grid graph, which is allowed to have holes. While terminals of the Ste trees may lie on the boundary of the graph as well as on those of the holes, topology of each tree is given. This problem is called *homotopic routing* in V layout design. The goal is to find vertex-disjoint Steiner minimal trees for the gi collection of terminal sets. Homotopic routing was first introduced by Leisers Maley,[12] but they only dealt with problems where each terminal set has cardina of 2. We present an efficient algorithm for terminal sets of cardinality ≥ 2 by us the Steiner tree algorithm described in Secs. 2 to 4.

2. The Dynamic Programming Algorithm

Since the graph boundary P is a simple polygon and all terminals lie on it, define an *interval* $[a, b]$ of K to be the set of terminals, including a and b, by traversing P counterclockwise from a to b. Interval $(a, b]$ or $[a, b)$ is s defined except that a or b is not included. The following lemma can be f Ref. 15 and is essentially due to Erickson *et al.*[6]

Lemma 1: Let K be a set of terminals lying on the boundary of a plana G, and T a Steiner tree for K in G. The removal of any edge $e = (u, v)$ splits T into two subtrees $T(e, u)$ and $T(e, v)$ such that the terminals in eac subtrees form an interval of K.

Lemma 1 is used in Refs. 6 and 15 to design recursive equations for namic programming algorithm. For each interval $[a, b]$ of K and vertex $v \in$ $C(v, [a, b])$ represent the length of a Steiner minimal tree for terminal set $[a,$ and let $B(v, [a, b])$ represent the minimum length of a Steiner tree for the sa minal set subject to the constraint that v has degree ≥ 2 in the tree. $B($ can be computed as the sum of the lengths of two Steiner minimal trees fo subintervals of $[a, b]$. That is

$$B(v, [a, b)) = \min_{a \neq x \in [a,b]} \{C(v, [a, x)) + C(v, [x, b])\}.$$

A Steiner minimal tree for $(v, [a, b])$ consists of a path from u to v (u and v identical) and a Steiner minimal tree for $(u, [a, b])$ with degree$(u) \geq 2$ or $u \in$ Let $d(u, v)$ denote the shortest distance between u and v in G. $C(v, [a, b])$ computed as follows.

$$C(v, [a, b]) = \min \left\{ \min_{u \in V}\{B(u, [a, b]) + d(u, v)\}, \min_{u \in \{a,b\}} \{C(u, [a, b] \backslash \{u\}) + d(\right.$$

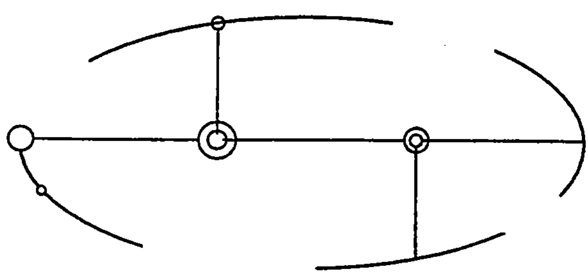


Fig. 2. Decomposition of a Steiner minimal tree. The sizes of the circles indicate the ord decomposition.

The computation of the B- and C-values proceeds in order of the card of the interval $[a, b]$. The initial conditions are $C(v, \emptyset) = 0$ and $B(v, \emptyset) = 0$

$v \in V$. At the end of the computation, the length of a Steiner minimal tree will be $C(v, K\backslash\{v\})$ for any $v \in K$. The tree itself can be recovered by retaining a record of the trees corresponding to the B- and C-values. The number of B- and C-values to be computed is of the same order as the number of possible choices of vertices $v \in V$ and intervals $I \subseteq K$, which is $O(k^2 n)$. A simple-minded approach requires $O(k)$ time for computing a B-value and $O(n)$ time for a C-value, which leads to a total running time of $O((n+k)nk^2)$. In the following we introduce a restricted type of subtrees whose length can be computed more efficiently.

3. The Restricted Subtrees

First we give some necessary notions. A *line* in a graph (G or its subgraphs) is a maximal line segment, i.e., no collinear extension in the graph is possible. A line may be subject to further restrictions, e.g., a *line from* a vertex v is a maximal line segment starting at v. In rectilinear graphs, a line from a vertex can have one of four directions, coded with the numbers 1 to 4. Vertices and lines on boundary P are called *boundary* vertices and lines; all others are *interior* vertices and lines.

We define a *reduced graph* G_K of G by deleting all grid lines of G which do not contain terminals in K or boundary corners of G. The vertices in G_K are the intersection points of the grid lines in G_K. Hanan[9] proved that a Steiner minimal tree for K in G_K is also a Steiner minimal tree for K in G. Therefore, we only need to consider the reduced graph G_K. It should be noted that the boundary P of G remains the boundary of G_K, because each boundary line contains a corner. The vertices of G_K on P are called *nodes*. There are at most $4k$ nodes altogether, and the number of vertices in G_K is $O(\min\{n, k^2\})$.

Definition 1: For any vertex $v \in V$ and interval $[a, b] \subseteq K$, a Steiner tree T is saic to be a *restricted tree* or an *R-tree* if it satisfies the following two conditions.

(i) Every line in T contains a node; every line from v also contains a node.

(ii) For every node u with $deg(u) \geq 2$ in T, $[a, b] \cup \{u\}$ is an interval of $K \cup \{u\}$.

An R-tree is called an *R_i-tree* if it contains a line from v pointing in direction i Similarly, an *R_{ij}-tree* is an R-tree which contains lines from v pointing in directio i and j.

For a Steiner tree, an *interior component* is a connected component of the tre after removing all the boundary lines. Hwang[10] proved that any interior componen of a Steiner minimal tree can be transformed without increasing the length into on of the two types depicted in Fig. 3. For Type 1, all the Steiner vertices lie o one line and the other lines incident to the first line point alternatively in th opposite directions. For Type 2, the Steiner vertices lie on two lines, which form *interior corner*, such that one of the lines has at most one Steiner vertex and th

Every line in T has at least one end on the boundary, and hence satisfies the fi condition of the R-tree. The second condition holds trivially. For any terminal with $deg(v) = 1$, T is an R-tree for $(v, K \setminus \{v\})$.

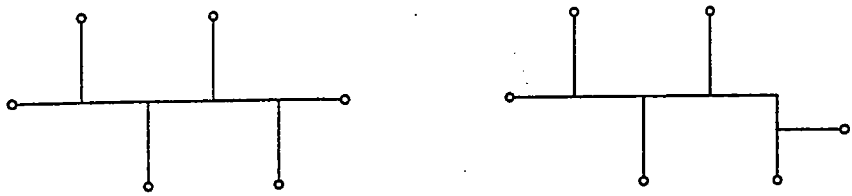


Fig. 3. The two types of interior components. The circles indicate nodes.

The following rules will be applied to break ties among the Steiner minimal tre First, choose the Steiner minimal trees whose *node degree*, i.e., the total degree the nodes in the tree, is maximal. Among the Steiner minimal trees with t maximum node degree, choose those whose interior components only have the t types depicted in Fig. 3. We called a Steiner minimal tree satisfying the tie-breaki rules an *optimal Steiner tree.* An R-tree for an interval of K is said to be *optin* if it is a subtree of an optimal Steiner tree for K. The length of an optimal R_i-tı or R_{ij}-tree for $(v, [a, b])$ will be denoted by $C_i(v, [a, b])$ or $B_{ij}(v, [a, b])$, respective In the following two lemmas we show that there is an optimal Steiner tree for which can be recursively decomposed into optimal R_i- and R_{ij}-trees for interv of K.

Lemma 2: If there is an optimal R_i-tree for $(v, [a, b])$, then there is an optimal R tree which is composed of a path p from v to u and an $R_{i'}$-tree for $(u, [a, b] \setminus \{u$ with $u \in [a, b]$ or an $R_{i'j'}$-tree for $(u', [a, b])$ with the line from u' to u pointi in direction i'. Path p consists of up to three interior lines and a sequence consecutive boundary lines between the interior lines.

Proof: Let T be an optimal R_i-tree for $(v, [a, b])$. If $deg(v) \geq 2$, then it is a an R_{ij}-tree. Otherwise, let p be the path in T from v to the first vertex u wi $u \in [a, b]$ or $deg(u) \geq 3$. If $u \in [a, b]$, then $T \setminus p$ obviously satisfies the conditions the R-tree and hence is an $R_{i'}$-tree for $(u, [a, b] \setminus \{u\})$. If $deg(u) \geq 3$, we distingui between two cases: (1) the last segment of p is an entire line of T, (2) it is a pa of an interior corner. In the first case every line in $T \setminus p$ has a node. If u lies on interior corner, let u' be the bending point of the corner. Then $T \setminus p$ is an optin $R_{i'j'}$-tree for $(u', [a, b])$, and it contains a line from u through u' if $u \neq u'$. In t second case, let w be the bending point. The line between u and w in $T \setminus p$ does r contain any node. We *flip* the corner bending at w to a new corner bending at u' depicted in Fig. 4. This operation does not change the length. It does not chan the node degree either because w is not a node and T has the maximal node deg as an optimal R-tree. That means u' is not a node either. Since T is an optin

R-tree, the resulting tree T' from T is a subtree of an optimal Steiner t
Now $T' \setminus p$ has the same properties as the $T \setminus p$ in the first case, and hen
optimal $R_{i'j'}$-tree for $(u', [a, b])$.

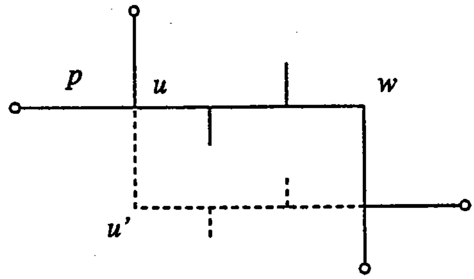


Fig. 4. Transform an optimal R-tree to another by flipping the corner.

Let $u_1, \ldots, u_m$ be the nodes on p excluding u and $p(u_1, u_m)$ the corr subpath of p. Further, let $P(u_1, u_m)$ be one of the two boundary parts b and u_m which does not contain terminals in $K \setminus [a, b]$. This assumption i because $[a, b] \cup \{u_1, u_m\}$ is an interval of $K \cup \{u_1, u_m\}$ as required of definition of the R-tree. If $P(u_1, u_m)$ contains any terminal, it belongs to must be connected to T. Such a connection has to cross $p(u_1, u_m)$ becaus formed by $P(u_1, u_m)$ and $p(u_1, u_m)$ separates the terminal from u. That c the assumption that $p \setminus \{v, u\}$ does not contain Steiner vertices or term $deg(v) = 1$. That means $P(u_1, u_m)$ may only have non-convex corners convex corner always has a terminal. Therefore the shortest path $p(u_1, u_m$ u_1 and u_m is identical to $P(u_1, u_m)$. Finally, since every line in p excep ending at u must contain a node, p can have at most three interior lines: v to u_1, one from u_m and the third one ending at u.

Now we redefine $B(u, [a, b])$ to be the minimum length of the R_{ij} $(u', [a, b])$ which contains a line from u' to u in direction i or j. That is

$$B(u, [a, b]) = \min_{1 \leq i,j \leq 4} \left\{ \min_{u'u \in \{i,j\}} \{B_{ij}(u', [a, b])\} \right\}$$

Further, $C(u, [a, b])$ is redefined to be the minimum of $C_i(u, [a, b])$ over Then we can use the right-hand side of Eq. (2) to compute $C_i(v, [a, b])$ constraint that the path from v to u has the property of Lemma 2 an segment points in direction i.

Lemma 3: An optimal R_{ij}-tree T for $(v, [a, b])$ can be split into an optim for $(v, [a, x))$ and R_j-tree for $(v, [x, b])$. The separating terminal x can be d by i, j and v.

Proof: Let $e_r = (v, u_r)$ with $1 \leq r \leq deg(v)$ be the edges in T incident t
From v, one of these edges points in direction i and another in direction j.
Lemma 1, the removal of these edges split T into $deg(v)$ subtrees $T(e_r, u_r)$ v
each connecting an interval of K. We can combine these intervals into two inter
I_i and I_j of K such that they are connected by two subtrees T_i and T_j o
containing the lines from v in directions i and j, respectively. It should be nc
that v is not included in I_i or I_j. To satisfy the second condition of Definitio
in case v is a terminal, we have to make sure that if $deg(v) \geq 2$ in any of the
subtrees, say T_i, then $I_i \cup \{v\}$ is an interval of K. To see that the condition holds
any other node u with $deg(u) \geq 2$, just imagine it as a terminal and consider T
and T_j as Steiner subtrees for $K \cup \{u\}$. Then the above argument can show t
$I_i \cup \{u\}$ or $I_j \cup \{u\}$ is an interval of $K \cup \{u\}$. Every line in T_i and T_j is either a
or a line from v in T, which contains a node because T is an R-tree. That me
T_i and T_j also satisfy the first condition of the R-tree. Therefore, the length of
optimal R_{ij}-tree for $(v, [a, b])$ can be calculated by the following equation simila
Eq. (1).

$$B_{ij}(v, [a, b]) = \min_{a \neq x \in [a, b]} \{C_i(v, [a, x)) + C_j(v, [x, b])\}$$

Let y and z be the first nodes on the lines from v in directions i and j, res
tively. Any R_{ij}-tree for $(v, [a, b])$ contains y and z. Furthermore, $[a, b] \cup \{y, z$
an interval of $K \cup \{y, z\}$ by Definition 1. Therefore, there is an interval $[y, z$
$K \cup \{y, z\}$ that does not contain any terminals in $K \setminus [a, b]$. The lines from v to y
z completely separate the terminals in $[y, z]$ from those in $[a, b] \setminus [y, z]$ (cf. Fig.
Only the terminals in $[y, z]$ determine the subtree that spans them and v. T
means the separating terminal x in the minimal R_{ij}-tree for $(v, [y, z])$ can be u
to split the R_{ij}-tree for any $(v, [a, b])$ as long as $[y, z] \setminus [a, b] = \emptyset$ holds.

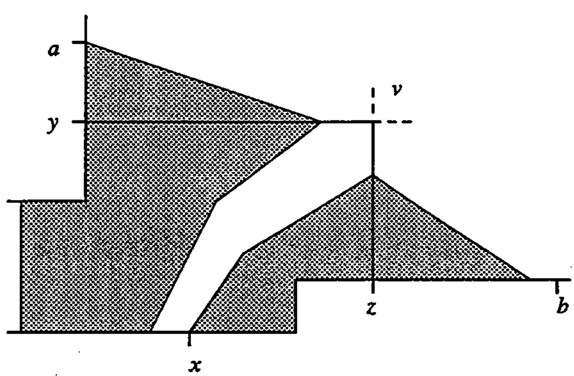


Fig. 5. An optimal R_{ij}-tree for $(v, [a, b])$ and its separating terminal.

Nodes y and z in the above lemma may or may not be terminals in K. Howe
$[y, z]$ is an interval of $K \cup \{y, z\}$, which is called a *special* interval for vertex v. Tl

are up to ten special intervals for vertex v because the two lines emana can form ten different angles: four of 90°, two of 180° and four of 270°. computation of B- and C-values we will also consider the special interv

4. Computation of the B- and C-Values

The computation of $B_{ij}(v,[a,b])$ and $C_i(v,[a,b])$ is carried out in the order of the interval size. The values for the same interval $[a,b]$, but fo choices of v, are computed in one step.

For the computation of $B_{ij}(v,[a,b])$, we distinguish between two case is a special interval of v, and (2) it is not. In the first case, we have possible terminals in $[a,b]$ to find the separating terminal x which mir sum $C_i(v,[a,x)) + C_j(v,[x,b])$. It takes $O(k)$ time for every $(v,[a,b])$ in Since there are $O(n)$ choices of v and each of them have $O(1)$ specia the entire computation for the first case takes $O(kn)$ time. In the se the separating terminal x is that of the corresponding special interval. calculation of the B_{ij}-value for a non-special interval $[a,b]$ and a verte completed in $O(1)$ time. The number of vertices v is $O(\min\{k^2,n\})$, and t of intervals $[a,b]$ is $O(k^2)$. Therefore, the time for the computation of all is $O(\min\{k^4,k^2n\})$.

Following an idea proposed by Erickson *et al.*,[6] the computation of $B(u,[a,b])$ for one interval $[a,b]$ and all vertices $u \in V_K$ can be consi problem of finding all single-source shortest paths. For each direction create a directed graph G'_K from G_K by deleting all grid edges perpen direction i, giving the remaining edges direction i, and then adding a source s, from which there is an arc to each vertex $u \in V$. The cost of th s to u is the minimum of $B_{ij}(u,[a,b])$ over $1 \le j \le 4$; the cost of any oth G'_K is an acyclic graph, the shortest paths from s to all u can be found i linear to the number of edges and vertices in G_K, which is $O(\min\{k^2,$ length of the shortest path from s to u represents the minimum of $B_{ij}(u'$ $1 \le j \le 4$ and $\mathbf{u'u}$ in direction i. The value $B(u,[a,b])$ can be determined shortest path algorithm is performed for all four directions.

Similarly, $C_i(v,[a,b])$ can be computed from $B(u,[a,b])$ in four itera for each interior line or the sequence of boundary lines in the path from each iteration four different directions are computed separately as discus except that the cost of each grid edge is its length instead of 0. In the for the sequence of boundary lines, only boundary lines appear in G'_K separate directions are considered: clockwise and counterclockwise. For iteration, the cost of the arc from s to u is $B(u,[a,b])$; for any of the iterations, the cost is the shortest length from s to u from the last iterati iteration takes $O(\min\{k^2,n\})$ time. Therefore, the total running time for C is $O(\min\{k^4,k^2n\})$.

Lemma 4: If all terminals are located on the boundary of a grid graph without holes, then the problem of finding a Steiner minimal tree in the graph can be solved in time $O(\min\{k^4, k^2n\})$ and space $O(\min\{k^4, k^2n\})$.

5. Homotopic Planar Routing of Multi-Terminal Nets

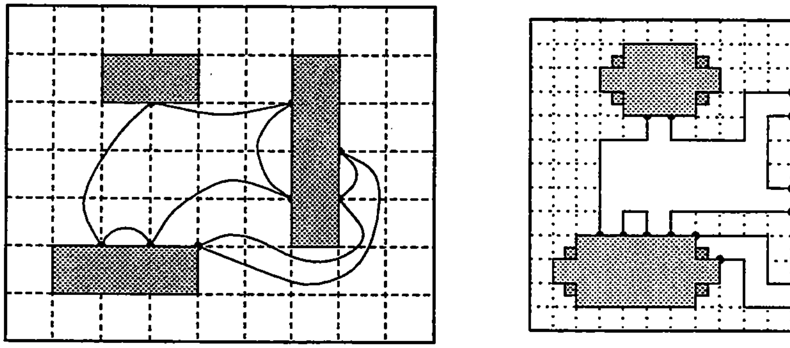
In this section we apply the above described Steiner tree algorithm to construct minimum-length interconnections for a collection of terminal sets in a grid graph. In this case, the grid graph may contain holes, i.e., finite faces enclosed by more than four grid edges. Terminals are located on the boundary of the graph as well as on those of the holes. The interconnection topology for each terminal set is given. This problem is called *homotopic routing.* In the homotopic planar routing interconnections for different terminal sets must be vertex-disjoint (routing on one layer). Homotopic routing can handle different types of routing areas, even areas containing holes, while other routing methods only deal with very restricted routing areas such as channels and hence require partitioning of routing areas and interconnections. Therefore, homotopic routing has found more and more applications in VLSI layout design.[4,13]

The first algorithms for homotopic planar routing were proposed by Cole–Siegel[5] and Leiserson–Maley.[12] Maley[14] later established a general theory on homotopic planar routing. However, these algorithms can only deal with the case of 2-terminal nets. An idea of splitting each multi-terminal nets into a ring of 2-terminal nets was put forward in Ref. 12 and detailed in Ref. 8. In this section we propose a routing algorithm which constructs minimum-length solutions for multi-terminal nets by means of Steiner minimal trees. We first employ the results of Ref. 8 and Ref. 12 to divide the routing area into a set of disjoint subregions where the interconnections for individual nets are to be accommodated. We show that the underlined grid graph in each subregion is connected and contains a Steiner minimal tree for the corresponding net. The grid graph does not contain any holes and all terminals lie on its boundary. Therefore, our Steiner tree algorithm can be applied to find a minimum-length connection for each multi-terminal net.

5.1. *Definitions and previous results*

The problem of homotopic planar routing is given by a *sketch* $S = (M, W)$ which consists of a set M of rectilinear polygons, called *modules*, and a set W of *nets* that interconnect *terminals* on module boundaries. Modules are placed on a rectilinear grid so that module boundaries are aligned with grid edges and terminals are located on grid vertices. The grid graph $G = (V, E)$ formed by grid vertices and edges which are not covered by the modules is called the *routing graph* of the sketch. The goal is to construct a detailed routing for S, which is a set of vertex-disjoint Steiner trees for the input nets. To describe the net topology, each k-terminal net is represented by a set of k curves (called *subnets*) which form a simple ring by intersecting the k terminals (cf. Fig. 6(a)). Except for the terminals, this ring may not cross o

enclose any modules. The subnets are two-terminal nets, and have ha
of the original net. Any other representations (including trees) for mu
nets can be transformed to a ring of two-terminal nets by slicing the r
centerline.



(a) (b)

Fig. 6. (a) A sketch S that contains multi-terminal nets. (b) A detailed routing in th sketch S'. Grid edges in envelopes are omitted.

A sketch is *routable* if there is a detailed routing for it. The rou be determined by a so-called cut condition. A *cut* X is an open-endec connects two module points and intersects no other modules. The *flo* is half the number of crossings of X by nets which are necessary a the topology of the sketch. The *capacity* of X is the maximum of t of the horizontal and vertical grid lines which X crosses. A cut X is flow(X) = capacity(X) and *oversaturated* if flow(X) > capacity(X). A similarly defined except that one of its endpoints is on a net. It is prove and 14 that the routability is equivalent to the non-existence of oversatu Based on the cut condition, Leiserson–Maley proposed an efficient al testing the routability. They also developed an algorithm to determi routings for routable sketches. Let $|M|$ denote the number of terminals corners, $|W|$ the number of line segments that represent the net topolog summarize their results in the following lemma.

Lemma 5: A sketch $S = (M, W)$ which contains two-terminal nets if and only if there is no oversaturated cut. A detailed routing can l $O(|M||W| \log |M||W|)$ time and $O(|M||W|)$ space. The solution has th properties:

(i) Every net has the minimum length; and

(ii) For every net segment, there is a saturated half cut that ends on it

5.2. *The routing algorithm for multi-terminal nets*

To transform a problem instance with multi-terminal nets into one with two-termin nets, we adopt the idea in Ref. 8. Every grid line l in the routing graph G of S replaced by a pair of lines which is parallel to and 1/4 unit away from the orig l. At the same time, modules are stretched in all the four directions (left, rigl up and down) by 1/4 unit except for the convex corners, which are flipped to no convex ones (cf. Fig. 6(b)). For every terminal t in G, we create two terminals 1 unit away from the origin t on the new module boundary, while a corner termin is replaced by two terminals on the new, neighboring corners. Any k-terminal ne which is represented by a ring of two-terminal nets in S, is split into k separat and intersection-free two-terminal nets. Let S' be the resulting sketch and G' i routing graph. The length of edges in G' is 1/2 unit, while nets in S' also have h width. Therefore, S' can be considered as a sketch only containing two-termin nets. The transformation preserves the routability: for each oversaturated cut in one sketch there is an oversaturated cut X' in the other sketch whose endpoir are next to those of X.

Now we can apply the results for two-terminal nets by Leiserson–Maley test the routability and to find a detailed routing for S' if it is routable. In t solution the two-terminal nets which are subnets of a multi-terminal net, togeth with the edges on module boundaries, form a rectilinear polygon (cf. Fig. 6(b This polygon is called the *envelope* of the multi-terminal net. Envelopes of diffe ent multi-terminal nets are area-disjoint, i.e., the boundary segment of the envelop do not cross each other and no envelope encloses any other envelopes. This is b cause the routing algorithm for two-terminal nets does not change the topology S' and it constructs vertex-disjoint paths. Each envelope encloses a subgraph of which will be used to find a Steiner tree for the corresponding net.

Lemma 6: Every envelope U encloses a connected part of G, which contains Steiner minimal tree for the corresponding multi-terminal net.

Proof: Because U is a simple polygon, the only possibility that the enclosed pa of G is not connected is that two parallel segments of U are next to each otl and have different origins. According to Lemma 5, there is a saturated half c X that crosses the both segments. It is not possible for a saturated cut to cro two segments of an envelope consecutively, while the two segments have differe origins.

Let T be a Steiner minimal tree for the corresponding net. As mentioned befo T can also be considered as a ring of subnets which connect the terminals in t same order as the subnets of U does. The length of T is half of the total leng of its subnets, because each edge of T is shared by two subnets. If T does not totally within U, then there is a subnet p of U crossing a subnet q of T. Since and T have the same topology, there is an even number of crossings of p and q. I $p(u, v)$ ($q(u, v)$) denote the part of p (q) between two points u and v. We call $q(u$

a *outer path* of T if u and v are two consecutive crossings of q by p, and outside of U in the immediate vicinity of u and v. Every outer path $q(u$ can be replaced by a path of G which is 1/4 unit away from $p(u, v)$. As T is transformed to a Steiner tree T' lying totally within U. The replacem not increase the total length of the subnets length, because p is a short according to Lemma 5. On the other hand, T' has the same property a every tree edge is shared by two subnets. This means the length of T' is als the total length of its subnets, and hence is not larger than that of T. T T' is a Steiner minimal tree lying totally within U.

Lemma 6 shows that finding minimum-length interconnections for a se can be treated as a set of separate instances of the Steiner tree probler Steiner minimal tree is in a grid graph enclosed by the envelope of the n the terminals are located on the boundary of the graph. Since the envelope contain any modules, the grid graph does not have holes. Therefore, the a described in the previous sections can be applied to Steiner minimal tree case. It takes $O(k^2 n)$ time for a k-terminal net in a grid graph with n For an input sketch $S(M, W)$, let $|K|$ denote the total number of termi the number of the vertices in the routing graph $G = (V, E)$. Then Stei algorithm can find minimum-length solutions for all the nets in $O(|K|^2|V$ The routing algorithm for a two-terminal net requires $O(|M||W| \log |M||V$ according to Lemma 5. $|M|$ is the total number of module corners and te i.e., $|M| \geq |K|$, while $|W|$ can be expected to have the same order of $|V|$. other steps can be carried out in $O(|V|)$ time.

Lemma 7: The problem of homotopic planar routing for multi-terminal be solved in $O(|M||W| \log |M||W| + |K|^2|V|)$. In the solution, the length net is minimized.

6. Conclusion

We have presented an algorithm for finding Steiner minimal trees in grid This algorithm can also handle non-convex boundaries, and is faster than viously known algorithms for this case. We also apply the algorithm to con collection of Steiner minimal trees for the homotopic routing problem. Ou show that any Steiner tree problem in grid graphs can be solved in polynom if the topology is given.

For the case that the boundary of a grid graph is convex, the algori Richards and Salowe[16] can be more efficient if the number boundary sides smaller than the number of terminals. An obvious open question is how tc their techniques in the case of non-convex boundaries.

Acknowledgment

Thanks are due to an unknown reviewer for suggestions which led to substantia
improvements of the paper.

References

1. M. W. Bern, "Network design problems: Steiner trees and spanning k-trees", Ph.I Dissertation, University of California, Berkeley, 1987.
2. M. W. Bern, "Faster exact algorithms for Steiner trees in planar networks", *Networl* **20** (1990) 109–120.
3. R. Cole and A. Siegel, "River routing every which way, but loose", *Proc. of the 25t Symp. on Foundations of Computer Science*, 1984, pp. 65–73.
4. W. W. Dai, T. Dayan, and D. Staepelaere, "Topological routing in SURF: Generatin a rubber-band sketch", *Proc. of the 28th Design Automation Conf.*, 1991, pp. 39–4
5. S. E. Dreyfus and R. A. Wagner, "The Steiner problem in graphs", *Networks* **1** (197 196–207.
6. R. E. Erickson, C. L. Monma, and A. F. Veinott, "Send-and-split method for minimun cost network flows", *Math. Oper. Res.* **12** (1987) 634–664.
7. M. R. Garey and D. S. Johnson, "The rectilinear Steiner tree problem is NP-complete' *SIAM J. Appl. Math.* **32** (1977) 826–834.
8. R. I. Greenberg and F. M. Maley, "Minimum separation for sigle-layer channel rou ing", *Information Process. Lett.* **43** (1992) 201–205.
9. M. Hanan, "On Steiner's problem with rectilinear distance", *SIAM J. Appl. Math.* **1** (1966) 255–265.
10. F. K. Hwang, "On Steiner minimal trees with rectilinear distance", *SIA J. Appl. Math.* **30** (1976) 104–114.
11. M. Kaufmann and K. Mehlhorn, "Routing through a generalized switchbox", *J. Alg rithms* **7** (1986) 510–531.
12. Ch. Leiserson and F. M. Maley, "Algorithms for routing and testing routability planar VLSI layouts", *Proc. of the 17th Symp. on Theory of Computing*, 198 pp. 69–78.
13. F. M. Maley, "Compaction with automatic jog introduction", *Proc. of the 1985 Chap Hill Conf. on VLSI*, 1985, pp. 261–284.
14. F. M. Maley, *Single-Layer Wire Routing and Compaction*, MIT Press, Cambridg MA, 1990.
15. J. S. Provan, "Convexity and the Steiner tree problem", *Networks* **18** (1988) 55–72
16. D. S. Richards and J. S. Salowe, "A linear-time algorithm to construct a rectiline Steiner tree for k-extremal points", *Algorithmica* **7** (1992) 246–276.