Routing Problems in Grid Graphs

Michael Kaufmann and Kurt Mehlhorn

1. Introduction

The routing problem lies at the heart of VLSI design. A routing problem is given by a routing region and a set of nets. In this paper the routing region will always be a grid graph, i.e., a finite subgraph of the integer grid.

Definition. The integer grid has vertices $x \in \mathbb{Z}^2$ and edges $\{x, y\}$ where $x = (x_1, x_2)$, $y = (y_1, y_2)$ and $|x_1 - y_1| + |x_2 - y_2| = 1$.

Grid graphs model the popular constraint that wires in a VLSI layout can only run horizontally and vertically in a natural way. A net or demand is specified by a pair of vertices (also called terminals of the net); a rough routing (also called global routing or homotopy) may also be specified. A solution to a routing problem called layout consists of a set of grid paths, one for each net, such that the following two conditions hold:

1) The paths are pairwise edge-disjoint.
2) For each net, the path for this net connects the two terminals of the net. In addition, if a rough routing for the net is specified, then the path must be homotopic to the rough routing; cf. Section 3 for a definition of homotopy.

A routing problem is naturally viewed as a multi-commodity flow problem. Each net represents the demand to send one unit of flow of a certain commodity from one terminal to the other terminal of the net; also each edge has capacity one. The additional constraint is that a commodity has to be sent along a single path and cannot be split up into pieces. If all nets have the same terminals then Menger's theorem provides us with a solution: The number of edge-disjoint paths is given by the capacity of a minimum cut. In other words, a set of demands can be satisfied, if there is no oversaturated cut, i.e., the cut condition holds. Okamura and Seymour extend Menger's Theorem to multi-commodity flow problems in planar graphs where all terminals lie on the boundary of the *same* face. They show that the cut condition together with an evenness condition implies solvability. This theorem together with an algorithmic version of it is discussed in Section 2; cf. also Frank's paper in this volume. In Section 3 we then turn to the homotopic routing problem in grid graphs. The terminals are now allowed to lie on the boundary of many faces; however global routings have to be specified. Again, the cut condition together with evenness implies routability. Section 4 is then devoted to a discussion of the evenness condition. Section 5 deals with special cases of the problem of Section 2. It is shown that for grid graphs without holes, e.g. rectangles or convex polygons, faster algorithms than for the general case can be obtained. Section 6 discusses the routing problem for grid graphs with a single hole. Terminals are allowed to lie on both non-trivial faces (the outer face and the hole) and no rough routing is given.

Mathematical models frequently suppress important features of real life problems. This is also true for the routing problem as discussed in Section 2 to 6; in particular, *layer assignment* is not dealt with and *multi-terminal nets* are not treated. They are the subject of Sections 7 and 8.

2. Edge-Disjoint Paths in Planar Graphs

Problem. Planar Edge-Disjoint Paths Problem (PED).

Input:

a) An embedded planar graph $G = (V, E)$.
b) A set $\mathcal{N}$ of nets where each $N \in \mathcal{N}$ is a pair of vertices on the boundary of the unbounded face of G.

Output: A family $\{p(N); N \in \mathcal{N}\}$ of paths such that

1) if $N = \{s, t\}$ then $p(N)$ is a path with endpoints s and t.
2) $p(N)$ and $p(N')$ are edge-disjoint for $N, N' \in \mathcal{N}$, $N \neq N'$. □

Subsets $X \subseteq V$ are also called *cuts* in the sequel. For a subset $X \subseteq V$ we define the *capacity* $cap(X)$ of X as the number of edges having exactly one endpoint in X and the *density* $dens(X)$ of X as the number of nets having exactly one terminal in X, i.e.,

$$cap(X) = \{\{a,b\} \in E; |\{a,b\} \cap X| = 1\}$$
$$dens(X) = \{\{s,t\} \in \mathcal{N}; |\{s,t\} \cap X| = 1\}.$$

The *free capacity* of X is then given by $fcap(X) = cap(X) - dens(X)$. A cut X is *saturated* if $fcap(X) = 0$ and *oversaturated* if $fcap(X) < 0$. An edge-disjoint path problem $(V, E, \mathcal{N})$ is *even* if $fcap(X)$ is even for every cut X; it satisfies the *cut condition* if $fcap(X) \geq 0$ for all cuts X.

Theorem 1 (Okamura/Seymour). *Let $P = (V, E, \mathcal{N})$ be an even planar edge-disjoint path problem. Then P is solvable iff P satisfies the cut condition.*

For a proof of this result we refer the reader to the paper by A. Frank in this volume. The proof directly yields an algorithm which we now discuss. Let P be an even problem. The idea is to construct a sequence $P_0, P_1, \ldots$ of problems such that

1) $P_0 = P$;
2) P_{i+1} has one less edge than P_i;
3) P_i is even;

4) if P_i satifies the cut condition then P_{i+1} does;
5) if P_{i+1} is solvable then P_i is solvable.

The construction ends when the algorithm either detects a violation of the cut condition or reaches a problem $P_m = (V_m, E_m, \mathcal{N}_m)$ which has an empty set of nets and is therefore trivially solvable. The problem P_{i+1} is constructed from P_i as follows. Let $e_0, e_1, \ldots, e_{k-1}$ be the edges on the boundary of the outer face in clockwise order with $e_0 = \{a, b\}$. Let us call a cut X *simple* if there are precisely two edges e_j having exactly one endpoint in X and let us call a cut X *a cut through* e_0 if $a \notin X$ and $b \in X$. We now distinguish cases.

Case a): There is a simple cut through e_0 with negative free capacity: Stop and declare the problem unsolvable.

Case b): There is no saturated simple cut through e_0: Then P_{i+1} is obtained by deleting the edge e_0 and adding the net $\{a, b\}$ to the set of nets.

Case c): There is a saturated simple cut through e_0: Let X be a saturated simple cut through e_0 of minimal cardinality and let $\{s, t\}$ be a net such that $s \in X, t \notin X$ and t is as close as possible to a in a counterclockwise traversal of the boundary of G. We delete the edge e_0 and replace the net $\{s, t\}$ by the two nets $\{a, t\}$ and $\{b, s\}$. If $a = t$ then $\{a, t\}$ is not added and if $b = s$ then $\{b, s\}$ is not added. Case c) is illustrated by Figure 2.

In the above case distinction it is assumed that the first case which applies is taken. For the proof of correctness of this algorithm, we again refer the reader to the paper by A. Frank. The problem of implementation is discussed by Matsumoto/Nishizeki/Saito [MNS85] and Becker/Mehlhorn [BM85].

Theorem 2 [BM85]. *Let* $P = (V, E, \mathcal{N})$ *be an even planar edge-disjoint path problem with* $n = |V|$.

a) The solvability of P *can be decided in time* $O(n^2)$. *Moreover, within the same time a solution can be constructed, if there is one.*
b) If (V, E) *is a grid graph then time* $O(n^{3/2})$ *suffices.*

Open Problem. Improve the running time.

For the following Sections it is useful to have a more "topological" definition of cuts and nets. Consider the dual of the planar graph $G = (V, E)$ and let $M = \{F_{ext}\}$; here F_{ext} denotes the unbounded or external face. A cut C (in the new sense) is any non-trivial simple path in the dual having both of its endpoints in M. A cut C in the new sense induces a cut in the old sense as follows: Remove the edges intersected by C from G and let X be one of the two connected components obtained in this way. For two distinct cuts C_1 and C_2 starting with the dual of the same edge e_0 it is natural to define the ordering relation "*C_1 is left of C_2*" as follows. Traverse C_1 and C_2 starting at their common origin until

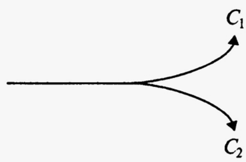


Fig. 1 C_1 is left of C_2

they separate. If at the point of separation the situation is as shown in Figure 1 then C_1 is left of C_2. With this definition, we can reformulate the selection of cut X in case c) as follows:

Cut Selection Rule. Let X be the leftmost saturated cut through e_0, i.e., X is saturated and X is left of any other saturated cut through e_0, cf. Figure 1.

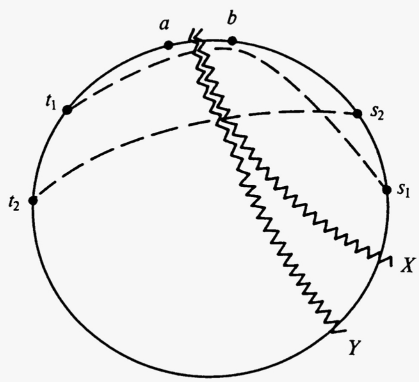


Fig. 2 X is left of Y, $\{a, t_1\}$ is right of $\{a, t_2\}$

Let us turn to the selection of the net N next. For $N = \{s, t\}, s \in X, t \notin X$ let us call the net $\{a, t\}$ the tail of the decomposition of N with respect to the edge $e_0 = \{a, b\}$. We may view the tail $\{a, t\}$ as a path from a to t running counterclockwise along the boundary of G and turning to the right at t. With this interpretation the selection of $\{s, t\}$ in part c) can be formulated as follows:

Net Selection Rule. Choose $N = \{s, t\}, s \in X, t \notin X$, such that the tail of the decomposition of N with respect to the edge e_0 is *rightmost*, cf. Figure 2.

3. Homotopic Edge-Disjoint Path Problems

We first state the problem and then define the concepts net, grid path and homotopy used in its definition.

Problem. Homotopic Routing Problem in Grid Graphs (HRP).

Input: A grid graph R and nets $q_1, \ldots, q_k$.

Output: Pairwise edge-disjoint grid paths $p_1, \ldots, p_k$ such that p_i is homotopic to q_i, $1 \leq i \leq k$, or an indication that no such paths exist. □

We call a bounded face F of R trivial if it has exactly four vertices on its boundary and nontrivial otherwise. We use M to denote the set of nontrivial bounded faces together with the unbounded face F_{ext} and $\mathcal{O}$ to denote the union of the interiors of the faces in M. A nontrivial face is also called a hole.

A path P is a continuous function $p : [0,1] \rightarrow \mathbb{R}^2 - \mathcal{O}$. A path p is called a *net* if $\{p(0), p(1)\} \subseteq V \cap \partial\mathcal{O}$ where $\partial\mathcal{O}$ is the boundary of $\mathcal{O}$. Two paths p and q are homotopic, denoted $p \sim q$, if there is a continuous function $F : [0,1] \times [0,1] \rightarrow \mathbb{R}^2 - \mathcal{O}$ such that $F(0,x) = p(x)$ and $F(1,x) = q(x)$ for all $x \in [0,1]$, and $F(t,0) = p(0)$ and $F(t,1) = p(1)$ for all $t \in [0,1]$. A path p is called a *grid path* if $p(x)$ belongs to R for all x. Fig. 3 gives an example of an HRP.

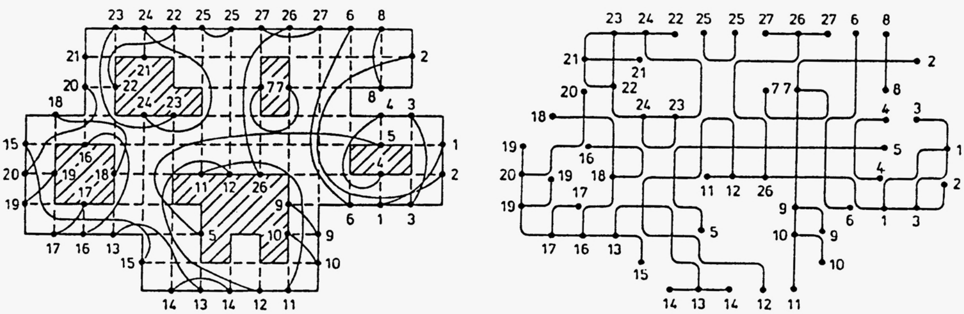


Fig. 3

Remark. In the previous Section all terminals had to lie on the infinite face. With $M = \{F_{ext}\}$ any two paths with the same endpoints are homotopic. It therefore sufficed to specify a net by its endpoints.

Theorem 3 [KM86]. *Let $P = (R, \mathcal{N})$ be an even bounded HRP. Here R is a grid graph and $\mathcal{N}$ a set of nets:*

$$P \text{ is solvable if and only if } fcap(X) \geq 0 \text{ for every cut } X.$$

As before, a cut is a simple path in the dual of R connecting two (not necessarily distinct) faces in M. The capacity $cap(C)$ of a cut C is the number of intersections with edges of R. If C is a cut and p is a path then $cross(p, C)$ is the number of intersections of p and C and $mincross(p, C) = \min\{cross(q, C); q \sim p\}$. Finally, the *density* $dens(C)$ of cut C is defined by

$$dens(C) = \sum_{p \in \mathcal{N}} mincross(p, C)$$

and the *free capacity* $fcap(C)$ is given by

$$fcap(C) = cap(C) - dens(C).$$

A cut C is *saturated* if $fcap(C) = 0$ and *oversaturated* if $fcap(C) < 0$.

An HRP is *even* if $fcap(C)$ is even for every cut C.

Let v be a vertex in R. We denote the degree of v by $deg(v)$ and the number of nets having v as an endpoint by $ter(v)$. An HRP is *bounded*, if $deg(v) + ter(v) \leq 4$ for all vertices v.

Remark. The definition of density given above extends the definition in the previous Section. Clearly, in the situation discussed there, $mincross(p, C) = 1$ if the endpoints of p lie on different sides of the cut C and $mincross(p, C) = 0$ otherwise.

The algorithm to solve HRP's is very similar to the algorithm given in the previous Section. We assume for simplicity that P satisfies the cut condition. We again construct a sequence $P_0, P_1, \ldots$ of HRP's such that

1) $P_0 = P$;
2) P_{i+1} has one less edge than P_i;
3) P_i is even and bounded;
4) if P_i satisfies the cut condition then P_{i+1} does;
5) if P_{i+1} is solvable then P_i is solvable.

The construction stops when a trivial problem is reached. The problem P_{i+1} is obtained from P_i as follows. Again we have to distinguish several cases. In all cases we use juxtaposition to denote concatenation of paths, i.e., if p and q are paths with $p(1) = q(0)$ then $pq(\lambda) = p(2\lambda)$ for $0 \leq \lambda \leq 1/2$ and $pq(\lambda) = q(2\lambda - 1)$ for $1/2 \leq \lambda \leq 1$. We also use a more careful definition for the ordering relation "right-of" on nets. If N is a net let $can(N)$ be the shortest path homotopic to N. We think of $can(N)$ as slightly extended into the incident non-trivial faces at both its terminals, cf. Figure 4a. Consider now two nets N_1 and N_2 with the same starting point. Then, if $N_1 \neq N_2$, $can(N_i)$ is not a prefix of $can(N_{3-i})$ for $i = 1, 2$. We say that N_1 is right-of N_2 if $can(N_1)$ and $can(N_2)$ separate as shown in Figure 4b.

We are now ready for the algorithm. Again we distinguish cases:

Case a): There is a cut X with $cap(X) = 1$: Let N be the unique net with $mincross(N, X) = 1$, let e be the edge intersected by X and let N_1 and N_2 be such that $N \sim N_1 e N_2$ and $mincross(N_1, X) = mincross(N_2, X) = 0$. Delete e and replace N by the nets N_1 and N_2.

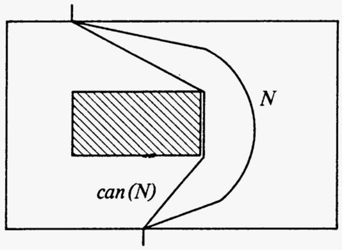


Fig. 4a $can(N)$

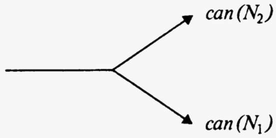


Fig. 4b The relation "right-of"

Case b): There is a vertex v with $deg(v) = ter(v)$. Let v be a vertex with $deg(v) = ter(v)$; let $e_i, 1 \leq i \leq 2$, be the edges incident to v and let $N_i, 1 \leq i \leq 2$, be the nets incident to v where the edges are numbered as shown in Figure 5 and N_1 is right-of N_2. Let $N_1 \sim e_1 N_1'$ where N_1' does not use edge e_1; remove edge e_1, reserve it for net N_1 and replace net N_1 by net N_1'.

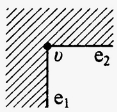


Fig. 5 A vertex v with $deg(v) = 2$ and the edges incident to it. The face in $\mathcal{O}$ is shown hatched

Case c): No cut of capacity one exists and $ter(v) < deg(v)$ for all v. Let vertex a be the left upper corner of the routing region, i.e. there is no vertex with either larger y-coordinate or smaller x-coordinate and same y-coordinate. Let b be the lower neighbor of a and $e^* = \{a, b\}$. The edge e^* plays the role of the edge e_0 in Section 2.

A cut is called a *straight-line cut* if it consists of a sequence of horizontal and vertical straight-line segments. For this definition to make sense, we view the dual as a grid graph. A straight-line cut is a *1-bend* cut if it consists of at most two segments. The 1-bend cuts play the role of the simple cuts in Section II.

Case c1): There is no saturated 1-bend cut through e^*: Remove edge e^* and add net N where N is the path from a to b following the boundary of the trivial face incident to e^*.

Case c2): There is a saturated 1-bend cut through e^*. We use the same cut selection rule as in Section II, i.e., we let X be the leftmost saturated 1-bend cut through e^*.

Let us turn to net selection next. For a net N with $mincross(N,X) > 0$, a decomposition with respect to X and e^* is a triple (N_1, e^*, N_2) such that $N \sim N_1 e^* N_2$ and $mincross(N_1, X) + mincross(N_2, X) = mincross(N, X) - 1$. A decomposition (N_1, e^*, N_2) is *rightmost* if N_2 is right-of M_2 for all decompositions (M_1, e^*, M_2) of nets M with $mincross(M, X) > 0$.

Net selection rule: Choose a net N with a rightmost decomposition (N_1, e^*, N_2) with respect to X and e^*.

Delete edge e^* and replace N by N_1 and N_2.

For the proof of correctness of this algorithm we refer the reader to [KM86]. We turn to the running time of the algorithm next. In order to measure the size of the input we assume that the nets are specified as polygonal paths. Then n denotes the number of vertices of the grid graph plus the total number of bends in the paths. In [KM86] an $O(n^2)$ algorithm to solve even, bounded HRP's was given. This was recently improved.

Theorem 4 [KM88]. *The solvability of even, bounded HRP's can be decided in linear time $O(n)$. Moreover a solution can be determined in time $O(n)$ if there is one.*

What can be done beyond grid graphs? [K87] showed that Theorem 3 is also valid for other types of grids, e.g. the grids shown in Figure 6. In these cases an instance is called bounded if $deg(v) + ter(v) \leq odeg(v)$ for all vertices v, where $odeg(v)$ is the degree of v in the infinite grid of the respective type.

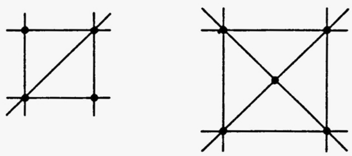


Fig. 6

Theorem 5 [K87]. *Theorem 3 is also valid for subgraphs of the grids shown in Figure 6. The runtime of the algorithm is $O(n^2)$.*

The most general type of graphs for which Theorem 3 is known to hold are *straight-line graphs*. Let G be an embedded planar graph and let $\mathcal{O}$ be the union of some of its faces (including the outer face) considered as open sets. The pair $(G, \mathcal{O})$ is called a straight-line graph if there are line segments $L_1, \ldots, L_t$ such that the endpoints of each L_i lie in the boundary of $\mathcal{O}$, such that the vertices of G are exactly the endpoints and intersections of the line segments, and such that the edges are exactly the induced fragments of the line segments. Boundedness is defined as above; the $odeg(v)$ of a vertex v on the boundary of a face in $\mathcal{O}$ is the degree of v plus the number of lines ending in v.

Theorem 6 [S87]. *Theorem 3 holds for straight-line graphs. Moreover, solvability can be decided in polynomial time.*

Open Problem. Design an algorithm for homotopic routing in straight-line graphs with quadratic or even linear running time.

Open Problem. Extend Theorem 3 beyond straight-line graphs. Note that the theorem does not hold for planar graphs as Figure 7 shows. What is the appropriate theorem for planar graphs?

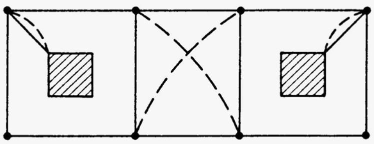


Fig. 7 Nets are indicated by dashed lines and finite non-trivial. This problem is even and satisfies the cut condition. It is not solvable, however

4. The Evenness Assumption

In Theorems 1 and 3 we assumed the routing problem to be even. Figure 8 shows that this assumption is crucial. What can be done without the evenness assumption?

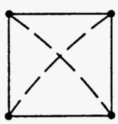


Fig. 8 An unsolvable, non-even problem satisfying the cut condition

For PED this problem was treated by Frank and Becker/Mehlhorn. Let us call an instance of PED half-even if the degree of every vertex v which does not border the outer face is even. Frank extends the cut condition so that it applies to half-even instances; we refer the reader to his paper for the details. Becker/Mehlhorn consider the algorithmic side.

Lemma. *Let P be an instance of PED. $fcap(X)$ is even for every cut X iff $deg(v) + ter(v)$ is even for every vertex v.*

We call $deg(v) + ter(v)$ the extended degree of v.

It is easy to see that each solvable half-even instance can be extended by adding some nets (called fictitious) which have vertices with odd extended degree as terminals, such that the new instance is even and solvable. Becker/Mehlhorn give an algorithm to find an appropriate extension of any given solvable half-even instance.

Theorem 7 [BM86]. *The solvability of half-even instances of PED can be decided in time $O(bn)$ where b is the number of vertices bordering the outer face.*

The main idea in their method is captured in the following definition:

Definition. Let P be a solvable half-even instance. Let U be the set of vertices with odd extended degree and let X be a saturated cut. Let $u_1, u_2, \ldots, u_{2k}$ be the vertices in $X \cap U$ in clockwise order along the boundary of the outer face.

a) X is *U-minimal* if $X \cap U \neq \emptyset$ and there is no saturated cut Y with $Y \cap U = \{u_i, \ldots, u_j\}$ where $1 < j - i + 1 < 2k$.
b) *The canonical extension* of P with respect to X is given by adding the nets (u_{2i-1}, u_{2i}), $1 \leq i \leq k$, to the set of nets in P.

The following lemma is crucial for the correctness of their method.

Lemma. *Let P be a solvable half-even instance. If X is a U-minimal cut, then the canonical extension of P with respect to X is a half-even solvable instance.*

The algorithm determines a U-minimal cut and extends the problem canonically with respect to this cut. This preserves solvability and reduces the size of U. After at most $O(b)$ iterations U is empty. Each iteration takes time $O(n)$.

If the routing region is a grid with no non-trivial inner face we can take advantage of the fact that we only need consider 1-bend cuts to find U-minimal cuts. Hence, each step can be executed in time $O(|U|)$.

Theorem 8 [KM85]. *Half-even instance of PED on grids with n vertices and no non-trivial inner face and $|U|$ vertices of odd extended degree can be extended to even problems in time $O(\log^2 n + |U|^2)$.*

Open Problem. For grid graphs, even instances can be solved in time $O(n^{3/2})$, but extending a half-even instance to an even instance takes time $O(bn)$; b might be as large as $\Omega(n)$. Find a faster algorithm for half-even grid graph problems.

For the homotopic routing problem in grid graphs the situation is more complicated. As before, we call an instance half-even, if the degree of v is even for every vertex not on the boundary of a non-trivial face. We call an instance locally even, if $deg(v) + ter(v)$ is even for every vertex v. If all terminals of nets are on the outer face then a locally even instance is necessarily even, but in general this is not the case.

As before we know that for any solvable half-even instance there exists an extension to a solvable locally even instance by adding some fictitious nets.

Furthermore every solvable locally even instance can be extended to a solvable even instance by adding some non-trivial circular nets. Circular nets are simple cycles of a certain homotopy (Kaufmann/Maley). Unfortunately, the extensions are not easy to find in this case.

Theorem 9 [KMa88]. *The homotopic routing problem in grid graphs is NP-complete for locally even instances.*

Kaufmann/Maley also obtained a positive result. Suppose that we allow to move modules by one unit.

Theorem 10 [KMa88]. *The homotopic routing problem with movable modules is solvable in linear time in the case of locally-even instances and is NP-complete in the case of half-even instances.*

5. Routing Regions Without Holes

In this Section we come back to the problem considered in Section 2, i.e. all terminals lie on the boundary of the outer face. The algorithm discussed in Section 2 solves this problem in time $O(n^2)$ for general planar graphs and $O(n^{3/2})$ for grid graphs. We now turn attention to grid graphs where all bounded faces are trivial. Such graphs are called generalized switchboxes. In the case of generalized switchboxes we only need to consider a special kind of cuts, namely straight cuts (they consist of only one straight path segment) or 1-bend cuts (consist of at most two straight segments).

Lemma [KM85]. *Let P be an even bounded instance of PED on a grid.*
a) If there is an oversaturated cut then there is an oversaturated 1-bend cut.
b) If there is an oversaturated cut then there is an oversaturated 0-bend cut or an oversaturated 1-bend cut connecting two concave corners.

A concave corner is a pair $((v', v), (v, v''))$ of boundary edges of some nontrivial face sharing a vertex v of degree 4. A 1-bend cut connects 2 concave corners $((v', v), (v, v''))$ and $((w', w), (w, w''))$ if the rectangle defined by the two corners v and w is non-empty and does not contain any boundary vertices except v and w. We call two such corners *rectilinear visible.*

Using the algorithm of Kaufmann/Mehlhorn for homotopic edge-disjoint paths we can solve each solvable generalized switchbox problem in time $O(n)$. However simpler and/or faster algorithms are known for some cases of generalized switchboxes.

Kaufmann/Mehlhorn [KM85] give an algorithm to solve PED for generalized switchboxes in time $O(n \log^2 n + |U|^2)$. It works for half-even instances; U is the set of the vertices with odd extended degree. Although the runtime is worse than the runtime of the algorithm for the homotopic case, the implementation is much less complicated. The algorithm is almost the same as in the planar case, but takes advantage of the fact that nets and cuts are given by their endpoints and that the capacities of 1-bend cuts are easily computed by the difference of the coordinates of their endpoints. Nets and possibly critical cuts are represented by intervals and are stored in a range tree. This data stucture supports the necessary operations like "determine the leftmost saturated cut X" and "find the rightmost net crossing X" in time $O(\log 2n)$. This is the most important point to get a runtime $O(n \log^2 n)$.

Simpler and faster algorithms are known for convex grids. Nishizeki/Saito/Suzuki [NSS85] define convex grids as the subclass of generalized switchboxes, where any two vertices can be connected by a path with at most 1 bend. They show that in convex grids only straight cuts have to be considered in this case. Based on this observation they achieve a runtime linear in the size of the routing region by a simple algorithm. Lai and Sprague [LS86] show the correct condition for the solvability of half-even instances in this class.

Kaufmann [K87] extends the notion of convex grids to generalized switchboxes where any horizontal or vertical line crosses the boundary at most twice. He shows that the instances in this class are extremely simple to solve. The algorithm works roughly as follows:

If there is a vertex v with $deg(v) = ter(v) = 2$, route both nets to the adjacent boundary vertices in the obvious way. If not, consider any certain corner v, and an adjacent boundary vertex w. If any net starts at w, then route it on the edge (v, w) and throw (v, w) away. If not, add a net $\{v, w\}$ and throw (v, w) away. Iterate until the routing region is empty.

Note that for this class straight cuts and cuts with 1 bend have to be considered, but only in the correctness proof of the algorithm. In the algorithm itself no cuts are considered. The runtime is $O(n)$.

The same algorithm works also for generalized switchboxes where any horizontal line crosses the boundary at most twice. This problem class is called half-convex grids. A weakness of the algorithm is that it can only be applied to even instances.

Open Problem. Find a condition for the solvability of half-even instances of PED in half-convex grids.

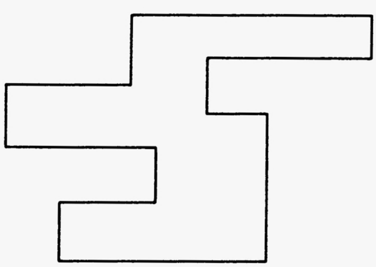
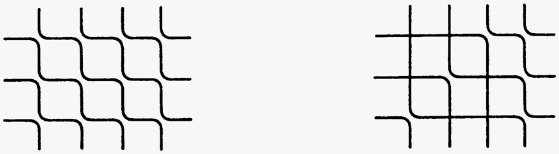


Fig. 9 A half-convex region

All algorithms mentioned so far remove one or two edges per step such that runtime $O(n)$ is optimal in a certain sense. Mehlhorn/Preparata [MP83] show that for rectangular routing regions better results are achievable. They present an algorithm for even and half-even instances with runtime $O(b \log b)$ where b is the perimeter of the rectangle. The main idea is to avoid to produce layouts as shown in Figure 10a (such layouts are typically produced by the algorithm of Section 2) and instead to route as shown in Figure 10b. Note that in the case of n nets the layout of Fig. 10a has $O(n^2)$ bends but the layout of Fig. 10b has only $O(n)$ bends.



(a) (b)

Fig. 10

Kaufmann/Klär [KK188] extend the result to generalized switchboxes without rectilinear visible corners and gets an algorithm for the solution of even and half-even instances of runtime $O(b \log^2 b)$.

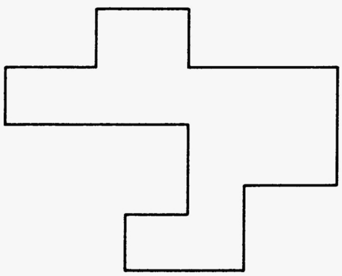


Fig. 11 A generalized switchbox without any rectilinear visible corners

Open Problem. Extend the last algorithms to larger problem classes like the homotopic routing problem.

6. Routing Regions with Exactly One Inner Hole

We consider routing regions with exactly two non-trivial faces one of which is the outer face. Also, all terminals of nets are supposed to lie on the boundary of these faces and no homotopies are given.

Okamura [O83] gives necessary and sufficient conditions for the solvability of such problems on planar graphs in the same style as Okamura/Seymour. In her paper the positions of the terminals are restricted in the following way: For each net both terminals have to lie on the same nontrivial face.

Matsumoto/Nishizeki/Saito give an algorithm with time complexity $O(bn^2 \log n)$ for this class of problems. Here n denotes as usual the size of the graph and b denotes the number of vertices on the boundary of the two non-trivial faces.

From now on we restrict ourselves to a very simple case: The routing region is a grid graph with two non-trivial faces (we call them the inner and the outer face) and the boundaries of both non-trivial faces are rectangles. Different types of problems are obtained by putting different restrictions on the positions of the terminals.

Under the restriction that both terminals of a net lie on the same face Suzuki/Ishiguro/Nishizeki [SIN87] give an algorithm with a running time linear in the size of the graph. It is based on the principles developed by Okamura and Matsumoto et al.

An alternative technique is the following: First determine a fixed homotopy for each net, such that no cut condition is violated. Then use the algorithm for homotopic routing [KM86] or a similar algorithm to find the final edge-disjoint paths.

This techniques can be used in the case where all terminals lie on the inner face. The first algorithm was given by LaPaugh [L82]. We describe a solution due to Suzuki et al. They propose to replace the routing region by a cycle with multiple edges. The length of the cycle is equal to the perimeter of the inner rectangle. The multiplicity of an edge e of the cycle is given by the capacity of the straight-line cut through the corresponding edge of the inner rectangle, cf. Figure 12.

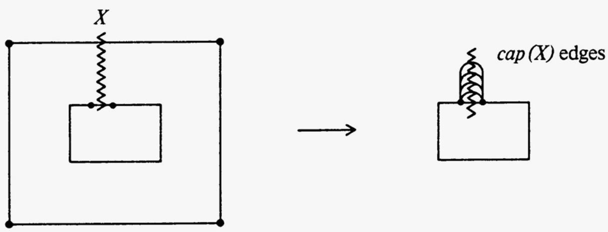


Fig. 12

The reduced problem can be solved by any of the standard algorithms for planar graphs [MNS85,BM86]. This determines the homotopies for the nets. The second step is then a standard homotopic routing problem. In [SNSFT88] a special data structure is used which supports the necessary operation efficiently such that a runtime of $O(k + n)$ for the first step and $O(\min(b_{out}, k \log k))$ for the second step can be achieved. k denotes the number of the nets and b_{out} the number of the vertices on the boundary of the outer face. The correctness proof for this method is based on the observation that only a limited set of cuts have to be considered, namely cuts which consist of two straight-line segments connecting the outer and the inner face. The capacity and density of such cuts is the same for the original and the reduced problem. Furthermore the solutions are 2- resp. 3-layer wireable, cf. Section 7 for a discussion of wireability.

If the terminals may lie anywhere on the two faces but not in the four corners of the outer rectangle (cf. Figure 13), the same technique is applicable, since also in this case only 2-segment cuts have to be considered. In [SIN87] an algorithm with $O(n)$ runtime for this problem class is presented.

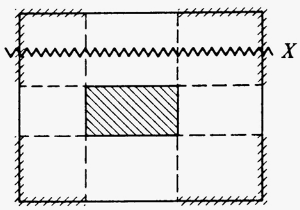


Fig. 13

Kaufmann/Klär ([KKl88]) extend their technique and get faster algorithms ($O(k \log k)$) for wider problem classes.

Open Problem. Is there an efficient algorithm for problems without any restrictions on the terminal positions?

7. Layer Assignment

A *layout* is a set of edge-disjoint paths. The problem of wiring a given layout in a grid is as follows: We have k copies of the routing region stacked vertically on top of each other. The k copies of each vertex are connected in the form of a line. We call the obtained graph the wiring graph and each copy of the routing region a layer. A layer assignment or wiring lifts a grid path p into a path P in the wiring graph such that p is the vertical projection of P. We require that the liftings of different paths p and q are *vertex-disjoint*. A layer assignment is a k-layer wiring if only k layers are used. Clearly, if the layout consists of vertex-disjoint paths, it can be wired within one layer. If the paths may cross each other but are not allowed to bend on a common vertex (knock knee), we call the layout Manhattan mode layout. Manhattan layouts can always be wired in two layers by assigning horizontal path segments to layer 1 and vertical segments to layer 2.

For arbitrary layouts in grids the situation is more complex. First of all, it is easy to find examples which are not 2- and 3-layer wirable.

Lipski [L84] showed that it is NP-complete to decide whether an arbitrary layout can be wired using three layers. Brady and Brown [BB84] showed that every layout can be wired using four layers. Both papers use the concept of two-colorable maps which we now briefly discuss. We show first that any layer assignment gives rise to a two-colorable map. We start by shifting the infinite grid by one-half unit in x- and y-direction. The squares of the shifted grid are called wiring tiles, cf. Figure 14. A wiring tile is either used by only one path and is then called trivial or by two paths. In the latter case the two paths either cross in the tile or bend in the tile (knock-knee). Trivial tiles can be removed because arbitrary layer changes can be performed in these tiles. This leaves us with the non-trivial tiles. In a non-trivial tile containing a crossing the horizontal wire segment either runs above the vertical segment (color I) or below the vertical wire segment (color II). In a non-trivial tile containing a knock-knee the tile is divided by a 45° or 135° degree line as shown in Figure 14. In one part of the tile the horizontal wire segment uses a higher-numbered layer (color I) and in the other part the vertical wire segment uses the higher-numbered layer (color II). In this way any layer assignment gives rise to a two-colorable map. The boundary between differently colored regions consists of diagonals which "cut" the knock-knees and horizontal and vertical tile boundaries, cf. Figure 14 c.

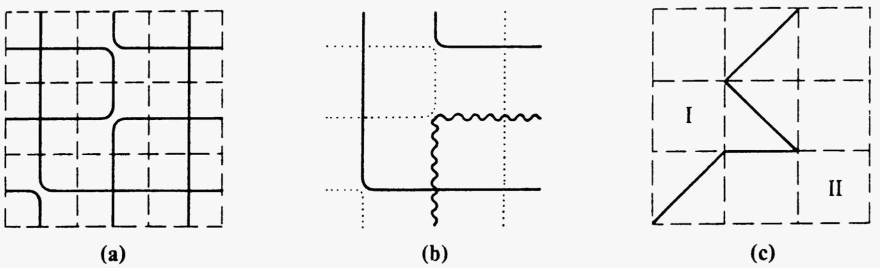


Fig. 14 (**a**) shows a layout consisting of 16 tiles, (**b**) shows a 3-layer wiring (layer 1 = ····, layer 2 = ∿∿, layer 3 = —) and (**c**) indicates the corresponding 2-colorable map

The problem of layer assignment can now be formulated as follows. Start with the diagonals induced by the knock-knees. Then add horizontal and vertical tile boundaries so as to turn the layout into a two-colorable map. In addition, avoid either the patterns shown in Figure 15 or the patterns shown in Figure 16. In the former case the layout is 3-layer wirable and in the latter case the layout is 4-layer wirable. Brady and Brown have shown that the pattern shown in Figure 16 can always be avoided and hence every layout is 4-layer wirable.

An alternative technique for 4-layer wiring which is not based on two-colorable maps, but on constraint graphs, was recently proposed by Tollis [T88]. His algorithm uses the 4th layer only if it is necessary and thus frequently uses only three layers.

Fig. 15

Fig. 16

A different approach is to produce the layout and the wiring simultaneously and not in separate phases. Preparata/Lipski [PL82] show in the first paper on 3-layer wirability that three layers are sufficient to solve a two-terminal channel routing problem with the minimum number of tracks. Gonzalez/Zheng [GZ88] extend this result to 3-terminal and multiterminal net channel routing.

Open Problems. Prove or disprove that every grid graph routing problem has a three-layer wiring.

Investigate techniques to modify and/or stretch the layout by a small factor to get a 2- resp. 3-layer wirable layout. Some results on the second problem can be found in [GZ88] and [BS87].

8. Multiterminal Nets

In the previous Sections we have considered nets with two terminals. We now allow nets with more than two terminals and call such nets multi-terminal nets. A layout is now a collection of edge-disjoint trees, one for each net. The tree for a net must connect the terminals of the net. The PED-problem with multi-terminal nets is difficult even for the simple case of channel routing. In channel routing the routing region is a rectangle and all terminals lie on the horizontal sides of this rectangle. The height of the rectangle is called the channel width and the goal is to minimize channel width. Sarrafzadeh [Sa87] shows that it is NP-complete to solve multiterminal net problems in knock-knee mode with optimal channel width. Mehlhorn et al. [MPS86] give simple approximation algorithms with unified approach for 2-, 3- and multiterminal net problems. For 2-terminal nets their method achieves optimal channel width, for 3-terminal nets it yields solutions being within a factor 3/2 of the optimum and for multi-terminal nets it achieves a factor of 2. Gao and Kaufmann [GK87] showed recently that the factor $3/2 + o(1)$ can always be achieved.

All these results are based on the same principle: The idea is to split the multiterminal nets into simpler parts such as 2-terminal nets or nets with all terminals on the same side of the channel. These simple parts are then routed independently as indicated in Figure 17.

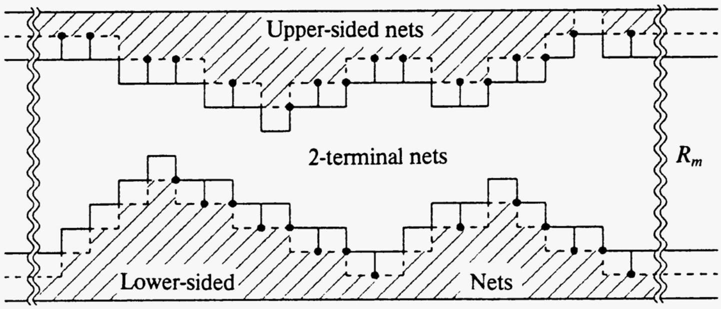


Fig. 17

The one-sided nets are routed close to the shores of the channel and the two-terminals nets are routed in the "middle" of the channel. The crucial step is to find a decomposition which does not increase channel density too much.

Consider a net N with terminals $v, \ldots, v_{l-1}$. Assume that the terminals appear in that order in a clockwise traversal of the boundary of the rectangle. If one replaces each net N by the l two-terminal nets $\{v; v_{(i+1) mod\ l},\ 0 \leq i < l\}$, then the density of every horizontal or vertical cut is at most doubled. Thus, if the original instance satisfies the cut condition and if one inserts a new grid column and row between every pair of columns and rows, then an instance satisfying the cut criterion is obtained. The obtained instance can be solved by the algorithm of [MP86].

Open Problems. Develop better approximation algorithms for multi-terminal net problems in rectangles.

Give a nontrivial lower bound for the multiterminal net problem in a channel.

References

[BM86] Becker, M., Mehlhorn, K. (1986): Algorithms for routing in planar graphs. Acta Inf. **23**, 163–176

[BB84] Brady, M., Brown, D. (1984): VLSI routing: Four layers suffice. M.I.T. VLSI Conference 1984

[BS87] Brady, M., Sarrafzadeh, M. (1987): Layout stretching to ensure wirability. Technical Report

[F82] Frank, A. (1982): Disjoint paths in a rectilinear grid. Combinatorica **2**, 361–371

[GK87] Gao, S., Kaufmann, M. (1987): Channel routing of multiterminal nets. Proc. 28th IEEE FOCS, pp. 316–325

[GZ88] Gonzalez, T., Zheng, S.-Q. (1988): Simple three-layer channel routing algorithms. In: Reif, J.H. (ed.): VLSI algorithms and architectures. Springer Verlag, Berlin, pp. 237–246 (Lect. Notes. Comput. Sci., Vol. 319)

[K87a] Kaufmann, M.: Über lokales Verdrahten von 2-Punkt-Netzen. Dissertation

[K87b] Kaufmann, M. (1987): A linear-time algorithm for routing in a convex grid. Technical Report

[KKl88] Kaufmann, M., Klär, G.: Routing around a rectangle – towards the general case. Preprint

[KKl88] Kaufmann, M., Klär, G.: Routing in generalized switchboxes without rectilinear visible corners. Technical Report

[KMa88] Kaufmann, M., Maley, F.M. (1988): Parity conditions in homotopic knock-knee routing. Preprint

[KM85] Kaufmann, M., Mehlhorn, K. (1985): Generalized switchbox routing. J. Algorithms **7**, 510–531 (and ICALP 1985)

[KM86] Kaufmann, M., Mehlhorn, K. (1986): On local routing of two-terminal nets. Technical Report 03/1986, FB 10, Universität des Saarlandes; an extended abstract appeared in STACS 87, LNCS 247, pp. 40-52

[KM88] Kaufmann, M., Mehlhorn, K. (1988): A linear-time algorithm for the local routing problem. Technical Report, FB 10, Universität des Saarlandes

[KvL82] Kramer, M.E., van Leeuwen, J. (1982): Wire routing is NP-complete. Technical Report RUU-CS-82-4, Department of Computer Science, Rijksuniversiteit te Utrecht

[LS87] Lai, T.-H., Sprague, A. (1987): On the routability of a convex grid. J. Algorithms **8**, 372–384

[LaP80] LaPaugh, A.S. (1980): A polynomial-time algorithm for optimal routing around a rectangle. Proc. 21st IEEE FOCS, pp. 282–293

[L84] Lipski jr., W. (1984): On the structure of three-layer wirable layouts. In: Preparata, F.P. (ed.): VLSI theory. JAI Press, London, pp. 231–243 (Adv. Comput. Res., Vol. 2)

[MNS86] Matsumoto, K., Nishizeki, T., Saito, N. (1986): Planar multicommodity flows, maximum matchings and negative cycles. SIAM J. Comput. **15**, 495–510

[MP86] Mehlhorn, K., Preparata, F.P. (1986): Routing through a rectangle. J. Assoc. Comput. Mach. **33**, 60–85

[MPS86] Mehlhorn, K., Preparata, F.P., Sarrafzadeh, M. (1986): Channel routing in knock-knee mode: Simplified algorithms and proofs. Algorithmica **1**, 213–221

[NSS85] Nishizeki, T., Saito, N., Suzuki, K. (1985): A linear-time routing algorithm for convex grids. IEEE Trans. Comput.-Aided Design **4**, 68–76

[O83] Okamura, H. (1983): Multicommodity flows in graphs. Discrete Appl. Math. **6**, 55–62

[OS81] Okamura, H., Seymour, P.D. (1981): Multicommodity flows in planar graphs. J. Comb. Theory, Ser B **31**, 75–81

[PL82] Preparata, F.P., Lipski jr., W. (1982): Three layers are enough. Proc. 23rd IEEE FOCS, pp. 350–357

[Sa87] Sarrafzadeh, M.: On the complexity of the general channel routing problem in the knock-knee mode. IEEE Trans. Comput.-Aided Design *6*

[S86] Schrijver, A.: Decompositions of graphs on surfaces and a homotopic circulation theorem. J. Comb. Theory, Ser. B, to appear

[S87] Schrijver, A. (1987): Edge-disjoint homotopic paths in straight-line planar graphs. Report OS-R8718, Mathematical Centre, Amsterdam (SIAM J. Discrete Math., to appear)

[SIN87] Suzuki, H., Ishiguro, A., Nishizeki, T. (1987): Edge-disjoint paths in a grid bounded by two nested rectangles. Technical Report, Tohoku University

[SIN88] Suzuki, H., Ishiguro, A., Nishizeki, T. (1988): Variable-priority queue and doughnut routing. Technical Report, Tohoku University

[SNSFT88] Suzuki, H., Nishizeki, T., Saito, N., Frank, A., Tardos, É. (1988): Algorithms for routing around a rectangle. Technical Report, Tohoku University

[T88] Tollis, I.G. (1988): A new algorithm for wiring layouts. In: Reif, J.H. (ed.): VLSI algorithms and architectures. Springer Verlag, Berlin, pp. 257–267 (Lect. Notes Comput. Sci., Vol. 319)