The Steiner Tree Packing Problem in VLSI-Design

M. Grötschel A. Martin R. Weismantel

Abstract

In this paper we describe several versions of the routing problem arising in VLSI design and indicate how the Steiner tree packing problem can be used to model these problems mathematically. We focus on switchbox routing problems and provide integer programming formulations for routing in the knock-knee and in the Manhattan model. We give a brief sketch of cutting plane algorithms that we developed and implemented for these two models. We report on computational experiments using standard test instances. Our codes are able to determine optimum solutions in most cases, and in particular, we can show that some of the instances have no feasible solution if Manhattan routing is used instead of knock-knee routing.

1 Introduction

The design of electronic circuits is a hierarchical process consisting of several phases. The beginning is a description of the task the circuit to be designed must perform. Such a task can be viewed as a complex logical function that consists of many elementary logic operations. Usually several of these elementary logic operations are combined into a logical unit (for example an adder). In the *logical design phase* chip designers specify which of these predefined logical units are to be used, and determine which of the chosen logical units must be connected by wires so that the chip performs in the way it should.

The logical units are also called *cells*. Each cell is characterized by its width, its height, its contact points (so-called *terminals*) and its electric properties. A *net* is a set of terminals that must be connected by a wire (as specified in the logical design phase). The list of cells and the list of nets are the input of the second phase, the *physical design*. Here, the task is to assign the cells to a certain rectangular area and connect (route) the nets by wires. The physical design problem is, of course, more complicated than the sketch above suggests, since certain design rules have to be taken into account, an objective function is to be minimized, etc. The design rules strongly depend on the given layout style and specify, for instance, the distance two nets must stay apart, whether certain cells are preassigned to certain locations and so on. This applies especially to the objective function. Usually, the primary goal is to minimize the whole area of the chip or, if the chip area is fixed in advance, to guarantee routability, i. e., to solve the problem of placing the cells on the chip such that there exists a feasible solution to the routing problem.

However, routability can hardly be measured and expressed in terms of an objective function. Thus, minimizing the total length of all routes is very often used instead. Another reason for minimizing the routing length is that an electronic circuit with small routing length usually needs little area on the whole. Thus, minimizing the overall area is (somehow) implicitly taken into account by minimizing the routing length.

Any reasonably precise version of the physical design problem is $\mathcal{NP}$-hard, even very simple models are. Moreover, most real world problem instances involve several thousands of cells and nets, so that today's algorithmic knowledge makes it very improbable that they can be solved to optimality. Therefore, the physical design problem is (heuristically) decomposed into subproblems. The first subproblem typically consists of finding appropriate locations for the cells (*placement problem*). Subsequently, the nets must be realized by wiring the appropriate terminals (*routing problem*) and finally, a compaction step is performed if required. This process is iterated with different parameters if the final result is not satisfactory.

In this paper we will focus on the routing problem in more detail. We survey in Section 2 different types of routing models used in practice and relate them to the packing of Steiner trees in certain graphs. In Section 3 we state an integer programming formulation of the Steiner tree packing problem and describe several classes of valid and facet-defining inequalities for the associated Steiner tree packing polyhedron. Specializing this model to switchbox routing we distinguish between routing in knock-knee and Manhattan style by using an additional class of inequalities (the *Manhattan inequalities*) to meet the requirements of the latter routing style.
In Section 4 we report on our computational experiments with a cutting plane algorithm that we designed and implemented for switchbox routing in Manhattan style; and we compare these in Section 5 with our results for the same instances when knock-knees are allowed.

2 The Routing Problem in VLSI Design: A Short Survey

We assume in this section that the placement problem has been solved. We seek for a solution of the routing problem. In technical terms, we are given a list of nets. Each net consists of a set of terminals. The terminals specify the points at which wires have to contact the cells. The routing problem is to connect the nets by wires on the routing area subject to certain technical side constraints. As mentioned above, the objective usually is to minimize the overall wiring length.

We say a net is *routed* if its terminals are connected by (electric) wires. We speak of a *k-terminal net*, if k is the number of terminals of the net. If $k > 2$, the term *multiterminal net* is often used. In the following we will not distinguish between a net and the route of a net unless this may lead to confusion.

The routing itself takes place on so-called *layers*. If some net changes a layer, a hole, called *via*, must be "drilled". Usually, each layer is subdivided into horizontal and vertical lines, so-called *tracks* to which the wires of the nets must be assigned. If there does not exist such a division into tracks we speak of a *free* or *grid-free routing*. Further side constraints include, for instance, the distance two wires must stay apart from each other, how long two different nets may run on top of each other on two different layers, or that some wires must not exceed a certain length.

In practice, the routing problem itself is also decomposed because of its inherent complexity and large scale. In the *global routing phase* the homotopy of the nets is determined, i. e., it is determined how the wires "maneuver around the cells". Thereafter, in the *detailed routing phase* the wires are assigned to the layers and tracks according to the homotopy specified in the global routing step.

The routing problems arising in both phases are usually expressed in graph-theoretic terminology. To describe these models precisely, we introduce some graphtheoretic notation.

We denote graphs by $G = (V, E)$, where V is the node set and E the edge set. All graphs we consider are undirected and finite. For a given edge set $F \subseteq E$, we denote by $V(F)$ all nodes that are incident to an edge in F. We call a sequence of nodes and edges $K = (v_0, e_1, v_1, e_2, \ldots, v_{l-1}, e_l, v_l)$, where each edge e_i is incident with the nodes v_{i-1} and v_i for $i = 1, \ldots, l$, and where the edges are pairwise different and the nodes distinct (except possibly v_0 and v_l), a *path from v_0 to v_l*, if $v_0 \neq v_l$, and a *cycle*, if $v_0 = v_l$ and $l \geq 2$. We call a graph G a *complete rectangular $h \times b$ grid graph*, if it can be embedded in the plane by h horizontal lines and b vertical lines such that the nodes of V are represented by the intersections of the lines and the edges are represented by the connections of the intersections. A *grid graph* is a graph that is obtained from a complete rectangular grid graph by deleting some edges and removing isolated nodes (i. e., nodes that are not incident to any edge).

Let $G = (V, E)$ be a graph and $T \subseteq V$ a node set of G. An edge set S is called a *Steiner tree for T in G*, if the subgraph $(V(S), S)$ contains a path from s to t for all pairs of nodes $s, t \in T, s \neq t$. Following the notation in VLSI-design, we call T a *terminal set* or a *net* and each $t \in T$ a *terminal*. "Routing some net T in a graph G" means in graphtheoretic terms, "finding a Steiner tree for T in G". We will use both phrases in the following.

Note that our definition of a Steiner tree differs from the standard terminology used in the literature. A Steiner tree is usually supposed to be a tree. For our purposes, however, the above definition is more convenient for our polyhedral investigations. A Steiner tree that is a tree and whose leaves are terminals is called *edge-minimal*. Observe that, since objective functions in practice are positive, every shortest Steiner tree is edge-minimal.

There are many ways to model the global routing problem as a graphtheoretic problem. Usually, the routing area is subdivided into subareas. This is done in a way such that the resulting subareas have certain special properties, for instance, they contain no holes (i. e., there are no cells located within the areas) or they have simple shapes (for example, rectangles). These subareas are represented by the nodes or the edges of some graph. We describe the node representation. Here, two nodes are connected by an edge, if the corresponding subareas are adjacent. Additionally, a capacity is assigned to an edge limiting the number of nets that may run between the subareas associated with the two endnodes of this edge. The weight of an edge corresponds to the distance between the two midpoints of the according subareas. Every terminal of a net is assigned to that node, whose corresponding subarea contains the terminal or is closest to the position of the terminal. The global routing problem consists in routing all nets in the graph contructed this way (or in graphtheoretic terms, finding a Steiner tree for each terminal set) such that the capacity constraints are satisfied and the total wiring length (that is the sum of the weights of the Steiner trees) is as small as possible.

After having solved the global routing problem every subarea that corresponds to a node in the global routing graph must be routed in detail. The number of different detailed routing models which are studied in the literature or which are used in practice is tremendous. Usually, the problems coming up are formulated in a grid graph. We restrict ourselves to this case, too. The detailed routing problems can be classified according to two criteria (see 1. and 2. below) which are independent of each other. We introduce these classifications now and discuss o few important subcases. For a more complete and detailed treatment we refer to [L90].

1. The detailed routing problems are distinguished according to the shape of the routing area and the locations of the terminals. As mentioned before,

the nodes in the global routing graph represent subareas of the whole routing area. Depending on the subdivision, different shapes of detailed routing areas arise. At the end of the global routing phase it is known which nets go across which subareas. Suppose, some net crosses the border of two adjacent subareas. Of course, from the information of the global routing solution it is not clear at which point the net meets the border. Each such crossing point is interpreted as a "pseudo"-terminal. In order to solve the routing problems for each of these subareas independently, locations for the pseudo-terminals must be determined. This usually is done by applying heuristics. Concerning the shape of the routing area and the locations of the terminals the following detailed routing models are of particular interest in practice.

(a) (Channel routing) Here, we are given a complete rectangular grid graph. The terminals of the nets are exclusively located on the lower and upper border (see Figure 1). It is possible to vary the height (= number of horizontal tracks) of the channel. Hence, the size of the routing area is not fixed in advance.

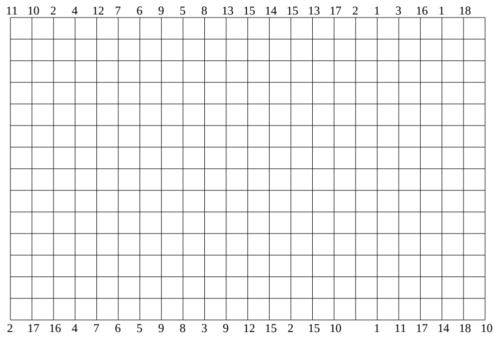


Figure 1:

(b) (Switchbox routing) Again, we are given a complete rectangular grid graph. The terminals may be located on all four sides of the grid graph (see Figure 2). Thus, the size of the routing area is fixed.

(c) (General routing) In this case, an arbitrary grid graph is considered. The terminals are located at any hole of the grid (see Figure 3). Here, the homotopy of the nets must be taken into account (which is trivial in (a) and (b)).

2. The detailed routing problems are distinguished by the extent to which the layers are taken into account when the wires of the nets are assigned to the tracks.

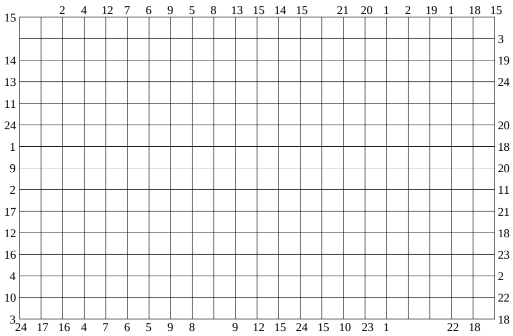


Figure 2:

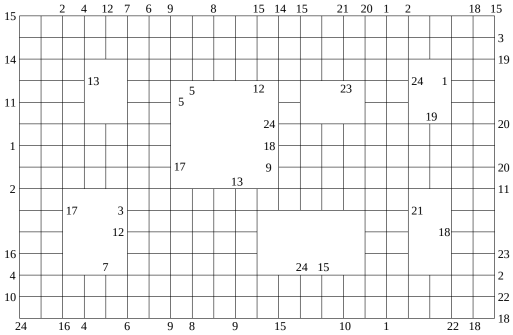


Figure 3:

(a) (Multiple layer model) Given a k-dimensional grid graph (that is a graph obtained by stacking k copies of a grid graph on top of each other and connecting corresponding nodes by perpendicular lines), where k denotes the number of layers. The nets have to be routed in a node disjoint fashion. The multiple layer model is well suited to reflect reality. The disadvantage is that, in general, the resulting graphs are very large.

(b) (Manhattan model) Given some (planar) grid graph. The nets must be routed in an edge disjoint fashion with the additional restriction that nets that meet at some node are not allowed to bend at this node, i. e., so-called *knock-knees* (cf. Figure 4) are not allowed. This restriction guarantees that the resulting routing can be laid out on two layers at the possible expense of causing long detours.

(c) (Knock-knee model) Again, some (planar) grid graph is given and the task is to find an edge disjoint routing of the nets. In this model knock-knees are possible. Very frequently, the wiring length of a solution in this case is smaller than in the Manhattan model. The main drawback is that the assignment to layers is neglected. Brady and Brown [BB84]

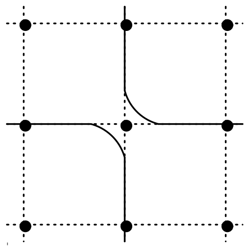


Figure 4:

have designed an algorithm that guarantees that any solution in this model can be routed on four layers. It was shown in [Li84] that it is $\mathcal{NP}$-complete to decide whether a realization on three layers is possible.

The models coming out of these two kinds of classifications can be combined in all possible ways. For example, combining 1 (b) and 2 (c) we obtain a switchbox routing problem in the knock-knee model, or in graphtheoretic terms, the problem of finding edge disjoint Steiner trees in a complete rectangular grid graph, where all terminals are located on the outer face. Moreover, depending on the model, different objective functions are considered. Possible objective functions are, for example, minimizing the routing area or minimizing the routing length. Minimizing the routing area is typically the objective in channel routing problems, whereas the routing length is usually minimized, if the routing area is fixed in advance.

It is not surprising that most of these routing problems are $\mathcal{NP}$-hard. For example, the problem of finding a (with respect to some weighting of the edges) minimum Steiner tree in a graph G for some terminal set T is $\mathcal{NP}$-hard (see [K72], [GJ77]). Even the problem of deciding whether there exists a feasible solution for the switchbox routing problem in the knock-knee model ([S87]) or in the Manhattan model ([Sz85]), respectively, is $\mathcal{NP}$-complete. In the next section we present a model that is applicable to the global routing problem and the switchbox routing problem in the knock-knee model and Manhattan model, respectively, and attack it from a polyhedral point of view.

3 A Polyhedral Approach to The Knock Knee and Manhattan Routing Model

To get started let us formally introduce the Steiner tree packing problem.

Problem 3.1 (The weighted Steiner tree packing problem)
Instance:

A graph $G = (V, E)$ with positive, integer capacities $c_e \in \mathbb{N}$ and nonnegative weights $w_e \in \mathbb{R}_+$, $e \in E$.
A net list $\mathcal{N} = \{T_1, \ldots, T_N\}$, $N \geq 1$, with $T_k \subseteq V$ for all $k = 1, \ldots, N$.

Problem:

Find edge sets $S_1, \ldots, S_N \subseteq E$ such that

(i) S_k is a Steiner tree in G for T_k for all $k = 1, \ldots, N$,

(ii) $\sum_{k=1}^{N} |S_k \cap \{e\}| \leq c_e$ for all $e \in E$,

(iii) $\sum_{k=1}^{N} \sum_{e \in S_k} w_e$ is minimal.

If requirement (iii) in Problem 3.1 is omitted we call the corresponding problem the *Steiner tree packing problem* without the prefix "weighted". We call an N-tupel $(S_1, \ldots, S_N)$ of edge sets a *Steiner tree packing* or *packing of Steiner trees* if the sets $S_1, \ldots, S_N$ statisfy (i) and (ii) of Problem 3.1. We will refer to an instance of the weighted Steiner tree packing problem by $(G, \mathcal{N}, c, w)$ and to an instance of the Steiner tree packing problem by $(G, \mathcal{N}, c)$.

We assume throughout the paper that every terminal set of the net list $\mathcal{N}$ has at least cardinality two and that $N \geq 1$.

Many routing problems introduced in the previous section can be formulated as Steiner tree packing problems in certain graphs with, possibly, some additional constraints reflecting the design rules. In this section we focus in detail on two such cases

- the switchbox routing problem in the knock-knee style and
- the switchbox routing problem in the Manhattan style.

We start with modelling the switchbox routing problem in the knock-knee style as an integer program. Before doing so let us fix some further notation.

We are given a graph $G = (V, E)$ with capacities $c_e \in \mathbb{N}$ for all $e \in E$ and a net list $\mathcal{N} = \{T_1, \ldots, T_N\}$, $N \geq 1$. Let $\mathbb{R}^{\mathcal{N} \times E}$ denote the $N \cdot |E|$ – dimensional vector space $\mathbb{R}^E \times \ldots \times \mathbb{R}^E$, where the components of each vector $x \in \mathbb{R}^{\mathcal{N} \times E}$ are indexed by x_e^k for $k \in \{1, \ldots, N\}$, $e \in E$. Moreover, for a vector $x \in \mathbb{R}^{\mathcal{N} \times E}$ and $k \in \{1, \ldots, N\}$, we denote by $x^k \in \mathbb{R}^E$ the vector $(x_e^k)_{e \in E}$, and, for notational simplicity, we write $x = (x^1, \ldots, x^N)$ instead of $x = ((x^1)^T, \ldots, (x^N)^T)^T$. For an edge set $F \subseteq E$, χ^F denotes the incidence vector of F. The *incidence vector of a Steiner tree packing* $(S_1, \ldots, S_N)$ is denoted by $(\chi^{S_1}, \ldots, \chi^{S_N})$.

With every $e \in E$ and $k \in \{1, \ldots, N\}$ we associate a Boolean variable x_e^k with the interpretation $x_e^k = 1$ if edge e is used to connect terminal set T_k and $x_e^k = 0$ otherwise. Then it is easy to see that each incidence vector of a Steiner tree packing satisfies the constraints (3.2) (i) – (iv), and vice versa, each vector $x \in$

$\mathbb{R}^{\mathcal{N}\times E}$ satisfying (3.2) (i) – (iv) is the incidence vector of a Steiner tree packing. Hence, (3.2) is an integer programming formulation for the weighted Steiner tree packing problem.

$$
(3.2) \qquad
\begin{array}{lll}
\min & \displaystyle\sum_{k=1}^{N}\sum_{e\in E} w_e x_e^k & \\
\text{(i)} & \displaystyle\sum_{e\in\delta(W)} x_e^k \geq 1, & \text{for all } W \subset V,\ W\cap T_k \neq \emptyset,\ (V\setminus W)\cap T_k \neq \emptyset, \\
& & k=1,\ldots,N. \\
\text{(ii)} & \displaystyle\sum_{k=1}^{N} x_e^k \leq c_e, & \text{for all } e\in E. \\
\text{(iii)} & 0 \leq x_e^k \leq 1, & \text{for all } e\in E,\ k=1,\ldots,N. \\
\text{(iv)} & x_e^k \in \{0,1\}, & \text{for all } e\in E,\ k=1,\ldots,N.
\end{array}
$$

The inequalities (3.2) (i) are called *Steiner cut inequalities*, inequalities (3.2) (ii) are called *capacity inequalities* and the ones in (3.2) (iii) *trivial inequalities*.

We define the *Steiner tree packing polyhedron* STP $(G,\mathcal{N},c)$ as the convex hull of all incidence vectors of Steiner tree packings, i.e.,

$$\text{STP}\,(G,\mathcal{N},c) := \text{conv}\,\{x\in\mathbb{R}^{\mathcal{N}\times E} \mid x \text{ satisfies (3.2) (i) – (iv)}\}.$$

If G is a complete rectangular grid graph, then every edge-minimal solution of (3.2) is obviously a switchbox routing in the knock knee style, and vice versa.

To model the Manhattan routing style, where knock-knees are not allowed, we have to introduce additional inequalities that make it impossible for two Steiner trees to bend at the same node.

Let G be a grid graph and uv, vw be two consecutive horizontal (or vertical) edges. Let N_1, N_2 be a partition of $\{1,\ldots,N\}$. Then, the constraint

$$(3.3) \qquad \sum_{k\in N_1} x_{uv}^k + \sum_{k\in N_2} x_{vw}^k \leq 1$$

is called *Manhattan inequality.*

Again it is easy to see that if G is a complete rectangular grid graph, then every edge-minimal packing of Steiner trees that satisfies, for every pair of consecutive edges and for every 2-partition of the set of nets, the corresponding Manhattan inequality (3.3) and the constraints (3.2) (i) – (iv) corresponds to a feasible switchbox routing in the Manhattan style. Conversely, the incidence vector of a switchbox routing in Manhattan style satisfies the inequalities (3.2) (i) – (iv) and all Manhattan inequalities.

We define the Steiner tree packing polyhedron in Manhattan style $\mathrm{STP}_M(G, \mathcal{N}, c)$ as

$$\mathrm{STP}_M(G, \mathcal{N}, c) := \mathrm{conv}\,\{x \in \mathrm{STP}\,(G, \mathcal{N}, c) \mid x \text{ satisfies all inequalities (3.3)}\}.$$

In the remainder of this section we present some inequalities that are valid for $\mathrm{STP}\,(G, \mathcal{N}, c)$. Since $\mathrm{STP}_M(G, \mathcal{N}, c) \subseteq \mathrm{STP}\,(G, \mathcal{N}, c)$, every inequality that is valid for $\mathrm{STP}\,(G, \mathcal{N}, c)$ is valid for $\mathrm{STP}_M(G, \mathcal{N}, c)$ as well. For a detailed discussion under which conditions some of these inequalities define facets of $\mathrm{STP}\,(G, \mathcal{N}, c)$, we refer to [GMW92a].

The Steiner Partition Inequalities

Let a graph $G = (V, E)$ and a set of terminals $T \subseteq V$, $|T| \geq 2$ be given. A partition $V_1, \ldots, V_p$, $p \geq 2$, of V is called a *Steiner partition (with respect to T)* if $V_i \cap T \neq \emptyset$ for $i = 1, \ldots, p$. The inequality

$$x(\delta(V_1, \ldots, V_p)) \geq p - 1$$

induced by a Steiner partition $V_1, \ldots, V_p$ is called a *Steiner partition inequality.* (Note that a Steiner cut inequality is the special case, where $p = 2$.) Obviously, each Steiner partition inequality is valid for $\mathrm{STP}\,(G, \{T\}, \mathbb{1})$ (cf. [GM90]).

The Alternating Cycle Inequalities

Let $G = (V, E)$ be a graph and $\mathcal{N} = \{T_1, T_2\}$ a net list. We call a cycle F in G an *alternating cycle with respect to* T_1, T_2, if $F \subseteq [T_1 : T_2]$ and $V(F) \cap T_1 \cap T_2 = \emptyset$ (see Figure 5). Moreover, let $F_1 \subseteq E(T_2)$ and $F_2 \subseteq E(T_1)$ be two sets of diagonals of the alternating cycle F with respect to T_1, T_2. The inequality

$$(\chi^{E \setminus (F \cup F_1)}, \chi^{E \setminus (F \cup F_2)})^T x \geq \tfrac{1}{2}|F| - 1$$

is called an *alternating cycle inequality.*

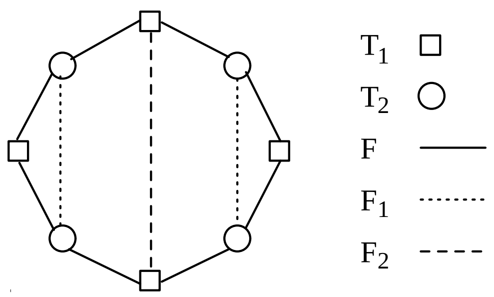


Figure 5:

It is not difficult to see that the basic form of an alternating cycle inequality, i. e., $F_1 = F_2 = \emptyset$, is valid for $\mathrm{STP}\,(G, \mathcal{N}, \mathbb{1})$, but in general, it is not facet-defining.

The sets F_1 and F_2 are used to strengthen the basic form; in fact, choosing them appropriately we can obtain valid and even facet-defining inequalities (see [GMW92a] for details).

The next type of inequalities to be considered here are the so-called grid inequalities.

The Grid Inequalities

Let $G = (V, E)$ be a graph and $\mathcal{N} = \{T_1, T_2\}$ a net list. Furthermore, let $\hat{G} = (\hat{V}, \hat{E})$ be a subgraph of G such that $\hat{G}$ is a complete rectangular $h \times 2$ grid graph with $h \geq 3$. Assume that the nodes of V are numbered such that $\hat{V} = \{(i,j) \mid i = 1, \ldots, h,\ j = 1, 2\}$. Moreover, let $(1,1), (h,2) \in T_1$ and $(1,2), (h,1) \in T_2$. We call the inequality

$$(\chi^{E \setminus \hat{E}}, \chi^{E \setminus \hat{E}})^T x \geq 1$$

an $h \times 2$ *grid inequality* (see Figure 6). In [GMW92a] we derived (very technical) conditions for an $h \times 2$ grid inequality to define a facet. The following theorem characterizes the conditions under which an $h \times 2$ grid inequality is valid.

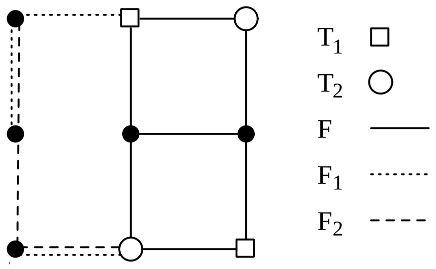


Figure 6:

Theorem 3.4 *Let $\hat{G} = (\hat{V}, \hat{E})$ be a complete rectangular $h \times 2$ grid graph with $h \geq 3$. Let $\mathcal{N} = \{T_1, T_2\}$ be a net list where $T_1 = \{(1,1), (h,2)\}$ and $T_2 = \{(1,2), (h,1)\}$. Furthermore, let $G = (V, E)$ be a graph with $\hat{V} \subseteq V$, $\hat{E} \subseteq E$ such that the following set of horizontal edges $\{uv \in \hat{E} \mid$ there exists an $i \in \{1, \ldots, h\}$ with $u = (i,1)$ and $v = (i,2)\}$ is a cut in G. Set $F := \hat{E}$ and let $F_1, F_2 \subset E \setminus F$, then the inequality*

$$(\chi^{E \setminus (F \cup F_1)}, \chi^{E \setminus (F \cup F_2)})^T x \geq 1$$

is valid for $STP(G, \mathcal{N}, \mathbb{1})$ if and only if F_1 and F_2 satisfy the following properties:

(i) For all $u, v \in V(F)$, $u \neq v$ there does not exist a path from u to v in (V, F_k) for $k = 1, 2$.

(ii) F_1 and F_2 are maximal with respect to property (i).

The Critical Cut Inequalities

Finally, we recall the so-called critical cut inequalities introduced in [GMW92a]. Let $G = (V, E)$ be a graph with edge capacities $c_e \in \mathbb{N}$, $e \in E$. Moreover, let $\mathcal{N} = \{T_1, \ldots, T_N\}$ be a net list. For a node set $W \subseteq V$, we define $S(W) := \{k \in \{1, \ldots, N\} \mid T_k \cap W \neq \emptyset,\ T_k \cap (V \setminus W) \neq \emptyset\}$. We call a cut induced by a node set W *critical for* $(G, \mathcal{N}, c)$ if $s(W) := c(\delta(W)) - |S(W)| \leq 1$, i. e., if the sum of the capacities of the edges leaving W exceeds the number of nets that must use at least one edge leaving W by at most 1.

Suppose that V_1, V_2, V_3 is a partition of V such that $\delta(V_1)$ is a critical cut. Moreover, assume that, for some $j \in \{1, \ldots, N\}$, $T_j \cap V_1 = \emptyset$ and $T_j \cap V_i \neq \emptyset$ for $i = 2, 3$. Then, the inequality

$$x^j([V_2 : V_3]) \geq 1$$

is called a *critical cut inequality with respect to T_j*.

It is easy to see that the critical cut inequality with resepct to T_j is valid for STP $(G, \mathcal{N}, c)$.

4 Computational Results for the Manhattan Model

In this section we present the computational results we obtained with our cutting plane algorithm for the switchbox routing problem in Manhattan mode. The Steiner partition inequalities, the alternating cycle inequalities, the grid and the critical cut inequalities together with the Manhattan inequalities form the basis of our cutting plane algorithm.

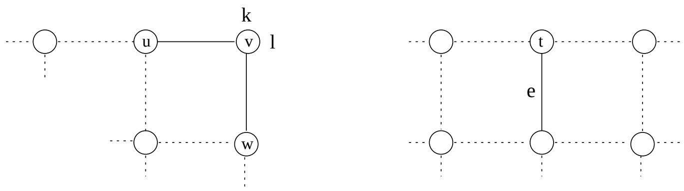
Our code is an extension and modification of the cutting plane algorithm for switchbox routing in knock-knee style that we described in [GMW92b]. We could use all separation routines for the Steiner partition inequalities, the alternating cycle inequalities, the grid and the critical cut inequalities, all special features (preprocessing, ...) and implementational tricks (perturbation, ...) developed for the routing problem in the knock-knee model. In addition, we designed and implemented a separation routine for the Manhattan inequalities (3.3), and some (minor) changes were needed or useful to apply the code to Manhattan routing problems.

Our procedure for separating Manhattan inequalities works as follows. Let us assume that that the capacity inequalities are satisfied (of course, this can be checked in linear time). Let $uv \in E$ and $vw \in E$ be two horizontal edges that are incident to node $v \in V$ (the same arguments apply to the case of two consecutive vertical edges). For every net $k \in \{1, \ldots, N\}$, we determine $\max\{x^k_{uv}, x^k_{vw}\}$. Set $N_1 := \{k \in \{1, \ldots, N\} \mid x^k_{uv} > x^k_{vw}\}$ and $N_2 := \{k \in \{1, \ldots, N\} \mid x^k_{uv} \leq x^k_{vw}\}$. If

$N_1 = \emptyset$ or $N_2 = \emptyset$, we can conclude that no violated Manhatten inequality exists, since the capacity inequalities are all satisfied. Otherwise, N_1, N_2 is a partition of $\{1, \ldots, N\}$ and the inequality $\sum_{k \in N_1} x^k_{uv} + \sum_{k \in N_2} x^k_{vw} \leq 1$ is a Manhattan inequality with maximal left hand side. This procedure obviously solves the separation problem for the class of Manhattan inequalities.

We also modified the LP-based primal heuristic described in [GMW92b] to guarantee that only Steiner tree packings are feasible that contain no knock-knees. We omit the technical details here.

Moreover, we exploit the fact that nets must not bend against each other in order to fix variables at the initial phase of the code. If two terminals of different nets k and l are located at the same corner v of the grid graph, i.e., $v \in T_k,\ v \in T_l$, then the input data specifies which of the two edges that are incident to v is used by which of the two nets. For example, in Figure 7 (a) edge vu must be used by net k and edge vw must be used by net l. Since the capacities of the edges are equal to one, all variables x^i_{vu} $(i \in \{1, \ldots, N\} \setminus \{l\})$ and x^i_{vw} $(i \in \{1, \ldots, N\} \setminus \{k\})$ can be fixed to zero.



(a) (b)

Figure 7:

Furthermore, suppose a terminal t of net k is not located at any corner of the grid graph. Then the edge e that is incident to t but not included in the outer face cycle cannot be used by any net, except k. Hence, the variables x^i_e, $i \in \{1, \ldots, N\} \setminus \{k\}$, can be fixed to zero. This situation is illustrated in Figure 7 (b).
Many variables can be fixed by using critical cuts and logical implications derived from them. How these can be found is described in [GMW92b].

The problem instances to which we applied our code are taken from VLSI literature. Table 1 summarizes the data. Column 1 presents the name used in the literature. In column 2 and 3 the height and width of the underlying grid graph is given. Column 4 contains the number of nets. Columns 5 to 9 provide information about the distribution of the nets; more precisely, column 5 gives the number of 2-terminal nets, column 6 gives the number of 3-terminal nets and so on. Finally, the last column states the reference to the paper the example is taken from.

The standard input format for switchbox routing problems used in the literature

Example	h	b	N	Distribution of the Nets					Ref.
				2	3	4	5	6	
difficult switchbox	15	23	24	15	3	4	1	1	[BP83]
more difficult switchbox	15	22	24	15	3	5		1	[CH88]
terminal intensive switchbox	16	23	24	8	7	5	4		[Lu85]
dense switchbox	17	15	19	3	11	5			[Lu85]
augmented dense switchbox	18	16	19	3	11	5			[Lu85]
modified dense switchbox	17	16	19	3	11	5			[CH88]
pedagogical switchbox	16	15	22	14	4	4			[CH88]

Table 1:

differs slightly from the representation in this paper. The input graph in the literature is obtained from a complete rectangular grid graph by removing the outer cycle, see Figure 8 (a). Hence, every terminal is incident to a unique edge, and obviously, every Steiner tree must contain this edge. It is easy to see that by contracting all pending edges an equivalent problem is obtained, see Figure 8 (b). The graph resulting this way is a complete rectangular grid graph with terminals on the outer face. This instance is the input to our problem.

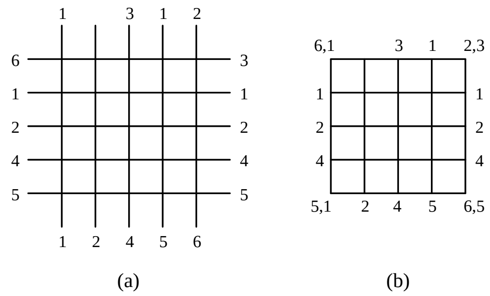


Figure 8:

In Table 2 we present the computational results we have obtained with our branch and cut algorithm. In Column 2 the objective function value of the best feasible solution we found is shown. The entries in Column 3 correspond to the objective function values of the linear program when no further violated constraints are found, i. e., when branching is performed for the first time. These values are obviously lower bounds for the whole problem. Column 4 contains the percental derivation of the best solution from the lower bound. Column 5 (resp. 6) gives the number of cutting plane iterations (resp. the number of nodes in the branching tree). Finally, the last column reports on the running times. The values are stated in minutes obtained on a SUN 4/50.

Example	Best Sol.	LP Value	Gap	Iter.	B & C	CPU-Time
difficult switchbox	469	469	0.0%	167	3	3452:55
more difficult switchbox	461	461	0.0%	124	5	3540:14
terminal intensive switchbox	537	537	0.0%	29	1	480:18
dense switchbox	-	∞	-	20	1	122:29
augmented dense switchbox	469	469	0.0%	30	1	583:50
modified dense switchbox	-	∞	-	48	1	686:27
pedagogical switchbox	343	341	0.6%	615	7	5230:35

Table 2:

For all instances we could either find an optimal solution or prove that the problem is infeasible. The latter situation occured in the two cases "dense switchbox" and "modified dense switchbox". To our knowledge, it was up to now open whether there exists a packing of Steiner trees in the Manhattan model for these instances. Actually, the two examples "modified dense switchbox" and "augmented dense switchbox" are extensions of the problem "dense switchbox" in which additonal tracks are added ("augmented dense switchbox" has an additional vertical track on the right and "modified dense switchbox" has an additional vertical track near the middle and an additional horizontal track at the bottom). In fact, these modifications have been introduced, because no routing algorithm could find a feasible solution for "dense switchbox" in any routing style. Whereas a Manhattan routing is known for the problem "augmented dense switchbox", the heuristics described in literature were unable to find one for "modified dense switchbox". Our algorithm yields a mathematical proof that, indeed, no routing routine can ever be successful for the latter example.

Second, the results show that except for the example "pedagogical switchbox" the objective function value of an optimal solution, provided it exists, was found without branching. The optimal LP-solution was, however, fractional and in two cases it took a few branching steps to find a feasible solution with the same value. Only for the test instance "pedagogical switchbox" the objective function value of the root LP differed from the optimal objective function value by 0.6%. This gap was closed by applying the enumerative phase of our code.
In all these cases the number of branch and cut nodes needed to solve the problems is very small (below 10). This indicates that the cutting planes we use as well as the corresponding separation routines perform quite well at least for the case of switchbox routing problems in Manhattan style.
Of course, there is a prize to pay: the high running times. The reason for that is that we aimed at finding an optimal solution or proving that no solution exists at all. If we just look at the time (measured in minutes) after which the lower bound deviates by at most 5, 2, 1 or 0 percent from the optimal value, the results look much more friendly. Table 3 shows in particular that in all these instances for which a feasible solution exists, the lower bound deviates at most 5% from the optimal objective function value within 4 : 11 minutes.

Example	5%	2%	1%	0%
difficult switchbox	2:48	54:59	248:02	1383:06
more difficult switchbox	2:35	35:20	268:43	1471:21
terminal intensive switchbox	4:11	26:52	99:48	343:00
augmented dense switchbox	1:38	1:38	146:13	583:50
pedagogical switchbox	2:15	84:19	199:38	5230:35

Table 3:

5 Knock Knee Versus Manhattan: A Comparison

From a practical point of view a very interesting question with probably never ending discussions is the question which model should be prefered: the knock-knee model or the Manhattan model. The theory says that in the knock-knee model two layers may not suffice, whereas in the Manhattan model they do. On the other hand, one can expect that the wiring length that is needed when Steiner trees are packed in the knock-knee model is smaller than in case of the Manhattan model. But, does the knock-knee model substantially provide shorter wiring lengths? We have tried to answer these questions for the problem instances introduced in the last section. In [GMW92b] we report in detail on our computaional experiences for the knock-knee model. The best lower and upper bounds we have obtained are summarized in Columns 2 and 3 of Table 4. We are able to solve all problem instances to optimality except the examples "dense switchbox" and "augmented dense switchbox". For comparison, the corresponding results for the Manhattan model are shown in Columns 4 and 5.

Example	Knock-knee Model		Manhattan Model	
	lower bound	upper bound	lower bound	upper bound
difficult switchbox	464	464	469	469
more difficult switchbox	452	452	461	461
terminal intensive switchbox	536	537	537	537
dense switchbox	438	441	∞	-
augmented dense switchbox	467	469	469	469
modified dense switchbox	452	452	∞	-
pedagogical switchbox	331	331	341	343

Table 4:

The results are quite different for different instances. For two of the examples the wiring length in the Manhattan model is just the same as in the knock-knee model though the solutions reported in [GMW92a] have knock-knees indeed (for pictures of the solutions, see [M92]). For three other problem instances the wiring length in the Manhattan model exceeds that in the knock-knee model by a small amount (for "difficult switchbox" by 5 (= 1.1%), for "more difficult switchbox" by 9 (= 2.0%) and for "pedagogical switchbox" by 12 (= 3.6%)). Of course, the shorter lengths in the knock-knee model must be paid by additional layers.

Since the percental increase in length is quite small one may tend to prefer the Manhattan model. However, for the examples "dense switchbox" and "modified dense switchbox", for which we could prove that there does not exist a feasible solution in the Manhattan model, we are able to find feasible solutions in the knock-knee model. This makes the knock-knee model more attractive.

Comparing the running times we observe similar phenomena (see Table 5). Some examples are quite easy for the knock-knee model but rather hard for the Manhattan model, and vice versa, some are solved quite fast in the Manhattan model, but are difficult in the knock-knee style. Based on these results we cannot decide whether one model is superior towards the other. The issue of choosing the "correct" model must be left to practitioners and depends on the chosen fabrication technology and the given design rules.

Example	CPU Time (min:sec)	
	Knock Knee	Manhattan
difficult switchbox	1564:15	3432:55
more difficult switchbox	983:23	3540:14
terminal intensive switchbox	3755:44	480:18
dense switchbox	1017:43	122:29
augmented dense switchbox	4561:41	583:50
modified dense switchbox	387:03	686:27
pedagogical switchbox	251:58	5230:35

Table 5:

Finally, we have compared our results with those published in the literature. In Table 6 we summarize the objective function values of the – to our knowledge – best Manhattan solution reported in the literature (Column 2). No entry means that we did not find any Manhattan solution for the corresponding problem instance that was published in the literature. In Column 3 the objective function value of the Manhattan solution that was obtained by our code is shown. The values differ from those reported in Table 2 and Table 4, respectively, by the total number of terminals of the original data due to preprocessing (see Section 4, page 13 for further explanations). For the instances "dense switchbox" and "modified dense switchbox" no Manhattan solution exists which is expressed by the symbol "$*$" in Column 3. For the problem instance "augmented dense switchbox" the solution given in [Lu85] is optimal, whereas for the two problems "difficult switchbox" and "terminal intensive switchbox" the solution found by our code improves the best solution reported in the literature by 2.2% and 2.7%, respectively.

Example	Best Manhattan Solution from the Literature	our Code
difficult switchbox	547 [JLSW90]	535
more difficult switchbox	-	527
terminal intensive switchbox	632 [Lu85]	615
dense switchbox	-	$*$
augmented dense switchbox	529 [Lu85]	529
modified dense switchbox	-	$*$
pedagogical switchbox	-	400

Table 6:

Of course, there are further routing algorithms presented in the VLSI literature. To our knowledge, all of them apply to the 2-layer model (i. e., the multiple layer model on a 2-dimensional grid graph), see, for instance, [LHT88], [JS86], [CH88], [JLSW90], [GH89], [TS88]. A comparison of the knock-knee or Manhattan model to the 2-layer model is difficult. In the 2-layer model two different nets may run on the same horizontal or vertical edges of the two layers. The number of consecutive edges that are used on both layers is usually limited in order to avoid so-called cross-talk problems. The value of this upper bound depends on the design rules and technological constraints, but is mostly neglected by the routing algorithms.

The fact that the wires can run on top of each other along arbitrary lengths may lead to routings with shorter wiring lengths than in the Manhattan model, because a solution in the Manhattan model is feasible for the 2-layer model. Nevertheless, we have compared our Manhattan solutions to the best 2-layer solutions reported in the literature for the instances described in Section 4. It turns out that for all examples for which a Manhattan solution exists, the objective function values are at most 1% worse than the objective function values of the corresponding 2-layer solutions. In fact, for the two examples "terminal intensive switchbox" and "augmented dense switchbox" the Manhattan solution provides the same wiring length, and for the switchbox "more difficult switchbox" we even find a better solution. For one of two examples ("modified dense switchbox") for which a Manhattan solution does not exist, the wiring length of the best 2-layer solution is by a value of 2 shorter than the one of the optimal knock-knee solution. For the instance "dense switchbox", we are not aware of any feasible routing that can be realized on two layers.

References

[BB84] M. L. Brady, D. J. Brown: *VLSI routing: Four layers suffice*, in: F. P. Preparata (ed.): "Advances in Computing Research", Vol. 2: VLSI theory, Jai Press, London, 1984, 245 – 258.

[BP83] M. Burstein, R. Pelavin: *Hierarchical wire routing*, IEEE Transactions on Computer-Aided-Design CAD-2, 1983, 223 – 234.

[GJ77] M.R. Garey, D.S. Johnson: *The rectilinear Steiner tree problem is $\mathcal{NP}$-complete*, SIAM J. Appl. Math. 32, 1977, 826 – 834.

[CH88] J. P. Cohoon, P. L. Heck: *BEAVER: A computational-geometry-based tool for switchbox routing*, IEEE Transactions on Computer-Aided-Design CAD-7, 1988, 684 – 697.

[GH89] S.H. Gerez, O.E. Herrmann: *Switchbox routing by stepwise reshaping*, IEEE Transactions on Computer-Aided-Design CAD-8, 1989, 1350 – 1361.

[GM90] M. Grötschel, C. L. Monma: *Integer polyhedra associated with certain network design problems with connectivity constraints*, SIAM Journal on Discrete Mathematics 3, 1990, 502 – 523.

[GMW92a] M. Grötschel, A. Martin, R. Weismantel: *Packing Steiner trees: polyhedral investigations*, Konrad-Zuse-Zentrum für Informationstechnik Berlin, Preprint SC 92-8, 1992.

[GMW92b] M. Grötschel, A. Martin, R. Weismantel: *Packing Steiner trees: a cutting plane algorithm and computational results*, Konrad-Zuse-Zentrum für Informationstechnik Berlin, Preprint SC 92-9, 1992.

[GMW93] M. Grötschel, A. Martin, R. Weismantel: *Packing Steiner trees: separation algorithms*, Konrad-Zuse-Zentrum für Informationstechnik Berlin, Preprint SC 93-2, 1993.

[JLSW90] J.M. Jou, J.Y. Lee, Y. Sun, J.F. Wang: *An efficient VLSI switch-box router*, IEEE Design and Test, 1990, 52 – 65.

[JS86] R. Joobbani, D.P. Siewiorek: *WEAVER: A knowledge-based routing expert*, IEEE Design and Test, 1986, 12 – 23.

[K72] R.M. Karp: *Reducibility among combinatorial problems*, in: R.E. Miller, J.W. Thatcher (eds.), "Complexity of Computer Computations", Plenum Press, New York, 1972, 85 – 103.

[L90] T. Lengauer: *Combinatorial algorithms for integrated circuit layout*, Wiley, Chichester, 1990.

[LHT88] Y.L. Lin, Y.C. Hsu, F. S. Tsai: *A detailed router based on simulated evolution*, in Proc. Int. Conf. Computer-Aided-Design, 1988, 38 – 41.

[Li84] W. Lipski: *On the structure of three-layer wireable layouts*, F. P. Preparata (ed.): "Advances in Computing Research", Vol. 2: VLSI theory, Jai Press, London, 1984, 231 – 244.

[Lu85] W. K. Luk: *A greedy switch-box router*, Integration 3, 1985, 129 – 149.

[M92] A. Martin: *Packen von Steinerbäumen: Polyedrische Studien und Anwendung*, Ph.D. Thesis, Technische Universität Berlin, 1992.

[S87] M. Sarrafzadeh: *Channel-routing problem in the knock-knee mode is $\mathcal{NP}$-complete*, IEEE Transactions on Computer-Aided-Design CAD-6, 1987, 503 – 506.

[Sz85] T. G. Szymanski: *Dogleg channel routing is $\mathcal{NP}$-complete*, IEEE Transactions on Computer-Aided-Design CAD-4, 1985, 31 – 40.

[TS88] P. Tzeng, C.H. Séquin: *Codar: a congestion-directed general area router*, in Proc. Int. Conf. Computer-Aided Design, 1988, 30 – 33.