Technische Universität Berlin

FACHBEREICH 3
MATHEMATIK

On the NP–Completeness of Channel and Switchbox Routing Problems

by

Stephan Hartmann

No. 542/1996

On the NP–Completeness of Channel and Switchbox Routing Problems

Stephan Hartmann*

Technische Universität Berlin
July 1997

Abstract

The design of integrated circuits has achieved a great deal of attention in the last decade. In the routing phase, an open layout problem has survived which is important from both the theoretical and the practical point of view. The channel routing problem has been known to be solvable in polynomial time when there are only 2–terminal nets, and is proved by Sarrafzadeh to be NP–complete in case that there exists nets containing at least six terminals. Also the 5–terminal case is claimed to be NP–complete. In our paper, we give a simple proof for the NP–completeness of the 5–terminal channel routing problem. This proof is based on a reduction from a special version of the satisfiability problem. Based on the techniques introduced in this paper and a result of [HSS97] stating the NP–completeness of the 3–terminal switchbox routing problem, we prove the 4–terminal 3–sided switchbox routing problem to be NP–complete.

1 Introduction

Very large scale integrated circuit layout (VLSI) is one of the amazingly growing areas in discrete mathematics and computer science due to both its practical relevance and its importance as a trove of combinatorial problems. Usually in VLSI design we distinguish between the phase of placing physical components and the subsequent *routing phase* realizing the conducting connections between them.

The routing phase itself consists of the layout problem and the corresponding layer assignment. We refer the reader to the book of Lengauer [Len90] and to the survey of Möhring, Wagner and Wagner [MWW95] for a detailed description of this process as well as for comprehensive surveys of the use of combinatorial and graph–theoretic methods in VLSI design. Here, we concentrate on the layout problem where the course of the wires to connect the cells in a single plane has to be determined.

Most generally, the problem is to find an edge–disjoint packing of Steiner trees in a given planar graph. To be more precise, we are given a graph $G = (V,E)$, the so–called *routing graph*, and k sets $N_1,\dots,N_k \subseteq V$ called *nets*. In this context, the elements of the nets are referred to as *terminals*. The task is to find k pairwise edge–disjoint Steiner trees $T_1,\dots,T_k \subseteq E$ such that T_i connects the terminals of net N_i. A solution of the Steiner tree packing problem is called *layout*. For arbitrary planar graphs, Kramer and van Leeuwen [KvL84] showed that the Steiner tree packing problem is NP–complete even if there are only two–terminal nets involved. Korte, Prömel and Steger [KPS90] complemented the NP–completeness result by Kramer and van Leeuwen by proving the NP–completeness of the problem if there are only two multi–terminal nets. If all terminals are assigned to the outer face of the routing graph, Okamura and Seymour [OS81] gave sufficient conditions for instances that can be solved in polynomial time.

*Technische Universität Berlin, MA 6–1, Straße des 17. Juni 136, 10623 Berlin, Germany, E-mail: hartmann@math.TU-Berlin.DE

The routing graphs arising in VLSI design are actually very special planar graphs, most frequently they are even rectangular grids, corresponding to the usual shape of physical layout areas. Such routing problems have been attacked by quite different methods ranging from purely bottom–up methods over floor–planning techniques up to polyhedral combinatorics (see, e.g., [Len90, GMW93]). There are two types of problems on a grid which are of particular importance, namely switchbox routing and channel routing. In both cases, all terminals are placed on the boundary of the grid. In switchbox routing the terminals may be placed on all four sides, or on exactly three sides in the restricted case of 3–sided switchbox routing. Channel routing is another special case of switchbox routing, placing the terminals only on the lower and the upper side of the grid.

There are two major routing models to distinguish. In the layout, all paths must have disjoint edges but they may meet at the intersection points of the grid. In the so–called *knock-knee model*, two different paths may cross or both change their direction (forming a double–bend, called knock–knee) at an intersection point of the grid. In the *Manhattan model*, knock–knees are not allowed.

From the theoretical perspective, the switchbox routing problem in the knock–knee model is completely solved since Preparata and Mehlhorn [MP86] presented a polynomial time algorithm for the two–terminal case while Hartmann, Schäffter and Schulz [HSS97] recently proved that the switchbox routing problem with 3–terminal nets is NP–complete.

For the Manhattan model, Szymanski [Szy85] showed that channel routing with 4–terminal nets is NP–complete, and hence, so is switchbox routing. This result is improved upon by Middendorf [Mid93] showing that even the 2–terminal channel routing problem in the Manhattan model is NP–complete.

For the channel routing problem in the knock–knee model, Sarrafzadeh [Sar87] proved the NP–completeness if some of the nets involved have six or more terminals. He also claimed (without giving a proof) the NP–completeness of the 5–terminal channel routing problem. In this paper, we give a much simpler proof even for the 5–terminal channel routing problem. The complexity status of the 3–terminal and 4–terminal channel routing problem remains open. We would like to mention that the given reduction is based on ideas in [Szy85] and [Sar87] and on new techniques introduced in [HSS97].

The NP–completeness result of Sarrafzadeh [Sar87] also applies for the 3–sided switchbox routing problem if some of the nets involved have six or more terminals. Using a result of [HSS97], we are able to reduce the gap between the polynomial solvable 2–terminal and the NP–complete multi–terminal case by showing the 4–terminal 3–sided switchbox routing problem to be NP–complete. Again, the complexity status of the 3–terminal case remains open.

On the other hand, there is a variety of polynomial time algorithms for the channel routing problem with only 2-terminal nets using at most d tracks where d is the maximum horizontal density of the given channel routing instance. An overview can be found in [MWW95]. Some of them are even linear time algorithms ([LP84], [MPS86]). Area-optimal layouts are produced by the algorithms of [Fra82] and [KWW89]. Layouts with a minimal total wire length are constructed by the algorithms of [FWW93] and [Wag93].

Approximation algorithms are described in [MPS86] stating that $2d-1$ tracks always suffice for a layout of multi–terminal nets. If all nets have at most three terminals then even $\lfloor 3d/2 \rfloor$ tracks suffice. The best known algorithm for the multi–terminal channel routing problem is given in [GK87]. Their algorithm has a guaranty for the number of tracks of $3d/2 + \mathcal{O}(d\ log\ d)$.

The paper is organized as follows. In Section 2, we define the 3–bounded 3–SAT problem and the channel routing problem and give a first, introductory description of the transformation. In Section 3, we give a new and simple proof for the NP–completeness of the 5–terminal channel routing problem. In Section 4, we apply our techniques to the 3–sided switchbox routing problem. Thereby, we prove the former problem to be NP–complete if some of the nets involved have four or more terminals. We conclude with some remarks in Section 5.

2 A Brief Description of the Reduction

An instance of the channel routing problem consists of a *routing region* and a set of *nets*. The routing region is assumed to be a rectangular grid, called *channel*, with n vertical lines and an arbitrary number of horizontal lines, also called *tracks*. The set of nets consists of k nets $N_1,..,N_k$, where each net is a set of so–called *terminals* which here are intersection points at the upper and the lower boundary of the grid.

A solution of the channel routing problem, called *layout* or *routing*, is given by pairwise edge–disjoint Steiner trees $T_1,...,T_k$ embedded in the grid such that T_i connects the terminals of net N_i, $i = 1,\ldots,k$.

Before explaining the basics of our reduction, we introduce some notions which prove useful in the discussions to follow. We number the tracks top–down and the vertical lines from the left to the right. The segment of the grid between the vertical lines with indices i and $i+1$ is called a *column* and is denoted by $\vec{\imath}$. The *(local) horizontal density* $d_h(i)$ of column $\vec{\imath}$ is defined as the number of nets that have to cross column $\vec{\imath}$, i.e., it is the number of nets that contain terminals at the left–hand side as well as terminals at the right–hand side of the column $\vec{\imath}$. We call the vertical line with index i *density–increasing* if $d_h(i) > d_h(i-1)$ and *density–decreasing* if $d_h(i) < d_h(i-1)$. Otherwise, we call the vertical line *density–preserving*. We call the number of tracks the *horizontal capacity* and the number of vertical lines the *vertical capacity* of the channel. The *free horizontal capacity* of a column is the difference of the horizontal capacity minus the horizontal density of the column.

The l–terminal channel routing problem

Instance: A rectangular grid consisting of n vertical lines and d horizontal lines. A collection $\{N_1,..,N_k\}$ of nets, each consisting of at most l terminals. Terminals are assigned to the intersection points at the upper and lower boundary of the grid. The number of horizontal lines d corresponds to the maximum horizontal density of the problem.

Question: Is there an edge–disjoint knock–knee routing for the nets in the given routing region?

We interchangeably use the term net for a set of terminals and for the realization of its Steiner tree in the layout. The respective meaning should always be clear from the context.

The following theorem was claimed (without giving a proof) by Sarrafzadeh [Sar87].

Theorem 2.1. *The 5-terminal channel routing problem is NP–complete.*

The mayor part of this paper is devoted to give a simple proof for this result. It is clear that the l–terminal channel routing problem is in NP. The following reduction is from a special version of the 3–SAT problem.

The 3–bounded 3–SAT problem

Instance: A set $\mathcal{X} = \{x_1,\ldots,x_N\}$ of Boolean variables, and a collection $\mathcal{C} = \{C_1,\ldots,C_M\}$ of clauses over $\mathcal{X}$. Thereby, for each variable x_i there are at most three clauses in $\mathcal{C}$ that contain either x_i or $\overline{x}_i$. Moreover, each clause contains at most three literals.

Question: Is there an assignment for the variables in $\mathcal{X}$ such that every clause in $\mathcal{C}$ is satisfied?

The 3–bounded 3–SAT problem was proved to be NP–complete by [GJ79, p. 259]. Without loss of generality, we assume that all clauses contain strictly more than one literal and that every Boolean variable occurs negated as well as unnegated.

The main ideas of the transformation are as follows.

- For each Boolean variable $x_i \in \mathcal{X}$ we introduce two nets X_i and $\overline{X}_i$, called *real–nets*. If x_i occurs in two clauses, X_i and $\overline{X}_i$ consist of four terminals each, otherwise they have five terminals. In order to route the real–nets, we introduce two horizontal lines for each variable x_i, called *variable–tracks*.

- A variable x_i is meant to be TRUE if and only if net X_i is routed below net $\overline{X}_i$ in the layout, and FALSE otherwise.
- Each clause C_j is modelled as a block B_j of consecutive vertical lines. The constructed channel consists of these clause blocks chained from the left to the right. Furthermore, we add a start block L to the left and an end block R to the right side of the channel (see Figure 1).

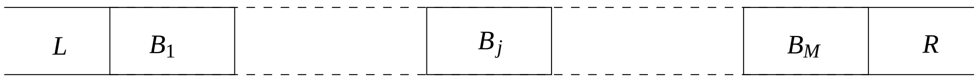

Figure 1: The coarse–structure of the resulting channel instance.

- In contrast to [Sar87], we do not need so–called enforcer blocks such that the number of terminals per net can simply be bounded by five. We insert at most three terminals of the real–nets corresponding to the occurrence of the Boolean variable x_i in the clauses. Furthermore, we introduce two additional terminals for each real–net, one in the start block L and one in the end block R.
- The link of the Boolean variables to the clauses in which they occur is essentially captured as follows. If the Boolean variable x_i appears in the clause C_j, both nets X_i and $\overline{X}_i$ have exactly one terminal on the same vertical line (called the *variable–line*) of the clause block B_j.
- These variable–lines are surrounded by a certain collection of vertical lines (and terminals of different nets placed on these lines) to guarantee that the terminals of X_i and $\overline{X}_i$ can be connected to its corresponding nets, respectively. This makes sure that each literal can be TRUE or FALSE. The collections of vertical lines and nets of the same topological structure are called *detour modules* explained in Subsection 3.2. The precise structure of clause blocks is described in Subsection 3.1.
- We introduce a bunch of *dummy nets* in the start and the end block in order to force the detour–nets to the lowest tracks.

Thereby, a 3–bounded 3–SAT instance I consisting of N Boolean variables and M clauses is transformed into an instance I' of the 5–terminal channel routing problem with $n = 2N + 4$ vertical, $m = 8N + 9M_2 + 11M_3 + 4$ horizontal lines, and $k = 6N + 4M + 4$ nets, where M_2 denotes the number of 2-literal and M_3 the number of 3-literal clauses of the 3–bounded 3–SAT instance, respectively.

3 The Transformation

In this section, we describe the transformation from the 3-bounded 3–SAT problem to the 5–terminal channel routing problem in detail. This leads to the first proof, the NP–completeness of the 5–terminal channel routing problem already claimed in [Sar87]. Additionally, our proof is much simpler than the proof of [Sar87] because we do not need structures like enforcer blocks. Hence, the number of involved nets is significantly smaller.

3.1 The Clause Block

In this subsection, we present the precise structure of the main structure of the transformation, the so–called clause block. There is exactly one clause block for each clause of the instance of the 3-bounded 3–SAT problem. Every clause block consists of one vertical variable–line for each Boolean variable of the associated clause. The variable–lines are surrounded by detour modules explained in detail in Sub–section 3.2. The length of each clause block depends on the number of literals in the corresponding clause.

For a given clause C, we introduce a variable–line for each variable occurring in C. If a variable x appears unnegated in C, we place a terminal of net X at the lower position of the variable–line associated with x and a terminal of net $\overline{X}$ at the upper position. If x appears negated in C, the assignment is vice versa (see Figure 2).

So far, we have mainly been concerned with the construction of the channel. To improve the accessibility of the discussions to follow, we turn for a moment to the interpretation point of view. As mentioned before, a variable x is meant to be TRUE if and only if net X is routed below net $\overline{X}$. Given the terminal assignment at the variable–lines and the interpretation of the routing of the real–nets, we can distinguish two possible routing types at each variable–line. If a variable x is TRUE (FALSE, respectively) and appears negated (unnegated) in the clause under consideration, then not both terminals of the associated variable–line can directly be connected to their dedicated track. Consequently, additional horizontal and vertical capacity is needed to route the necessary detour (see Figure 3 b). Such a routing is called a FALSE–routing. Every other kind of local routing (see, for instance, Figure 3 a) is called a TRUE–routing. It corresponds to a literal with value TRUE.

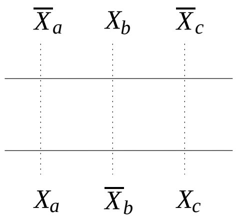

Figure 2: Var.–lines for $C = x_a \vee \overline{x}_b \vee x_c$

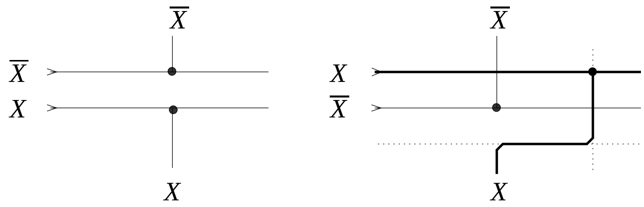

Figure 3: a) TRUE–routing b) FALSE–routing.

In order to allow a TRUE– or a FALSE–routing at every variable–line, the detour modules provide the required horizontal and vertical capacity.

3.2 The Detour Module

Detour modules surround the variable–lines. They are intended to provide the capacity needed for a FALSE–routing and, at the same time, to keep the horizontal and vertical density high enough to prevent a change of the vertical ordering of the nets.

Consider, say, the i–th detour module. This module consists of four vertical lines, and of seven terminals of four so–called *detour–nets* e_{i-1}, e_i, g_{i-1}, and g_i. These detour–nets serve to keep the capacity low which is caused by the special terminal assignment at the introduced vertical lines. Detour–nets have at most four terminals. The first terminal of the net e_{i-1} and the first two terminals of the net g_{i-1} are located in the previous detour module. The third terminal of g_{i-1} is placed on the upper endpoint of the first vertical line of the ith detour module. We put this line itself to the left of the variable–lines. The other three vertical lines of this detour module are placed to the right side of the variable–lines (see Figure 4). The vertical line directly to the right of the variable–lines is called the *detour–line*. Terminals of the detour–net g_i are assigned to the bottom endpoints of the second and the fourth of these vertical lines whereas the bottom endpoints of the first and the third vertical line of the detour module are used for the nets e_{i-1} and e_i, respectively. In order to force net g_i to be routed below net e_i as shown in Figure 4, we assign a terminal of net e_i to the top of the fourth vertical line. The exact terminal assignment within the detour module can be taken from Figure 4.

The name detour–line is motivated by its functionality. The detour caused by a FALSE–routing cannot be realized without occupying the vertical capacity of the associated detour–line. For the same reason, the detour module provides free horizontal capacity of one track between its first density–decreasing vertical line and its density–increasing detour–line. The remaining vertical lines of the detour module, however, are density–preserving. The capacity provided by the detour module may be used to realize a TRUE– as well as a FALSE–routing. Notice that the free horizontal capacity of one track of the detour module must be used in both cases (see Figure 4).

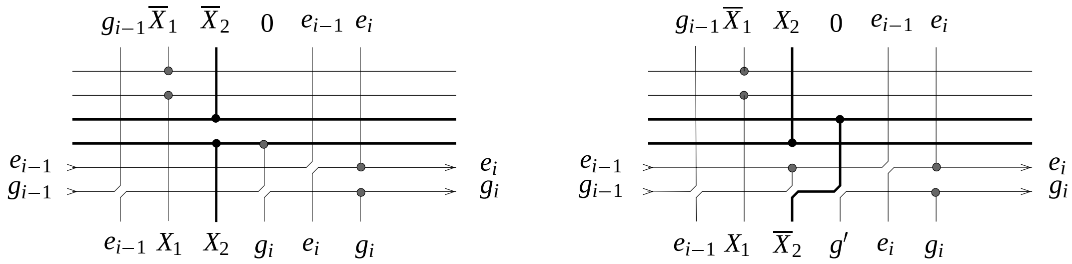

Figure 4: A layout for the detour module in a 2–literal clause block: a TRUE– and a FALSE–routing (left/right).

Up to this point, the transformation of the clauses is completely described. For a 3–literal clause, we embed three consecutive variable–lines into two detour modules. The nets of the second detour module are denoted with f_i and h_i. Thereby, the first vertical line of each detour module is placed on the left side of the variable–lines. The remaining vertical lines are on the right side. For a 2–literal clause, we embed a block of two variable–lines into one detour module (see Figure 4). Furthermore, we place a block of three consecutive vertical lines at the end of the clause block. We assign terminals of nets f_i, h_i and f_i to the top and f_{i-1}, h_{i-1} and h_i to the bottom of these lines, in this order. Thus, a clause block corresponding to a 3–literal clause consists of eleven vertical lines, and a clause block corresponding to a 2–literal consists of nine vertical lines. In order to route the detour nets, four horizontal lines are added to the channel routing problem.

Figure 4 depicts two variable–lines embedded in a detour module. The terminal assignment and two possible layouts for the nets X_2 and $\overline{X}_2$ inside the detour module are shown. If grid points at the upper boundary are not occupied by terminals, then this is denoted by zero as an index for the empty net.

3.3 Start & End Block

The function of the start block L and the end block R is to introduce the detour– and the real–nets, such that the density is at its maximum at the transition between block L and the left–most clause block B_1 as well as between the right–most clause block B_M and R, respectively. Within these new blocks, additional nets, so–called *dummy–nets*, induce an ordering of the nets at the transition to the clause blocks, such that the detour–nets are forced to the four lowest tracks. The start and the end block are symmetric, but in a reverse order. Both blocks consists $5N+5$ vertical lines.

The start block can clearly be divided into three sub–blocks. The first sub–block L_{ins} consist of $N+2$ density–increasing vertical lines. The detour–nets and some additional nets are inserted in L_{ins}. The corresponding terminal assignment is depicted in Figure 5. At the right side of this sub–block, the density is at its maximum, such that each net occupies exactly one track. The second sub–block L_{ind} consists of $2N+3$ density–preserving vertical lines. The following list of terminals is assigned the upper side of the channel: $(g_0, f_0, e_0, N, \overline{N}, .., 2, \overline{2}, 1, \overline{1})$. At the lower side of the channel, a shifted list is assigned to. This list is shifted by one position to the right while a terminal of net h_0 is assigned to the head of the list. The terminal of net $\overline{1}$ is omitted (see Figure 5). Due to this assignment, the detour–nets and some dummy–nets are forced to the lowest tracks, which will be proved by Lemma 3.1.

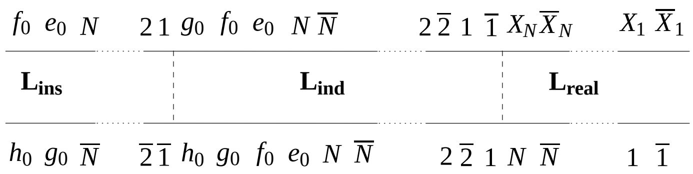

Figure 5: The start block L in detail

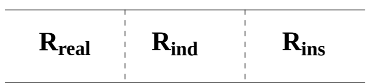

Figure 6: The structure of the end block R

The third sub–block L_{real} consists $2N$ density–preserving vertical lines. At each of these lines, one of the dummy–nets is replaced by the associated real–net (see Figure 5). For the end block R (see Figure 6), the detour–nets indexed with zero are replaced by detour–nets indexed with $M+1$. For the reason of simplification, the dummy nets of the start and the end block are not distinguished furthermore, but they are in fact different from each other.

3.4 The Proof for the 5–Terminal Case

At this point, the reduction from an instance of the 3–bounded 3–SAT problem to an instance of the 5–terminal channel routing problem is completely described. The NP–completeness proof has the following structure: Lemma 3.1 states that the detour–nets are always forced to the lowest track. Lemma 3.2 shows how a satisfying variable assignment for an instance of the 3–bounded 3–SAT problem induces a layout for the resulting channel routing problem. For the reverse direction, we show that the determination of the values of the Boolean variables from the given layout is well defined, i.e. the determined values are the same for each vertical line of the layout. This is captured by Lemma 3.3. Once this lemma is proved, it only remains to show that the assignment induced by the given layout satisfies all clauses of the underlying instance of the 3–bounded 3–SAT problem. This is done by Lemma 3.4.

Lemma 3.1 (Area of the Detour–Nets). *For every layout of the channel routing problem, the detour–nets e_i, f_i, g_i, and h_i, $i = 0,..,M+1$, occupy tracks $2N+1$ to $2N+4$. Furthermore, the nets e_i and f_i are routed above the nets g_i and h_i, $i = 0,..,M+1$, in every layout.*

Proof. Each net occupies exactly one track at the transition between L_{ins} and L_{ind} since horizontal density is at its maximum. Hence, at each of the density–preserving vertical lines of L_{ind}, the track–assignment must correspond to the terminal assignment, since there is neither horizontal nor vertical capacity to route a detour which would be necessary at a vertical line which terminal assignment contradicts the track assignment. This means that the track–assignment induced by L_{ind} is unique. Starting from the top, the track assignment is as follows: $(\overline{1},1,\overline{2},2,..,\overline{N},N,e_0,f_0,g_0,h_0)$. Since all real–nets are inserted from the upper side of the channel thereby replacing an associated dummy–net, the ordering of the detour–nets must not change. The same argumentation also holds for the detour-nets e_{M+1}, f_{M+1}, g_{M+1}, and h_{M+1} of the end block R.

It remains to show that this order also holds for the detour–nets of the clause blocks. In each clause blocks, only the ending detour–nets are replaced by new corresponding detour–nets. So, these new detour–nets cannot be routed to a higher track than the old detour–nets as all new detour–nets are inserted from the lower side of the channel. The last vertical lines of the detour modules are density–preserving and density is again at its maximum. Since detours cannot be routed at maximum density it is guaranteed that nets e_i and f_i are routed above the nets g_i and h_i, $i = 1,..,M+1$ according to their terminal assignment at those vertical lines. This completes the proof of Lemma 3.1. □

Lemma 3.2 (Existence of a Layout). *Every feasible solution of the 3–bounded 3–SAT problem induces a layout for the resulting 5–terminal channel routing problem.*

Proof. Given an instance of the 3–bounded 3–SAT problem and a satisfying variable assignment, we now describe how to obtain a layout for the resulting channel routing problem instance.

1. Route the sub–blocks L_{ind} and R_{ind} according to the track assignment as suggested in the proof of Lemma 3.1. Connect the terminals of the nets of L_{ins} and R_{ins} in respect to these track assignments.

2. Assign the first two real–nets X_1 and $\overline{X}_1$ to the highest free tracks. If variable x_1 is TRUE, route $\overline{X}_1$ above X_1, otherwise route them in reverse order. Connect both nets to their terminals in L_{real} and R_{real}. Continue in the same manner with the remaining real–nets.

3. Connect the terminals of the real–nets whenever the assignment of the terminals in the clause blocks corresponds to the ordering of the track assignment at the associated variable–lines. These are exactly the TRUE–routings.

4. Connect the remaining terminals of the real–nets on the upper side directly to their corresponding tracks. Connect the remaining terminals of the real–nets on the lower side by using the horizontal capacity in the area of the detour–nets and the vertical capacity of the detour–line as depicted in Figure 4, right side. These FALSE–routings can be realized as in each clause block, the number of the detour–lines is greater or equal as the number of necessary FALSE–routings.

5. Route the remaining detour–nets in a canonical way (refer to Figure 4).

This procedure yields a layout for the resulting channel routing problem. □

Lemma 3.3 (Detour Module). *For every layout of the channel routing problem, the track assignment of the real–nets does not change inside the detour module.*

Proof. Inside each detour module, there is a horizontal capacity of one between the density–decreasing vertical line on the left side of the variable–lines and the density–increasing detour–line on the right side of the variable–lines. For every layout, this horizontal capacity must be used by nets that occupy two tracks due to the terminal assignment of the detour module. In case of a TRUE–routing, these are the upper detour–nets e or f, and in case of a FALSE–routing, these are the real–nets which terminals are assigned to the bottom of the associated variable–lines (see Figure 4). The lack of more free horizontal capacity inside the detour module prevents changes of the track assignment since, as shown before, all tracks are occupied and at least horizontal capacity is necessary to re–route one net to another track. □

Lemma 3.4 (Satisfying Assignment). *Consider a layout for the resulting instance of the channel routing problem. Then, in every clause block, a* TRUE*–routing is realized for at least one variable of the associated clause.*

Proof. The proof follows by a simple contradiction argument. As mentioned before, a clause block corresponding to a 3–literal clause (2–literal clause, respectively) consists of 2 (1) detour–lines. At each detour-line, vertical capacity is available in order to route exactly one FALSE–routing. Suppose a layout of the channel routing problem implies a variable assignment which does not satisfy a clause C. Hence, at each variable–line of the corresponding clause block a FALSE–routing has to be realized. This means that for a 3–literal clause (2–literal clause, respectively), 3 (2) FALSE–routings have to be routed. Thus, the demand of vertical capacity is strictly greater than the number of detour–lines which contradicts the existence of a layout. This means that a layout of the channel routing problem implies a satisfying variable assignment for the instance of the underlying satisfiability problem which proves this lemma. □

Proof of Theorem 2.1. Obviously all introduced nets consist of at most five terminals. The theorem now follows directly from the above lemmas: Lemma 3.2 states that a satisfying variable assignment induces a layout of the corresponding instance for the resulting channel routing problem. Such a layout can be constructed by the algorithm which is given in the proof of Lemma 3.2. By Lemma 3.3, the variable assignment is shown to be well defined while Lemma 3.4 guarantees that every clause of the 3–bounded 3–SAT instance is satisfied by the obtained variable assignment. Hence, the 5–terminal channel routing problem is NP–complete. □

4 An Extension to 3–Sided Switchbox Routing

In the first part of this paper, we introduced techniques of constructing terminal assignments of sub–channels in order to fix nets to dedicated tracks. In this section, we apply these techniques and a result of [HSS97] to the 3–sided switchbox routing problem to prove its NP–completeness for the 4–terminal case. The channel and the 3–sided switchbox routing problem are special cases of the *switchbox routing problem*.

An instance of the switchbox routing problem consists of a routing region and a set of nets. Again, the routing region is assumed to be a rectangular grid, the so–called *switchbox*, with n vertical lines and m horizontal lines. The set of nets consists of k nets $N_1,..,N_k$, where each net is a set of terminals which are intersection points at the boundary of the grid.

A solution of the switchbox routing problem, called layout or routing, is given by pairwise edge–disjoint Steiner trees $T_1,...,T_k$ embedded in the grid such that T_i connects the terminals of net N_i, $i = 1,\dots,k$.

The l–terminal switchbox routing problem

Instance: A rectangular routing region consisting of n vertical and m horizontal lines. A collection $\{N_1,..,N_k\}$ of nets, each net consists of at most l terminals. The terminals are assigned to the intersection points at the boundary of the grid.

Question: Is there an edge-disjoint knock-knee routing for the nets in the given routing region?

The following result for the switchbox routing problem is established in [HSS97].

Theorem 4.1 ([HSS97]). *The 3-terminal switchbox routing problem is NP–complete.*

This means that the gap between the polynomial solvable 2–terminal case and the NP–complete multi–terminal cases is completely closed. For the channel routing problem, the gap is reduced from the 6–terminal to the 5–terminal case by Theorem 2.1 (see Table 1). In this section, we reduce the gap for the 3–sided switchbox routing problem proving the 4–terminal case to be NP–complete. The l–terminal *3–sided switchbox problem* is a special case of the switchbox routing problem where the terminals are only assigned to three sides of the boundary of the grid.

For the necessary transformation, we first apply the transformation from the 3–bounded 3–SAT problem suggested in [HSS97]. Then, we replace the right side of the switchbox by a sub–channel which fulfills the following tasks: it introduces all necessary nets and fixes them to dedicated tracks, so that we gain a transformation from the 3–bounded 3–SAT problem to the 4–terminal 3–sided switchbox routing problem. The disadvantage of this replacement is that the number of terminals of some nets increases from three to four. If we also replace the left side of the switchbox by a corresponding sub–channel, the number of terminals of some nets even increases to five. But, we gain an alternative proof for the NP–completeness of the 5–terminal channel routing problem. The replacement and its correctness is presented in detail in the next subsection.

4.1 The Replacement

We start with an instance of the 3–bounded 3–SAT problem and apply the transformation given in [HSS97] to gain a terminal assignment for a 4–sided switchbox. Terminals of a bunch of different extension–, detour–, gate– and sandwich–nets are assigned to the right boundary of this switchbox. Each of these nets consists of at most three terminals and exactly one terminal of these nets is assigned to the right boundary. For the reason of simplification, we denote these nets by 1 to $2K$ in this order from top to bottom. Replacing the right side of the switchbox by a sub–channel, we add another terminal to each of these nets. Furthermore, we introduce $2K$ dummy nets denoted by $\overline{1}$ to $\overline{2K}$ in order to induce a special track assignment.

The replacing channel consists of $6K-1$ vertical lines. It can clearly be decomposed into three sub–channels: R^s_{repl}, R^s_{ind}, and R^s_{ins} in this order from left to right. The terminal assignments of R^s_{ind} and R^s_{ins} are depicted in

Figure 7. Their functionality is already described in detail in Subsection 3.3. The terminal assignments of these sub–channels guarantee that the dummy nets $\overline{1}$ to $\overline{2K}$ are assigned to the tracks of the switchbox in this order from top to bottom at the first column left of R^s_{ind} which is already shown in the proof of Lemma 3.1.

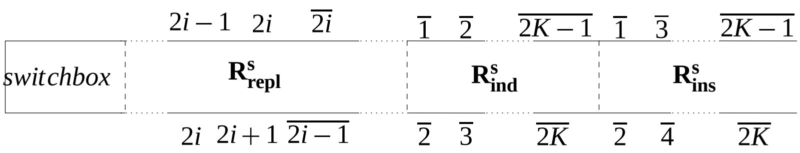

Figure 7: The replacing channel

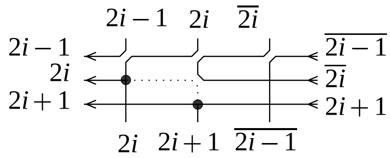

Figure 8: Routing detail in R^s_{repl}

The new and essential sub–channel R^s_{repl} consists of $3K$ vertical lines. In this sub–channel, we replace the dummy nets by the nets involved in the switchbox transformation, such that the track assignment left of R^s_{repl} is almost the same as the terminal assignment at the right side of the switchbox to be replaced. The sub–channel contains K modules each consisting of three vertical lines. For each $i \in \{1,..,K\}$, we assign terminals of the nets $2i-1$, $2i$, and $\overline{2i}$ to the top of these lines, and terminals of $2i$, $2i+1$, and $\overline{2i-1}$ to the bottom (see Figure 7). Notice that the lower terminal of net $2K+1$ which does not exist is replaced by a terminal of the dummy net $\overline{2K}$. Except local exchanges between nets which are direct neighbors, this terminal assignment induces the dedicated track assignment left of R^s_{repl}. This is proved by Lemma 4.2. With this lemma and Theorem 4.1, we are able to prove the 4–terminal 3–sided switchbox routing problem to be NP–complete.

Before formulating the lemma, we introduce some further notations. Analogously to a column $\vec{\imath}$, a *row* $\vec{\jmath}$ is the segment between the tracks j and $j+1$. The vertical density of a row $\vec{\jmath}$ is defined analogously to the horizontal density of a column $\vec{\imath}$. The Steiner tree resulting from an embedding of a 3–terminal net into the grid contains a node with degree 3. This so–called *uneven* point has to be placed on a grid point with even degree of 4, such that the remaining unused edge of this grid point cannot be used to embed another net. Each uneven point is connected to another one by a sequence of unused edges of the grid. We call this sequence a *virtual* path. Such a virtual path between uneven points of the Steiner trees associated with nets $2i$ and $2i+1$ is marked by a dotted line in Figure 8.

Lemma 4.2 (Replacing Channel). *A layout for the replacing channel exists and for every layout of the resulting 3–sided switchbox routing problem, the terminal assignment of the replacing channel induces the following track assignment at the first column at its left side:*

1. *Net* 1 *(respectively net* $2K$*) is assigned to the highest (resp. lowest) track of the 3–sided switchbox.*

2. *For each* $i \in \{1,..,K-1\}$*, the nets* $2i$ *and* $2i+1$ *are assigned to tracks* $2i$ *and* $2i+1$*, or vice versa.*

Proof. First, we prove that a layout exists for the replacing channel. For the sub–channels R^s_{ind} and R^s_{ins}, this has been done in the proof of Lemma 3.1, such that it suffices to give a solution for R^s_{repl}. At the first column left (right, respectively) of R^s_{repl}, the nets 1 ($\overline{1}$) to $2K$ ($\overline{2K}$) have to be assigned from top to bottom. The routing inside sub–channel R^s_{repl} can be found by applying the routing detail described in Figure 8 for each $i \in \{1,..,K\}$.

For the remaining proof of this lemma, we again focus on the sub–channel R^s_{repl}. Since the track assignment at its left side is unique as stated in the proof of Lemma 3.1, this sub–channel itself can be considered as a 3–sided switchbox S with $3K$ vertical and $2K$ horizontal lines. Terminals are assigned to the upper, lower and right side of S (see Figure 9 for $K=3$). Regarding this switchbox, three observations can be made.

1. For each $i \in \{0,..,K\}$, the horizontal density is at its maximum at columns $\vec{3i}$.

2. For each $j \in \{1,..,K\}$, the vertical density is at its maximum at rows $\overrightarrow{2j-1}$.

3. The considered switchbox S contains $2K$ 3–terminal nets, such that all layouts must contain K virtual paths connecting uneven points. These virtual paths must not cross columns or rows of maximum density.

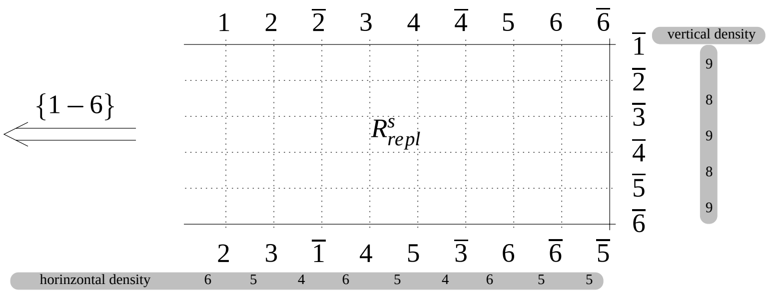

Figure 9: The 3–sided switchbox S (for $K = 3$) with horinzontal and vertical densities

In order to prove this lemma, it remains to show that net $2i-1$ ($2i$) cannot be assigned to track $2i$ ($2i-1$) for an arbitrary $i \in \{1, .., K\}$. Considering the described 3–sided switchbox, we suppose that at the first column left of S, net $2i-1$ is assigned to track $2i$ and net $2i$ is routed to track $2i-1$.

Due to the first observation, each net occupies exactly one track at column $3i\vec{-}3$. Additionally, net $2i$ is still routed above net $2i-1$ since the considered nets cannot exchange their tracks without crossing row $2i\vec{-}1$ according to the second observation. Within the sub–channel between columns $3i\vec{-}3$ and $\vec{3i}$, there exists two 3–terminal nets, namely $2i$ and $2i+1$, which have to be connected by a virtual path according to the third observation. This virtual path has to cross row $2i\vec{-}1$ which contradicts the first observation. Hence, a layout does not exist for the assumed track assignment. This proves the lemma. □

This yields the following result.

Theorem 4.3. *The 4–terminal 3–sided switchbox routing problem is NP–complete.*

Proof. Obviously, all nets consist of at most four terminals. Considering the track assignment which is induced by the replacing channel, we can apply Theorem 4.1 as differences to the dedicated track assignment of this switchbox are strictly local which is proved by Lemma 4.2. These local exchanges have no influence on the fact that the values of the Boolean variables are still well defined because for this proof it is only necessary that nets which belongs to different vertical regions are separated. As each of these regions consists of a great number of nets of the same topological type, exchanges between direct neighbors are of any importance. Nets of different topological type are still separated. Hence, it follows that the 4–terminal 3-sided switchbox routing problem is NP–complete. □

5 Concluding Remarks

In our paper, we have given a simple proof for the NP–completeness of the 5–terminal channel routing problem. Replacing the right side of the switchbox by a sub–channel R and applying a result of [HSS97], we have proved the 3–sided switchbox routing problem with 4–terminal nets to be NP–complete. If we also replace the left side of the switchbox by sub–channels L^s_{ins}, L^s_{ind}, and L^s_{repl} which are similarly constructed as the corresponding sub–channels for the right side, we gain an alternative proof for the NP–completeness of 5–terminal channel routing problem. But, the complexity status of the 3–terminal and the 4–terminal channel routing problem and the 3–terminal 3–sided switchbox routing problem remains open. Table 1 displays an overview on the complexity status of the channel and switchbox routing problems under consideration.

Acknowledgements

The author would like to thank Markus W. Schäffter and Andeas S. Schulz for proof–reading this paper. The author is grateful to Dorothea Wagner for helpful comments and discussions.

<table>
<tr><th rowspan="2">problems</th><th colspan="5">number of terminals</th></tr>
<tr><th>2</th><th>3</th><th>4</th><th>5</th><th>>5</th></tr>
<tr><td>channel routing</td><td>P
[MWW95]</td><td colspan="2">open</td><td>NP–c
[Theorem 2.1]</td><td>NP–c
[Sar87]</td></tr>
<tr><td>3–sided switchbox routing</td><td rowspan="2">P
[MP86]</td><td>open</td><td colspan="3">NP–c
[Theorem 4.3]</td></tr>
<tr><td>switchbox routing</td><td colspan="4">NP–c
[HSS97]</td></tr>
</table>

Table 1: Overview on the complexity status of channel and switchbox routing problems

References

[Fra82] A. Frank. Disjoint paths in a rectilinear grid. *Combinatorica*, 2(4):361 – 371, 1982.

[FWW93] M. Formann, D. Wagner, and F. Wagner. Routing through a dense channel with minimum total wire length. *Journal of Algorithms*, 15:267 – 283, 1993.

[GJ79] M. R. Garey and D. S. Johnson. *Computers and Intractability – A Guide to the Theory of NP–Completeness*. W. H. Freeman and Company, New York, 1979.

[GK87] S. Gao and M. Kaufmann. Channel routing of multiterminal nets. In *Proceedings of the 28th Annual Symposium on Foundations of Computer Science, FOCS'87*, pages 316–325, 1987.

[GMW93] M. Grötschel, A. Martin, and R. Weismantel. Routing in grid graphs by cutting planes. In G. Rinaldi and L. Wolsey, editors, *Third IPCO Conference*, pages 447 – 461, 1993.

[HSS97] S. Hartmann, M. Schäffter, and A. S. Schulz. Switchbox routing in VLSI design: Closing the complexity gap. In F. d'Amore, P. G. Franciosa, and A. Marchetti-Spaccamela, editors, *Proc. of 22th International Workshop on Graph–Theoretic Concepts in Comp. Sc. (WG'96)*, number 1197 in Lecture Notes in Comp. Sc., pages 196–210. Springer: Berlin, 1997.

[KPS90] B. Korte, H.-J. Prömel, and A. Steger. Steiner trees in VLSI–layout. In B. Korte, L. Lovasz, H.-J. Prömel, and A. Schrijver, editors, *Paths, Flows and VLSI–Layout*, pages 185–214. Springer Verlag, 1990.

[KvL84] M. R. Kramer and J. van Leeuwen. The complexity of wire routing and finding minimum area layouts for arbitrary VLSI circuits. In F. P. Preparata, editor, *Advances in Computing Research, Vol. 2 VLSI theory*, pages 129–146. JAI Press, Reading, MA, 1984.

[KWW89] R. Kuchem, D. Wagner, and F. Wagner. Area–optimal three layer channel routing. In *Proc. 30th Ann. Symp. on Foundations of Computer Science (FOCS'89)*, pages 506 – 511. IEEE, 1989.

[Len90] Th. Lengauer. *Combinatorial Algorithms for Integrated Circuit Layout*. Teubner/Wiley&Sons, 1990.

[LP84] W. Lipski Jr. and F. P. Preparata. Optimal three–layer channel routing. In *IEEE Trans. Comp.*, pages 427 – 437. IEEE, 1984.

[Mid93] M. Middendorf. Manhattan channel routing is NP–complete under truly restricted settings. Preprint, Universität Karlruhe, 1993. To appear in the Chicago Journal of Theoretical Computer Science.

[MP86] K. Mehlhorn and F. P. Preparata. Routing through a rectangle. *Journal of the Association for Computing Machinery*, 33:60 – 85, 1986.

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

[MWW95] R. H. Möhring, D. Wagner, and F. Wagner. VLSI network design. In M. O. Ball, T. L. Magnanti, C. L. Monma, and G. L. Nemhauser, editors, *Network Routing*, volume 8 of *Handbooks in Operations Research and Management Science*, chapter 8, pages 625 – 712. Elsevier, 1995.

[OS81] H. Okamura and P. D. Seymour. Multicommodity flows in planar graphs. *Journal of Computer Theory*, 31:75 – 81, 1981.

[Sar87] M. Sarrafzadeh. Channel–routing problem in the knock–knee mode is NP–complete. *IEEE Transaction on Computer–Aided Design*, 6:503 – 506, 1987.

[Szy85] T. G. Szymanski. Dogleg channel–routing is NP–complete. *IEEE Transaction on Computer–Aided Design*, 4:31 – 41, 1985.

[Wag93] D. Wagner. Optimal routing through dense channels. *Journal of Computational Geometry & Applications*, 3(3):269 – 289, 1993.

Reports from the group

“Algorithmic Discrete Mathematics”

of the Department of Mathematics, TU Berlin

559/1997 *Matthias Müller–Hannemann and Karsten Weihe:* Improved Approximations for Minimum Cardinality Quadrangulations of Finite Element Meshes

554/1997 *Rolf H. Möhring and Matthias Müller–Hannemann:* Complexity and Modeling Aspects of Mesh Refinement into Quadrilaterals

551/1997 *Hans Bodlaender, Jens Gustedt and Jan Arne Telle:* Linear-Time Register Allocation for a Fixed Number of Registers and no Stack Variables

549/1997 *Andreas S. Schulz and Martin Skutella:* Random-Based Scheduling: New Approximations and LP Lower Bounds

542/1996 *Stephan Hartmann:* On the NP–Completeness of Channel and Switchbox Routing Problems

536/1996 *Cynthia A. Phillips, Andreas S. Schulz, David B. Shmoys, Cliff Stein, and Joel Wein:* Improved Bounds on Relaxations of a Parallel Machine Scheduling Problem

535/1996 *Rainer Schrader, Andreas S. Schulz, and Georg Wambach:* Base Polytopes of Series-Parallel Posets: Linear Description and Optimization

533/1996 *Andreas S. Schulz and Martin Skutella:* Scheduling–LPs Bear Probabilities: Randomized Approximations for Min–Sum Criteria

530/1996 *Ulrich H. Kortenkamp, Jürgen Richter-Gebert, Aravamuthan Sarangarajan, and Günter M. Ziegler:* Extremal Properties of 0/1-Polytopes

524/1996 *Elias Dahlhaus, Jens Gustedt and Ross McConnell:* Efficient and Practical Modular Decomposition

523/1996 *Jens Gustedt and Christophe Fiorio:* Memory Management for Union-Find Algorithms

520/1996 *Rolf H. Möhring, Matthias Müller-Hannemann, and Karsten Weihe:* Mesh Refinement via Bidirected Flows: Modeling, Complexity, and Computational Results

519/1996 *Matthias Müller-Hannemann and Karsten Weihe:* Minimum Strictly Convex Quadrangulations of Convex Polygons

517/1996 *Rolf H. Möhring, Markus W. Schäffter, and Andreas S. Schulz:* Scheduling Jobs with Communication Delays: Using Infeasible Solutions for Approximation

516/1996 *Leslie A. Hall, Andreas S. Schulz, David B. Shmoys, and Joel Wein:* Scheduling to Minimize Average Completion Time: Off-line and On-line Approximation Algorithms

515/1996 *Christophe Fiorio and Jens Gustedt:* Volume Segmentation of 3-dimensional Images

514/1996 *Martin Skutella:* Approximation Algorithms for the Discrete Time-Cost Tradeoff Problem

509/1996 *Soumen Chakrabarti, Cynthia A. Phillips, Andreas S. Schulz, David B. Shmoys, Cliff Stein, and Joel Wein:* Improved Scheduling Algorithms for Minsum Criteria

508/1996 *Rudolf Müller and Andreas S. Schulz:* Transitive Packing

506/1996 *Rolf H. Möhring and Markus W. Schäffter:* A Simple Approximation Algorithm for Scheduling Forests with Unit Processing Times and Zero-One Communication Delays

505/1996 *Rolf H. Möhring and Dorothea Wagner:* Combinatorial Topics in VLSI Design: An Annotated Bibliography

504/1996 *Uta Wille:* The Role of Synthetic Geometry in Representational Measurement Theory

502/1996 *Nina Amenta and Günter M.Ziegler:* Deformed Products and Maximal Shadows of Polytopes

500/1996 *Stephan Hartmann and Markus W. Schäffter and Andreas S. Schulz:* Switchbox Routing in VLSI Design: Closing the Complexity Gap

498/1996 *Ewa Malesinska, Alessandro Panconesi:* On the Hardness of Allocating Frequencies for Hybrid Networks

496/1996 *Jörg Rambau:* Triangulations of Cyclic Polytopes and higher Bruhat Orders

Reports may be requested from:

S. Marcus
Fachbereich Mathematik, MA 6–1
TU Berlin
Straße des 17. Juni 136
D-10623 Berlin – Germany

e-mail: Marcus@math.TU-Berlin.DE

Reports are available via anonymous ftp from:

ftp.math.tu-berlin.de
cd pub/Preprints/combi
file Report–<number>–<year>.ps.Z