PERIODICA POLYTECHNICA SER. EL. ENG. VOL. 38, NO. 2, PP. 191–196 (1994)

OPTIMAL ROUTING ON NARROW CHANNELS

Tibor SZKALICZKI

Department of Mathematics and Computer Science
Faculty of Electrical Engineering and Computer Science
Technical University of Budapest
H–1521 Budapest, Hungary
email: sztibor@vma.bme.hu
phone: 463–1111/3150

Received: Sept. 27, 1994

Abstract

Channel routing is one of the basic problems in VLSI routing. While the minimum width can be found in linear time in the single row routing problem, the complexity of the channel routing problem is not fully understood yet. A solution can be found, even in linear time, in the unconstrained model, but the complexity of determining the minimum width is not known. The present article concentrates on the Manhattan model where horizontal and vertical wire segments are positioned on different sides of the board. In this case, the routing problem is known to be NP-complete. Hence there is no hope to find an algorithm whose running time is polynomial both in the length and the width of the channel. The width of the channel is usually much smaller than the length, thus, an algorithm, whose running time is exponential in the width and polynomial in the length can be efficient in the case of a narrow channel. We show that the channel routing problem in the Manhattan model is solvable in linear time if the length of the input is proportional to the length of the channel, and the width does not belong to the input.

Keywords: channel routing, NP-complete, minimum channel width, minimum wire length.

A channel given with length n and width w. The terminals appear on both sides of the channel. A net is a collection of terminals. A channel routing problem is a set of pairwise disjoint nets. The solution of a routing problem is a set of subgraphs (wires) where each subgroup connects all the terminals of the corresponding net under the conditions of the wiring model. In the restricted version of the Manhattan model each wire could occupy only one horizontal row (track). This model is called the dogleg-free model. We need an algorithm which finds a solution for the given routing problem in the dogleg-free model if there is any or else indicates that there is no solution. LA PAUGH (1980) and SZYMANSKI (1985) proved that the channel routing in the Manhattan model was NP-complete. We suppose that the width of the channel is small ,thus, the algorithm can be efficient if its running time is exponential in the width and polynomial in the length.

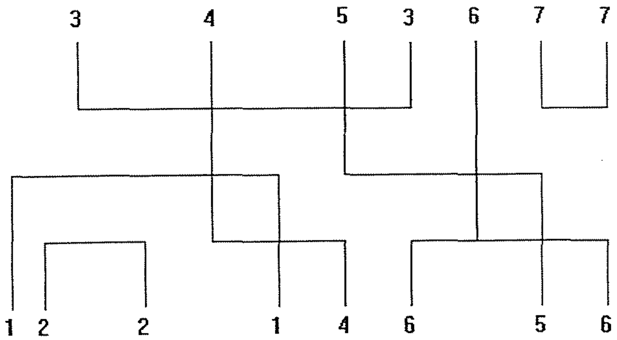

Fig. 1a. Solution of a routing problem

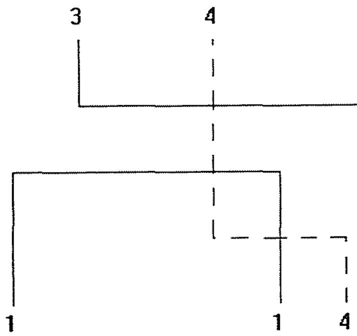

Fig. 1b. One of the profiles belonging to the first 3 wires.
Continuous line: wires forming the profile
Dotted line: 4th wire.

The algorithm proceeds on the channel from left to right. It takes the nets one after the other. It takes a net earlier if its leftmost terminal is nearer to the left end of the channel. If two nets begin in the same column, no matter which one is taken first. The actual net has to be connected, it means in the dogleg-free model that it has to be assigned to an appropriate free horizontal line. Suppose that the first i nets have been connected. Which previously placed wires influence the rest of the process? The wires passing a column containing terminals of not connected nets. All nets not connected yet nets start at or after the beginning of the last connected net while none of the already placed wires begin after the beginning of the last one. Hence, all the already placed wires reaching a column containing not connected terminals pass the column belonging to the beginning of the last placed wire. So the number of such wires is at most the width of the channel (w). Consequently, the arrangement of at most w wires influences the remaining process. At most w wires passing a common column can be assigned to tracks in at most $w!$ different ways. Let the arrangements of wires influencing the placement of the $(i+1)$th or later wires be called profiles belonging to the first i wires, see *Fig. 1.*

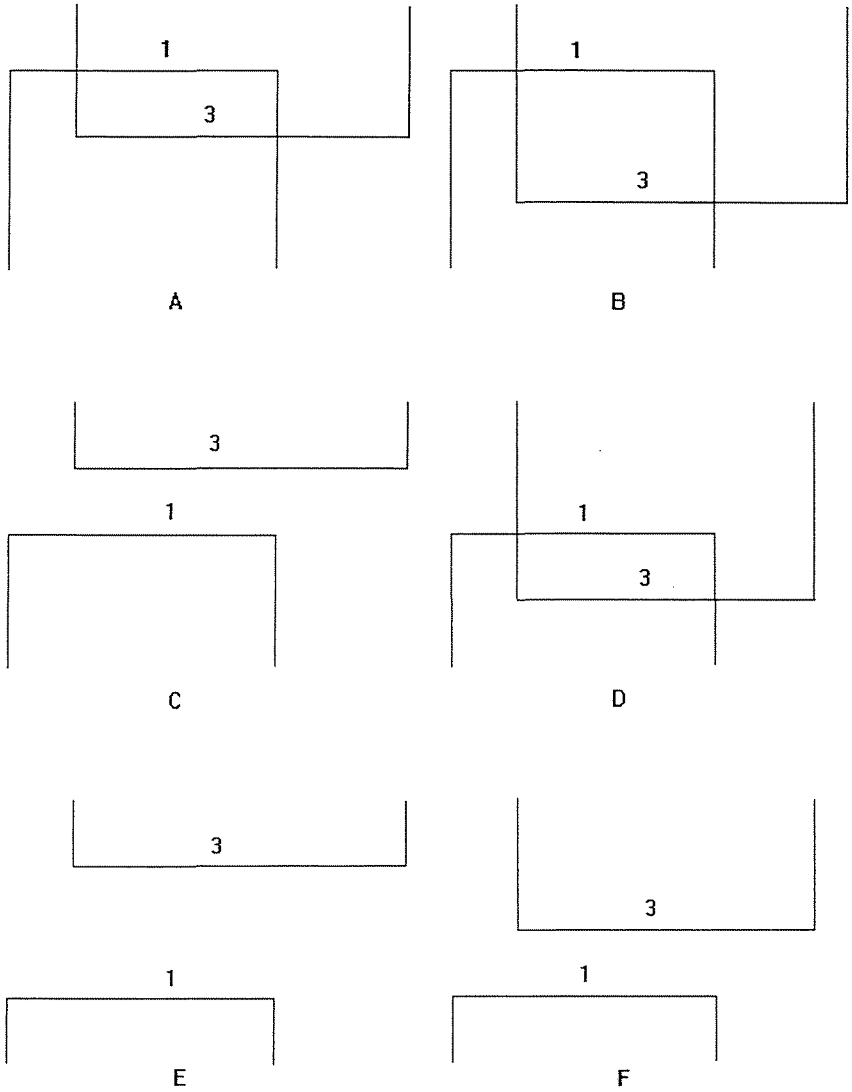

Fig. 2. All profiles belonging to the first 3 wires

The algorithm:

Let i denote the number of wires already placed.

0. Initially $i = 0$, the corresponding profile is the empty profile.
1. $i = i + 1$. Take the ith net.
2. Take the profiles belonging to the first $i - 1$ wires one after the other. See *Fig. 2* if $w = 3$, $i = 4$.
3. Try to continue the profile with the ith wire in all the possible ways. (Try to place the ith wire in every empty line of the profile in such a way that its vertical segments don't overlap vertical segments of another wire.) See *Fig. 3*.

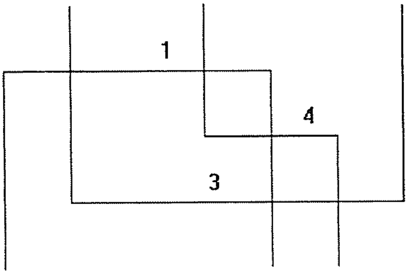

Fig. 3. Profile B of *Fig. 2* can be continued only in one way with the 4th wire

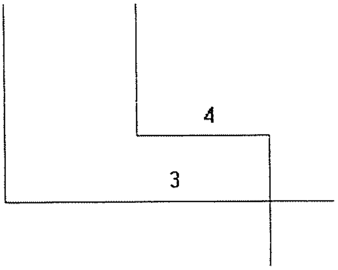

Fig. 4. The profile arising after placing the 4th wire in profile B of *Fig. 2*

4. Determine which profiles belonging to the first i wires arise after placing the ith wire, see *Fig. 4*. If we have already got this profile, we don't have to do anything with it (unless we are searching for an optimal solution). Else we got a new profile belonging to the first i wires. Store this profile (we will use it at the next net). The corresponding placement of the first i wires can be stored efficiently, too.
5. If any profile belonging to the first $i - 1$ wires is left, go to Step 2.

6. a) If there was no profile which was able to be continued with the ith wire, there is no solution.
   b) If no wire is left, we are ready, we have found a solution. The empty profile belongs to the last wire. The corresponding placement of the wires can be easily retrieved.
   c) Otherwise go to Step 1.

Running time:
the number of nets: $\leq n$
the number of possible profiles of the same wires: $\leq w!$
the number of possible placements of a wire in a profile: $\leq w$
Step 4: $\leq c \cdot w$ (c is a constant)
Total : $\leq c \cdot n \cdot w! \cdot w^2$

The algorithm can be easily modified to find the solution with minimum wire length. (No polynomial time algorithm is known which determines the minimum wire length even in single row routing problem where every terminal is located at the upper boundary.) In this case we have to store the minimum wire length belonging to the placement of the corresponding wires in addition to the profile.

We have implemented the algorithm on IBM PC. The program finds the solution with minimum wire length. The program was tested on an IBM PC 486 (33 MHz). The solution was found for $w < 8$ but for $w = 8$ the program failed because of insufficient memory. The example of *Fig. 5* consists of 18 wires, the width of the channel is 7, the length of the channel is 35. The minimum wire length is 344. The program determined the solution in 0.66 s. The solution of *Fig. 6* belongs to a channel with twice that length. The minimum wire length is 694, the running time is 0.83 s. The channel consists of relatively many wires and there are a lot of opposite vertical segments. Hence, the number of possible profiles decreases after placing the first few wires. In such a case the algorithm finds the solution especially quickly. That is why the running time of the second example increases less than expected.

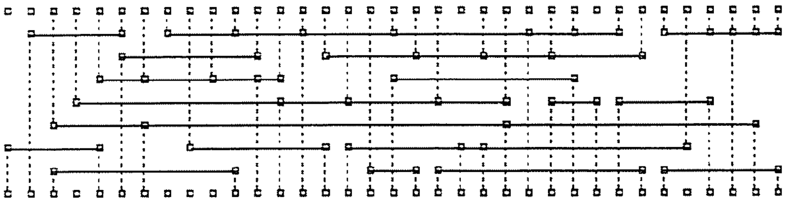

Fig. 5.

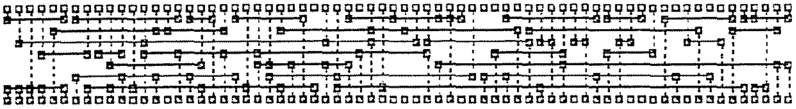

Fig. 6.

Acknowledgement

Useful conversations with A. Recski are gratefully acknowledged.

References

La Paugh, A. S. (1980): A Polynomial Time Algorithm for Optimal Routing around a Rectangle, *Proc. 21st FOCS Symp.*, 1980, pp. 283–293.

Szymanski, T. G. (1985): Dogleg Channel Routing is NP-complete, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, CAD–4(1), pp. 31–41.