# AN EFFICIENT ALGORITHM FOR WIDTHS OF CHANNEL ROUTING WITH GIVEN HORIZONTAL CONSTRAINT GRAPH

**Xianya Geng**[1]
**Xianwen Fang**
**Dequan Li**
**Jing Chu**

*Department of Mathematics and Physics*
*Anhui University of Science and Technology*
*Huainan, Anhui 232001*
*China*

**Abstract.** In VLSI design, One of the most important detailed routings is the channel routing. Channel routing in the 2-layer Manhattan model is one of the most investigated problem in VLSI design. In this paper, we consider the channel with horizontal constraint graph is a star. An efficient graph theoretic algorithm is presented, compared with the latest results, our algorithm yields a better bound on the width of the channel.
**Keywords:** channel routing, Manhattan model, VLSI.
**AMS Subject Classification:** 05C50, 15A18.

## 1. Introduction

In VLSI design the problem of completing the necessary interconnections among different modules is known as the routing problem. Typically, the routing problem can be divided into two steps due to the problem complexity: global routing and detailed routing. One of the most important detailed routing is channel routing [9,10,12]. The channel routing problem (CRP) is the problem of interconnecting all the nets in a channel using minimum possible routing area.

We use the expression of graph theory to describe the channel routing problem, a channel is defined by a rectangular grid $G$ of size $(w+2)\times n$ consisting of horizontal tracks (numbered from 0 to $w+1$) and vertical columns (numbered from 1 to $n$), where $w$ is the *width* and $n$ is the *length* of the channel. Top and bottom sides points of $G$ are called *terminals*.

A *channel routing problem* is a set $\mathcal{N}=\{N_1,\dots,N_t\}$ of pairwise disjoint nets. A channel routing problem is called *bipartite* if each net contains exactly two terminals, one on the top, and one on the bottom side. A channel routing problem is *dense* if each terminal on the top and bottom sides belongs to some net. A net is called *trivial* if it consists of two terminals which are situated in the same column. In this paper we always assume that each net contains at least two terminals.

---

[1]Corresponding author. E-mail: gengxianya@sina.com.

There are two constraints for the net in a channel: horizontal constraint and vertical constraint.

The constraint that two nets cannot overlap on the horizontal layer is called the *horizontal constraint.* Let $l_i$ be the leftmost and $r_i$ be the rightmost column of net $i$. A net $i$ is said to *span* the $c$-th column if $l_i \leq c \leq r_i$. The set of columns $[l_i, r_i]$ is called the span of net $i$.

There is a horizontal constraint between net $i$ and net $j$ if and only if their spans overlap. The horizontal constraints are often represented by an undirected graph, the horizontal constraint graph (HCG) (see Figure. 1), where vertices represent the nets and edges represent the horizontal constraints.

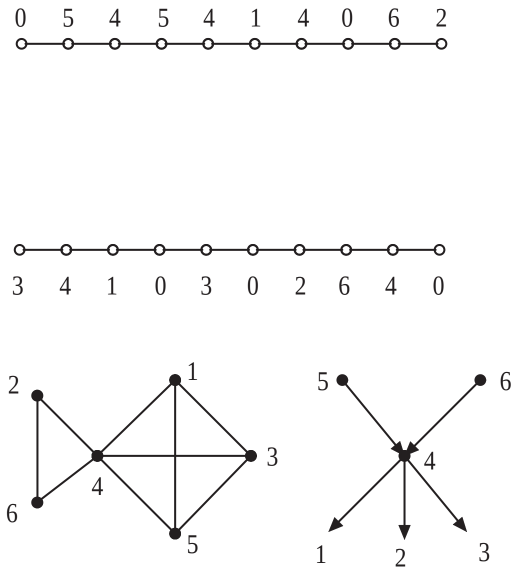


Figure 1.

Let $Z_i$ be the set of nets that span the $i-$th column, $d_{\max} = \max\{|Z_i| : i$ is a column$\}$ is called the density of the CRP. Clearly, $d_{\max}$ is a lower bound on $S^*$ because nets spanning the same column cannot be assigned to the same track.

The constraint that two nets cannot overlap on the vertical layer is called the *vertical constraint*. Note that if net $i$ connects to the $c$-th column in the top row and net $j$ connects to the $c$-th column in the bottom row, $i \neq j$, then net $i$ must be assigned to a track *higher* than net $j$. In this case, we say that net $i$ must precede net $j$ and there is a vertical constraint from $i$ to $j$ . The vertical constraints define a partial ordering between nets. The vertical constraints are often represented by a directed graph, the vertical constraint graph (VCG) (see Figure.1), where vertices represent the nets and arcs represent the vertical constraints.

A solution of a channel routing problem is said to belong to the Manhattan model if consecutive layers contain wire segments of different directions only. That is, layers with horizontal(east-west) and vertical (north-south) wire segments alternate. In this paper we restrict that the length of the channel cannot be extended by introducing extra columns.

2-layer Manhattan routing has always been one of the most popular and most investigated problems in VLSI routing. The first classic result in the topic of VLSI routing is probably Gallai's linear time algorithm that solves the single row routing problem – a special case of channel routing – with optimal width in the 2-layer Manhattan model.

**Theorem 1.1.** (T. Gallai [2]) *The minimum width of a solution of a single row routing problem in the* 2*-layer Manhattan model is equal to the density of the problem. Moreover, such a minimum width routing can be found in linear time.*

The proof makes use of the fact that interval graphs are perfect. A horizontal interval is associated with every net, stretching from its leftmost terminal to its rightmost terminal. The density is equal to the clique number of the corresponding interval graph. A coloring with an equal number of colors can easily be transformed into an optimal width routing.

In light of the above theorem it might be surprising that channel routing turns out to be much more complicated than single row routing.

**Theorem 1.2.** (T.G. Szymanski, 1985 [10]) *It is NP-complete to decide whether a channel routing problem is solvable in the* 2*-layer Manhattan model with width at most $k$* (*where $k$ is part of the input*).

In [9], the authors restrict the problem of 2-layer Manhattan channel routing, they do not allow extra empty columns to be added to the grid. They give a complete characterization of all specifications that are solvable if the width can be arbitrarily large, but the length of the channel cannot be extended by introducing extra columns. Furthermore, the author present a linear time algorithm to solve these specifications with a width at most constant times the length of the problem.

**Theorem 1.3.** (Dávid Szeszlér [8]) *A channel routing problem is not solvable in the* 2*-layer Manhattan model* (*with an arbitrary width*) *if and only if it is bipartite, dense and has at least one non-trivial net. Moreover, if a specification is solvable then it can be solved with width at most $\frac{3}{2}n$ in the bipartite, and $\frac{7}{4}n$ in the general case* (*where $n$ is the length of the channel*).

## 2. Main results

Recall that a channel routing problem is called bipartite if each net contains exactly two terminals, one on the top, and one on the bottom side. Now we consider horizontal constraint graph of the net in this problem is a star.

**Theorem 2.1.** *If a channel with horizontal constraint graph be a star, then the vertical constraint graph has directed path with length at most* 1.

**Proof.** Let the horizontal constraint graph $G$ has $k$ vertices, denoted by $v_1, v_2, ..., v_k$. Without loss of generality, we let the vertex $v_k$ has degree $k-1$. Since the graph $G$ is a star, each vertex of vertices $v_1, v_2, ..., v_{k-1}$ has an edge joined with vertex $v_k$. Then is to say, each net of $v_1, v_2, ..., v_{k-1}$ has horizontal constraint with net $v_k$ and each two nets belong to $v_1, v_2, ..., v_{k-1}$ have not horizontal constraint.

Now, we proof by contradiction. If these is an direct path belongs to the vertical constraint graph has length $s(s \geq 2)$, denoted by $u_1, u_2, ..., u_s$. Recall that if two nets have vertical constraint, then they also have horizontal constraint. Then the net $u_i$ and $u_{i+1}(1 \leq i \leq s-1)$ have horizontal constraint, the vertex $u_i$ and $u_{i+1}(1 \leq i \leq s-1)$ have edge in horizontal constraint graph, a contradiction.

So if a channel with horizontal constraint graph $v_1, v_2, \ldots, v_k$ be a star, then the vertical constraint graph has directed path with length at most 1, and if the vertical constraint graph has directed path with length 1, one of the vertex belong to this path has the largest degree. Since we consider channel routing problem is bipartite, that is all the net has two terminals, the directed path with length 1 of vertical constraint graph have at most two. ∎

### 2.1. Our Algorithm

An algorithm for 2-layer Manhattan routing problem with horizontal constraint graph be a star, based on the notions of horizontal constraint graph and vertical constraint graph, is presented in this section. The algorithm proceeds in three cases outlined below.

Route all trivial nets straight down in the obvious fashion. Henceforth we do not include these columns and nets below.

#### 2.1.1. Algorithm

**Case 1.** Vertical constraint graph of the net have no directed path, that is to say, vertical constraint graph have not edge, each two nets have no vertical constraint.

In this case, we only consider horizontal constraint. Since horizontal constraint graph is a star, we only need to consider the corresponding net with maximum degree in the horizontal constraint graph. We assign the first track on the bottom layer to this net, then the rest nets can not use this track. The rest nets can route in the second track since each two nets belong to the rest nets have not horizontal constraint and vertical constraint. Then we route them in the most straightforward way: in the corresponding track of the bottom layer we introduce a horizontal wire segment connecting the columns of the two terminals, we switch to the top layer at both ends of this segment and connect to the two terminals.

**Case 1.** Vertical constraint graph of the net have no directed path, that is to say, vertical constraint graph have not edge, each two nets have no vertical constraint.

In this case, we only consider horizontal constraint. Since horizontal constraint graph is a star, we only need to consider the corresponding net with maximum degree in the horizontal constraint graph. We assign the first track on the bottom layer to this net, then the rest nets can not use this track. The rest nets can route in the second track since each two nets belong to the rest nets have not horizontal constraint and vertical constraint. Then we route them in the most straightforward way: in the corresponding track of the bottom layer we introduce a horizontal wire segment connecting the columns of the two terminals, we switch to the top layer at both ends of this segment and connect to the two terminals.

**Case 2.** Vertical constraint graph of the net have a directed path with length 1.

By Theorem 2.1, one of the vertex of this directed path is the center of star. If this vertex is the starting point, denoted by $v_k$, and the terminal vertex of this directed path denoted by $v_{k+1}$. By vertical constraint, the track assigned to the net corresponding to vertex $v_k$ is above to the net corresponding to vertex $v_{k+1}$. In this case, we assign the first track on the bottom layer to the net corresponding to vertex $v_k$ and assign the second track on the bottom layer to the net corresponding to vertex $v_{k+1}$. The rest nets can route in the second track since each nets belong to the rest nets have horizontal constraint with $v_k$ but have not horizontal constraint with $v_{k+1}$. Then using the same method in Case 1, we can assign a track to each of these nets and route these nets in the above straightforward way.

If this vertex is the terminal point, denoted by $v_k$, and the starting vertex of this directed path denoted by $v_{k-1}$. By vertical constraint, the track assigned to the net corresponding to vertex $v_k$ is below to the net corresponding to vertex $v_{k-1}$. In this case, We assign the first track on the bottom layer to the net corresponding to vertex $v_{k-1}$ and assign the second track on the bottom layer to the net corresponding to vertex $v_k$. The rest nets must route in the third track since each nets belong to the rest nets have horizontal constraint with $v_k$. Then using the same method in case 1, we can assign a track to each of these nets and route these nets in the above straightforward way.

**Case 3.** Vertical constraint graph of the net have two directed paths with length 1.

We denote the center of star by $v_k$. By Theorem 2.1, one of the vertex of directed path is the center of star. Since vertical constraint graph of the net have two directed paths with length 1, then $v_k$ is terminal point of one directed path and is starting point of another directed path. When $v_k$ is the terminal point of directed path, then the starting vertex of this directed path denoted by $v_{k-1}$. When vertex $v_k$ is the starting point of directed path, the terminal vertex of this directed path denoted by $v_{k+1}$. By vertical constraint, the track assigned to the net corresponding to vertex $v_k$ is below to the net corresponding to vertex $v_{k-1}$, the track assigned to the net corresponding to vertex $v_k$ is above to the net corresponding to vertex $v_{k+1}$. In this case, we assign the first track on the bottom layer to the net corresponding to vertex $v_{k-1}$, assign the second track on the bottom layer to the net corresponding to vertex $v_k$, and assign the third track on the bottom layer to the net corresponding to vertex $v_{k+1}$. The rest nets can route in the third track since each nets belong to the rest nets have not horizontal constraint with $v_{k+1}$. Then using the same method in Case 1, we can assign a track to each of these nets and route these nets in the above straightforward way.

### 2.1.2. Running time analysis

When we choose the directed path in algorithm, we need constant time to route the nets of this path. When we route the second directed path in step 2, we need 2 compares. Since every compare need constant time, So we give a constant time algorithm to solve this routing problems.

### 2.1.3. The upper bound

In the above algorithm, we give a solution for this particular routing problem. Now, we analyze the upper bound of our algorithm and compare to others algorithms. In [9], the author route this problem with $n$ tracks, he assign a separate track on the bottom layer to each net. In our algorithm, we consider the horizontal constraint of the nets, we can route this problem with no more than 3 tracks. So we give a better polynomial time algorithm to solve the 2-layer Manhattan channel routing problem.

**Acknowledgement.** The research is partially supported by National Science Foundation of China (11401008, 61472003,61272153 and 61401011) and Natural Science Foundation of Anhui Provincial Education Department (KJ2014A064).

### References

[1] Baker, S.B., Bhatt, S.N., Leighton, F.T., *An approximation algorithm for Manhattan routing*, Proc. 15th STOC Symp. (1983), 477–486.

[2] Gallai, T., *His unpublished results were announced in A. Hajnal and J. Surányi Über die Auflösung von Graphen in vollständige Teilgraphen*, Annales Univ. Sci. Budapest. Eötvös Sect. Math., 1 (1958), 115–123.

[3] Gao, S., Kaufmann, M., *Channel routing of multiterminal nets,* J. Assoc. Comput. Mach., 41 (4) (1994), 791–818.

[4] Geng, X.Y., *A polynomial time Algorithm for 2-layer Manhattan Channel*, International Journal of Applied Mathematics and Statistics, 29 (5) (2012), 76–83.

[5] Johnson, D.S., *The NP-completeness column: An ongoing guide*, J. Algorithms, 5 (1984), 147–160.

[6] Marek-Sadowska, M., Kuh, E., *General channel-routing algorithm*, Proc. IEE (GB), 130 (1983), 83–88.

[7] Recski, A., Strzyzewski, F., *Vertex-disjoint channel routing on two layers,* Integer programming and combinatorial optimization (Ravi Kannan and W.R. Pulleyblank, ed.), University of Waterloo Press (1990), 397–405.

[8] Recski A., *Some polynomially solvable subcases of the detailed routing problem in VLSI design*, Discrete Appl. Math., 115 (1-3) (2001), 199–208.

[9] Szeszlér D., *A New Algorithm for 2-layer Manhattan Channel Routing*, Proc. 3rd Hungarian-Japanese Symposium on Discrete Mathematics and Its Applications, (2003), 179–185.

[10] Szkaliczki T., *Optimal routing on narrow channels*, Period. Polytech. Ser. El. Engrg., 38 (1994), 191–196.

[11] Szymanski, T.G., *Dogleg channel routing is NP-complete*, IEEE Trans. Computer-Aided Design of Integrated Circ. Syst., CAD-4 (1985), 31–41.

[12] Yoeli, U., *A robust channel router*, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 10 (2) (1991), 212–219.

[13] Yoshimura, T., Kuh, E.S., *Efficient Algorithms for Channel Routing*, IEEE Trans. on CAD of Integrated Circuits and Systems, 1 (1982), 25–35.

Accepted: 25.04.2015