A General Greedy Channel Routing Algorithm

Tai-Tsung Ho, S. Sitharama Iyengar, *Senior Member, IEEE,* and Si-Qing Zheng, *Member, IEEE*

***Abstract*—This paper presents a new channel routing algorithm which assigns wires track by track in a greedy way. The simple underlying data structures and strategy used in this algorithm can be generalized to obtain a class of heuristic channel routing algorithms. The proposed new algorithm has a backtracking capability to increase the chance of completing the routing with a minimum number of tracks. Since the concepts described in this paper are general, they can be applied to other channel problems such as switchbox routing, three-layer routing, and multilayer routing, or it can be even applied to the overlap model with only a few modifications. It successfully routes both the Burstein's difficult switchbox problem and Deutsch's difficult example with 19 tracks in the Manhattan model without any backtracking. The extensions of this algorithm are presented with examples.**

***Keywords and Phrases*—Channel routing algorithm, data structure, backtracking, Manhattan model.**

I. Introduction

Efficient chip design reflects the capabilities of the semiconductor lithography in the areas of logic, circuit, layout, and processing. The routing portion of the VLSI layout problems is to realize a particular interconnection among different modules in as small area as possible. In general, many routing strategies exist for efficient interconnections among different modules of the VLSI layout problem.

One of the most important forms of routing strategies is called "channel routing." This approach allows us to reduce the extremely difficult VLSI layout problem to a collection of simpler subproblems. Typically a channel router is designed to assign wires that interconnect terminals on two opposite sides of a rectangular region called a "channel." Normally the virtual grid is assumed and the Manhattan model is adopted, i.e., all the horizontal wire segments are routed in one layer and all vertical wire segments in the other, and the connection between wires in these two layers is through some electrical contacts called *vias*. Since the interconnection area usually represents 65–80% of the total area in a typical polycell integrated circuit design, the primary goal of a channel router is to minimize the above mentioned area by minimizing the number of tracks used. The number of vias and the length of wires are also important in evaluating the quality of the routing.

In recent years we have witnessed a tremendous surge in the development of efficient channel routing algorithms for many types of problems. Normally, all the algorithms can be divided into two categories: those in the *overlap* model, which allow some wire segments in different conducting layers to be overlapped, and those in the *nonoverlap* model, which do not allow the overlaps. Traditionally, the nonoverlap model can be further divided into the *Manhattan* model, which restricts all the horizontal wire segments to run in one layer and all the vertical wire segments to run in the other, and the *knock-knee* model which is free of this restriction. Major past research on the channel routing includes [1]–[4], [6], [10] in the Manhattan model [15], [16], [23], [24] in the knock-knee model, and [8], [12]–[14] in the overlap model.

Unlike the above mentioned algorithms, most of which use some parameters derived from experiments, the algorithm presented in this paper approaches the solution in the Manhattan model in a greedy way, systematically based on simple concepts. The advantages of this algorithm are its simplicity and generality. The data structures used are vertical constraint graph, column density, and spans of nets which are the only information needed to assign wires for the nets. In addition, since the algorithm has a general nature, it treats cyclic and noncyclic constraints in the same way, and can be extended easily to the switchbox router, three-layer channel router, multilayer channel router, or even be applied in the overlap routing environment without much modification.

The algorithm proceeds in a track-by-track fashion. First, possible routings for each net in a certain track are calculated. Depending on channel density d_{max} and structure of the vertical constraint graph, nets are then chosen to minimize the channel density d_{max} or length of the directed longest path in the vertical constraint graph, and to maximize the total length of significant horizontal wires in a track through some priority. The performance of the algorithm is shown by experimental results. It successfully routes both Burstein's difficult switchbox problem and Deutsch's difficult example in the Manhattan model without any backtracking. This algorithm can be easily extended to solve other routing problems. For details on the extension of this algorithm see [25]–[28].

The remainder of this paper is organized as follows: some terms and definition of the problem are given in Section II; a framework for a class of heuristic algorithms is given in Section III; the description of our algorithm is given in Section IV, and its performance is given in Section V. In Section VI, we describe several extensions of our algorithm. Finally, the concluding remarks are given in Section VII.

II. Preliminaries

A rectangle *channel* C of length l and width w consists of 1) the set V of *grid points* (x, y) such that x and y are integers, $0 \leq x \leq l + 1$, and $0 \leq y \leq w + 1$; and 2) the set E of *grid edges* connecting points (x, y) and (x', y') whenever these points are at distance 1 from each other. A vertical (horizontal) grid line is a line formed by the grid edges connecting grid points (x, y) with $x = i$ $(y = i)$; furthermore if $i \neq 0$ and $i \neq l + 1$ $(i \neq w + 1)$, then we call such a grid line an ith *column(track)* of the channel C. The horizontal (vertical) lines formed by edges connecting grid points with $y = 0$ and $y = w + 1$ $(x = 0$ and $x = l + 1)$, whose grid points are represented by ordered lists

Manuscript received November 9, 1988; revised October 5, 1989. This paper was recommended by Associate Editor, R. H. J. M. Otten.

The authors are with the Department of Computer Science, Louisiana State University, Baton Rouge, LA 70803.

IEEE Log Number 9040261.

0278-0070/91/0200-0204$01.00 © 1991 IEEE

$T(L)$ and $B(R)$, are called the top (left) and bottom (right) boundary of C, respectively.

Let $N = (N_1, N_2, \cdots N_n)$ be a collection of mutually disjoint subsets of boundary grid points (not including the corner points) of C. Each N_i is called a *net* i, and the grid points in N_i are called the *terminals* of net i. A *routing* of nets for N is a collection $W = (W_1, W_2, \cdots W_n)$ of subgraphs of the channel grid such that all the terminals in N_i are connected by edges in W_i. Each W_i is called the *wire* for N_i. Given a channel grid C and a set $N = (N_1, N_2, \cdots N_n)$ of nets, the *Manhattan mode channel routing problem* (MCRP) is to find $W = (W_1, W_2, \cdots W_n)$ in C such that two distinct wires W_i and W_j cannot share a grid edge, and moreover, they can only share a grid point by crossing each other. Generally, terminals can be located on any boundary side of C. Such a MCRP is called a *switchbox* MCRP. A simpler version of a MCRP restricts terminals to be located on two opposite boundary sides, say, the top and bottom boundary sides of C. Such a MCRP is called a *two-shore* MCRP. The objective of a two-shore MCRP is to find a routing $W = (W_1, W_2, \cdots W_n)$ with a minimum number of tracks. In this paper, we first concentrate on the two-shore MCRP and then extend our algorithm to some related routing problems.

Let $l(N_i)$ ($r(N_i)$) be the x coordinate value of the leftmost (rightmost) terminal of N_i. An obvious lower bound on the number of tracks necessary for a routing solution W is:

$$d_{\max} = \max \{ d_i | \ 1 \le i \le l \}$$

where d_i = the number of nets N_j such that $l(N_j) \le i \le r(N_j)$. Here d_i is called the *local density* at column i, and $d_{\max}$ is called the *channel density*.

In addition to the channel density, another condition that any routing solution must satisfy is that if N_i contains a terminal at the top of column k, and N_j contains a terminal at the bottom of column k and $i \neq j$, then the horizontal wire segment of W_i which connects with the top-side terminal i through a vertical wire segment in column k should be above the horizontal wire segment of W_j which connects with the bottom-side terminal j through a vertical wire segment in column k. This constraint can be characterized by a *vertical constraint graph* $G_{\text{vc}} = (V_{\text{vc}}, E_{\text{vc}})$, where $V_{\text{vc}} = \{ v_i | 1 \le i \le n \}$ such that v_i corresponds to N_i, $e_{ij} = v_i \to v_j \in E_{\text{vc}}$ iff there is a column k such that N_i has a terminal on the top of column k and N_j has a terminal on the bottom of column k. Normally, we say that v_i is an ancestor vertex of v_j if there exists a directed path from v_i to v_j in G_{vc}. In this case, we also say that v_j is a descendant vertex of v_i. In [11], it is shown that there exist MCRP instances with $d_{\max} = 2$ and the longest directed path of length $n - 1$ in a vertical constraint graph that needs at least $\sqrt{2n}$ tracks. In such an instance, almost every wire for each net i must be assigned to more than one track, which is referred to as a *dogleg* for net i. This indicates that in some cases, the channel width is related to the structure of the vertical constraint graph.

III. A Framework for a Class of Algorithms

Since the channel routing problem is *NP*-complete [17]–[19], it is necessary to develop heuristic algorithms for it. In this section we present a framework for a class of heuristic routing algorithms. An algorithm in this class is track oriented. Starting at the topmost track, it processes track after track downwards. The invariant part of this process is that once track t is processed, by treating track t as the top boundary of the channel, we have a new MCRP. Since the combination of the wires above track $t + 1$ and the routing solution for the new MCRP is a solution for the original MCRP, the operations for a single track, which defines the new MCRP, will critically determine the performance of the algorithm. The operations for the current topmost track t can be divided into three phases.

i) Introduce a set W_H of nonoverlapped horizontal wire segments in track t such that after connecting the wire segments in W_H with the related terminals on the top and bottom boundaries of the channel, the number of tracks required for the new MCRP is as small as possible.

ii) Introduce a set W_V of vertical wire segments for each wire segment in W_H when possible, such that $W_H \cup W_V$ and the routing solution of the new MCRP form a routing solution of the original MCRP.

iii) Define a new MCRP for the tracks below track t by identifying a set of grid points in track t and the bottom boundary of the channel as new terminals, and the set of columns that are available.

In the following subsections, we briefly describe the operations needed to process the topmost track of a given channel C.

A. Finding W_V When W_H Is Available

Let $T(k)$ and $B(k)$ represent the grid points on the top and bottom sides of the current channel in column k, respectively. We call a column k an *empty column* if both of $T(k)$ and $B(k)$ are not terminals of any net. When W_H is available, to complete the routing for the topmost track t and construct the remaining MCRP, it is necessary to introduce a set $W_V = W_{V1} \cup W_{V2}$ of vertical wire segments, where W_{V1} is a set of vertical segments from the current top boundary (track $t - 1$) to track t for those columns k with $T(k)$ being terminals, and W_{V2} is a set of vertical segments from track t to the bottom boundary of the channel C. Finding W_{V1} is trivial. We can simply introduce a unit vertical segment that is incident at every terminal on the current top boundary. W_{V2} can be constructed in the following way: for a given column k, if $B(k)$ is a terminal of N_i, $T(k)$ is not a terminal of N_j other than N_i and (k, t) is in $W_{i,t}$, where $W_{i,t} \in W_H$ denotes the horizontal wire segments of W_i in track t, then include a vertical wire segment from track t to the bottom boundary in column k into W_{V2}. If W_{V2} is constructed in this way, then it is impossible that some net can never be completely connected by a wire without overlapping with other wires.

B. Finding W_H for the Current Topmost Track

There are many ways of finding W_H. As mentioned in the previous section, both the vertical constraint graph and channel density can be used to characterize the lower bound for the number of tracks needed for an MCRP instance. Hence, it is reasonable to use the vertical constraint graph and channel density as heuristics in choosing the wire segments in the current topmost track. We propose some useful concepts which may be applied to devise different routing algorithms. In the next section, we present one algorithm based on these concepts.

Generally, the vertical constraint graph corresponding to a problem instance may have cycles. Since among wires connecting a set of nets whose corresponding vertices in G_{vc} are in a cycle, at least one dogleg must be introduced; determining where a dogleg is to be introduced is very important to the total number of tracks used in the routing solution. Of course, we prefer to introduce horizontal wires that form a dogleg among those columns with lower densities. That is, when we use vertical constraint graph G_{vc} to guide the selection of horizontal wires for the current topmost track, we need to select a subset of nets which we do not want to use wires forming doglegs to connect them.

Such a selection can be based on an *acyclic vertical constraint graph* G_{avc} obtained by deleting a minimal number of edges from G_{vc}. Clearly, there is a unique G_{vc} for a given MCRP, and there can be more than one G_{avc}. Among these G_{avc}, we are particularly interested in a G_{avc} constructed as follows:

procedure *AVC-GRAPH* ($G_{avc} = (V_{avc}, E_{avc})$)
  Let colno be the total number of columns for a given MCRP instance.
  Let $c[1], c[2], \cdots, c[\text{colno}]$ be the sequence of columns such that $d_{c[k]} \geq d_{c[k+1]}$, where $1 \leq k \leq \text{colno} - 1$.
  $V_{avc} = V_{vc}$, $E_{avc} = \emptyset$
  for $k = 1$ to colno **do**
    $i = T(k)$, $j = B(k)$
    if ($i \neq 0$ and $j \neq 0$ **and** $i \neq j$ **and** v_i is not a descendant vertex of v_j) **then** $E_{avc} = E_{avc} \cup \{v_i \rightarrow v_j\}$
  endfor
end *AVC-GRAPH*.

The above process for constructing G_{avc} tends to reserve edges in G_{vc} connecting two nets that share columns with higher column densities. When W_H is being constructed, we only consider those constraints which are present in G_{avc}. The missing constraints will be considered in the subsequent steps, processing lower tracks. Consequently, doglegs will likely occur among columns with lower densities. In determining which net is to be connected (or partially connected) in the current topmost track, a parameter *level* for each net is computed where G_{avc} is available. The level of N_i is defined as the length of the longest path from v_i to its descendant vertices in G_{avc}. We identify a subset N_F of N as *free nets*, each of which is a net whose corresponding vertex in G_{avc} does not have incoming edges.

In the Manhattan routing model, it is easy to see that for each wire segment in $W_{i,t}$, both endpoints should locate in columns k such that $T(k)$ is not a terminal of N_j other than N_i. Let

$$E_i = \{k \mid \text{either } T(k) \text{ is a terminal of } N_i \text{ or a nonterminal}\}.$$

Let the horizontal wire segment for N_i in the top-most track with two endpoints located in column k, where $k \in E_i$, be the *feasible wire* for N_i, denoted as f_i. The set of nonoverlapped feasible wires is denoted as H_f. For an $f_i \in H_f$, it is possible that there exists a column k such that $B(k)$ is a terminal of N_i, $T(k)$ is a terminal of N_j other than N_i, and the left and right endpoints of f_i are to the left and right of column k, respectively. In such a case, one cannot introduce a vertical wire segment in column k into W_{V2}. Let $C(f_i)$ denote the set of columns passed by f_i. For a given feasible wire f_i, if there exists no column $k \in C(f_i)$ such that $T(k)$ is a terminal other than a terminal of N_i and $B(k)$ is a terminal of N_i, then we call such a feasible wire f_i a *safe wire* for N_i, denoted as s_i. Let H_s be the set of nonoverlapped safe wires. Clearly, for each $s_i \in H_s$, all the bottom-side terminals of N_i in column k, where $k \in C(s_i)$, can be connected with s_i through a vertical wire segment in column k without causing any congestion problem in the new MCRP.

We may insist on finding an H_s such that after processing the current topmost track, the remaining MCRP can be simplified so that the operations on subsequent tracks result in a routing solution with a number as small as possible of tracks. Some heuristics can be used to choose the H_s.

Let $E_i^* = E_i^1 \cap E_i^2 \cap E_i^3$, where

$$E_i^1 = \{k \mid k \in E_i, B(k) \text{ is a terminal of } N_j \text{ other than } N_i \text{ and level } (N_i) > \text{level } (N_j)\} \cup \{k \mid k \in E_i, B(k) \text{ is a terminal of } N_i \text{ or } B(k) = 0\}$$

$$E_i^2 = \{k \mid k \in E_i, k = l(N_i) \text{ or } k = r(N_i) \text{ or } d_k < d_{\max}\}$$

$$E_i^3 = \{k \mid k \in E_i, l(N_i) \leq k \leq r(N_i)\}.$$

For each safe wire s_i, if both endpoints of s_i are located in the columns in E_i^*, then such a safe wire is called an *optimal wire* for N_i, denoted as o_i. A set of nonoverlapped optimal wires is denoted as H_0. Now consider the relationship between the original MCRP and the new MCRP constructed by introducing an H_0. Since for each wire segment o_i in H_0 both endpoints of o_i are located in the columns in E_i^1, it is easy to verify that the following property holds.

P1) The longest path of G_{avc} of the new MCRP will be less than or equal to that of the original MCRP if no cycle exists in G_{vc}.

Since for each wire segment o_i in H_0 both endpoints of o_i are located in the columns in E_i^2, it is easy to verify that the following property holds.

P2) The value of $\max\{d_k' \mid k \in C(W_{i,t}), W_{i,t} \in H_0\}$ of the new MCRP is less than the $d_{\max}$ of the original MCRP.

Since for each wire segment o_i in H_0 both endpoints of o_i are in E_i^3, it is easy to verify that the following property holds.

P3) The *total span* of the new MCRP is less than that of the original MCRP, where the total span is defined as $\Sigma_{i=1}^{n} (r(N_i) - l(N_i))$.

It is desirable to choose an nonoverlapped H_0^* such that it is a set of safe wires satisfying properties P1), P2), and P3), and satisfying the following additional condition.

H1) When H_0^* is chosen, the channel density $d_{\max}$ of the new MCRP is less than that of the original MCRP.

To enforce ($H1$), in addition to the restrictions of choosing endpoints of each o_i from E_i^*, it is necessary that for each column k such that $d_k = d_{\max}$, there exists an o_i passing through or ending at column k.

After H_0^* is selected, it is possible that there is still some space available for the routing in the topmost track. A degenerated case is that $H_0^* = \emptyset$, that is, there does not exist a set of wire segments that satisfy the conditions for H_0^*. To fully use the current topmost track, one needs to introduce a set of wire segments into the unused gaps. There are two possibilities: either introducing a set H_f^* of feasible wires or introducing a set H_s^* of safe wires. The criterion for choosing an H_s^* or H_f^* is to simplify the remaining MCRP as much as possible. Since feasible wires have less restrictions than safe wires, more flexibilities can be provided by feasible wires when simplifying the problem is concerned. There are many possible ways of choosing H_f^*. Here we discuss one way of introducing H_f^*.

As mentioned in the previous section, both the vertical constraint graph and the channel density characterize the complexity of an MCRP instance. Again, we use both of G_{avc} and d_{max} as heuristics in choosing H_f^*. If we cannot decrease d_{max} of the MCRP through the current topmost track, we choose to change the structure of the vertical constraint graph. From [11], we know that the length of the longest path of G_{avc} is an important factor for determining the lower bound of the channel width for a given MCRP; on the other hand, from [1] we also know that for a given MCRP, if there are no vertical constraints (i.e., every net is a free net), then the optimal routing solution can be easily achieved. Subject to possible existence of H_0^*, we can select H_f^* in the following way: if the length of the longest path in G_{avc} is greater than or equal to d_{max}, then introduce a set H_f^* of wire segments to minimize the length of the longest path of G_{avc}; otherwise a set H_f^* of wire segments is introduced to maximize the number of free nets that can be created. Thus the set W_H of nonoverlapped horizontal wire segments in the current topmost track is $H_0^* \cup H_f^*$.

C. Defining the New MCRP

After the W_H and W_V are introduced, the next step is to identify a set of grid points in track t and on the bottom boundary of the channel as terminals of the new MCRP below track t. For each N_i such that there is no corresponding wire in W_H, all terminals of N_i located in track t-1 are shifted downward to track t as the terminals of N_i in the new MCRP and the terminals of net N_i located at the bottom boundary of channel remain as the terminals of N_i in the new MCRP. If all terminals of N_i are completely connected by the wires in $W_{i,t} \cup W_V$, then N_i will not be considered in the operations for the tracks below. For each $W_{i,t}$ in W_H such that N_i is not completely connected by the wires in $W_{i,t} \cup W_V$, we need to identify a set of new nets with terminals in track t and on the bottom boundary of the channel. In such a case, net N_i may be *split* into at most $|W_{i,t}| + 1$ new nets in the new MCRP, where $|W_{i,t}|$ is the number of horizontal wire segments in $W_{i,t}$. The basic idea for net splitting is as follows: all the terminals of N_i in the gap between each pair of adjacent wire segments in $W_{i,t}$ can be treated as a new net in the new MCRP. The objective of the net splitting is to generate a new MCRP with minimum total span and minimum channel density d_{max}. It is possible that there are more nets in the new MCRP than the nets in the original MCRP; however, if we define the new nets carefully, as the use of concept of net splitting, then the new MCRP problem will have a much simpler structure.

There are several additional issues which must be considered in defining a new MCRP. For example, a wire segment in $W_{i,t}$ can be a feasible wire, a safe wire, or an optimal wire. Wires of different types may need different treatments. Another aspect we must consider is that a column cannot always be available in the process of routing. Once a vertical wire segment in a column k is introduced into W_{V2} to connect a bottom-side terminal of N_i, the section of column k from track t downward is fully occupied. Thus column k must be marked unavailable for the use by wires connecting the terminals of N_j other than N_i. For simplicity, we will not go into details on these issues.

D. Designing a Channel Routing Algorithm

Based on the framework of a class of track oriented greedy channel routing algorithms discussed so far, we can design different heuristic algorithms. Among the three phases, the operations corresponding to the constructions of W_H, W_V, and a new MCRP, for the current topmost track, the way of selecting a set W_H of horizontal wire segments is a determining factor in obtaining a routing solution with a smaller number of tracks. There are not many choices for the selection of W_V that exist. Also, options for defining a new MCRP, when W_H and W_V are present, are limited.

It is important to note that the feasible wires, safe wires, and optimal wires form an interesting hierarchy. Since an optimal wire is also a safe wire and a safe wire is also a feasible wire, it seems that by setting optimal wires with a priority higher than safe wires and safe wires a priority higher than that of feasible wires, better routing solutions can be always possible. Unfortunately, this is not always true. Since a MCRP instance can be extremely complex, global information about an MCRP instance must be examined when operations for a single track are considered. The vertical constraint graph and the channel density capture some of these global information and that is why they can be used for selecting W_H. However, there are other factors to be considered in determining how these pieces of information are used to guide the operations for a single track so that a better solution can be achieved. The notions that may be useful are not limited to those given in this section. Concepts similar to feasible wires, safe wires, and optimal wires can be defined by adding or dropping some conditions. Our proposed framework for the channel routing algorithms is general and powerful enough to allow us to obtain algorithms for the MCRP problem and several related routing problems, as shown in the remaining sections of this paper.

IV. An Algorithm for MCRP

In this section we present an algorithm for the MCRP problem. The design of this algorithm directly follows the framework given in the previous section. Since finding the W_H is the major issue in the framework, the presentation of our algorithm will focus on constructing the W_H for each track. Our algorithm constructs W_H in the following way: if there exists more than one H_0^*, then we choose a H_0^* which minimizes the total span of the new MCRP; if a H_0^* cannot be found, then we choose a nonoverlapped H_f^* as discussed in the previous section; if more than one such H_f^* exists, the one which minimizes the total span of the new MCRP is selected.

Algorithm MCRP-ROUT

1) Construct the G_{avc} using procedure *AVC-GRAPH*, and assign the level of each net.

2) Do the following operations track by track downwards until all nets are routed.

a) Find the longest optimal wire $W_{i,t}$ for each net i.

b) Select the set H_0 of nonoverlapped optimal wires $W_{i,t}$, from the high density columns to lower density columns according to the following priorities: whenever possible a) the wire of free net is selected; otherwise b) the wire of maximal span is selected.

c) If the selected H_0 is not a H_0^* and an alternative H_0 exists, then go to step b) and rechoose the H_0; if the selected H_0 is a H_0^* then go to step d); if none of H_0 is an H_0^* then let $H_0^* = \varnothing$ and go to step d).

d) Select H_f^*.

e) Let $W_H = H_0^* \cup H_f^*$. Construct W_V from W_H. Define the new MCRP. If there exists a net in the new MCRP that is

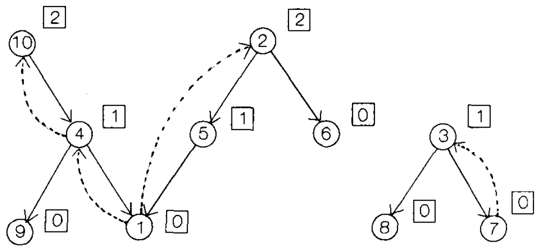


Fig. 1. Acyclic vertical constraint graph.

not completely connected, then update G_{avc} and densities; advance to next track, and go to step a).

3) Minor adjustment to reduce the numbers of bends (vias) and the wirelength of the routing solution.

end *MCRP-ROUT.*

Let us use a simple example to demonstrate the above algorithm *MCRP-ROUT*. For a given MCRP in Fig. 2, in step 1), we apply the procedure *AVC-GRAPH* to construct the G_{avc} as shown in Fig. 1, where numbers inside the circles and rectangles represent the net numbers and related levels, respectively; solid lines represent acyclic vertical constraints, and dotted lines represent the cyclic vertical constraints, which are ignored in G_{avc}.

In step 2.a), all longest optimal wire segments $W_{i,t}$ constructed from the candidate endpoints located in E_i^* are shown in Fig. 2, where dots represent the candidate endpoints in E_i^*.

In step 2.b), we select an H_0^* from all possible optimal wire segments constructed in step 2.a). The details are shown as follows: in Fig. 2, we check the left highest density column 9, where $d_9 = d_{max}$, and find four wires ($W_{1,t}$, $W_{2,t}$, $W_{3,t}$, and $W_{7,t}$) pass this column. From Fig. 1, we know only N_2 and N_3 are in N_F. Since the span of $W_{3,t}$ is greater than that of $W_{2,t}$, $W_{3,t}$ is selected. After wire $W_{3,t}$ is selected, the wires ($W_{2,t}$, $W_{5,t}$, $W_{7,t}$, $W_{8,t}$, and $W_{10,t}$) are ignored, and wires ($W_{1,t}$, $W_{4,t}$, and $W_{9,t}$) are partially chopped off. Then we check the left highest density column 16 ($d_{16} = 6$) which is passed by the remaining candidate wire segments. Since only the wire segment of $W_{4,t}$ passes this column, the wire segment of $W_{4,t}$ from column 16 to column 18 is selected. Since the wire segments of $W_{1,t}$ and $W_{9,t}$ overlap with the selected wire segment of $W_{4,t}$, both wire segments can also be ignored. Hence, H_0, which is also an H_0^* in this case, selected in track t will include the wire segment of $W_{3,t}$ from column 4 to column 15 and the wire segment of $W_{4,t}$ from column 16 to column 18.

Since the selected H_0 is an H_0^* and no feasible wires exist in the unused space, it is easy to see that $H_f^* = \varnothing$ and W_H is an H_0^*.

The consideration for the limited backtracking in step 2.c) is obvious. Since if a column has the highest density and is not used for some optimal wire, it is very likely that the total number of tracks in the final solution is larger than it should be just because an H_0^* is not used. Thus the backtracking in step 2.c) for finding an H_0^* is necessary for obtaining better solutions, although considerable running time may be required to search for an H_0^*. Since the algorithm always routes the nets which pass the high density columns first, in most cases, it is very unlikely that backtracking capability is needed in this step. For example, when our algorithm is applied to both Burstein's difficult switchbox and Deutsch's difficult example, no backtracking occurs. However, when the given problem is more difficult, the backtracking capability is a powerful tool to reach a good solution.

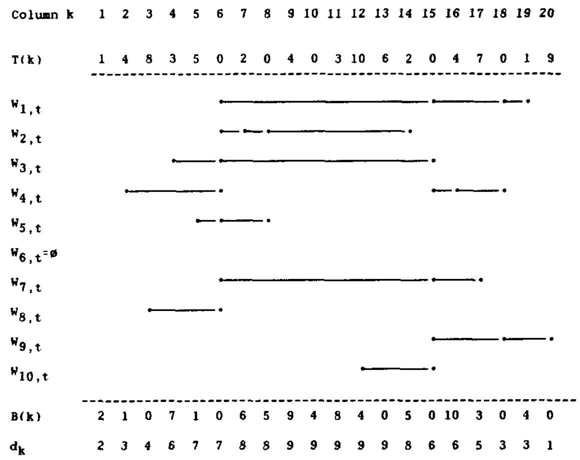


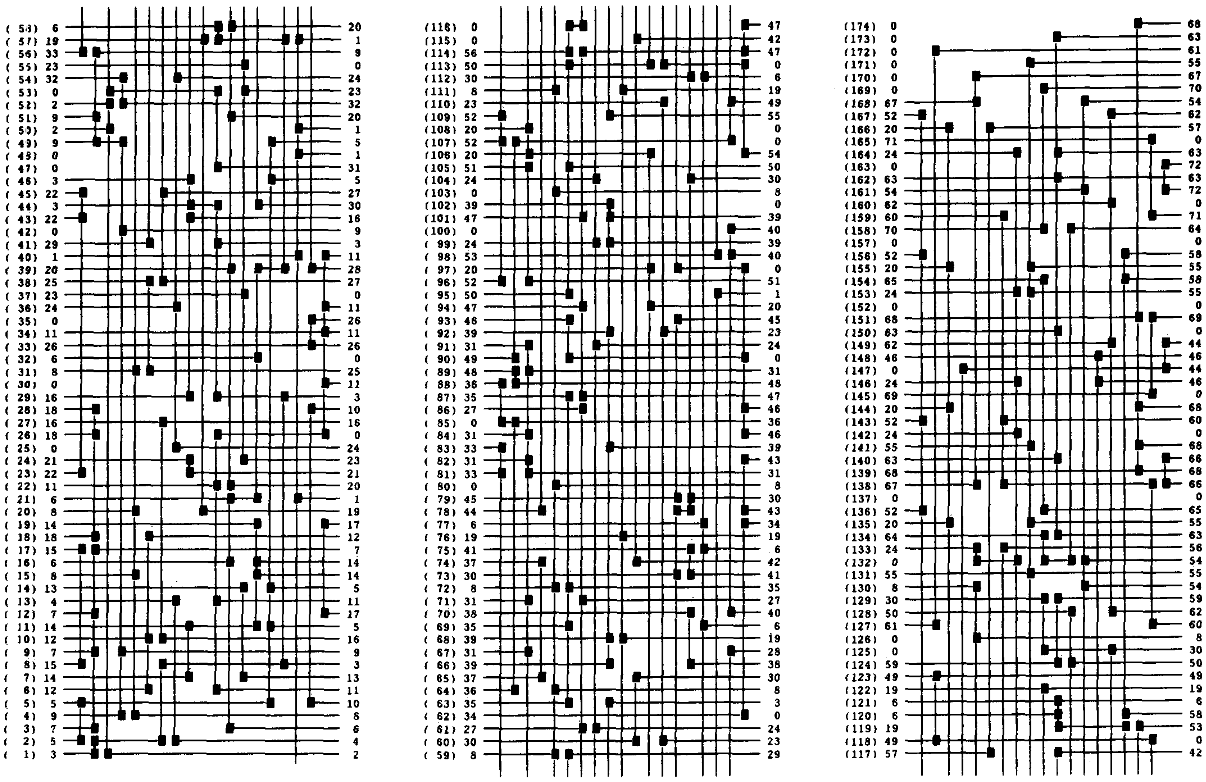
Fig. 2. The set of all the longest safe wires $W_{i,t}$ for the topmost track.

Since the main idea of this algorithm is to fully utilize one track before going to the next one, it is not always desirable to assign wires passing through high density columns as soon as possible. In fact, in some cases delaying assigning wire segments through high density columns may result in better overall routing performance in terms of wirelength and number of vias. That is why, in the final step, adjustment operations are necessary to improve the routing result. For simplicity, we omit the details of these operations.

V. Performance

We applied our algorithm *MCRP-ROUT* to many problem instances. It turns out that our algorithm generates optimal routing solutions in most cases. For example, for the Deutsch's difficult example [3], we achieved a routing solution with 19 tracks, which is equal to the lower bound, as shown in Fig. 3. It is worth mentioning that although minimizing the wirelength and via number is not as important as minimizing the number of tracks emphasized in designing our algorithm, the total wirelength of 5004 and via number of 333 in this solution are better than those achieved by hierarchical wire routing [10], which is the only known algorithm that achieves the lower bound of channel width for this problem instance in the Manhattan model. We also applied our algorithm to the Burstein's difficult channel [7]. As shown in Fig. 4, the solution by our algorithm uses 6 tracks without allowing wires to go beyond the leftmost and rightmost nonempty columns. We believe this is the optimal solution in terms of tracks used in the Manhattan model.

It is easy to see that the running time of the algorithm is dominated by the operations for choosing H_0^* and H_f^* in each track. Let c be the number of columns and n be the number of nets. Clearly, sorting the columns in the order of nondecreasing column density can be done in $O(c \log c)$ time. To obtain H_0^*, we need to find the longest optimal wire for each net. Finding the longest optimal wire $W_{i,t}$ for net N_i takes $O(c)$ by a linear scanning of the columns from left to right. Thus finding the longest optimal wires for all the nets requires $O(nc)$ time. Choosing the H_0^* from these optimal wires can be done in $O(cd_{max})$ time since in each column at most d_{max} wires need to be considered. Therefore, the total time for choosing H_0^* takes $O((\log c + n + d_{max})c)$. For the purpose of deleting a certain vertical constraint, each wire in H_f^* incident at the top boundary of the channel should find a column to dogleg. So choosing H_f^* takes $O(c^2)$. Normally, after H_0^* is chosen, only a small portion of



(a) (b) (c)

Fig. 3. Deutsch's difficult example.

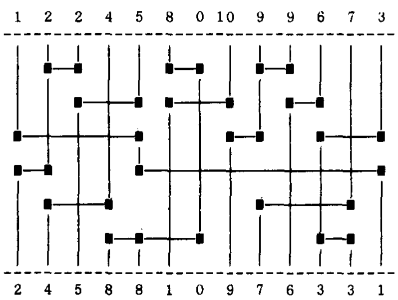


Fig. 4. Burstein's difficult channel.

columns need to be considered for H_f^*. In this case, the constant factor for $O(c^2)$ will be small. The resulted time complexity for the whole procedure is $O((\log c + n + d_{\max})cd_{\max} + c^2k)$, where k is the extra track needed for the final routing beyond the $d_{\max}$. In above analysis, we did not consider backtracking which takes $O(ncd_{\max})$ for each track, when finding H_0^* from high density columns to low density columns always finds the H_0^* satisfying $(H1)$ at the first time if it exists. Since not enough evidence shows that such a mechanism is sufficient to find such H_0^*, the backtracking routine is included.

VI. Extensions of the Algorithm *MCRP-ROUT* to Other Problems

In this section, we briefly discuss several extensions of our algorithm *MCRP-ROUT*. The detailed results will be reported in subsequent papers.

A. Manhattan Switchbox Problem

Our algorithm *MCRP-ROUT* can be extended to the switchbox problem easily. Since that is a switchbox problem, terminals can be located on any of four boundaries of the rectangular channel; the definitions of vertical constraint graph G_{vc}, acyclic vertical constraint graph G_{avc}, feasible wires, safe wires, and optimal wires need to be slightly modified. Then a modified version of algorithm *MCRP-ROUT* for the switchbox problem can be designed. The modified *MCRP-ROUT* connects nets in a track-by-track fashion. By its greedy feature of fully utilizing a track before proceeding to the next track, terminals on the left and right sides of the channel should be connected as soon as possible so that more of the space in the remaining tracks can be available. Consequently, a higher success rate of routing inside the given channel area can be expected.

We have modified algorithm *MCRP-ROUT* to obtain an algorithm for the switchbox problem. We applied the modified version of *MCRP-ROUT* to Burstein's difficult switchbox [10] and successfully found a solution without backtracking. The result is shown in Fig. 5.

B. Channel Routing in Two-layer Overlap Model

Currently, our algorithm is implemented under the Manhattan model, i.e., all the horizontal wire segments run in one layer, and all the vertical wire segments run in the other. To adapt our algorithm to the overlap model, we must allow horizontal wire segments or vertical wire segments to run in different layers with overlaps. We can simply modify the definitions of feasible wires, safe wires, and optimal wires in a way that nonoverlap restrictions are released, with all other features re-

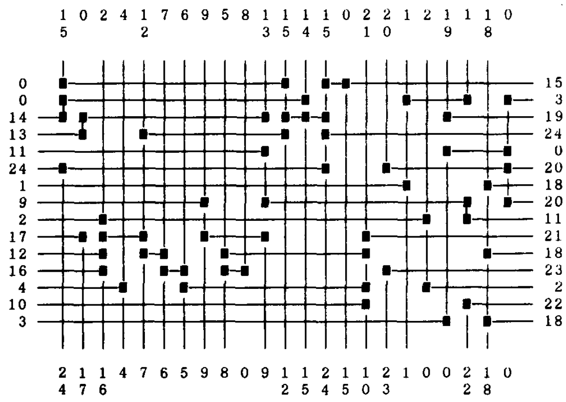


Fig. 5. Burstein's difficult switchbox.

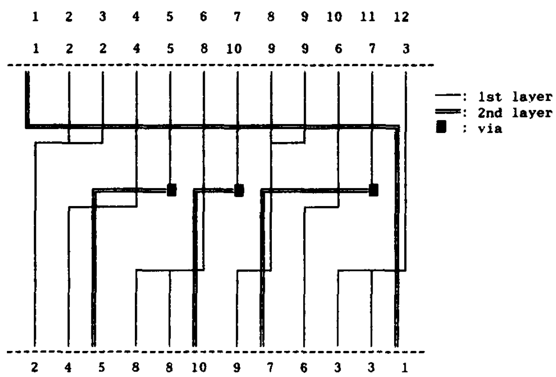


Fig. 6. Burstein's difficult channel in overlap model.

maining unchanged. In the Burstein's difficult channel [7], all previous works either add one or two empty columns in the middle to achieve the routing with five or six tracks [7], [8], or allow wire overlaps to achieve the routing with four tracks [12]. Applying our preliminary modified algorithm to this problem, the routing solution with only three tracks is achieved. The result is shown in Fig. 6.

C. Channel Routing with More Than Two Layers

As a two-layer channel routing problem, there are two versions of the problem of routing with more than two layers. The first version does not allow wire segments in different layers to overlap. Thus two wire segments can only share a grid point only by crossing each other or by forming a knock-knee. Usually, this version of multilayer channel routing is called knock-knee mode channel routing. Typically, knock-knee mode channel routing algorithms consist of two phases. In the first phase, a layout $W = (W_1, W_2, \cdots, W_n)$ is constructed, where W_i is a subgraph of the channel grid that connects all terminals of N_i and no two distinct W_i and W_j share a grid line segment. Such a layout is also called a *planar* layout. In the second phase, each wire segment in a planar layout W is assigned to a layer such that any two wire segments belonging to the wires connecting different nets do not share a grid point in the same layer. It is well known that any planar layout W can be wired in four layers [20] and the problem of determining whether W is three-layer wirable is NP-complete [21]. The necessary and sufficient conditions for constructing a three-layer wiring of W are given in [22]. The only modification needed to adapt our algorithm *MCRP-ROUT* to the knock-knee mode channel routing is to allow wires sharing a grid point by knock-knees. The second version of problem of routing with more than two layers is the one that allows wires in different layers to overlap not only by crossings and knock-knees. Techniques developed for (27) can be generalized to cope with layers more than two. For details of this section, see [26]–[28].

VII. Concluding Remarks

In this paper, as a framework for a class of heuristic routing algorithms, a general approach for the channel routing problem is presented. As an example, we showed how to follow this approach to design a particular routing algorithm *MCRP-ROUT* for the two-shore Manhattan channel routing problem. The performance of this algorithm has been tested on many problem instances and good results have been obtained. Applying our algorithm *MCRP-ROUT* to the benchmark Deutsch's difficult problem and Burstein's difficult problem, we obtained routing solutions of 19 tracks and 6 tracks, respectively. These solutions are either optimal or believed optimal. It should be mentioned that algorithms similar to *MCRP-ROUT* can be easily developed by following the same lines and using different heuristics. For example, for the current topmost track, there are many ways to select W_H. Instead of vertical constraint graph and channel density, other heuristics may be used.

We also showed that track oriented greedy algorithms can be modified to solve other channel routing problems. As examples, we described how to modify algorithm *MCRP-ROUT* to solve the Manhattan switch-box problem and channel routing problems in the overlap and knock-knee models. By our preliminary experiments, the modified algorithms have good performance and show strong potential to out perform the existing algorithms.

Many refinements can be incorporated into our framework and algorithm *MCRP-ROUT*. There is a tradeoff between the quality of the routing solutions and the computing resources required for the more complicated algorithms. Our framework provides the flexibility for running an algorithm to achieve better routing performance.

Acknowledgment

The authors wish to thank all the reviewers for their comments, which improved their presentation of this paper.

References

[1] A. Hashimoto A. and J. Stevens, "Wiring routing by optimizing channel assignment within large apertures," in *Proc. 8th Design Automation Workshop*, 1971, pp. 155–169.
[2] B. W. Kernighan, D. G. Schweikert, and G. Persky, "An optimal channel routing algorithm for polycell layouts of integrated circuits," in *Proc. 10th Design Automation Workshop*, 1973, pp. 50–59.
[3] David N. Deutsch, "A dogleg channel router," in *Proc. 13th Design Automation Conf.*, 1976, pp. 425–433.
[4] R. L. Rivest and C. M. Fiduccia, "A greedy channel router," in *Proc. 19th Design Automation Conf.*, 1982, pp. 418–424.
[5] R. J. Enbody and H. C. Du, "General purpose router," in *Proc. 24th ACM/IEEE Design Automation Conf.*, 1987, pp. 637–640.
[6] Y. Takeshi and E. S. Kuh, "Efficient algorithm for channel routing," *IEEE Trans. Computer-Aided Design*, vol. CAD-1, pp. 298–307, Jan. 1982.
[7] M. Burstein and R. Pelavin, "Hierarchical channel router," in *Proc. 20th Design Automation Conf.*, 1983, pp. 591–597.
[8] J. Reed, A. Sangiovanni-Vincentelli, and M. Santomauro, "A new symbolic channel router: YACR2," *IEEE Trans. Computer-Aided Design*, vol. CAD-4, pp. 208–219, July 1985.

[9] M. R. Edward, N. Jury, and D. Narsing, *Combinatorial Algorithms: Theory and Practice*. Englewood Cliffs, NJ: Prentice Hall, 1977.
[10] M. Burstein and R. Pelavin, "Hierarchical wire routing," *IEEE Trans. Computer-Aided Design*, vol. CAD-2, pp. 223–234, Oct. 1983.
[11] D. J. Brown and R. L. Rivest, "New lower bounds for channel width," in *Proc. CMU Conf. on VLSI Systems and Computations*, 1981, pp. 178–185.
[12] H. Shin and A. Sangiovanni-Vincentelli, "MIGHTY: 'Rip-up and reroute' detailed router," in *Proc. Int. Conf. on CAD*, Nov. 1986, pp. 2–5.
[13] G. T. Hamachi and J. K. Ousterhout, "A switch-box router with obstacle avoidance," in *Proc. 21st Design Automation Conf.*, June 1984, pp. 173–179.
[14] Y. Hsieh and C. Chang, "A modified detour router," in *Proc. Int. Conf. on CAD*, Nov. 1985, pp. 301–303.
[15] R. L. Rivest, A. E. Baratz, and G. Miller, "Probably good channel routing algorithm," in *Proc. CMU Conf. on VLSI Systems and Computations*, 1981, pp. 153–159.
[16] T. Bolognesi and D. J. Brown, "A channel routing algorithm with bounded wire length," unpublished manuscript, Coordinated Science Lab., University of Illinois at Urbana-Champaign, 1982.
[17] T. G. Szymanski, "Dogleg channel routing Is *NP*-complete," *IEEE Trans. Computer-Aided Design*, vol. CAD-4, pp. 31–40, 1985.
[18] D. S. Johnson, "The NP-completeness column: An ongoing guide," *J. Algorithms*, vol. 3, pp. 381–395, 1983.
[19] S. Sahni and A. Bhatt, "The complexity of design automation problems," in *Proc. 17th Design Automation Conf.*, 1980, pp. 402–411.
[20] M. L. Brady and D. J. Brown, "VLSI routing: Four layers suffice," *Advances Comput. Res.*, vol. 2, 1984.
[21] W. Lipski, Jr., "An NP-complete problem related to three-layer channel routing," *Advances Comput. Res.*, vol. 2, 1984.
[22] W. Lipski, Jr. and F. P. Preparata, "An unified approach to layout wirability," *Mathematic. Syst.* Theory, vol. 19, 1987.
[23] K. Mehlhorn and F. P. Preparata, "Routing through a rectangle," *J. ACM.*, vol. 33, no. 1, 1986.
[24] F. P. Preparata and W. Lipski, Jr., "Optimal three-layer channel routing," *IEEE Trans. Comput.*, vol. C-33, p. 5, 1984.
[25] T. T. Ho, S. S. Iyengar, and S. Q. Zheng, "A new channel routing algorithm," Tech. Rep. #88-055, Louisiana State Univ., 1988.
[26] —— "Density-based general greedy channel routing," Tech. Rep. #89-014, Louisiana State Univ., 1989.
[27] T. T. Ho and S. S. Iyengar, "A density-based three-layer router," Tech. Rep. #89-015, Louisiana State Univ., 1989.
[28] T. T. Ho, "A density-based general greedy channel routing algorithm in VLSI design automation," Ph.D. dissertation, Louisiana State Univ., Aug. 1989.
[29] Y. L. Lin, Y. C. Hsu, and F. S. Tsai, "SILK: A simulated evolution router," *IEEE Trans. Computer-Aided Design*, vol. 8, pp. 1108–1114, Oct. 1988.

Tai-Tsung Ho (M'89) received the B.S. degree in industrial engineering from National Tsing Hua University, Taiwan, Republic of China, in 1980, and the M.S. degree in industrial engineering and the Ph.D. degree in computer science from Louisiana State University in 1985 and 1989, respectively.

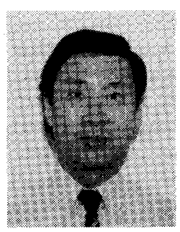
From 1980 to 1982, he served as a Lieutenant in the Army of the Republic of China. He is currently an Assistant Professor in the Department of Mathematics, Computer Science, and Statistics, McNeese State University. His major research interests are in VLSI design automation, parallel and distributed processing, and management information systems.

Dr. Ho is a member of ACM.

S. Sitharama Iyengar (M'87–SM'89) received the Ph.D. degree from Mississippi State University in 1974.

He is currently a Professor of computer science at Louisiana State University, and has been the Director of the Robotics Research Laboratory since its inception in 1986. He is currently working on research projects supported by the Office of Naval Research, National Aeronautics and Space Administration, National Science Foundation/Laser Programme, Jet Propulsion Laboratory-California Institute of Technology, Department of Navy-NORDA, Department of Energy through Oak Ridge National Laboratory, LEQFS-Board of Regents, and APPLE Company Inc. He has coauthored two volumes of *A Tutorial on Autonomous Intelligent Systems* IEEE Computer Society Press, to be published, and has edited two books and over 140 publications including 70 archival journal papers in areas of high performance parallel and distributed algorithms and data structure for image processing and pattern recognition, autonomous navigation, distributed sensor networks etc. He is a technical consultant to Jet Propulsion Laboratory-California Institute of Technology, Oak Ridge National Laboratory, and the Indian Institute of Science.

Dr. Iyengar is an ACM National Lecturer. He is a series editor of Neuro Computing of Complex Systems and area editor of *Journal of Computer Science and Information*. He has been coguest editor of IEEE TRANSACTIONS ON SOFTWARE ENGINEERING (1988), *IEEE Computer Magazine* (1989), IEEE TRANSACTIONS ON SYSTEM, MAN, and CYBERNETICS, IEEE TRANSACTIONS ON DATA AND KNOWLEDGE ENGINEERING and *Journal of Computers and Electrical Engineering*. He was awarded the Phi Delta Kappa Research award of distinction at LSU (1989). He won the Best Teacher of the year award in 1978.



Si-Qing Zheng (M'90) received the B.S. degree from Jilin University, China, in 1973, the M.S. degree in computer science from the University of Texas at Dallas in 1982, and the Ph.D. degree in computer science from the University of California, Santa Barbara, in 1987.

He is currently an Assistant Professor with the Department of Computer Science, Louisiana State University. His research interests include VLSI, computational geometry, combinatorial algorithms, and parallel computing.