

An Efficient and Fast Algorithm for Routing Over the Cells

KUO-EN CHANG and SEI-WANG CHEN

Department of Information and Computer Education, National Taiwan Normal University, Taipei, Taiwan, R.O.C.

(Received 7 August 1992; In final form 20 August 1994)

A linear time algorithm for routing over the cells is presented. The algorithm tries to reduce maximum channel density by routing some connections over the cells. The algorithm first defines a new scheme for channel representation and formulates the problem based on an intersection graph derived from the new scheme. Then, a feasible independent set of the intersection graph is found for routing some subnets over the cells. The algorithm is implemented and evaluated with several well known benchmarks. In comparison with previous research, our results are satisfactory, and the algorithm takes substantially less CPU time than those of previous works. For Deutsch's difficult example, the previous algorithms take about 29.25 seconds on an average but our new algorithm needs only 5.6 seconds.

Keywords: Channel routing, routing over the cells, independent set, and intersection graph

1. INTRODUCTION

In VLSI design automation, channel routing is one of the most important steps in circuit design. A channel router receives two lists of terminals specified at both the top and the bottom sides of the channel, and finds all the interconnections of terminals in order to minimize the channel width, the length of interconnections, and the number of vias. Conventional channel router assumes that the areas used for interconnections are restricted to the two routing layers *within* a channel. This type of channel routing is very well-known as evidenced by numerous studies in the past [1-4].

Due to the advent of semiconductor fabrication technology, more layers (e.g. 2 or 3 metal layers) are

available for interconnections in integrated circuit design. As more metal layers are made for routing in standard cell design or III logic array technologies, routing with extra layer over the cells becomes both practical and important. To further reduce routing area or channel width, a channel router is allowed to use extra areas over the cells for interconnections. These channel routers are called the over-the-cell channel router.

The basic idea behind over-the-cell routing is that two routing areas, one metal layer outside the channel and two layers (one polysilicon layer and one metal layer) inside the channel, are used to connect all nets. The router finds some subnets to be planarly connected over the cells so that the maximum channel density is minimized after the remaining subnets are

routed inside the channel [7]. Figure 1 is an example of the over-the-cell routing.

The over-the-cell channel routing has been the subject of several studies [5–7]. The router proposed in [5] completed the detailed routes in IIL LSI. The procedure consists of three distinct operations. The first is to determine which nets or net segments are to be routed over the cells and which are to be connected within the channel. Then, the net segments selected to be routed over the cells are connected physically. Finally, the interconnections routed inside the channel are completed by the existing dogleg channel router [1].

Another algorithm, called the permeation router, was presented in [6]. This router was developed to give an approximate solution to the routing problem. The basic idea is to use the areas over the cells as optional regions for routing. The available tracks, therefore, just permeate from the channel between two cell rows into that on the top/bottom cell row. The algorithm consists of two phases. First, a vertex coloring technique is used to divide the given track set into three subsets corresponding to the tracks to be routed in the area on the top cell row, bottom cell row, or between cell rows. The second phase is to route the tracks on the top/bottom cell row using just one metal layer, and to route the channel between two rows using two wiring layers.

The over-the-cell routing proposed in [7] was composed of three steps: (1) routing over the cells, (2) choosing net segments within the channel, and (3)

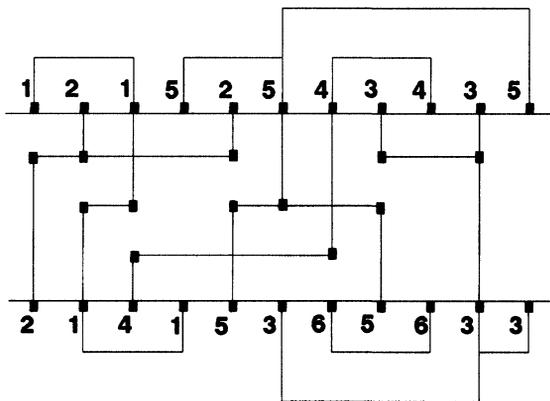


FIGURE 1 The example of over-the-cell routing.

routing within the channel. In the first step, some terminals at the same side of channel are selected to be connected using the over-the-cell routing area on that side. The goal is to find as many subnets routed over the cells as possible such that fewer subnets are left within the channel. At completion of the first step, a net is partially routed over the cells. During the second step, a proper subset of terminals which have been connected over the cells is selected as a net segment, and this net segment will be routed within the channel at the final step. Finally, the physical connections of subnets are routed within the channel. Obviously, this final step can be accomplished using a conventional channel router.

In this paper, we only consider the problem involved in the first step of [7]. The previous approaches were based on the idea that fewer subnets routed within the channel yields fewer tracks used in the channel. Hence, the goal of these approaches is to find as many subnets as possible for routing over the cells, and the selected subnets can be collected as a maximum planar subset since there is only one routing layer over the cells at each side of channel. Unfortunately, this idea does not meet the characteristics of channel routing. It is well-known that the number of tracks needed in the channel is proportional to the channel density of a routing problem [3]. The selection of subnets, which contribute to the reduction of maximum channel density after these subnets have been routed over the cells, is more critical than the selection of subnets which can be collected as a *maximum* planar subset. The approach that blindly routes as many subnets as possible over the cells only contributes slightly to the reduction on the channel width. The critical subnets routed over the cells are those whose removal from the channel will contribute to the reduction of maximum channel density. Therefore, our consideration is the critical planar subset but not the maximum planar subset.

Lin, et. al. [8, 9] have studied the channel density reduction by routing over the cells. They transformed this problem into a constrained covering problem and formulated it as an integer linear programming problem. In comparison with previous research, their approach reduces more channel densities while using

fewer tracks over the cells. However, the drawback of the integer linear programming formulation is its computational inefficiency; the CPU time complexity is not proportional to the channel size. When the problem size becomes large, the CPU time grows very rapidly. A heuristic algorithm is thus needed to replace the integer linear programming for CPU time consideration. In this paper, we present a heuristic algorithm for routing the nets over the cells. The experiment showed that our algorithm is efficient in both reductions of the channel width and the CPU time.

In the following section, we present the problem of routing over the cells as a new scheme of channel representation. The graph model and problem formulation are presented in Section 3. On the basis of the graph model, a heuristic algorithm to determine the critical subnets for routing over the cells is addressed in Section 4. Finally, results and conclusions are presented.

2. CHANNEL REPRESENTATION

A channel representation is defined by two number lists of terminals specified at the top side and the bottom side of a channel, respectively. A net is an interconnection of terminals with the same number in the channel. A segment is an interval between two adjacent terminals of a net in the channel. The segments can be classified into three types: top, bottom and dummy segments. Figure 2 shows the definitions of the three types of segments. Top segments are the segments defined on the top side of a channel, and bottom segments are those on the bottom side. Dummy segment is an interval of two terminals which are located on different sides of a channel. For a net j with k terminals on the top side of channel,

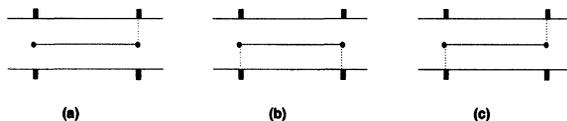


FIGURE 2 The definitions of (a) top segment, (b) bottom segment, and (c) dummy segment.

there are $k - 1$ top segments denoted by $h_{j1}^t, h_{j2}^t, \dots, h_{j(k-1)}^t$, respectively. The notation for the bottom segments is similar to that of top segments. The i -th bottom segment of net j is denoted by h_{ji}^b . Every dummy segment of net j is designated as h_j^d . Figure 3 specifies all instances of segments, in which there are two top segments h_{51}^t and h_{52}^t , one bottom segment h_{51}^b , and one dummy segment h_4^d for the net 4.

For each column i in a channel, a column-set $z(i)$ is constructed. Column-set $z(i)$ is a set of net-segments defined at column i . A net-segment defined at column i is a set of segments which belong to the same net and cross over column i in the channel. For each column i , there are some net-segments defined. The example shown in Figure 4 is the column-set representation of Figure 3. For example, four segments $h_{21}^t, h_{51}^t, h_{51}^b$, and h_4^d cross over column 5 in Figure 3, so there are three net-segments (h_{21}^t) , (h_{51}^t, h_{51}^b) , and (h_4^d) to be defined. Therefore, $z(5) = \{(h_{21}^t), (h_{51}^t, h_{51}^b), (h_4^d)\}$ as shown in Figure 4. For channel routing, all segments in each net-segment have to be connected together and be realized as a connection within the channel. The number of net-segments decides the number of tracks used in the channel. Hence, local density $d(i)$ is defined as the number of net-segments in the column-set $z(i)$. Figure 4 also shows the local densities of all columns. The maximum channel density d_{max} is then defined as the maximal $d(i)$ for all columns i . The d_{max} of Figure 4 is 4.

Furthermore, the net-segments are classified into three types. A simple net-segment contains only one top or bottom segment. The compound net-segment contains at least two non-dummy segments. The dummy net-segment is a collection of segments in which at least one dummy segment is included. In the

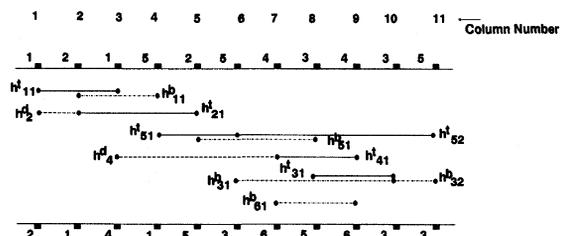


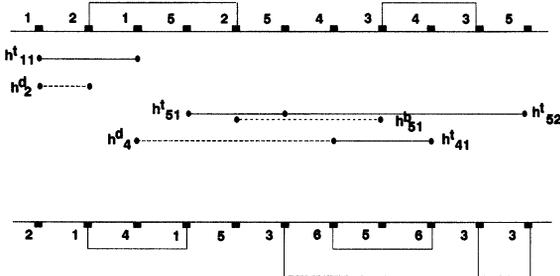
FIGURE 3 Definitions of segments shown in Figure 1.

Column i	Column-sets $z(i)$	Local density $d(i)$
1	$(h_{11}^t) (h_2^d)$	2
2	$(h_{11}^t, h_{11}^b) (h_2^d, h_{21}^t)$	2
3	$(h_{11}^t, h_{11}^b) (h_{21}^t) (h_4^d)$	3
4	$(h_{11}^t) (h_{21}^t) (h_4^d) (h_{51}^d)$	4
5	$(h_{21}^t) (h_4^d) (h_{51}^t, h_{51}^b)$	3
6	$(h_4^d) (h_{51}^t, h_{52}^t, h_{51}^b) (h_{31}^b)$	3
7	$(h_{41}^t, h_4^d) (h_{52}^t, h_{51}^b) (h_{31}^b) (h_{61}^b)$	4
8	$(h_{41}^t) (h_{52}^t, h_{51}^b) (h_{31}^t, h_{31}^b) (h_{61}^b)$	4
9	$(h_{41}^t) (h_{52}^t) (h_{31}^t, h_{31}^b) (h_{61}^b)$	4
10	$(h_{52}^t) (h_{31}^t, h_{31}^b, h_{32}^b)$	2
11	$(h_{52}^t) (h_{32}^b)$	2

FIGURE 4 Channel representation of column-sets.

column-set $z(7)$ of Figure 4, there exists a compound net-segment (h_{52}^t, h_{51}^b) , two simple net-segments (h_{31}^b) and (h_{61}^b) , and a dummy net-segment (h_{41}^t, h_4^d) . The net-segment (h_{41}^t, h_4^d) is dummy because of the inclusion of dummy segment h_4^d .

Our problem of routing over the cells is to find some critical top or bottom segments such that their removal from the channel to the area over the cells can significantly reduce the maximum channel density. For the example of Figure 3, if these segments $h_{31}^t, h_{31}^b, h_{32}^b, h_{61}^b, h_{21}^t$, and h_{11}^b are selected to be routed over the cells or to be removed from the channel, the maximum channel density will be reduced to 2 as shown in Figure 5. Compared with the result of $d_{max} = 3$ presented in Cong and Liu [7], the result in Figure 5 is better. Since only one layer over the cell is permitted for routing, the selected segments have to be routed for planar connections. In short, the aim of our algorithm is to find some critical segments that can be routed by planar connections in order to minimize d_{max} in the final result.

FIGURE 5 The layout with $d_{max} = 2$ after routing over-the-cells.

3. GRAPH MODEL AND PROBLEM FORMULATION

For the selection of critical segments, a graph called segment intersection graph denoted by $SIG = (V_T \cup V_B, E)$ is used to define the relation among the top segments and the bottom segments. The segment intersection graph is defined as follows. Vertices in V_T (or V_B) correspond to the top (or the bottom) segments in the channel. Two vertices v_i and v_j in V_T (or V_B) are connected by an edge in E when the two corresponding top (or bottom) segments do not belong to the same net and are intersected with each other. Two top (or bottom) segments are said to be intersecting if their intervals are partially rather than completely overlapped on the top (or the bottom) side. It is clear that the associated connections of two top (or bottom) segments, which do not belong to the same net, cannot be routed planarly over the cells if these two segments are intersecting. For the example of Figure 3, two top segments h_{41}^t and h_{31}^t are intersecting, but the segments h_{52}^t and h_{41}^t are not intersecting. Figure 6 shows the graphical representation of the relation among the segments in Figure 3.

LEMMA 1: *The vertices in any independent set in SIG correspond to the segments that can be routed planarly over the cells.*

Proof: Because only one layer is used for routing over the cells, all selected segments should be routed planarly over the cells. It is noted that two segments can be routed planarly over the cells if their intervals

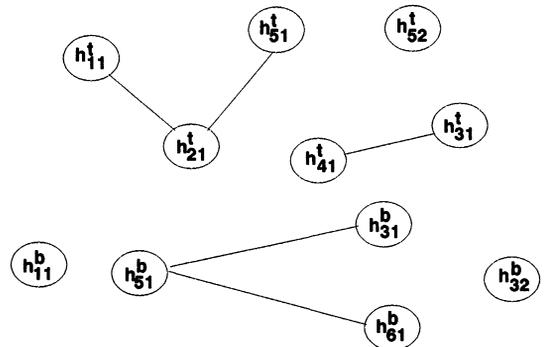


FIGURE 6 Segment-intersection graph of Figure 3.

do not intersect in the channel. Based on the definition of a *SIG*, two vertices not connected in *SIG* means that their intervals are not intersecting. Therefore, all associated segments of the vertices in any independent set can be routed planarly over the cells. Q.E.D.

According to the graph model and Lemma 1, our problem can now be formulated as follow.

PROBLEM ROC (or Routing Over the Cells): Given a graph $SIG = (V_T \cup V_B, E)$, the problem is to find two subsets $R_T \subseteq V_T$ and $R_B \subseteq V_B$ where R_T and R_B are independent sets in *SIG* and the removal of corresponding segments in R_T and R_B over the cells can reduce the maximum channel density significantly.

The algorithm solving the PROBLEM ROC is heuristically executed in the following steps:

1. Segments considered to be critical for routing over the cells in order to reduce the d_{max} are the segments crossing over the columns j with $d(j) = d_{max}$ for all j . These critical segments are collected as a set S_{max} . The formal definition of S_{max} is an union of all segments in the column-sets $z(j)$ with $d(j) = d_{max}$ for all j . As shown in Figure 3, $d_{max} = 4$ and $S_{max} = z(4) \cup z(7) \cup z(8) \cup z(9) = \{h_{11}^b, h_{21}^t, h_{51}^t, h_{52}^t, h_{51}^b, h_{41}^t, h_{61}^b, h_{31}^t, h_{31}^b\}$. The first step is to obtain the set S_{max} .
2. Construct a subgraph H from the graph *SIG* with vertices in S_{max} . Figure 7 shows the subgraph H of Figure 6
3. According to Lemma 1, an independent set in the subgraph H is obtained and the corresponding seg-

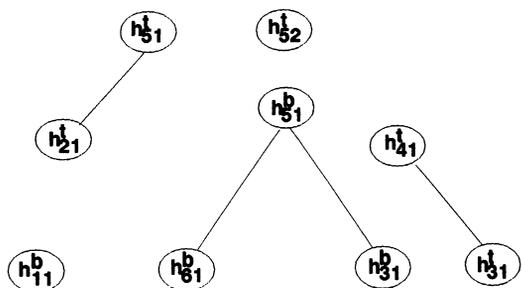


FIGURE 7 The subgraph H with vertices in S_{max} of *SIG* shown in Figure 6.

ments are removed from the channel for reducing the channel densities. The removed segments are those that are going to be routed over the cells.

4. After the vertices of corresponding segments obtained from Step 3 have been removed, the graph *SIG* is simplified and d_{max} is reduced. The set S_{max} has to be redefined based on the reducing d_{max} , and the subgraph H is reconstructed for the next selection of critical segments.
5. Repeat Steps 3 and 4 until H is null or no feasible segment remains.

In selecting the feasible segments in Step 3, the issues raised in Lemma 2 and Lemma 3 below must be considered.

LEMMA 2: *It is not helpful to reduce the channel densities after only one segment in any compound net-segment was removed. In order to reduce the channel densities, it is necessary to remove all segments in the compound net-segment over the cells.*

Proof: According to the definition of compound net-segment, all segments in the compound net-segment belong to a net. Therefore, these segments have to be connected together. After removing only one segment over the cells, the remaining segments also need to be routed inside the channel, and the connection will use a track. This removal is of no help for reducing the channel densities. Q.E.D.

For example, as shown in Figure 3, the removal of segment, that is only h_{52}^t instead of all segments in the net-segments (h_{52}^t, h_{51}^b) at column 7, cannot reduce the local density $d(7)$.

According to Lemma 2, the consideration is the removal of the entire net-segments, neither the top segments nor the bottom segments. Therefore, the selection problem in Step 3 is actually the consideration of the critical net-segments.

LEMMA 3: *Dummy net-segments are not considered for reducing the channel densities.*

Proof: A dummy net-segment consists of at least a dummy segment. The dummy segment is defined as an interval that has its two terminals located on two different sides of a channel. Hence, the dummy seg-

ment cannot be routed over the cells on either side of the channel. Lemma 2 states that a net-segment may be critical when all its segments can be removed or routed over the cells. The dummy net-segment is not critical for reducing channel densities since the dummy segments in it cannot be routed over the cells. Q.E.D.

As the result of Lemma 3, in each column-set, the net-segments to be potentially critical are the simple net-segments and the compound net-segments.

A Net-segment Intersection Graph (or abbreviated as *NIG*) defined below is used to determine which net-segments are critical. Each vertex in *NIG* corresponds to a simple or compound net-segment, and two vertices are connected in *NIG* when any two segments, which belong to the two corresponding net-segments respectively, are intersecting. The *NIG* can be constructed from the subgraph *H* by merging two or more vertices in *H* when they are in the same net-segment. Figure 8 is the *NIG* constructed from the subgraph *H* of Figure 7.

By the *NIG* definition, the removal of all segments in a net-segment in *NIG* may lead to the reduction of local channel densities. For the example of Figure 3, the simple net-segment (h^t_{41}) is useful for reducing the local densities $d(8)$ and $d(9)$ when it is routed over the cells, but it cannot reduce the local density $d(7)$ because it is a member of the dummy net-segment (h^t_{41}, h^d_4) at column 7. On the other hand, the net segment (h^t_{31}, h^b_{31}), (h^t_{52}, h^b_{51}), or (h^b_{61}) is feasible to reduce the local densities $d(7)$, $d(8)$ and $d(9)$. Based on the above discussion, the removal of segments in the net-segment (h^t_{31}, h^b_{31}), (h^t_{52}, h^b_{51}),

or (h^b_{61}) is better than that of the net-segment (h^t_{41}). As a conclusion, the suitable selection of net-segments in *NIG* for significantly reducing the maximum channel density is required for our algorithm.

For the suitable selection of net-segments, two parameters are weighted at the vertices in a *NIG*. For a vertex k of a *NIG*, these parameters are defined as:

$$\alpha_k = \text{the number of edges connected to vertex } k \text{ in } NIG. \quad (1)$$

$$\beta_k = \text{the number of column-sets } j \text{ with } d(j) = d_{max} \text{ and net-segment } k \text{ is in column-set } j. \quad (2)$$

The weights α_k and β_k can be easily computed by the following algorithm:

- **ALGORITHM WEIGHT** (k)
 1. $\alpha_k = \text{deg}(NIG, k)$, where $\text{deg}(NIG, k)$ is the degree of vertex k in the graph *NIG*.
 2. $\beta_k = 0$
 3. **FOR** all columns j in the channel **DO**
 4. **IF** $d(j) = d_{max}$ and $k \in z(j)$ **THEN**
 $\beta_k = \beta_k + 1$
 - END ALGORITHM**

The complexity of the algorithm *WEIGHT* (k) is linear with respect to the number of columns in the channel. Figure 9(a) shows the weighted graph of Figure 8.

4. THE ALGORITHM

The critical net-segments in graph *NIG* can be determined heuristically by two steps:

- (1) Select one vertex based on the weight (α, β).
- (2) The graph *NIG* is reconstructed after the segments in the net-segment associated with the selected vertex are routed over the cells and removed from the *SIG*.

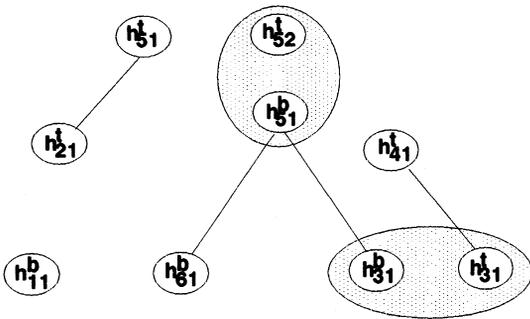
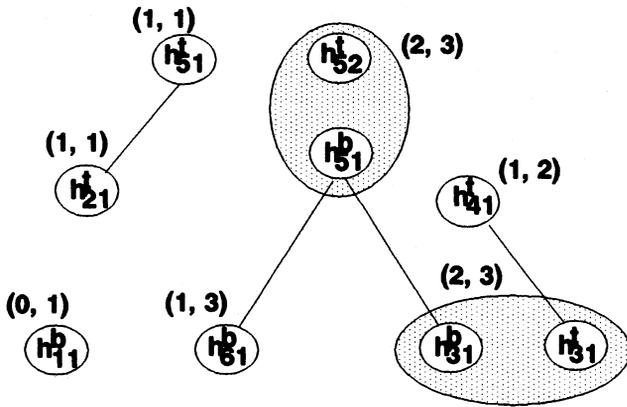


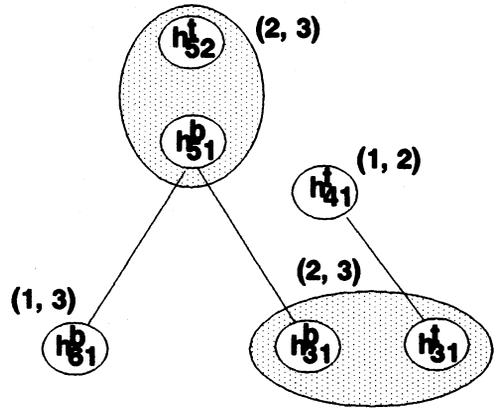
FIGURE 8 The graph *NIG* constructed from the subgraph *H* shown in Figure 7.



$$S_{max} = \{h_{11}^b, h_{21}^t, h_{51}^t, h_{52}^t, h_{51}^b, h_{61}^b, h_{41}^t, h_{31}^b, h_{31}^t\}$$

$$\Psi = \phi$$

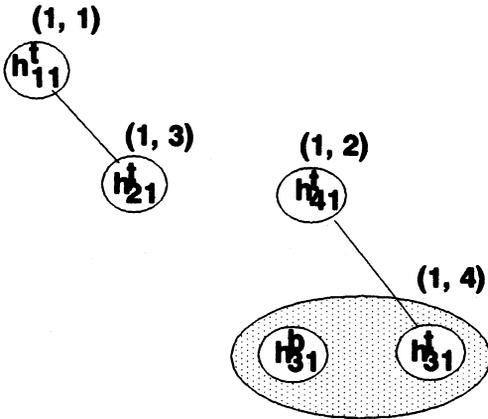
(a) The weighted NIG ($d_{max} = 4$)



$$S_{max} = \{h_{52}^t, h_{51}^b, h_{61}^b, h_{41}^t, h_{31}^b, h_{31}^t\}$$

$$\Psi = \phi$$

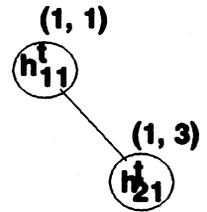
(b) The reduced NIG after h_{11}^b is removed



$$S_{max} = \{h_{11}^t, h_{21}^t, h_{41}^t, h_{31}^b, h_{31}^t\}$$

$$\Psi = \{h_{52}^t, h_{51}^b\}$$

(c) The graph NIG ($d_{max} = 3$) after h_{61}^b is removed



$$S_{max} = \{h_{11}^t, h_{21}^t\}$$

$$\Psi = \{h_{52}^t, h_{51}^b, h_{41}^t\}$$

(d) The reduced NIG after (h_{31}^b, h_{31}^t) is removed

$$S_{max} = \phi$$

$$\Psi = \{h_{52}^t, h_{51}^b, h_{41}^t, h_{11}^t\}$$

(e) The NIG ($d_{max} = 2$) is null graph after h_{21}^t is removed and it cannot further to remove net-segments.

FIGURE 9 A step-by-step exmaple for the removal of net-segments, and the final outputs are $h_{11}^b, h_{61}^b, (h_{31}^b, h_{31}^t)$, and h_{21}^t .

The algorithm starts with a *NIG* with vertices in the initial S_{max} and executes the above two steps iteratively until no feasible vertex exists.

During step (1), vertices with minimal α are considered first, and then a vertex with maximal β is selected from them. If $W_j = \alpha_j + 1/(\beta_j)$ for all vertices j , the vertex with minimal W_j is selected. For the instance of Figure 9(a), the vertex h_{II}^b is selected first.

After a vertex is selected, the segments in the corresponding net-segment will be routed over the cells and removed from the *SIG*. The *NIG* will be reconstructed for the next selection. Four steps are required to reconstruct a new *NIG* after a selected vertex is removed.

- (1) Mark all segments in the vertices that are connected to the selected vertex in *NIG* to be inactive in the successive selection.
- (2) Recalculate a new set S_{max} after the segments in the corresponding net-segment are routed over the cells or removed from the channel.
- (3) Based on the new set S_{max} , reconstruct the subgraph H in which the vertices are all active.
- (4) From the new subgraph H , a new *NIG* is built.

Because the selected vertices should not be interconnected in *NIG* for the planar routing over the cells, all segments in the vertices connected to the selected vertex will not be considered in the next selection. These inactive segments will be marked and neglected from the recalculation of new S_{max} . Let Ψ be a set of the inactive segments. The members in Ψ will grow during the process of selecting the critical segments. Figure 9 shows a step-by-step example for solving the problem shown in Figure 3. The algorithm halts when no feasible net-segment exists as shown in Figure 9(e).

In summary, the procedure of routing over the cells is presented as follows.

• **MAIN PROCEDURE**

1. Input two lists of terminals on the top side and the bottom side of the channel.
2. Define column-sets and local densities.

3. Construct the graph *SIG*, subgraph H , and the corresponding graph *NIG*.
 4. **REPEAT**
 5. Select a vertex j with minimum W_j for all vertices in *NIG*;
 6. Mark all segments in the vertices that are adjacent to j in *NIG* to be inactive.
 7. **FOR** all segments u in net-segment j **DO**
 8. **IF** u corresponds to the top-segment **THEN** $R_T = R_T \cup u$;
 9. **ELSE** $R_B = R_B \cup u$;
 10. Recalculate the local densities and d_{max} after the net-segment j is removed.
 11. Reconstruct a set of S_{max} and the weighted graph *NIG*
 12. **UNTIL** (*NIG* is null or no feasible vertex remains)
 13. **RETURN** (R_T, R_B)
- END PROCEDURE

The final results are stored in R_T and R_B . All segments corresponding to the vertices in R_T and R_B are feasible for reducing the d_{max} after they have been routed over the cells at the top side and the bottom side, respectively.

The time complexity of algorithm is easily analyzed as follows. Each time of scanning all columns in a channel from left to right, the local densities, S_{max} , *SIG*, and *NIG* are constructed. It is easy to show that the complexity of Steps 1–3 is $O(M)$, where M is the number of columns in the channel. A *NIG* is reconstructed once for each iteration in REPEAT-UNTIL loop. During each iteration of REPEAT-UNTIL loop, the algorithm takes $O(M + |S_{max}|)$ since the updates in both the local densities and the S_{max} in Steps 10 and 11 are done in $O(M)$. Assume that K is the number of net-segments selected for routing over the cells (or the number of iterations in REPEAT-UNTIL loop), the worst-case complexity of the algorithm is $O(KM + \sum_K S_{max}) = O(KM + N)$, where N is the number of segments including the top and the bottom segments. However, it is clear that $O(N) = O(\sum n_i) = O(M)$, where n_i is the number of terminals in net i . We conclude that the complexity of algo-

rithm is $O(M)$ since $K \ll M$ based on our experiments.

5. RESULTS AND CONCLUSIONS

The algorithms of routing over the cells were coded in *C* and tested on a *SUN 4 SPARC II* workstation. The algorithms are efficient and have been evaluated using several examples including examples 1, 3a, 3b, 3c, 4b, 5, and Deutsch's Difficult example from [7]. Table I compares our results with that of [7] and [8]. In terms of the number of tracks used in the channel or over the cells, our results are comparable to that of [8] and are better than [7]. However, taking CPU time in account, our algorithm is much faster than the approaches presented by [7] and [8].

In this paper, we have proposed a new fast algorithm for routing over the cells. We heuristically choose a "critical" set of segments for significantly reducing the maximum channel density. The proposed algorithm improved the computational efficiency over those of [7] and [8] yet obtained satisfactory results.

Acknowledgements

This work in the paper was supported in part by the National Science Council, R.O.C., under Contract NSC-82-0408-E003-004.

References

- [1] D. N. Deutsch, "A dogleg router," Proc. 13rd Design Automation Conference, pp. 425-433, 1976.
- [2] R. L. Rivest and C. M. Fiduccia, "A 'greedy' channel router," Proc. 19th Design Automation Conference, pp. 418-424, 1982.
- [3] T. Yoshimura and E. S. Kuh, "Efficient algorithms for channel routing," IEEE Trans. on CAD of ICAS, Vol. 1, No. 1, pp. 25-35, Jan. 1982.
- [4] M. Burstein and R. Pelavin, "Hierarchical wire routing," IEEE Trans. on CAD of ICAS, Vol. 2, No. 4, pp. 223-234, Oct. 1983.
- [5] D. N. Deutsch and P. Glick, "An over-the-cell router," Proc. 17th Design Automation Conference, pp. 32-39, 1980.
- [6] Y. Shiraishi and Y. Sakemi, "A permeation router," IEEE Trans. on CAD of ICAS, Vol. 6, No. 3, pp. 462-471, May 1987.
- [7] J. Cong and C. L. Liu, "Over-the-cell channel routing," IEEE Trans. on CAD of ICAS, Vol. 9, No. 4, pp. 408-418, Apr. 1990.
- [8] M. S. Lin, H. W. Perng, C. Y. Hwang, and Y. L. Lin, "Channel density reduction by routing over the cells," Proc. 28th Design Automation Conference, pp. 120-125, 1991.
- [9] M. S. Lin, H. W. Perng, C. Y. Hwang, and Y. L. Lin, "Channel density reduction by routing over the cells," IEEE Trans. on CAD of ICAS, Vol. 10, No. 8, pp. 1067-1071, Aug. 1991.

Authors' Biographies

Kuo-En Chang received the B.S. degree in electronic engineering from National Taiwan Institute of Technology, Taipei, Taiwan, in 1984, and M.S. and Ph.D. degrees in electrical engineering from national Taiwan University, Taipei, Taiwan, in 1986 and 1990, respectively. He has been with a director of Computer Center and is currently an Associate Professor of the Department of Information and Computer Education

TABLE I Experimental Results

Examples	Original Density	Density over the lower cells			Density over the upper cells			Final density			CPU-time (sec.)		
		[7]	[8]	ours	[7]	[8]	ours	[7]	[8]	ours	[7]	[8]	ours
1	12	3	3	2	4	2	2	9	9	9	2.0	13.3	0.4
3a	15	6	3	3	3	3	3	12	12	12	2.9	34.8	1.0
3b	17	5	4	3	2	2	2	13	13	12	3.5	276.6	1.1
3c	18	4	4	3	3	4	3	14	13	13	4.5	55.1	1.2
4b	17	4	3	3	5	2	4	16	13	13	9.7	909.2	1.6
5	20	3	4	2	4	6	6	14	11	12	4.9	17.8	1.0
Diff	19	7	3	3	8	2	5	16	16	16	25.1	33.4	5.6

[7] uses Pyramid computer, [8] uses VAX-11/8550, and we use SUN 4 SPARC II.

at National Taiwan Normal University, Taipei, Taiwan, R.O.C. His research interests include the VLSI design automation and neural network.

Sei-Wang Chen received the B.S. degree in atmospheric and space physics, the M.S. degree in geophysics from National Central University, Chung-Li, Taiwan, in 1974 and 1976, respectively, and the M.S. and Ph.D. degrees in computer science from Michigan State University, East Lansing, Michigan, in 1985 and 1989, respectively. From 1977 to 1983, he worked as

a Research Assistant in the Computer Center of Central Weather Bureau, R.O.C. In 1990, he was a researcher in the Advanced Technology Center of the Computer and Communication Laboratories at the Industrial Technology Research Institute. He is currently a professor of the Department of Information and Computer Education at the National Taiwan Normal University, Taipei, Taiwan, R.O.C. His areas of research interest include neural networks, robotics and automation, computer vision, and image processing.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

