Short Papers

A Robust Over-the-Cell Channel Router

Lih-Der Chang, Pei-Yung Hsiao, Jin-Tai Yan
and Paul-Waie Shew

***Abstract*—In this paper, we present an efficient algorithm for over-the-cell routing in the standard cell layout design technology. Two variations are discussed: one aims to minimize the channel density with fewest tracks over the cells while the other aims to minimize the final channel width. Our algorithm can fit both the two-layer and three-layer routing models. With the two-layer model, there is a single routing layer over the cells for intercell connections. While with the three-layer model, there are two disjoint routing layers over the cells for intercell connections. In our approach, we decompose the problem into two phases: (1) over-the-cell routing and (2) conventional channel routing. The over-the-cell routing phase, which is executed iteratively, consists of two steps, routing over the cells and choosing net segments within the channel. For each iteration in the over-the-cell routing phase, our algorithm removes a net or a subnet which intersects the column with highest column density and route it over the cells according to some prioritized criteria. In comparison with the previous researches, our approach achieved the best effectiveness and has used the least CPU-time. On the average, the execution speed of our router is 163 and 4163 times faster than that of [8] and [9], respectively. Besides, our algorithm can produce results comparable to those produced by the WISER algorithm [10].**

I. Introduction

Obtaining smaller chip size is always an important goal in the automatic layout design of VLSI circuits. In general, a large portion of the chip area is reserved for channel routing. In view of this, in the last decade many researchers have made efforts to reduce the number of tracks used in the channel to complete the net connections. Many good algorithms such as [1]–[4] have been proposed to solve the problem. With the progress in semiconductor fabrication technology, more routing layers are now available for interconnections in the layout design. As a result, several variations to the channel routing problem have emerged in recent years, among them include *multi-layer channel routing* and *over-the-cell (OTC) channel routing*.

In this paper, the idea of utilizing the routing area over the cells for interconnections to further reduce the channel width for the standard cell layout design style is discussed. This kind of routing that uses the extra routing area over the cells, adjacent to the channels, has been adopted by several routers [5]–[12]. These routers, called *OTC channel routers*, in most cases complete the interconnections with fewer tracks in the channel as compared to traditional channel routers since the channel density can be reduced by routing some connections over the cells.

Three layout models for the physical standard cell design using one layer of polysilicon and two layers of metal were presented in [7]. These general models, widely used in the industry for standard cell design were used in [5]–[10]. In order to verify the effectiveness of our algorithm, we follow [8] and [9] using the two and a half layer routing technology. Apparently, under this layout model, there is only one extra layer that can be used for routing over the cells; therefore, we are only allowed to perform single layer planar routing on the area over the cells. This model reserves the first metal layer (M1) for intra-cell connections; and uses the second metal layer (M2) for routing over the cells. Clearly, the number of tracks available over the cells is fixed since the cell height is limited. Recently, [11] and [12] proposed a new model with three layers for OTC channel routing. Accordingly, we have extended our algorithm to handle the three-layer model. We also assume that vias are not allowed on the area over the cells for this routing model. We can only select two planar sets of net segments to be routed over the cells; one on M2 and the other on M3.

It is well known that the OTC channel routing problem is NP-hard; therefore, many heuristic algorithms have been presented over the past. A common approach in the OTC channel routing problem is to divide the problem into following three steps and solve them independently:

1) routing over the cell;
2) choosing net segments in the channel;
3) routing in the channel.

The authors of [8] solved the problem in the first step by finding a maximum independent set of all the nets on one side of the channel to be routed over the corresponding row of cells. It was based on the assumption that the larger amount of nets being routed over the cells, the easier the routing can be accomplished in the channel. This approach does not consider the congestion in the channel when deciding which net is to be routed over the cells, hence it may only result in slight reduction in the channel density as indicated in [9]. Clearly, in selecting the net segments to be routed over the cells, the precedence should be given to those intersecting the columns whose densities are highest; otherwise, the channel density will not be reduced. This is illustrated in Fig. 1. Suppose before routing over the cells, column c has the highest column density which is equal to the channel density. In order to find a maximum independent set among n_1, n_2, and n_3 without causing any circuit violation, n_1 and n_2 may be chosen instead of n_3. However, only n_3 intersects column c. In this case, the channel density remains and, as a result, the removal of nets n_1 and n_2 from the channel does not contribute to the reduction in channel density. From this observation, the authors of [9] indicated that only the removal of critical nets contributes to the reduction in channel density. Therefore they transformed the OTC channel routing problem into a constrained covering problem and formulated it as an integer linear program-

Manuscript received April 17, 1992. This work was supported in part by the National Science Council, Republic of China under Contracts NSC 81-0404-E009-525 and NSC 81-0404-E009-129. This paper was recommended by Editor M. Marek-Sadowska.

L.-D. Chang, P.-Y. Hsaio, and J.-T. Yan are with the Department of Computer and Information Science, National Chaio Tung University, Hsinchu, Taiwan 300, Republic of China.

P.-W. Shew was with the Department of Electrical Engineering, National University of Singapore, Singapore 0511. He is now with the Department of Computer and Information Science, National Chaio Tung University, Taiwan, Republic of China.

IEEE Log Number 9212222.

0278-0070/93$03.00 © 1993 IEEE

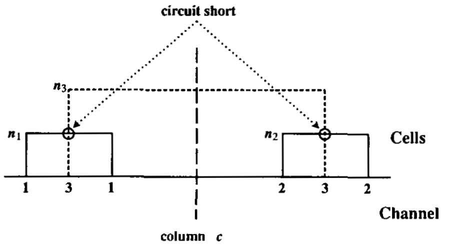

Fig. 1. For finding the maximum independent set among nets n_1, n_2, and n_3 nets n_1 and n_2 are chosen while n_3 is sacrificed. However, column c has the highest column density.

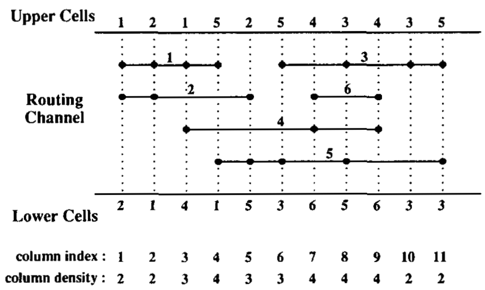

Fig. 2. The initial connection intervals for each net before the step of routing over the cells is activated. The original channel density is 4.

ming problem. They showed that their approach reduced more channel densities while using fewer tracks over the cells.

The main drawback of the integer linear programming formulation is its computational inefficiency. It remained as an open question whether if there is any efficient algorithm to solve the problems in the first and second step optimally. Different from previous approaches, our algorithm solves the first and second problems in a single iterative procedure which will be described in Section III. After routing some segments of a net over the cell and choosing a proper subset of its terminals for connection in the channel, our algorithm begins the phase of routing in the channel. Certainly, any conventional channel router can be used in this phase.

We have developed two variations to our algorithm: OTC-I and OTC-II. In brief, OTC-I stops its attempt to move net segments to area over the cells once a net segment at the columns with highest column density cannot be moved. On the other hand, OTC-II continues to select net segments intersecting the critical columns and route them over the cells until none can be moved. The latter variation follows the assumption of [8] to leave lesser net connections in the channel so that in the final phase, routing in the channel, less tracks may be required. Our experiments have verified the correctness of this assumption.

We compared our algorithm with the test examples used in [8] and [9]. The results verified that our OTC router is very much faster than the previous two algorithms and is able to produce comparable final channel densities. In order to compare the effectiveness of our algorithm, we also conducted experiments using test examples from [10]. The algorithm proposed in [10] makes use of the vacant terminals to produce better OTC channel routing results. The experiments show that our algorithm can produce routing results comparable to those presented in [10]. These experiments are described in Section IV.

II. Preliminaries

In the two layers routing model, we assume that there is only one routing layer over the cells that can be used for intercell connections. The three layers routing model, vias between M2 and M3 may or may not be allowed over the cells depending on the fabrication process. In this paper, we shall assume that vias over the cells are not used. Considering the fixed cell height, we further assume that there are six tracks on M2 and 7 tracks on M3 available over the cells for routing. In our algorithm, we decompose the problem into two phases: the OTC routing phase and the channel routing phase. The OTC routing phase includes two steps, *viz.* routing over the cells and choosing net segments.

The input of the OTC channel routing problem is two rows of terminals namely the top terminals, TOP, and bottom terminals, BOT. Each terminal is assigned a net number. All terminals with the same net number must be connected except the vacant terminals which are assigned with number 0. Two terminals $n.i$ and $n.j$ of net n are *adjacent* if there is no other terminal of net n located between column i and column j on the same row. We shall call each pair of adjacent terminals a *candidate* since it has a chance to be connected over the corresponding row of cells.

For convenience, we divide the entire routing channel into columns and we define the *net interval* of a net to be a horizontal segment which is specified by its leftmost and rightmost columns. Under the two-layer H–V routing model, distinct nets whose horizontal net intervals intersect a same column i must be assigned to different tracks at column i. We can scan every column on the routing channel from left to right, to construct the corresponding horizontal net interval. The net interval covers all its terminals and for each terminal there is a corresponding endpoint (via) on the net interval from which we may direct a vertical wire to the location of that terminal if it does not violate the vertical constraints. Thus, the net interval of a net can be divided into several pieces of horizontal connection intervals intersected by each other only at their endpoints. Accordingly, we define a *connection interval* to be a horizontal interval representing the partial connection which is routed within the channel. Furthermore, the length of a connection interval is zero when both its corresponding terminals are located on the opposite sides of the channel at the same column. We define the *column density* $d(i)$ of column i as the number of different nets that have at least one and at most two connection intervals (in the case of zero length intervals), intersected by column i. The channel density D is equal to $max\ (d(i))$ and those columns whose column densities are equal to the channel density D are called *critical columns*. As an illustration, Fig. 2 shows all the connection intervals and the column densities in a channel.

Whenever part of the connections of a net has been moved from the channel to the routing area over the cells, the connectivities of that net must be reconsidered. It is essential to select a proper subset from those already connected terminals to be further connected with others in the channel such that the channel density is minimized. Such selection is done in the net segments selection step which will be described in Section 3.4. Now, we define a *hyperterminal* of a net as a set of terminals that have been connected by wires in the OTC routing area on one side of the channel. In the beginning, every single terminal on either side of the channel is a hyperterminal. In other words, initially, every hyperterminal contains only one element. The more the terminals are connected in the area over the cells, the more the elements a hyperterminal has. The total number of hyperterminals of a net is decreased by one for each merging operation as shown in Fig. 3.

To complete the net connectivities in the channel, we only need to consider the possible net segments connecting all the hyperterminals of a net instead of connecting all its terminals. A *net seg-*

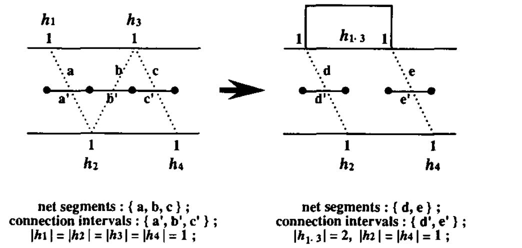

Fig. 3. The merging operation of hyperterminals and the reconstruction of connection intervals.

ment is a set of two terminals that belongs to different hyperterminals of the same net. As shown in Fig. 3, when two hyperterminals of a net are merged into one by routing the corresponding net segment over the cells, we need to find a minimum proper subset of all the relevant net segments again for connection in the channel. Finally, after the OTC routing phase has been completed the remaining connections can be solved by a conventional channel router in the second phase.

III. The OTC Routing Phase

To reduce the channel density, we must reduce the density of every critical column. This can be achieved by removing some net segments at the critical columns and route them over the cells. Such a task is handled by the first step of the OTC routing phase, i.e., routing over the cells. The second step of the OTC routing phase is choosing net segments within the channel. Before activating the OTC routing phase, we have to calculate the density for each column to identify all the critical columns.

3.1. Identification of the Initial Connection Intervals and the Candidates

To begin with, the initial connection intervals for each net must be identified before we can compute the column density at each column. Suppose there are N nets in a given channel ϕ(TOP, BOT) with c columns, and the ith net has k_i terminals, then $\Sigma_{i=1}^{N} k_i \leq 2c$, since some terminals may be vacant. As such, the total number of initial connection intervals in ϕ is bounded by

$$\sum_{i=1}^{N} (k_i - 1) = \sum_{i=1}^{N} k_i - N \leq 2c - N < 2c \quad (3.1)$$

After the determination of the highest density columns, we identify all the candidates for routing over the cells. For illustration purpose, Fig. 4 shows the candidates for the initial connection intervals in Fig. 2. For the N nets in channel ϕ the total number of candidates, χ is bounded by

$$\sum_{i=1}^{N} (k_i - 2) \leq \chi \leq \sum_{i=1}^{N} (k_i - 1)$$

Or,

$$2c - 2N \leq \chi \leq 2c - N \quad (3.2)$$

It can be shown from the above that identification of the corresponding initial connection intervals and candidates can be done in $Q(c)$ time. Note also that the OTC routing phase is executed at most $2c - N$ times.

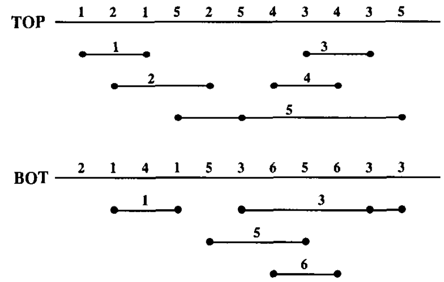

Fig. 4. The candidates for each net corresponding to the example shown in Fig. 2 before the step of routing over the cells is activated.

3.2. Two OTC Routing Algorithms, OTC-I and OTC-II

In each iteration of the OTC routing phase our OTC routing algorithms identify those candidates that are intersected by the critical columns and choose the proper one among them to be routed over the cells. In our approach, routing over the cells and choosing net segments within the channel may change the density of a column. It is clear that if there is any critical column on which none of the candidates can be routed over the cells, then the channel density will not be reduced any further by the effect of routing over the cells even if there are some other critical columns whose densities can be reduced.

On the other hand, intuitively, the fewer the nets is left within the channel, the simpler the subsequent routing in the channel will be. Based on these two observations, we have designed two algorithms, OTC-I and OTC-II for OTC routing. They differ from each other only at their ending conditions. The objective of the first algorithm, OTC-I is to complete the routing over the cells and obtain a minimum final channel density using as few tracks over the cells as possible. Therefore, it terminates the OTC routing phase once there exists any critical column on which none of the candidates can be routed over the cells under the planarity constraint. In other words, this algorithm will check all the critical columns to make sure that every one of them intersects at least one candidate which can be routed over the cells. The second algorithm, OTC-II continues its iterations of the OTC routing phase until no more candidates can be routed over the cells. For each iteration of this algorithm, if the considered column intersects none of the candidates that can be routed over the cells, then it is marked and will not be considered any more in the subsequent iterations. When all the columns have been marked, then the OTC routing phase will terminate.

3.3. Routing Over the Cells

3.3.1. Candidate Selection— Let $S(i)$ be the set of the candidates that are intersected by column i. For a critical column c, in order to make the decision on which element in $S(c)$ is the most suitable for routing over the cells, we specify some criteria with different priorities to evaluate all the possible candidates. First of all, the chosen candidate must result in a valid planar routing on either M2 or M3 over the cells. Those candidates that violate the constraint of planar routing on both M2 and M3 can be deleted from $S(c)$ during the evaluation. Two relations are defined to help deciding whether the considered candidates can be routed over the cells. We say that two candidates s_1 and s_2 have either the overlapping relation $\vee$ or the covering relation $\wedge$, if they satisfy the following conditions. Let ab be the candidate with endpoints a and b. We denote s_1 and s_2 as a_1b_1 and a_2b_2 respectively. If $a_1 < a_2 < b_1 < b_2$ or $a_2 < a_1 < b_2 < b_1$, then $s_1 \vee s_2$. That means s_1 and s_2 are over-

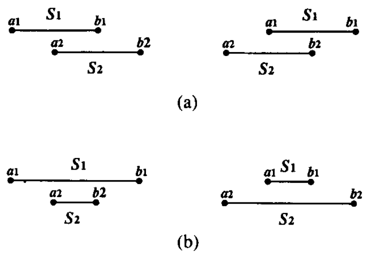

Fig. 5. (a) The overlapping relation $S_1 \vee S_2$. (b) The covering relation $S_1 \wedge S_2$.

lapping each other as shown in Fig. 5(a). On the other hand, if $a_1 < a_2 < b_2 < b_1$ or $a_2 < a_1 < b_1 < a_2$, then $s_1 \wedge s_2$. That means s_1 is covering s_2 or vice versa as shown in Fig. 5(b).

Let $L_{M2}(0)$ and $L_{M3}(0)$ be the sets of the two-terminal nets that have already been selected and routed on layers M2 and M3 respectively over the upper row of cells, and $L_{M2}(1)$ and $L_{M3}(1)$ be the sets of the two-terminal nets that have already been routed on layers M2 and M3 respectively over the lower row of cells in the previous stages. For every candidate ϵ in $S(c)$ which is on side i (0 or 1), we compare all the two-terminal nets in $L_{M2}(i)$ or $L_{M3}(i)$ with ϵ to decide whether candidate ϵ can be routed on M2 or M3 over its corresponding row of cells without causing any circuit violation.

If there is no element in $L_{M2}(i)$ or $L_{M3}(i)$ satisfying relation $\vee$ with ϵ, then ϵ is called a *valid candidate*. Only valid candidates need to be considered further. Relation $\wedge$ is allowed to exist among candidate ϵ and elements either in $L_{M2}(i)$ or $L_{M3}(i)$, because these two-terminal nets with relation $\wedge$ can be safely routed over the cells by using different tracks on the same layer. We define a covering set, $C_{Mj}(i)$ as the set of the two-terminal nets in $L_{Mj}(i)$ such that there exists a covering relation between every two elements. It is easy to show that given a channel ϕ with c columns, it takes $O(c)$ time to decide if a candidate is routable over the cells.

There can only be at most six tracks on M2 and seven tracks on M3 over either side of the channel for OTC routing. Routing over the cells involves the track assignment process which can be easily done as the following. Firstly, the tracks in the OTC routing area of the upper (lower) row of cells are numbered 1-6 on M2 and 1-7 on M3 from the bottom (top) to the top (bottom). For each element in $C_{Mj}(i)$, we process the tracks in ascending track number order. The smaller the span of a two-terminal net in $C_{Mj}(i)$, the smaller the track number is assigned to it. Since the number of tracks is fixed and the candidates are enumerated and considered from left to right, this assignment for a selected candidate has a constant complexity order. An example of the track assignment is shown in Fig. 6.

For each iteration of the OTC routing phase, in order to find a proper candidate among the remaining valid candidates in $S(c)$ to be routed over its corresponding row of cells, we introduce the following weights: *span density* (α), *overlap ratio* (β) and *average density* (γ). The span density of a candidate ϵ in $S(c)$, $\alpha(\epsilon)$ is equal to the highest column density of some column (except column c) within the coverage of its span. Suppose that candidate ϵ is on side i. All the candidates on side i are grouped into a set, $T(i)$. The overlap ratio of candidate ϵ, $\beta(\epsilon)$ is a value obtained by dividing the number of the candidates in $T(i)$ that have relation $\vee$ with ϵ by $|T(i)|$. The average density of candidate ϵ, $\gamma(\epsilon)$ is an average of all the column densities that belong to those columns within the coverage of its span. With these weights, we select the candidates according to the following criteria:

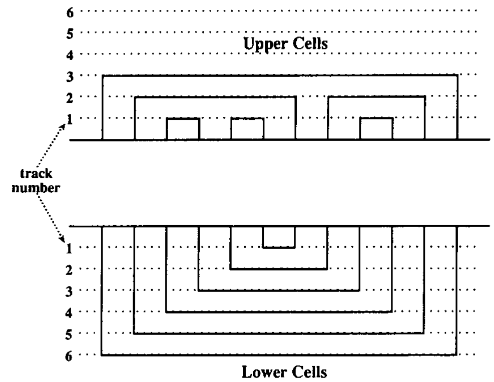

Fig. 6. An example for the track assignment.

1) the candidate with maximum α is selected;
2) in case of tie in (1), the one with minimum β is selected;
3) in case of tie in (2), the one with the largest span is selected;
4) in case of tie in (3), the one with maximum γ is selected.

The first criterion, suggests that a candidate with larger span density has a higher urgency to be routed over the cells. For the second criterion, the smaller the overlap ratio of a candidate, the more chances the others have for being routed over the cells in the subsequent iterations. Both the third and fourth criteria are based on the same reason as that of the first criterion. From (3.2) and the above discussion, the following theorem is easily established.

Theorem 3.1: Given a channel ϕ with c columns, the candidate selection can be solved in $O(c^2)$ time in each iteration of the OTC routing phase.

3.3.2. Merging of Hyperterminals—After one candidate has been selected, the connection of its two corresponding terminals can be established over the cells. Let t_1 and t_2 denote these two terminals. Let h_1 and h_2 be two hyperterminals of net n that contain terminals t_1 and t_2 respectively. A new hyperterminal, $h_{1 \cdot 2}$ is obtained by merging h_1 and h_2 so as to replace h_1 and h_2. Thus, $h_{1 \cdot 2}$ is the union of h_1 and h_2, i.e., $h_{1 \cdot 2} = h_1 \cup h_2$. It is clear that both h_1 and h_2 are located on the same side. The degree of a hyperterminal h is defined as the number of terminals contained in h, denoted by deg (h). Note that deg (h_1) + deg (h_2) $\leq c$, where c is the number of columns in the channel. It is easy to see that hyperterminals h_1 and h_2 can be merged into $h_{1 \cdot 2}$ in linear time. Thus, we established the following theorem.

Theorem 3.2: Given a channel ϕ with c columns, the merging operation can be completed in $O(c)$ time in each iteration of the OTC routing phase.

After hyperterminals h_1 and h_2 have been merged, the total number of hyperterminals of net n is decreased by one. From now on, the required connections among all the hyperterminals of net n must be reconsidered. As a result, the original connection intervals of net n as well as the column density, $d(i)$ of each affected column i, have to be updated. The modification of the column density for all the affected columns can be done in linear time since the connection intervals of a net are intersected by each other only at their endpoints. The new connection intervals of net n are reconstructed in the second step of the OTC routing phase, choosing net segments within the channel.

3.4. Net Segments Selection

After routing a candidate over the cells for net n in the first step of the OTC routing phase, a new set of hyperterminals of net n is obtained. All the terminals in each hyperterminal of net n have been connected by OTC connections. Next, we want to choose net segments to connect all the hyperterminals of net n such that the channel density is minimum. This problem called the *net segments selection problem* is NP-hard as indicated in [8]. In the following, we present an efficient heuristic algorithm to solve the problem.

In our method, the required net segments for net n are constructed by selecting a proper subset among all the possible connections. We divide the entire selection process into two steps. The objective of the first step, step A, is to determine a suitable connection interval to represent the connection between each pair of hyperterminals for net n. Let h_1 and h_2 be two hyperterminals of net n. For every terminal $n.i$ in h_1 and for every terminal $n.j$ in h_2, there exists an interval $[i, j]$ such that it has a chance to be a connection segment for h_1 and h_2. Clearly, if h_1 contains k_1 terminals and h_2 contains k_2 terminals, then there are $k_1 * k_2$ choices that may connect h_1 and h_2. In consideration of minimizing the channel density, we find the most suitable one among the $k_1 * k_2$ net segments by the following criteria. Here, we define the span density (α) of an interval as the highest column density of some column that is within the coverage of its span, and we use the same definition for average density (γ) as in Section 3.3.1.

1) The interval with minimum α is selected.
2) In case of tie in (1), the one with the smallest span is selected.
3) In case of tie in (2), the one with minimum γ is selected.

We can prove the following theorem.

Theorem 3.3: Given an instance I of the net segment selection problem for net n, the selection process in step A can be completed in $O(c^2)$ time, where c is the number of columns in I.

Algorithm OTC routing;
Input Two rows of terminals on the top and bottom edges of the channel;
Output Routing solutions over the upper and lower cells;
A new channel to be further routed in the second phase;
/* The ending conditions for both algorithms OTC-I and OTC-II are as described in Section 3.2.
1. OTC-I: There exists any critical column on which none of the candidates can be selected to be routed over the cells;
2. OTC-II: All the columns have been marked; in other words, there are no more candidates that can be routed over the cells; */
Begin
Step 1. Create the initial connection intervals and the candidates for each net;
Step 2. Check the ending condition. If it is satisfied, then go to Step 7;
Step 3. Select a proper candidate intersected by the critical column and assign it a track number for routing over the cells;
Step 4. Merge the two hyperterminals that contain the terminals corresponding to the selected candidate;
Step 5. Construct the connection graph *CG* for the net being considered;
Step 6. Find the minimum cost spanning tree of *CG* using Prim's algorithm to obtain the connection intervals for that net. Then go to Step 2;
Step 7. Generate a new channel as specified by the connection intervals of each net;
End.

Proof: Assume that net n contains k terminals ($k \geq 3$) and p hyperterminals ($2 \leq p < 2c$). Let the degree of hyperterminal h_i be t_i ($1 \leq i \leq p$). For every two hyperterminals h_i and h_j, we shall select one net segment among the $t_i * t_j$ choices. Clearly, the total number of possible net segment is $p(p-1)/2$.

Note that $\Sigma_{i=1}^{p} t_i = k \leq 2c$. Without loss of generality, let $t_i = t$ ($1 \leq i \leq p$), i.e., $p * t = k$. Then the total number of intervals in I is bounded by

$$2 \leq [p(p-1)/2] * t^2 = (1 - 1/p) * k^2/2 < 2c^2 - c. \tag{3.3}$$

Thus we can conclude that the selection process in step A has the time complexity claimed. □

After the corresponding net segment for every two hyperterminals of net n has been identified, we create a weighted complete graph called the *connection graph* of net n, $CG(n) = \langle V, E, w_1, w_2, w_3 \rangle$. Each node in V represents a hyperterminal. For every two hyperterminals, there is a corresponding edge in E, and each edge is a possible net segment for connection within the channel. There are three weights, w_1, w_2 and w_3 attached in each edge. The first weight of each edge I, $w_1(I)$, is defined as the span density of I, $\alpha(I)$. The second weight, $w_2(I)$, is the span of I. The third weight, $w_3(I)$, is the average density of I, $\gamma(I)$. For example, when the example shown in Fig. 2 is proceeded to the second iteration, a candidate of net 5 is selected for routing over the upper row of cells. The connection graph corresponding to net 5 is shown in Fig. 7. Given a net n, since we have to connect all its hyperterminals, we need to find a *spanning tree* of $CG(n)$. Moreover, since we want to minimize the channel density, the criteria used in step A are applied again to the selection in step B. Thus, the objective of the selection in step B is to find a *minimum cost spanning tree* of $CG(n)$. Prim's algorithm is used to solve this problem. It is well known that this algorithm has a complexity of $O(|\nu|^2)$ where ν is the set of vertices, hence we can formally state the following theorem.

Theorem 3.4: Given an instance I of the net segment selection problem for net n, the selection process in step B can be solved in $O(c^2)$ time, where c is the number of columns in I.

Finally, with the new set of connection intervals for net n, we need to update the column density, $d(i)$ for each affected column i with respect to each connection interval of net n.

3.5. The Entire Algorithm of the OTC Routing Phase

Our OTC routing algorithm is summarized in the following.

Let c be the number of columns in a channel. The time complexity of our algorithm is concluded by the following theorem.

Theorem 3.5: The time complexity of our OTC routing algorithm is $O(c^3)$.

Proof: If it obvious that, Step 1 takes $O(c)$ time. According to Theorem 3.1, both Step 2 and Step 3 take $O(c^2)$ time. Step 4 can be completed in $O(c)$ time according to Theorem 3.2. In Step 5, the selection process at step A needs $O(c^2)$ time to construct a connection graph in compliance with Theorem 3.3. According to Theorem 3.4, Step 6 takes $O(c^2)$ time to find the minimum cost spanning tree of CG. It is clear that Step 7 takes the same $O(c)$ time as Step 1. Moreover, the loop from Step 2 to Step 6 is repeated

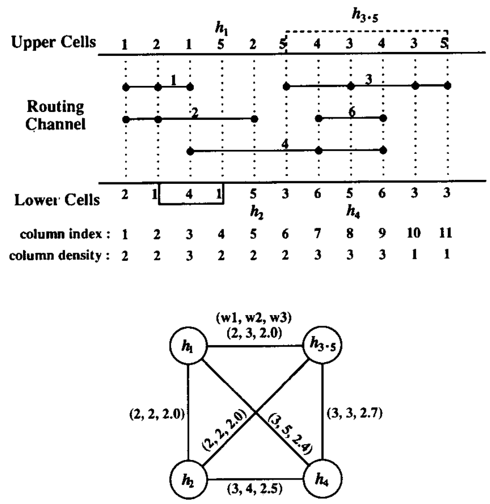

Fig. 7. The connection graph of net 5 (see Fig. 2) at the second iteration of the OTC routing phase.

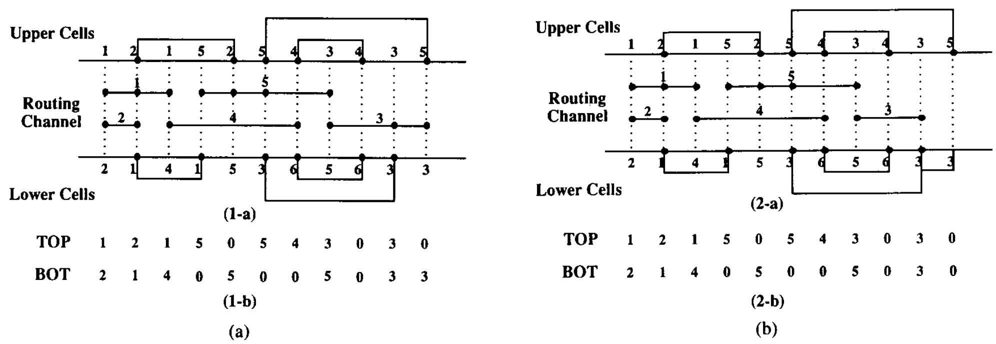

Fig. 8. The routing results and the new channels for the example shown in Fig. 2. (a) The final connection intervals for each net after the OTC routing phase has been completed. The final channel density is 2. (b) The new channel as specified by the connection intervals of each net.

TABLE I
COMPARISONS OF OUR EXPERIMENTAL RESULTS PRODUCED BY OTC-I USING TWO-LAYER MODEL WITH [8]'S AND [9]'S

Example	Original Density	Density Over the Lower Cells			Density Over the Upper Cells			Final Density			Time (s)		
		[8]	[9]	Ours	[8]	[9]	Ours	[8]	[9]	Ours	[8]	[9]	Ours
1	12	3	3	3	4	2	1	9	9	9	2.0	13.3	0.014
3a	15	6	3	3	3	3	2	12	12	12	2.9	34.8	0.027
3b	17	5	4	3	2	2	2	13	13	13	3.5	276.6	0.028
3c	18	4	4	4	3	4	2	14	13	14	4.5	55.1	0.033
4b	17	4	3	3	5	2	2	16	13	13	9.7	909.2	0.049
5	20	3	4	4	4	6	6	14	11	11	4.9	17.8	0.042
De	19	7	3	2	8	2	3	16	16	16	25.1	33.4	0.132

for $O(c)$ iterations at most. Therefore, the total time complexity is

$$O[c + c(2c^2 + c + c^2 + c^2) + c] = O(c^3). \quad \square$$

We apply both algorithms OTC-I and OTC-II using the two layer model to the example shown in Fig. 2. The corresponding routing solutions over the cells and new channels are shown in Fig. 8. After the OTC routing phase has been completed, any three-layer channel router can be used to solve the remaining connections within the channel.

IV. EXPERIMENTAL RESULTS

We have implemented both of our algorithms, OTC-I and OTC-II in C language and run them on a SPARC-2 station under the UNIX operating system. In order to compare the run times for

TABLE II
SUMMARY OF TABLE I

Total Density Reductions ($\Sigma \Delta d_i$)			Total Densities Over the Cells (n)			Average Time (sec)			Effectiveness ($\Sigma \Delta d_i / n$)		
[8]	[9]	Ours	[8]	[9]	Ours	[8]	[9]	Ours	[8]	[9]	Ours
24	31	30	61	45	40	7.5	191.5	0.046	0.39	0.69	0.75

TABLE III
COMPARISONS OF THE ROUTING RESULTS PRODUCED BY OTC-I AND OTC-II USING THREE-LAYER MODEL

		Density Over the Lower Cells				Density Over the Upper Cells				Total Density Using Two Horizontal Layers		Lower Bound of Channel Width [13]	
		M2		M3		M2		M3		M2		M3	
Example	Original Density	I	II	I	II	I	II	I	II	I	II	I	II
1	12	3	4	2	3	1	2	1	1	7	7	6	5
3a	15	3	4	3	5	3	3	1	2	11	10	6	5
3b	17	3	4	3	3	2	2	2	2	12	12	6	6
3c	18	4	4	2	2	3	4	2	3	11	11	6	6
4b	17	3	4	4	5	3	4	2	2	10	10	6	5
5	20	4	4	2	2	4	4	3	4	9	9	5	5
De	19	5	6	2	3	4	5	3	4	13	12	7	6

all the test examples with those of [8] and [9], we have also ported our programs to the same machine, VAX-11/8550 running the ULTRIX operating system. All the benchmarks were taken from [3] except Deutsch's difficult channel denoted by *De*.

Table I shows our results produced by algorithm OTC-I using the two-layer model together with those of [8] and [9]. The comparison is summarized in Table II. It is observed that in our results, the total number of density reductions for these seven examples is almost as good as that of [9] and is greater than that of [8]. On the other hand, the total number of tracks over the cells we use is fewer than that of both [8] and [9]. We denote the density reduction by Δd_i and their total by $\Sigma \Delta d_i$. We also denote the total number of tracks used over the cells by n. The effectiveness, E is defined as a positive real number obtained by dividing $\Sigma \Delta d_i$ by n, i.e., $E = \Sigma \Delta d_i / n$. Our approach achieves the best effectiveness in these benchmarks. It means that our method utilizes the routing area over the cells most efficiently. Furthermore, our OTC router takes much less CPU-time than that of both [8] and [9] did. On the average, the execution speed of our program is 163 and 4163 times faster than that of [8] and [9] respectively.

For the purpose of verifying the correctness of the assumption made by [8], that the fewer the nets left within the channel, the fewer tracks a conventional channel router will need to complete the routing. The results obtained by OTC-I and OTC-II, using the three-layer model are compared in Table III. It is very natural that algorithm OTC-II uses more tracks over the cells than algorithm OTC-I does, since algorithm OTC-II routes as many nets over the cells as it can while algorithm OTC-I stops once it encounters a critical column on which none of the candidates can be routed over the cells. As a result, algorithm OTC-II achieves better total density for the channel width than algorithm OTC-I does. We have

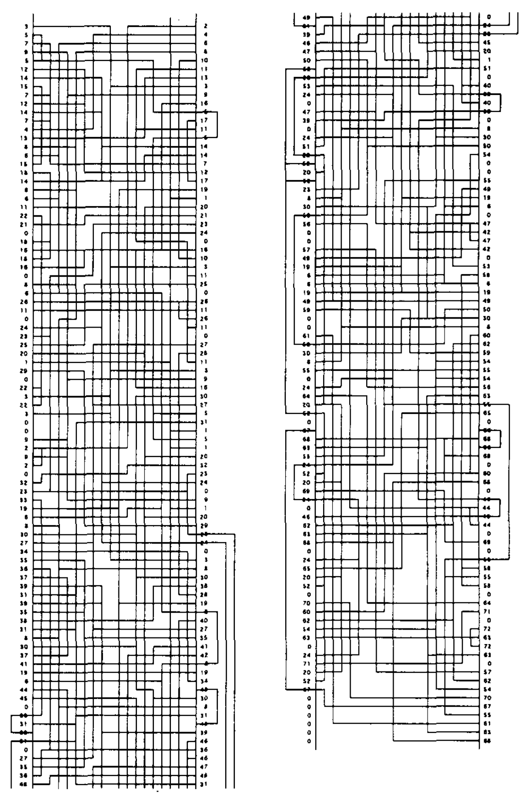

Fig. 9. Routing results of OTC-I for Deutsch's difficult example.

TABLE IV
COMPARISON WITH WISER [10] USING THE PRIMARY-I TEST EXAMPLES

Channel No.	% Vacant Terminals Available	Number of Tracks			
		Gr.	OTC	WISER	Ours
1	85	11	9	5	6
2	74	16	13	12	14
3	64	21	19	16	16
4	60	24	24	21	20
5	64	21	20	17	17
6	52	29	21	18	17
7	58	22	21	14	14
8	50	24	18	18	17
9	53	21	20	16	15
10	60	15	13	11	10
11	63	17	14	12	11
12	64	15	14	11	10
13	62	13	11	9	10
14	64	13	13	9	11
15	63	11	9	7	7
16	67	11	9	7	8
17	66	14	10	8	9
18	91	6	4	3	4

observed that the density of a column may be changed by the step of routing over the cells and by the step of choosing net segments within the channel. In particular, for the Deutsch's difficult channel, the channel density is reduced from 19 down to 13 by algorithm OTC-I and further reduced to 12 by algorithm OTC-II. This is due to the fact that algorithm OTC-II repeats the OTC routing phase many times more than algorithm OTC-I does; thus the condition within the channel can be further improved by algorithm OTC-II, and as a result, the smaller final density may be obtained. In particular, algorithm OTC-II finds a six-track lower bound for the final channel width for Deutsch's difficult channel. It is the best known lower bound for this example. A solution for the Deutsch's difficult channel produced by OTC-I for single layer routing over the cells is shown in Fig. 9.

We have also performed experiments using the same set of Primary I test examples used in [10]. This is to compare our algorithm with the WISER algorithm presented in [10], which makes use of the vacant terminals for the OTC routing. The results produced by OTC-I are presented in Table IV. Again, since our approach reduces the congestion in the channel by routing those net segments on the critical columns over the cells, we obtained results comparable to those produced by the WISER algorithm.

V. CONCLUSION

We have proposed a heuristic algorithm for the over-the-cell channel routing problem. It is simple yet very efficient. Two variations to the algorithm are presented: OTC-I uses the least number of tracks over the cells and OTC-II routes as many net segments over the cells as it can to minimize the final channel width. Our experimental results have shown that our algorithms are very fast, on the average the speed is 163 and 4163 times faster than algorithms presented in [8] and [9]. In addition, the effectiveness we achieve is superior to that of these algorithms. Besides, our algorithms can produce the final channel width comparable to those produced by the WISER algorithm [10] for the Primary-I test examples.

ACKNOWLEDGMENT

The authors thank Y. L. Lin at National Tseng Hua University, R.O.C., for providing them the benchmarks and the VAX-11/8550 machines. They are grateful to Editor M. Marek-Sadowska and N. A. Sherwani at Western Michigan University for making the Primary-I benchmarks available to them. They would also like to thank Y. C. Lim at National University of Singapore for supporting Mr. Shew, a visiting Ph.D. student at National Chiao Tung University since February 1992.

REFERENCES

[1] D. N. Deutsch, "A dogleg router," in *Proc. 13th Design Automation Conf.*, pp. 425–433, 1976.

[2] R. L. Rivest and C. M. Fiduccia, "A 'greedy' channel router," in *Proc. 19th Design Automation Conf.*, pp. 418–424, 1982.

[3] T. Yoshimura and E. S. Kuh, "Efficient algorithms for channel routing," *IEEE Trans. Computer-Aided Design* vol. CAD-1, pp. 25–35, Jan. 1982.

[4] M. Burstein and R. Pelavin, "Hierarchical wire routing," *IEEE Trans. Computer-Aided Design*, vol. CAD-2, pp. 223–234, Oct. 1983.

[5] D. N. Deutsch and P. Glick, "An over-the-cell router," in *Proc. 17th Design Computer-Aided Design Automation Conf.*, pp. 32–39, 1980.

[6] Y. Shiraishi and Y. Sakemi, "A permeation router," *IEEE Trans. Computer-Aided Design*, vol. CAD-6, pp. 462–471, May 1987.

[7] J. Cong, B. Preas, and C. L. Liu, "General models and algorithms for over-the-cell routing in standard cell design," in *Proc. 27th Design Automation Conf.*, pp. 709–715, 1990.

[8] J. Cong and C. L. Liu, "Over-the-cell channel routing," *IEEE Trans. Computer-Aided Design*, vol. 9, pp. 408–418, Apr. 1990.

[9] M. S. Lin, H. W. Perng, C. Y. Hwang, and Y. L. Lin, "Channel density reduction by routing density reduction by routing over the cells," in *Proc. 28th Design Automation Conf.*, pp. 120–125, 1991.

[10] N. D. Holmes, N. A. Sherwani and M. Sarrafzadeh, "New algorithm for over-the-cell channel routing using vacant terminals," in *Proc. 28th Design Automation Conf.*, pp. 126–131, 1991.

[11] M. Terai, K. Takahashi, K. Nakajima, and K. Sato, "A new model for over-the-cell channel routing with three layers," in *Proc. Int. Conf. on CAD-91*, pp. 432–435, 1991.

[12] N. D. Holmes, N. A. Sherwani, and M. Sarrafzadeh, "Algorithms for three-layer over-the-cell channel routing," in *Proc. Int. Conf. on CAD-91*, pp. 428–431, 1991.

[13] Y. K. Chen and M. L. Liu, "Three-layer channel routing," *IEEE Trans. Computer-Aided Design*, CAD-3, pp. 156–163, 1984.

On the Over-Specification Problem in Sequential ATPG Algorithms

Kwang-Ting Cheng and Hi-Keung Tony Ma

***Abstract*—Most sequential ATPG (Automatic Test Pattern Generation) programs employ the time-frame expansion technique. Within a time-frame, combinational test generation algorithms that are variations of D-algorithm or PODEM are used. In this paper, we show that some ATPG programs may err in identifying untestable faults. In other**

Manuscript received July 7, 1992; revised January 28, 1993. This paper was recommended by Associate Editor B. Krishnamurthy.

An earlier version of this paper appeared in DAC-92.

K.-T. Cheng is with AT&T Bell Laboratories, Murray Hill, NJ 07974.

H.-K. T. Ma is with Synopsis Inc., Mountain View, CA 94043.

IEEE Log Number 9212220.