Over-the-Cell Channel Routing

JINGSHENG CONG, MEMBER, IEEE, AND C. L. LIU, FELLOW, IEEE

Abstract—**A common approach to the over-the-cell channel routing problem is to divide the problem into three steps: 1) routing over the cells, 2) choosing net segments, and 3) routing within the channel. In this paper, we show that the first step can be reduced to the problem of finding a maximum independent set of a circle graph, and thus can be solved optimally in quadratic time. Also, we show that to determine an optimal choice of net segments in the second step is *NP*-hard in general, and we present an efficient heuristic algorithm for this step. The third step can be carried out using a conventional channel router. Based on these theoretical results, we design an over-the-cell channel router that produces solutions which are better than the optimal two-layer channel routing solutions for all test examples. Our over-the-cell channel router also outperforms the over-the-cell channel router in [19]. In particular, for the famous Deutsch's difficult example, our solution yields a saving of 10.5 percent in channel routing area when compared with the optimal two-layer channel routing solution, and a saving of 15 percent in channel routing area when compared with the routing solution produced by the over-the-cell channel router in [19].**

I. Introduction

CHANNEL ROUTING is a basic yet very important step in the automatic layout design of VLSI circuits. For the standard cell design style, after the cells are placed in rows and necessary feedthroughs are inserted, a channel router is used to complete the interconnections between cells (Fig. 1). The conventional channel routing problem is restricted to utilizing two routing layers in the channel for interconnections. Extensive studies have been carried out on the conventional channel routing problem, and there are several channel routers which can produce solutions that use at most one or two tracks beyond the channel density for most of the practical test examples. (For example, see [6], [22], [18], [2], [16].) To further reduce the channel routing area, several channel routers have been designed to take advantage of the possibility of utilizing the routing area over the cells for interconnections [7], [13], [19], [12]. These routers are called *over-the-cell channel routers*. In most cases, over-the-cell channel routers can complete the interconnections using fewer tracks in the channel than the density of the channel. Since a large portion of the area of a VLSI circuit is used for channel routing, savings in channel area obtained by using over-the-cell routers are usually significant. As

Manuscript received December 20, 1988; revised June 27, 1989. This work was supported in part by grants from the National Science Foundation, the Semiconductor Research Corporation, and AT&T Bell Laboratories. This paper was recommended by Associate Editor R. H. J. M. Otten.

This is an expanded version of the paper presented at ICCAD-88.

The authors are with the Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL 61801.

IEEE Log Number 8933514.

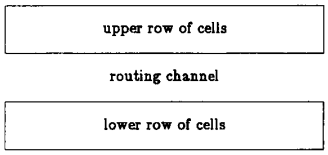


Fig. 1. Channel routing between cell rows.

two and a half layer routing technology (using one layer of polysilicon and two layers of metal) becomes more and more widely used in standard cell design, routing with one extra layer over the cells becomes both practical and important.

Since the conventional channel routing problem is known to be *NP*-hard [21], and the over-the-cell channel routing problem is a generalization of the conventional channel routing problem, it is easy to see that the over-the-cell channel routing problem is also *NP*-hard [12]. A common approach to the over-the-cell channel routing problem is to divide the problem into three steps as follows:

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

Obviously, the third step can be accomplished using a conventional channel router. Little was known about the complexity of the first step and the second step. Several heuristic algorithms have been proposed previously [13], [19], [12]. It was an open question whether there are efficient algorithms for solving the problems in the first step and the second step optimally.

In this paper, we show that the first step can be formulated in a very natural way as the problem of finding a maximum independent set of a circle graph. Since the latter problem can be solved in quadratic time optimally, we obtain an efficient optimal algorithm for the first step. Also, we show that the second step can be formulated as the problem of finding a minimum density spanning forest of a graph. We demonstrate that the minimum density spanning forest problem in *NP*-hard. We also present an efficient heuristic algorithm which produces very satisfactory results. Based on these algorithms together with a greedy channel router [18] for the third step, we design an efficient over-the-cell channel router which performs very well for all test examples. On the average, our routing solution attains a saving of 19.6 percent of the channel routing area when compared with the optimal two-layer channel routing solutions, and attains a saving of 9.6 per-

cent when compared with the routing solutions obtained by the over-the-cell router in [19].

The rest of this paper presents these results in detail. In Section II we give precise formulations of the three steps. In Section III we present a polynomial time optimal algorithm for the first step. In Section IV we show that the second step is *NP*-hard and describe a heuristic algorithm for this step. The computational complexity of our over-the-cell channel router is summarized in Section V. Finally, experimental results are presented in Section IV.

II. Formulation of the Problem

We assume that there are two routing layers in the channel, and that there is a single routing layer over the cells for intercell connections. Clearly, the over-the-cell routing must be planar. Our routing model is based on the two and a half layer routing technology for standard cells which is now widely used in the industry. In the two and a half layer routing technology, we have one layer of polysilicon and two layers of metal. We can use the layer of polysilicon and one layer of metal for the layout of standard cells. Thus we can use the other layer of metal for routing over the cells. We shall use the two metal layers for routing in the channel. We also assume that terminals are accessible at the edges of the cells on all layers. Fig. 2 shows a valid over-the-cell channel routing solution in our model.

The first step of the over-the-cell channel routing problem is to connect terminals on each side of the channel using over-the-cell routing area on that side. We carry out the same procedure for each side (upper or lower) of the channel independently. Let $n \,.\, c$ denote the terminal of net n at column c. In a given planar routing on one side of the channel, we define a *hyperterminal* of a net to be a maximal set of terminals which are connected by wires in the over-the-cell routing area on that side. For example, for the terminals in the upper side of the channel in Fig. 2, {5.4, 5.6, 5.11} is a hyperterminal of net 5. {2.2} is also a hyperterminal. Obviously, when we proceed to the routing within the channel step (the third step), we only need to connect all the hyperterminals of a net instead of connecting all the terminals of the net, because the terminals in each hyperterminal have already been connected in the over-the-cell routing area. Intuitively, the fewer hyperterminals we end up with after routing over the cells, the simpler the subsequent channel routing problem. Thus the first step of our problem can be formulated as: *to route a row of terminals using a single routing layer on one side of the row such that the number of hyperterminals is minimum*. We shall show in Section III how to solve this problem optimally in polynomial time.

After the completion of the over-the-cell routing step, the second step is to choose net segments to connect the hyperterminals that belong to the same net. A *net segment* is a set of two terminals of the same net that belong to two different hyperterminals. For example, for the two hyperterminals of net 1 on the opposite sides of the channel in Fig. 3, there are four possible net segments that can

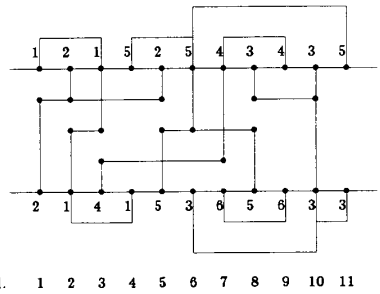


Fig. 2. A valid over-the-cell routing solution.

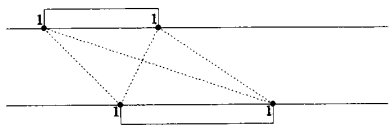


Fig. 3. Possible net segments for connecting two hyperterminals.

be used to connect these two hyperterminals (indicated by dashed edges), while only one of them is needed to complete the connection. Thus the second step of our problem is *to choose net segments to connect all the hyperterminals of each net such that the resulting channel density is minimum*. In Section IV, we shall prove that this problem is *NP*-hard. Also, we shall present an efficient heuristic algorithm for solving this problem.

After we have chosen the net segments for all the nets, we shall connect the terminals which are specified by the selected net segments selected using the routing area in the channel. Our problem is now reduced to the conventional two-layer channel routing problem. We use a greedy channel router [18] for this step. Other well-known two-layer channel routers may be used as well.

III. Routing Over the Cells

As stated in the preceding section, the first step of the over-the-cell channel routing problem is to route a row of terminals using a single routing layer on one side of the channel such that the resulting number of hyperterminals is minimized. We call this problem the *multiterminal single-layer one-sided routing problem* (MSOP). Fig. 4(a) shows an instance of the problem for the upper side of the channel in Fig. 2. A *valid routing solution* is a set of non-intersecting wires which connect terminals in the same net on one side of the channel such that all the wires lie on one side of the channel. For example, Fig. 4(b) is a valid routing solution for the instance in Fig. 4(a). If the number of terminals that belong to each net is no more than 2, we call the corresponding routing problem the *two-terminal single-layer one-sided routing problem* (TSOP). We have the following theorem.

Theorem 3-1: Given an instance I of MSOP, we can transform I into an instance I' of TSOP in $O(c^2)$ time such that from a given optimal solution S' of I', we can construct an optimal solution S of I in $O(c^2)$ time, where c is the number of terminals.

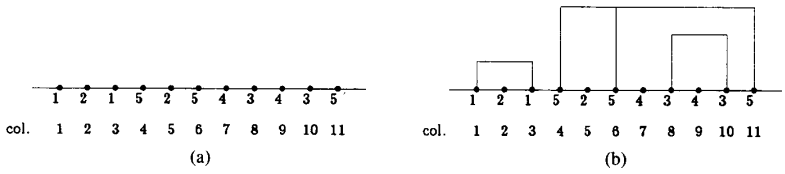


Fig. 4. (a) An instance of MSOP. (b) One of its valid solutions.

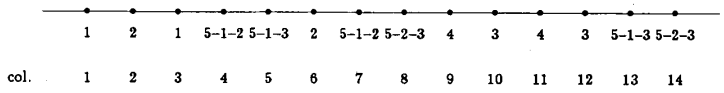


Fig. 5. Instance of TSOP problem obtained after transformation.

Proof: First, we present the transformation procedure. Given an instance I of MSOP, we construct I' of TSOP as follows: let n be a net with k terminals ($k \geq 3$). Let $p_1, p_2, \cdots, p_k$ denote the k terminals. We shall replace net n by $k(k-1)/2$ two-terminal nets which will be denoted n - i - j, ($1 \leq i < j \leq k$). Specifically, we split p_l, ($1 \leq l \leq k$), into $k-1$ terminals $p_{l,1}, p_{l,2}, \cdots, p_{l,k-1}$. (These terminals are placed next to each other as a group.) $p_{l,i}$ is assigned to net n - i - l for $1 \leq i \leq l-1$, and $p_{l,j}$ is assigned to net n - l - j + 1 for $l \leq j \leq k$ - 1. It is easy to see that I' thus constructed is an instance of TSOP. For example, the instance in Fig. 4(a) can be transformed into the instance in Fig. 5. Note that nets 5 - 1 - 2, 5 - 1 - 3, and 5 - 2 - 3 are nets introduced to replace the 3-terminal net, net 5. Let N denote the total number of nets. Let $k_1, k_2, \cdots, k_N$ denote the numbers of terminals in each net. Let $K = \max_{1 \leq i \leq N} k_i$, i.e., K is the maximum number of terminals in a net. Note that $\sum_{i=1}^{N} k_i = c$. Then, the total number of terminals in I' are bounded by

$$\sum_{i=1}^{N} k_i^2 \leq \sum_{i=1}^{N} K \cdot k_i = K \cdot \sum_{i=1}^{N} k_i = K \cdot c \leq c^2. \quad (3.1)$$

Thus we can conclude that our transformation procedure has the time complexity claimed.

We now show the correctness of the transformation procedure. We define first the degree of a hyperterminal. In a MSOP routing solution S, a hyperterminal h is said to have *degree* t if h contains t terminals. For example, in Fig. 4(b) there is a hyperterminal of degree 3 which contains the three terminals at column 4, 6, and 11. We have the following lemmas.

Lemma 3-1: Let d_i denote the number of hyperterminals of degree i in a given MSOP routing solution S. Then S has $c - \sum_{i \geq 2} (i-1) d_i$ hyperterminals in total, where c is the number of columns.

Proof: It is clear that $\sum_{i \geq 1} i \cdot d_i = c$. Thus the total number of hyperterminals equals $\sum_{i \geq 1} d_i = c - \sum_{i \geq 2} (i-1) d_i$. □

Lemma 3-2: Let I be an instance of MSOP and I' the corresponding instance of TSOP constructed according to our transformation procedure. Let S be a routing solution for I. For each hyperterminal h in S, assume that the degree of h is t and h contains the $i_1, i_2, \cdots, i_t$th terminals of net n. We connect the $k-1$ two-terminal nets n - i_1 - i_2, n - i_2 - i_3, $\cdots$, n - i_{t-1} - i_t in I' to obtain a routing solution S' of I'. Then S is a valid routing solution for I if and only if S' is a valid routing solution for I'.

Proof: First, for each hyperterminal h of degree t in S, the wires connecting the corresponding $t-1$ two-terminal nets in S' do not intersect. Moreover, the wires connecting hyperterminal h_1 of net n_1 intersects the wires connecting hyperterminal h_2 of net n_2 in S if and only if some of the corresponding 2-terminal nets of h_1 intersect some of the corresponding 2-terminal nets of h_2 in S', because our transformation keeps the relative ordering of the terminals. (For example, if terminal p_i of net n_1 is to the left of terminal p_j of net n_2 in I, then all the terminals corresponding to p_i are to the left of every terminals corresponding to p_j in I'.) □

Based on these two lemmas, we can complete our proof of Theorem 3-1 as follows. According to Lemma 3-1, minimizing the total number of hyperterminals in S is equivalent to maximizing $\sum_{i \geq 2} (i-1) d_i$ in S. According to Lemma 3-2, each hyperterminal of degree k in S corresponds to $k-1$ connected 2-terminal nets in S'. Therefore, minimizing the total number of hyperterminals in S is equivalent to maximizing the number of connected 2-terminal nets in S' in I'. Moreover, according to Lemma 3-2, after we obtain an optimal solution S' for I', we can construct the corresponding solution S for I simply by doing a linear scan of the connected 2-terminal nets in S'. This concludes the proof of Theorem 3-1. □

We concentrate now on TSOP. Let $n_1 = \{n_1 . i, n_1 . j\}$ and $n_2 = \{n_2 . k, n_2 . l\}$ be two two-terminal nets. We say that net n_1 *intersects* net n_2 if the locations of their terminals satisfy the relation $i < k < j < l$ (Fig. 6). For an instance I of TSOP, the *intersection graph* of I is defined to be an undirected graph $G(I) = \langle V, E \rangle$. Each node in V represents a net in I. There is an edge (n_1, n_2) in E if and only if net n_1 intersects net n_2. For example, the intersection graph for the instance in Fig. 5 is shown in Fig. 7. Given a routing solution S for I, according to

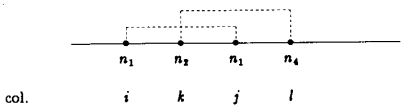


Fig. 6. Two nets which intersect.

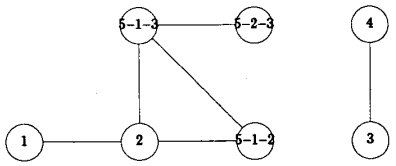


Fig. 7. The intersection graph for the example in Fig. 5.

Lemma 3-1, the number of hyperterminals in S is equal to $c - H$ where H is the number of connected two-terminal nets. Furthermore, S is a valid solution if and only if these connected two-terminal nets do not intersect. Therefore, the problem of finding a routing solution for I that has a minimum number of hyperterminals is equivalent to the problem of finding a maximum independent set of $G(I)$. In general, the problem of finding a maximum independent set of a graph is *NP*-hard [9]. However, we can show that the intersection graph thus defined for any instance of TSOP is always a circle graph. A circle graph is defined as follows [8]: let C be a set of chords in a circle. The corresponding *circle graph* $G(C)$ is an undirected graph in which each vertex represents a chord, and two vertices are connected if and only if the corresponding chords intersect (Fig. 8). Given an instance I of TSOP, we can imagine that we bend the upper (lower) edge of the channel such that the two ends of the edge meet to form a circle. Consequently, a two-terminal net becomes a chord in the circle thus formed. It is not difficult to see that the corresponding circle graph is the intersection graph of I. It is known that the problem of finding a maximum independent set of a circle graph can be solved in polynomial time [10], [1], [20]. In particular, using the dynamic programming approach presented in [20], we have the following lemma.

Lemma 3-3: TSOP can be solved optimally in $O(c^2)$ time, where c is the number of columns.

Combining Theorem 3-1 and Lemma 3-3, we obtain the following theorem.

Theorem 3-1: If the number of terminals in each net is bounded by a constant, MSOP can be solved optimally in $O(c^2)$ time, where c is the number of columns.

Proof: According to Theorem 3-1, reduction from MSOP to TSOP can be done in $O(c^2)$ time. Moreover, according to (3.1), the number of terminals in the instance of TSOP is $O(K \cdot c)$, where K is the maximum number of terminals in a net. According to Lemma 3-3, an optimal solution of the instance of TSOP can be found in $O(K^2 \cdot c^2)$ time. Since K is bounded by a constant, the complexity is $O(c^2)$. □

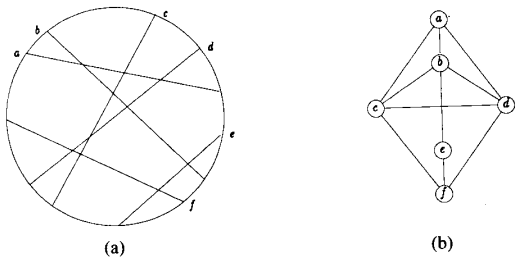


Fig. 8. (a) A set of chords. (b) The corresponding circle graph.

It is practical to assume that the number of terminals in a net is bounded by a constant. In fact, for circuits from industrial examples, it is known that the average number of terminals per net is between 2 and 3, and the maximum number of terminals in a net is between 8 and 16. However, for theoretical interest, when K is not bounded by any constant, we can show that MSOP can be solved optimally in $O(c^3)$ time.

In fact, MSOP can be solved directly by a dynamic programming method in $O(c^3)$ time without being transformed to TSOP. Given an instance I of MSOP, let $I(i, j)$ denote the instance resulting from restricting I to the interval $[i, j]$. Let $\mathbb{S}(i, j)$ denote the set of all the possible routing solutions for $I(i, j)$. Let:

$$M(i, j) = \max_{S \in \mathbb{S}(i,j)} \left\{ \sum_{k \geq 2} (k - 1) \cdot d_k(S) \right\}$$

where $d_k(S)$ is the number of hyperterminals of degree k in S. If there is no terminal at column i, clearly, $M(i, j) = M(i + 1, j)$. Otherwise, assume that the terminal at column i belongs to net n. Let $x_{n1}, x_{n2}, \cdots, x_{ns}$ be the column indexes of other terminals that belong to net n in interval $[i, j]$. Then, it is easy to verify that

$$M(i, j) = \max \Big(M(i + 1, j), \max_{1 \leq l \leq s} \{ M(i + 1, n_l) + M(n_l, j) \} \Big).$$

We leave it to the reader to complete the proof that this recurrence relation leads to an $O(c^3)$ time dynamic programming solution to MSOP.

IV. Net Segments Selection

After the completion of the over-the-cell routing for both the upper cell row and the lower cell row, we obtain a set of hyperterminals. The terminals in each hyperterminal are connected together by over-the-cell connections. We now want to choose net segments to connect all the hyperterminals of each net such that the channel density is minimized. We call this problem the *net segment selection problem*. Several heuristic algorithms were proposed in [14], [12]. However, the complexity of the problem was not known before. In this section, we show that the general net segment selection problem is *NP*-hard. Then we present an efficient heuristic algorithm to solve the problem.

For an instance I of the net segment selection problem, we define the *connection graph* $CG(I) = \langle V, E, w \rangle$ to be a weighted multigraph. Each node in V represents a hyperterminal. Let h_1 and h_2 be two hyperterminals that belong to the same net n. For every terminal $n \cdot i$ in h_1 and for every terminal $n \cdot j$ in h_2 there is a corresponding edge (h_1, h_2) in E, and the weight of this edge $w((h_1, h_2))$ is the interval $[i, j]$ (assume that $i \leq j$, otherwise, it will be $[j, i]$). Clearly, if h_1 contains p_1 terminals and h_2 contains p_2 terminals, then there are $p_1 \cdot p_2$ parallel edges connecting h_1 and h_2 in $CG(I)$. Furthermore, corresponding to each net in I there is a connected component in $CG(I)$. For example, the connected component corresponding to net 3 in the example in Fig. 2 is shown in Fig. 9. Given an instance I of the net segment selection problem, since we have to connect all the hyperterminals in the same net together for every net in I, we need to find a spanning forest of $CG(I)$. Moreover, since we want to minimize the channel density, we need to minimize the density of the set of intervals associated with the edges in the spanning forest. Therefore, the net segment selection problem can be formulated as the following problem.

Minimum Density Spanning Forest Problem (MDSFP)

Instance: A weighted graph $G = \langle V, E, w \rangle$ in which the weight $w(e)$ for each edge $e \in E$ is an interval, and an integer D.

Question: Is there a subset of edges $N \subseteq E$ that form a spanning forest of G, and the density of the interval set $\{w(e) \mid e \in N\}$ is no more than D?

In the rest of this paper, we use MDSFP as the general formulation of the net segment selection problem. It turns out that MDSFP is a very important problem because many density-related minimization problems can be reduced to it. For example, we show that the following minimum density representative problem can be reduced to MDSFP.

Minimum Density Representative Problem

Instance: A collection of interval sets $\Omega = \{A_1, A_2, \cdots, A_m\}$, where each A_i is a set of intervals, and an integer M.

Question: Can we choose an interval I_i from A_i as its representative ($1 \leq i \leq m$) such that the density of the set of representatives $\{I_1, I_2, \cdots, I_m\}$ is no more than M? (Interval I_i is called a *representative* of interval set A_i. The set $\{I_1, I_2, \cdots, I_m\}$ is called a *representative set* of Ω).

The minimum density representative problem has applications to several VLSI routing problems. The following lemma shows that the minimum density representative problem can be reduced to MDSFP.

Lemma 4-1: The minimum density representative problem can be reduced to MDSFP in linear time.

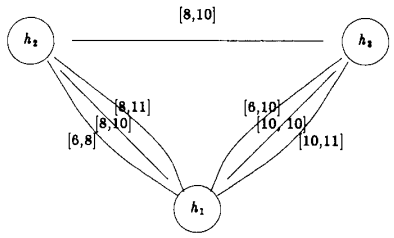


Fig. 9. The connected component induced by net 3 in Fig. 2.

Proof: Given an instance I of the minimum density representative problem, let $\Omega = \{A_1, A_2, \cdots, A_m\}$ be the collection of interval sets in I. Let M be the threshold density in I. We construct an instance I' of MDSFP as follows: the weighted graph G in I' consists of m connected components, $G_1, G_2, \cdots$, and G_m. Component G_i corresponds to interval set A_i. Assume that $A_i = \{I_{i,1}, I_{i,2}, \cdots, I_{i,p_i}\}$. Then, G_i consists of two vertices which are connected by p_i parallel edges. The weights of these p_i edges are the intervals $I_{i,1}, I_{i,2}, \cdots$, and I_{i,p_i}. (See Fig. 10.) Let the threshold density D in I' equal M. Since any spanning forest of G contains exactly one edge from each G_i ($1 \leq i \leq m$), it is easy to verify that Ω has a representative set of density no more than M if and only if G has a spanning forest of density no more than D. Moreover, it is not difficult to see that I' can be constructed in linear time. □

We shall conclude that MDSFP is *NP*-complete by showing that the minimum density representative problem is *NP*-complete [9]. Lemma 4-2 shall show that the minimum density representative problem is *NP*-complete. In fact, from the construction in the proof of Lemma 4-2, we can see that the minimum density representative problem remains *NP*-complete if each interval set contains at most two intervals.

Lemma 4-2: The minimum density representative problem is *NP*-complete.

Proof: First, let Ω be a collection of m interval sets. Let M be the threshold of density. Given a representative set R of Ω, we can determine whether the density of R is larger than M in $O(m \log m)$ time. Therefore, we can check if a given representative set is a valid solution in polynomial time. Thus the minimum density representative problem is in *NP*.

Next, we show that the minimum density representative problem is *NP*-complete by constructing a reduction from the monotone 3-SAT problem [9]. An instance of the monotone 3-SAT problem is a Boolean formula B in conjunctive normal form in which each clause contains either 3 un-negated variables or 3 negated variables. For example, $B = (\bar{x}_1 + \bar{x}_2 + \bar{x}_3)(x_1 + x_2 + x_4)(x_1 + x_3 + x_4)(\bar{x}_2 + \bar{x}_3 + \bar{x}_4)(x_2 + x_3 + x_4)$ is an instance of the monotone 3-SAT problem. The question is to determine whether there is a truth assignment to the variables in B such that B is satisfied (i.e., $B = 1$). The monotone 3-SAT problem was shown to be *NP*-complete [11].

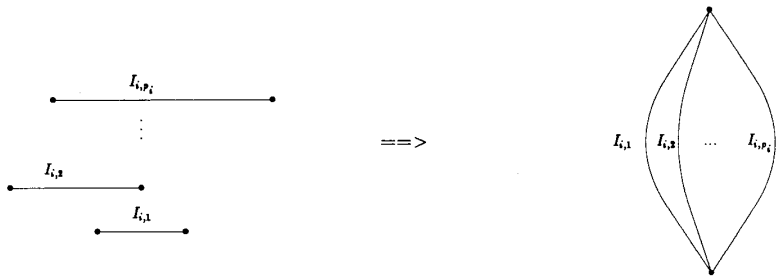


Fig. 10. Reduction from the minimum density representative problem to MDSFP.

Given an instance B of the monotone 3-SAT problem, we reduce B to an instance I of the minimum density representative problem in the following way. Assume that B contains n variables and m clauses. Let $x_1, x_2, \cdots, x_n$ denote the variables and $c_1, c_2, \cdots, c_m$ the clauses in B A clause is called a *positive clause* if it contains only un-negated variables. Similarly, a clause is called a *negative clause* if it contains only negated variables. Without loss of generality, we assume that each variable x_i in B appears in at least one clause. (Otherwise, we can assign an arbitrary truth value to x_i.) Moreover, the intervals we shall construct are open intervals instead of close intervals. (Since we shall construct only a finite (in fact, polynominal) number of open intervals, we can replace each open interval (a, b) by a close interval $[a + \epsilon, b - \epsilon]$. We can choose ϵ small enough so that two open intervals intersect if and only if their corresponding close intervals intersect.)

For each variable x_i ($1 \le i \le n$), assume that x_i or $\bar{x}_i$ appears in p_i clauses. Let the indexes of these clauses be $\alpha(i, 1), \alpha(i, 2), \cdots, \alpha(i, p_i)$ in increasing order. We introduce $2p_i + 2$ interval sets of size two: $A_{i,1} = \{I_{i,1}, \bar{I}_{i,1}\}$, $A_{i,2} = \{I_{i,2}, \bar{I}_{i,2}\}, \cdots, A_{i,p_i} = \{I_{i,p_i}, \bar{I}_{i,p_i}\}$, and $B_{i,0} = \{J_{i,0}, \bar{J}_{i,0}\}$, $B_{i,1} = \{J_{i,1}, \bar{J}_{i,1}\}, \cdots, B_{i,p_i+1} = \{J_{i,p_i+1}, \bar{J}_{i,p_i+1}\}$, where

$$I_{i,j} = (4\alpha(i, j), 4\alpha(i, j) + 2), \qquad 1 \le j \le p_i$$

$$\bar{I}_{i,j} = (4\alpha(i, j) + 2, 4\alpha(i, j) + 4), \qquad 1 \le j \le p_i$$

$$J_{i,j} = (4\alpha(i, j) + 3, 4\alpha(i, j + 1)), \qquad 1 \le j \le p_i - 1$$

$$\bar{I}_{i,j} = (4\alpha(i, j) + 4, 4\alpha(i, j + 1) + 1), \qquad 1 \le j \le p_i - 1$$

and

$$J_{i,0} = (0, 4\alpha(i, 1))$$

$$\bar{J}_{i,0} = (1, 4\alpha(i, 1) + 1)$$

$$J_{i,p_i} = (4\alpha(i, p_i) + 3, 4m + 7)$$

$$\bar{J}_{i,p_i} = (4\alpha(i, p_i) + 4, 4m + 8)$$

$$J_{i,p_i+1} = (4m + 7, 4m + 8)$$

$$\bar{J}_{i,p_i+1} = (0, 1).$$

These $2p_i + 2$ 2-interval sets are called the *assignment family* of x_i, denoted Ω_i, i.e., $\Omega_i = \{A_{i,j} | 1 \le j \le p_i\} \cup \{B_{i,j} | 0 \le j \le p_i + 1\}$. See Fig. 11 for an illustration. We shall show later that the assignment family Ω_i thus constructed has the following property: let R_i denote the representative set of Ω_i. Then, if x_i is assigned the truth value 1, then $I_{i,j}$ and $J_{i,j}$ are chosen as representatives for $A_{i,j}$ and $B_{i,j}$, respectively, i.e., $R_i = \{I_{i,j} | 1 \le j \le p_i\} \cup \{J_{i,j} | 0 \le j \le p_i + 1\}$. Otherwise, if x_i is assigned the truth value 0, then $\bar{I}_{i,j}$ and $\bar{J}_{i,j}$ are chosen as representatives for $A_{i,j}$ and $B_{i,j}$, respectively, i.e., $R_i = \{\bar{I}_{i,j} | 1 \le j \le p_i\} \cup \{\bar{J}_{i,j} | 0 \le j \le p_i + 1\}$.

For each clause c_i ($1 \le i \le m$), we introduce an interval set of a single interval, i.e., $C_i = \{K_i\}$, where

$$K_i = (4i + 1, 4i + 2) \qquad \text{if } c_i \text{ is a negative clause}$$

and

$$K_i = (4i + 2, 4i + 3) \qquad \text{if } c_i \text{ is a positive clause.}$$

See Fig. 12 for an illustration. Note that the representative of C_i can only be K_i.

Finally, we choose the threshold density M in I to be the number of variables in B, which is n.

We have now completed the construction of the instance I of the minimum density representative problem. Since the collection of the interval sets in I is $\Omega = \Omega_1 \cup \Omega_2 \cup \cdots \cup \Omega_n \cup \{C_i, C_2, \cdots, C_m\}$, the total number of interval sets introduced is $\Sigma_{i=1}^{n} (2p_i + 2) + m = 2\Sigma_{i=1}^{n} p_i + 2n + m = 2n + 7m$. (Note that $\Sigma_{i=1}^{n} p_i = 3m$). Moreover, since each interval set contains at most 2 intervals, the total number of intervals introduced is at most $4n + 14m$. Therefore, the transformation can be carried out in polynomial time. We shall show that B is satisfiable if and only if Ω has a representative set of density no more than n.

Let δ be a truth assignment that satisfies B. If $\delta(x_i) = 1$, we choose $R_i = \{I_{i,j} | 1 \le j \le p_i\} \cup \{J_{i,j} | 0 \le j \le p_i + 1\}$ to be the representative set of Ω_i. If $\delta(x_i) = 0$, we choose $R_i = \{\bar{I}_{i,j} | 1 \le j \le p_i\} \cup \{\bar{J}_{i,j} | 0 \le j \le p_i + 1\}$ to be the representative set of Ω_i. Thus the representative set of Ω is $R = R_1 \cup R_2 \cup \cdots \cup R_n \cup \{K_1, K_2, \cdots, K_m\}$. It is easy to see that the density of R_i is 1 for all $1 \le i \le n$. Therefore, the density of $R_1 \cup R_2 \cup \cdots \cup R_n$ is at most n. Moreover, if the truth assignment of x_i satisfies clauses $c_{i_1}, c_{i_2}, \cdots, c_{i_s}$, then the density of

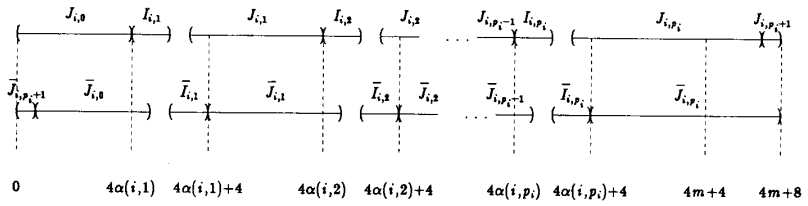


Fig. 11. Assignment family Ω_i of variable x_i.

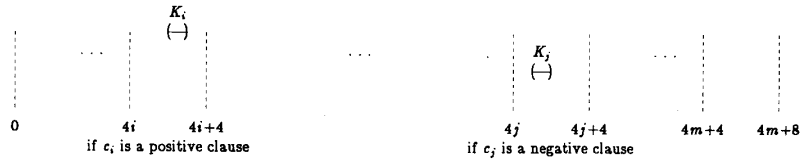


Fig. 12. Intervals corresponding to the clauses.

$R_i \cup \{K_{i_1}, K_{i_2}, \cdots, K_{i_s}\}$ is still 1. In this case, we say that $K_{i_1}, K_{i_2}, \cdots, K_{i_s}$ are *absorbed* by R_i. Since δ is a truth assignment that satisfies B, each clause c_j is satisfied by at least a truth assignment for some x_i. Thus each K_j is absorbed by some R_i. Therefore, the density of R is the same as the density of $R_1 \cup R_2 \cup \cdots \cup R_n$, which is at most n.

On the other hand, suppose that Ω has a representative set R of density no more than n. We construct the truth assignment δ in the following way. Let R_i be the representative set of Ω_i. Since the density of each R_i at $4\alpha(i, j) + 0.5$ is at least 1, R_i can not contain both $I_{i,j}$ and $\bar{J}_{i,j-1}$. Otherwise, the density of R_i at $4\alpha(i, j) + 0.5$ is 2, and the density of R at this point is at least $n + 1$. Similarly, since the density of each R_i at $4\alpha(i, j) + 3.5$ is at least 1, R_i can not contain both $\bar{I}_{i,j}$ and $J_{i,j}$. Therefore, either $R_i = \{I_{i,j} | 1 \le j \le p_i\} \cup \{J_{i,j} | 0 \le j \le p_i + 1\}$ or $R_i = \{\bar{I}_{i,j} | 1 \le j \le p_i\} \cup \{\bar{J}_{i,j} | 0 \le j \le p_i + 1\}$. In the former case, let $\delta(x_i) = 1$ and $\delta(\bar{x}_i) = 0$. In the latter case, let $\delta(x_i) = 0$ and $\delta(\bar{x}_i) = 1$. Moreover, it is easy to see that the density of $R = R_i \cup R_2 \cup \cdots \cup R_n \cup \{K_1, K_2, \cdots, K_m\}$ and the density of $R_1 \cup R_2 \cup \cdots \cup R_n$ are both n. Therefore, each K_j is absorbed by some R_i. Furthermore, given that K_j is absorbed by R_i, it is easy to verify that if c_j is a positive clause, then x_i appears in c_j and $R_i = \{I_{i,j} | 1 \le j \le p_i\} \cup \{J_{i,j} | 0 \le j \le p_i + 1\}$, and if c_j is a negative clause, then $\bar{x}_i$ appears in c_j and $R_i = \{\bar{I}_{i,j} | 1 \le j \le p_i\} \cup \{\bar{J}_{i,j} | 0 \le j \le p_i + 1\}$. In the former case, clause c_j is satisfied by the truth assignment $\delta(x_i)$. In the later case, clause c_j is satisfied by the truth assignment $\delta(\bar{x}_i)$. Therefore, δ is a truth assignment that satisfies B. □

Combining Lemma 4-1 and Lemma 4-2, we can conclude the following theorem.

Theorem 4-1: MDSFP is *NP*-complete.

Proof: Let $G = \langle V, E, w \rangle$ be a weighted graph and D the threshold in an instance I of MDSFP. For any subset N of E, we can check if N constitutes a spanning forest of E in $O(|V| + |E|)$ time. Moreover, we can compute the density of the interval set associated with N in $O(|E| \log |E|)$ time to see whether it is larger than D. Therefore, we can determine if N is a valid solution of I in polynomial time. It follows that MDSFP is in *NP*. Moreover, according to Lemma 4-1, we can reduce the minimum density representative problem to MDSFP in polynomial time. According to Lemma 4-2, the minimum density representative problem is *NP*-complete. Thus we conclude that MDSFP is also *NP*-complete. □

We present now an efficient heuristic algorithm for solving the net segment selection problem. First, we notice that the connection graph can be simplified. For example, in Fig. 9 the edge with weight [8, 11] connecting h_1 and h_2 is redundant. If this edge is ever chosen to connect h_1 and h_2, it can always be replaced by the edge with weight [8, 10] without increasing the channel density, because [8, 10] is subinterval of [8, 11]. To be more precise, we say that two terminals $n \cdot i$ and $n \cdot j$ of net n are *adjacent* if there is no other terminal of net n located between column i and column j (assume that $i < j$). For an instance I of the net segment selection problem, the *simplified connection graph* $SCG(I)$ is a subgraph of $CG(I)$ such that an edge of net n with weight $[i, j]$ in $CG(I)$ is also in $SCG(I)$ if and only if the two terminals $n \cdot i$ and $n \cdot j$ of net n are adjacent. For example, the simplified connection graph of the connection graph in Fig. 9 is shown in Fig. 13. We can prove the following lemma.

Lemma 4-3: $SCG(I)$ contains a minimum density spanning forest of $CG(I)$. Moreover, the number of edges in $SCG(I)$ is at most linear in c, where c is the number of columns in I.

Proof: Let T be a minimum density spanning forest of $CG(I)$. If all the edges in T are also in $SCG(I)$, clearly, $SCG(I)$ contains a minimum density spanning forest. Otherwise, let (p_1, p_2) be an edge in T but not in $SCG(I)$. Suppose that $q_1, q_2, \cdots, q_l$ are the terminals of the same net between p_1 and p_2. It is easy to show that we can replace (p_1, p_2) by one of the edges $(p_1, q_1), (q_1, q_2), \cdots, (q_l, p_2)$ to obtain another spanning forest with-

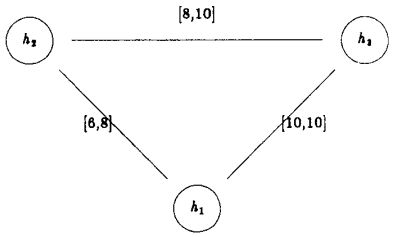


Fig. 13. The simplified connection graph for the example in Fig. 9.

out increasing the density of the forest. We apply such a replacement operation to every edge in $T - SCG(I)$. It is easy to see that we shall eventually end up with a minimum density spanning forest in $SCG(I)$.

Assume that I contains k hyperterminals and the degree of hyperterminal h_i is t_i $(1 \leq i \leq k)$. Assume that hyperterminal h_i belongs to net n. Let v_i be the vertex in $SCG(I)$ that represents the hyperterminal h_i. Let $d(v_i)$ denote the degree of v_i. We claim that $d(v_i)$ is upper bounded by $2t_i$, since each terminal p in h_i contributes at most two edges that are incident upon v_i in $SCG(I)$; one corresponds to the connection from p to a terminal that is closest to its left and belongs to another hyperterminal of net n, and the other corresponds to the connection from p to a terminal that is closest to its right and belongs to another hyperterminal of net n. Note that $\Sigma_{i=1}^{k} t_i \leq 2c$. Therefore, the number of edges in $SCG(I)$ is bounded by

$$\frac{1}{2}\sum_{i=1}^{k} d(v_i) \leq \frac{1}{2}\sum_{i=1}^{k} 2t_i = \sum_{i=1}^{k} t_i \leq 2c. \qquad \square$$

Our heuristic algorithm works as follows: given an instance I of the net segment selection problem, we construct $SCG(I)$. Let N denote the set of edges of $SCG(I)$. For each edge e in N, we compute the *relative density* of e, $RD(e)$, which is defined to be $d(e)/d(N)$, where $d(e)$ is the density of the set of intervals which intersect with the interval $w(e)$, and $d(N)$ is the density of the interval set $\{w(e) \mid e \in N\}$. The relative density of an edge measures the degree of congestion over the interval associated with the edge. The rest of the algorithm is a loop which repeatedly removes edges from N until N is a spanning forest. First, we determine the set of noncritical edges in N, denoted $X(N)$. An edge is *noncritical* if the removal of this edge from $SCG(I)$ does not increase the number of connected components in $SCG(I)$. We can identify all the noncritical edges by generating all the biconnected components of $SCG(I)$, because an edge is noncritical if and only if it belongs to a biconnected component of size larger than 2. A depth-first search algorithm can be used to generate all the biconnected components in linear time. (See [17] for a detailed description of the algorithm.) Next, we choose from $X(N)$ an edge $\hat{e}$ which has the maximum relative density and delete $\hat{e}$ from N. (We break ties by comparing edge lengths and choosing the longer one.) Clearly, selection and deletion of edge $\hat{e}$ can be done in linear time, since N contains at most a linear number of edges. After we delete the edge $\hat{e}$, we update $d(N)$ and recompute $d(e)$ for each affected edge e. In the worst case, there are $\Omega(c)$ edges affected. A straightforward computation takes $\Omega(c)$ time to recompute the value $d(e)$ for an edge e. Therefore, updating all the $d(e)$'s could take $\Omega(c^2)$ time. However, since the basic operation is to compute the density of a subset of intervals, by maintaining a segment tree [15] throughout the execution of the algorithm, we can update $d(e)$ in $O(\log c)$ time after the deletion of an edge. (See [15] for details in using segment trees in the solution of interval related problems.) Hence, updating of all the $d(e)$'s can be done in $O(c \log c)$ time. We repeat such a deletion procedure until N becomes a spanning forest of $SCG(I)$, (or, equivalently, until $X(N)$ is empty). We call this algorithm the *iterative deletion algorithm*. Our heuristic algorithm has two advantages. First, because we begin with all possible net segments, we know where the most congested area is. Thus our deletion process will help to reduce the channel density. Second, we can show that for any given instance I, there is always an edge deletion sequence for the edges in $SCG(I)$ that leads to a minimum density spanning forest of $CG(I)$, thus yielding an optimal net segment selection for I. However, our heuristic algorithm might not always produce such an optimal deletion sequence (note that finding an optimal solution to the net selection problem is *NP*-hard). The experimental results in Section VI show that the iterative deletion algorithm performs very well in general. From Lemma 4-3, it is not difficult to conclude the following.

Theorem 4-2: The iterative deletion algorithm connects all the hyperterminals in each net after no more than $O(c)$ steps of edge deletion, where c is the number of columns in the channel.

V. Overall Complexity

Fig. 14 shows a summary of the over-the-cell channel routing algorithm. Let c be the number of columns in a channel. We have the following theorem:

Theorem 5-1: The time complexity of our over-the-cell channel routing algorithm is $O(c^2 \log c)$.

Proof: According to Theorem 3-2, line 1 and line 2 take $O(c^2)$ time. Line 3 can be completed in $O(c \log c)$ time using an optimal sorting algorithm [17]. According to Theorem 4-2, line 4 consists of $O(c)$ iterations of edge deletion. In each of those iterations, identifying all the noncritical edges takes $O(c)$ time. Selecting and deleting the edge with the maximum relative density takes $O(c)$ time. Moreover, by maintaining a segment tree [15], each density update can be done in $O(\log c)$ time. Thus updating all the relative densities takes $O(c \log c)$ time. Therefore, each iteration of edge deletion takes $O(c \log c)$ time. Thus, line 4 takes $O(c^2 \log c)$ time. Line 5 takes $O(c)$ time since the greedy channel routing algorithm does a linear scan of the columns in the channel. Therefore, the total time complexity is

$$O(c^2 c \log c + c^2 \log c + c) = O(c^2 \log c). \qquad \square$$

Algorithm Over-the-Cell Channel Routing;
Input Two lists of terminals on the top and the bottom edge of the channel;
Output Routing solutions in the channel and over the upper and lower cells ;
Begin
1. Route over the upper cell row;
2. Route over the lower cell row;
3. Construct the simplified connection graph *SCG*
4. Find an approximated minimum density spanning forest of *SCG* using the iterative deletion algorithm;
5. Use the greedy channel routing algorithm to connect the terminals in the channel as specified by the obtained spanning forest.

End.

Fig. 14. Summary of over-the-cell channel routing algorithm.

TABLE I
EXPERIMENTAL RESULTS OF OUR OVER-THE-CELL CHANNEL ROUTER

Ex.	density	density over the lower cells	density over the upper cells	density within the channel	channel width	running time
1	12	3	4	9	10	2.0 sec
3a	15	6	3	12	12	2.9 sec
3b	17	5	2	13	13	3.5 sec
3c	18	4	3	14	15	4.5 sec
4b	20	4	5	16	16	9.7 sec
5	20	3	4	14	14	4.9 sec
De	19	7	8	16	17	25.8 sec

TABLE II
COMPARISONS WITH THE OPTIMAL TWO-LAYER ROUTING SOLUTIONS AND THE ROUTING SOLUTIONS BY THE OVER-THE-CELL CHANNEL ROUTER IN [19]

Ex.	optimal 2L solutions # tracks	solutions in [19] # tracks	our solutions # tracks	improvement on opt 2L solutions	improvement on solutions in [19]
1	12	10	10	16.7%	0.0%
3a	15	15	12	20.0%	20.0%
3b	17	16	13	23.5%	25.0%
3c	18	16	15	16.7%	6.3%
4b	20	16	16	20.0%	0.0%
5	20	14	14	30.0%	0.0%
De	19	20	17	10.5%	15.0%
Ave.				19.6%	9.5%

VI. EXPERIMENTAL RESULTS

We implemented an over-the-cell channel router in Pascal and executed it under Unix 4.3 BSD on a Pyramid computer. Table I shows some of our experimental results. All the examples were taken from Yoshimura and Kuh's paper [22]. The famous Deutsch's difficult example is labeled *De*. Note that the final channel widths of our routing solutions are always several tracks fewer than the density d of the original problem. Also, running times for all the examples are very short. Note also that for some examples the final channel width is larger than the resulting channel density. This is due to the fact that the greedy channel router does not always produce an optimal channel routing solution. It indicates that using a more sophisticated channel router we might be able to obtain even better routing solutions. Note that for some very dense routing examples, such as Deutsch's difficult example, the densities of over-the-cell routings can be quite high. However, in practice, there is a fixed number of tracks available over the cells, which is determined by the maximum cell height. This problem shall be discussed in the next section.

Table II shows a comparison of our routing solutions with the optimal two-layer routing solutions and with the routing solutions produced by the over-the-cell router in [19] without using diffusion underpass (which is equivalent to our routing model). It was observed that our routing solutions are consistently better than the optimal two-layer routing solutions. The average reduction on the number of tracks in channel is 19.6 percent. Our router also outperformed the over-the-cell router in [19]. On the average, our solutions use 9.6-percent fewer tracks in channel than the solutions in [19]. In particular, for the Deutsch's difficult example, our routing solution (see Fig. 15) uses 2 fewer tracks in the channel than the optimal

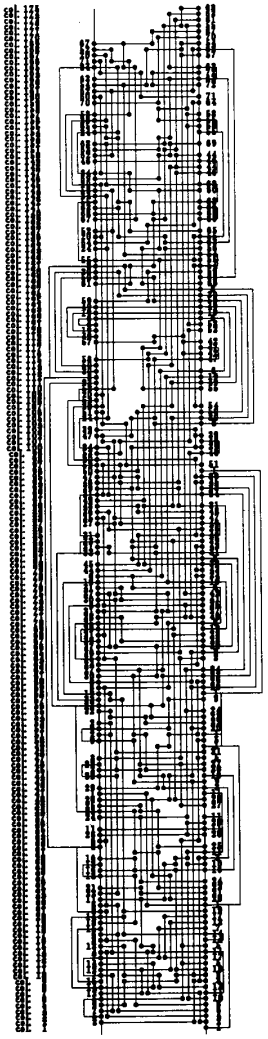


Fig. 15. Our solution to Deutsch's difficult example.

two-layer solution, and uses 3 fewer tracks in the channel than the solution in [19]. (Apparently, the channel router used in [19] is not a very powerful one. For Deutsch's difficult example, after over-the-cell routing, their routing solution still used 20 tracks in channel. Unfortunately, they did not specify the channel density after over-the-cell routing. We conjecture that the results in [19] can be improved by using a more powerful channel router.) We were unable to compare our router with the over-the-cell routers in [7] and [13] due to a lack of common test examples. No experimental results were reported in [12].

VII. Conclusions and Remarks

In this paper, we presented results on the complexity of each step involved in over-the-cell routing. Combining these results, we designed an over-the-cell channel router which runs in $O(c^2 \log c)$ time, where c is the number of columns in the routing problem. Significant channel area reductions were achieved using our over-the-cell router. Our approach can easily be generalized to three layer channel routing simply by using a three layer channel router (see, for example, [3]) in our third step.

In the first step of our algorithm, we assume that there are unlimited number of tracks available over the cells. A more realistic model would assume a fixed number of tracks over the cells as suggested in [7]. We can show that the first step can still be solved optimally with such restriction, but in $O(c^3)$ time instead of in $O(c^2)$ time. This result and discussions of other physical design restrictions will be presented in another paper [5]. The minimum density spanning forest problem formulated in Section IV also plays an important role in some other layout design problems. A further study to find more efficient and effective heuristic algorithms and to find optimal solutions to its subproblems is in progress.

Acknowledgment

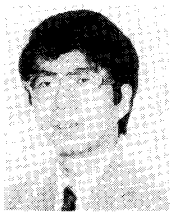
The authors thank Prof. D. F. Wong and Prof. H. W. Leong for their assistance. They also thank Dr. Deutsch for his helpful discussions. They are grateful to the reviewers for their valuable comments and suggestions.

References

[1] T. Asano, T. Asano, and H. Imai, "Partitioning a polygonal region into trapezoids," *J. ACM*, vol. 33, pp. 290–312, 1986.
[2] M. Burstein and R. Pelavin, "Hierarchical channel router," *Integration, VLSI J.*, vol. 1, pp. 21–38, 1983.
[3] J. Cong, D. F. Wong, and C. L. Liu, "A new approach to the three layer channel routing," in *Proc. ICCAD-87*, pp. 378–381, Nov. 1987.
[4] J. Cong and C. L. Liu, "Over-the-cell channel routing," in *Proc. Int. Conf. Computer-Aided Design*, pp. 80–83, 1988.
[5] J. Cong, B. Preas, and C. L. Liu, "Over-the-cell routing for standard cells," Manuscript, 1989.
[6] D. N. Deutsch, "A dogleg channel router," in *Proc. 13th Design Automation Conf.*, pp. 425–433, 1976.
[7] D. N. Deutsch and P. Glick, "An over-the-cell router," in *Proc. 17th IEEE/ACM Design Automation Conf.*, pp. 32–39, 1980.
[8] S. Even and A. Itai, "Queues, stacked, and graphs," in *Theory of Machines and Computations*, A. Kohavi, ed. New York: Academic, 1971, pp. 71–86.
[9] M. R. Garey and D. S. Johnson, *Computer and Intractability: A Guide to the Theory of NP-Completeness*. San Francisco, CA: Freeman, 1979.
[10] F. Gavril, "Algorithms for a maximum clique and a maximum independent set of a circle graph," *Networks*, vol. 3, pp. 261–273, 1973.
[11] E. M. Gold, "Complexity of automaton identification from given data," unpublished manuscript, 1974.
[12] G. Gudmundsson and S. Ntafos, "Channel routing with superterminals," in *Proc. 25th Allerton Conf. on Computing, Control and Communication*, pp. 375–376, 1987.
[13] H. E. Krohn, "An over-the-cell gate array channel router," in *Proc. 20th IEEE/ACM Design Automation Conf.*, pp. 665–670, 1983.
[14] H. W. Leong and C. L. Liu, "A new channel routing problem," in *Proc. 20th IEEE/ACM Design Automation Conf.*, pp. 584–590, 1983.
[15] F. P. Preparata and M. I. Shamos, *Computational Geometry*. New York: Springer-Verlag, 1985.
[16] 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, 1985.
[17] E. Reingold, J. Nievergelt, and N. Deo, *Combinatorial Algorithms: Theory and Practice*. Englewood Cliffs, NJ: Prentice-Hall, 1977.
[18] R. L. Rivest and C. M. Fiduccia, "A 'greedy' channel router," in *Proc. 19th Design Automation Conf.*, pp. 418–424, 1982.
[19] Y. Shiraishi and Y. Sakemi, "A permeation router," *IEEE Trans. Computer-Aided Design*, vol. CAD-6, pp. 462–471, May 1987.

[20] K. J. Supowit, "Finding a maximum plannar subset of a set of nets in a channel," *IEEE Trans. Computer-Aided Design*, vol. CAD-6, pp. 93–94, Jan. 1987.
[21] T. G. Szymanski, "Dogleg channel routing is *NP*-complete," *IEEE Trans. Computer-Aided Design*, vol. CAD-4, pp. 31–41, 1985.
[22] T. Yoshimura and E. S. Kuh, "Efficient algorithms for channel routing," *IEEE Trans. Computer-Aided Design*, vol. CAD-1, pp. 25–35, Jan. 1982.

*



Jingsheng Jason Cong (M'00) received the B.S. degree in computer science from Peking University in 1985, and the M.S. degree in computer science from University of Illinois at Urbana-Champaign in 1987. Currently, he is a Ph.D. candidate in computer science at University of Illinois at Urbana-Champaign.

Since 1986, he has been a research assistant in Computer Science Department of University of Illinois. He worked at Xerox Palo Alto Research Center in the summer of 1987. He worked at National Semiconductor Corporation in the summer of 1988. His research interests include design and analysis of efficient combinatorial and geometric algorithms, computer-aided design of integrated circuits.

Mr. Cong received the Best Graduate Award from Peking University in 1985. He received Ross J. Martin Award for outstanding research from University of Illinois at Urbana-Champaign in 1989. He is a member of ACM.

*

C. L. Liu (M'00–SM'00–F'00) received the B.Sc. degree from the National Cheng Kung University, Taiwan, in 1956, and the M.S., E.E. and Sc.D. degrees in electrical engineering from the Massachusetts Institute of Technology in 1960 and 1962, respectively.

He is currently a Professor of Computer Science at the University of Illinois at Urbana-Champaign. His research interests include design and analysis of algorithms, computer-aided design of integrated circuits, and combinatorial mathematics.

Dr. Liu currently is a member of the editorial boards of IEEE TRANSACTIONS ON COMPUTERS, *Graphs and Combinatorics*, and *Algorithmica*. He is also an editorial advisor for the Computer Science Series published by the World Scientific Publishing Company.