NOVEL ROUTING SCHEMES FOR IC LAYOUT PART I: TWO-LAYER CHANNEL ROUTING

by

Deborah Wang and E. S. Kuh

Memorandum No. UCB/ERL M90/101

2 November 1990

NOVEL ROUTING SCHEMES FOR IC LAYOUT PART I: TWO-LAYER CHANNEL ROUTING

by

Deborah Wang and E. S. Kuh

Memorandum No. UCB/ERL M90/101

2 November 1990

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

NOVEL ROUTING SCHEMES FOR IC LAYOUT PART I: TWO-LAYER CHANNEL ROUTING

by

Deborah Wang and E. S. Kuh

Memorandum No. UCB/ERL M90/101

2 November 1990

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

Novel Routing Schemes for IC Layout
Part I: Two-Layer Channel Routing

ABSTRACT

We present new channel routing algorithms and theory that consider the characteristic of net crossings. The routing strategy is based on parallel bubble sorting and river routing techniques. A function named "POTENTIAL", can be evaluated to indicate the required channel height for a given channel without actually carrying out the routing steps. Non-Manhattan wires as well as overlapping wires are introduced. Preliminary results show that a class of channel routing problems can be routed in height less than the Manhattan density.

1 Introduction

Channel routing is one of the most important phases of physical design of VLSI chips as well as PC boards. Many breakthroughs [26, 9, 20, 7, 5, 2, 22, 10, 18, 24] in channel routing theory and algorithms have been reported. The classical and well-studied model for channel routing is the directional Manhattan routing model: one layer is used exclusively for vertical wires , another is used for horizontal wires, and vias are introduced for each layer change. The density of a channel gives a lower bound on the channel height. However, routing in density is frequently difficult to achieve due to the complication introduced by the vertical constraints. It is known that the restrictive routing problem with cyclic vertical constraints does not have a solution [6] and in general, the Manhattan problem in the presence of vertical constraints is NP-complete [13, 25].

In this paper, we propose new methods which perform successfully on practical channels and attempt to provide accurate and formal analysis on the quality of the solutions. The approach was developed with an interest in the intrinsic complexity of channel routing problems and their combinatorial structure. Two new routing models, the *mini-swap* model and the *overlap* model are introduced. Non-Manhattan geometry as well as rectilinear wires are used. The routing strategy is based on parallel sorting and river routing techniques and is very different from the Manhattan approach. In particular, vertical constraints no longer exist. Furthermore, the solution produced by either model consists of a minimal set of net crossings which leads to a small number of vias. To characterize and to evaluate the performance of each model, a function called "POTENTIAL" is used. Intuitively, the "POTENTIAL" function measures the degree of difficulty for a given channel routing problem. Based on the performance analysis, we attempt to combine the strength of both models to form a new *hybrid* router. The routing solution produced by our router has unambiguously layer assignment and is design-rule correct. Finally, extensions to handle multi-terminal nets and multi-layers will be discussed.

The remainder of this paper is organized as follows. In Section 2, we present the algorithm and theory associated with the *mini-swap* model. The *overlap* routing model is introduced and analyzed in Section 3. Section 4 describes the new hybrid router and its performance. Routing results are presented in Section 5. Section 6 contains concluding remarks and discussion on future work.

2 The Mini-Swap Model

The mini-swap model involves diagonal wires oriented in the +45° and -45° angles as well as vertical wires. The routing strategy is based on parallel sorting techniques [1]. To characterize the channel routing problem under this model, a function called "POTENTIAL", originally defined for the parallel bubble sorting [21], is used. Intuitively, the "POTENTIAL" function measures the degree of difficulty for a given channel routing problem. We can mechanically evaluate the POTENTIAL function to compute the exact channel height required to route the channel under the *mini-swap* model without actually carrying out the routing steps. This feature makes this model very attractive since knowing the required channel height in the placement stage is valuable.

Furthermore, we prove the solution is optimal under the mini-swap model. The routing solution produced by applying the parallel bubble sorting technique can be unambiguously wired using two interconnect layers. Unlike other Non-Manhattan routers [17] [16] [23] that magnify both column and track spacing by a factor of $\sqrt{2}$ which implies the channel area is doubled, this router ensures that the solution is design rule correct and does not magnify spacing in either direction.

2.1 The Problem

A channel is a pair of vectors of nonnegative integers - TOP and BOT - of the same dimension.

TOP = t(1), t(2), ... t(n)

BOT = b(1), b(2), ... b(n)

We assume that these numbers are the labels of grid points located along the top and bottom edge of a rectangle. Points having the same positive label have to be interconnected, i.e. they define nets. A 100% routing completion is required and the objective is to minimize the channel area and the number of vias.

Definition 1 *A channel is dense iff* $t(i) \neq 0$ *and* $b(i) \neq 0$ *for all i, that is, every grid point on the top and bottom boundaries is occupied by a terminal.*

Definition 2 *A non-trivial 2-terminal net is a net that has exactly two terminals, one on the top and another on the bottom.*

Let { 1, 2, ... N } denote the set of nets. Then in a dense 2-terminal net channel, TOP and BOT are permutations of { 1, 2, ... N }. Without loss of generality, we may assume that nets are arbitrarily ordered on the bottom and are naturally ordered on the top. This is stated as:

Definition 3 *A dense 2-terminal net (D2TN) channel routing problem is specified by:*

TOP = 1, 2, 3, ... N

BOT = a permutation of { 1, 2, ... N }

We denote the top and bottom terminals of net i by (i, q_i).

2.2 Definition of the Mini-Swap Model

The basic idea of the new model is to swap a pair of neighboring nets by two wires, one in the +45° direction, another in the -45° direction (Figure 1). We call such a swap a "mini-swap".

Routing of a D2TN channel can be viewed as a vertical stack of steps. A step is a unit-high horizontal strip which lies between two tracks. In each step, a set of mini-swaps is performed simultaneously. If a net does not change position in a step, it simply propagates to the next track by a unit vertical wire. An example is shown in Figure 2.

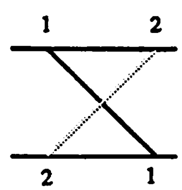


Figure 1. A mini-swap

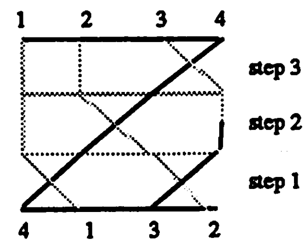


Figure 2. In each step, a set of mini-swaps occur

A solution for the D2TN channel routing problem in this model can be constructed in a bottom-to-top step-by-step fashion . The final channel height is equal to the number of steps required. Clearly, a D2TN channel routing problem can have many possible solutions under the mini-swap model. Figure 3 shows 2 solutions to an instance of a D2TN channel. To measure the performance of a routing strategy, we need to define the optimum solutions under the mini-swap model.

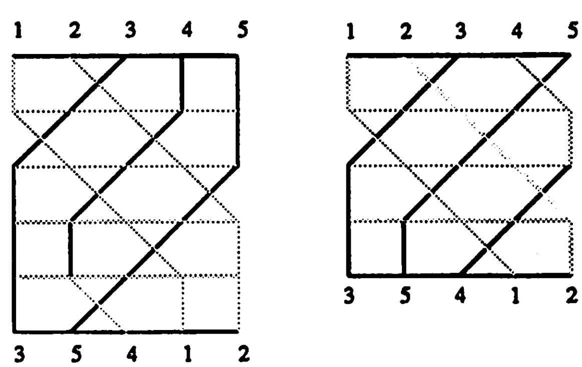


Figure 3. Two realizations of the same channel

Definition 4 *The optimal solution of a two-terminal net channel routing problem under the mini-swap model is one that has*

(a) the minimum channel height and

(b) the shortest total wire length

In search of the optimum solution, the router must determine which pairs of nets to swap in each step so that the final solution is optimum. In the example shown in figure 3, mini-swaps were performed only on pairs of nets that were not ordered. Is it possible that in search of the optimum solution we may want to swap pairs that are already ordered? The answer to this question is given in theorem 1 which states a necessary optimality condition. Before we prove the theorem, we need the following definition.

Definition 5 *A pair of nets is said to be "planar" if it is in the natural order on the bottom. Otherwise, it is said to be "intersecting".*

Theorem 1 *A solution is optimal under the mini-swap model if it has the minimum number of tracks and properties 1 and 2 hold:*
(property 1) planar pairs do not intersect and
(property 2) intersecting pairs intersect only once

[proof] For the sake of contradiction, assume a solution S, has the minimum number of

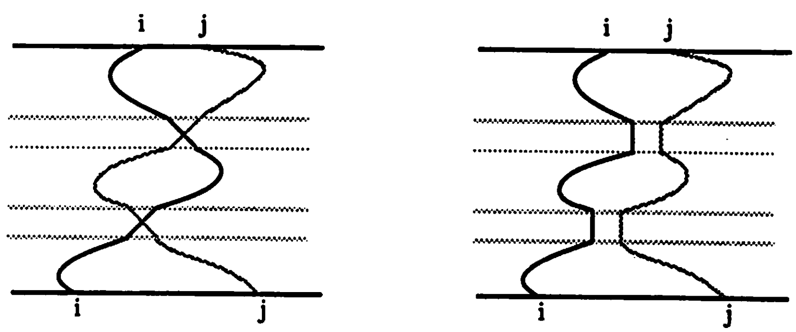


Figure 4. Solution S Solution S'

tracks and some planar nets intersect twice. Then there exists a new solution S' (see Figure 4), which has the same number of tracks and no intersecting planar nets.

Clearly S' has shorter total wire length than S, which implies that S is not optimal.

The proof for property 2 is similar. QED.

In other words, a router never needs to swap a pair of nets that are already in the natural order to obtain an optimum solution. This implies that the set of mini-swaps that leads to the solution corresponds to a minimal set of net crossings.

2.3 The Routing Strategy

The basic routing strategy is to construct a wiring path for each net in a bottom-to-top step-by-step manner. In each step, the router selects a set of intersecting pairs to swap. Since decisions made in the earlier steps can affect choices in later steps, choosing an appropriate set of mini-swaps to perform in each step is crucial to the quality of the final solution.

We now describe a routing algorithm based on the parallel bubble sort [1]. In the first step, nets located at odd grid points are compared with the net on their right. If the pair is an planar pair, the pair does not switch positions; otherwise, a mini-swap is performed. The second step is identical to the first one, except this time nets located at even grid points are compared with the net on their right. These two steps are repeatedly performed in this order. The algorithm stops when for two consecutive steps, no pairs of nets switch places. Once the algorithm terminates, the nets are ordered in the natural order. The algorithm must terminate since each pair of nets can switch places only once. Let us demonstrate the algorithm on the example shown in Figure 5.

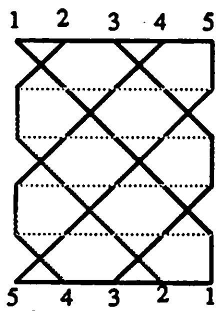


Figure 5. Routing by odd-even transposition

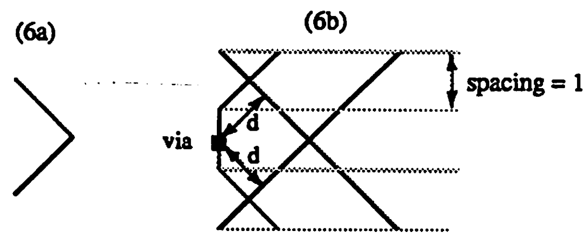


Figure (6a) An impossible situation
(6b) A viaa is placed at the center of a vertical segment to ensure d > 1.

Layer assignment for the wiring path is trivial. Layer 1 is assigned to wires oriented in the +45° angle; layer 2 is assigned to wires oriented in the -45° angle. Vertical wires can be assigned to any layer since they do not cross any other wires. Vias are introduced for layer changes between a +45° wire and -45° wire. Due to the odd-even transposition procedure, the situation shown in Figure 6a could never happen: the +45° wire of a net always connects to a -45° wire through a vertical segment as shown in Figure 6b. We place a via at the midpoint of the vertical segment. This ensures that the wires satisfy the design rule so that there is no need to magnify either the column spacing or the row spacing by $\sqrt{2}$.

Formally, let us denote the permutation of nets on track t by:

$A^{(t)} = (a^{(t)}(1), a^{(t)}(2), \ldots\ldots a^{(t)}(N))$

Then, $A^{(0)} = (a^{(0)}(1), a^{(0)}(2), \ldots\ldots a^{(0)}(N))$, where $a^{(0)}(i) = b(i)$, for all i. We also assume there is an infinite number of auxiliary nets, represented by .... a(-1), a(0), and a(N+1), a(N+2),...., where for all $i \geq 0$ and $t > 0, a^{(t)}(i) = -\infty$ and for all $i > N$ and $t > 0$, $a^{(t)}(i) = +\infty$. These auxiliary nets do not disturb the routing of regular nets at any time. For each track $t \geq 0$ and each integer i, the net, $a^{(t)}(i)$, at the ith grid point at track t is given by:

if (i + t) is even , $a^{(t)}(i) = \min(a^{(t-1)}(i), a^{(t-1)}(i+1))$

if (i + t) is odd , $a^{(t)}(i) = \max(a^{(t-1)}(i-1), a^{(t-1)}(i))$

The dynamic behavior of this routing scheme is the same as that of the parallel bubble sorter realized on a two-way infinite linear array. If, for every i, $a^{(t)}(i) \leq a^{(t)}(i+1)$, then $A^{(t)}$ is said to be sorted. The computing time of the parallel bubble system is the smallest t such that $A^{(t)}$ is sorted. A router similar to ours has been developed independently by [8]. The main difference is that their method is based on the sequential bubble sort.

2.4 The POTENTIAL function

Since the routing scheme is a direct adaptation of the parallel bubble sort, a characterization of the bubble system can also be used to analyze the routing process. In particular, we are interested in the computing time of a parallel bubble system which corresponds to the required height of the D2TN channel routing problem. A function called "POTENTIAL", first introduced by [21], proved to be very useful in evaluating the computing time of the bubble system. This section defines the POTENTIAL function and reviews the theorem proved by [21]. Before we introduce the POTENTIAL function, we need to define the following four terms.

Definition 6 *For each (i,j,t), where $1 \le i, j \le N$, and $t \ge 0$, and the set of nets, S, we define:*

- *ORDER(i,j,t,S) is the number of indices $p \in S$ such that $i < p \le j$ and $a^{(t)}(i) \le a^{(t)}(p)$, or such that $j \le p < i$ and $a^{(t)}(p) \le a^{(t)}(i)$*
- *NOTORDER(i,j,t,S) is the number of indices $p \in S$ such that $i < p \le j$ and $a^{(t)}(p) \le a^{(t)}(i)$, or such that $j \le p < i$ and $a^{(t)}(i) \le a^{(t)}(p)$*
- *MAXLT(i,t,S) = max(0 $\cup$ { ORDER(i,p,t,S) - NOTORDER(i,p,t,S) + 1 | $p \in S, p < i$ and $a^{(t)}(i) \le a^{(t)}(p)$ })*
- *MAXGT(i,t,S) = max(0 $\cup$ { ORDER(i,p,t,S) - NOTORDER(i,p,t,S) + 1 | $p \in S, i < p$ and $a^{(t)}(p) \le a^{(t)}(i)$ })*

We are now ready to define the POTENTIAL function.

Definition 7 *For any position indexing $i \le N$, and any $t \ge 0$, the function POTENTIAL(i,t) is defined as:*

- *When NOTORDER(i,1,t,S) = 0,*
  POTENTIAL(i,t) = NOTORDER(i,N,t,S) + MAXGT(i,t,S)
- *When NOTORDER(i,N,t,S) = 0,*
  POTENTIAL(i,t) = NOTORDER(i,1,t,S) + MAXLT(i,t,S)
- *When NOTORDER(i,N,t,S) $\neq$ 0 and NOTORDER(i,N,t,S) $\neq$ 0,*
  POTENTIAL(i,t) = NOTORDER(i,1,t,S) + NOTORDER(i,N,t,S) +
  max(1, MAXLT(i,t,S), MAXGT(i,t,S))

The POTENTIAL function for the entire bubble system is defined by:
POTENTIAL(t) = max(POTENTIAL(i,t) $| 1 \leq i \leq N$ *)*

Note that when both NOTORDER(i,1,t,S) = NOTORDER(i,N,t,S) = 0, POTENTIAL(i,t) = 0 from either (1) or (2). Hence the function is well defined. From the above definition, it is clear that POTENTIAL(i,t) = 0 if and only if NOTORDER(i,1,t,S) = NOTORDER(i,N,t,S) = 0. An immediate consequence of this fact is:

Fact 1 If POTENTIAL(t) = 0, then $A^{(t)}$ is sorted.

Fact 2 If POTENTIAL(i,t) = 0, then POTENTIAL(i,r) = 0 for all r > t

We now state the main theorem proved by [21].

Theorem 2 *If* $t \geq 1$ *and* $A^{(t)}$ *is not sorted, then*
POTENTIAL(t+1) = POTENTIAL(t) - 1

Corollary 2.1 *The computing time of the bubble system is POTENTIAL(0) or POTENTIAL(0)+1.*

The salient feature of the bubble system is that the POTENTIAL value consistently decreases by 1 per step. In other words, when POTENTIAL(0) = k, the required height of the D2TN channel is k or k+1. The value of POTENTIAL(0) is defined solely by the initial permutation $A^{(0)}$, without referring to the intermediate configurations $A^{(t)}$ for $t > 0$, that is we can mechanically and precisely evaluate the POTENTIAL function to compute the number of tracks require to route the D2TN channel without actually carrying out the routing steps. Another feature is that the decision of whether to swap a pair or not is local and does not depend on the locations of the rest of the nets. This feature makes this algorithm attractive in a parallel mode of operation. We also observe that the wiring path for each net is monotonic in the vertical direction.

2.5 The Main Theorem

The parallel bubble sort provides a simple routing strategy to produce a solution in the mini-swap model. Given the initial value of the POTENTIAL function, we saw the bubble scheme consistently decrease the POTENTIAL value by one until the target value zero was reached. Do better algorithms exist for which the POTENTIAL value decreases rapidly , say, by more than 1 per step ? For the example shown in Figure 7, the first step of the parallel bubble algorithm would switch the 2 pairs shown in 7a. But as many as 6 pairs could have been switched (7b). In the following theorem, we prove a sufficient optimality condition for any algorithm under the mini-swap model. See Appendix A and B for detail of the proof.

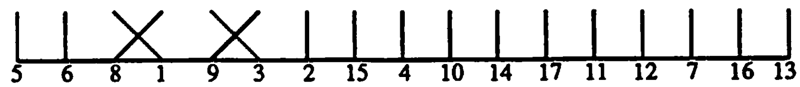


Figure (7a)

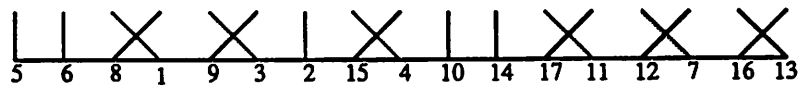


Figure (7b)

Theorem 3 *Potential value decreases at most by 1 per net per step under the mini-swap model.*

The following theorem gives a lower bound on the POTENTIAL function.

Theorem 4 *For each net i, we define the displacement(i) as the horizontal distance between its two terminals. Then* $POTENTIAL(0) \geq max(\{displacement(i) \mid 1 \leq i \leq N\})$.

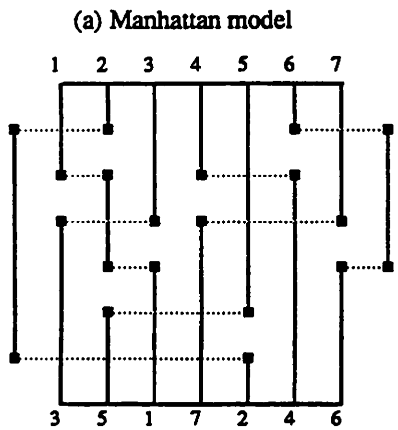
2.6 Results and Comparison

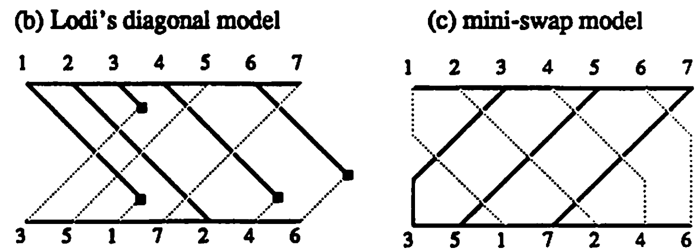
Routing results of two D2TN channels are shown in Figure 8 and 9. Compared to the Manhattan routers, our router has the following advantages:

- There is no need to deal with vertical constraints
- The required channel height, POTENTIAL(0), can be precisely computed
- For D2TN channels that have a POTENTIAL(0) value less than the Manhattan density, our router can out-perform any Manhattan routers
- Extra columns outside the channel's span are never used
- Wirelength is expected to be shorter due to the use of diagonal wires
- Being a minimal crossing solution, we expect only a small number of vias is required. We observed that most nets require zero or one via, whereas Manhattan routing typically requires at least 2 vias per net.
- It is inherently suitable for parallel mode of operations.

- In standard cell design, it is very difficult to optimize the assignment of pins to feedthroughs when the objective function is the density. In our environment, one would want to minimize the maximum displacement of a net. This can be easily done by linear assignment.

Figure 8. A channel is routed three models shown in (a), (b), and (c). The results are tabulated in (d).





(d) Comparison

	Man	Lodi	Mini Swap
height	7	3	3
area	56	42	18
vias	20	4	0
wire length	71	42	28

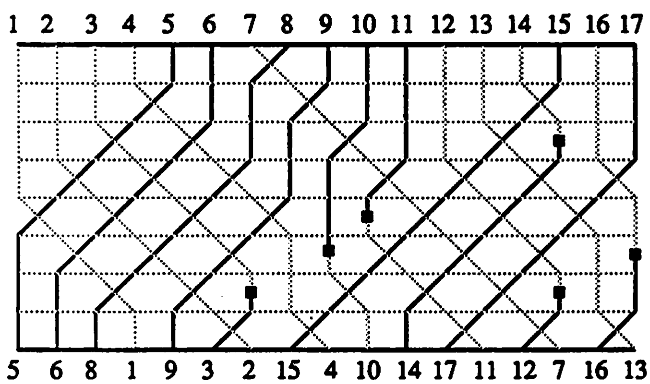


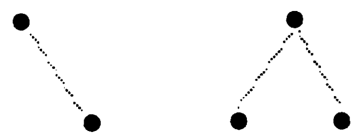
Figure 9. A channel with density = 7 is optimally routed in the mini swap model.

The diagonal model channel router proposed by [17] can complete routing of a D2TN channel in $max(\{displacement(i) \mid 1 \leq i \leq N\})$ tracks. Although this number is smaller than the channel height required by our router (see theorem 4), the diagonal router [17] magnifies both the column spacing and row spacing by root(2), which implies that the channel area is doubled. Our router ensures that the wiring paths are design rule correct and does not magnify spacing in either direction.

2.7 Multi-Terminal Net

A multi-terminal net is partitioned into 2-terminal and 3-terminal subnets. The parallel bubble sort algorithm can easily handle 2-terminal subnets and 3-terminal subnets which have a single destination point on TOP (Figure 10a). Routing the latter subnets corresponds to sorting multiple copies of the same number (see example in Figure 10a). However, 3-terminal subnets that have two destination points on TOP (Figure 10b) does pose a problem to the sorting algorithm. To overcome this problem, we transform such a subnet into two 2-terminal subnets, which can be handled by the parallel bubble sort, by introducing a pseudo terminal next to the bottom terminal. Horizontal subnets with terminals on the same side are routed by horizontal wires and do not participate in the parallel bubble sort.

Figure 10a. Parallel bubble sort can easily handle these types of subnets:



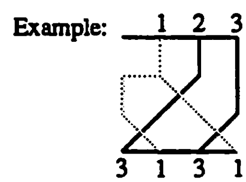
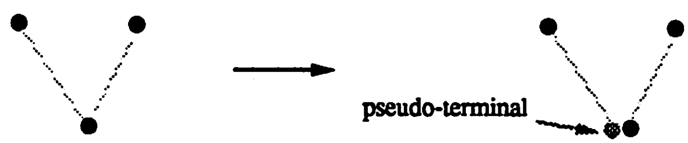


Figure 10b. 3-terminal subnet with two destination points are transformed into two 2-terminal subnets by introducing a pseudo terminal



We artificially divide the channel into three regions: UPPER, MIDDLE, and LOWER. A high level description of our routing algorithm is as follow:

step 1) Route $S1 = \{$ subnets with terminals on TOP only $\}$ in the UPPER region.

step 2) Route $S2 = \{$ subnets with terminals on BOTTOM only $\}$ in the LOWER region.

step 3) Introduce pseudo terminals

step 4) Use parallel bubble sort to route the remaining subnets, denoted by the set $S3$, across the channel in the MIDDLE region.

The height of the UPPER (LOWER) region equals to the density of the set $S1$ ($S2$) since all terminals are on the same side, ie. no vertical constraints are present. The height of the MIDDLE region is the initial POTENTIAL value evaluated with the set $S3$. The required channel height, $density(S1) + density(S2) + POTENTIAL(S3)$, can be precisely computed.

2.8 Concluding Remarks on the Mini-Swap Model

In section 2, we have proposed a new channel routing model, the "mini-swap" model. The routing algorithm is based on the parallel bubble sorting technique. A function named "POTENTIAL", originally defined for characterizing the bubble system, can be evaluated to compute the precise number of tracks required to route the channel. The function evaluation of POTENTIAL for a given channel can be obtained without referring to the intermediate routing steps. We have established necessary and sufficient optimality conditions for routing under the "mini-swap" model. The final routing solution has an unambiguous layer assignment and is design-rule correct. Our results show that a class of dense two-terminal net channels can be routed in a height less than the Manhattan density. This method is also extended to handle multi-terminal nets.

3 An Overlap Model

3.1 Introduction

The results of the previous section show the advantages of using the mini-swap model for routing channels whose POTENTIAL value is lower than or equal to the Manhattan density. Using the mini-swap model to route channels with "long" nets (nets spanning a number of columns which is larger than the Manhattan density), however, leads to unsatisfactory results. How can we overcome this deficiency? Since theorem 3 confirmed that we could not reduce the required channel height by cleverly selecting pairs to swap, the only alternative that may lead us to a better solution is to to relax the constraint of pair-wise transpositions. In other words, we shall allow long nets to swap with more than one net at a time so that they may reach their target positions more rapidly. In this section, we introduce an overlap model. The routing algorithm is based on the river routing technique, in which, like the mini-swap model, vertical constraints do not exist. To analyze the routing performance, the POTENTIAL function again proves to be a useful tool in providing an upper bound of the channel height. We also establish a lower bound of the channel height that incorporates the relative ordering of the nets. For a subclass of CRP, this demonstrates that density is not a tight lower bound. Unlike other overlap models [12, 19, 4] that used two or more vias per net, the routing solution produced by our router can be wired using two interconnect layers so that at most one via is required per net. Furthermore, all wires are monotonic in both the horizontal and vertical directions.

3.2 Algorithm for the Overlap Model

The basic idea of the new scheme is to allow long nets to swap with more than one net while maintaining the integrity of net crossings. To improve the results of the mini-swap model, the long nets must be given a better chance in swapping with other nets. At the same time, we want to keep the number of net crossings minimal so that the number of vias will not increase.

The routing algorithm is based on the river routing technique. A dense 2-terminal net channel routing problem is specified in Definition 3. A net is said to be a *right* net if $B(i) > i$; otherwise, it is said to be a *left* net (Figure 12). The basic idea of the overlap model is to divide the nets into two disjoint subsets: $S1$ = { *right* nets } and $S2$ = { *left* nets } and routes each set independently. A top level description of the algorithm is given below:

step 1) Nets are divided into two sets: S1 = { right nets }; S2 = { left nets }

step 2) Sort S1 in decreasing order of their top terminals. Route S1.

step 3) Sort S2 in decreasing order of their bottom terminals. Route S2.

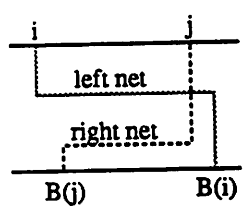


Figure 12.

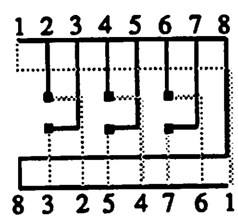


Figure 13. Routing in the overlap model

The solution for the D2TN channel routing problem is constructed in a bottom- to-top net-by-net fashion. Let us demonstrate the algorithm on the example shown in Figure 13. The right nets are sorted and routed sequentially in a river routing fashion. The procedure begins by constructing a rectilinear wiring path for the first right net. The path begins at the bottom terminal and ends at the top terminal. The wiring path for the second right net simply follows the path of the first right net and ends at its top terminal. This process continues until all of the right nets are routed. All of the paths are monotonic in both the horizontal and vertical directions. Step 3 is identical to step 2, except that this time the left nets are routed.

Layer assignment for the wiring paths is trivial. For the right (left) nets, the vertical segment attached to their bottom (top) terminals is assigned layer 2 (1) while the remaining wire segments are assigned layer 1 (2), see Figure 13. A via is introduced for each layer change. Clearly, the routing paths for a pair of planar nets do not intersect and intersecting pairs intersect only once. This that implies the routing solution contains a minimal set of net crossings.

3.3 Performance Analysis

We establish four results associated with the overlap model: lower- and upper- bounds of channel height, wire length and number of vias. The lower- and upper- bounds of the channel height are derived by considering the permutation of nets. Suppose the D2TN channel routing problem is specified as a pair of vectors: *TOP* = { *1, 2, ..., N* } and *BOT* = { *b(1), b(2), ... b(N)* } . Let us define the term *NOTORDER* and the function *POTENTIAL* as in Definition 6 and 7.

Theorem 5 *The lower Bound of channel height under the arbitrary overlap model is max NOTORDER(1,i,0,S1) + NOTORDER(i,N,0,S2)* $| 1 \leq i \leq N$

Proof: At each column i, the number of horizontal tracks required on layer 1 is at least *NOTORDER(i,1,0,S1)*. The number of horizontal tracks required on layer 2 is at least *NOTORDER(i,N,0,S2)*. Since the layer 2 tracks must be stacked on top of the layer 1 tracks, this implies the height of the channel at column i is at least the sum of the two terms. The overall channel height is then the maximum over all columns. ■

The above lower bound is derived by considering the permutation of the nets. It is not only a tighter bound than the Manhattan density, d, but also demonstrates that $d/(L-1)$ [14, 11] is not a universal lower bound under the unrestricted overlap models, where L is the number of layers. The example in Figure 14 is routed by the overlap router in a height equal to the lower bound, which is half of the Manhattan density. We prove in the following theorem that, in contrast to the bubble system, the POTENTIAL value will decrease by more than 1 per track (amortizedly) under the overlap model.

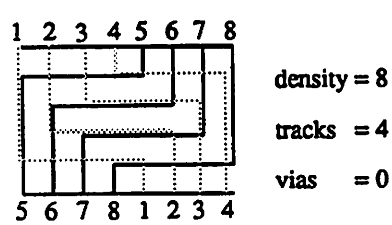


Figure 14.

Theorem 6 *The upper Bound of channel height under the arbitrary overlap model is* $POTENTIAL(0) + 2$.

nets is similar. ■

3.4 Refinement of Routing Strategy

One weakness of the overlap routing method is its inefficiency in handling a consecutive sequence of planar nets, as shown in Figure 15a. A modification of the routing procedure to improve routing results in such situations is proposed. The basic idea is that instead of routing all right (left) nets in one step, we route the right and left nets alternatingly until the nets are sorted. The remaining nets are planar and can be optimally routed by the "layer per net" method [3, 15]. The "layer per net" method states that, given 2 interconnect layers and a set of planar nets, 1, 2, ... M , we assign the odd-numbered nets to layer 1 and the even-numbered nets to layer 2, solve the two one-layer river routing problems that are thus formed to yield the minimum width channel.

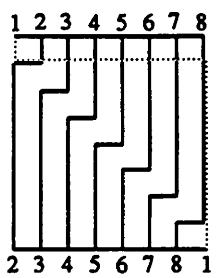


Figure 15a. Before refinement

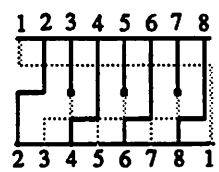


Figure 15b. After refinement

Refined Routing Algorithm for Rectilinear Overlap Model

step 1) Divide the nets into two sets: S1 = {right nets} and S2 = {left nets}

step 2 Sort right nets in decreasing order of their top terminals: S1 = { r1, r2, r3, ... }

step 3) Sort *S2* in decreasing order of their bottom terminals: S2 = { l1, l2, l3, ... }

step 4) Alternatingly route the right and left nets, ie. route in the order: r1, l1, r2, l2, r3, l3, ... until the nets are sorted.

step 5) Route remaining planar nets by the "layer per net" model

The example in Figure 15a is rerouted by the refined procedure to yield a much better solution as shown in Figure 15b.

3.5 Multi-terminal Nets and Side Nets

To extend the overlap routing algorithm to handle multi-terminal nets, we partition each multi-terminal net into 2-terminal subnets and classify each subnet as a *right* net or a *left* net. A 2-terminal subnet with one terminal on TOP and another on BOTTOM is said to be a *2-sided* subnet; otherwise, it said to be a *1-sided* subnet. The 2-sided subnets can be categorized as left or right easily. However, the classification of 1-sided subnets is ambiguous and may affect the quality of solution. In the following lemmas, we show that not all 1-sided subnets can be classified either way.

Definition 8 *Given I, a two-terminal 1-sided subnet:*

- *I is top-sided iff both terminals of I lie on TOP.*
- *I is bottom-sided iff both terminals of I lie on BOTTOM.*

Lemma 2 *Given a 1-sided subnet I,*

i) I is top-sided and I intersects a right net ⇒ I must be a left net.

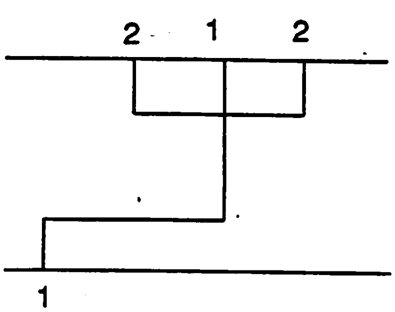
ii) I is bottom-sided and I intersects a left net ⇒ I must be a right net.

iii) I is top-sided and I intersects another top-sided subnet J ⇒ I and J can not both be right nets.

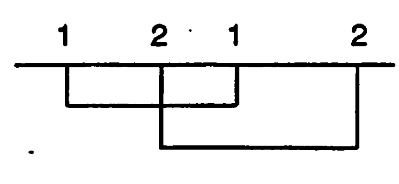
iv) I is bottom-sided and I intersects another bottom-sided subnet J ⇒ I and J can not both be left nets.

Proof:

i) Suppose for the sake of contradiction that net 2 is classified as a right net. Then by the algorithm, net 2 must be routed before net 1 on layer 1. This shorts the two wires.



i



ii

ii) Suppose for the sake of contradiction that both net 1 and net 2 are classified as right nets. Then the routing algorithm would route both nets on layer 1 and result in conflict.

iii) Similar to i).

iv) Similar to ii).

■

After applying Lemma 2 to the 1-sided subnets, most 1-sided nets in practical examples are classified. The classification of the remaining 1-sided subnets is guided by local congestion analysis. The left subnets and right subnets then are sorted independently by their terminal positions and routed in the rivering routing fashion as discussed in section 3.2. We observe this method for routing multi-terminal nets is straight-forward but not necessarily optimal. Future work should investigate better strategy for handling multi-terminal nets.

In a channel routing problem, nets may exit the channel on either its left end or its right end. Routing of these so-called side nets is done by the following procedure:

The analysis and theorems introduced for the D2TN channel routing problem in Section 3.2 can be generalized for the multi-terminal nets problem by replacing nets with subnets in the equations.

3.6 Concluding Remarks on the Overlap Model

Compared to the Manhattan model, the overlap model has the following advantages:

- There is no need to deal with vertical constraints
- For channels that have POTENTIAL(0) value less than the Manhattan density, our model can out-perform any Manhattan routers
- Extra columns outside the channel's span are never used
- Because it is a minimal crossing solution, we expect only a small number of vias to be required.

The overlap model is better than the mini-swap model in two respects: (1) It yields a small channel area for channel with long nets. (2) Variable wire widths can be incorporated easily. However, overlapping wires may increase crosstalk between signals and unlike the mini-swap algorithm, the overlap algorithm is not suitable for parallel operations.

4 The Hybrid Router

We combine the advantages of the overlap and the miniswap models to form a new hybrid router. This router pre-routes long nets using the overlap algorithm so that the amount of overlapping wire is limited. The remaining nets, ie. relatively short nets, are routed by the miniswap algorithm. The user can specify the maximum amount of overlap allowed.

4.1 Algorithm

The basic idea is as follows: When the channel is routed solely by the miniswaps, we know the channel height is POTENTIAL(0). Suppose we pick some long nets which would cause the mini-swap model to disgrace itself, and pre-route them using the overlap strategy. If the resulting channel height is better than before, we can choose more long nets and continue the process provided the maximum amount of overlap is not exceeded. A top level description of the algorithm is given below:

For $i \leftarrow 1$ to $n-1$ do

i) pre-route i longest nets by the overlap model

amount of overlap $= i$,

calculate current channel height, t(i),

$t(i) =$ (tracks for pre-routes) + (POTENTIAL value of remaining channel)

ii) compare height to last iteration:

if $(t(i) < t(i-1))$ and (max overlap is not exceeded)

$i = i + 1$

continue

else stop

After selecting a set of long nets, it is not necessary to pre-route them to their final positions. It suffices to pre-route them to intermediate positions with the objective of minimizing the POTENTIAL value. Our aim is to simultaneously process the long nets and to find an intermediate position for each net. Assignment of a net to a position should be done in such a way that the displacement of that net is as small as possible. Using this criterion, many nets may compete for the same position. To find a good assignment of nets to positions, we use a linear bottleneck assignment approach. We build a complete bipartite graph: G(V,U,E), with two sets of nodes. One set, V = P(i), i = 1,...K, represents the nets; the other set, U = 1,2,...K, represents the positions. K is the number of long nets. Edge (i,

j) connects net node i to position node j. Edge (i, j) is labeled with cost c[i,j] indicating the the POTENTIAL value of net i when it is assigned to position j. (see Fig. 16). Standard methods [25, 6, 21, 17] may be used to solve this problem. An example routed by the hybrid router is shown in Figure 17.

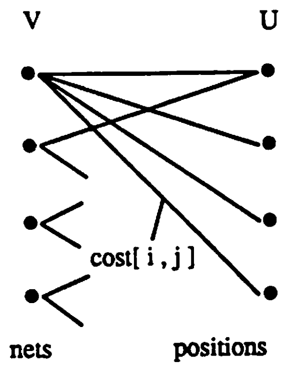


Figure 16. Linear assignment

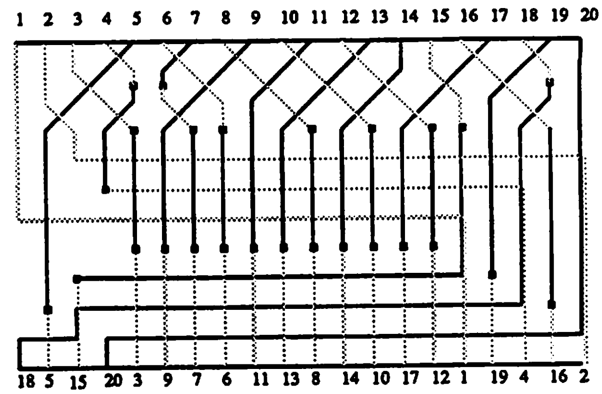


Figure 17. A channel is routed by the hybrid router in density.

4.2 Performance Analysis

Since the hybrid router is a combination of the mini-swap model and the overlap model, it should perform at least as well as either model in the worst case. Given a channel routing problem, if the algorithm terminates in the first ((N − 1)th) iteration, the solution has purely mini-swaps (overlaps). Therefore, the mini-swap model and the overlap model can be considered special cases of the hybrid router. We summarize the performance bounds of the hybrid router in the following theorem.

Theorem 7 *Lower Bound of channel height required by the hybrid router*

$$= max\{NOTORDER(i,1,0,S1) + NOTORDER(i,1,N,S2) \mid 1 \leq i \leq N\})$$

Upper Bound of channel height required by the hybrid router

$= POTENTIAL(0) + 1.$

5 Results

The hybrid router is implemented in the C language on a DEC3100 running Ultrix Worksystem V2.1. Figure 18 shows the routing result of a channel with 48 nets. We have attempted to run this example using other routers available[yacr,glitter] but they either failed to complete routing or produced substantially worse results. Table I lists the channels tested with 100% routing completion in all cases. Several D2TN channels are routed in a height less than the density. Due to the simple heuristics used for routing multi-terminal nets, the difficult channel was routed with 4 tracks more than the density. However, our result has 25% less vias than [7, 22].

Table 1. Experimental Results

Example	density	final height	CPU
Ex1	8	7	0.1
Ex2	7	8	0.1
Ex3	4	3	0.1
Ex4	18	22	0.1
Ex5	4	5	0.1
Ex6	8	5	0.1

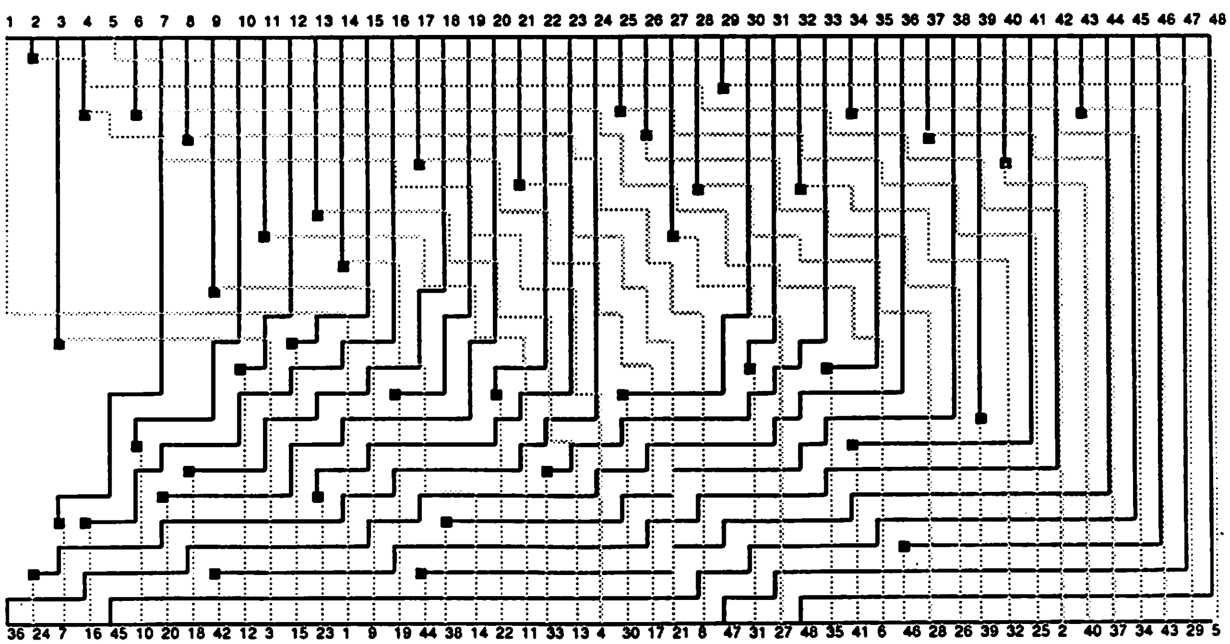


Figure 18. Routing result of Ex4

6 Conclusion and Future Work

In this paper, we propose new methods which perform well on practical channels and attempt to provide accurate and formal analysis on the quality of the solutions. Two new routing models, the *mini-swap* model and the *overlap* model are introduced. Non-Manhattan geometry as well as rectilinear wires are used. The routing strategy is based on the parallel sorting and river routing techniques and is very different from the Manhattan approach. In particular, vertical constraints no longer exist. Furthermore, the solution produced by either model consists of a minimal set of net crossings which leads to a small number of vias. To characterize and to evaluate the performance of each model, a function called "POTENTIAL" is used. Intuitively, the "POTENTIAL" function measures the degree of difficulty for a given channel routing problem. Based on the performance analysis, we attempt to combine the strength of both models to form a new *hybrid* router. The routing solution produced by our router has unambiguously layer assignment and is design-rule correct. Finally, a straight-forward extension to handle multi-terminal nets and and side-nets is proposed. Preliminary results show that a class of two-terminal net channel routing problems can be routed in height less than the Manhattan density. Future work should optimize the extension to handle multi-terminal nets and multi-layers.

A Proof of Theorem 3

```
For brevity of notation,

Let A  = ( a(1), a(2), .... , a(N) )  denote the permutation of
      nets on track t.

Let A' = ( a'(1), a'(2), .... , a'(N) )  denote the permutation of
      nets on track t+1.

Let DELTA(i,j,t) = ORDER(i,j,t) - NOTORDER(i,j,t) + 1


{Case 1.}  When NOTORDER(i,N,t) = 0

  POTENTIAL(i,t) = NOTORDER(i,1,t) + MAXLT(i,t)                          (AA)

  where MAXLT(i,t) = max({0} U {DELTA(i,j,t) | j < i, and a(i) < a(j)} )


  {Subcase 1.1}  If net at position i does not switch position at t,

      NOTORDER(i,1,t+1) = NOTORDER(i,1,t)                                (BB)

      Let J = { j | 1 <= j < i, a(i) < a(j) }
      Let j be any memeber of J.


      {Subcase 1.1a}  If the net at position j switches position
                      with its right neighbor:

          We have a(j+1) < a(j) and a(i) < a(j) at time t.

          If a(j+1) < a(i), then

              ORDER(i,j+1,t+1) = ORDER(i,j,t) - 1   and
              NOTORDER(i,j+1,t+1) = NOTORDER(i,j,t)
              Hence, DELTA(i,j+1,t+1) = DELTA(i,j,t) - 1

          Otherwise, i.e. if a(j+1) > a(i), then

              ORDER(i,j+1,t+1) = ORDER(i,j,t) and
              NOTORDER(i,j+1,t+1) = NOTORDER(i,j,t)
              Hence, DELTA(i,j+1,t+1) = DELTA(i,j,t)


      {Subcase 1.1b}  If the net at position j switches position
                      with its left neighbor:

          We have a(i) < a(j) < a(j-1) at time t.

          Hence,
              ORDER(i,j-1,t+1) = ORDER(i,j,t)
              NOTORDER(i,j-1,t+1) = NOTORDER(i,j,t)
              DELTA(i,j-1,t+1) = DELTA(i,j,t)
```

{Subcase 1.1c} If the net at position j does not switch position:

```
Clearly, ORDER(i,j,t+1) = ORDER(i,j,t)
         NOTORDER(i,j,t+1) = NOTORDER(i,j,t)
         DELTA(i,j,t+1) = DELTA(i,j,t)
```

Since in subcases 1.1a, 1.1b and 1.1c, DELTA(i,j,t) decreases at most by 1, MAXLT(i,t) decreases at most by 1. Substitute MAXLT(i,t) and (BB) into (AA), we can conclude POTENTIAL decreases at most by 1 for subcase 1.1.

{Subcase 1.2} If net at position i switches position at t,

then it must switch with the net on its left since

NOTORDER(i,N,t) = 0.

Hence, we have NOTORDER(i-1,1,t+1) = NOTORDER(i,1,t) - 1 (CC)

Let J = { j | 1 <= j < i, a(i) < a(j) }
Let j be any memeber of J.

{Subcase 1.2a} If the net at position j does not switch position

```
Then, ORDER(i-1,j,t+1) = ORDER(i,j,t) and
   NOTORDER(i-1,j,t+1) = NOTORDER(i,j,t) - 1
      DELTA(i-1,j,t+1) = DELTA(I,J,T) + 1
```

{Subcase 1.2b} If the net at position j switches with rihgt neighbor

If a(j+1) < a(i) < a(j), then

```
ORDER(i-1,j+1,t+1) = ORDER(i,j,t) - 1
NOTORDER(i-1,j+1,t+1) = NOTORDER(i,j+1,t) - 1
DELTA(i-1,j+1,t+1) = DELTA(i,j,t)
```

Otherwise, i.e. a(i) < a(j+1) < a(j), we have

```
ORDER(i-1,j+1,t+1) = ORDER(i,j,t)
NOTORDER(i-1,j+1,t+1) = NOTORDER(i,j,t) - 1
DELTA(i-1,j+1,t+1) = DELTA(i,j,t) + 1
```

{Subcase 1.2c} If the net at position j switches with left neighbor

We have a(i) < a(j) < a(j-1). Then

ORDER(i-1,j-1,t+1) = ORDER(i,j,t)
NOTORDER(i-1,j-1,t+1) = NOTORDER(i,j,t) - 1
DELTA(i-1,j-1,t+1) = DELTA(i,j,t) + 1

Since in subcases 1.2a, 1.2b and 1.2c, DELTA(i,j,t) increases by 0 or 1, MAXLT(i,t) increases by 0 or 1. Substitute MAXLT(i,t) and (CC) into (AA), we can conclude POTENTIAL decreases at most by 1 for subcase 1.2.

{Case 2.} When NOTORDER(i,1,t) = 0

POTENTIAL(i,t) = NOTORDER(i,1,t) + MAXGT(i,t)

The proof of this case is similar to that of {Case 1}.

{Case 3.} When NOTORDER(i,1,t) = 0 and NOTORDER(i,N,t) = 0

POTENTIAL(i,t) = NOTORDER(i,1,t) + NOTORDER(i,1,t) + max{ 1, MAXLT(i,t), MAXGT(i,t) } (**)

{Subcase 3.1} If net at position i does not switch position at t,

then,

ORDER(i,1,t+1) = ORDER(i,1,t) (DD)
NOTORDER(i,1,t+1) = NOTORDER(i,1,t) (EE)

Similar to the proof of {Subcase 1.1}, we can show

MAXLT(i,t+1) = MAXLT(i,t) or MAXLT(i,t) - 1 (FF)
MAXGT(i,t+1) = MAXGT(i,t) or MAXGT(i,t) - 1 (GG)

Substituting equations (DD), (EE) , (FF) and (GG) into equation (**), we conclude POTENTIAL decreases at most by 1 for {Subcase 3.1}.

{Subcase 3.2} If net at position i switches position with its

left neighbor, then,

ORDER(i-1,1,t+1) = ORDER(i,1,t) - 1 (HH)
NOTORDER(i-1,1,t+1) = NOTORDER(i,1,t) (II)

Similar to the proof of {Subcase 1.2}, we can show

MAXLT(i-1,t+1) = MAXLT(i,t) or MAXLT(i,t) + 1 (JJ)

To see the changes in MAXGT(i-1,t+1), we again do a case analysis.

Let J' = { j | i <= j < N, a(j) < a(i) }
Let j be any memeber of J'.

{Subcase 3.2a} If the net at position j does not switch position

then,

```
ORDER(i-1,j,t+1)    = ORDER(i,j,t) + 1
NOTORDER(i-1,j,t+1) = NOTORDER(i,j,t)
DELTA(i-1,j,t+1) = DELTA(i,j,t) + 1
```

{Subcase 3.2b} If the net at position j switches position with its right neighbor:

then,

```
a(j+1) < a(j) < a(i)
ORDER(i-1,j+1,t+1)    = ORDER(i,j,t) + 1
NOTORDER(i-1,j+1,t+1) = NOTORDER(i,j,t) + 1
DELTA(i-1,j+1,t+1)    = DELTA(i,j,t)
```

{Subcase 3.2c} If the net at position j switches position with its left neighbor:

then, a(j) < a(j-1) and a(j) < a(i)

If a(j) < a(j-1) < a(i)

```
ORDER(i-1,j-1,t+1) = ORDER(i,j,t) + 1
NOTORDER(i-1,j-1,t+1) = NOTORDER(i,j,t) - 1
DELTA(i-1,j-1,t+1) = DELTA(i,j,t) + 2
```

Otherwise, i.e. a(j-1) > a(i),

```
ORDER(i-1,j-1,t+1) = ORDER(i,j,t)
NOTORDER(i-1,j-1,t+1) = NOTORDER(i,j,t)
DELTA(i-1,j-1,t+1) = DELTA(i,j,t)
```

Since in subcases 3.2a, 3.2b and 3.2c, DELTA(i,j,t) increases by 0, 1, or 2, MAXGT(i,t) increases by 0, 1 or 2. Substitute MAXGT(i,t), (HH), (II), and (JJ) into (**), we can conclude POTENTIAL decreases at most by 1 for {subcase 3.2}.

{Subcase 3.3} If net at position i switches position with its right neighbor, the proof is similar to {Subcase 3.2}.

QED.

B Proof of Theorem 4

```
For any net i, the top terminal of net i is in column i, and
the bottom terminal is in, say, column j.

{Case 1}  j >= i

    {Subcase 1.1}  When NOTORDER(i,N) = 0

        Obviously NOTORDER(i,1) != 0

        Let S = { i+1, i+2, ...., N }

        Let S1    S be the nets that have bottom terminals

        in positions j+1, j+2, ...., N.  Clearly,

        |S1| <= N - j

        Let S2 = S - S1 be the nets that have bottom terminals

        in positions 1, 2, ...., j-1.  Then

        |S2| >= (N-i) - (N-j) = j - i = displacement(i)

        POTENTIAL(i,t) = NOTORDER(i,1,t) + MAXLT

                       = |S2| + MAXLT

                       > displacement(i)                QED.

    {Subcase 1.2}  When NOTORDER(i,1) != 0  and NOTORDER(i,N) != 0

        Let S = { i+1, i+2, ...., N}

        Let S1    S be the nets that have bottom terminals

        in positions 1, 2, .... j - 1.  Then

        |S1| = NOTORDER(i,1) >= displacement(i)

        POTENTIAL(i,t) = NOTORDER(i,1,t) + NOTORDER(i,N,t) +

                         max( 1, MAXLT, MAXGT )

                       = |S1| + NOTORDER(i,N,t) + max( 1, MAXLT, MAXGT )

                       > displacement(i)

    The proof for {Case 2} is similar.                 QED.
```

C Proof of Theorem 6

[Proof]

Without loss of generality, the channel can be viewed as having two sections: the bottom section and the top section. In the bottom section, the right nets are routed to their target positions; in the top section, the left nets are routed.

```
-----------------------------------------

   top section ( for routing left nets )

.........................................

 bottom section ( for routing right nets )

-----------------------------------------
```

Let us consider the change in the POTENTIAL values of nets in the bottom section. The proof is by Lemma I, II and III. The main result of these three lemmas is that the POTENTIAL value of every net decreases by at least 1 (amortized) per track in the bottom section. The proof for the top section is similar.

Lemma I. In the bottom section, If NOTORDER(i, N, t) = 0,

then POTENTIAL(i,t+1) = POTENTIAL(i,t) - m

where m >= 1, unless

POTENTIAL(i,t) = 0 which implies POTENTIAL(i,t+1) = 0.

[Proof]

When NOTORDER(i, N, t) = 0,

POTENTIAL(i,t) = NOTORDER(i,1,t) + MAXLT(i,t)

= NOTORDER(i,1,t) + max{ DELTA(i,j,t) } (AA)

{ Case 1 } Net i is a right net:

Claim. Net i has reached its target position,

ie. POTENTIAL(i,t) = 0

[proof]

Suppose for the sake of contradiction that net i has reached position i' < i. Now, there are (N - i) nets which are greater than i. These nets can not fill up the positions in the interval [i'+1, N]. Hence some net j < i is in this interval. This contradicts the fact NOTORDER(i, N, t) = 0.

QED

{ Case 2 } Net i is a left net and i crosses a net i':

When i crosses a net,

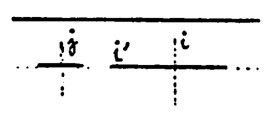
NOTORDER(i,1,t+1) = NOTORDER(i,1,t) - 1   (BB)

Let I = { j | j < i, a(i) < a(j) }
Let j be a member of I.

By examining equations (AA) and (BB), we need to show that DELTA(i,j,t) does not increase.

{ Case 2.1 } When j crosses another net:

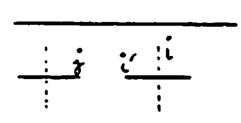
If j propagates vertically upwards as shown in Figure (a):



(a)

Then DELTA(i,j,t+1) = DELTA(i,j,t)

Else if j propagates horizontally to the right as shown in Figure (b):



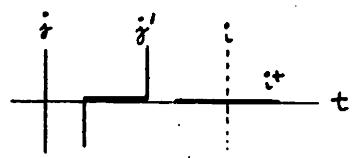
(b)

Then there exist a net j' > j such that j' > i. Thus

```
ORDER(i,j,t+1) = ORDER(i,j',t)
NOTORDER(i,j,t+1) = NOTORDER(i,j',t)
DELTA(i,j,t+1) = DELTA(i,j',t)
```

{ Case 2.2 } When j does not cross another net:

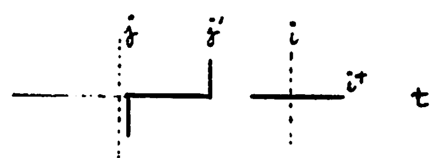
If j is a right net as shown in Figure (c):



(c)

then, DELTA(i,j,t+1) = DELTA(i,j',t)

Else, ie, j is a left net as shown in Figure (d):



(d)

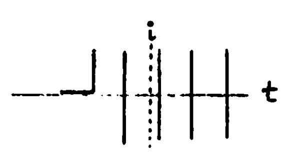
Then there exists a right net j' overlapping with j (else j would have been crosses by a right net) such that j' > j. Hence,

DELTA(i,j,t+1) = DELTA(i,j',t) - 1.

In subcases (2.1) and (2.2), DELTA(i,j,t+1) does not increase. Hence we conclude in {Case 2}, POTENTIAL(i,t+1) = POTENTIAL(i,t) - m, where m = 1 or 2.

QED

{ Case 3 } Net i is a left net and i does not cross another net as shown in Figure (e):



(e)

Since i is a left net and NOTORDER(i,N,t) = 0, net i can be routed to its target position at the t'th track in the top section which reduces POTENTIAL(i,t) to 0 (see proof of Lemma 4). This implies the amortized decrease in POTENTIAL value of net i is greater than 1. Note: once POTENTIAL(i,t+1) becomes zero, net i will be dropped from analysis in future steps.

QED

Lemma II. In the bottom section, If NOTORDER(i, 1, t) = 0,

then POTENTIAL(i,t+1) = POTENTIAL(i,t) - m

where m >= 1.

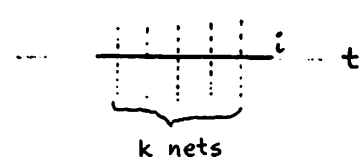
[Proof]

In the bottom section, only a right net could satisfy NOTORDER(i,1,t) = 0. By the routing algorithm, net i would propagte horizontal toward right until another right net i', i' > i is reached. Since NOTORDER(i,1,t) = 0, the POTENTIAL value of net i is defined as

POTENTIAL(i,t) = NOTORDER(i,N,t) + MAXGT(i,t)

= NOTORDER(i,N,t) + max{ DELTA(i,j,t) } (CC)

{ Case 1 } Net i crosses k nets as shown in Figure (f):



(f)

Thus,

NOTORDER(i,N,t+1) = NOTORDER(i,N,t) - k (DD)

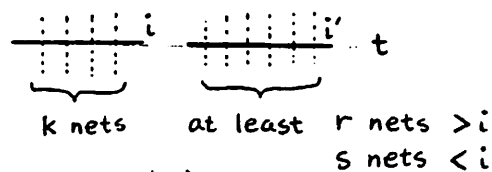
Let I = { j | j < i, a(i) < a(j) }
Let j be a member of I.

By examining equations (CC) and (DD), we We need to show DELTA(i,j,t) does not increase more than k.

If MAXGT(i,t+1) = 0, then

POTENTIAL(i,t+1) = POTENTIAL(i,t) - k

Else, MAXGT(i,j,t+1) > 0 corresponds to the nets shown in Figure (g).



(g)

Hence,

$$\begin{aligned} DELTA(i,j,t+1) &= ORDER(i,j,t+1) - NOTORDER(i,j,t+1) \\ &= r - s \\ &= (r + 1) - (k + s) + (k - 1) \\ &= ORDER(i,j,t) - NOTORDER(i,j,t) \\ &\quad + (k - 1) \\ &= DELTA(i,j,t) + (k - 1) \end{aligned}$$

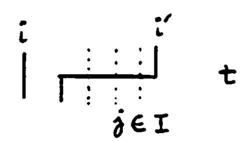
QED

{ Case 2 } When Net i does not cross any net,

NOTORDER(i,N,t+1) = NOTORDER(i,N,t)

We need to show that DELTA(i,j,t) decreases.

If MAXGT(i,t) > 0 as shown in Figure (h):

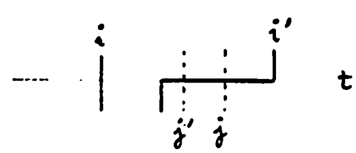


(h)

Then

$$\begin{aligned} DELTA(i,j,t+1) &= ORDER(i,j,t+1) - NOTORDER(i,j,t+1) \\ &= (ORDER(i,j,t) -1) - NOTORDER(i,j,t) \\ &= DELTA(i,j,t) - 1 \end{aligned}$$

Else if MAXGT(i,t) = 0 as shown in Figure (p):



(p)

This implies j' < i and net i will cross j' and j in track t+1. Hence the amortized decrease in the POTENTIAL value of net is 1. QED

Lemma III. In the bottom section, If NOTORDER(i, 1, t) != 0 and

NOTORDER(i, N, t) != 0

then POTENTIAL(i,t+1) = POTENTIAL(i,t) - m

where m >= 1, unless

POTENTIAL(i,t) = 0 which implies POTENTIAL(i,t+1) = 0.

[Proof]

When NOTORDER(i,1,t) != 0 and NOTORDER(i,N,t) != 0,

net i propagates vertically upwards on layer 2 in the

bottom section and

POTENTIAL(i,t) = NOTORDER(i,1,t) + NOTORDER(i,N,t) + max{1, MAXLT, MAXGT} (EE)

{ Case 1 } If net i crosses another net, then

NOTORDER(i,1,t+1) = NOTORDER(i,1,t) - 1 (FF)

NOTORDER(i,N,t+1) = NOTORDER(i,N,t) (GG)

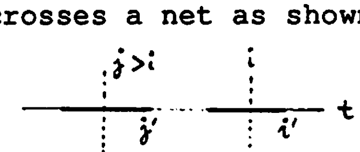
By examining equations (EE), (FF) and (GG), we
We need to show DELTA(i,j,t) does not increase.

{ Case 1.1 } i is a right net

To show the change in MAXLT,

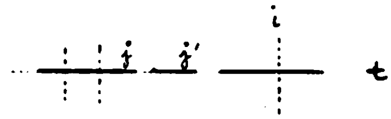
let I = { j | j > i, a(i) > a(j) }.
Let j be a member of I.

(1.1A) If j crosses a net as shown in Figure (q):



Then j' > j > i. DELTA(i,j,t+1) = DELTA(i,j,t)

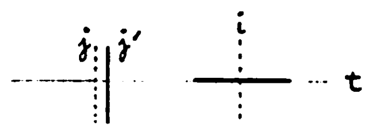
(1.1B) if j crosses a net as shown in Figure (r):



(r)

Then j' > j > i. DELTA(i,j,t+1) = DELTA(i,j',t) which implies DELTA(i,j,t+1) does not increase.

(1.1C) If j does not cross any net as shown in Figure (s),



(s)

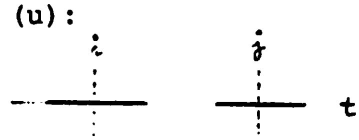
then DELTA(i,j,t+1) = DELTA(i,j',t)

From (1.1A), (1.1B) and (1.1C), we conclude MAXLT(i,t) decreases by >= 0.

To show the change in MAXGT,

Let I' = { j | j < i, a(i) < a(j) }
Let j be a member of I'.

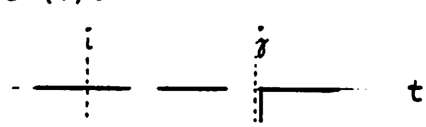
(1.1D) If j crosses another net as shown in Figure (u):



(u)

Then, DELTA(i,j,t+1) = DELTA(i,j,t)

(1.1E) net j does dnot cross any net as shown in Figure (v):



(v)

If DELTA(i,j,t+1) <= 1, then done. Else, ie. DELTA(i,j,t+1) >= 2, then net j will cross >= 2 nets in some track in the top section. This implies the amortized decrease in the POTENTIAL value of net i is >= 1.

QED

{ Case 1.2 } i is a left net

The proof is analogous to { Case 1.1 }.

{ Case 2 } If net i does not cross any net, then

NOTORDER(i,1,t+1) = NOTORDER(i,1,t) (HH)

NOTORDER(i,N,t+1) = NOTORDER(i,N,t) (II)

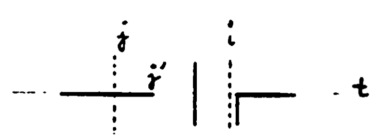
By examining equations (EE), (HH) and (II), we We need to show DELTA(i,j,t) decreases.

{ Case 2.1 } i is a right net

To show the change in MAXLT,

let I = { j | j > i, a(i) > a(j) }.
Let j be a member of I.

(2.1A) If j crosses a net as shown in Figure (w):



(w)

Then j' > j > i.
DELTA(i,j,t+1) = ORDER(i,j,t+1) - NOTORDER(i,j,t+1)
= ORDER(i,j,t) - (NOTORDER(i,j,t)+1)
= DELTA(i,j,t) - 1

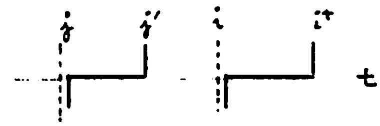
(2.1A) If j crosses a net as shown in Figure (x):

(x)

Then j' > j > i.
DELTA(i,j,t+1) = ORDER(i,j,t+1) - NOTORDER(i,j,t+1)
= ORDER(i,j,t) - (NOTORDER(i,j,t)+1)

= DELTA(i,j,t) - 1

(2.1B) If j does not cross any net as shown in Figure (y):



(y)

Then, DELTA(i,j,t+1)

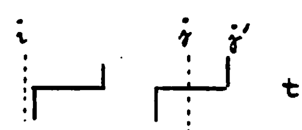
= DELTA(i,j,t)
= ORDER(i,j,t) - NOTORDER(i,j,t)
= ORDER(i,j',t) - (NOTORDER(i,j',t) + 1)
= DELTA(i,j',t) - 1

From (2.1A) and (2.1B), we can conclude that MAXLT(i,t) decreases by 1.

To show the change in MAXGT,

Let I' = { j | j < i, a(i) < a(j) }
Let j be a member of I'.

(2.1C) If j crosses another net as shown in Figure (z):



(z)

Then j' > i. Hence,

DELTA(i,j,t+1) = ORDER(i,j,t) - 1 - NOTORDER(i,j,t)
= DELTA(i,j,t) - 1

(2.1D) net j does not cross another net

Proof is similar to (1.1E)

From (2.1C) and (2.1D), we can conclude that MAXGT(i,t) decreases by 1.

QED

References

[1] S.G. Akl. Linear arrays. In *Parallel sorting algorithms*, pages 41–59. 1985.

[2] B.S. Baker, S.N. Bhatt, and F.T. Leighton. An approximation algorithm for manhattan routing. In *Proc. 15th Annual Symp. on Theory of Computing*, 1983.

[3] A.E. Baratz. *Algorithms for integrated circuit signal routing.* PhD thesis, Massachusetts Institute of Technology, August 1981.

[4] M.L. Brady and D.J. Brown. Optimal multilayer channel routing with overlap. In *Proc. 4th MIT Conference on Advanced Research in VLSI*, pages 281–296, 1986.

[5] D.J. Brown and R.L. Rivest. New lower bounds for channel width. In *Proc. CMU Conf. VLSI Systems and Computations*, pages 178–185, 1981.

[6] M. Burstein. Channel routing. In *Layout Design and Verification*, pages 133–164. 1986.

[7] M. Burstein and R. Pelavin. Hierarchical channel router. In *Proc. 20th Design Automation Conference*, pages 591–597, 1983.

[8] K. Chaudhary and P. Robinson. Private communication. 1990.

[9] D.N. Deutsch. A dogleg channel router. In *Proc. 13th Design Automation Conference*, pages 425–433, 1976.

[10] A. Frank. Disjoint paths in a rectilinear grid. In *Combinatorica, 2(4)*, pages 361–371, 1982.

[11] S.E. Hambrusch. Using overlap and minimizing contact points in channel routing. In *Proc. 21st Annual Allerton Conference on Communication, Control and Computing*, pages 256–257, 1983.

[12] S.E. Hambrusch. Channel routing algorithms for overlap models. In *IEEE Trans. on CAD of Integrated Circuits and Systems, V. CAD-4*, pages 23–30, 1985.

[13] A. LaPaugh. *Algorithms for Integrated Circuit Layout: An Analytic Approach.* PhD thesis, Massachusettes Institute of Technology, November 1980.

[14] F.T. Leighton. New lower bounds for channel routing. In *unpublished manuscript.* 1982.

[15] C.E. Leiserson and R.Y. Pinter. Optimal placement for river routing. In *SIAM Journal on Computing, 12(3)*, pages 447–462, 1983.

[16] E. Lodi, F. Luccio, and L. Pagli. Channel routing for strictly multiterminal nets. In *Integration, the VLSI journal*, pages 143–153, 1989.

[17] E. Lodi, F. Luccio, and L. Pagli. A preliminary study of a diagonal channel-routing model. In *Algorithmica*, pages 585–597, 1989.

[18] K. Mehlhorn, F.P. Preparata, and M. Sarrafzadeh. Channel routing in knock-knee mode: simplified algorithms and proof. In *Algorithmica, 1(2)*, pages 213–221, 1986.

[19] R.L. Rivest, A.E. Baratz, and G.L. Miller. Provably good channel routing algorithms. In *Proc. of the CMU Conference on VLSI Systems and Computations*, pages 153–159, 1981.

[20] R.L. Rivest and C.M. Fiduccia. A greedy channel router. In *Proc. 19th Design Automation Conference*, pages 418–424, 1982.

[21] K. Sado and Y. Igarashi. A function for evaluating the computing time of a bubbling system. In *Theoretical Computer Science (54)*, pages 315–324, 1987.

[22] A. Sangiovanni-Vincentelli and M. Santomauro. Yacr: Yet another channel router. In *Proc. Custom Integr. Circuits Conf.*, pages 460–466, 1982.

[23] M. Sarrafzadeh. *Hierarchical approaches to VLSI circuit layout.* PhD thesis, University of Illinois, December 1986.

[24] M. Sarrafzadeh. Channel-routing problem in the knock-knee mode is np-complete. In *IEEE Trans. on CAD, CAD-6(4)*, pages 503–506, 1987.

[25] T.G. Szymanski. Dogleg channel routing is np-complete. In *IEEE Trans. on CAD, CAD-4(1)*, pages 31–40, 1985.

[26] T. Yoshimura and E.S. Kuh. Efficient algorithms for channel routing. In *IEEE Trans. on CAD of Integrated Circuits and Systems, V. CAD-1*, pages 25–35, 1982.