Provably Good Channel Routing Algorithms

Ronald L. Rivest, Alan E. Baratz, and Gary Miller
Massachusetts Institute of Technology
Laboratory for Computer Science
Cambridge, Massachusetts 02139-1986

I. Introduction

In this paper we present three new two-layer channel routing algorithms that are provably good in that they never require more than $2d$-1 horizontal tracks where d is the channel density, when each net connects just two terminals. To achieve this result we use a slightly relaxed (but still realistic) wiring model in which wires may run on top of each other for short distances as long as they are on different layers. Two of our algorithms will never use such a "parallel run" of length greater than $2d$-1 and our third algorithm will require overlap only at jog points or cross points. Since in this wiring model at least $d/2$ horizontal tracks are required, these algorithms produce a routing requiring no more than four times the best possible number of horizontal tracks. The second algorithm also has the property that it uses uses at most $4n$ contacts, where n is the number of nets being connected.

II. The Model

The (infinite) channel of width t consists of (1) the set V of *grid points* (x,y) such that the integers x and y satisfy the conditions $0 \leq y \leq t+1$ and $-\infty < x < \infty$, (2) the set P of *poly segments* consisting of all unit length line segments connecting pairs of adjacent grid points which do not both have $y=0$ or $y=t+1$, (3) the set M of *metal segments* which is isomorphic to but disjoint from P. The channel (V,P,M) thus forms a multigraph with vertex-set V and edge-set $P \cup M$. If two vertices are adjacent in this graph they are connected by precisely two edges - one of type poly and one of type metal. We define *track* i of the channel (V,P,M) to be the subgraph composed of all grid points in V with y-coordinate equal to i, and all segments of $P \cup M$ which connect pairs of these grid points.

A *wire* W consists of a sequence of distinct grid points separated by segments which connect them:

$$W = (p_0, s_1, p_1, s_2, \ldots, s_k, p_k).$$

Here $p_0, \ldots, p_k$ are the grid points and s_i connects p_{i-1} to p_i. Each s_i may be of either type, poly or metal, and we define the sets of poly segments and metal segments of wire W as follows:

$$P(W) = \{s_i \mid s_i \in P\},$$
$$M(W) = \{s_i \mid s_i \in M\}.$$

Similarly, the *contact points* $C(W)$ is defined to be the set of grid points where W starts, ends or changes layers:

$$C(W) = \{p_0, p_k\} \cup \{p_i \mid 0 < i < k \text{ and } \mathrm{type}(s_i) \neq \mathrm{type}(s_{i+1})\}.$$

$(\text{type}(s_i)=\text{poly if } s_i \in P(W) \text{ and type}(s_i)=\text{metal if } s_i \in M(W))$

We say that two wires W_1 and W_2 are *compatible* if there does not exist a pair of segments $s_i \in W_1$ and $s_j \in W_2$ such that s_i and s_j are incident on a common grid point and $\text{type}(s_i)=\text{type}(s_j)$. Notice that two compatible wires may "overlap" by connecting to common grid points with segments of different type, as illustrated in Figure 1.

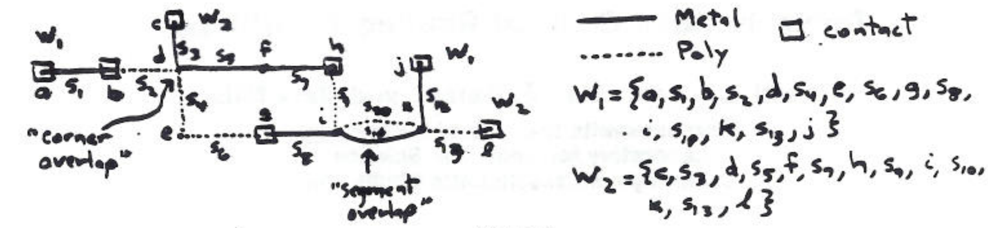

Figure 1.

Many previous channel routing algorithms employ a more restricted wiring model in which no such "overlap" is permitted. We do not know how to prove our current results without making use of a modest amount of overlap. The current model is certainly a realistic two-layer model, although it does permit wirings which are susceptible to "cross-talk" via the capacitive coupling of long overlapping wires. Our wirings will not have any long sections of overlapping wires - the longest such section will have length at most the width of the channel.

A *net* $N_i = (p_i, q_i)$ is an ordered pair of integers specifying an *entry (x-)coordinate* p_i and an *exit coordinate* q_i. A net is said to be *rising* if $q_i < p_i$, *falling* if $p_i < q_i$, and *trivial* if $p_i = q_i$. A *channel routing problem* is simply a set of n nets, for some integer n, such that no two nets have a common entry coordinate or a common exit coordinate. A *solution* to a channel routing problem consists of an integer t and a set of n compatible wires $W_1,\ldots,W_n$ in the channel of width t, such that W_i begins at grid point $(p_i, t+1)$ and ends at grid point $(q_i, 0)$. The *optimal width* for a channel routing problem is defined to be the least integer t such that the problem has a solution in a channel of width t.

For any real number x, we say that a net $N_i = (p_i, q_i)$ "crosses x" if either $p_i \leq x < q_i$ or $q_i \leq x < p_i$. The channel *density* of a channel routing problem is defined to be the maximum over all $x \in R$ of the number of nets crossing x. It is simple to show that a problem has optimal width at least $d/2$ if it has density d.

III. A Provably Good Channel Routing Algorithm

Let $CRP = \{N_1,\ldots,N_n\}$ denote any channel routing problem. We assume without loss of generality that $1 \leq p_i, q_i \leq m$ for all $1 \leq i \leq n$ and some integer m. Thus the nets $N_i \in CRP$ specify end-points which lie within some m "columns" of the channel. We will now describe a polynomial time algorithm which is guaranteed to compute a solution to CRP having channel width exactly $t = 2d-1$, where d is the channel density of CRP. Since $d/2$ is a lower bound on the optimal channel width for CRP, this algorithm will never generate a solution with channel width more than four times optimal.

Algorithm 1.

This algorithm proceeds column by column routing all nets which cross j in step j. The solution generated will have the properties that $t=2d-1$, there will be at most d wires passing from column j to column $j+1$ for any j, and for some j there will be at least d such wires. Further, wires will pass from a column j to column $j+1$ only on the odd-numbered tracks; there will be no horizontal segments on the even-numbered tracks. In addition, if there are k nets which cross j then there will be exactly k horizontal segments connecting columns j and $j+1$. These segments will all lie on distinct odd-numbered tracks and they may be of either type, poly or metal, independently. Finally, if exactly r of the k nets which cross j are rising and f are falling (so that $r+f=k$), then between columns j and $j+1$:

(1) The top-most r odd tracks will be devoted to wire segments for the r rising nets,
(2) The "middle" $d-r-f$ odd tracks will be empty, and
(3) The bottom-most f odd tracks will be devoted to wire segments for the f falling nets.

It now remains to demonstrate that this set of invariant properties can be maintained as the algorithm proceeds from column to column. If a column contains a trivial net, the net is wired straight across the column using the even numbered tracks to change layers as necessary. No other wiring is needed in such a column.

If a falling net $N_i=(p_i,j)$ enters column j from column j-1 on track t_z, the algorithm drops a vertical connection from grid point (j,t_z) down to grid point $(j,0)$. The algorithm then "closes up ranks" in column j so that all the empty odd tracks are in the middle of the channel. Figure 2 illustrates how such a wiring can be generated. Rising nets with entry coordinate j are handled similarly.

Finally, any rising net $N_i=(p_i,j)$ is routed in column j with a vertical connection from grid point $(j,0)$ up to grid point (j,t_w), where t_w is the top-most odd track which would be empty (i.e. contain no horizontal segment between grid points (j,t_w) and $(j+1,t_w)$) if net N_i were not present. Similarly any falling net is routed down to the lowest odd track that would otherwise be empty. If both of these situations occur in the same column, a modest amount of "overlap" is required as indicated in Figure 3. However, the situation of Figure 3 is the only place where overlap is needed.

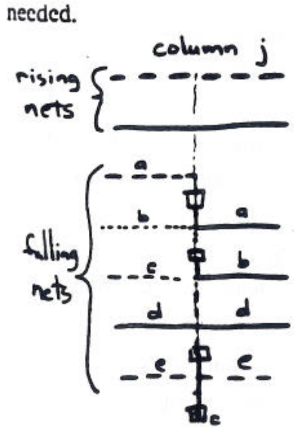

Figure 2.

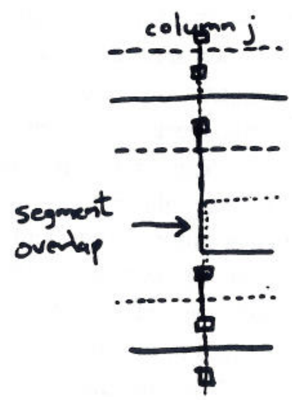

Figure 3.

Theorem 1:

Algorithm 1 is guaranteed to compute a solution to CRP having channel width no more than four times optimal.

Proof:

The proof of Theorem 1 follows directly from the above discussion.

At this point it should also be clear that the running time of Algorithm 1 is bounded by a polynomial in m, the number of columns which must be processed. In practice, however, we only need process columns having index equal to some net entry or exit coordinate. Thus with the appropriate output representation, Algorithm 1 is $O(d \cdot n)$ for a channel routing problem containing n nets and having density d.

Although Algorithm 1 never generates a solution with channel width more than four times optimal, it does generate solutions containing as many as $d \cdot n$ contact points. Further, it generates solutions containing overlapping parallel runs as long as length $2d$-1. In the remainder of this paper we present algorithms which cope with these two problems independently.

IV. Bounding the Number of Contacts

In this section we will describe a polynomial time algorithm which, like Algorithm 1, is guaranteed to compute a solution to *CRP* having channel width no more than four times optimal, but unlike Algorithm 1 requires no more than $4n$ total contact points. This new algorithm employs the same basic approach as Algorithm 1 and thus its description will be facilitated by simply noting the differences between the two algorithms.

Algorithm 2.

Similar to Algorithm 1, this algorithm proceeds column by column routing all nets which cross j in step j. Further, a solution generated by Algorithm 2 will have essentially the same properties as a solution generated by Algorithm 1 with only two significant exceptions. The first of these exceptions is that all horizontal segments belonging to wires of falling nets (with the possible exception of the top-most such segment in each column) will be of type metal. A similar property will hold for rising nets and poly horizontal segments. The second significant exception is that for each column j there may be at most one distinct horizontal segment which is associated with a falling net and connects columns j and $j+1$ while lying on an even-numbered track. Further, the net of such a segment will not have exit coordinate equal to $j+1$ and the odd-numbered track immediately below the segment will be empty between columns j and $j+1$. A similar property will also hold relative to rising nets.

The maintenance of this new set of invariant properties requires a somewhat different set of wiring rules from those employed by Algorithm 1. Consider the case where a falling net $N_i=(p_i,j)$ enters column j from column j-1 on track t_z. As in the previous algorithm, a vertical connection is dropped from grid point (j,t_z) down to grid point $(j,0)$. Notice, however, that at most one contact point will be required along this connection since all segments which must be crossed will have the same type. The algorithm must now "close up ranks" so that all blank columns remain in the middle of the channel. It should be clear that the technique employed by Algorithm 1 in solving this problem can be of no use here. However, the problem can be easily solved by dropping a vertical connection from the top-most track containing a falling net which crosses j down to grid point (j,t_z+1), as shown in Figure 4a. The only problem that occurs when the net to be dropped has exit coordinate equal to $j+1$. In this case, however, the algorithm simply drops the next lower net (if any) as shown in Figure 4b. Rising nets with entry coordinate j are handled similarly and all other cases are handled as in Algorithm 1.

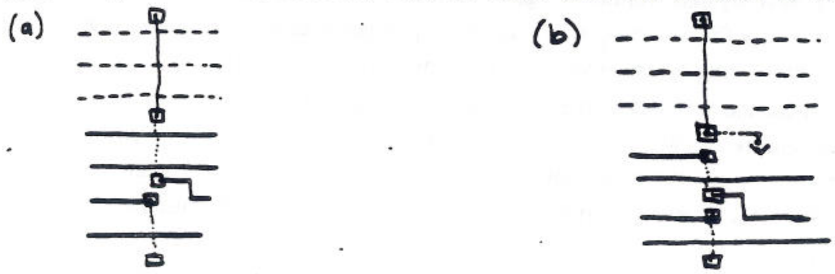

Figure 4.

Theorem 2:

Algorithm 2 is guaranteed to compute a solution to CRP with channel width no more than four times optimal and with no more than $4n$ total contact points.

Proof:

The proof of Theorem 2 follows from the above discussion and a more detailed case analysis of the wiring rules applied within each column.

Finally, we note that Algorithm 2 has time complexity $O(d \cdot n)$ for a channel routing problem containing n nets and having density d.

V. Reducing Overlap

Let us now assume that we wish to compute a solution to CRP which has minimal channel width and no segment overlap. In this section we will describe a polynomial time algorithm which is guaranteed to compute a solution to CRP having channel width no more than four times optimal and requiring only "corner overlap". However, the number of contact points required by this algorithm will be $O(d \cdot n)$ rather than $O(n)$.

Algorithm 3.

This algorithm proceeds track by track rather than column by column. The processing at each step involves a pair of adjacent tracks, i and $i+1$, such that i is odd. Furthermore, the algorithm proceeds bottom-up beginning with tracks 1 and 2. At each step the algorithm extends all existing wires across both track i and track $i+1$, in such a way that the density of the subproblem between track $i+1$ and the top of the channel decreases. This reduction in density will result from horozontal wire extension along the odd-numbered track. Once again the final solution will have the properties that $t=2d-1$ and there will be horizontal wire segments lying only on odd-numbered tracks: the even-numbered tracks will be used solely for layer changes along vertically running wires.

When the algorithm begins processing a pair of tracks i and $i+1$, there will exist exactly n distinct vertical segments connecting a grid point in track $i-1$ to a grid point in track i. Further, each of these segments will belong to a distinct wire. Since track $i-1$ is even-numbered and thus used solely for layer changes, we note that the type, poly or metal, of each of these segments can always be assigned as a function of the horizontal routing in track i. We will now describe the procedure for routing nets across track i.

The processing of track i is performed in either a *left-to-right* or a *right-to-left* fashion depending on how track i-2 was processed. The processing direction for track i is initially set to be the opposite of that for track i-2.

Let us assume that track i is to be processed in a left-to-right fashion: an analogous procedure is employed for the right-to-left case. Further, assume that column j_l is the left-most column containing a vertical segment connecting grid point $(j_l, i\text{-}1)$ to grid point (j_l, i) and belonging to a rising net $N_k=(p_k,q_k)$ for which $p_k > j_l$. Thus net N_k requires extension to the right. If no such column exists then track i is processed in a right-to-left fashion.

Now let W_k denote the wire associated with net N_k. Note that W_k ends at grid point (j_l, i). The algorithm then simply extends wire W_k horizontally to the right from grid point (j_l, i) until it reaches either column p_k (the entry coordinate of net N_k) or a column j_2 containing the terminus of a wire W_r for a net $N_r=(p_r,q_r)$ with $p_r > p_k$. (i.e. W_r is a wire which must be extended farther right than W_k.)

In the latter case wire W_k ends at column j_2 and wire W_r is extended to the right in a manner similar to the extension of W_k. In the former case wire W_k ends at column p_k and the algorithm searches to the right for the first wire requiring some extension.

Let column j_3 denote the left-most column (if any) such that $j_3 \geq p_k$ and the point (j_3, i) is the terminus of a wire W_s for a net $N_s=(p_s,q_s)$ with $p_s \neq j_3$. Thus wire W_s requires some horizontal extension; either to the right or to the left. Further, if $j_3 > p_k$ then N_s must additionally be a rising net so that W_s requires extension to the right. The wire W_s is then the next wire to be extended. The only difference in the manner of extension occurs when N_s happens to be a falling net. In this case W_s is extended to the right only until it reaches a column j_4 such that the point (j_4, i) is not the terminus of a wire for a net with entry coordinate equal to j_4. This will allow W_s to extend to the left, without generating segment overlap, when track $i+2$ is processed.

Once the processing of track i has been completed, all wires are extended vertically across track $i+1$ and the horizontal processing of track $i+2$ begins. The entire procedure for tracks i and $i+1$ is illustrated in Figure 5. Notice that a wire W for a net $N=(p,q)$ is never extended horizontally once its terminus lies in column p. Therefore, Algorithm 3 terminates when no further horizontal extension is necessary.

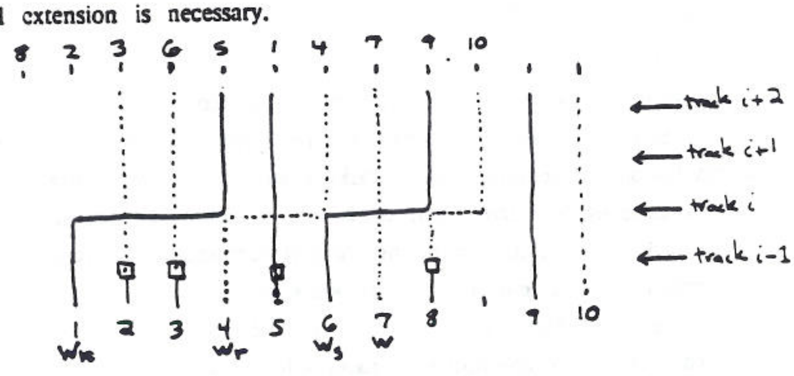

Figure 5.

Theorem 3:

Algorithm 3 is guaranteed to compute a solution to *CRP* with channel width no more than four times optimal and requiring only "corner overlap".

Proof:

It follows directly from the above discussion that Algorithm 3 will always generate a solution in which the only type of overlap is corner overlap. The upper bound on channel width then follows from the observation that the density between track i and the top of the channel is strictly decreasing as the algorithm proceeds and i increases.

We now point out that Algorithm 3, like the previous two algorithms, has time complexity $O(d \cdot n)$. Unlike the previous two algorithms, however, this algorithm may generate wires which are non-monotonic (i.e. weave back and forth across the channel), thus resulting in increased total wire length.

VI. Conclusions

We have presented three channel routing algorithms which are guaranteed to compute a wiring requiring no more than four times the optimal channel width. Furthermore, one of these algorithms requires only a small number of contact cuts and another requires only a minimal amount of overlap. However, many open questions still remain:

(1) Can the upper bound be improved (e.g. to $3d/2$)?

(2) Can this bound be proved in more restricted wiring models (e.g. the model of [D76])?

(3) Can this bound be proved for multi-terminal nets?

(4) Can both the number of contact cuts and the amount of overlap be simultaneously minimized?

VII. Acknowledgements

We would like to thank Charles Leiserson, Ron Pinter and Brenda Baker for helpful discussions.

VIII. References

[D76] Deutsch, D.N., "A Dogleg Channel Router," Proceedings of the 13th IEEE Design Automation Conference (1976), 425-433.

[DKSSU81] Dolev, D.; Karplus, K.; Siegel, A.; Strong, A. and Ullman, J.D., "Optimal Wiring Between Rectangles," 13th Annual ACM Stoc Proceedings (Milwaukee, 1981), 312-317.

[HS71] Hashimoto, A. and Stevens, J., "Wire Routing By Optimizing Channel Assignment Within Large Apertures," Proceedings of the 8th IEEE Design Automation Workshop (1971), 155-169.

[T80] Tompa, M., "An Optimal Solution to a Wire-routing Problem," 12th Annual ACM Stoc Proceedings (Los Angeles, 1980), 161-176.

This research was supported by NSF grant MCS78-05849 and by DARPA grant N00014-80-C-0622.