Minimizing Channel Density with Movable Terminals*

Ronald I. Greenberg† Jau-Der Shih‡

December 13, 1994

Abstract

We give algorithms to minimize density for VLSI channel routing problems with terminals that are movable subject to certain constraints. The main cases considered are channels with linear order constraints, channels with linear order constraints and separation constraints, channels with movable modules containing fixed terminals, and channels with movable modules and terminals. In each case, we improve previous results for running time and space by a factor of $L/\lg n$ and L, respectively, where L is the channel length, and n is the number of terminals.

1 Introduction

The channel routing problem has received a great deal of attention in VLSI layout design. In the usual model, terminals lie on grid points along two horizontal line segments which delimit the channel. Each terminal is labeled with a net number, and the problem is to connect terminals belonging to the same net, using horizontal and vertical wire segments in a grid of two layers, one reserved for horizontal wires and one for vertical wires. Nets can connect from one layer to another by way of a *via*; nets cannot intersect one another on the same layer. Figure 1 shows a routing of an example problem. We refer to each of the vertical grid lines as a column, while the horizontal grid lines are referred to as rows or tracks.

Usually, it has been assumed that the positions of terminals on each side (top and bottom) are fixed but that the distance between the sides (the channel width) can be varied, and the minimum width is sought. While determining the width required to route a channel is NP-complete [9], a good estimate in practice is the channel *density*, the maximum over all columns of the number of nets that must cross the column. In fact, many existing channel routers achieve widths that are usually within one of the density, e.g., [8]. (Focusing on density may also be appropriate when more than two interconnection layers are available, in which case the lower bound on width becomes density divided by the number of layers allowing horizontal routing; e.g., see [5] for multilayer channel routing.)

In this paper we consider the situation in which the orderings of the terminals along each side of the channel are fixed, but the exact positions may vary. There are a number of practical situations in which such flexibility arises [2], and it can lead to substantial reduction in channel density and width [2, 4]. When only the ordering of terminals on each side is fixed, Gopal, Coppersmith, and

*This work was supported in part by NSF grants CCR-9109550 and CCR-9321388.

†Electrical Engineering Department, University of Maryland, College Park, MD 20742 (rig@eng.umd.edu)

‡Department of Information Engineering, Kaohsiung Polytechnic Institute, Ta-Hsu, Kaohsiung, Taiwan 84008, Republic of China (jdshih@nas04.kpi.edu.tw)

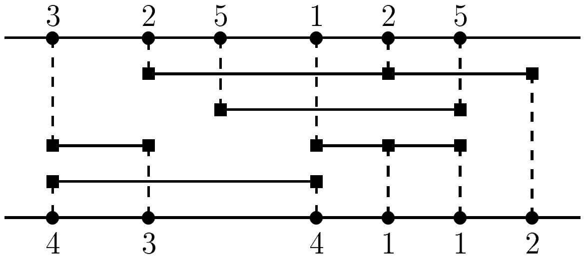

Figure 1: A representative channel routing problem in two layers. The horizontal wires (solid) are in one layer and the vertical (dashed) in the other layer. The vias are represented by squares and the terminals by circles.

Wong [4] give an $O(n^2)$ algorithm to minimize the width[1], where n is the number of terminals. LaPaugh and Pinter [7] presented an $O(n^2 \lg n)$ algorithm to minimize the channel density with the additional constraint that the relative positions of the terminals on each side are fixed. That is, the terminals lie on a single top module and a single bottom module, and the only freedom is to shift the modules relative to each other. More recently, Johnson, LaPaugh, and Pinter [6] provided an $O(n^3)$ algorithm to minimize density when there are multiple modules and terminal positions are fixed within each module, but the only other constraint is a fixed order for the modules on each side.

In the above works, however, the resulting channel length may be as large as $p+q$, where p is the number of top terminals and q is the number of bottom terminals (or as large as the sum of the module lengths in the module-based version of the problem). In contrast, Cai and Wong [1, 2] minimize density for a channel of fixed length L (perhaps as small as $\max\{p,q\}$) under a wide variety of constraints on the terminal positions. For channels with only *linear order constraints* (the orderings of the terminals on each side of the channel are fixed), they proposed an $O(pqL)$ algorithm to minimize the channel density. If we add *separation constraints* (the distance between each pair of consecutive terminals is within a certain range), their running time and space become $O(pqL^3)$ and $O(pqL^2)$, respectively. With multiple modules and fixed terminals within each module, they obtain $O(L^3)$ time and space. If the terminals within the modules are also movable, then the running time and space become $O(pqL^3)$.

In this paper we provide more efficient algorithms for these four problems of Cai and Wong [1, 2]. In each case, we improve the running time by a factor of $L/\lg(p+q)$ and the space by a factor of L. (Unlike Cai and Wong, however, we do not handle "position constraints", which specify a set of allowable columns for each terminal.) The third of these four problems can also be solved by a method of Chao and LaPaugh [3] that is discussed further and compared to our method in Section 7.

The remainder of this paper is organized as follows. In Section 2, we introduce some additional terminology and notation which will be used throughout this paper. Section 3 describes an algorithm to find the minimum channel density for channels with linear order constraints by using a dynamic programming approach. The algorithm is then extended in Sections 4, 5, and 6 to handle channels with separation constraints, channels with movable modules, and channels with movable modules and movable terminals, respectively. Finally, in Section 7, we provide some concluding

[1] This does not contradict the NP-completeness result, due to the use of a model in which there is complete freedom to choose the amount of space between adjacent terminals.

remarks.

2 Preliminaries

We begin by giving a more formal problem definition and some notation. We define $t_1, t_2, \ldots, t_p$ and $b_1, b_2, \ldots, b_q$ to be the terminals on the top and bottom side of the channel, which are ordered from left to right. We are given L column positions in which to place the terminals while retaining the given ordering on each side. The goal is to find the positions of the terminals such that the channel density is minimized.

Note that the density at any given column depends only on the fixed order of the terminals on each side and the position of that column within those orderings. Then let $d_1(i,j)$ be the density at the column of t_i when t_i is placed between b_j and b_{j+1}, let $d_2(i,j)$ be the density at the column of b_j when b_j is placed between t_i and t_{i+1}, and let $d_3(i,j)$ be the density at the column of t_i and b_j when they are aligned. These density functions can be computed in $O(pq)$ time for all possible i, j. The computation is a simple double loop over i and j; for example, $d_1(i+1,j)$ can be computed in constant time from $d_1(i,j)$ by looking at which terminals are connected to t_i and t_{i+1}. (If there are many terminals per net, we can perform a preprocessing step that removes all but the leftmost and rightmost terminal of each net on the top and bottom of the channel.) We assume throughout this paper that the d_1, d_2, and d_3 values have been computed and saved. Also, for any given target density d, we define an indicator variable $\delta_1^d(i,j)$ as follows

$$\delta_1^d(i,j) = \begin{cases} 1 & \text{if } d_1(i,j) \le d \\ \infty & \text{if } d_1(i,j) > d \ , \end{cases}$$

and we define $\delta_2^d(i,j)$ and $\delta_3^d(i,j)$ analogously. We use these δ values throughout our algorithms to express the feasibility, at a given density, of certain relative positionings of terminals.

The high-level structure of all our algorithms is as follows. Given a target density d, we compute the minimum channel length required to achieve the density. Based on the computed channel length and L, we increase or decrease the target density. By using a binary search on all the possible channel densities, we can find the minimum density achievable in length L.

3 Channels with Linear Order Constraints

In this section, we give an algorithm to minimize the channel density for channels with linear order constraints. We begin by showing how to find the minimum channel length at a given target density d. To do that, we introduce some subproblems used as the basis for a solution by dynamic programming. (We show in detail only how to find the minimum channel length, but one can readily retrace the computations leading to this result to determine the corresponding terminal placement.)

The length function $L^d(i,j)$ is defined to be the minimum number of columns spanned by top terminals $t_1, \ldots, t_i$ and bottom terminals $b_1, \ldots, b_j$, with the restriction that each of those columns has density at most d when all the other terminals are placed to the right of both t_i and b_j. If the target density d is unachievable, then $L^d(i,j)$ is defined to be ∞. We define $L_1^d(i,j)$ the same way as $L^d(i,j)$ but with the constraint that t_i is to the right of b_j. $L_2^d(i,j)$ and $L_3^d(i,j)$ are defined similarly but with the constraint that t_i is to the left of b_j, and t_i is aligned with b_j, respectively. We now show how to compute these functions recursively using the shorthand

$$L^d(i,j) = \min\{L_1^d(i,j), L_2^d(i,j), L_3^d(i,j)\} \ .$$

The final answer to our problem is $L^d(p,q)$.

Consider first the computation of $L_1^d(i,j)$. By the definition of $L_1^d(i,j)$, t_i must be to the right of b_j. Thus we require one column more than are spanned by $t_1, t_2, \ldots, t_{i-1}$ and $b_1, b_2, \ldots, b_j$, and we must check the density constraint in this new column:

$$L_1^d(i,j) = (L^d(i-1,j) + 1)\delta_1^d(i,j) \ .$$

Similarly, we can express $L_2^d(i,j)$ and $L_3^d(i,j)$ as

$$L_2^d(i,j) = (L^d(i,j-1) + 1)\delta_2^d(i,j)$$

and

$$L_3^d(i,j) = (L^d(i-1,j-1) + 1)\delta_3^d(i,j) \ .$$

For initial conditions, we have, for $c = 1, 2, 3$,

$$L_c^d(0,j) = j \prod_{k=1}^{j} \delta_c^d(0,k), \qquad j = 0, 1, \ldots, q \ ,$$

and

$$L_c^d(i,0) = i \prod_{k=1}^{i} \delta_c^d(k,0), \qquad i = 0, 1, \ldots, p \ ,$$

where we think of t_0 and b_0 as dummy terminals at the left of their respective sides that do not contribute to density.

Theorem 1 *Given a target density d, the minimum channel length subject to linear order constraints can be computed in $O(pq)$ time and space.*

Proof. We have already noted that the δ values can be computed in $O(pq)$ time, and an additional $O(p+q)$ time suffices to determine the initial conditions. Then we compute the values of the three length functions together in order of increasing i and j using the recurrences above. There is a total of $O(pq)$ values to compute, and each can be computed in $O(1)$ time from previously computed values. ■

Corollary 2 *The minimum density of a channel subject to linear order constraints can be found in $O(pq \lg(p+q))$ time and $O(pq)$ space.*

Proof. The minimum density problem can be solved by binary search on density, which is at most $p+q$. ■

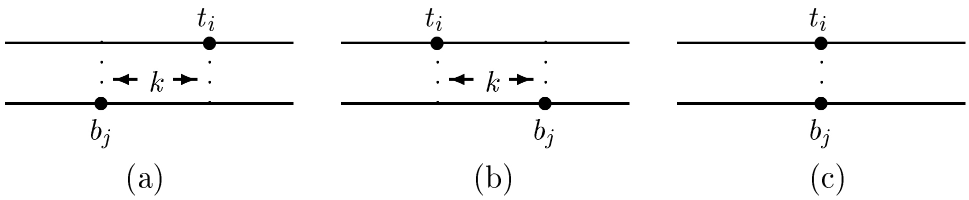

Figure 2: Three types of length functions: (a) $L_1^d(i,j,k)$ (b) $L_2^d(i,j,k)$ (c) $L_3^d(i,j)$

4 Channels with Linear Order Constraints and Separation Constraints

In this section, we extend the algorithm of Section 3 to handle channels with linear order constraints and separation constraints. Let the separation constraints have the following form: the distance s_i between t_i and t_{i+1} must satisfy $l_i \le s_i \le r_i$, and the distance s'_j between b_j and b_{j+1} must satisfy $l'_j \le s'_j \le r'_j$.

To handle the distance constraints, we have to modify the length functions. Let $L_1^d(i,j,k)$ and $L_2^d(i,j,k)$ be defined as in Section 3 but with the restriction that the horizontal distance between t_i and b_j equals k (in absolute value). We define $L_3^d(i,j)$ exactly as before. The constraints for the three length functions are illustrated in Figure 2. Then, $L^d(i,j)$ is obtained by minimizing over the three types of length functions and all possible k's.

Consider $L_1^d(i,j,k)$ first. There are three cases: (1) t_{i-1} is to the right of b_j, (2) t_{i-1} is to the left of b_j, and (3) t_{i-1} is aligned with b_j. And the minimum among the three cases is the minimum channel length. In the first case,

$$L_1^d(i,j,k) = \min_{k'}\{L_1^d(i-1,j,k') + k - k'\}\delta_1^d(i,j) \ ,$$

with $l_{i-1} \le k - k' \le r_{i-1}$. Figure 3(a) illustrates the restriction on k'. The second case can be analyzed similarly, and we have

$$L_1^d(i,j,k) = \min_{k'}\{L_2^d(i-1,j,k') + k\}\delta_1^d(i,j) \ ,$$

with $l_{i-1} \le k + k' \le r_{i-1}$. In the third case, which is possible only when $l_{i-1} \le k \le r_{i-1}$, we find

$$L_1^d(i,j,k) = (L_3^d(i-1,j) + k)\delta_1^d(i,j) \ .$$

The three cases are shown in Figure 3. In all cases, we have $0 < k < L$, and we assign a length function value of ∞ for values of k that are impossible given the other constraints.

From the above argument, $L_1^d(i,j,k)$ can be expressed as

$$L_1^d(i,j,k) = \begin{cases} (\min A_1)\delta_1^d(i,j) & \text{if } l_{i-1} \le k \le r_{i-1} \\ (\min A_2)\delta_1^d(i,j) & \text{otherwise} \end{cases}$$

where

$$A_1 = \{L_3^d(i-1,j) + k\} \cup A_2 \ ,$$

and

$$A_2 = \{\min_{l_{i-1} \le k-k' \le r_{i-1}} \{L_1^d(i-1,j,k') + k - k'\}, \min_{l_{i-1} \le k+k' \le r_{i-1}} \{L_2^d(i-1,j,k') + k\}\} \ .$$

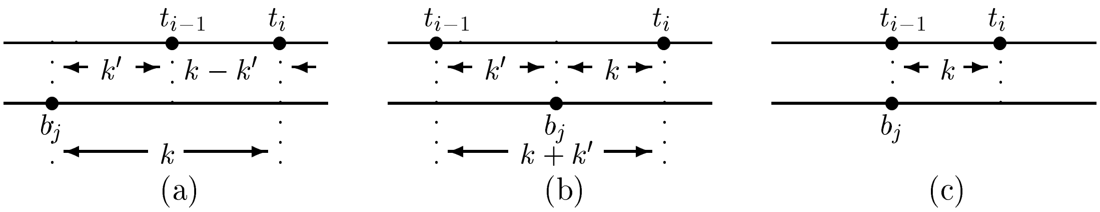

Figure 3: Three possibilities of $L_1^d(i,j,k)$: (a) t_{i-1} is to the right of b_j. (b) t_{i-1} is to the left of b_j. (c) t_{i-1} is aligned with b_j.

Similarly, $L_2^d(i,j,k)$ and $L_3^d(i,j)$ can be expressed as follows:

$$L_2^d(i,j,k) = \begin{cases} (\min B_1)\delta_2^d(i,j) & \text{if } l'_{j-1} \le k \le r'_{j-1} \\ (\min B_2)\delta_2^d(i,j) & \text{otherwise} \end{cases}$$

and

$$L_3^d(i,j) = \begin{cases} (\min C_1)\delta_3^d(i,j) & \text{if } [l_{i-1}, r_{i-1}] \cap [l'_{j-1}, r'_{j-1}] \ne \emptyset \\ (\min C_2)\delta_3^d(i,j) & \text{otherwise} \end{cases}$$

where

$$\begin{aligned} B_1 &= \{L_3^d(i,j-1)+k\} \cup B_2 \ , \\ B_2 &= \{\min_{l'_{j-1} \le k+k' \le r'_{j-1}} \{L_1^d(i,j-1,k')+k\}, \min_{l'_{j-1} \le k-k' \le r'_{j-1}} \{L_2^d(i,j-1,k')+k-k'\}\} \ , \\ C_1 &= \{L_3^d(i-1,j-1)+max\{l_{i-1}, l'_{j-1}\}\} \cup C_2 \ , \\ C_2 &= \{\min_{(k'',k') \in S_{i,j}} \{L_1^d(i-1,j-1,k')+k''\}, \min_{(k'',k') \in T_{i,j}} \{L_2^d(i-1,j-1,k')+k''\}\} \ , \\ S_{i,j} &= \{(k'',k') | l_{i-1} \le k'' \le r_{i-1} \text{ and } l'_{j-1} \le k''+k' \le r'_{j-1}\} \ , \end{aligned}$$

and

$$T_{i,j} = \{(k'',k') | l'_{j-1} \le k'' \le r'_{j-1} \text{ and } l_{i-1} \le k''+k' \le r_{i-1}\} \ .$$

Theorem 3 *Given a target density d, the minimum channel length subject to linear order constraints and separation constraints can be computed in $O(pqL^2)$ time and $O(pqL)$ space.*

Proof. We compute values of the length functions in order of increasing i, j and k, and then the minimum channel length is

$$\min\left\{ \min_{0<k<L} L_1^d(p,q,k), \min_{0<k<L} L_2^d(p,q,k), L_3^d(p,q) \right\} \ .$$

There are $O(pqL)$ values of L_1^d and L_2^d to be computed, and each can be computed from previously computed values in $O(L)$ time. In addition, there are $O(pq)$ values of L_3^d to be computed, each in time $O(L^2)$. ■

Corollary 4 *The minimum density of a channel subject to linear order constraints and separation constraints can be found in $O(pqL^2 \lg(p+q))$ time and $O(pqL)$ space.* ■

5 Channels with Movable Modules

This section considers the problem of channels with movable modules but with the terminals at fixed positions within their modules. We first augment the set of terminals to include the endpoints of the modules. Then we insert pseudo-terminals on the modules until every column in the modules contains a terminal or a pseudo-terminal as in [2]. As a result, the separation constraints between terminals inside a top module have the form $l_i = r_i = 1$ (an adjacency constraint), and the separation constraints between the right endpoint of a top module and the left endpoint of the module immediately to its right are $l_i = 1$, and $r_i = \infty$. (The constraints on the bottom are similar.) Now we can see this problem as a channel subject to linear order constraints and special separation constraints.

The length functions used in this section are as defined in Section 3. The approach to calculate these length functions is the same except for a modification to handle adjacency constraints. Using the notational shorthand

$$L^d_{x,y}(i,j) = \min\left\{L^d_x(i,j), L^d_y(i,j)\right\} \ ,$$

we have:

$$L^d_1(i,j) = \begin{cases} (L^d(i-1,j)+1)\delta^d_1(i,j) & \text{if } r_{i-1} = \infty \\ (L^d_{1,3}(i-1,j)+1)\delta^d_1(i,j) & \text{if } r_{i-1} = 1 \end{cases}$$

$$L^d_2(i,j) = \begin{cases} (L^d(i,j-1)+1)\delta^d_2(i,j) & \text{if } r'_{j-1} = \infty \\ (L^d_{2,3}(i,j-1)+1)\delta^d_2(i,j) & \text{if } r'_{j-1} = 1 \end{cases}$$

$$L^d_3(i,j) = \begin{cases} (L^d(i-1,j-1)+1)\delta^d_3(i,j) & \text{if } r_{i-1} = r'_{j-1} = \infty \\ (L^d_{1,3}(i-1,j-1)+1)\delta^d_3(i,j) & \text{if } r_{i-1} = 1 \text{ and } r'_{j-1} = \infty \\ (L^d_{2,3}(i-1,j-1)+1)\delta^d_3(i,j) & \text{if } r_{i-1} = \infty \text{ and } r'_{j-1} = 1 \\ (L^d_3(i-1,j-1)+1)\delta^d_3(i,j) & \text{if } r_{i-1} = r'_{j-1} = 1 \end{cases}$$

and

$$L^d(i,j) = \min\{L^d_1(i,j), L^d_2(i,j), L^d_3(i,j)\} \ .$$

Theorem 5 *Given a target density d, the minimum channel length for channels with movable modules can be computed in $O(L^2)$ time and space.*

Proof. We can compute $L^d_1(i,j)$, $L^d_2(i,j)$, and $L^d_3(i,j)$ from previously computed values in $O(1)$ time. Including the pseudo-terminals, there are $O(L)$ terminals on each side of the channel, which yields $O(L^2)$ length function values to be computed. ■

Corollary 6 *The minimum density with movable modules can be found in $O(L^2 \lg(p+q))$ time and $O(L^2)$ space.* ■

6 Channels with Movable Terminals and Modules

In this section, we consider channels with movable terminals and modules. That is, the modules on each side of the channel are movable as in Section 5, and we also allow the terminals to move within their modules. To handle this situation, we have to introduce new definitions and length functions.

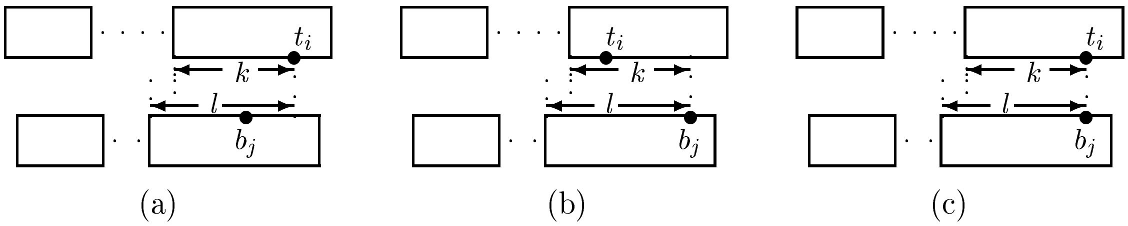

Figure 4: Three types of length functions: (a) $L_1^d(i,j,k,l)$ (b) $L_2^d(i,j,k,l)$ (c) $L_3^d(i,j,k,l)$

Define a *left terminal* to be the leftmost terminal of a module. Also define $M(p)$ to be the module where terminal p is located, v_i to be the length of $M(t_i)$, and w_j to be the length of $M(b_j)$. The length functions used here have four variables i, j, k, and l as illustrated in Figure 4; here k and l represent the distance from the rightmost of t_i and b_j to the left edges of their modules. The length function $L^d(i,j)$ is equal to the minimum of the three types of length functions for all possible k's and l's (where each length function accounts for the lengths of the modules containing $t_1, t_2, \ldots, t_i$ and $b_1, b_2, \ldots, b_j$).

For many values of k and l, we can immediately set length function values to ∞. For example, if terminal t_i is the mth terminal in its module, then $L_1^d(i,j,k,l) = \infty$ for any $k < m-1$. In what follows we give recurrences for the length functions under the assumption that such restrictions have already been taken into account.

To simplify the presentation, we define notational shorthand as in Section 5:

$$L_{x,y}^d(i,j,k,l) = \min\left\{L_x^d(i,j,k,l), L_y^d(i,j,k,l)\right\}$$

and

$$L^d(i,j,k,l) = \min\{L_1^d(i,j,k,l), L_2^d(i,j,k,l), L_3^d(i,j,k,l)\} \ .$$

We first consider $L_1^d(i,j,k,l)$. There are two cases according to whether t_i is a left terminal or not. We seek the minimum among the channel lengths obtained in the following three subcases: (1) t_{i-1} is to the right of b_j, (2) t_{i-1} is to the left of b_j, and (3) t_{i-1} is aligned with b_j. Note that if the relative position of $M(t_i)$ and $M(b_j)$ is fixed, then the actual positions of the terminals on the two modules have no effect on the value of the length functions as long as the density is less than or equal to d.

Case (A): t_i is not a left terminal.

(1): In the subcase where t_{i-1} is to the right of b_j, we know that we can place t_{i-1} in the column just before t_i, since t_{i-1} and t_i are on the same module, and the definition of $L_1^d(i,j,k,l)$ implies that there are no bottom terminals between b_j and t_i. Thus we have

$$L_1^d(i,j,k,l) = L_1^d(i-1,j,k-1,l-1)\delta_1^d(i,j) \ .$$

(2) and (3): In the subcases where t_{i-1} aligned with or to the left of b_j, we know that we can place b_j in the column just before t_i if $w_j \geq l-1$; otherwise, we can place b_j at the right end of its module.

Putting the subcases together, we have

$$L_1^d(i,j,k,l) = \begin{cases} L^d(i-1,j,k-1,l-1)\delta_1^d(i,j) & \text{if } w_j \geq l-1 \\ \min\{L_1^d(i-1,j,k-1,l-1), L_{2,3}^d(i-1,j,w_j+k-l,w_j)\}\delta_1^d(i,j) & \text{if } w_j < l-1 \end{cases}$$

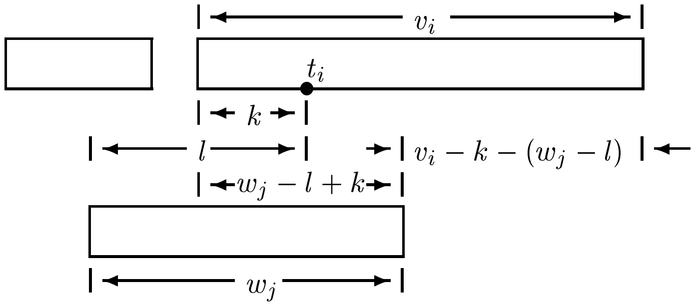

Figure 5: This figure shows how to calculate the channel length when t_i is a left terminal.

Case (B): t_i is a left terminal.

(1): In the subcase where t_{i-1} is to the right of b_j, we know that we can push t_{i-1} to the right edge of its module, giving us

$$L_1^d(i,j,k,l) = \min_{l'<l-k}\{L_1^d(i-1,j,v_{i-1},l')\}\delta_1^d(i,j) + \max\{0, v_i - k - (w_j - l)\} \ .$$

The term added at the end accounts for the possible increase in channel length when module $M(t_i)$ is included, as shown in Figure 5.

(2): In the subcase where t_{i-1} is to the left of b_j, we know that we can place b_j in the column just before t_i if $w_j \geq l-1$; otherwise we can push b_j to the right edge of its module.

(3) In the subcase where t_{i-1} is aligned with b_j, we can push t_{i-1} to the right edge of its module if $l - w_j \leq k$; otherwise we can push b_j to the right edge of its module.

Putting the subcases together gives:

$$\begin{aligned} L_1^d(i,j,k,l) &= \min_{l',k',k'',l''}\{L_1^d(i-1,j,v_{i-1},l'), L_2^d(i-1,j,k',\min\{w_j,l-1\}), L_3^d(i-1,j,k'',l'')\}\delta_1^d(i,j) \\ &\quad + \max\{0, v_i - k - (w_j - l)\} \ , \end{aligned}$$

where $l' < l-k$, $k' > v_{i-1} + k + \min\{w_j - l, -1\}$, and k'' and l'' are defined as follows. If $l - w_j > k$, then $l'' = w_j$ and $k'' > v_{i-1} + w_j + k - l$. If $l - w_j \leq k$, then $k'' = v_{i-1}$ and $l'' < l - k$.

We can write recurrences for L_2 in a fashion similar to L_1. When b_j is not a left terminal,

$$L_2^d(i,j,k,l) = \begin{cases} L^d(i,j-1,k-1,l-1)\delta_2^d(i,j) & \text{if } v_i \geq k-1 \\ \min\{L_2^d(i,j-1,k-1,l-1), L_{1,3}^d(i,j-1,v_i,v_i+l-k)\}\delta_2^d(i,j) & \text{if } v_i < k-1 \end{cases}$$

When b_j is a left terminal,

$$\begin{aligned} L_2^d(i,j,k,l) &= \min_{k',l',l'',k''}\{L_2^d(i,j-1,k',w_{j-1}), L_1^d(i,j-1,\min\{v_i,k-1\},l'), L_3^d(i,j-1,k'',l'')\}\delta_2^d(i,j) \\ &\quad + \max\{0, w_j - l - (v_i - k)\} \ , \end{aligned}$$

where $k' < k-l$, $l' > w_{j-1} + l + \min\{v_i - k, -1\}$, and l'' and k'' are defined as follows. If $k - v_i > l$, then $k'' = v_i$ and $l'' > w_{j-1} + v_i + l - k$. If $k - v_i \leq l$, then $l'' = w_{j-1}$ and $k'' < k - l$.

Finally, we consider L_3. It is easy to see that when t_i is not a left terminal,

$$L_3^d(i,j,k,l) = L_2^d(i-1,j,k,l)\delta_3(i,j) \ .$$

Similarly, when b_j is not a left terminal,

$$L_3^d(i,j,k,l) = L_1^d(i,j-1,k,l)\delta_3(i,j) \ .$$

Finally, if t_i and b_j are both left terminals,

$$L_3^d(i,j,k,l) \quad = \quad \min_{w_{j-1}+l<l'} L_1^d(i,j-1,k,l') + \max\{0, w_j - l - (v_i - k)\} \ .$$

Theorem 7 *Given a target density d, the minimum channel length problem for channels with movable modules and terminals can be computed in $O(pqL^2)$ time and space.*

Proof. All the length functions $L_1^d(i,j,k,l)$, $L_2^d(i,j,k,l)$, and $L_3^d(i,j,k,l)$ can be computed from the previously computed values in $O(1)$ time because all the minimizations appearing in our recurrences can be performed on the fly. In fact the minimizations never depend on the values of both k and l; for example the minimization over $l' < l - k$ needs only be performed for each value of $l - k$, and there is no need for more than $O(1)$ extra storage as long as these minimizations are peformed in order of the value of $l - k$. There is a total of $O(pqL^2)$ length functions, which yields the stated running time and space. ■

Corollary 8 *The minimum density of a channel with movable modules and terminals can be solved in $O(pqL^2 \lg(p+q))$ time and $O(pqL^2)$ space.* ■

7 Conclusion and Extensions

We have presented algorithms to minimize the channel density for a variety of problems. These algorithms improve the previous known results by $O(L/\lg(p+q))$ in running time and $O(L)$ in space. These algorithms can also easily be extended to channels with exits or channels with irregular boundaries as in [1] without increasing the complexity. In the process of minimizing density for a fixed channel length, we have provided even more efficient algorithms to minimize length at a fixed density. By running the latter type of algorithm $O(p+q)$ times, we can also minimize more complex cost measures, such as area (where density is treated as width) in a channel of length at most L. We can also improve the space bound for our algorithms to find minimum channel length or minimum density if we are not worried about recovering the actual terminal placement. Since the length function values for a given sum of i and j depend only on values with a lesser sum of i and j, we need only store the values for one previous sum at a time. Thus all the space requirements decrease by a factor of $\max\{p,q\}$ (or L for the case of movable modules with fixed terminals).

For the case of movable modules with fixed terminals, density can be minimized in a channel of length L in $O(n^3 \lg n)$ time independent of L (which improves upon the time in Section 5 for $L > n^{3/2}$) using the method of Chao and LaPaugh [3]. Like our approach, this would involve using binary search along with a dynamic programming method that determines minimum channel length for a fixed density [3, p. 4]. Their length functions include one more parameter than ours, and they require a more complicated method to compute each value quickly, including a preprocessing step to analyze the overlap of individual pairs of modules. Their method cannot be extended to handle channels with movable terminals as well as movable modules [3, p. 44]. Obviously, their method can be applied to the problem considered in Section 3 (linear order constraints for independent terminals) by thinking of each terminal as a module by itself, but the running time is never as good as in Section 3. Their method may be applicable to the problem considered in Section 4 (with separation constraints), but the running time would be worse than the $O(n^3 \lg n)$ time obtained in the other case [3, p. 44]. An interesting open question is to solve the problems of Sections 4 and 6 in time polynominal in n only.

References

[1] Yang Cai and D. F. Wong. Minimizing channel density by shifting blocks and terminals. In *IEEE International Conference on Computer-Aided Design (ICCAD-91)*, pages 524–527. IEEE Computer Society Press, 1991.

[2] Yang Cai and D. F. Wong. Optimal channel pin assignment. *IEEE Trans. Computer-Aided Design of Integrated Circuits*, 10(11):1413–1424, November 1991.

[3] Liang-Fang Chao and Andrea S. LaPaugh. Finding all minimal shapes in a routing channel. Technical Report CS-TR-384-92, Princeton University Department of Computer Science, August 1992.

[4] Inder S. Gopal, Don Coppersmith, and C.K. Wong. Optimal wiring of movable terminals. *IEEE Trans. Computers*, C-32(9):845–858, September 1983.

[5] Ronald I. Greenberg, Alexander T. Ishii, and Alberto L. Sangiovanni-Vincentelli. MulCh: A multi-layer channel router using one, two, and three layer partitions. In *IEEE International Conference on Computer-Aided Design (ICCAD-88)*, pages 88–91. IEEE Computer Society Press, 1988.

[6] D. S. Johnson, A. S. LaPaugh, and R. Y. Pinter. Minimizing channel density by lateral shifting of components. In *Proceedings of the 5th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 122–131, 1994.

[7] Andrea S. LaPaugh and Ron Y. Pinter. On minimizing channel density by lateral shifting. In *IEEE International Conference on Computer-Aided Design (ICCAD-83)*, pages 123–124. IEEE Computer Society Press, 1983.

[8] James Reed, Alberto Sangiovanni-Vincentelli, and Mauro Santomauro. A new symbolic channel router: YACR2. *IEEE Trans. Computer-Aided Design of Integrated Circuits*, CAD-4(3):208–219, July 1985.

[9] Thomas G. Szymanski. Dogleg channel routing is NP-complete. *IEEE Trans. Computer-Aided Design of Integrated Circuits*, CAD-4(1):31–41, January 1985.