To the Graduate Council:

I am submitting herewith a thesis written by Leanne Tuggle Metzger entitled "Optimizing Total Wire Length for Internal-External Routings." I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Computer Science.

Jean R. S. Blair

Dr. Jean R. S. Blair, Major Professor

We have read this thesis

and recommend its acceptance:

Dave Mutchler

Heather Booth

Accepted for the Council:

Carolyn Winkel

Associate Vice Chancellor
and Dean of The Graduate School

STATEMENT OF PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for a Master's degree at The University of Tennessee, Knoxville, I agree that the Library shall make it available to borrowers under rules of the Library. Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgement of the source is made.

Permission for extensive quotation from or reproduction of this thesis may be granted by my major professor, or in her absence, by the Head of Interlibrary Services when, in the opinion of either, the proposed use of the material is for scholarly purposes. Any copying or use of the material in this thesis for financial gain shall not be allowed without my written permission.

Signature Leanne Juggle Metz

Date 4/29/94

Optimizing Total Wire Length for Internal-External Routings

A Thesis

Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

Leanne Tuggle Metzger

May 1994

Dedication

This thesis is dedicated to my husband, Brad Metzger, for his unfailing support and encouragement.

Acknowledgements

I would like to thank my major professor, Dr. Jean R. S. Blair, for her direction in the research and writing of this thesis. In addition, I would like to thank Dr. Heather Booth and Dr. David Mutchler for their valuable suggestions. I would like to thank my parents, Lois Hart and Joe Tuggle, for their interest and encouragement. Finally, I would like to give my reverent thanks to God for giving me the strength to meet this challenge.

Abstract

The architecture of present day computers consists of very-large-scale integrated (VLSI) chips. VLSI technology is used in designing high-performance multiprocessors for the purpose of parallel computing. In addition, VLSI technology is used in the design of semi-conductor storage to achieve fast memory access. Of particular interest here is the use of computers to aid in the routing stage of VLSI design. We focus on one aspect of this routing process, namely, the river routing problem.

Popular classes of routings for the river routing problem include internal routings, internal-external routings, and mixed routings. Each class is a superset of the preceding class. Optimization criteria typically used in the VLSI river routing problem are area of enclosing rectangle, vertical density (number of tracks), number of jogs, maximum individual wire length, and total wire length. An algorithm exists to produce internal routings with minimum total wire length, but total wire length has not been considered as an optimization criterion for internal-external or mixed routings. Here, we present a new polynomial time algorithm that produces internal-external routings with minimum total wire length.

Contents

List of Figures

1 Introduction

Given two parallel rows of n terminals each the basic river routing problem is to connect in a single layer each corresponding pair of terminals. The traditional river routing model restricted the connecting wires to paths between the terminal rows (*internal routings*) [4, 5, 6, 8, 9, 12]. In [7] this restriction was relaxed to allow wires to be routed outside the terminal rows (*internal-external routings*). In [2] the restrictions on the routings of wires were even further relaxed (*mixed routings*). The lifting of restrictions has produced models that allow for richer classes of routings.

The goal of the river routing problem is to produce a routing in which the corresponding pairs of terminals are wired together in an *optimal* fashion. Optimization criteria typically used in the VLSI river routing problem are area of enclosing rectangle, vertical density(number of tracks used), number of jogs, maximum individual wire length, and total wire length. Total wire length has been used as an optimization criteria for the class of internal routings. However, the literature [2, 7] pertaining to the internal-external routings and the mixed routings has concentrated on the area of the enclosing rectangle and vertical density, two closely related criteria. Here, we focus on the internal-external routings with total wire length as the optimization criterion. The motivation for using total wire length as an optimization criterion is that long wires have greater resistance,

therefore short wire lengths are desirable.

First, we review the algorithms presented in [2, 7] for producing internal-external routings. The method used in the algorithm produced in [2, 7] is the *greedy routing method*, and it produces *greedy internal-external routings*. We establish that this existing algorithm does not produce an optimal routing (minimum total wire length), but we then define a relationship between the greedy internal-external routing and an optimal routing. From this relationship, we develop a polynomial time algorithm to produce an optimal internal-external routing.

In the next section we describe the traditional river routing model and the no component constraint model, and we also introduce the necessary terminology and notation. Section 3 reviews the algorithm presented in [2, 7]. A discussion of optimization criteria and an objective function for minimizing total wire length is presented in Section 4. Our algorithm is then presented in Section 5.

2 Background

Many of the descriptions and definitions in this section follow closely those given in [2, 7].

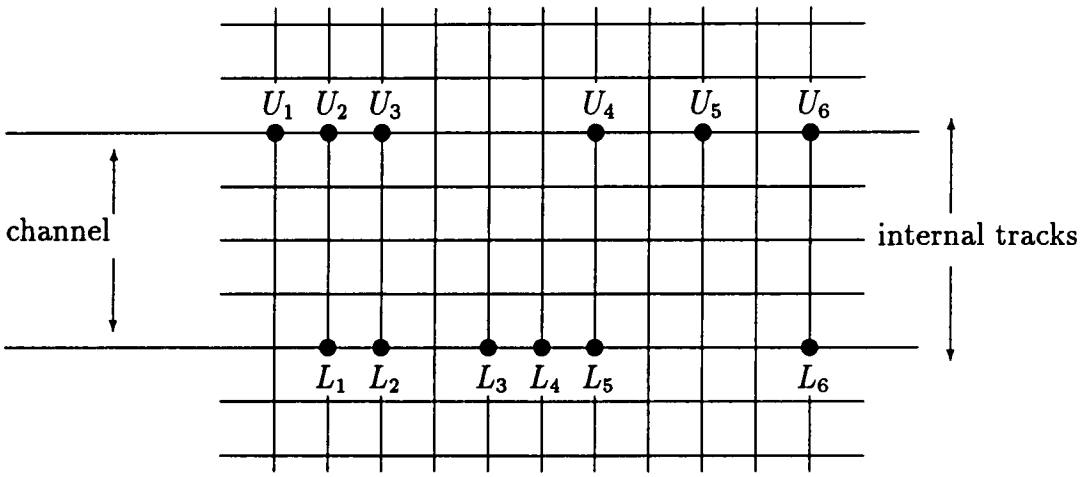
2.1 The River Routing Problem and Terminology

The *basic configuration* for the *river routing* problem consists of a unit square grid with terminals restricted to grid intersection points and arranged in two rows parallel to the x-axis. Interconnections are made in a single layer using wires that are restricted to paths along the grid lines.

An instance of the *river routing problem* consists of two horizontal rows of n terminals, with fixed horizontal positions for the terminals in each row (Figure 1). The i^{th} terminal, from left to right, of the upper (lower) row is denoted by U_i (L_i). For each $1 \leq i \leq n$, the pair of terminals (U_i, L_i) make up *net* i. Wire W_i connects net i.

The leftmost terminal has a *horizontal position* of zero. The horizontal position of each other terminal (on either row) is equal to the number of vertical grid lines that the terminal is away from this leftmost terminal. The symbol U_i (L_i) will be used to represent the horizontal position of the terminal U_i (L_i). For the instance shown in Figure 1, $U_1 = 0$ and $U_4 = L_5 = 6$.

The *offset* of net i is $|U_i - L_i|$. Net i is *offset-by-one* if $|U_i - L_i| = 1$. Net i has a



Blocks:
$\{(U_1, L_1), (U_2, L_2), (U_3, L_3)\}$ – a left block
$\{(U_4, L_4), (U_5, L_5)\}$ – a right block
$\{(U_6, L_6)\}$ – a straight block

Figure 1: Basic configuration.

right orientation if $U_i > L_i$. Likewise, it has a *left (straight) orientation* if $U_i < L_i$ ($U_i = L_i$). None of the results presented in this paper rely on any properties that are inherent to a particular orientation (right, left, or straight) of the nets. Thus, in presenting the results we will assume that all nets are right oriented. The modifications necessary for left and straight oriented nets are straight-forward.

A *block* is a maximal set of consecutive nets such that every net in that block has the same orientation. A block is a *right (left, straight) block* if the nets in the block are right (left, straight) oriented nets (Figure 1).

Every horizontal grid line is a *track*. The tracks between the terminal rows, as well as the terminal rows themselves, are *internal tracks*. The region defined by the two tracks containing the terminal rows and the two vertical lines passing through the leftmost and rightmost terminals is the *base*.

The horizontal routing region between the terminal rows is the *channel*. The *channel width* is the number of tracks between the upper terminal row and lower terminal row. The upper terminal row is defined as track 0. In Figure 2 the lower terminal row is track 9, thus the channel width for this instance is 9. The *separation* is the number of internal tracks utilized in the routing. If routing is allowed on the terminal rows then the separation is equal to the channel width, otherwise given a channel width s the separation is $s - 1$. Thus, in Figure 2 the separation 8. The channel width for a particular river routing instance may be a

fixed channel width or a *variable channel width*, that is, we have the freedom to specify the channel width.

By restricting the nature of wires in a routing, we obtain a class of routings. The traditional river routing model requires that every interconnecting wire be internal to the *channel.* An interconnecting wire that is internal to the base is an *internal wire.* A routing that consists solely of internal wires is an *internal routing* (Figure 2).

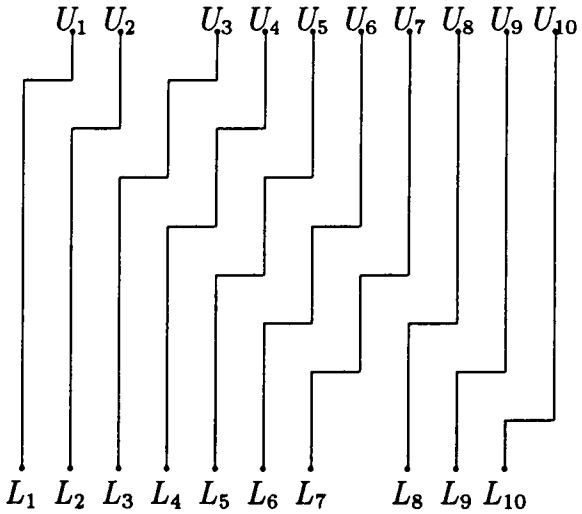


Figure 2: Internal Routing

2.2 The No Component Constraint Model

In the *no component constraint (NCC) model* it is assumed that the only objects (other than the wires) in the routing layer are the terminals. That is, the active

circuits of each component are effectively insulated from the routing layer. Unlike the traditional river routing models, this allows wires to travel along paths that utilize the area above the upper terminal row and below the lower terminal row, as well as utilizing the area between the terminal rows. To simplify the presentation, the model does not allow wires with horizontal segments on the terminal rows.

The arguments for restricting wires to the area between the terminal rows (the traditional model) are based on the assumption that the modules where the terminals originate also utilize the routing layer. However, this need not be the case. Wires can be routed **outside** the terminal rows without placing any additional restrictions on the layout of the modules (e.g. in a nMOS VLSI chip) if the terminal rows are extended away from the modules by a layer (e.g. polysilicon) electrically unrelated to the layer (e.g. metal) used for routing.

In the NCC model (Figure 3) the entire routing area is divided into three horizontal regions. Now, in addition to the *channel* between the terminal rows for routing there is the *upper channel* (area above upper terminal row), and the *lower channel* (area below lower terminal row). The tracks in the upper channel and lower channel are *external tracks.*

The NCC model allows for an additional class of wires. Wires are classified according to the region(s) they intersect. As stated in the previous section, an *internal wire* is routed completely within the base. A wire that intersects the

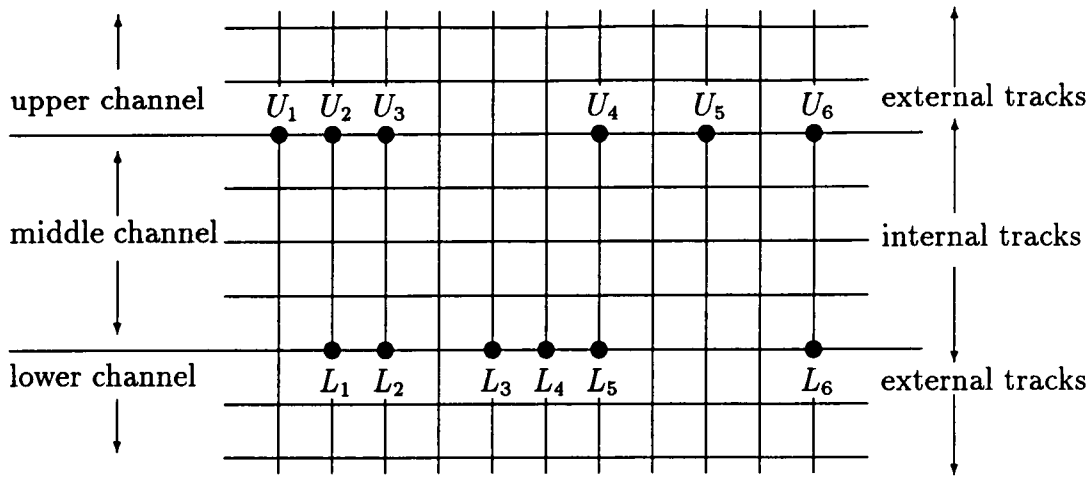


Figure 3: No Component Constraint Model

base only at its endpoints is an *external wire*. By allowing external wires, the NCC model allows for a richer class of routings. A routing consisting of internal wires and external wires is an *internal-external routing* (Figure 4). The class of internal-external routings forms a superset of the class of internal routings.

The NCC model is a more attractive model than the traditional river routing model because it is less restrictive. Using the traditional model with a fixed channel width, it may be that no internal routing exists for a particular instance. The same instance using the NCC model may have a valid internal-external routing. Thus, river routing instances with a fixed channel width that could not be routed under the traditional model are now routable using the NCC model.

There are several additional considerations which make this model

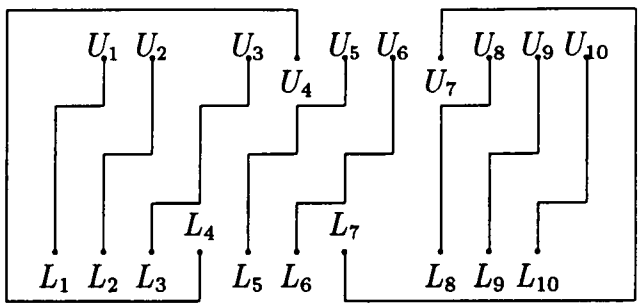


Figure 4: Internal-External Routing

attractive[2]. First, the model is consistent with the "one-active-level" realization of three-dimensional VLSI proposed by Rosenberg[11]. In that realization, active components are present only on one level of the chip and all additional levels are used solely for routing. Second, the generalized model lends itself to the routing of printed circuit boards (PCBs), since the absence of constraints on the layout of the modules is inherent to the routing problem for PCBs. Finally, the generalized model is consistent with the models typically used for single row problems [1, 10]. This consistency is useful not only for the additional variety it affords, but also because it places the one and two row problems on a uniform footing. (To date, routing techniques have not carried over from one type of problem to the other).

3 Greedy Routing Method

In Section 2.1 and Section 2.2 we reviewed two types of river routing models, and the classes of routings they allow. In this section, we review a popular method for producing routings. This method is known as the *greedy routing method*, and the routings it produces are referred to as *greedy routings*. This method was introduced in [4] and later used in [2, 7]. We begin our discussion by reviewing the greedy method for internal routings and the concept of conflict numbers. Then in Subsection 3.2 we examine how the technique is used to produce internal-external routings, and we discuss how the concept of conflict numbers is expanded so it is applicable to routings with external wires. In Subsection 3.1 and Subsection 3.2 we limit our dicussion to instances with fixed channel width.

3.1 Internal Routings and Conflict Numbers

Here, we consider the greedy routing method for producing internal routings only for instances with n terminals that have a fixed channel width s. The algorithm used in [4, 7] divides the process into two steps.

1. Find the minimum separation for which an internal routing exists. The minimum separation can be found in $O(n)$.

2. Compare the minimum separation required with the fixed channel width s. If $s \leq$ the minimum separation then state no internal routing exists, otherwise produce the routing in the following manner. Route each wire W_i, starting at $i = 1$ and proceeding right until the last net $i = n$, as follows: Beginning with U_i, a wire drops one track and then travels left as far as it can without intersecting another wire or the vertical grid line containing L_i, it drops vertically until it can again start travelling left; otherwise it drops vertically to meet its lower row terminal. This is repeated until the net i is connected. This is the *greedy method*, and it requires $O(n^2)$ time in the worst case.

The method for finding the minimum separation described in [4] and utilized in [2, 7] is based on the concept of conflict numbers. The description below of conflict numbers is taken directly from [2].

For two nets i and j (in a right block[1]) the *conflict number* $c(i,j)$ is defined as follows:

$$c(i,j) = \begin{cases} 0 & \text{if } (i < j \text{ and } U_i - i + j \leq L_j) \text{ or } (i > j) \\ j - i + 1 & \text{otherwise.} \end{cases}$$

Intuitively, a conflict number $c(i,j)$ represents the number of internal wires, between the i^{th} and j^{th} wires inclusive, that force the placement of the lowest hori-

[1]The remainder of the definition presented in [4] pertains to left and straight blocks.

zontal segment of wire j in a greedy routing. That is, it is the number of tracks necessary to route internal wires i through j. For example, in the instance shown in Figure 5(a), $c(2,4) = 3$ and 3 tracks would be necessary to route wires 2 through 4 in isolation.

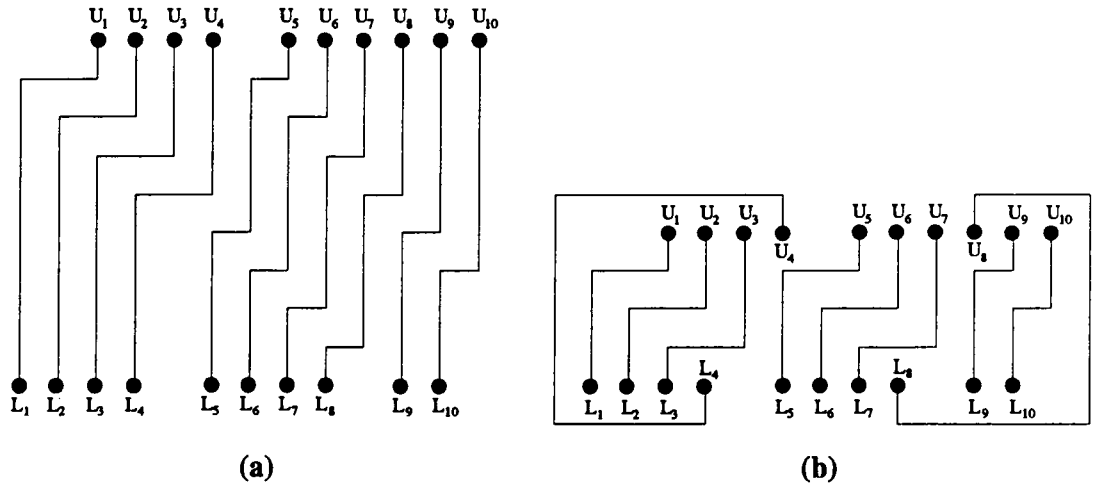


Figure 5: An example instance to illustrate conflict numbers. (a) An internal routing. (b) An internal-external routing.

Conflict numbers relate specific information about the tracks containing horizontal segments in a greedy internal routing. The following is the formal statement of this property as presented in [2].

Property 1 *The conflict number $c(i,j) \neq 0$ if and only if in a greedy routing there is a horizontal segment of wire j that is on the $c(i,j) - 1^{st}$ track below the rightmost horizontal segment of wire i.*[2]

[2]Since nets are right oriented, the "rightmost horizontal segment of wire i" in a greedy routing is the one with a right endpoint at horizontal position U_i.

For net j, the *maximum conflict number* $C(j)$ is defined as:

$$C(j) = \max\{c(i,j) \text{ for } 1 \leq i \leq n\}.$$

Note that $C(j) = c(i,j)$ for only one index i. The symbol $\rho(j)$ will represent the i for which $C(j) = c(i,j)$. Columns 4 and 5 respectively of Table 1 give the $\rho()$ and $C()$ values for all of the nets in the instance shown in Figure 5.

Table 1: Original and modified conflict numbers for nets in the example.

j	U_j	L_j	$\rho(j)$	$C(j)$	$\chi_I(j)$	$\pi_I(j)$	$\rho_I(j)$	$C_I(j)$
1	2	0	1	**1**	1	1	1	**1**
2	3	1	1	**2**	1	2	1	**2**
3	4	2	1	**3**	1	3	1	**3**
4	5	3	1	**4**	0	—	—	—
5	7	5	1	**5**	1	4	5	**1**
6	8	6	1	**6**	1	5	5	**2**
7	9	7	1	**7**	1	6	5	**3**
8	10	8	1	**8**	0	—	—	—
9	11	10	5	**5**	1	7	9	**1**
10	12	11	5	**6**	1	8	9	**2**

The minimum separation for an internal routing is equal in number to the largest maximum conflict number. This result relating minimum separation and maximum conflict numbers easily follows from Property 1. This property also holds for greedy internal-external routings. The intuition behind this result is

that if wires 1 through $j-1$ are routed internal in a greedy fashion, then wire j can be routed internal if at least $C(j)$ tracks are available in the channel. For the example, the minimum separation is $C(8) = c(1,8) = 8$, indicating that 8 tracks are necessary to route the wires internally. The reader is referred to [4] for details.

3.2 Internal-External Routings and Modified Conflict Numbers

Here, we describe the greedy method for producing internal-external routings for an instance with a fixed channel width s.

1. Using the modified conflict numbers as described below, each wire is marked internal or external. Once every wire has been marked the routing proceeds.

2. Each internal wire is routed in the same fashion described in Subsection 3.1.

3. Each external wire is routed around one of the ends of the base. More formally, let m be the number of external wires and let j be any number such that $1 \leq j \leq m$. Each external wire E_i for $1 \leq i \leq j$ is routed by proceeding upward from U_i until it reaches the i^{th} external track above the upper terminal row, then it moves left until it reaches the i^{th} vertical grid line past L_1, then it drops to the i^{th} external track below the lower terminal row, then it moves right until it reaches the vertical grid line containing its

lower terminal L_i, then it moves upward until it reaches the lower terminal row. Each external wire E_k for $j+1 \leq k \leq m$ is routed by proceeding upward from U_k until it reaches the $(m-k+1)^{th}$ external track above the upper terminal row, then it moves right until it reaches $(m-k+1)^{th}$ vertical grid line past U_n, then it drops until it reaches the $(m-k+1)^{th}$ external track below the lower terminal row, then it moves left until it reaches the vertical grid line containing L_k, then it moves upward to the lower terminal row.

Here, we describe the modified version of conflict numbers presented in [2]. The modification made in [2] was to account for the existence of non-internal (external) wires. Let $\mathrm{I} \subseteq \{1,\ldots,n\}$ be the subset of wires that are to be routed internal and let $E = \{1 \ldots n\} - \mathrm{I}$ be the subset of wires that are to be routed external. (Thus, E is the complement of I.) The characteristic function $\chi_{\mathrm{I}}()$ of set I is defined as follows,

$$\chi_{\mathrm{I}}(j) = \begin{cases} 1 & \text{if } j \in \mathrm{I} \\ 0 & \text{otherwise.} \end{cases}$$

The modified conflict numbers effectively ignore the existence of all nets j for which $\chi_{\mathrm{I}}(j) = 0$. For $j \in \mathrm{I}$, the normalized index function $\pi_{\mathrm{I}}(j)$ is the number of

internal nets to the left of net j inclusive. That is,

$$\pi_{\mathrm{I}}(j) = \sum_{k=1}^{j} \chi_{\mathrm{I}}(k)$$

In modified conflict numbers, the normalized index function will be used in lieu of indices i and j whenever i and j refer to a count of internal wires rather than a specific wire. Thus, given a particular set of internal wires I, the modified conflict number for $i, j \in \mathrm{I}$ is defined by

$$c_{\mathrm{I}}(i,j) = \begin{cases} 0 & \text{if } (i < j \text{ and } U_i - \pi_{\mathrm{I}}(i) + \pi_{\mathrm{I}}(j) \leq L_j) \text{ or } (i > j) \\ \pi_{\mathrm{I}}(j) - \pi_{\mathrm{I}}(i) + 1 & \text{otherwise.} \end{cases}$$

For $j \in \mathrm{I}$, the modified maximum conflict numbers are defined by

$$C_{\mathrm{I}}(j) = \max\{c_{\mathrm{I}}(i,j) \text{ for } 1 \leq i \leq n,\ i \in \mathrm{I}\}.$$

As before, note that $C_{\mathrm{I}}(j) = c_{\mathrm{I}}(i,j)$ for only one index i. We use symbol $\rho_{\mathrm{I}}(j)$ to represent the i for which $C_{\mathrm{I}}(j) = c_{\mathrm{I}}(i,j)$, when $j \in \mathrm{I}$. Referring back to our example, consider the routing shown in Figure 5(b) where $\mathrm{I} = \{1,2,3,5,6,7,9,10\}$. Columns 6 through 9 of Table 1 show the functional values used in the calculation of modified maximum conflict numbers.

In addition to the above definitions, [2] introduced equivalent definitions that use a more suitable notation for our purposes. Let $I_1 < I_2 < ... < I_r$ represent the indices of the nets in I. Thus from [2] we have, $\mathrm{I}_r = i$ if and only if $\pi_{\mathrm{I}}(i) = r$. For the example we have $\mathrm{I}_1 = 1, \mathrm{I}_2 = 2, \mathrm{I}_3 = 3, \mathrm{I}_4 = 5, \ldots, \mathrm{I}_8 = 10$ where $r = 8$ is the number of internal wires. This representation of internal wires gives rise to the following equivalent definitions for conflict numbers.

$$c_{\mathrm{I}}(\mathrm{I}_p, \mathrm{I}_q) = \begin{cases} 0 & \text{if } (\mathrm{I}_p < \mathrm{I}_q \text{ and } U_{\mathrm{I}_p} - p + q \leq L_{\mathrm{I}_q}) \text{ or } (\mathrm{I}_p > \mathrm{I}_q) \\ q - p + 1 & \text{otherwise.} \end{cases}$$

$$C_{\mathrm{I}}(\mathrm{I}_q) = \max\{c_{\mathrm{I}}(\mathrm{I}_p, \mathrm{I}_q) \text{ for } \mathrm{I}_p \leq \mathrm{I}_q\}.$$

The definitions of modified conflict number provide three different notations that refer to the same net or wire: i, $\pi_{\mathrm{I}}(i)$, and I_r where $\pi_{\mathrm{I}}(i) = r$. As in [2] we will always use i or I_r to represent the net i or wire i. The normalized index $\pi_{\mathrm{I}}(i) = r$ will be used only when we need a count of the number of internal wires up to and including wire i. Thus, in the condition for $c_{\mathrm{I}}(\mathrm{I}_p, \mathrm{I}_q) = 0$, the inequality $\mathrm{I}_p < \mathrm{I}_q$ means that net I_p is to the left of net I_q, while the value $q-p+1 = \pi_{\mathrm{I}}(\mathrm{I}_q) - \pi_{\mathrm{I}}(\mathrm{I}_p) + 1$ is the number of internal wires forcing the placement of a horizontal segment in wire I_q.

In addition to the definition of a modified conflict number, a linear time algorithm MARK-INTERNAL was presented in [2] to produce the sets I and E using the modified conflict number. The details of MARK-INTERNAL are given below.

Algorithm MARK-INTERNAL

INPUT: A set of n right oriented nets and a channel width $d \geq 0$.

OUTPUT: Each wire is marked I or NI, so that all I wires can be routed internal using a channel width of d. Moreover, the number of wires marked I is maximized.

METHOD: As the algorithm proceeds, it will always be the case that all wires marked I can be routed internal using a channel width of d. The wires are processed from left to right. Wire j is processed by computing the maximum conflict number $C_{\text{I}}(j)$. This number is used to determine whether wire j will "fit" in the channel. If so, then j is marked I, otherwise it is marked NI. Details are given in Figure 6.

Throughout the remainder of this paper, unless otherwise stated, the use of the terms conflict number and maximum conflict number will mean modified conflict number and modified maximum conflict number respectively.

```
01  π_I(0) ← 0; i ← 1
02  for j ← 1 to n do
03      π_I(j) ← π_I(j − 1) + 1
04      mark(j) ← I
05      while (mark(i) = NI) or (U_i − π_I(i) + π_I(j) ≤ L_j) do
06          i ← i + 1
07      if π_I(j) − π_I(i) ≥ d then
08          π_I(j) ← π_I(j) − 1
09          mark(j) ← NI
```

Figure 6: Details of the algorithm MARK-INTERNAL.

We will also make reference to a very useful property about conflict numbers that was established in [2]. This property states that for fixed I_q the defined, non-zero conflict numbers $c_I(I_p, I_q)$ form a decreasing sequence of consecutive integers ending with 1. In fact, if I_p is the i for which $c(i, I_q)$ is maximum then any I_r such that $I_r < I_p$ or $I_r > I_q$ implies $c(I_r, I_q) = 0$. The following Property is the formal statement from [2].

Property 2 *For* $I_p = \rho_I(I_q)$,

$$\begin{aligned} c_I(I_p, I_q) &= q - p + 1; \\ c_I(I_{p+1}, I_q) &= q - (p+1) + 1; \\ c_I(I_{p+2}, I_q) &= q - (p+2) + 1; \\ &\cdots \end{aligned}$$

$$c_{\mathrm{I}}(I_q, I_q) \quad = \quad q - (p + (q - p)) + 1 = 1.$$

Moreover, for all $I_r \in I - \{I_p, I_{p+1}, \ldots, I_q\}$, $c_{\mathrm{I}}(I_r, I_q) = 0$.

These definitions and properties of conflict numbers will be crucial to the results presented in this paper. We will also make use of the algorithm MARK-INTERNAL.

4 Optimization Criteria

The goal of the river routing problem is to produce a routing in which the corresponding pairs of terminals are wired together in an *optimal* fashion. Optimization criteria typically used in the VLSI river routing problem are area of enclosing rectangle, vertical density (number of tracks used), number of jogs, maximum individual wire length, and total wire length. Total wire length has been used as a criterion for optimization in traditional river routing. The motivation for using total wire length as an optimization criterion is that long wires have greater resistance, therefore short wire lengths are desirable. Recent literature [2, 7] pertaining to the NCC model has concentrated on the area of enclosing rectangle and vertical density as optimization criteria, two closely related criteria. Our goal for this paper is to produce an internal-external routing, for any given instance with fixed channel width, that has minimum total wire length. Thus an *optimal internal-external routing* for a given instance with fixed channel width is one with *minimum total wire length.* In Subsection 4.1 we present the motivation for developing a new routing algorithm, and in Subsection 4.2 we present the objective function by which a new routing algorithm may minimize total wire length.

4.1 Motivation for a New Routing Algorithm

In Section 3, we discussed a popular routing method known as the greedy routing method. Here, we examine whether the greedy routing method always produces an optimal routing. First, we consider the topology of wires in a greedy internal-external routing.

4.1.1 Monotonic wires

Consider the greedy internal-external routing in Figure 7. Recall that given an instance with fixed channel s width the greedy method (using conflict numbers) produces two disjoint sets of wires. Let I^G (E^G) be the set of wires to be routed internal (external) using the greedy method. Thus in Figure 7, I^G and E^G would

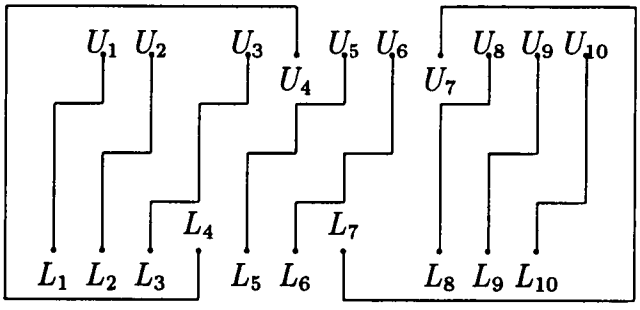


Figure 7: Greedy Internal-External Routing

be as follows:

$$I^G = \{W_1, W_2, W_3, W_5, W_6, W_8, W_9, W_{10}\}$$

$$E^G = \{W_4, W_7\}$$

Observe the topology of every wire in I^G. Every wire in I^G is routed in a greedy fashion, and every wire in I^G is a *monotonic internal wire*. An internal wire is monotonic if and only if it never backtracks horizontally or vertically. Thus the greedy method for routing internal wires produces monotonic internal wires. This follows directly from the description provided in Section 3.1. The length of any monotonic internal wire can be defined as follows:

$$Length(W_i) = (U_i - L_i) + s, \text{ for each } W_i \in I.$$

Recall that the NCC model defines an external wire to be a wire that intersects the base only at its endpoints. Therefore, an external wire must be routed around the outside of the base (either to the left or right of the base). It is obvious that an external wire must backtrack, thus the definition of a *monotonic external wire* cannot be the same definition given for a monotonic internal wire. The definition of a monotonic external wire is divided into three parts, and an external wire is monotonic if and only if all three parts of the definition are true. The definition of a monotonic external wire is as follows:

1. The continuous portion of wire in the upper channel never backtracks hori-

zontally, and it backtracks only once in the vertical direction.

2. The continuous portion of wire in the middle channel never backtracks horizontally or vertically.

3. The continuous portion of wire in the lower channel never backtracks horizontally, and it backtracks only once in the vertical direction.

Figure 8a thru Figure 8c highlight the different portions of a monotonic external wire. Clearly, the greedy method described in Section 3.2 produces only monotonic external wires.

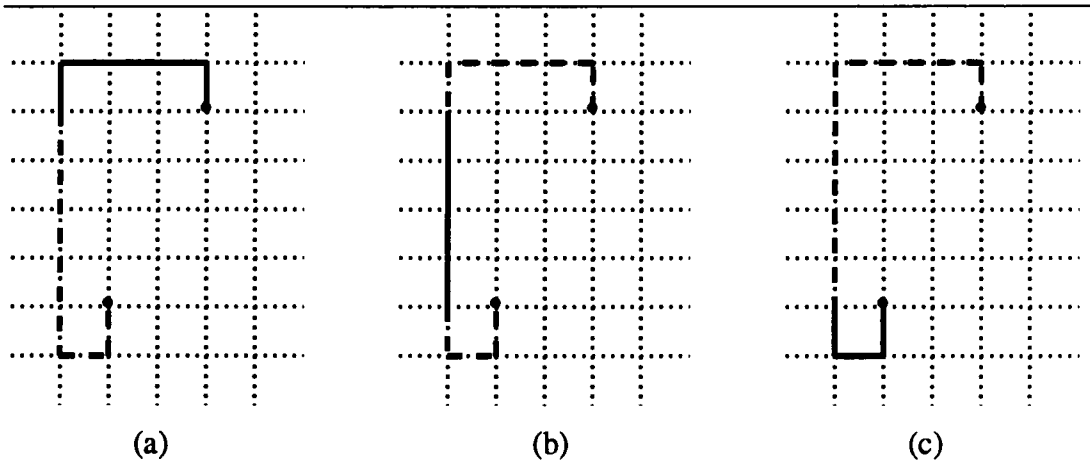


Figure 8: An example to illustrate monotonic external wires (a) the continuous portion of wire in the upper channel (b) the continuous portion of wire in the middle channel (c) the continuous portion of wire in the lower channel (d) a non-monotonic external wire.

Any routing composed solely of monotonic wires (internal and external) is a *monotonic routing*. We will restrict our attention to monotonic routings, but not

necessarily greedy routings.

4.1.2 The greedy method and optimal internal-external routings

The greedy routing method has been the method of choice in much of the literature pertaining to the river routing problem. It was shown in [4] that the greedy method would produce an internal routing with minimum area for both the fixed and variable channel width problems. In addition, the greedy method was shown in [2, 7] to produce an internal-external routing with minimum area and minimum vertical density by routing half the external wires to the left of the base and the other half to the right of the base. This was true for both the fixed and variable channel width problem. An example of a greedy routing for fixed channel width 3 is given in Figure 9.

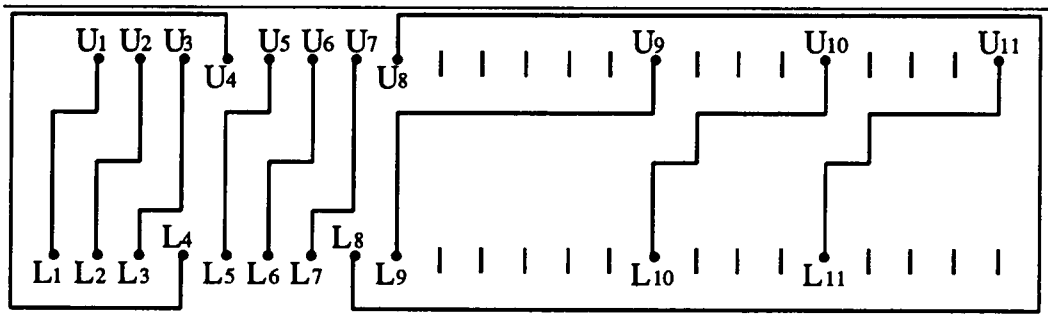


Figure 9: A greedy internal-external routing with minimum area and minimum vertical density.

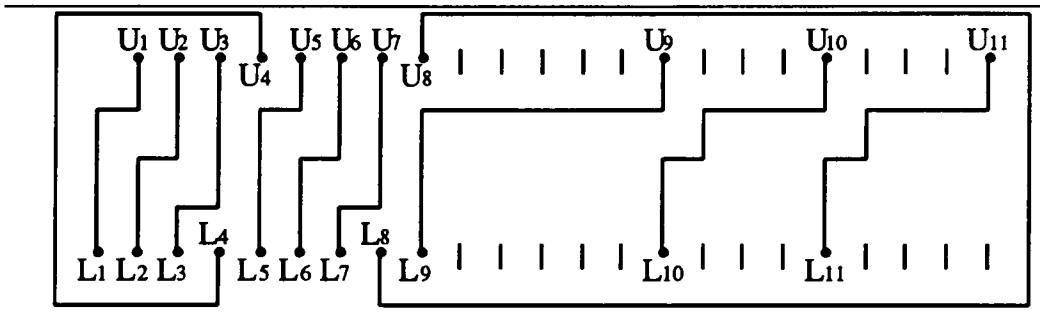
Consider the greedy routing method for minimizing total wire length for a given instance with fixed channel width s. The results of [7] indicate that if an

internal routing is produced then it is of minimum total wire length. This follows directly from the monotonicity condition. This can be formally stated as the following property.

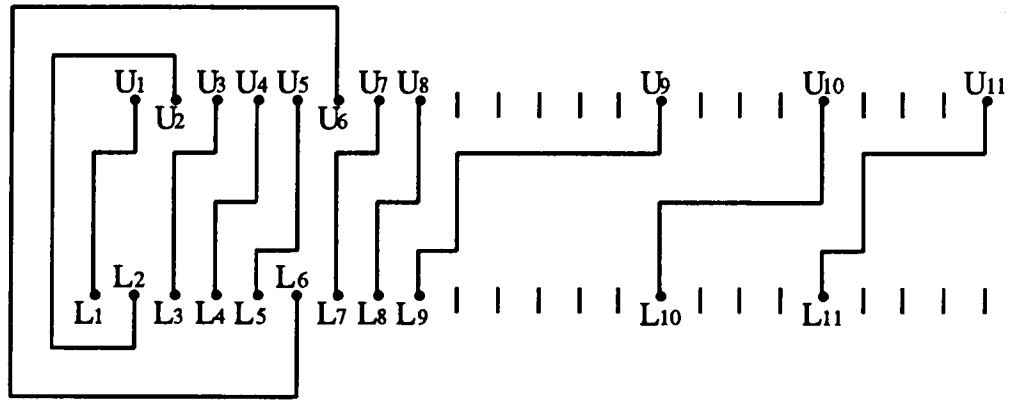
Property 3 *Given a river routing instance with a fixed channel width* s, *the length of any monotonic internal wire W_i is of minimum length. Moreover, if W_i is internal for all i then the routing has minimum total wire length.*

Proof: Since the channel width is fixed at s then that is the minimum vertical distance that W_i must travel to connect U_i and L_i. The horizontal distance between U_i and L_i is the offset, which is defined as $(U_i - L_i)$. Thus $(U_i - L_i) + s$ is as short as W_i can be and still connect the corresponding terminals U_i and L_i. Clearly, if every W_i is of minimum length then the sum is minimized. ∎

The results in [7] do not state whether a greedy internal-external routing for an instance with fixed channel width has minimum total wire length. Observe the internal-external routings (fixed channel width) in Figure 10. Figure 10a illustrates the sets I^G and E^G produced by the greedy method for this instance with a fixed channel width of 4, where the wires in E^G are routed around the ends of the base using the method described in [2, 7]. The total wire length in Figure 10a is 112. The internal-external routing shown in Figure 10b has sets I and E such that $I \neq I^G$ and $E \neq E^G$ for the same instance. The routing in



(a)



(b)

Figure 10: A example to show that an optimal greedy internal-external is not always possible (a) A greedy routing (b) An optimal internal-external routing.

Figure 10b has a total wire length of only 96, which is smaller than that of the routing in Figure 10a. In fact, the reader may verify that internal-external routing in Figure 10b is optimal by trying all other possible routings. Thus, the sets I^G and E^G produced by the greedy routing method do not always admit an optimal routing.

Clearly, the monotonicity condition of external wires is not sufficient to guarantee minimum total wire length. The choice of external wires also determines whether an internal-external routing will be optimal as does the end of the base around which each external wire is routed.

4.2 An Objective Function for Minimizing Total Wire Length

In this subsection we present the objective function for minimizing total wire length. It is clear from Section 4.1.2 that to minimize total wire length we must focus on our choices for external wires.

According to Property 3, if an internal routing exists for a given instance with a fixed channel width s then it is of minimum total wire length. We may then consider every internal-external routing for a given instance with fixed channel width s to consist of a *total base cost* plus a *total additional cost*, where cost is

measured in terms of wire length. We may define the total base cost as follows:

$$\begin{aligned} \textit{total base cost} &= \sum_{i=1}^{n} \textit{base cost}(W_i) \\ &= \sum_{i=1}^{n}((U_i - L_i) + s). \end{aligned}$$

Notice that the total base cost is just the sum of the minimum lengths of all wires.

Consider the total additonal cost for an internal-external routing of a given instance with fixed channel width s. Let E be the set of external wires in an internal-external routing, and let $E_1 < E_2 < ... < E_m$ be the indices of the external wires in E. The total additional cost incurred by a particular set E can be defined as follows:

$$\textit{total additional cost} = \sum_{i=1}^{m} \textit{additional cost}(E_i).$$

To define the additional cost of an individual external wire E_i we need the following lemma.

Lemma 1 *If E_i is routed left around the base then $E_1, E_2, ..., E_{i-1}$ are routed left around the base, and if E_i is routed right around the base then $E_{i+1}, E_{i+2}, ..., E_m$ are routed right around the base.*

Proof: The proof of this lemma comes directly from the definition of the NCC

model. The routing of all wires in any given instance takes place in one layer, and no overlap between wires are allowed. Thus, it is not possible for E_i to be routed left around the base and E_j such that $j < i$ be routed right around the base, because then either there would be an overlap between the two wires or the wire would must be routed internal to the base, which is not allowed. Thus, the routing produced would not be valid. ∎

The monotonicity requirement and Lemma 1 provides for the following definitions of the additional cost of a single external wire E_i. Recall that E_i may be routed left or right around the base. Therefore, we give a definition for the additional cost of routing E_i left around the base, and a definition for routing E_i right around the base. The additional cost of E_i is defined to be the minimum of the left and right definition, or the left routing if their additional costs are equivalent.

$$\begin{aligned} \mathit{left}(E_i) &= 2((L_{E_i} - L_1) + i) + 4i; \\ \mathit{right}(E_i) &= 2((U_n - U_{E_i}) + (m - i + 1)) + 4(m - i + 1); \\ \mathit{additional\ cost}(E_i) &= \mathit{Min}(\mathit{left}(E_i), \mathit{right}(E_i)). \end{aligned}$$

The reader may verify this definition by considering the external wires in Figure 11. The highlighted portion of each external wire is its additional cost.

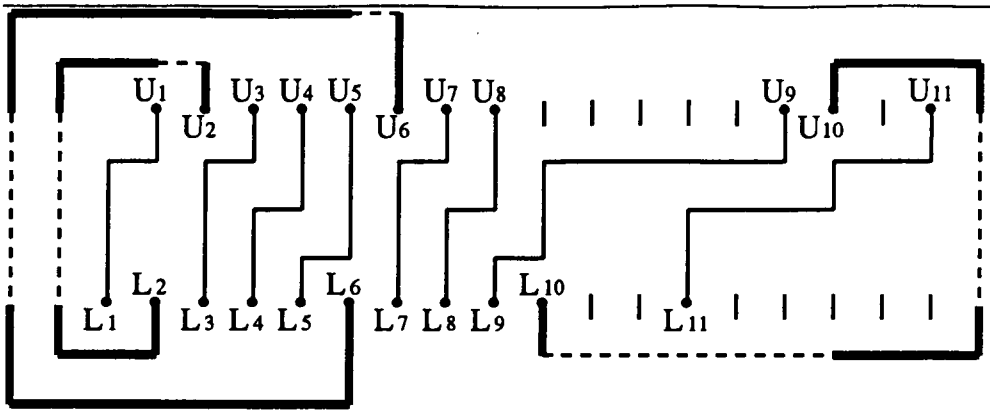


Figure 11: An example to show the additional cost of an external wire.

The remaining portion of wire is the base cost of the wire.

From Property 3, the total base cost of an internal-external routing for a given instance with fixed channel width s cannot be reduced. Thus for producing an optimal internal-external routing our objective function is to choose I and E so as to minimize *total additional cost.*

5 An Algorithm For Optimal Internal-External Routings

In this section, we present an algorithm that produces sets I^M and E^M, where I^M (E^M) is the set of internal (external) wires, such that an optimal internal-external routing for a given instance with fixed channel width s may be produced.

It was shown in Section 4.1 that the greedy routing method did not necessarily produce an optimal internal-external routing for a given instance with fixed channel width. However, there are two legitimate reasons for not abandoning the greedy method altogether. The first reason is the concept of conflict numbers. Recall that MARK-INTERNAL produces the sets I^G and E^G, where I^G (E^G) is the set of internal (external) wires of a greedy routing, in linear time. Secondly, the greedy routing method produces a minimal set of external wires for a given instance with fixed channel width. It seems reasonable (though not necessarily true) that a minimal set of external wires would imply a minimum *total additional cost.*

Our algorithm will make use of the algorithm MARK-INTERNAL, and its development will be presented in two phases. In Subsection 5.1 we present an algorithm that minimizes the total additional cost for a fixed number m of external wires, where $m = |E^G|$. That is, the algorithm will find a set of external wires

E^M such that $|E^M| = |E^G|$ and the total additional cost is minimized. In Subsection 5.2, we remove the restriction that $|E^M| = |E^G|$. We show, with this slight modification, that our algorithm will produce I^M and E^M such that $I^M = I^O$ and $E^M = E^O$, where I^O (E^O) is the set of internal (external) wires that can produce an optimal internal-external routing for a given instance with fixed channel width s.

5.1 An Algorithm for a Fixed Number of External Wires

In this subsection we present a description of our algorithm, we examine its time complexity, and then we provide a proof of correctness.

5.1.1 Description of MOVE_EXTERNAL

Given a right block instance of n nets with a fixed channel width s, our algorithm first uses MARK-INTERNAL to obtain initial sets of internal and external wires (I^G, E^G). Recall that $I_1^G < I_2^G < ... < I_r^G$ are the indices of the nets in I^G, and let $E_1^G < E_2^G < ... < E_m^G$ be the indices of the nets in E^G. Our algorithm produces new sets I^M and E^M from I^G and E^G. This production is done by the algorithm MOVE_EXTERNAL.

MOVE_EXTERNAL produces I^M and E^M through two intermediary steps. At the lower level it tries to find the optimal net location for wire E_i^M for some i.

Here, we state the following property without proof to illustrate the possible net locations for wire E_i^M and still maintain m external wires. (Lemma 2 provides the proof).

For all $1 \leq i \leq m$, $E_i^G - (s-1) \leq E_i^O \leq E_i^G$.

This property states that E_i^O must be within $s-1$ nets left of E_i^G for all i, therefore MOVE_EXTERNAL tries E_i^M at every net within that range for each particular i.

Consider the example in Figure 12. Here, it suffices to show only the upper terminal row of an instance. Initially, MOVE_EXTERNAL may start with $E_i^M = E_i^G$. In trying to find the optimal net location for E_i^M, MOVE_EXTERNAL considers a net, initially X_1, left of E_i^G. If $X_1 \in I^M$, then net X_1 is "switched" with E_i^G for the location of E_i^M. Temporary sets, E^{temp} and I^{temp}, are built by MOVE_EXTERNAL in the following manner. MOVE_EXTERNAL initializes E^{temp} to contain E_1^M thru E_{i-1}^M plus net X_1, and it initializes I^{temp} to contain every net currently in I^M prior to net X_1. MOVE_EXTERNAL then builds E^{temp} and I^{temp} by marking all nets right of X_1 using the standard greedy method (MARK-INTERNAL). Once completed, the *total additional cost* of E^{temp} is calculated as described in Section 4.2. If $|E^{temp}| = |E^M|$ and its *total additional cost* is less than or equal to that of E^M, then E^M and I^M are replaced with E^{temp} and I^{temp}, respectively. For the instance in Figure 12, MOVE_EXTERNAL continues to

switch E_i^M with each net X_1 thru X_5, skipping over E_{i-1}^M. Once E_{i-1}^M has been skipped, the greedy method can only be used to mark nets right of E_{i-1}^M. That is, the location of E_{i-1}^M is fixed.

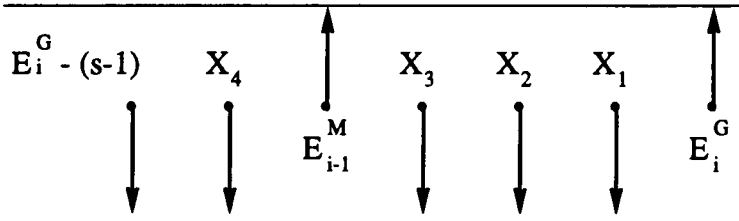


Figure 12: An example to illustrate the movement of E_i^M.

The fact that external wires are not counted in the $s-1$ nets left of E_i^G presents no problem. That is, $E_i^G - (s-1) \leq E_i^M \leq E_i^G$ will still be true. This must be since we have defined $E_1^M < E_2^M < ... < E_m^M$.

The lower level steps are performed in order for each external wire in E^M. After the final sets I^M and E^M are produced, MOVE_EXTERNAL finds the j such that E_j^M is the first external wire to be routed right around the base. This j is found by using the formula, described in Section 4.2, for the *additional cost* of an external wire and marking the first wire for which the minimum is the right-routing. Lastly, MOVE_EXTERNAL calls a greedy router, which was described in Section 3, that produces the actual routing.

Algorithm MOVE_EXTERNAL

INPUT: A set of n right oriented nets and a channel width $s \geq 0$.

OUTPUT: The sets I^M and E^M, where I^M (E^M) is the set of internal (external) wires such that $|E^M| = |E^G|$ and the total additional cost is minimized.

METHOD: Run MARK-INTERNAL to obtain the sets I^G and E^G. Save the information regarding the conflict numbers, maximum conflict numbers, and the normalized index function. Assign I^G and E^G as the initial values of I^M and E^M, respectively. The following steps are repeated for each $E_i^M \in E^M$, starting with $i = 1$ and proceeding until $i = m$. Consider the net X immediately left of E_i^M. If $X \in E^M$ continue moving left until a net X is found such that $X \in I^M$. Create the sets I^{temp} and E^{temp}. Initialize I^{temp} to contain every net in I^M up until net X. Initialize E^{temp} to contain the external nets E_1^M thru E_{i-1}^M plus net X. Run MARK-INTERNAL2 with an input of E_i^M to complete sets I^{temp} and E^{temp}. MARK-INTERNAL2 is a slightly modified version of MARK-INTERNAL. It takes a net as input and marks only nets strictly to the right of that net. If any external nets were skipped, then mark net E_{i-1}^M so that it remains the input to MARK-INTERNAL2 for all further iterations on this particular i. Compute the total additional cost of both E^M and E^{temp}. If the *total additional cost* of E^{temp} is less than or equal to that of E^M, then replace I^M

and E^M with I^{temp} and E^{temp}, respectively. If any external nets were skipped then initialize I^{temp}, for the next iteration on this i, to contain the current net E_i^M. Once the final sets of I^M and E^M have been produced, calculate the j such that E_j^M is the first net to be routed right. The net j is found by calculating the additional cost of each net as described in Section 4.2, and marking the first net for which the right routing incurs the smallest additional cost. Once j has been found, call the greedy router described in Section 3. Figure 13 gives the details of the algorithm.

For example, consider Figure 14a. We initially have

$$I^M = I^G = \{W_1, W_2, W_3, W_6, W_7, W_8, \}$$

$$E^M = E^G = \{W_4, W_5, W_9, W_{10}\}.$$

In the first iteration of MOVE_EXTERNAL, we have $E_1^M = 4$. MOVE_EXTERNAL begins by assigning the following:

$$I^{temp} \leftarrow \{W_1, W_2\}$$

$$E^{temp} \leftarrow \{W_3\}.$$

```
01 run MARK-INTERNAL
02 I^M ← I^G, E^M ← E^G
03 for i ← 1 to m do
04    GreedyNet ← E_i^M
05    SwitchNet ← E_i^M − 1
06    SkipExternal ← FALSE
07    I^temp ← φ
08    if (E_i^M > E_i^G − (s − 1)) then
09       for j ← 1 to (s − 1) do
10          k ← 1
11          while (SwitchNet ∈ E^M) do
12             SwitchNet ← SwitchNet − k
13             k ← k + 1
14          if (k > 1) then
15             SkipExternal ← TRUE
16          I^temp ← I^temp ⋃ (I^M − {SwitchNet})
17          for net ← (GreedyNet + 1) to n do
18             if net ∈ I^M then
19                I^temp ← I^temp − {net}
20          E^temp ← (E^M − {GreedyNet, ..., E_m^M}) ⋃ {SwitchNet}
21          run MARK-INTERNAL2(GreedyNet)
22          TAC(E^M) ← total additional cost(E^M)
23          TAC(E^temp) ← total additional cost(E^temp)
24          if (TAC(E^temp) ≤ TAC(E^M)) and (|E^temp| = |E^M|) then
25             I^M ← I^temp, E^M ← E^temp
26          if SkipExternal then
27             I^temp ← {SwitchNet}
28          else
29             GreedyNet ← E_i^M − 1
30             I^temp ← φ
31          SwitchNet ← SwitchNet − 1
32          i ← 1
33          while (i ≤ m) and (right = 0) do
34             if right(E_i^M) < left(E_i^M) then
35                right ← i
36             else
37                i ← i + 1
38 run GREEDY-ROUTER(I^M,E^M)
```

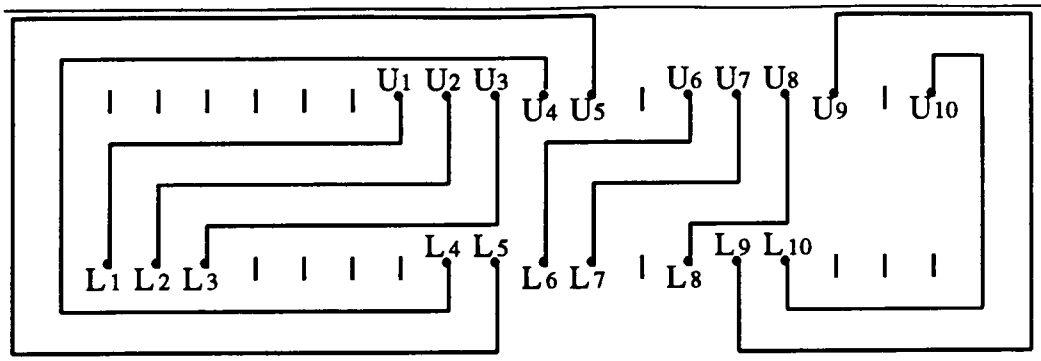
Figure 13: Details of the algorithm MOVE_EXTERNAL.

MOVE_EXTERNAL completes I^{temp} and E^{temp} on line [21] in Figure 13. The result of this first iteration is shown in Figure 14b, which indicates that *total additional cost* of E^{temp}, as calculated in line [23] of Figure 13, was less than for E^M. Thus, I^M and E^M were replaced with I^{temp} and E^{temp}, respectively. Next, E_1^M is tried at nets 1 and 2, but at these net locations the *total additional cost* of E^{temp} is greater than that of E^M. Therefore, net 3 is chosen for E_1^M. Now, MOVE_EXTERNAL finds the best location for E_2^M. After finding the best location for E_1^M, E_2^M is still located at net 5. First, MOVE_EXTERNAL tries E_2^M at net 4, and then it skips net 3, maintaining this location for wire E_1^M, and tries E_2^M at both nets 1 and 2. The final choice for E_2^M is shown in Figure 14c. This process is repeated for all the external wires, but the routing in Figure 14c is also the final routing.

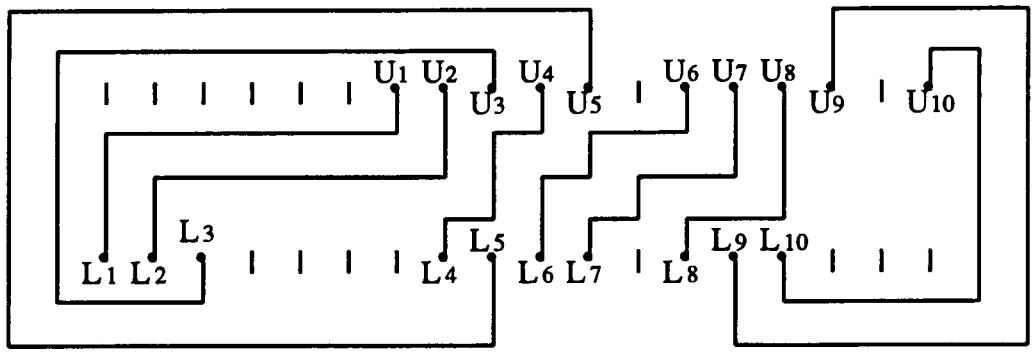
5.1.2 Time complexity of MOVE_EXTERNAL

In this section we calculate the time complexity of MOVE_EXTERNAL.

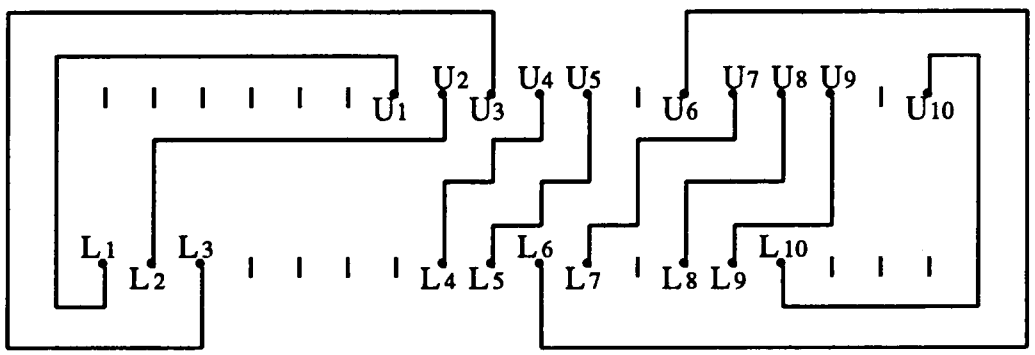
The first line in MOVE_EXTERNAL is a call to MARK-INTERNAL, and in [2] it was shown that MARK-INTERNAL is $O(n)$. Line [2] takes $O(n)$ time since lists of total size n are being copied. The outermost loop (line [3]) executes m times, where m is the number of external wires produced by MARK-INTERNAL. Lines [4] thru [8] take constant time. The first inner loop (line [9]) is executed $s-1$



(a)



(b)



(c)

Figure 14: An example to illustrate MOVE_EXTERNAL (a) a greedy routing (b) after one low-level step in MOVE_EXTERNAL (c) after MOVE_EXTERNAL completes.

times, where s is the fixed channel width. The while loop (line [11]) is executed at most m times. Each statement within the while loop takes constant time. The lines [14] and [15] take constant time. Like line [2], line [16] is a copy that takes $O(n)$ time. The for loop of line [17] is executed at most n times, and each statement within the for loop takes constant time. Line [20] takes $O(n)$ time. Line [21] is a call to MARK-INTERNAL2. MARK-INTERNAL2 is a slightly modified version of MARK-INTERNAL, and it is also of $O(n)$. Both lines [22] and [23] take $O(m)$ time, the description of this subroutine is given in the equation for *total additional cost.* The lines [24] thru [32] take constant time. The while loop (line [33]) executes at most m times. The statements within this while loop take constant time. The last line is a call to a greedy router, which was shown to be $O(n^2)$ in [2, 7]. MOVE_EXTERNAL at the inner most level is of $O(n)$, and at the second level is of $O(s)$, and at the outermost level is of $O(m)$. Thus, we may conclude that the time complexity of MOVE_EXTERNAL is $O(m \cdot s \cdot n)$. Note that $m \leq n$, and $s \leq n$. Therefore, the time complexity is no worse than $O(n^3)$.

5.1.3 Proof of correctness

In this subsection we prove that MOVE_EXTERNAL minimizes total additional cost for routing with a fixed number of external wires m, where m is the number of external wires produced by MARK-INTERNAL.

Theorem 1 *Let m be the number of external wires in a greedy routing for fixed channel width s. The algorithm MOVE_EXTERNAL produces an optimal internal-external routing with m external wires for a channel width of s.*

Let s be a fixed channel width and m be the number of external wires produced by the greedy internal-external routing. We will show that of all internal-external routings with channel width s and m external wires, the routing produced by MOVE_EXTERNAL is optimal. First it is necessary to establish the possible net choices for external wires in an optimal internal-external routing with m external wires.

Consider a right block instance of n nets with a fixed channel width of s and fixed number of external wires of m, where m is the number of external wires produced by the greedy routing. Let $E^G = \{E_1^G, E_2^G, ..., E_m^G\}$ be the set of indices of external wires in the greedy routing such that $E_1^G < E_2^G < ... < E_m^G$. In addition, let $E^O = \{E_1^O, E_2^O, ..., E_m^O\}$ be the set of indices of external wires in an optimal routing such that $E_1^O < E_2^O < ... < E_m^O$. I^G and I^O are defined similarly, except they contain the indices of the internal wires for the greedy and optimal routings, respectively. Lemma 2 defines the possible net choices for the external wires in the optimal routing in relation to their positions in the greedy routing.

Lemma 2 *For all $1 \leq i \leq m$, $E_i^G - (s-1) \leq E_i^O \leq E_i^G$ such that $|E^O| = |E^G|$.*

Proof(Part I: $E_i^G - (s-1) \leq E_i^O, \forall\, i$): By way of contradiction, we will prove $E_i^G - (s-1) \leq E_i^0 \;\; \forall\, i$. Assume $\exists$ an optimal routing where $E_k^G - (s-1) \leq E_k^O \;\; \forall\, k$ such that $k < i$, but $E_i^O < E_i^G - (s-1)$. We will show in this case $\exists$ an $E_{j+1}^O \leq E_j^G \;\; \forall \;\; E_j^G, j \geq i$. This would mean there are $m+1$ external wires in the optimal routing, which is a contradiction since we are maintaining a fixed number of external wires, m. The proof will be an induction on j.

Basis($j = i$, show $E_{i+1}^O \leq E_i^G$): Recall that we have assumed $E_i^O < E_i^G - (s-1)$. If $E_{i+1}^O > E_i^G$ then the nets $E_i^O + 1$ thru E_i^G inclusive must be routed internal in the optimal routing of the same instance.

Consider the greedy routing. Recall that I^G is the set of internal wires in the greedy routing. The net E_i^G was marked external in the greedy routing because $\exists\, I_p^G \in I^G$ such that $c^G(I_p^G, E_i^G) \geq s$. According to Property 1 this means there would be a horizontal segment of E_i^G on or below the lower terminal row. Consequently $\pi_I^G(E_i^G) - \pi_I^G(I_p^G) + 1 \geq s$, therefore $I_p^G \leq E_i^G - (s-1)$. According to Property 2, since $c_I^G(I_p^G, E_i^G) \neq 0$ and $I_p^G \leq E_i^G - (s-1)$ it must be the following:

$$c_I^G(E_i^G - (s-1), E_i^G) \neq 0. \tag{1}$$

Equation 1 implies the following:

$$c_I^G(E_i^G - (s-1), E_i^G) = \pi_I^G(E_i^G) - \pi_I^G(I_p^G) + 1 \quad (2)$$

$$= s \quad (3)$$

Since we have assumed $E_i^O < E_i^G - (s-1)$ then, as we stated above, $E_i^O + 1$ thru E_i^G inclusive must be routed internal in the optimal routing. This being the case we have encountered the same number of internal wires by and inclusive of net E_i^G in the optimal routing. That is,

$$\pi_I^O(E_i^G) = \pi_I^G(E_i^G). \quad (4)$$

If $E_i^O < E_i^G - (s-1)$ and $E_{i+1}^O > E_i^G$ then by net $E_i^G - (s-1)$ in the optimal routing we have encountered exactly one less internal wire by the same net in the greedy routing. Thus, we have:

$$\pi_I^O(E_i^G - (s-1)) = \pi_I^G(E_i^G - (s-1)) - 1. \quad (5)$$

Now consider the conflict number $c_I^O(E_i^G - (s-1), E_i^G)$. By equations 4 and 5 we have the following

$$\begin{aligned} c_I^O(E_i^G - (s-1), E_i^G) &= \pi_I^G(E_i^G) - (\pi_I^G(E_i^G - (s-1)) - 1) + 1 \\ &= \pi_I^G(E_i^G) - 1 - \pi_I^G(E_i^G - (s-1)) + 2 \\ &= c_I^G(E_i^G - (s-1), E_i^G) + 1 \\ &> c_I^G(E_i^G - (s-1), E_i^G). \end{aligned}$$

Thus, from equation 2 we have

$$c_I^O(E_i^G - (s-1), E_i^G) \geq s.$$

There is no routing on the terminal rows, thus E_i^G will not fit internal in the optimal routing. Therefore, $E_{i+1}^O \leq E_i^G$.

Induction Step: We have shown Part I of Lemma 2 holds for $j = i$. Assume it holds for $i, ..., j-1$. We will show that $\exists$ an $E_{j+1}^O \leq E_j^G$. Assume $E_{j+1}^O > E_j^G$. Consider the greedy routing, where E_j^G is routed external because $\exists$ $I_p^G \in I^G$ such

that $c^G(I_p^G, E_j^G) \geq s$. By the induction hypothesis $E_{i+1}^O \leq E_i^G, ..., E_j^O \leq E_{j-1}^G$. Consequently, by every net after net E_i^O up until net E_{j-1}^G in the optimal routing we have encountered one additional external wire, and one less internal wire. Thus, $\pi_I^O(E_j^G) = \pi_I^G(E_j^G) - 1$ if $E_{j+1}^O > E_j^G$ in the optimal routing. Consider $c_I^O(I_r^O, E_j^G)$ such that $I_r^O = I_p^G$ if $I_p^G \in I^O$, otherwise it is the nearest internal to the right of I_p^G in the greedy routing. Clearly, $\pi_I^O(I_r^O) = \pi_I^O(I_p^O)$. To calculate $c_I^O(I_r^O, E_j^G)$ we have two cases to consider. The first case to consider is when $I_r^O < E_j^O$. In this case, we can use an argument analogous to that given in the basis to show that $c_I^O(E_i^G - (s-1), E_j^G) \geq s$. Thus, in this case E_j^G must be routed external in the optimal routing. The second case is when $I_r^O > E_j^O$. In this case we have $\pi_I^O(I_p^G) \leq \pi_I^G(I_p^G) - 1$. Therefore, we can say the following:

$$\begin{aligned} c_I^O(I_r^O, E_j^G) &= \pi_I^O(E_j^G) - \pi_I^O(I_r^O) + 1 \\ &\geq \pi_I^G(E_j^G) - 1 - (\pi_I^G(I_p^G) - 1) + 1 \\ &\geq \pi_I^G(E_j^G) - \pi_I^G(I_p^G) + 1 \\ &\geq c_I^G(I_p^G, E_j^G). \end{aligned}$$

Since $c_I^G(I_p^G, E_j^G) \geq s$, then $c_I^O(I_r^O, E_j^G) \geq s$. Thus, we have shown in both cases that E_j^G must be external in the optimal routing, thus $E_{j+1}^O \leq E_j^G$. ∎

Proof(Part II: $E_i^O \leq E_i^G$, $\forall$ i): By way of contradiction, we will prove $E_i^O \leq E_i^G$ $\forall$ i. This proof will be an induction on i.

Basis($i = 1$, show $E_1^O \leq E_1^G$): We will show this by way of contradiction. Assume $E_1^O > E_1^G$. Since we have defined $E_1^O < E_2^O < ... < E_m^O$, we know E_1^O is the first external wire in the optimal routing. Thus, all W_j such that $1 \leq j \leq E_1^O$ must routed internal in the optimal routing. However, we know from the results in [7] that if wires 1 through $E_1^G - 1$ are internal then E_1^G cannot be internal. Thus, E_1^O could not be the first external wire in the optimal routing. Therefore, $E_1^O \leq E_1^G$.

Induction Step: We have shown Part II of Lemma 2 to hold when $i = 1$. Assume it holds for $1, ..., i-1$, we will show that it holds for i. By way of contradiction we will show $E_i^O \leq E_i^G$. Assume $E_i^O > E_i^G$. Consider the greedy routing, where E_i^G was routed external because $\exists$ $I_p^G \in I^G$ such that $c_I^G(I_p^G, E_i^G) \geq s$. E_i^G must be routed external, because according to Property 1 its interconnecting wire would have a horizontal segment on or below the lower terminal row. By the induction hypothesis and Part I of Lemma 2 we know that $E_1^G - (s-1) \leq E_1^O \leq E_1^G, ..., E_{i-1}^G - (s-1) \leq E_{i-1}^O \leq E_{i-1}^G$. Therefore, if $E_i^O > E_i^G$ then by net E_i^G in the optimal routing we have encountered the same number of external wires as in the greedy routing by net E_i^G. In addition, by the net I_p^G in the optimal routing we have encountered the same number of external wires or perhaps more external

wires than in the greedy routing by net I_p^G. Let $I_r^O = I_p^G$ if I_p^G is routed internal in the optimal routing, otherwise let I_r^O be the closest internal net to the right of I_p^G in the optimal routing. Then using the same argument as used in the proof of Part I of Lemma 2, $\pi_I^O(I_r^O) \leq \pi_I^G(I_r^G)$. Thus, we have:

$$\begin{aligned} c_I^O(I_r^O, E_i^G) &= \pi_I^O(E_i^G) - \pi_I^O(I_r^O) + 1; \\ &\geq \pi_I^G(E_i^G) - \pi_I^G(I_p^G) + 1; \\ &\geq c_I^G(I_p^G, E_i^G). \end{aligned}$$

In the greedy routing $c_I^G(I_p^G, E_i^G) \geq s$, therefore $c_I^O(I_r^O, E_i^G) \geq s$. Thus, E_i^G must be routed external in the optimal routing, therefore $E_i^O \leq E_i^G$. ∎

Lemma 2 established the only possible net locations for the external wires in an optimal routing. Given the results of Lemma 2, we may now prove the correctness of MOVE_EXTERNAL. First, we restate Theorem 1.

> Let m be the number of external wires in a greedy routing for fixed channel width s. The algorithm MOVE_EXTERNAL produces an optimal internal-external routing with m external wires for a channel width of s.

Proof(Theorem 1): By way of contradiction we will prove that the additional cost of MOVE_EXTERNAL is no greater than that of the optimal routing. Assume that the positioning of E_j^O constitutes a savings in additional cost over that of the position of E_j^M.

Consider the same right block instance as described for Lemma 2. In addition to sets I^G, E^G, I^O, and E^O defined for the greedy and optimal routings, let E^M be the set of indices of external wires produced by MOVE_EXTERNAL such that $E_1^M < E_2^M < ... < E_m^M$, and let I^M be the set of indices of internal wires produced by MOVE_EXTERNAL such that $I_1^M < I_2^M < ... < I_{n-m}^M$. The particular optimal routing we are referencing is identical to the routing produced by MOVE_EXTERNAL up until some net i. Our goal is to prove that the additonal cost incurred by MOVE_EXTERNAL is no greater than that incurred by the optimal routing. In order to prove that, we must examine the eight possible cases for that scenario. The first four cases assume that $E_j^M = i$, and these four cases then take into account the different possible routings in the above assumption. The last four cases assume that $E_j^O = i$, and these four cases then take into account the different possible routings in that assumption.

Case 1($E_j^M = i$ and both E_j^M and E_j^O are routed left): In this case net $i \in E^M$ in the routing produced by MOVE_EXTERNAL, and the same net $i \in I^O$ in the optimal routing, thus $E_j^M < E_j^O$ since the routings match until that point. We also know that $E_j^G - (s-1) \leq E_j^M \leq E_j^G$ since MOVE_EXTERNAL starts with net E_j^G and only attempts to place the external wire within that range. We also know $E_j^G - (s-1) \leq E_j^O \leq E_j^G$ according to Lemma 2. Since $E_j^M < E_j^O$ and both wires are routed left, we know that the *additional cost* of wire E_j^M is less than that of wire E_j^O. However, the position chosen for E_j^O constituted a savings in the *total additional cost.* There are only two ways this savings could have been achieved.

By leaving E_j^O right of E_j^M the right-routed external wires E_k^O thru E_m^O, where E_k^O is the first external wire to be routed right in the optimal routing, did not accumulate as large an *additional cost* as the right routed wires E_l^M thru E_m^M, where E_l^M is the first external to be routed right in the MOVE_EXTERNAL routing. Consider how MOVE_EXTERNAL works, it starts with the greedy routing. It then creates a temporary routing that matches the greedy routing up until net $E_j^G - 1$, it marks $E_j^G - 1$ external and recalculates the conflict numbers for the remaining nets to complete the sets I^{temp} and E^{temp}. It then calculates the total additional cost of the greedy routing and that of the temporary routing. If and only if the total additional cost of the temporary routing is less than that of

the greedy routing then $I^M = I^{temp}$ and $E^M = E^{temp}$, otherwise $I^M = I^G$ and $E^M = E^G$. This process is repeated until each of the $s-1$ nets left of the net E_j^G have been marked external and a new routing produced. Therefore, E_j^M was placed at the net that achieved minimum total additional cost. By performing a greedy routing on the nets following the net E_j^M we know the rightmost net positions for E_l^M thru E_m^M. MOVE_EXTERNAL chose the net position for E_j^M that minimized the total additional cost, therefore the wires E_l^M thru E_m^M could not have gained enough in their additional costs to increase the total additional cost. If the total additional cost had increased E_j^M would not have been placed where it was. In fact, if E_j^O had achieved a savings in total additional cost then E_j^M would have been positioned at the same net as E_j^O. Since $E_j^M \neq E_j^O$ then no savings was achieved.

There is one other way that savings in total additional cost could have been achieved by leaving E_j^O right of E_j^M, and that is if the left-routed external wires E_{j+1}^O thru E_{k-1}^O were allowed to move further left than E_{j+1}^M thru E_{l-1}^M. First, consider when the nets E_j^O and E_j^M are not within $s-1$ nets of the net E_{j+1}^G in their respective routings. Then we know that the furthest left either of E_{j+1}^O A and E_{j+1}^M could move is to the net $E_{j+1}^G - (s-1)$. Consider why E_{j+1}^M could not be positioned at $E_{j+1}^G - (s-1)$. When E_j^M is positioned it can only cause E_{j+1}^M to move left, and if net E_{j+1}^M is not moved as far left as $E_{j+1}^G - (s-1)$ that was

determined by the greedy routing that followed its placement. That applies for all the external nets that follow E_j^M. Thus, the net choice for position E_j^M does not prevent any leftward movement of the external wires that follow it, therefore the net choice for E_j^O did not create any savings in additional cost for left routed external nets. Now consider the case when E_j^O is within the $s-1$ nets left of net E_{j+1}^G in the optimal routing. By definition $E_j^O < E_{j+1}^O$ then the furthest left E_{j+1}^O could be is net $E_j^O + 1$. Similarily, the furthest left E_{j+1}^M could be is $E_j^M + 1$. Since $E_j^M < E_j^O$ implies that $(E_j^M + 1) < (E_j^O + 1)$. Thus, E_{j+1}^M could move further left than E_{j+1}^O, which would only be prevented by the greedy routing that followed. In this case the net choice for E_j^O prevented leftward movement of E_{j+1}^O. Thus, no savings in additional cost could have been achieved.

It has been shown that savings in additional cost was not achieved by the net choice for E_j^O. This is contradiction.

Case 2($E_j^M = i$, and E_j^M is routed left while E_j^O is routed right): Assume that the total additional cost of the optimal routing is strictly less than that of the routing produced by MOVE_EXTERNAL. According to Lemma 1, if E_j^O is routed right then E_{j+1}^O thru E_m^O must also be routed right. Since the optimal and the MOVE_EXTERNAL routings match up until net E_j^M then the savings in additional cost achieved by the optimal routing had to come from its choices

for E_j^O thru E_m^O. Since the optimal routing chose to route them right then the rightmost positions for these external wires would achieve the minimum additional cost for each of these wires. After the placement of E_{j-1}^O a greedy routing of the following nets would have place E_j^O thru E_m^O in their rightmost positions. However, these would also be the initial choices for MOVE_EXTERNAL after it had placed E_{j-1}^M since $E_{j-1}^M = E_{j-1}^O$. Therefore, MOVE_EXTERNAL would have only replaced this routing with one achieved by its placement of E_j^M if and only if this routing had a lower total additional cost. This is a contradiction. Therefore, MOVE_EXTERNAL would not have chosen net i for E_j^M, or the total additional cost of the optimal routing is not less than that of the MOVE_EXTERNAL routing.

Case 3($E_j^M = i$, and E_j^M is routed right and E_j^O is routed left): Assume that the total additional cost of the optimal routing is strictly less than that of the MOVE_EXTERNAL routing. Since the routings match up until net i, the savings achieved by the optimal routing had to come from the external wires E_j^O thru E_m^O. Consider how MOVE_EXTERNAL works. When MOVE_EXTERNAL places E_j^M at net i, it routes E_j^M in the direction that minimizes the additional cost of E_j^M. If the best routing for E_j^M is right around the base at net i, then we know that net E_j^O would have achieved even more savings if it had been chosen

for E_j^M. We know the location for E_j^M was not forced left of E_j^O since the two routings match up until net i. Therefore, MOVE_EXTERNAL would not have chosen net i for E_j^M.

Case 4($E_j^M = i$, and E_j^M and E_j^O are both routed right): Since E_j^M is routed right, we know that E_{j+1}^M thru E_m^M are also routed right. The same can be said for E_j^O thru E_m^O. This is true because this is a single-layer routing, and no wires are allowed to cross each other. By chosing to move E_j^M to the left of the E_j^O we know that it increased its additional cost since it is routed right. In addition, this could have only caused the additional costs of wires E_{j+1}^M thru E_m^M to be greater than or equal to their corresponding wires E_{j+1}^O thru E_m^O additional costs. Since all the nets prior to E_j^M are routed exactly the same as in the optimal routing we know the net choice for E_j^M was not forced to its position by an earlier routing, therefore MOVE_EXTERNAL would not have chosen this net for E_j^M.

Case 5($E_j^O = i$, and E_j^O and E_j^M are both routed left): Assume that the total additional cost of the optimal routing is strictly less that that of the MOVE_EXTERNAL routing. This implies that by moving E_j^O left of the net E_j^M the optimal routing achieved some savings in total additional cost. Consider then how the savings in total additional cost could have been achieved by the optimal

routing. The savings could not have been achieved by any right-routed external wires E_k^O thru E_m^O, where E_k^O is the first external wire to be routed right, since they may be no further right than their corresponding external wires E_k^M thru E_m^M. In addition, the two routings match up until net E_j^O, therefore no savings could have been achieved by external wires E_1^O thru E_{j-1}^O. Thus, the only external wires that can account for the savings of the optimal routing are the left-routed external wires E_{j+1}^O thru E_{k-1}^O.

Given that the only wires that can account for the savings of the optimal routing are the left-routed external wires E_{j+1}^O thru E_{k-1}^O, consider how that savings was achieved. By placing the j^{th} external wire at net E_j^O, the optimal routing allowed nets E_{j+1}^O thru E_{k-1}^O to move further left than their corresponding wires E_{j+1}^M thru E_{k-1}^M. However, from Case 1 we know that unless E_j^M is within the $s-1$ nets left of E_{j+1}^G its placement cannot affect the placement of the $(j+1)^{th}$ external wire. Consider when $E_j^M \geq E_{j+1}^G - (s-1)$. In this case, MOVE_EXTERNAL just skips over E_j^M without counting it as one of the $s-1$ nets left of E_{j+1}^G, and then runs MARK-INTERNAL2 only on the nets following E_j^M. By running MARK-INTERNAL2 only on the nets following E_j^M, MOVE_EXTERNAL maintains its net location for E_j^M. Therefore, given the optimal net location for E_{j+1}^M left of E_j^M we must show that it would achieve no savings to try moving E_j^M further left, again.

Assume that if we moved E_j^M further left that a savings is achieved. If we move E_j^M further left, say to net k, then we are trying E_j^M at a location that it had been previously tried at in an earlier iteration of MOVE_EXTERNAL. Compare this attempt to place E_j^M at net k with its earlier attempt.

First, consider the earlier attempt to place E_j^M at net k. Since E_j^M is routed left, we know that E_j^M at net k would have achieved a savings in the additional cost of the j^{th} external wire over its current net choice for E_j^M. The results of placing E_j^M at net k was that the savings achieved by the j^{th} external wire and the other left-routed external wires following net k was less than the gain incurred by the right-routed external wires. Therefore, net k was not chosen for E_j^M.

Consider the effect that the placement of E_{j+1}^M would have on an attempt to place E_j^M at net k. If a savings were to occur on this attempt to place E_j^M at net k then the placement of E_{j+1}^M would have to allow the left-routed external wires following net k to move further left. Consider the nets such that $\rho_1(net) = E_{j+1}^M$ prior to this net being marked external. It can be seen from the definition of conflict numbers in Section 3.2 that the relationship between the maximum conflict numbers of these nets for the current attempt and the prior attempt is as follows:

$$C_{current}(net) = C_{prior}(net) - 1. \tag{6}$$

Thus, any external wires within these nets could not move leftward from the prior attempt to the current attempt. This is true because leftward movement is forced by an increase in maximum conflict numbers of earlier nets. Now, consider all nets such that $\rho_I(net) = E_j^M$ when it is marked internal. Again, from the definition of conflict numbers the relationship between the maximum conflict numbers of these nets for the current attempt and the prior attempt is as follows:

$$C_{current}(net) = C_{prior}(net) + 1. \tag{7}$$

Thus, every net that was forced external by net k+1 in the earlier iteration is also external in this iteration, and any net with a horizontal segment on the lowest track above the terminal row in the earlier iteration is forced external in this iteration. For example, if E_l^M was at net x in the earlier attempt to place E_j^M at net k, then on this attempt E_l^M is at net $x-1$ and E_{l+1}^M is at net x. Since net k was not chosen for E_j^M in the earlier attempt, the savings achieved by moving E_j^M to net k and E_l^M to net x and any movement of the other left-routed wires did not outweigh the gain incurred by the right-routed wires. The external wire now at net x was a wire routed further to the right than E_l^M, and if on this attempt there is a savings we know it was achieved by moving E_{l+1}^M to a net following net x. Therefore, an even greater savings could be achieved by stopping short of net

x. This is known from the results of the earlier attempt. Therefore, we know that by attempting to move E_j^M left, again, we cannot achieve the greatest savings in total additional cost. That is the choice for E_j^M is best at its fixed location.

We have shown that if the optimal routing could have achieved a savings in the manner described then we are not comparing the j^{th} external wire of the optimal routing to the j^{th} external wire of the MOVE_EXTERNAL routing. Thus, we have a contradiction.

Case 6($E_j^O = i$, and E_j^O is routed left and E_j^M is routed right): Assume the optimal routing achieves a savings in total additional cost over the MOVE_EXTERNAL routing. We have two cases to consider. The first being that the additional cost of wire E_j^O is less than that of wire E_j^M, and the second being that it is greater than that of wire E_j^M.

Assume wire E_j^O routed left has a smaller additional cost than that of E_j^M. Given this assumption, MOVE_EXTERNAL would have only placed E_j^M right of E_j^O if the additional cost in the right-routed wires E_{j+1}^M thru E_m^M gained more in additional cost than E_j^O saved. The routings match up until net E_j^O, therefore the savings had to come from external wires E_j^O thru E_m^O. The savings could only be achieved if the optimal external wires E_{j+1}^O thru $E_{j+(E_j^M - E_j^O)}^O$ ($E_j^M - E_j^O$ is the maximum number of internal nets available between E_j^O and E_j^M) were allowed

to move left past E_j^M, and they achieved the savings by being routed left. We may use the same argument as in Case 4 to show either no savings was achieved, or we are no longer comparing the j^{th} external wire in the optimal routing to the j^{th} external wire in the MOVE_EXTERNAL routing.

Assume E_j^O routed left has a greater additional cost than that of E_j^M routed right. Again, the savings had to come from nets E_j^O thru E_m^O. Since E_j^O is left of E_j^M we know that E_{j+1}^O thru E_m^O could be no further right than their corresponding external wires E_{j+1}^M thru E_m^M. Thus, the additional cost of the optimal external wires E_j^O thru E_m^O is greater than or equal to their corresponding external wires E_j^M thru E_m^M. Thus, the optimal routing could not have achieved any savings in total additional cost over the MOVE_EXTERNAL routing.

Case 7($E_j^O = i$, and E_j^O is routed right and E_j^M is routed left): Assume that the total additional cost of the optimal routing is strictly less than that of the MOVE_EXTERNAL routing. Since the routings match up until net i, the savings achieved by the optimal routing had to come from the external wires E_j^O thru E_m^O. Consider how MOVE_EXTERNAL works. When MOVE_EXTERNAL places E_j^M at net i, it routes E_j^M in the direction that minimizes the additional cost of E_j^M. This implies that E_j^M has a smaller additional cost than E_j^O. Since wire E_j^O is routed right from Lemma 1 we know E_{j+1}^O thru E_m^O must also be

routed right. By placing E_j^O left of E_j^M, the locations of E_{j+1}^O thru E_m^O can be no further right than their corresponding wires in the MOVE_EXTERNAL routing. Therefore, any right-routed wires in the optimal routing can achieve no savings in additional cost. Thus, the optimal routing achieved no savings in total additional cost over the MOVE_EXTERNAL routing.

Case 8($E_j^O = i$, and E_j^O and E_j^M are both routed right): Assume that by moving E_j^O further left, even though it is routed right, a savings in total additonal cost was achieved. This is only possible if the left-routed external wires to the left of E_j^O were allowed to move further left, but we know that this is not true since the optimal routing and the routing produced by MOVE_EXTERNAL are exactly the same until net E_j^O. Thus, no savings could have been achieved. ∎

We have shown that MOVE_EXTERNAL produces an optimal routing for a given instance with fixed channel width s such that $|E^O| = |E^G|$.

5.2 An Algorithm for an Arbitrary Number of External Wires

In the previous subsection we presented an algorithm MOVE_EXTERNAL that minimizes the *total additional cost* for a given instance with fixed channel width

s and a fixed number of external wires m, where $m = |E^G|$. Here, we consider whether a set of $|E^G|$ external wires guarantees that it is possible to produce an optimal routing. Observe the internal-external routings (fixed channel width) in Figure 15. Figure 15a is an internal-external routing with minimized *total additional cost* for an instance with a fixed channel width of 3 and 1 external wire. Figure 15b is an internal-external routing of the same instance except with 2 external wires , and the routing in Figure 15b has a smaller *total additional cost* than the routing in Figure 15a. In fact, the reader may verify that this routing is optimal by trying all other possible routings.

Here, we show that with only a slight modification that MOVE_EXTERNAL will produce an optimal internal-external routing for a given instance with fixed channel s without regard to the number of external wires. The only modification necessary for MOVE_EXTERNAL is the removal of the requirement that $|E^M| = |E^G|$ in line [24]. There are two things we must show that will prove that this is true. First, we must show that Lemma 2 holds for an optimal routing with an arbitrary number of wires. Second, we must show that if we allow an additional wire to go external that it is always routed right. That is, it would never create a savings to intentionally attempt to move it left.

Here, we restate Lemma 2 as Lemma 3 without the restriction on $|E^O|$.

Lemma 3 *For all* $1 \leq i \leq m$, $E_i^G - (s-1) \leq E_i^O \leq E_i^G$.

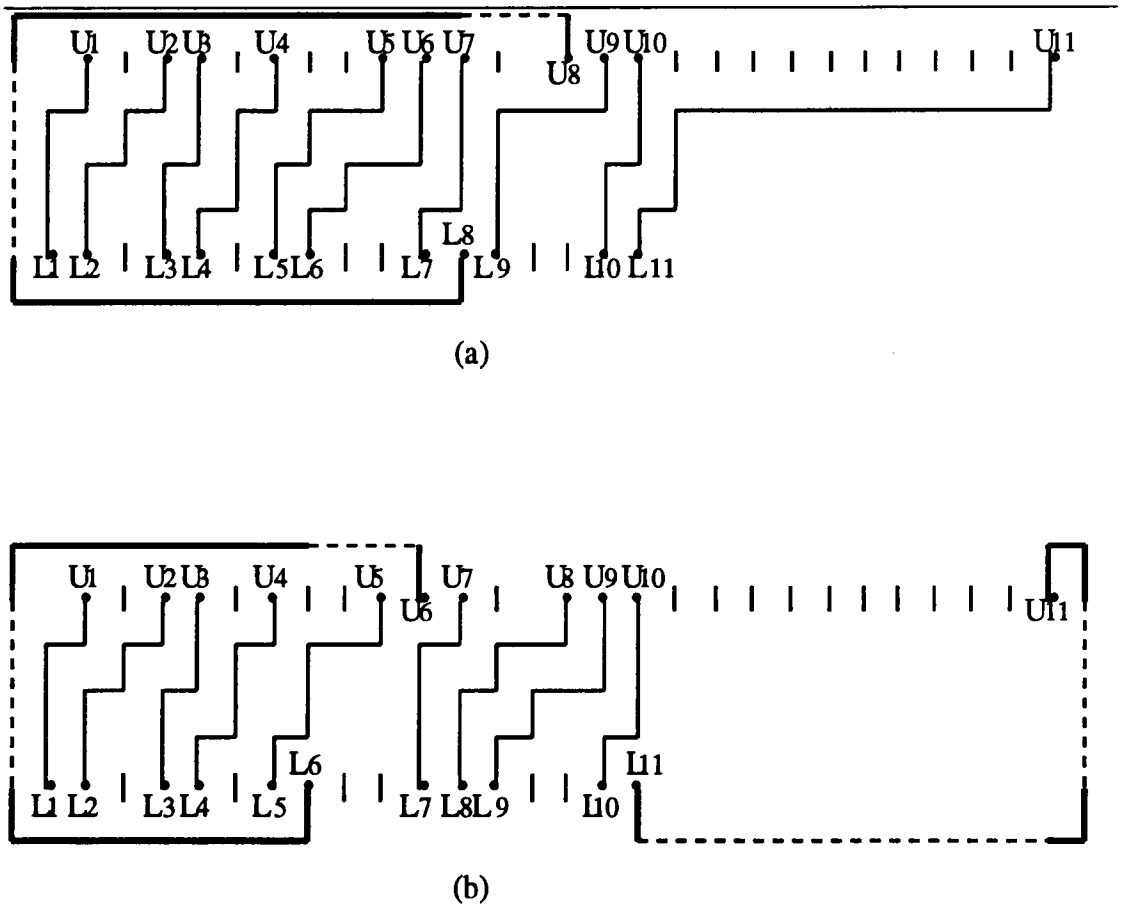


Figure 15: An example to illustrate the effect of $|E|$ (a) the best routing given a minimal set E (b) an optimal routing where E is not a minimal set.

Proof: Although we have shown that an optimal routing may have more external wires than the best routing with m external wires, it will suffice to show the savings cannot be achieved by marking any $E_i^O < E_i^G - (s-1)$. From Lemma 2, we know there must be at least one external wire in each of these ranges. Thus, there will be a corresponding external wire in the optimal routing for every E_i^{best} in the range of $E_i^G - (s-1)$ thru E_i^G. We know from Theorem 1 that we can produce a routing that minimizes the total additional cost. That is, we can position these wires within their respective range to minimize total additional cost for a routing with m external wires. Then wire E_i^O could not produce any savings. ∎

MOVE_EXTERNAL only attempts to move m wires, therefore we must show that any additional wire can never achieve any savings by moving left. That is, any additional wires are always routed right.

Lemma 4 *If the optimal internal-external routing contains more external wires than the best internal-external routing with a fixed number of external wires m, where $m = |E^G|$, then the additonal external wires must be routed right of the base.*

Proof: Assume that the placement of E_i^O forced an additional external wire, say E_{m+1}^O. This could only be a savings over the best internal-external routing with m external wires if the savings made by E_i^O and the left-routed external wires following E_i^O achieved a savings greater than the additional cost of E_{m+1}^O.

Consider if E^O_{m+1} is routed left, then the the additional cost of E^O_{m+1} would be as follows,

$$left(E_i) = 2((L_{E^O_{m+1}} - L_1) + (m+1)) + 4(m+1). \tag{8}$$

Consider only the factor $(L_{E^O_{m+1}} - L_1)$ of equation 8. Clearly, this factor alone would contain the total savings made by the placement of E^O_i. Therefore, either E^O_{m+1} must be routed right or the optimal routing contains only m external wires.

∎

From Lemma 3 and Lemma 4, we can conclude that the modification described earlier is the only modification necessary. This must be true since the proof presented for Theorem 1 never refers to the number of external wires directly.

6 Future Work

In this section we conclude this paper by exploring the areas into which our work may be expanded.

6.1 Variable Channel Width

Here, we restricted our investigation to internal-external routings with fixed channel width. There is an obvious method for extending our algorithm to the variable channel width problem. That is, given an instance of n terminals apply our algorithm for all reasonable channel widths. The reasonable channel widths are 1 thru n. This would produce an optimal routing over all possible routings for that instance. This approach would produce an algorithm with a time complexity of $O(n^2 \cdot s \cdot m)$. Therefore, it would be worthwhile to investigate whether an alternative approach can improve upon the performance.

6.2 Mixed Routings

Our algorithm will produce an optimal internal routing if one exists for a fixed channel width, and it will produce an optimal internal-external routing for a fixed channel width. In the previous subsection we explained how the algorithm would work for internal-external routings with a variable channel width. However, further investigation is required to see how it can be adapted for mixed routings.

6.3 Performance Bounds

In Subsection 6.1 we gave an approach to produce optimal routings for instances with a variable channel width. The next logical step is to investigate the relative performance of the internal-external routings and the internal routings. In [2], it was shown for optimization criteria of area of the enclosing rectangle and vertical density that each successively richer class of routings achieved significant improvement over the others. The mixed routings achieved an improvement of $O(\sqrt{n})$ over internal routings, and it achieves $O(\sqrt[4]{n})$ over internal-external routings. It would be interesting to see if the same level of improvement holds when the optimization criterion is total wire length.

LIST OF REFERENCES

References

[1] P B. Arnold. Complexity for single row routing. Technical Report TR-22-82, Center for Research in Computing Technology, Harvard University, Boston, Mass., 1982.

[2] J. Blair and E. Lloyd. Generalized river routing - algorithms and performance bounds. In *VLSI Algorithms and Architectures*, Springer-Verlag Lectures in Computer Science, pages 317–328, July 1986.

[3] J. Blair and E. Lloyd. Generalized river routing with a generalized model. submitted for publication, July 1993.

[4] D. Dolev, K. Karplus, A. Siegel, A. Strong, and J. D. Ullman. Optimal wiring between rectangles. In *Proceedings of the Thirteenth Annual ACM Symposium on Theory of Computing*, pages 312–317. ACM, 1981.

[5] D. Dolev and A. Siegel. The separation for general single-layer wiring barriers. In H. T. Kung et. al., editor, *VLSI Systems and Computations*, pages 143–152. Computer Science Press, 1981.

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

[7] E. L. Lloyd and S. S. Ravi. One-layer routing without component constraints. *Journal of Computer and System Sciences*, 28(3):420–438, June 1984.

[8] F. M. Maley. *Single-Layer Wire Routing and Compaction.* MIT Press, 1990.

[9] R. Y. Pinter. Optimal routing in rectilinear channels. In H. T. Kung et. al., editor, *VLSI Systems and Computations*, pages 160–177. Computer Science Press, 1981.

[10] R. Raghavan and S. Sahni. The complexity of single row routing. *IEEE Transactions on Circuits and Systems*, CAS-31(5):462–472, May 1984.

[11] A. L. Rosenberg. Three-dimensional vlsi: A case study. *Journal of the Association for Computing Machinery*, 30(3):397–416, July 1983.

[12] M. Tompa. An optimal solution to a wire routing problem. In *Proceedings of the Twelfth Annual ACM Symposium on Theory of Computing*, pages 161–176. ACM, 1981.

Vita

Leanne Tuggle Metzger was born in Memphis, Tennessee on December 15, 1964. In 1967, she and her family moved to Oak Ridge, Tennessee. She attended elementary and secondary schools in the Oak Ridge School District and graduated from Oak Ridge High School in June 1982. She received a Bachelor of Science degree in Computer Science from Mississippi State University in May 1987, and she received a Master of Science degree in Computer Science from the University of Tennessee at Knoxville in May 1994.

She is currently pursuing a Doctor of Philosophy degree in Computer Science from the University of Tennessee at Knoxville, where she is also employeed as Graduate Teaching Assistant.