



4-1995

Planar Over the Cell Routing

Danda

Follow this and additional works at: https://scholarworks.wmich.edu/masters_theses



Part of the Computer Sciences Commons

Recommended Citation

Danda, "Planar Over the Cell Routing" (1995). *Master's Theses*. 4860.
https://scholarworks.wmich.edu/masters_theses/4860

This Masters Thesis-Open Access is brought to you for free and open access by the Graduate College at ScholarWorks at WMU. It has been accepted for inclusion in Master's Theses by an authorized administrator of ScholarWorks at WMU. For more information, please contact wmu-scholarworks@wmich.edu.



PLANAR OVER THE CELL ROUTING

by

Srinivasa R. Danda

A Thesis
Submitted to the
Faculty of The Graduate College
in partial fulfillment of the
requirements for the
Degree of Master of Science
Department of Computer Science

Western Michigan University
Kalamazoo, Michigan
April 1995

To
my grand mother Narayanamma Danda,
and in the fond memory of
my grand father Veeraiah Danda

ACKNOWLEDGEMENTS

First and foremost, I would like to express my gratitude to my advisor, Dr. Naveed Sherwani, from whom I have learned many valuable skills, both research and otherwise. His consistent support, accessibility and his constant motivation, has been a source of encouragement for me. I expect our professional working relationship to continue for many years to come.

Dr. Donald Nelson, the Department Chair, has my gratitude for his continued support in providing research facilities that has been a great help in finishing this thesis work and conducting other research.

I am grateful to the members of my thesis committee, Dr. Alfred Boals and Dr. Ajay Gupta, who accepted this additional task with high enthusiasm.

My sincere gratitude to Sreekrishna for his help on this thesis. I would like to thank both the old and the new 'nitegroup' members, Aman, Arun, Anand, Sandeep, Qiyong, Siddharth, and Praveen for their help and friendship.

Special thanks to Sue and Phyllis from CS Department, who helped me in many administrative situations: their pleasant characters and helpful personalities are assets to us all.

Finally, I would like to thank my parents, sisters, and brother for their encouragement, guidance and support which has always helped me in accomplishing my goals.

Srinivasa R. Danda

PLANAR OVER THE CELL ROUTING

Srinivasa R. Danda, M.S.

Western Michigan University, 1995

Planar over the cell routing in standard cell layouts is an important problem and it has been studied quite extensively. In two layer standard cell design methodology, M1 layer is typically used for connections internal to the cell, and the M2 layer is available for routing over-the-cell. In this thesis, we consider the Two Row Maximum Planar Subset (TRMPS) problem in Over-The-Cell routing. The TRMPS problem requires selection of the maximum planar subset of nets, which can be routed between two rows of terminals in a cell row. This problem was first encountered by Cong, Liu, and Preas [3]. They stated the complexity of this problem to be unknown, and presented a $\min\{1, \frac{k}{d(S)}\}$ approximation algorithm, where k is the number of tracks available over the cell area and $d(S)$ is the density of a solution S .

We show that TRMPS problem can be solved optimally in polynomial time. We present a $O(kn^2)$ dynamic programming algorithm for the TRMPS problem, where n is the number of nets. Our algorithm can also be extended to solve the TRMPS problem, in the presence of pre-routed nets, a chosen subset of nets, as well as for planar channel routing. We also apply our technique to obtain a 0.5 approximation, for over the cell routing in middle terminal model, thus improving the best known existing algorithm. The weighted version of the TRMPS problem, as well as, all the extensions can also be solved in $O(kn^2)$ time.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	iii
LIST OF FIGURES	iv
CHAPTER	
I. INTRODUCTION	1
Over The Cell Routing	1
Preliminaries	5
II. THE OPTIMAL ALGORITHM	8
Overview of the Algorithm	8
Single Row Maximum Independent Subset	10
Two Row Maximum Planar Subset	13
Maximum Planar Subset in an L-Shaped Region	21
The Optimal Algorithm	27
III. EXTENSIONS	30
TRMPS With Chosen Subset and Prerouted Nets	30
Chosen Subset of Nets	31
Pre-Routed Nets	33
Zero Dogleg Planar Channel Routing	36
Approximation Algorithm for OTC Routing in MTMs	41
ALGO-MTM	45
IV. CONCLUSIONS	49
BIBLIOGRAPHY	50

LIST OF FIGURES

1.	An Example of the TRMPS Problem.	2
2.	BTM-HCVC Model.	6
3.	Schematic Overview of the Algorithm ALGO-TRMPS.	10
4.	Single Row Maximum Independent Set.	11
5.	L-Shaped Regions.	14
6.	Cases 1 and 2 in ALGO-TRMPS.	15
7.	Cases 3(a),(b) and (c).	16
8.	The Four Cases of L-shaped Solutions.	23
9.	Algorithm ALGO-TRMPS.	27
10.	Cases in Crossing Nets.	39
11.	Middle Terminal Model.	42
12.	Four Routing Choices.	45
13.	Four Routing Choices Representation in a Circle.	46
14.	An Instance of MTM Problem and Solution.	47

CHAPTER I

INTRODUCTION

Over The Cell Routing

Channel routing in standard cell layouts is an important problem in VLSI design and as a result, it has been studied quite extensively (see chapter 7 of [12]). In fact, existing channel routers can produce solutions, very close to optimal, for most channels [11]. However, further reduction in channel height has been obtained by routing some nets in over-the-cell areas. This style of routing is called *over-the-cell* routing [13, 5, 1, 4]. With the availability of more metal routing layers, over-the-cell routing has not only become feasible, but also necessary to achieve high density layout.

The over the cell channel routing problem(OTC-CRP), is a generalization of the conventional channel routing problem(CRP), and it follows naturally from the intractability of CRP, that the OTC-CRP is also NP-hard. Therefore research has been concentrated on the development of heuristic algorithms.

In two layer standard cell design methodology, M1 layer is typically used for connections internal to the cell, and the M2 layer is available for routing over-the-cell. Figure 1(a) shows the net connections, in three consecutive cell rows, R_0, R_1 , and R_2 , of a standard cell layout. Figure 1(b) shows the set of nets that are suitable for routing over the cell row R_1 . In a HCVC (Horizontally Connected Vertically Connected) model (See Section I), the main problem in two layer over the cell routing is to select a maximum planar subset of nets which are suitable for routing in a single layer, available over the cell rows [3]. The

remaining connections are completed in the channel. We call this problem, the Two Row Maximum Planar Subset (TRMPS) problem. Figure 1(c) shows the maximum planar subset of nets, that can be routed over R_1 , which is an optimal solution, for the instance of the TRMPS problem, shown in Figure 1(b). Notice, that the tracks are shared between the top row nets and the bottom row nets, so as to efficiently utilize the over-the-cell area.

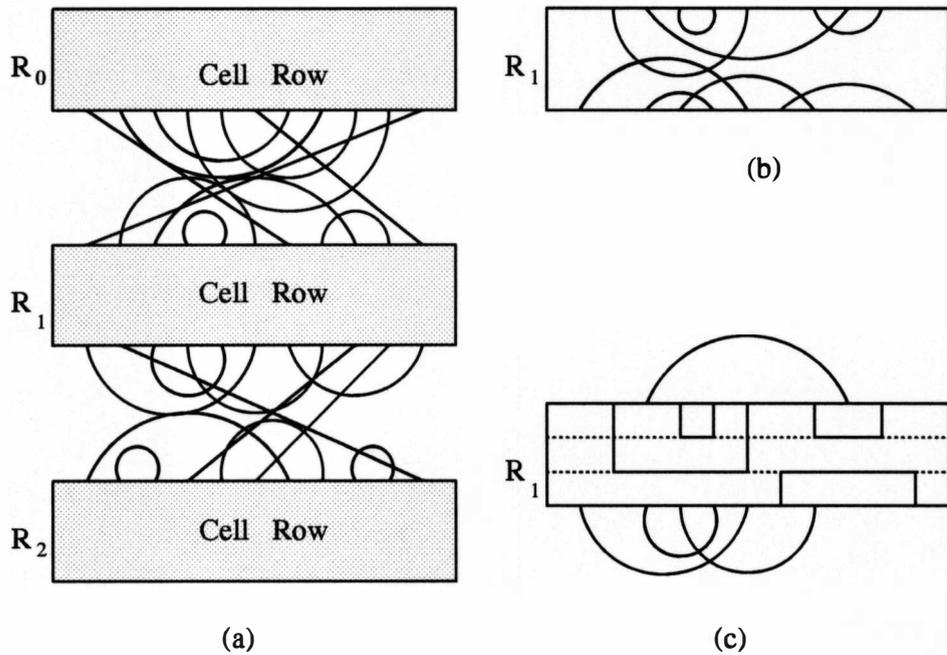


Figure 1. An Example of the TRMPS Problem.

The TRMPS problem, is formally defined as follows. Given two rows of terminals $\mathcal{T} = \{t_1, t_2, \dots, t_L\}$ and $\mathcal{B} = \{b_1, b_2, \dots, b_L\}$ and two sets of nets $\mathcal{N}_{\mathcal{T}} = \{(t_i, t_j) \mid t_i, t_j \in \mathcal{T}\}$ and $\mathcal{N}_{\mathcal{B}} = \{(b_i, b_j) \mid b_i, b_j \in \mathcal{B}\}$, where $|\mathcal{N}_{\mathcal{T}} \cup \mathcal{N}_{\mathcal{B}}| = n$, and k tracks between the two rows, find the maximum planar subset $\mathcal{N}_{\mathcal{P}} \subseteq \mathcal{N}_{\mathcal{T}} \cup \mathcal{N}_{\mathcal{B}}$ of the two sets in k tracks. We call this problem as the Two Row Maximum

Planar Subset problem (TRMPS). In the weighted version of this problem, each net $N_i \in \mathcal{N}_{\mathcal{T}} \cup \mathcal{N}_{\mathcal{B}}$ is assigned a weight W_i , and it is required to find the maximum weighted planar subset. We call this problem as the Weighted Two Row Maximum Planar Subset problem (WTRMPS).

The TRMPS problem was first encountered in Over the Cell routing in the BTM-HCVC model, by Cong, Liu and Preas [3]. They stated that the complexity of the problem is unknown and presented an approximation algorithm with a performance ratio of $\min\{1, \frac{k}{d(S)}\}$, where k is the number of tracks available and $d(S)$ the density of the solution, for solving this problem. Since then, the complexity of this problem has remained open.

The single-row version of the TRMPS problem is to find a maximum planar subset of nets, on a row of terminals, using a fixed number of tracks, on one side of the terminals. This problem is called the One-Row Fixed height Planar Routing (OFPR) problem, and a polynomial algorithm, that solves this problem optimally, was presented in [3]. Infact, we use this algorithm in the early stages of solving the TRMPS problem.

There are several simpler variants of the TRMPS problem, which can be easily solved in polynomial time. The simplest variant is, when there is only one track between the terminal rows. In this case, the optimal solution can be obtained, by taking the maximum independent set in the interval graph, constructed from the corresponding intervals of the nets in $\mathcal{N}_{\mathcal{T}} \cup \mathcal{N}_{\mathcal{B}}$. Another variant of this problem, that can also be solved easily, is the topological version of the problem. Here, the nets are not assigned to any of the tracks, but are topologically routed. In this case, the optimal solution can be found, by using the algorithm for solving the OFPR problem, independently for the two terminal rows \mathcal{T} and \mathcal{B} , assuming that atleast d number of tracks are available between the terminal rows,

where d is the density of maximum topologically planar subset of combined set of nets. Yet another variant of this problem, has the region between the terminal rows, vertically divided into two sub-regions. This is called the HCVD (Horizontally Connected Vertically Divided) model [3]. In this case, the upper and lower regions can be independently routed, by applying the algorithm for solving the OFPR problem, to the top and bottom terminal rows respectively.

In all the above variants, there is no *track sharing*, *i.e.*, no track is shared by the nets belonging to both the top and bottom terminal rows. It is precisely the track sharing and not the net selection, that causes the main difficulty in optimally solving the TRMPS problem. Infact, the TRMPS problem is very close to another problem called the Two Maximum Planar Subsets problem (TMPS). Given a row of terminals and a set of nets, the TMPS problem is to find two planar subsets of the nets, with maximum combined cardinality. This problem is equivalent to the problem of finding the maximum bipartite subgraph in a circle graph (or equivalently in an overlap graph), which is a known NP-complete problem [10, 8].

We show that the TRMPS problem can be solved optimally in $O(kn^2)$ time. We use a dynamic programming approach followed by a backtracking procedure to solve this problem. We have extended our algorithm to solve several variants of the TRMPS problem listed below, with no additional penalty in the time complexity.

1. Zero Dogleg Planar Channel Routing: There exist some nets of the form (t_i, b_j) , such that, $t_i \in \mathcal{T}$ and $b_j \in \mathcal{B}$. This problem arises in planar channel routing, which has been studied extensively [2, 9, 14].

2. TRMPS with Pre-routed Nets: Some of the nets which are critical, are already routed and the objective of this problem is to find the maximum planar subset of the remaining set of nets such that, the overall solution which includes

the pre-routed nets is planar and is maximum.

3. TRMPS with a Chosen Subset of Nets: This is a generalization of the TRMPS problem with pre-routed nets. Unlike the case of pre-routed nets, the tracks in which the chosen subset of nets are routed are not initially specified.

4. An Improved Approximation for Over The Cell Routing in MTM (Middle Terminal Model) Layouts: If the terminals on the same column in a cell row are equipotential, as in MTM layouts, then the problem of finding the maximum planar independent subset is shown to be NP-hard [13]. In the same paper, authors developed an approximation algorithm with a performance ratio of $\rho \geq 0.5 \max\{1, \frac{k}{d}\}$, where k is the number of tracks available between the terminal rows and d is the net density. In this thesis, we present a *0.5* approximation algorithm for this problem.

The rest of the thesis is organized as follows. In the next Section, we discuss the BTM-HCVC model and define the required terminology. In the next chapter we present a detailed description of our algorithm. In Chapter III, we discuss the extensions of our algorithm to several variants of the TRMPS problem, and conclude with Chapter IV.

Preliminaries

As mentioned earlier, the TRMPS problem was encountered in over-the-cell routing in the BTM-HCVC model. The HCVC (Horizontally Connected Vertically Connected) model, which is a type of BTM (Boundary Terminal Model), is shown in Figure 2. In this model, the terminals and the feedthroughs are on the M1 layer. The power bus is on the M2 layer, in the channel just above the top terminal row, and the ground bus is on the M2 layer, in the channel just below the bottom terminal row. In the channels, the horizontal wire segments (trunks) are routed

on M2, and the vertical wire segments (branches) are routed on M1. Hence, the entire M2 layer over the cells, is available for over-the-cell routing.

Let L denote the total number of columns in a cell row, numbered from left to right. In BTM-HCVC, the terminals are located at the intersection points of the upper or the lower horizontal boundaries of a cell row and the vertical columns. If a terminal is not used by any net, then that terminal is called a *vacant terminal*. If both the upper and lower terminals of a column are vacant, then that column is called a *vacant abutment*. The total number of tracks available in the OTC area of a cell row, for routing, is denoted by k (cell height), and the tracks are numbered from top to bottom. Then, an instance of the TRMPS problem can be formally represented as a 7-tuple $\mathcal{I} = (\mathcal{T}, \mathcal{B}, \mathcal{N}_{\mathcal{T}}, \mathcal{N}_{\mathcal{B}}, k, n, L)$. We call an instance of the TRMPS problem, as a *Canonical Instance*, if there are no vacant abutments in that instance. If n is the number of nets in a canonical instance \mathcal{I} , then the number of columns (L), can be at most $2n$. This is because, in the worst case, each column has at most one vacant terminal, either in the top or the bottom terminal row.

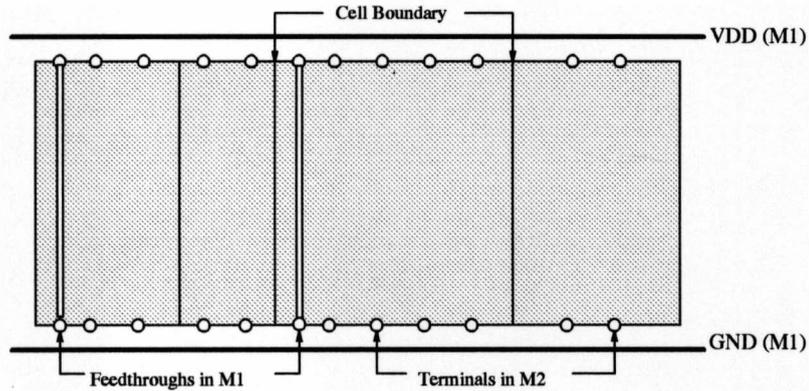


Figure 2. BTM-HCVC Model.

In this thesis, we consider only canonical instances, and we assume that all the nets are two terminal nets, unless explicitly stated otherwise. It is easy to see that all results reported in this thesis are applicable to non-canonical instances. A net is denoted by a pair of terminals. A net (t_i, t_j) , where $1 \leq i, j \leq L$, is called a top net. Similarly, a net (b_i, b_j) , where $1 \leq i, j \leq L$, is called a bottom net. We define the *span* of a two terminal net, as the absolute difference between the column numbers on which the terminals of the net are located. For example, the span of the net $N_\alpha = (t_i, t_j)$, is given by,

$$\text{span}(N_\alpha) = |i - j|$$

We define a region \mathcal{R}_m of a cell row, as a rectangular region of the cell row, containing the columns in the range $[1, m]$, where $1 \leq m \leq L$. A net (t_i, t_j) (or b_i, b_j), is said to be *completely contained* in the region \mathcal{R}_m , if $1 \leq i, j \leq m$.

CHAPTER II

THE OPTIMAL ALGORITHM

In this chapter, we discuss the optimal algorithm in detail. We will give an overview of the algorithm in the next section, and detailed description of each phase will be discussed in the subsequent sections.

Overview of the Algorithm

In this section, we present an overview of our algorithm for solving the TRMPS problem optimally. We use $T(j)$ to denote the optimal TRMPS solution in a rectangular region \mathcal{R}_j . The $T(j)$ solution is computed for all j , $1 \leq j \leq L$, using a dynamic programming technique. Finally, the $T(L)$ solution gives the optimal solution, for a given instance \mathcal{I} of the TRMPS problem. In order to compute the $T(j)$ solution, the region \mathcal{R}_j is partitioned into two or three sub regions. depending on the existence of top nets and bottom nets, completely contained in \mathcal{R}_j , with one of their terminals at column j , as shown in Figure 3.

Let $N_\alpha = (t_i, t_j)$ be the only net with a terminal at column j , and which is completely contained in \mathcal{R}_j . In this case, \mathcal{R}_j is divided into an L-shaped region R , and a rectangular region r which consists of a single row of terminals (Figure 3(a)). The optimal $T(j)$ solution, may or may not contain N_α . If N_α is included, then the $T(j)$ is summation of the optimal solutions in the L-shaped region R and the rectangular region r , and the net N_α itself. If N_α is not included, then the $T(j)$ solution is the same as the $T(j - 1)$ solution. The maximum of the above two solutions, is taken as the optimal $T(j)$ solution.

Let $N_\alpha = (t_i, t_j)$ and $N_\beta = (b_m, b_j)$ be the nets with terminals at column j , and which are completely contained in \mathcal{R}_j . Then, the optimal $T(j)$ solution may include

1. **None of the nets N_α and N_β :** In this case, the $T(j)$ solution is the same as the $T(j - 1)$ solution.

2. **Only the net N_α :** In this case, the $T(j)$ solution can be computed as shown in Figure 3(a).

3. **Only the net N_β :** In this case also, the $T(j)$ solution can be computed as shown in Figure 3(a).

4. **Both the nets N_α and N_β :** In this case, if $i \neq m$, then \mathcal{R}_j is partitioned into an L-shaped region R , and two rectangular regions r_1 and r_2 , which consist of a single row of terminals (Figure 3(b)). If $i = m$, then \mathcal{R}_j is partitioned into a rectangular region R , which consists of two rows of terminals, and two rectangular regions r_1 and r_2 , which consist of a single row of terminals (Figure 3(c)). Then, the $T(j)$ is simply summation of the optimal solutions in the regions R , r_1 and r_2 , and the nets N_α and N_β .

The optimal $T(j)$ solution, is the maximum among all the above four solutions.

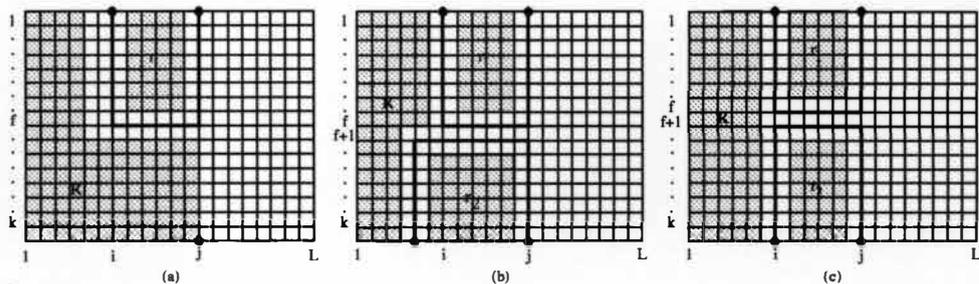


Figure 3. Schematic Overview of the Algorithm ALGO-TRMPS.

From the above discussion, it is clear that, the single row solutions and the solutions in the L-shaped regions need to be computed, before computing the two row solutions. Our algorithm consists of the following three phases.

1. In the first phase, we compute the single row solutions of the terminal rows \mathcal{T} and \mathcal{B} , individually. Each single row solution of a terminal row, is an (i, j, t) solution, where $1 \leq i, j \leq L$ and $1 \leq t \leq k$. This problem, is equivalent to the *one-row fixed-height planar routing (OFPR) problem*, described in Chapter 1. For the sake of completeness we describe the algorithm presented in [3] in this chapter. Using this algorithm we compute the (i, j, t) solutions of the terminal rows \mathcal{T} and \mathcal{B} , which we denote by $S_t(i, j, t)$ and $S_b(i, j, t)$ respectively.

2. In this phase, we compute the maximum two row planar subset ($T(j)$) for the given terminal rows, where $1 \leq j \leq L$ by using a dynamic programming approach. Here, the $S_t(i, j, t)$ and $S_b(i, j, t)$ solutions, computed in the first phase will be used. As described above, finding the $T(j)$ solution also involves finding the maximum planar subset in L-shaped regions. We describe this in detail, in this chapter.

3. The solution obtained in phase 2, gives the number of nets in the optimal solution for a given instance of the TRMPS problem. In this phase, the actual planar subset of nets in the optimal solution, is determined by backtracking.

Single Row Maximum Independent Subset

In this section, for the sake of completeness we present an overview of the algorithm presented in [3], which is essentially an extension of the algorithm for maximum independent set in circle graphs [6].

From a routing perspective, this problem is equivalent to assigning the maximum number of intervals to k tracks such that, if interval (i, j) is assigned

to track f , then no interval assigned to tracks $1, 2, \dots, f - 1$ should intersect columns i and j . Let $MIS(i, j, f)$ denote the solution of the OFPR problem resulting from restricting the intervals to be in the range of $[i, j]$ and allowing f tracks for routing, where $1 \leq i, j \leq L$ and $1 \leq f \leq k$. The (i, j, f) solution is computed using dynamic programming. Notice that, to compute $MIS(i, j, f)$, we have the following three cases:

1. If j is vacant, then

$$MIS(i, j, f) = MIS(i, j - 1, f)$$

2. There exists a net N_α with terminals j and m but $m \notin [i, j]$. Then,

$$MIS(i, j, f) = MIS(i, j - 1, f) \text{ if } m \notin [i, j]$$

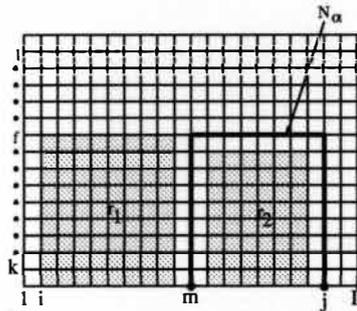


Figure 4. Single Row Maximum Independent Set.

3. There exists a net N_α with terminals j and m such that $m \in [i, j]$, then

(a) Excluding the net N_α in the solution leads to

$$MIS(i, j, f) = MIS(i, j - 1, f)$$

(b) Including the net N_α in the solution results in

$$MIS(i, j, f) = MIS(i, m - 1, f) + MIS(m + 1, j - 1, f - 1) + 1$$

As shown in Figure 4, if $m \in [i, j]$, we need to check if including N_α will lead to a better solution or not. Therefore,

$$MIS(i, j, f) = \max (MIS(i, j - 1, f), MIS(i, m - 1, f) + MIS(m + 1, j - 1, f - 1) + 1 \text{ if } j' \in [i, j])$$

The complexity of this algorithm is given by the following theorem, stated in [3].

Theorem 1 [3] *The two-terminal net OFPR problem can be solved in $O(kn^2)$ time, where n is the number of nets and k is the number of available tracks.*

Using the above algorithm the maximum k -planar subsets S_t and S_b are computed, for the top and bottom terminal rows respectively, and all the intermediate solutions are stored.

The weighted version of the OFPR problem is to find the maximum weighted independent subset of nets, that can be routed in k tracks, such that the selected nets will not overlap with each other.

Corollary 1 *The weighted version of the OFPR problem can be solved in $O(kn^2)$ time.*

Two Row Maximum Planar Subset

In this section, we describe the algorithm ALGO-TRMPS, that computes the $T(j)$ solutions. Since, computing the $T(j)$ solution, involves computing the solutions in L-shaped regions. we first describe a scheme to represent an L-shaped region.

Figure 5 shows two types of L shaped regions. We denote an L-shaped region shown in Figure 5(a), by the 3-tuple (i, j, f) , where

1. i is the column number of the terminal t_i , which is the rightmost corner of the L-shaped region, in the top terminal row.
2. j is the column number of the terminal b_j , which is the rightmost corner of the L-shaped region, in the bottom terminal row.
3. f is the track, that forms part of the horizontal boundary of the L-shaped region (See Figure 5(a)).

The maximum planar subset in the L-shaped region, shown in Figure 5(a), is denoted by $L(i, j, f)$. Following the same convention described above, the inverted L-shaped region, shown in Figure 5(b) is denoted by (j, i, f) , and the solution in this region is denoted by $L(j, i, f)$. The method of computing solutions in L-shaped regions is described in this chapter.

While computing the $T(j)$ solution in the rectangular region \mathcal{R}_j , the algorithm deals with the following three cases.

Case 1: There exists a top net $N_\alpha = (t_i, t_j)$, which is completely contained in \mathcal{R}_j (Figure 6(a)).

Case 2: There exists a bottom net $N_\beta = (b_i, b_j)$, which is completely contained in \mathcal{R}_j (Figure 6(b)).

Case 3: There exists a top net $N_\alpha = (t_i, t_j)$, and a bottom net $N_\beta =$

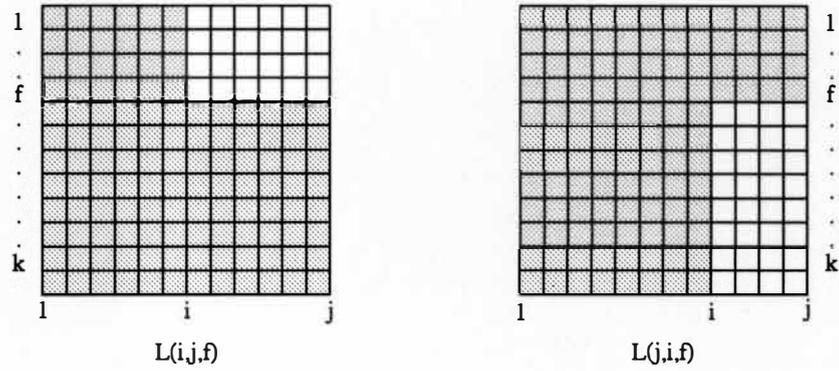


Figure 5. L-Shaped Regions.

(b_m, b_j) , which are completely contained in \mathcal{R}_j (Figure 6(c)).

Let us consider each of the above listed cases in detail.

Case 1: Depending on whether the net N_α is in the optimal $T(j)$ solution, or not, the algorithm has to deal with the following sub-cases.

Case 1(a): Excluding the net N_α leads to

$$T(j) = T(j - 1)$$

Case 1(b): If the net N_α is included, such that, it is assigned to a track f , $1 \leq f \leq k$, then we have the following solution, which we denote by $T'(j)$.

$$T'(j) = S_t(i + 1, j - 1, f - 1) + 1 + L(i - 1, j - 1, f + 1)$$

By considering all possible track assignments, the track to which N_α can be assigned is found, so as to maximize the $T(j)$ solution.

Then, the $T(j)$ solution obtained by choosing N_α , which we denote by $T''(j)$, is

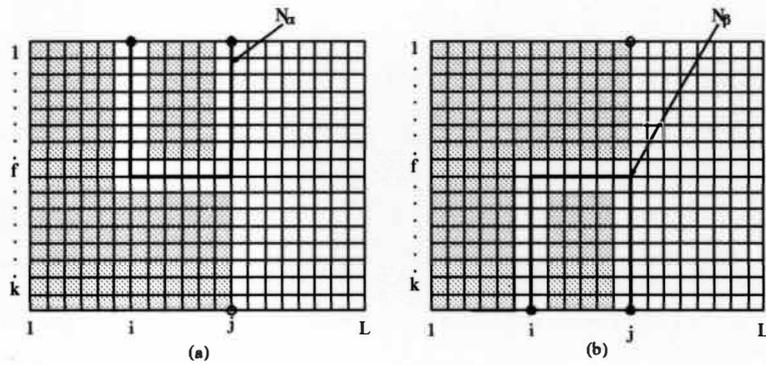


Figure 6. Cases 1 and 2 in ALGO-TRMPS.

given by,

$$T''(j) = \max_{f=1}^k \{T'(j)\}$$

The optimal $T(j)$ solution will then be the maximum of the two solutions obtained by including and excluding the net N_α . Therefore,

$$T(j) = \max\{T(j-1), T''(j)\}$$

Case 2: This is symmetric to Case 1.

Case 3: Here, the following three sub-cases are possible (Figure 7).

Case 3(a): $Span(N_\alpha) > Span(N_\beta)$

Case 3(b): $Span(N_\alpha) < Span(N_\beta)$

Case 3(c): $Span(N_\alpha) = Span(N_\beta)$

For each of the above sub-cases, the following solutions are computed.

$W_0(j)$: Two row solution of \mathcal{R}_j , which does not consist of N_α and N_β .

$W_1(j)$: Two row solution of \mathcal{R}_j , which consists of only N_α .

$W_2(j)$: Two row solution of \mathcal{R}_j , which consists of only N_β .

$W_{12}(j)$: Two row solution of \mathcal{R}_j , which consists of N_α and N_β .

The maximum of W_0, W_1, W_2 and W_{12} solutions is the optimal $T(j)$ solution.

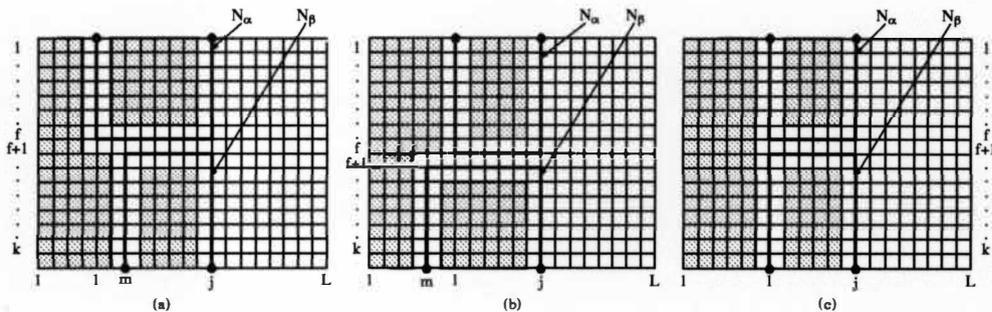


Figure 7. Cases 3(a),(b) and (c).

If both the nets N_α and N_β are included in the optimal solution $T(j)$, then a simple observation, regarding the track assignment of the nets N_α and N_β , is stated in the following lemma.

Lemma 1 *If $N_\alpha = (t_1, t_j)$ and $N_\beta = (b_m, b_j)$, are two nets, which are completely contained in \mathcal{R}_j , and the optimal $W_{12}(j)$ solution has the net N_α in track f_1 , and N_β in track f_2 such that $1 \leq f_1 < f_2 \leq k$, then,*

1. *If $\text{span}(N_\alpha) > \text{span}(N_\beta)$, then, the solution in which, the net N_β is assigned to a track $f_1 + 1$ is also an optimal $W_{12}(j)$ solution.*
2. *If $\text{span}(N_\alpha) < \text{span}(N_\beta)$, then, the solution in which, the net N_α is assigned to a track $f_2 - 1$ is also an optimal $W_{12}(j)$ solution.*
3. *If $\text{span}(N_\alpha) = \text{span}(N_\beta)$, then, the solution in which, the net N_β is assigned to a track $f_1 + 1$, and the solution in which, the net N_α is assigned to a track $f_2 - 1$ are also optimal $W_{12}(j)$ solutions.*

Proof: If $\text{span}(N_\alpha) > \text{span}(N_\beta)$ then, no other net could have been assigned to

any of the contiguous tracks, in the range (f_1, f_2) , between the columns in the range $[m, j]$, as this would violate the planarity property. Therefore, if the net N_β is assigned to the track $f_1 + 1$, then the resulting $W_{12}(j)$ solution, would still be optimal. The other cases stated in this lemma, can also be proved in a similar manner. \square .

We now consider the three sub-cases listed above, in detail.

Case 3(a): In this case, since $span(N_\alpha) > span(N_\beta)$, column l is to the left of column m (Figure 7(a)). The $W_0(j)$ solution, in which both the nets are excluded is given by,

$$W_0(j) = T(j - 1)$$

The $W_1(j)$ solution can be computed as follows. Suppose, the net N_α is assigned to track f , $1 \leq f \leq k$, then, the following solution, which we call $W'_1(j)$.

$$W'_1(j) = S_t(l + 1, j - 1, f - 1) + 1 + L(l - 1, j - 1, f + 1)$$

By trying all possible track assignments, the track to which N_α can be assigned is found, so as to maximize the $W_1(j)$ solution. The $W_1(j)$ solution is given by,

$$W_1(j) = \max_{f=1}^k \{W'_1(j)\}$$

The $W_2(j)$ solution can be computed in a similar manner as $W_1(j)$.

The $W_{12}(j)$ solution can be computed as follows. From lemma 1, it is clear that in the optimal $W_{12}(j)$ solution, the nets N_α and N_β are assigned to adjacent tracks. Suppose, the net N_α is assigned to track f , and N_β in track $f + 1$, then

the following solution, which we call W'_{12} is obtained.

$$\begin{aligned} W'_{12}(j) &= L(l-1, m-1, f+1) + S_t(l+1, j-1, f-1) \\ &+ S_b(m+1, j-1, f+2) + 2 \end{aligned}$$

The adjacent tracks, to which N_α and N_β can be assigned is found, so as to maximize $W_{12}(j)$.

$$W_{12} = \max_{j=1}^{k-1} \{W'_{12}(j)\}$$

Then, the optimal $T(j)$ solution will be the maximum of W_0, W_1, W_2 and W_{12} solutions. Therefore,

$$T(j) = \max\{W_0(j), W_1(j), W_2(j), W_{12}(j)\}$$

Case 3(b): This is symmetric to Case 3(a).

Case 3(c): In this case, $\text{span}(N_\alpha) = \text{span}(N_\beta)$. (Figure 7(c)).

Here, the $W_0(j), W_1(j)$ and $W_2(j)$ solutions are the same as for Case 3(a) and Case 3(b) However, the W_{12} solution differs slightly. According to the Lemma 1, the nets N_α and N_β can be assigned to adjacent tracks (say f and $f+1$ respectively). Then the W'_{12} will be

$$\begin{aligned} W'_{12}(j) &= T(l-1) + S_t(l+1, j-1, f-1) \\ &+ S_b(l+1, j-1, f+2) + 2 \end{aligned}$$

By trying all possible track assignments, we can find two adjacent tracks, on which

we can place N_α and N_β so as to maximize the W_{12} solution. Therefore,

$$W_{12}(j) = \max_{f=1}^{k-1} \{W'_{12}(j)\}$$

Then the optimal solution is given by,

$$T(j) = \max\{W_0(j), W_1(j), W_2(j), W_{12}(j)\}$$

We have the following theorems on the time complexity and optimality of ALGO-TRMPS.

Theorem 2 *The time complexity of ALGO-TRMPS is $O(kn^2 \times f(k, n))$, where n is the number of nets, k is the number of tracks available on over-the-cell area and $f(k, n)$ is the time to compute solution in each L-shaped region.*

Proof: At each column j the algorithm ALGO-TRMPS computes the following.

1. In the worst case at column j , ALGO-TRMPS may compute k number of $T(j)$ solutions and finds the maximum among them. The computation of each $T(j)$ solution may involve computation of two row solutions in $O(kj)$ number of L-shaped regions. Assume that time required to compute solution in each L-shaped regions is $f(k, n)$.

Hence at column j the total computation time is $O(k + kj \times f(k, n))$ and for L columns the total computation time will be

$$\sum_{j=1}^L (k + kj \times f(k, n))$$

Since $L = O(n)$, the total time is

$$\begin{aligned} & kn + k \times f(k, n) \frac{n(n+1)}{2} \\ &= kn + f(k, n) \times O(kn^2) \\ &= O(f(k, n) \times (kn^2)) \end{aligned}$$

Hence, the theorem. \square

Now, we state the corollary for the WTRMPS problem.

Theorem 3 *Given an instance \mathcal{I} , of TRMPS problem, ALGO-TRMPS produces an optimal solution.*

Proof: We prove this theorem by induction.

Basis: When $j = 1$, ALGO-TRMPS generates an optimal solution, which is null.

Induction Hypothesis: Assume that ALGO-TRMPS has generated an optimal solution, $T(j)$, for j columns.

Inductive Step: Now we have to prove that, for $j+1$ columns, the $T(j+1)$ solution, generated by this algorithm is also optimal.

If \mathcal{I} has an $j+1$ solution S with maximum number of nets, then S may be equal to $T(j)$ or greater than $T(j)$.

At column $j+1$, one can have any of the following situations.

1. There can be only one net, $N_\alpha = (t_i, t_{j+1})$ or $N_\beta = (b_i, b_{j+1})$, with a terminal at column $j+1$, such that $1 \leq i < j+1$. In this case, the optimal solution may or may not contain the net N_α or N_β . Since, ALGO-TRMPS considers both the cases, and takes the maximum of the two solutions obtained, by taking and not taking $N_\alpha(N_\beta)$, it generates an optimal $T(j+1)$ solution.

2. There are two nets $N_\alpha = (t_i, t_{j+1})$ and $N_\beta = (b_i, b_{j+1})$, with one of their terminals at column $j + 1$, such that $1 \leq i, m < j + 1$. Here, the optimal solution may consist of (a) none of the nets N_α and N_β , (b) only the net N_α , (c) only the net N_β , or (d) both the nets N_α and N_β .

Since, ALGO-TRMPS considers all the above cases exhaustively, and takes the maximum of the four solutions obtained, for all those cases, it generates an optimal $T(j + 1)$ solution \square .

In this section, we discussed all the possible cases the algorithm has to deal with, while computing the two row maximum planar subset. So far, we have assumed that, we already know the solutions in the L-shaped regions. In the following subsection we give a detailed description of computing the maximum planar subset in an L-shaped region.

Maximum Planar Subset in an L-Shaped Region

In this section, we describe the method of computing the solutions in an L-shaped region (i, j, f) , where $i < j$. The solutions in an inverted L-shaped region (where $i > j$), can also be computed in a similar manner. The $L(i, j, f)$ solutions can be classified into the following four types depending on the existence of a bottom net which is completely contained in R_j , with b_j as one of its terminals.

Case 1: There is no bottom net, which is completely contained in R_j , with b_j as one of its terminals (Figure 8(a)). In this case

$$L(i, j, f) = L(i, j - 1, f)$$

Case 2: There is a net $N_\beta = (b_m, b_j)$, which is completely contained in R_j , such that, $\text{span}(N_\beta) < (j - i)$, i.e., column m is to the right of column i , as shown in Figure 8(b). Excluding the net N_β leads to,

$$L(i, j, f) = L(i, j - 1, f)$$

Let us assume that, the $L(i, j, f)$ solution that includes the net N_β , is maximum, by assigning N_β to track f_1 , such that $f_1 \geq f$. Also notice that the optimal $L(i, j, f)$ solution cannot consist of any other nets, that lie entirely in the L-shaped region, represented by (i, j, f) , in the shaded area shown in Figure 8(b). If any such net exists, then the $L(i, j, f)$ solution, which includes the net N_β , would not be planar. Therefore the $L(i, j, f)$ solution remains maximum, even if N_β is assigned to track f_2 , such that $f < f_2 < f_1$. Therefore, we can assign N_β to track $f + 1$. Now, the $L(i, j, f)$ solution, which includes N_β , consists of

1. The nets enclosed by N_β , which is $S_b(m + 1, j - 1, k - f - 1)$.
2. The net N_β itself, and
3. The solution of the L-shaped region, represented by $L(i - 1, m - 1, f)$.

The $L(i, j, f)$ solution that includes N_β , which we denote by $L'(i, j, f)$ is given by

$$L'(i, j, f) = S_b(m + 1, j - 1, f + 2) + 1 + L(i, m - 1, f)$$

The optimal $L(i, j, f)$ solution will be, the maximum of the solutions obtained by excluding and including the net N_β . Therefore,

$$L(i, j, f) = \max\{L(i, j, f - 1), L'(i, j, f)\}$$

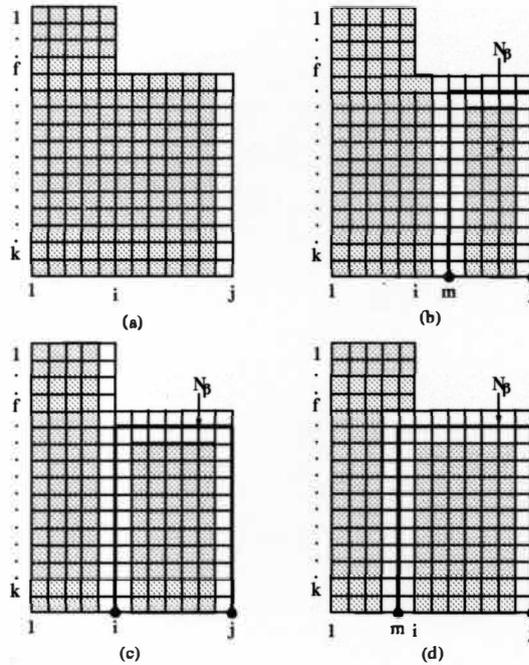


Figure 8. The Four Cases of L-shaped Solutions.

Case 3: There is a net $N_\beta = (b_m, b_j)$, which is completely contained in R_j , such that, $\text{span}(N_\beta) = (j - i)$, i.e., column m and column i are the same, as shown in Figure 8(c). This is similar to the Case 1, except that, the $L(i, j, f)$ solution, which includes the net N_β , consists of the single row solution, in the region enclosed by N_β , the net N_β , and the two row solution $T(i-1)$. Therefore, the $L(i, j, f)$ solution is given by,

$$L(i, j, f) = \max\{L(i, j-1, f), \\ S_b(i+1, j-1, f+2) + 1 + T(i-1)\}$$

Case 4: There is a net $N_\beta = (b_m, b_j)$, which is completely contained in R_j , such that, $\text{span}(N_\beta) > (j - i)$, i.e., column m is to the left of column i , as shown in

Figure 8(d). Excluding the net N_β leads to,

$$L(i, j, f) = L(i, j - 1, f)$$

Suppose we place the net N_β in track f_1 , $f < f_1 \leq k$, then The $L(i, j, f)$ solution, that includes the net N_β , in track f_1 , $f < f_1 \leq k$; denoted by $L'(i, j, f)$ is consists of

1. The nets enclosed by N_β , which is $S_b(m + 1, j - 1, k - f_1 - 1)$.
2. The net N_β itself, and
3. The solution of the L-shaped region, represented by $(i, m - 1, f)$.

Therefore the $L(i, j, f)$ solution, which includes N_β in track f_1 is given by

$$\begin{aligned} L'(i, j, f) &= S_b(m + 1, j - 1, k - f_1 - 1) + 1 \\ &+ L(i, m - 1, f) \end{aligned}$$

By varying f from $f + 1$ to k , we can find the track, on which we can place N_β so as to maximize the $L(i, j, f)$ solution. Then, the $L(i, j, f)$ solution we get, by choosing N_β , which we denote by $L''(i, j, f)$ is given by

$$L''(i, j, f) = \max_{f_1=f+1}^k \{L'(i, j, f)\}$$

The optimal $L(i, j, f)$ solution will be, the maximum of the solutions obtained by excluding and including the net N_β . Therefore

$$L(i, j, f) = \max\{L(i, j, f - 1), L''(i, j, f)\}$$

The computation of each $T(j)$ solution, involves the computation of solutions in

several L-shaped regions. Therefore, the worst case running time of the algorithm ALGO-TRMPS, depends on the the number of L-shaped regions. We have the following lemma on the number of L-shaped regions.

Lemma 2 *In canonical representation the number of L-shaped regions is $O(kn^2)$, where k is the number of tracks and n is the number of nets.*

Proof: Consider a $K \times L$ rectangular grid. The two types of L-shaped regions in Fig 5 are presented by the 3-tuples (i, j, f) and (j, i, f) respectively.

For a fixed value of i , where $1 < i < L$, the number of L-shaped regions obtained by varying j from $i + 1$ to L and f from 1 to k , which are of type (i, j, f) is $k(L - i)$, and the number of L-shaped regions of type (j, f, i) are also $k(L - i)$. Therefore the total number of the L-shaped regions, for a fixed value of i is $2K(L - i)$. As i ranges from 2 to $(L - 1)$, the total number of L shaped regions is given by

$$2 \sum_{i=2}^{L-1} K(L - i)$$

which is equal to $K(L^2 - 3L + 2) = O(kn^2)$ \square .

Lemma 3 *Each $L(i, j, f)$ solution ,where $1 \leq i, j \leq L$ and $1 \leq f \leq k$, is computed once and it takes constant time to compute the solution.*

Proof: By the time we compute $L(i, j, f)$ we have the following solutions at disposal.

$$L(i', f', j') \text{ where } 1 \leq i' \leq i, 1 \leq j' \leq j, \text{ and } 1 \leq f' < f$$

$$L(i', f', j') \text{ where } 1 \leq i' \leq i, 1 \leq j' < j, \text{ and } 1 \leq f' \leq f$$

$$L(i', f', j') \text{ where } 1 \leq i' < i, 1 \leq j' \leq j, \text{ and } 1 \leq f' \leq f$$

From the cases discussed above it can be concluded that

$$L(i, j, f) = \max\{L(i, f - 1, j), L(i, j, f') + S_b(i, j, f)\}$$

Since the values of $L(i, f - 1, j)$, $L(i, j, f)$, and $S_b(i, j, f)$ are available computation of $L(i, j, f)$ takes constant time. \square

Theorem 4 *The computation time of ALGO-LMPS is $O(kn^2)$, where k is the number of tracks in a cell row, and n is the number of nets.*

Proof: At each column j the algorithm ALGO-LMPS computes the following.

ALGO-LMPS may have to compute the solutions in $O(jk)$ number of L-shaped regions at column j . Computation of the solution for each L-shaped region takes constant time, since the solutions of the smaller L-shaped regions and smaller rectangular regions are already known.

Hence at column j the total computation time is $O(k + kj)$ and for L columns the total computation time will be

$$\sum_{j=1}^L (kj)$$

Since $L = O(n)$, the total time is

$$\begin{aligned} & k \frac{n(n+1)}{2} \\ & = O(kn^2) \end{aligned}$$

Hence, the theorem. \square

Theorem 5 *Given an Instance \mathcal{I} , ALGO-LMPS produces an optimal solution.*

Proof: Proof for this theorem is very similar to the proof presented for the Theorem 3. \square

In the next section we formally present our algorithm, that solves the TRMPS problem, optimally.

The Optimal Algorithm

Algorithm <i>ALGO-TRMPS</i> ($\mathcal{N}_T, \mathcal{N}_B, n, T, \mathcal{B}, \mathcal{N}, L$)
Input: Sets of nets, $\mathcal{N}_T, \mathcal{N}_B$, Sets of terminals, T, \mathcal{B} . Number of nets n , Number of columns L ,
Output: Optimal two planar subset \mathcal{N}_P .
<pre> Begin <i>SRMPS</i>(); for $j = 1$ to L case(<i>net_type</i>(j)): Type 1: <i>TRMPS</i>(j) = $T1$; Type 2: <i>TRMPS</i>(j) = $T2$; Type 3: case(<i>nets_at</i>(j)) type a: <i>TRMPS</i>(j) = $T3a$ type b: <i>TRMPS</i>(j) = $T3b$ type c: <i>TRMPS</i>(j) = $T3c$ End(for) for $j = L$ to 1 if($T(j-1) < T(j)$) if(<i>type</i> = 1) $\mathcal{N}_P = \mathcal{N}_P \cup N_\alpha$ else if(<i>type</i> = 2) $\mathcal{N}_P = \mathcal{N}_P \cup N_\beta$ else if(<i>type</i> = 3) $\mathcal{N}_P = \mathcal{N}_P \cup N_\alpha \cup N_\beta$ End(if) End(for) End; </pre>

Figure 9. Algorithm ALGO-TRMPS.

In this section, we present ALGO-TRMPS formally. The input to the

algorithm is an instance \mathcal{I} of the linear TRMPS problem, and the output is a set of nets $\mathcal{N}_{\mathcal{P}}$ to \mathcal{I} .

The algorithm to compute *TRMPS* is formally presented in Figure 9. Our results on the TRMPS problems are stated in the following two theorems.

From the theorems 3 and 5 we conclude the following theorem.

Theorem 6 *Given an instance \mathcal{I} , ALGO-TRMPS provides an optimal solution to the two row maximum planar subset problem.*

From the theorems 2 and 4 we conclude the following theorem.

Theorem 7 *The complexity of the ALGO-TRMPS is $O(kn^2)$, where k is the number of tracks available over-the-cell area and n is the number of nets.*

The TRMPS problem, can be easily modified to solve the weighted version of the problem (WTRMPS). For example the weighted version of the expression for Case 1(b), is as follows.

$$WT(j) = \max_{f=1}^k \{WS_t(i+1, j-1, f-1) + W_\alpha + WL(i-1, j, f+1)\}$$

In the above expression, WS_t is the maximum weighted single row solution, WL is the maximum weighted solution in the L-shaped region defined by the tuple $(i-1, j, f+1)$ and W_α is the weightage given to the net N_α .

The WTRMPS problem is important because, weights can be assigned to

each net so as to find a better planar subset of nets, such that, the nets which are selected will contribute significantly towards the density reduction in the channels.

The following corollary holds for WTRMPS problem.

Corollary 2 *The time complexity of WTRMPS problem is $O(kn^2)$ time.*

CHAPTER III

EXTENSIONS

In this chapter we will discuss some of the extensions for TRMPS problem, such as TRMPS with chosen subset of nets, pre-routed nets, TRMPS in presence of crossing nets. We also present 0.5 approximation algorithm for MTM planar routing problem.

TRMPS With Chosen Subset and Prerouted Nets

There are several important extensions of TRMPS problem which need attention. One important case is pre routing of critical nets. In many practical routing methodologies, timing critical nets are manually routed or special routers are used to pre-route the nets. The remaining nets must account for presence of these nets.

A set of nets is chosen if it is required to in the solution. Note, it should be a planar set by definition. A chosen subset of nets is called *pre-routed* if each net of the set has been assigned a track.

In this chapter, we present the optimal algorithms ALGO-CSS and ALGO-PRN that solves the *TRMPS* problem in the presence of a chosen subset of nets and in the presence of pre-routed nets, respectively.

Definition: A net $N_\alpha = (t_l, t_m)$ is said to be *contained* by a net $N_\beta = (t_i, t_j)$, if $i < l$ and $m < j$. In the other words, net N_β *contains* net N_α .

Definition: A net $N_\alpha = (t_l, t_m)$ is said to be *overlapping* with a net $N_\beta = (t_i, t_j)$, if $i < l < j < m$ or $l < i < m < j$.

Since the routing of the nets which overlap with the chosen subset of nets or pre-routed nets violates the planarity property, the nets which are overlapping with the chosen subset of nets and pre-routed nets are removed from the input of the problem.

Chosen Subset of Nets

In this section, we present the algorithm ALGO-CSN to compute the TRMPS solution in the presence of chosen subset of nets. Let $\mathcal{N}'_T \subseteq \mathcal{N}_T$ and $\mathcal{N}'_B \subseteq \mathcal{N}_B$. Let $\mathcal{N}'_{TD} = \mathcal{N}'_T \cup \mathcal{N}'_B$ be the chosen subset of \mathcal{N} , which should be included in the solution. In order to be in the solution the subset \mathcal{N}'_{TD} should be planar and routable in k tracks. The possible range of tracks that a net N_α can be assigned is stored as an additional information with each net $N_\alpha \in \mathcal{N}'_T \cup \mathcal{N}'_B$.

While computing the $T(j)$ solution, the algorithm has to deal with the following four cases.

1. There is no net $N_\alpha \in \mathcal{N}'_{TD}$, or $N_\beta \in \mathcal{N}'_{TD}$, with one of their terminals at column j . In this case, the $T(j)$ solution is computed, as explained in inprevious chapter.

2. There exists a net $N_\alpha = (t_l, t_j)$, such that, $N_\alpha \in \mathcal{N}_{TB}$ and it can be assigned to any track between tracks f_1 and f_2 . In this case N_α has to be in the $T(j)$ solution. Therefore, the $T(j)$ solution consists of the maximum planar subset in the region enclosed by N_α , the net N_α itself, and the maximum planar subset in the L-shaped region (l, f, j) , where f is the track assigned to N_α .

$$T(j) = \max_{f=f_1}^{f_2} \{S_t(l+1, j-1, f-1) + 1 + L(l, j, f)\}$$

The solution $T(j)$ is given by above equation.

3. There exists a net $N_\beta = (b_l, b_j)$, such that, $N_\beta \in \mathcal{N}_{TB}$, and it can be assigned to any track between tracks f_3 and f_4 . In this case N_β has to be included in the $T(j)$ solution. This case is very similar to the case discussed above.

4. There exists nets $N_\alpha = (t_l, t_j)$ and $N_\beta = (b_l, b_j)$, such that, $N_\alpha, N_\beta \in \mathcal{N}_{TB}$. In this case both N_α and N_β have to be included in the $T(j)$ solution.

$$T(j) = \max_{\substack{f=f_2, f'=f_3, f < f' \\ f=f_1, f'=f_3}} \{S_t(l+1, j-1, f-1) + S_b(m+1, j-1, k-f') + 2 \\ + L(l-1, m-1, f'-1)\}$$

Based on the above discussion we conclude the following theorem.

Theorem 8 *Given an instance \mathcal{I} of the TRMPS problem with the chosen subset of nets, ALGO-CSS solves the problem optimally in $O(kn^2)$ time.*

For the WTRMPS problem in the presence of chosen subset of nets the

following corollary holds.

Corollary 3 *The WTRMPS problem in the presence of chosen subset of nets can be solved optimally in $O(kn^2)$ time.*

Pre-Routed Nets

In this section, we present the optimal algorithm ALGO-PRN, to compute the TRMPS solution in the presence of pre-routed nets. Let $\mathcal{N}'_T \subseteq \mathcal{N}_T$ and $\mathcal{N}'_B \subseteq \mathcal{N}_B$. Let $\mathcal{N}'_{TB} = \mathcal{N}'_T \cup \mathcal{N}'_B$ be the prerouted subset of \mathcal{N} , which should be included in the solution. In order to be in the solution the subset \mathcal{N}'_{TB} should be planar and routable in k tracks.

If there exists a pre-routed net $N_\alpha = (t_i, t_j)$ then the subset of nets \mathcal{N}_{ur} , which are overlapping with the pre-routed nets are unroutable because routing of \mathcal{N}_{ur} violates the planarity property. The subset of nets \mathcal{N}_{cp} , which are contained by pre-routed nets are routable only in the tracks below the tracks used by the pre-routed nets. Since no other nets can use the tracks below the track of pre-routed net N_α , except the nets belong to \mathcal{N}_{cp} the maximum number of nets that can be routed under the pre-routed net N_α are $S_t(i+1, j-1, f-1)$. The same explanation applies to N_β also. Hence the optimal solution consists of $S_t(i, j, f)$.

Hence, the following lemma.

Lemma 4 *If there exists a pre-routed net N_α (or N_β) = (t_i, t_j) (or (b_l, b_m)) in the track f (g), then, the optimal solution \mathcal{S} consists of nets in $S_t(i, j, f)$ (or $S_b(i, j, f)$).*

The subset of nets \mathcal{N}_{ur} , which overlap with the set of pre-routed nets \mathcal{N}'_{TD} , are removed from the netlist. The subset of nets which are contained by the pre-routed net $N_{pn} = (t_i, t_j)$ which is in track f , and not in $S_t(i+1, j-1, f-1)$ are removed from the netlist. The subset of nets \mathcal{N}_{ps} , which are contained by N_{pn} and in $S_t(i+1, j-1, f-1)$ are added into the solution and \mathcal{N}_{ps} is removed from the netlist.

Now the netlist contains the subset of nets \mathcal{N}_{co} , where each net of \mathcal{N}_{co} contains one or more pre-routed nets. The possible range of tracks that a net $N_\alpha \in \mathcal{N}_{co}$ can be assigned is computed and stored as an additional information with each net.

There is a net $N_\alpha = (t_i, t_j) \in \mathcal{N}_{co}$, which can be assigned to any track between f_1 and f_2 , and another net $N_\beta = (b_m, b_j) \in \mathcal{N}_{co}$, which can be assigned to any track between f_3 and f_4 . Here, we have the following 4 cases.

Case 1: If both the nets are excluded then, the solution $W_1(j)$ is

$$W_1(j) = T(j-1)$$

Case 2: If the net N_α is included in the solution $W_2(j)$, and the net N_β is excluded from the solution.

$$W_2(j) = \max_{f=f_1}^{f=f_2} \{L(i-1, j-1, f+1) + S_t(i+1, j-1, f-1) + 1\}$$

Case 3: If the net N_β is included in the solution $W_3(j)$, and the net N_α is excluded from the solution.

$$W_3(j) = \max_{\substack{f=f_4 \\ f=f_3}} \{L(j-1, m-1, f-1) + S_b(m+1, j-1, f-1) + 1\}$$

Case 4: If both the nets N_α and N_β are included in the solution $W_4(j)$.

$$W_4(j) = \max_{\substack{f=f_2, f'=f_4, f < f' \\ f=f_1, f'=f_3}} \{L(i-1, m-1, f'-1) + S_t(i+1, j-1, f-1) \\ + S_b(m+1, j-1, k-f') + 2\}$$

Now the solution $T(j)$ is

$$T(j) = \max\{W_1(j), W_2(j), W_3(j), W_4(j)\}$$

Based on the above discussion we conclude the following theorem.

Theorem 9 *Given an instance \mathcal{I} of the TRMPS problem with a set of pre-routed nets, ALGO-PRN solves the problem optimally in $O(kn^2)$ time.*

For the WTRMPS problem in the presence of pre-routed nets, the following corollary holds.

Corollary 4 *The WTRMPS problem in the presence of pre-routed set of nets can be solved optimally in $O(kn^2)$ time.*

Zero Dogleg Planar Channel Routing

In this section, we extend our algorithm to zero dogleg planar channel routing. The planar routing model is also called the *river routing* model. The planar routing model is used for such tasks as routing the chip inputs and outputs to the pads on the chip boundary, or routing wires on a single layer in routing schemes where the layer assignment is determined by technological constraints, such as in power-supply routing. Planar channel routing has the best developed theory of all detailed-routing problems. Practically all optimization versions of the detailed planar routing problem that involve two terminal nets can be solved with efficient algorithms whose run time does not exceed $O(n^2)$ [7].

In channel routing, in addition to \mathcal{N}_T and \mathcal{N}_B , there exists another set of nets \mathcal{N}_C , called *crossing nets*. These nets are of the form (t_i, b_j) , such that $t_i \in \mathcal{T}$ and $b_j \in \mathcal{B}$. The *zero dogleg planar channel routing problem* is to find the maximum planar subset of nets, which can be routed in one layer without any doglegs, in the channel. This is a special case of the *general river routing problem* [2]. A restricted case of the general river routing problem, which has attracted a great deal of attention, is called, the *simple river routing problem* [2, 9, 14]. It is a planar channel routing problem for two terminal nets, such that, each net has one terminal on each terminal row i.e., each net is of the form (t_i, b_j) . The planar channel routing problem, we described is an extended version of the simple river routing problem, since there exists nets with both the terminals on

the same terminal row such as $N_\alpha = (t_i, t_j)$ and $N_\beta = (b_i, b_j)$.

Now, we describe the algorithm ALGO-PCR, to compute $T(j)$ solutions, where $1 \leq j \leq L$, in planar channel routing.

Phase I: In this phase, the single row solutions $S_u(i, j, f)$ and $S_l(i, j, f)$ are computed, ignoring the crossing nets.

Phase II: The two row solution for each column is computed in this phase. While computing the two row solution, the algorithm has to deal with the following cases.

(a) There are no crossing nets which lie entirely within the first j columns and have a terminal at column j , either in the top or bottom terminal rows. In this case, the $T(j)$ solution is computed, as explained in previous chapter.

(b) There exists a net $N_\alpha = (b_i, t_j)$ as shown in Figure 10(a). If N_α is excluded in the solution, then the $T(j)$ solution which is denoted as $T^\theta(j)$, can be computed, as explained in previous chapter. If N_α is included in the solution, then assume that it is assigned to track f . Let us denote this solution by $T'(j)$. Then, the $T'(j)$ solution consists of the single row solution in the region enclosed by N_α is $S_u(i + 1, j, f - 1)$, the net N_α itself, and the solution in the L-shaped region (j, f, i) . Therefore,

$$T'(j) = S_u(i + 1, j, f - 1) + 1 + L(i - 1, f + 1, j - 1)$$

By checking the all possible track assignments for N_α , the track on which the $T'(j)$

solution is maximum is found, such that N_α is in the solution. Let us denote this solution by $T''(j)$. Therefore,

$$T''(j) = \max_{f=1}^k \{T'(j)\}$$

then, the optimal $T(j)$ solution will be,

$$T(j) = \max\{T^\theta(j), T''(j)\}$$

the maximum of the two solutions obtained, by including and excluding the net N_α .

(c) There exists a net $N_\beta = (t_i, b_j)$ as shown in Figure 10(b). This case is very similar the case discussed above.

(d) There exists a net $N_\alpha = (t_j, b_j)$ as shown in Figure 10(c). In this case $T(j)$ will have N_β .

$$T(j) = T(j - 1) + 1$$

(e) There exists two nets $N_\alpha = (t_i, b_j)$ and $N_\beta = (t_j, b_m)$ as shown in Figure 10(d). Here, one can have the following two choices.

i. If N_α is chosen, then the $T(j)$ solution can be computed as in Case 2, discussed above. Let us denote this solution by $T_\alpha(j)$.

ii. If N_β is chosen, then the $T(j)$ solution can be computed as in Case 3,

discussed above. Let us denote this solution by $T_\beta(j)$.

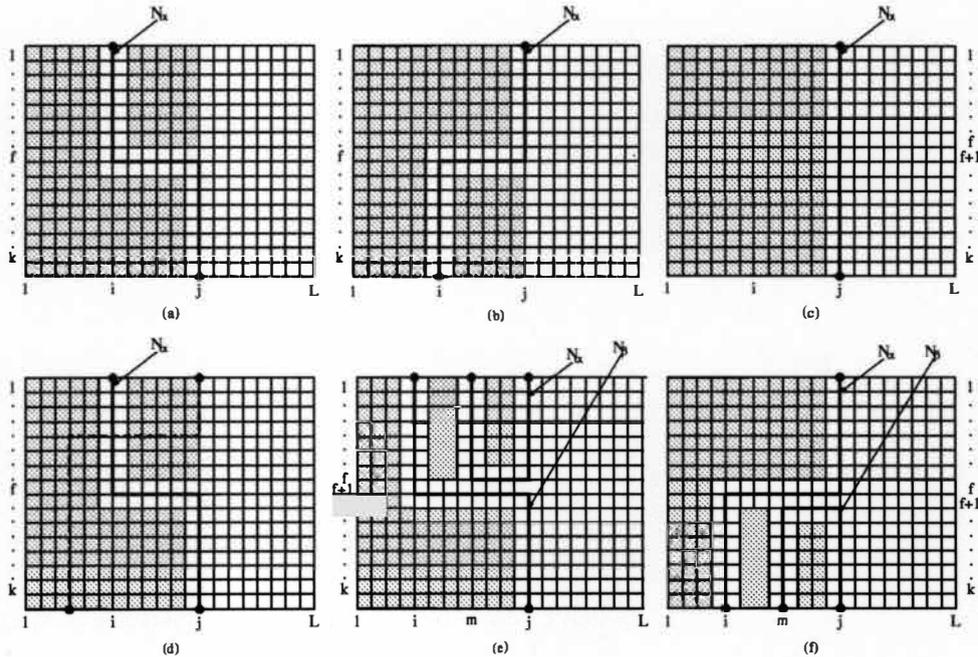


Figure 10. Cases in Crossing Nets.

Note that one cannot choose both N_α and N_β as both these nets cannot be routed on a single plane without doglegging. The optimal $T(j)$ solution will then be,

$$T(j) = \max\{T_\alpha(j), T_\beta(j)\}$$

the maximum of $T_\alpha(j)$ and $T_\beta(j)$ solutions.

(f) There exists a net $N_\alpha = (t_j, t_i)$ and a net $N_\beta = (b_j, u_l)$ as shown in Figure 10(e). Here one can have the following four cases.

i. Not taking both the nets N_α and N_β into the solution $T(j)$.

$$T(j) = T(j - 1)$$

ii. Including the top net N_α and excluding the bottom net N_β in $T(j)$ solution. This case is similar to the Case 1 described in previous chapter.

iii. Including the bottom net N_β and excluding the top net N_α in $T(j)$ solution. This case is similar to the case 2(c) described in this section.

iv. Including both the nets N_α and N_β in $T(j)$ solution.

$$T(j) = \max_{f=1}^k \{ S_u(i+1, m-1, f-1) + S_u(m+1, j-1, f-2) \\ + L(i-1, j-1, f+1) \} + 2$$

$T(j)$ solution is the maximum of the above four cases.

(g) There exists a net $N_\alpha = (t_j, l_l)$ and a net $N_\beta = (b_j, b_i)$ as shown in Figure 10(e). This case is symmetric to the above case.

In this section, we presented a detailed description on solving the planar channel routing problem. The following theorem states the complexity of ALGO-PCR for TRMPS with crossing nets.

Theorem 10 *The time complexity of ALGO-PCR, is $O(kn^2)$.*

The weighted version of the zero dogleg planar channel routing (WPCR) problem can also be solved using the algorithm presented above.

Corollary 5 *The time complexity of ALGO-PCR for WPCR problem is $O(kn^2)$.*

Approximation Algorithm for OTC Routing in MTMs

In this section, we present an improved approximation algorithm for over the cell routing in Middle Terminal Model (MTM) standard cell layouts [13]. It is a modification of the approximation algorithm presented in [13].

In MTM, the terminals are located in two rows. The top terminal row \mathcal{T} , which is located k_1 tracks below the top cell boundary, and the bottom terminal row \mathcal{B} , which is located k_3 tracks above the bottom cell boundary. There are k tracks available between \mathcal{T} and \mathcal{B} . The MTM based cells have the following three rectangular regions in M2 and M3 as described below (Figure 11):

T area: An area with k_1 tracks between the top cell boundary and the top terminal row.

C area: An area with k tracks between the top terminal row and the bottom terminal row.

B area: An area with k_3 tracks between the bottom terminal row and the bottom cell boundary.

In MTM cells, the terminals on the the same column are equi-potential. The MTM-V router which was described in [13] does not allow vias in over-the-cell area. Therefore, in MTM-V, the routing in over-the-cell areas must be planar. Furthermore, the terminals cannot be 'brought-up' to M3, as that would require a

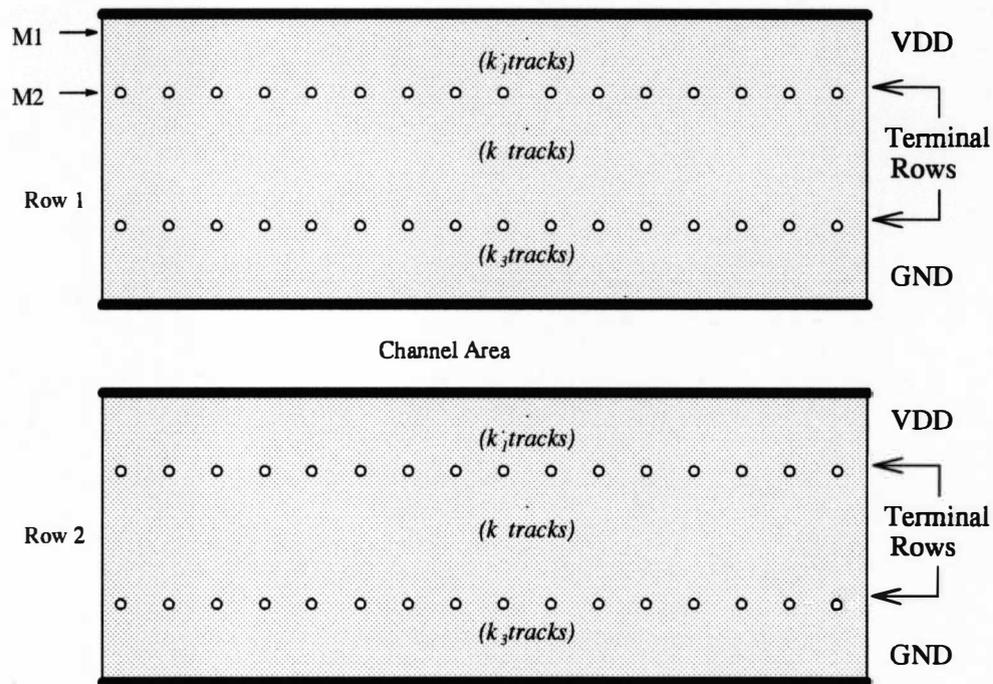


Figure 11. Middle Terminal Model.

via. Thus, the nets from all the terminals must be routed on M2, till they 'reach' the cell boundary, where vias may be used to complete the connections.

The basic steps of the MTM-V router described in [13] are given below:

1. Net classification, decomposition and weighting.
2. Maximum planar equi-potential set selection for the M2 in C area.
3. Boundary terminal assignment and M2 river routing.
4. Routing in M3 layer of OTC areas, and in channel areas.

All the nets are decomposed into two terminal nets and classified into the following two types.

TYPE I net: A net which has terminals on the same cell row.

TYPE II net: A net which has terminals on different cell rows (One terminal on \mathcal{T} and the other on \mathcal{B}).

In step 2 a set of nets is selected for routing in the M2 layer in C area. Let \mathcal{N} be the set of TYPE I nets for a given cell row. As terminals on the same column are equi-potential, four routing choices are available as described below for routing a net $N_i \in \mathcal{N}$ in the C area(Figure 12).

1. (u_i, u_j) top routing choice (t).
2. (l_i, l_j) bottom routing choice (b).
3. (u_i, l_j) right crossing routing choice (r).
4. (l_i, u_j) left crossing routing choice (l).

Given a positive integer α , a set of routing choices $\beta \subseteq \{t, b, r, l\}$, and a set \mathcal{N} of TYPE I nets, then a set $\mathcal{S} \subseteq \mathcal{N}$ is said to be an equipotential planar subset denoted by $EPS(\alpha, \beta, \mathcal{N})$, if all the nets $S_i \in \mathcal{S}$ are routable in a planar fashion, using one of the routing choices in β , using α tracks. The maximum weighted $EPS(\alpha, \beta, \mathcal{N})$ is referred to as $MEPS(\alpha, \beta, \mathcal{N})$. In short $MEPS(\alpha, \beta, \mathcal{N})$ is denoted by $S(\alpha, \beta, \mathcal{N})$.

The MES problem is finding $S(k, \{t, b, r, l\}, \mathcal{N})$ and in [13] the following result was proved.

Theorem 11 [13] *MES is NP-Hard.*

The algorithm presented in [13] solves the MES problem by transforming a planar problem into a topological problem as described below.

The terminals t_1 to t_L are represented as points p_1 through p_L on the circumference of a circle, in the clockwise direction. Similarly, the terminals b_1 through b_L are represented as points q_1 through q_L on the circumference of the circle, in the anti-clockwise direction. For each point p_i (and similarly for q_i), let $p'_i(q'_i) = p_i(q_i) + \epsilon$ and $p''_i(q''_i) = p_i(q_i) - \epsilon$ be new points on the right and the left of $p_i(q_i)$, respectively. Let C represent the set of chords of the circle. For each net $N_i = (t_i, t_j) \in \mathcal{N}$, C contains three chords $c_{ib} = (q_i, q_j)$, $c_{ir} = (p_i, q''_j)$, and $c_{il} = (q'_i, p_j)$ representing the bottom, right, and left routing choices, respectively (See Figure 13). Each chord $c_{iz} \in C$ ($z \in \{t, r, l\}$) has a weight $w(n_i)$ associated with it. Let C^* be the set of maximum weighted independent chords in C . Note that for a net N_i , C^* may contain at most one chord among c_{ib} , c_{ir} , and c_{il} as each pair of chords in $\{c_{ib}, c_{ir}, c_{il}\}$ intersect each other. S_2 is represented by $\{n_i | c_{iz} \in C^* \ \& \ z \in \{t, r, l\}\}$.

The following theorem states the performance ratio of the above algorithm.

Theorem 12 [13] *The performance ratio of ALGO-MTM is*

$$\rho = \frac{1}{2} \left(\min \left\{ 1, \frac{k}{d} \right\} \right)$$

where k is the number of tracks available in C area, and d is the optimal number of tracks required for routing the nets with ∞ number of tracks by using the (b, l, r) choices.

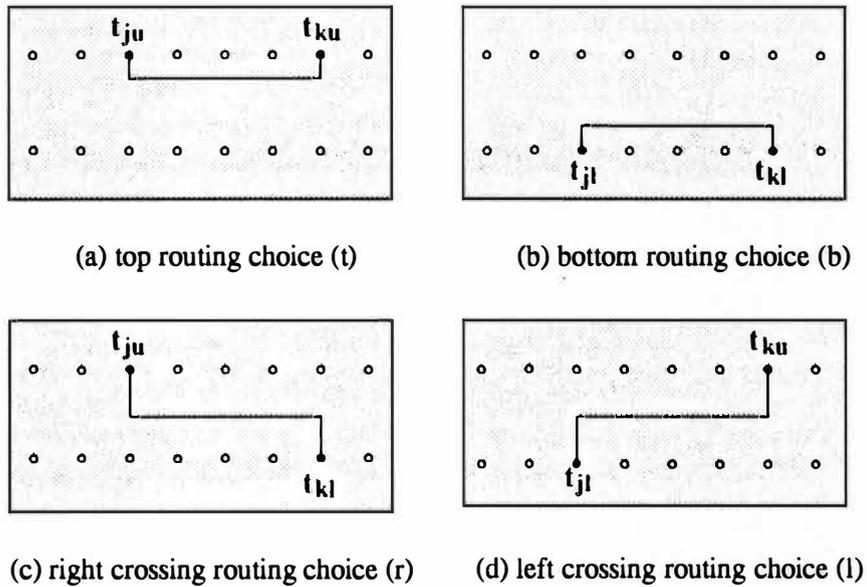


Figure 12. Four Routing Choices.

The solution obtained by the algorithm requires d number of tracks to route the nets, but k number of tracks are available in over-the-cell area. Hence, the factor $\frac{k}{d}$ is appearing in the approximation. If k value is low and d is very high then the performance the algorithm is less than 0.5, which is an unfavorable result. An instance of the MTM problem and a solution was presented in Figure 14.

In the next section we present an improved approximation algorithm ALGO-MTM that solves the MES problem and has a performance ratio of 0.5.

ALGO-MTM

In this section, we present algorithm ALGO-MTM to solve the MES problem discussed above. The solution $S = MEPS(k, \{t, l, r\}, N)$ is computed by

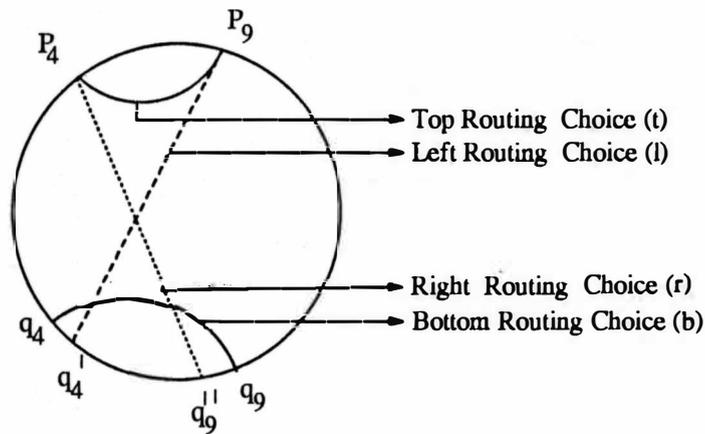


Figure 13. Four Routing Choices Representation in a Circle.

using the zero dogleg planar channel routing algorithm, described in the previous chapter. The steps in the computation of S are described below.

At each column j

1. Compute $T_l(j)$ for the left routing choice of the net.
2. Compute $T_r(j)$ for the right routing choice of the net.
3. Compute $T_t(j)$ for the top routing choice of the net using ALGO-TRMPS described in the previous chapter.
4. Maximum among $T_l(j)$, $T_r(j)$, and $T_t(j)$ is considered as $T(j)$.

$$T(j) = \max\{T_l(j), T_r(j), T_t(j)\}$$

Assume that the optimal solution, S^* consists of t^* , b^* , r^* , and l^* number of nets

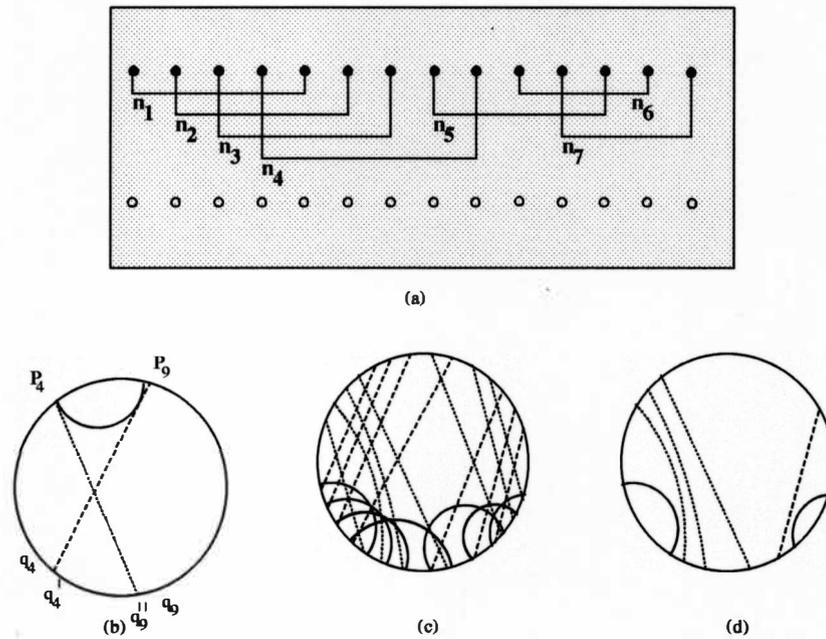


Figure 14. An Instance of MTM Problem and Solution.

of t , b , r , and l routing choices respectively.

$$S^* = t^* + b^* + r^* + l^*$$

but

$$S \geq t^* + r^* + l^*$$

In the worst case in the optimal solution $l^* = 0$, and $r^* = 0$. Hence the performance ratio, ρ , is

$$\rho = \frac{S}{S^*} = \frac{t^*}{t^* + b^*}$$

Since $l^* = 0$ and $r^* = 0$, in the solution S^*

$$t^* = b^*$$

Hence, the performance ratio

$$\rho = \frac{1}{2}$$

Therefore, the following theorem is concluded.

Theorem 13 *The performance ratio of ALGO-MTM is*

$$\rho = \frac{1}{2}$$

ALGO-MTM shows a better performance compared to the previous known result by eliminating the $\frac{k}{d}$ factor. This is achieved by directly routing the nets rather than obtaining a solution from a topological solution.

CHAPTER IV

CONCLUSIONS

In this thesis, we presented an algorithm for solving the TRMPS Problem optimally. Our algorithm runs in $O(kn^2)$ time, where k is the number of tracks in over-the-cell area of a cell row, and n is the number of nets. This algorithm can be effectively utilized for over the cell routing in standard cell layouts. Our algorithm can also be extended, to solve the TRMPS problem in the presence of pre-routed and a chosen subset of nets, as well as for zero dogleg planar channel routing *i.e.*, in the presence of crossing nets. By using our approach, we have improved the performance ratio of the existing best known approximation algorithm, for over the cell routing in MTM standard cell layouts, to 0.5.

In this thesis, we solved the TRMPS problem when the nets are not allowed to bend. However, by allowing bends more nets may be routed in over-the-cell area. A variant of the TRMPS problem, in which b or less number of bends are allowed for each net, is still an open problem.

BIBLIOGRAPHY

- [1] M. Holmes, N. A. Sherwani, and Sarrafzadeh M. New algorithm for over-the-cell channel routing using vacant terminals. In *IEEE Transactions in CAD*, pages 126–131. IEEE, June 1993.
- [2] C.P. Hsu. General river routing algorithm. In *Proceedings of 20th Design Automation Conference*, pages 578–583, 1983.
- [3] Cong J., Preas B., and Liu C. L. General models and algorithms for over-the-cell routing in standard cell design. In *Proc. 27th ACM/IEEE Design Automation Conference*, volume IEEE-90, pages 709–715. IEEE, 1990.
- [4] Cong J. and Liu C. L. Over-the-cell channel routing. In *Proc. International Conference on Computer-Aided Design*, volume ICCAD-88, pages 80–83. IC-CAD, 1988.
- [5] Cong J., Hossain M., and Sherwani N.A. A provably good multilayer topological planar routing algorithm in ic layout design. *IEEE Trans. CAD.*, 12(1):70–78, January 1992.
- [6] Supowit K. J. Finding a maximum planar subset of a set of nets in a channel. *IEEE Transaction on CAD*, CAD-6(1):93–94, January 1987.
- [7] T. Lengauer. *Combinatorial Algorithms for Integrated Circuit Layout*. A Wiley-Teubner Series, 1991.
- [8] D. S. Johnson M. R. Garey and L. Stockmeyer. *Some Simplified NP-Complete Graph Problems*. W.H. FREEMAN AND COMPANY, 1976.
- [9] M. Marek-Sadowska and T.T. Trang. Single-layer routing for vlsi: Analysis and algorithms. *IEEE Transactions on Computer Aided Design*, pages 246–259, 1983.
- [10] Garey M.R. and Johnson D.S. *Computers and Intractability*. W.H. FREEMAN AND COMPANY, 1979. A Guide to the Theory of NP-Completeness.
- [11] S. Bhingarde N. Sherwani, A. Panyam. *Routing in Third Dimension-From VLSI chips to MCMs*. Manuscript, 1994.
- [12] Sherwani N.A. *Algorithms for VLSI Physical Design Automation*. Kluwer Academic Publishers, 1993.

- [13] N. Sherwani S. Bhingarde, A. Panyam. Middle terminal cell models for efficient over-the-cell routing in high performance circuits. *IEEE Transactions on Very Large Scale Integration Systems*, pages 462–472, 1993.
- [14] T. Tuan and S.L. Hakimi. River routing with a small number of jogs. *SIAM Journal of Discrete Mathematics*, 1990.