Improving Over-The-Cell Channel Routing In Standard Cell Design

Xiaolin Liu[1] and Ioannis G. Tollis[1]

Abstract: *The first stage of over-the-cell routing in the horizontally connected vertically connected (HCVC) model is formulated as follows: Given two rows of terminals, find a planar routing to connect a subset of nets (with weights) on each row of terminals using a fixed number of tracks to maximize the total weight. This problem is called the two row fixed height planar routing (TFPR) problem [CPL93]. The complexity of the TFPR problem was unknown up to now. An approximation algorithm for the TFPR problem was presented in [CPL93]. In this paper we present a* $O(n^2 * h^2)$ *time algorithm to solve the TFPR problem optimally, where n is the number of terminals and h is the height of the standard cells. Our algorithm can be used to improve the performance of several over-the-cell channel routers including the ones in [CPL93] and [HSS93].*

1. Introduction

Over-the-cell channel routing for standard cell design has been studied extensively recently [CPL93, CL90, HSS93, LPHL91, SS87]. In the standard cell design, cells are placed in rows and channels are formed between adjacent cell rows. There are three routing layers: one layer P of polysilicon and two layers M1 and M2 of metal. It has been observed that intra-cell routing can be completed using one layer of polysilicon and one layer of metal. Therefore, it is possible to use the other metal layer over the cells for inter-cell routing in order to reduce the channel routing area. There are two physical models [CPL93]:

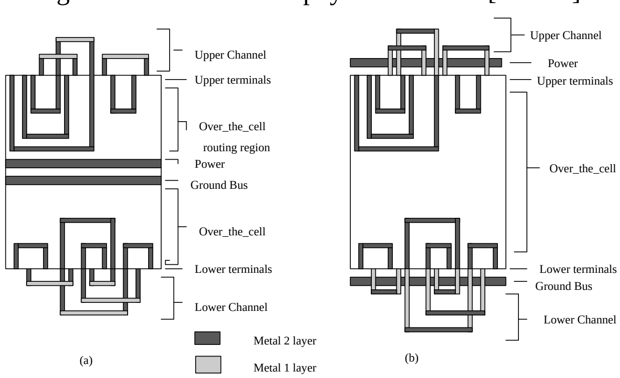

Figure 1: (a) HCVD model; (b) HCVC model.

* *The Horizontally connected vertically divided model (HCVD)*, Fig. 1(a): In this model, power/ground busses are routed on layer M2 in the middle of the cell row. Over-the-cell connections are also routed on layer M2. Clearly, the over-the-cell region is divided into two parts vertically by the power/ground busses in the middle of the cell.

* *The horizontally connected vertically connected model (HCVC)*, Fig.1(b): In this model, cell terminals and feedthroughs are both on layer M1. In the channel, horizontal wires are routed on layer M2 and vertical wires are routed on layer M1. The power bus is on layer M2 in the upper channel just above the upper terminals, and the ground bus is on layer M2 in the lower channel just below the lower terminals. Over-the-cell routing is carried out on layer M2. Clearly, layer M2 is available for over-the-cell routing, over the total area of the cells.

The first stage of over-the-cell channel routing in the horizontally connected vertically connected (HCVC) model can be formulated as follows: Given two rows of terminals, find a planar routing to connect a subset of nets (with weights) on each row of terminals using a fixed number of tracks to maximize the total weight. This problem is called the *two row fixed height planar routing (TFPR)* problem [CPL93]. There were no polynomial time algorithms to solve the TFPR problem optimally before. Hence, its complexity was unknown and an approximation algorithm for the TFPR problem was presented in [CPL93]. In this paper we present a dynamic programming algorithm to solve the TFPR problem optimally in $O(n^2 * h^2)$ time, where n is the number of terminals and h is the height of the standard cells. The time complexity is reduced to $O(n * h)$ if our algorithm is implemented in a SIMD parallel machine with $O(n * h)$ processors. The weight can be any kind of measurement for reducing channel width. Therefore our algorithm may be used to improve the performance of several over-the-cell channel routers including the ones in [CPL93] and [HSS93]. Due to space limitations we have removed the proofs, which can be found in [LT94]. We recently discovered that a similar result to ours is described in [DSMP94].

2. Over-The-Cell Routing

Since the over-the-cell channel routing problem is NP-hard [GN87], a common approach is the following [CPL93]:

Stage 1: Route over the cells (in one layer). Fig. 2(a) shows a routing solution for one side of the channel after the first step. The routing is valid if any two different nets do not cross or touch each other. The weight of the routing solution is equal to the sum of the weights of the nets which are connected in the over-the-cell routing area. The objective of stage 1 is to maximize the weight of the planar

[1] Dept. of Computer Science, The Univ. of Texas at Dallas, Richardson, TX 75083–0688 xliu@utdallas.edu, tollis@utdallas.edu

routing solution. For example, if each net has weight 1, then the routing solution in Fig. 2(a) is a valid solution with weight 3. The weight indicates the possibility to reduce the channel width. In [CL90] the number of nets routed over-the-cell is chosen as the weight. This means that we route as many nets as possible over-the-cell, which may not lead to a good solution. In [CPL93] the weight of a pair of terminals measures the degree of congestion in the channel between these two terminals.

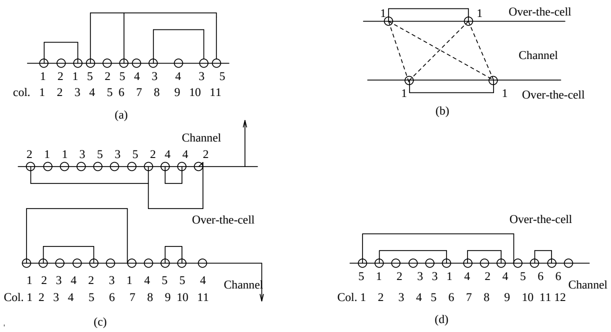

Figure 2: (a) Planar routing over the cell; (b) possible net segment connecting two hyperterminals;(c) A valid solution for TFPR (height is 3); (d) a valid solution to OFPR(height is 2).

Stage 2: Choose which net segments will be connected in the channel such that the resulting channel density is minimized. It was shown that the general net segment selection problem is NP-hard, and an efficient heuristic algorithm was presented in [CL90]. This algorithm is used in almost all over-the-cell channel routers [CL90, CPL93, HSS93].

Stage 3: Connect the terminals corresponding to the selected net segments using a conventional two layer channel router, such as [RF82].

In HCVC model, for each row of cells *R*, the entire M2 layer over cell *R* can be used for over-the-cell routing. Moreover, both the lower terminals of *R* and the upper terminals of *R* share the same over-the-cell routing region. Therefore, over-the-cell routing for both the lower and upper terminals of *R* has to be carried out simultaneously in order to use the common routing region efficiently. Furthermore, the number of tracks in the over-the-cell routing region is only limited by the height of cells. Thus the over-the-cell routing problem (stage 1) in the HCVC model can be formulated as follows:

Given two rows of terminals, we want to find a planar routing *S* to connect a subset of nets on each row of terminals, using a fixed number of tracks in order to maximize the weight of *S*. This problem is called the *two row fixed height planar routing (TFPR)* problem. Fig. 2(c) shows a valid solution to a TFPR problem with weight 40 if nets 2, 4, 1, and 5 have weight 10 and any other net has weight 1. It is easy to show that this solution is optimal since it has maximum weight. The important observation is that net 2 from top and net 1 from bottom share track 2. In Section 3 we will present a dynamic programming algorithm to find a valid routing solution with maximum weight for any TFPR instance.

In the HCVD model, for each row of cells of *R*, the over-the-cell routing region is divided vertically into two sub-regions by the power/ground busses. Therefore, over-the-cell routing for the terminals on the upper edge of the cells of *R* and the terminals on the lower edge of the cells of *R* is carried out independently in the upper and the lower sub-regions, respectively. Moveover, the height of each sub-region is limited to half the height of the cell. Thus, the over-the-cell routing problem in the HCVD model is to find a planar routing *S* to connect a subset of the nets on a row of terminals, using a fixed number of tracks on one side of the terminals such that the weight of *S* is maximum. We call this problem the *one row fixed height planar routing (OFPR) problem*. Fig. 2(d) shows a valid routing solution to an OFPR problem. In [CL90], the multi-terminal net routing problem was first transformed into a two-terminal net routing problem.

Theorem 2.1: For any instance *I* of the multi-terminal net OFPR problem, *I* can be transformed to an instance *I'* of the two terminal net OFPR problem in $O(n^2)$ time such that *I'* contains $O(k*n)$ terminals, where *n* is the number of terminals in *I* and *k* is the maximum size of a net in *I'*.

Based on a dynamic programming approach presented in [CPL93], the authors obtained the following result:

Theorem 2.2: The two terminal net OFPR problem can be solved in $O(t*n^2)$ time, where *n* is the number of terminals and *t* is the number of available tracks.

3. Solving the TFPR problem optimally

In over-the-cell routing (in the HCVC model), the multi-terminal net routing problem is first transformed into a two terminal net routing problem. By a result similar to Theorem 2.1, we need only consider two terminal net TFPR problems. In this section we will present a dynamic programming approach to the two terminal net TFPR problem.

For an instance *T* of the two terminal net TFPR problem, let *n* be the maximum number of terminals on the lower/upper edges of cells in *T* and *h* be the number of tracks used for over-the-cell routing, i.e., the height of the standard cells. Let *T(j)* be the instance of the two terminal net TFPR problem that results by restricting *T* to the interval [1, *j*]. A valid routing solution for *T(j)* *(j=11)* is shown in Fig. 2(c). Let *M(j)* denote the maximum weight of any routing solution for *T(j)*. Clearly, our goal is to find a maximum weighted routing solution for the instance *T(n)*, i.e., a routing solution that achieves *M(n)*. Let *T1(j,* *k, m)* ($1 \le k \le j$; $0 \le m \le h$) denote the instance of *T(j)* with a hole in its right lower corner. The routing solution for *T1(j, k, m)* cannot go inside this hole which is located between column *k* and column *j* and between track *m* and track *h*. A valid routing solution for *T1(j, k, m)* is shown in Fig. 3(a). Let *T2(j, k, m)* denote the instance of *T(j)* with a hole in its right upper corner. The routing solution for *T2(j, k, m)* cannot go inside this hole that is located between column *k* and column *j* and between track 1 and track *m*. An example for *T2(j, k, m)* is shown in Fig. 3(b). It is clear that if there is no hole in the routing region of *T1(j, k, m)* (or *T2(j, k, m)*), then *T1(j, k, m)* (or *T2(j, k, m)*) becomes *T(j)*. Let *M1(j, k, m)* and *M2(j, k, m)* denote the maximum weight of any routing solution for *T1(j, k, m)* and *T2(j, k, m)*, respectively.

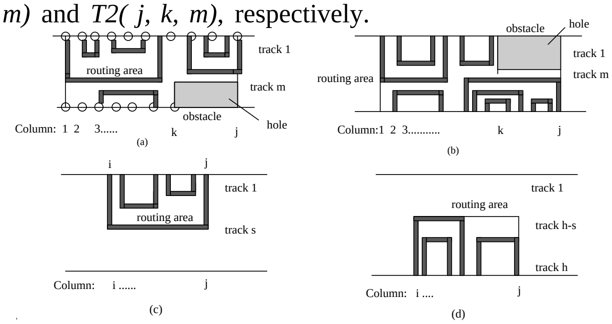

Figure 3: (a) A valid routing solution for T1(j,k,m); (b) a valid solution for T2(j,k,m); (c) A valid solution for I1(i,j,s); (d) a valid solution for I2(i,j,s).

Let *I1(i, j, s)* denote the instance of the two terminal net OFPR problem resulting from restricting nets on the upper edge of the cells inside interval *[i, j]*, and allowing tracks 1 to *s* for routing. An example is shown in Fig. 3(c). Let *I2(i, j, s)* be the instance of the two terminal net OFPR problem resulting from restricting nets on the lower edge of the cells inside interval *[i, j]*, and allowing track *(h-s) to h* for routing. A valid solution for *I2(i, j, s)* is shown in Fig. 3(d). Let *N1(i, j, s)* and *N2(i, j, s)* be the maximum weight of any routing solution for *I1(i, j, s)* and *I2(i, j, s)*, respectively. In the following, we will show how to compute *M1(j, k, m)* ($1 \le k < j$; $0 \le m \le h$).

Assume that the net at column *j* on the upper edge of the cell is net *a* and that the other terminal of net *a* is at column *j'* of the upper edge of the cell. There are three cases according to position of *j'*. We consider all cases for computing *M1(j, k, m)*.

Case 1: If *j'* is not in the interval [1, *j*] (Fig. 4(a) shows this case), we cannot route net *a* in any of the solutions for *T1(j, k, m)*. Thus, a routing solution for *T1(j, k, m)* is also a routing solution for *T1(j-1, k, m)*. Therefore,
M1(j, k, m) = M1(j-1, k, m), if j' is not in [1, j]. (Eq. 1)

If *j'* is in the interval [1, *j*], there are other two cases according to the relationship between *j'* and *k*.

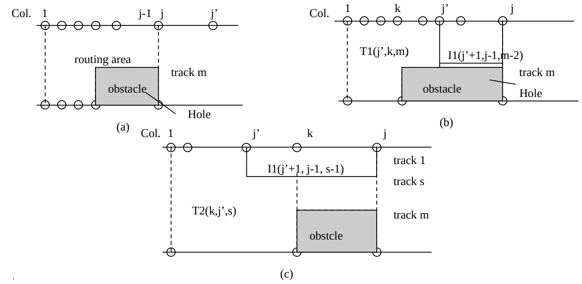

Figure 4: (a) An instance with j' out of routing region; (b) An instance with k<=j'<j; (c) An instance with 1<=j'<k.

Case 2: $k \le j' < j$ (Fig. 4(b)). A maximum weighted routing solution *S* for *T1(j, k, m)* may or may not route net *a*.
Subcase 2.1: If net *a* is not routed in *S*, clearly, we still have *M1(j, k, m) = M1(j-1, k, m)*.
Subcase 2.2: If net *a* is routed in *S* (without loss of generality, we assume that net *a* is routed in *S* using track *m-1*; otherwise, we waste part of track *m-1*), then net *a* partitions *T1(j, k, m)* into *T1(j', k, m)* and *I1(j'+1, j-1, m-2)*. Since the over-the-cell routing is planar, the routing solution for *T1(j', k, m) (I1(j'+1, j-1, m-2))*should not go inside routing region of *I1(j'+1, j-1, m-2) (T1(j', k, m))*. Therefore, we have *M1(j, k, m) = M1(j', k, m) + N1(j'+1, j-1, m-2) + w(a.j, a.j')*, where *w(a.j, a.j')* is weight of net *a*.

Thus, for case 2 we have the following equation:
M1(j, k, m) = max{M1(j-1, k, m), M1(j', k, m) + N1(j'+1, j-1, m-2) + w(a.j, a.j')}, if $1 \le k \le j' < j$. (Eq. 2)

Case 3: $1 \le j' < k$ (Fig. 4(c)). A maximum weighted routing solution *S* for *T1(j, k, m)* may or may not route net *a*.
Subcase 3.1: If net *a* is not routed in *S*, clearly, we still have *M1(j, k, m) = M1(j-1, k, m)*.
Subcase 3.2: If net *a* is routed in *S* using track *s* ($0 \le s \le m-1$), then net *a* partitions *T1(j, k, m)* into two sub-problems *I1(j'+1, j-1, s-1)* and *T2(k, j', s)*. Hence, we have: *M1(j, k, m)=M2(k, j', s) + N1(j'+1, j-1, s-1) + w(a.j, a.j')*

Therefore, for case 3 we have the following equation:
M1(j, k, m)=Max{M1(j–1, k, m), $Max_{0 \le s \le m-1}$*{M2(k, j', s) + N1(j'+1, j-1, s-1) + w(a.j, a.j')} }, if* $1 \le j' < k$. (Eq. 3)

We compute *M2(j, k, m)* in a similar fashion to the computation of *M1(j,k,m)* using the following three equations:
M2(j, k, m) = M2(j-1, k, m) if j' is not in [1, j] (Eq. 4)
M2(j, k, m) = max{M2(j-1, k, m), M2(j', k, m) + N2(j'+1, j-1, h-m+1) + w(a.j, a.j')}, if $1 \le k \le j' < j$. (Eq. 5)
M2(j, k, m) = Max{ $Max_{m \le s \le h}$*{M1(k, j', s–1) + N2(j'+1, j-1, h-s+1) + w(a.j, a.j')}, M2(j-1, k, m)}, if* $1 \le j' < k$. (Eq. 6)

Finally we discuss how to compute *M(j) (j=1, 2, ... n)*. Assume that the net at column *j* on the upper edge of the cells is net *a* and that the other terminal of net *a* is at column *j'* of the upper edge of the cells. Assume that the net at column *j* on the lower edge of the cells is net *b* and that the other terminal of net *b* is at column *j"* of the lower edge of the cells. Let *S* be the routing solution of maximum weight for *T(j)*. There are four cases according to whether *j'* or *j"* is out of the interval [*i*, *j*] as follows.

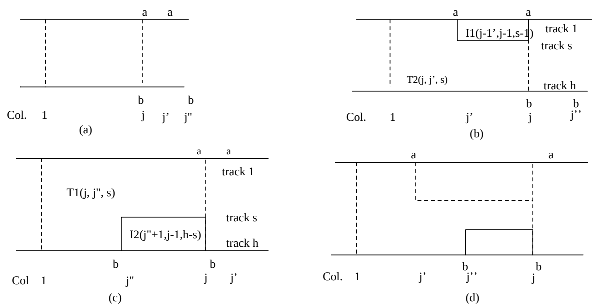

Figure 5: (a) an instance of T(j) with j'>j, j">j; (b) An instance of T(j) with j'<=j but j">j; (c) An instance of T(j) with j"<=j but j'>j; (d) An instance of T(j) with j', j"<=j.

Case I: If neither *j'* nor *j"* is in the interval [1, *j*] (shown in Fig.5(a)), then neither net *a* nor net *b* can be routed in any routing solution for *T(j)*. This means that any routing solution for *T(j)* is also a routing solution for *T(j-1)*. So we have the following equation:

M(j) = M(j-1), if neither j' nor j" is in the interval [1, j]. (Eq.7)

Case II: If *j'* is in the interval [1, *j*] but *j"* is not (shown in Fig. 5(b)), then net *b* should not be routed in *S* and net *a* may or may not be routed in *S*.

Subcase II.1: If net *a* is not routed in *S* , then we have *M(j) = M(j-1)*.

Subcase II.2: If net *a* is routed in *S* using track *s* $(1 \le s \le h)$, then it partitions *T(j)* into two routing subregions *T2(j, j', s)* and *I1(j'+1, j-1, s-1)*. Furthermore, *M(j) = M2(j, j', s) + I1(j'+1, j-1, s-1) + W(a.j, a.j')*. So in case II, we have:

$M(j) = \max\{M(j-1), \max_{1 \le s \le h}\{M2(j, j', s) + N1(j'+1, j-1, s-1) + W(a.j, a.j')\}\}$, *if j' is in [1,j] but j" is not.* (Eq. 8)

Case III: If *j"* is in the interval [1,*j*] but *j'* is not (shown in Fig.5(c)), then we have the following equation:

$M(j) = \max\{M(j-1), \max_{1 \le s \le h}\{M1(j, j'', s) + N2(j''+1, j-1, h-s-1) + W(b.j, b.j'')\}\}$, *if j" is in [1,j] but j' is not.* (Eq. 9)

Case IV: If both *j'* and *j"* are in the interval [1, *j*] (shown in Fig. 5(d)), then net *a* (or net *b*) may or may not be routed in *S*.

Subcase IV.1: If neither net *a* nor net *b* is routed in *S*, then we still have *M(j) = M(j-1)*.

Subcase IV.2: If net *a* is routed in *S* using track *s* $(1 \le s \le h)$, then it partitions *T(j)* into two routing subregions *T2(j, j', s)* and *I1(j'+1, j-1, s-1)*. This implies that *M(j) = M2(j, j', s) + N1(j'+1, j-1, s-1) + W(a.j, a.j')*.

Subcase IV.3: If net *b* is routed in *S* using track *t* $(1 \le t \le h)$, then *M(j) =M1(j, j", t) + N2(j"+1, j-1, h-t-1) + W(b.j, b.j")*.

So for case IV, we have the following equation:

$M(j) = \max\{M(j-1), \max_{1 \le s \le h}\{M2(j, j', s) + N1(j'+1, j-1, s-1) + W(a.j, a.j')\}, \max_{1 \le t \le h}\{M1(j, j'', t) + N2(j''+1, j-1, h-t-1) + W(b.j, b.j'')\}\}$, *if both j' and j" are in [1, j].* (Eq. 10)

If *n* is the number of terminals and *h* is the number of available tracks in a two terminal net TFPR problem, the maximum weight of solutions is *M(n)*. According to equations 1 to 10, *M(n)* can be computed using the dynamic programming algorithm described below:

Algorithm TFPRS

Begin

step 1: compute $\{N1(i, j, s) \mid 1 \le i \le j \le n, 1 \le s \le h\}, \{N2(i, j, s) \mid 1 \le i \le j \le n, 1 \le s \le h\}$ *using the method of [CLP93];*

step 2: ***for*** *j:=1* ***to*** *n* ***do begin*** *Phase j*

step 2.1: compute $\{M1(j, k, m), M2(j, k, m) \mid 1 \le k < j, 1 \le m \le h\}$ *using equations 1 to 6;*

step 2.2: compute M(j) using equations 7 to 10; ***end*** *Phase j;*

return(M(n));

End *of Algorithm TFPRS*

By Section 2 we know that step 1 takes $O(n^2*h)$ time. Since the computation of equation 3 or equation 6 takes O(*h*) time, step 2.1 takes $O(n*h^2)$ time in one iteration. It is clear that step 2.2 takes O(*h*) time in one iteration. Thus the time taken by step 2.1 dominates the time taken by step 2. Since step 2.1 is executed *n* times, step 2 takes $O(n^2 * h^2)$ time. Moreover, by keeping proper information at each step, not only can we compute the value of *M(n)*, but also we can construct the solution which achieves *M(n)* at the end of our algorithm.

Theorem 3.1 Algorithm TFPRS finds the maximum weighted routing solution for *T(n)* in $O(n^2 * h^2)$ time.

The time complexity of our algorithm can be reduced to $O(n*h)$ if it is implemented on a parallel computer.

Theorem 3.2 The two terminal net TFPR problem can be solved optimally in $O(n*h)$ time in a SIMD parallel computer with $O(n*h)$ processors.

Acknowledgement

We wish to thank Cheng-Hsi Chen, Achilleas Papakostas, and Chunliang Xia for their helpful comments.

References

[CL90] J. Cong and C.L.Liu, "Over-the-cell Channel Routing", *IEEE Trans. on CAD*, vol.9, no.4, pp.408–418, April 1990.

[CPL93] J.Cong, B. Preas, and C.L. Liu, "Physical Models and Efficient Algorithms for Over-the-cell Routing in Standard Cell Design", *IEEE Trans. on CAD*, vol. 12, no.5, pp. 723–734, 1993.

[DSMP94] S. Danda, N. Sherwani, S. Madhwapathy, and A. Panyam "An Optimal Algorithm for the Two Row Maximum Planar Subset Problem," Manuscript 1994.

[GN87] G. Gudmundsson, and S. Ntafos, "Channel Routing with Superterminals", *Proc. 25th Allerton Conf.*, pp. 375–376, 1987.

[HSS93] N.D Holmes, N. Sherwani, and M. Sarrafzadeh, "Utilization of Vacant Terminals for Improved over-the-cell channel routing", *IEEE Trans. on CAD*, vol.12, no.6, pp.780–792, 1993.

[LT94] X. Liu and I.G. Tollis, "Improving Over-The-Cell Channel Routing in Standard Cell Design," Tech. Rep. UTDCS-9–94, Jan. 1994.

[LPHL91] M. Lin, H. Perng, C.Hwang, and Y.Lin, "Channel Density Reduction by Routing Over The Cells", *Proc. 28th DAC*, pp. 120–125, 1991.

[RF82] R.L. Rivest and C. M. Fiduccia, " A 'Greedy' Channel Router ", *Proc. 19th DAC.*, pp. 418–424, 1982.

[SS87] Y. Shiraishi and Y. Sakemi, "A Permeation Router", *IEEE Trans. on CAD*, vol, CAD-6, pp. 462–471, May 1987.