A New Congestion-Driven Placement Algorithm Based on Cell Inflation

Wenting Hou, Hong Yu, Xianlong Hong, Yici Cai, Weimin Wu
Dept. Of Computer Science and Technology
Tsinghua University
Beijing 100084, China
email: {houwt,yuhong,hong,caiy,wuwm}@tiger.cs.tsinghua.edu.cn

Jun Gu
Department Of Computer Science and Technology,
Hong Kong University,
Hong Kong

William H. Kao
Arcadia Design System Inc.
San Jose CA 95134
U.S.A.

Abstract- In this paper, we describe a new congestion-driven placement based on cell inflation. In our approach, we have used the method of probability-estimation to evaluate the routing of nets. We also take use of the strategy of cell inflation to eliminate the routing congestion. Further reduction in congestion is obtained by the scheme of cell moving. We have tested our algorithm on a set of sample circuits from American industry and the results obtained have shown great improvement of routability.

I. INTRODUCTION

The problem of introducing performance constraints in automatic layout of integrated circuits has been addressed on several occasions in the past. With the advent of over-the-cell routing, the goal of every placement and route methodology has been to utilize area to prevent spilling of routes into channels. It is this overflow of routes that accounts for an increase in area. The multiple routing layers have enough routing resources to route most wires as long as there are not too many wires congested in the same region. Yet net list synthesis tools tend to generate net lists with high pin-count nets and poor porosity library cells, which often result in local routing congestion [5].

Congestion-driven placement based on multi-partitioning was proposed in [1]. It uses the actual congestion cost calculated from pre-computed Steiner trees to minimize the congestion of the chip, however, the number of partitions is limited due to the excessive computational load [2]. The use of minimal wire length as a metric to guide placement has been successful in achieving good placement. However, it only indirectly models congestion and the behavior of the router. Reducing the global wire length helps reduce the wiring demand globally, but does not prevent existing local congested spots. Therefore, traditional placement schemes that are based solely on wire length minimization cannot adequately account for congestion.

Recently, there appear some congestion-driven placement methods. P.N. Parakh proposed a method which takes use of the strategy of “region expanding” to eliminate the routing congestion [3]. However, this method is very easy to lead to congestion oscillation in two or more regions. M. G. Wang has presented another method in [4]. In this approach, a flow-based cell-centric algorithm is taken to move some cells to minimize the congestion. Nevertheless, this method depends on router too heavily.

In this paper, a new congestion-driven placement algorithm based on cell inflation is proposed. Our approach keeps on checking and eliminating the congestion while doing the placement. In order to check the routing congestion, we use the method of “resource estimation” so that the result of congestion will be more common and generic, not router-dependent. To alleviate and erase congestion, we make use of the strategy of “cell inflation”. All these works are done self-adaptively in our approach. By this means, our approach can not only avoid “congestion oscillation”, but also it will be of small loss of total wire length. Finally, we use a method of “cell moving” to refine our solution further.

II. CONGESTION ANALYSIS

2.1 Bin Structure

First, we will give some denotations.

Global bin structure: We partition a given chip into a set of rectilinear regions, each of these regions is called a bin, which is the same way as [2]. The congestion is defined based on the bin structure. For a certain bin, when its routing demand exceeds its routing resource supply, this bin is called congested-bin. The number of bin must be set properly, too small will be of poor accuracy, while too large will lead to high computing complexity

\[ D_{ij} \text{ (demand): In bin (i,j), } D_{ij} \text{ is the number of routes which are across or inside the bin. Since, there are horizontal and vertical routes in each bin, there are two } D_{ij} \text{ in each direction.} \]

\[ S_{ij} \text{ (supply): In bin (i,j), } S_{ij} \text{ is the number of tracks which are supplied by the route layer. There are also two } S_{ij} \text{ in vertical and horizontal direction. Existent wiring of power and clocking nets, standard cells are considered to be obstacles to routing.} \]

2.2 Wiring Modeling

We use a method of routing-estimation to estimate the
routing congestion.

Our method of routing estimation can be applied in standard cell layout with two or more routing layers. From the current cell positions and bin structure, the estimation gives out the routing estimation of each net across the chip plan in horizontal and vertical direction and the information about congested bin.

Since routing estimation is done during the placement, we can not know the real shape of each net. To build up the estimation model, we suppose to follow such requirements:
1) The net will be no detours, and the net should be routed within its bounding-box.
2) The estimation should be appropriate.
3) The estimation should make the routing estimation without very complicated computing.

In practice, we adopt "star-model" to estimate the routing of each net. To do this, we may first find the geometry center of the net. In order to optimize the routing and reduce the number of vias, we will adjust the center to the nearest coordinate of one of the nodes which the net connects to, then connect each node of this net with the center. As Fig. 1 shows, here is a five-pin net. First the net is transformed into a star-mode net just like on the right. Then we compute and adjust the coordinates of the net center.

As shown in Fig. 2, when we estimate the route that connects to Sink2 and Center, there are two possible routing path: A path and B path. We can not know which path the following router will choose after placement, so we assume that the probabilities of the following router may choose path A or path B are equal. As a result, we consider that this route will go along path A and path B 0.5 track each. According to our method, the result of routing estimation shows in Fig. 3.

We must notice that in some segments, the estimated number of required tracks is more than 1.0, which means that there are some duplicated nets in these segments. This is not consistent with real routing, so we modify that for a single net. Next, we should record the number of tracks required in the bins where the net has passed.

In fact, “star-model” is very likely to the “minimum steiner-tree” model, and the net center is rather an excellent approximation of the steiner point. According to our experiments, “star-model” is very close to the real routing in practice.

2.3 Routing Resource Supply

The information about routing resource supply includes the width of each metal layer, pitch, positions of pins in cells, obstacles and so on.

An existing wire decreases routing resource supply. Since these wires are fixed throughout placement, no updating is required during run time of placement once the resource supply is reduced.

A cell normally consists of a blockage in the first metal layer, which is as large as the cell outline, a few blockages in the second metal layer and pins in the first or the second metal layer. Note that even though pins might be in the first metal layer, pins are still considered to be obstacles in the second metal layer to other nets because of accessibility. These blockages will decrease the routing resource supply. The decremental values can be pre-processed before global optimization starts.

2.4 Cell Inflation

We introduce some denotations again.

Real Area of cell: The real area of the cell is the product of length and height of the cell. It is the actual size of the cell.

We use the real area of the cell for accurate computation, such as in the first process of global optimization to get the global optimized position of the cell, in routing estimation, etc.

Virtual Area of cell: The virtual area of the cell is the area of cell after inflated. The value reveals not only the size of the cell but also the routing demand of the bin where the cell locates in currently. We use the virtual area to eliminate the congestion, such as in the latter process of global optimization to decrease the routing congestion of the bins. We can get the value of virtual area through multiple a ratio to the real area. The ratio can be determined by the degree of the congestion of the bin.

Then we will introduce the method of cell inflation.

After doing the routing estimation and detecte the congested bins, in which we will use the real area of the cell for accurate, what we should do is inflate the areas of the cells which locate in the congested region. After we have found that bin A is a congested bin, we will expand the same ratio to the area of cells which located in bin A to get the virtual area of the cells. Then we implement next partition loop, during which we use the virtual area of cells, due to the rule of area-balance, some cells located in bin A previously will be forced to move to other bins. As a result, the number of cells located in bin A will decrease. Generally speaking, with the decrease of cell number, the number of nets which pass through bin A will also decrease. As a result, the routing resource supply in bin A will increase, and in the mean time, the routing demand will decrease. Therefore, the congestion...
in bin A is alleviated or eliminated.

As shown in Fig. 4a, there are four cells and five nets in the left-up bin A, and bin A is congested. Now we expand the areas of the four cells which located in bin A. After next partition loop, due to area-balance, cell C is forced to move to bin B. At this time, there are only three cells and four nets in bin A, so the routing resource supply increases while the routing demand decrease, as in Fig 4b.

2.5 Cell Moving

After the stage of global placement, we will implement a step of cell moving to erase the routing congestion more deeply. This is the greedy method to decrease congestion. We find the most congested bin S, and the most uncongested bin T currently. We draw a line from S to T as shown in Fig.5, then use a step line SABCD...T to fit the line. Next, we will do some operations along the step line. We will map the cells in bin S to bin S and A according to the coordinates, then mapping the cells which are located in A before and the new cells come from bin S to bin A and B. We will do the same operation until reach bin T. After that, we do the congestion detection, find another most congested bin and most uncongested bin and do the step of cell moving. We will iterate the cell moving several times. Experimental results have shown the cell moving can also improve the quality of our placement considerably.

III. MAIN ALGORITHM DESCRIPTION

The information of the circuit is read through LEF/DEF Format. Then we will build up an m × n bin structure.

3.1 Global Optimization and Slicing Partition

The main loop of the global placement is composed of global optimization, congestion detection, congestion eliminating and slicing partitioning.

In each global optimization step, a mathematical programming problem is derived and solved. The objective function is based on nets’ quadratic wire length model and we assign a weight \( w_n \) to net \( n \). Therefore, the objective is to minimize the weighted sum of the quadratic wirelength of all nets:

\[
\phi = \sum_{n=0}^{N} \left( (x_n - x_{j_n})^2 + (y_n - y_{j_n})^2 \right) w_n
\]

At the \( h \)th partition level, the placement plane will be divided into \( 2^h \) regions, each containing a subset of cells. If \( r \) is a region, we use \( \tau_r \) to denote the set of cells contained in \( r \), and use \( (\mu_r, \nu_r) \) to denote the coordinates of the center of region \( r \). Thus for each region \( r \), we get two constraints on the global placement:

\[
\sum_{i=0}^{m} x_i = \mu_r, \quad \sum_{i=0}^{n} y_i = \nu_r
\]

Combining the objective function with constraints, we get a constrained quadratic programming problem:

\[
LQP: \min \Phi(x) = 1/2 * X'QX + b'yX | A'X = u'X
\]

We use Lagrange Relaxation Method to solve the LQP problem. According to [6], we can get the global optima:

\[
x = Q'b, \quad y = Q'b_y
\]

where the vectors \( x \) and \( y \) denote the coordinates of the movable cells to be placed. We change the format of above formula and take use of preconditioned conjugate-gradient method (PCG) to solve this problem.

First we do the global optimization and partition on the chip plane, which is from level 0 to partition-level limit \( N \), to get the global optimization for each cell. In this first process, we use the real area of the cell for accuracy to get the optimized optimization. Then we will check the status of routing congestion, find the congested bins, in which also using the real area of cell for the same reason. Next, we will expand the areas of cells with the front method to get the virtual areas of cells and will adjust the positions of cells to decrease the congestion. To make the adjustment more naturally and to eliminate routing congestion more thoroughly, we will withdraw the partition level to level (N-6), restart our global optimization and partition. In partitions of this latter process, we use the virtual areas of cells. Because of the rule of partition, part of the cells in the congested bins will move out of the bins automatically and naturally. When reach the partition limit \( N \) again, we will do the congestion detection, expand the areas of cells to update the virtual areas of cells, withdraw the partition level again. After do the process several times, we will go out of the process and give the output. The loop flow shows in Fig.6.

The algorithm can decrease the maximum congestion markedly and have no congestion vibration. Of course, with this method the total wire length will increase. But because
we adjust the position of cells naturally, the total wire length only increase a little. Because we withdraw the partition level several times, the algorithm’s total time will increase.

3.2 Post-Processing

After the stage of global placement, we will perform a step of cell moving to erase the routing congestion more deeply as described in section 2.4. Finally, a final placement called FAME[7] is called to assign the cells into slots and refine the solution we have obtained. In the last two process, we use the real areas of cells. So the inflated method will not affect the purpose of final placement.

IV. EXPERIMENTAL RESULTS

We have implemented our new congestion-driven placement algorithm on ULTRA-SPARC workstations in C language. To investigate the efficiency of our approach, we have tested it with a set of sample circuits from American industry. The characteristics of the circuits are listed in Table I. We used our approach without congestion constraints to obtain results by conventional placement (CP), and by congestion-driven placement (CDP) for respective test cases. The experimental results are summarized in Table II. X_Max is the number of track shortage in the most congested bin in x direction, and Y_Max has similar meaning. From the results, it is shown that our approach can cut down the maximum congestion at about 30-40 percent. Although the total wirelength increases slightly, it is less important than routability by now. Even though the run time increases by about one time, our algorithm still runs fairly fast, it takes about one hour with circuit U28070 that consists of more than 160 thousand cells.

V. CONCLUSION

In this paper, we have presented a new congestion-driven placement based on cell inflation. In our approach, we have used the method of probability-estimation to evaluate the routing of nets. We also take use of the schemes of cell inflation to eliminate the routing congestion. Moreover, further reduction of congestion is obtained by means of cell moving. The experimental results show that our method is very effective and efficient.

<table>
<thead>
<tr>
<th>#Circuits</th>
<th>#Pin</th>
<th>#Cell</th>
<th>#Net</th>
<th>#Row</th>
<th>#Layer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table</td>
<td>142</td>
<td>4596</td>
<td>4674</td>
<td>64</td>
<td>5</td>
</tr>
<tr>
<td>Design</td>
<td>1028</td>
<td>44283</td>
<td>44943</td>
<td>164</td>
<td>5</td>
</tr>
<tr>
<td>U05614</td>
<td>388</td>
<td>32112</td>
<td>36452</td>
<td>186</td>
<td>3</td>
</tr>
<tr>
<td>U11228</td>
<td>836</td>
<td>64224</td>
<td>72966</td>
<td>263</td>
<td>3</td>
</tr>
<tr>
<td>U28070</td>
<td>1604</td>
<td>160560</td>
<td>181932</td>
<td>417</td>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Circuits</th>
<th>Wire (um)</th>
<th>R (s)</th>
<th>X Max</th>
<th>Y Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table</td>
<td>33643475</td>
<td>73.6</td>
<td>90</td>
<td>16.5</td>
</tr>
<tr>
<td>CDP</td>
<td>33934358</td>
<td>212.7</td>
<td>58.5</td>
<td>4.5</td>
</tr>
<tr>
<td>Design</td>
<td>50503240</td>
<td>542.4</td>
<td>193</td>
<td>106.5</td>
</tr>
<tr>
<td>CDP</td>
<td>50772567</td>
<td>1187.9</td>
<td>164</td>
<td>85</td>
</tr>
<tr>
<td>U05614</td>
<td>26148536</td>
<td>543.9</td>
<td>553</td>
<td>455</td>
</tr>
<tr>
<td>CDP</td>
<td>272246965</td>
<td>1255</td>
<td>410</td>
<td>400</td>
</tr>
<tr>
<td>U11228</td>
<td>62317492</td>
<td>938.1</td>
<td>1287</td>
<td>1078</td>
</tr>
<tr>
<td>CDP</td>
<td>64068224</td>
<td>1777.6</td>
<td>699.5</td>
<td>588.5</td>
</tr>
<tr>
<td>U28070</td>
<td>215642114</td>
<td>1714</td>
<td>4173</td>
<td>4032</td>
</tr>
</tbody>
</table>

REFERENCES