Congestion Prediction in Early Stages

Chiu-wing Sham
Department of Computer Science & Engineering
The Chinese University of Hong Kong
Hong Kong
cwsham@cse.cuhk.edu.hk

Evangeline F. Y. Young
Department of Computer Science & Engineering
The Chinese University of Hong Kong
Hong Kong
fyyoung@cse.cuhk.edu.hk

ABSTRACT

Routability optimization has become a major concern in the physical design cycle of VLSI circuits. Due to the recent advances in VLSI technology, interconnect has become a dominant factor of the overall performance of a circuit. In order to optimize interconnect cost, we need a good congestion estimation method to predict routability in the early stages of the design cycle. Many congestion models have been proposed but there's still a lot of room for improvement. Some existing models [6] are dependent on parameters that are related to the actual congestion of the circuits. Besides, routers will perform rip-up and re-route operations to prevent overflow but most models do not consider this case. The outcome is that the existing models will usually underestimate the routability. In this paper, we propose a new congestion model to solve the above problems. The estimation process is divided into three steps: preliminary estimation, detailed estimation and congestion redistribution. We have compared our new model and some existing models with the actual congestion measures obtained by global routing some placement results with a publicly available maze router [2]. Results show that our model has significant improvement in prediction accuracy over the existing models.

Categories and Subject Descriptors

B.8.2 [**Performance and Reliability**]: Performance Analysis and Design Aids; B.7.2 [**Integrated Circuits**]: Design Aids

General Terms

Performance

Keywords

Interconnect Estimation, Placement, Floorplanning

1. INTRODUCTION

1.1 Motivations

The routability problem is a demand and supply problem of the routing resources. In the early stages of the design cycle, the shapes and locations of the modules on a chip are planned, and the result of this planning step will greatly affect the overall performance of the final design. In some advanced systems using the deep submicron technology today, the extremely high design densities can result in a major escalation in routing demand. Over-congestion will deteriorate circuit performance or even lead to an unroutable solution. Thus, routability optimization has become a major concern in physical design. Unfortunately, minimizing total wirelength does not have significant impact on routability [14]. We need an accurate congestion prediction and an efficient congestion removal technique.

In an automated IC implementation flow, congestion information will be available only after detailed routing. Excessive congestion will result in a local shortage of routing resources. This will lead to a large expansion in area, or even an unroutable design failing to achieve timing closure after detailed routing. In this case, the design process must be restarted from an early stage such as floorplanning and placement. Thus, it is desirable to detect and remove congested regions in the early designing stages. A good congestion model is needed for accurate interconnect analysis and prediction during the early stages of the design process.

1.2 Related Works

Because of the importance of this congestion estimation problem, many models have been proposed. In the papers [5, 4, 10], a packing is divided into tiles and congestion is estimated in each tile, assuming that each net is routed in either L- or Z-shape. In the paper [8], the congestion model used is the average net density on the half-perimeter boundaries of different regions in a floorplan. In the papers [7, 9, 12], probabilistic analysis is used to estimate congestion and routability. They assume that all feasible routes have the same probability of being selected. In practice, routes of less bends are more desirable. In the papers [6, 15], extended versions of [9] are proposed. The authors take into account the impact of the number of bends in a routing path on the probability of occurrence of the path. However, the accuracies of their congestion models will depend on the accuracies of their predictions on the distribution of the number of bends. The paper [16] predicts congestion by using the Rent's rule. However, connections of the nets are already known in the floorplanning and placement stage, and we should be able to predict congestion more accurately than simply using the Rent's rule. The papers [11, 13, 14] use global routers to estimate congestion, which will be more accurate but the runtime penalty is high.

1.3 Our Contributions

Congestion prediction is an important part of interconnect planning in the early stages of the physical design cycle. Although some congestion models have been proposed, the accuracies of the predictions still have a lot of room for improvement. Routers will perform rip-up and re-route operations to avoid overflow but most models do not consider this case. The outcome is that the existing models will very often over-estimate the number of over-congested regions. In this paper, we propose a new congestion prediction method to solve the above problems. The estimation process is divided into three steps: preliminary estimation, detailed estimation and congestion redistribution. To avoid over-estimating congestion, we perform a preliminary estimation step first to determine which regions are likely to be over-congested. A region should be more attractive to net routing if it is less congested. Then, we will make use of this information to predict the congestion measures during the detailed estimation step. We use a diagonal based congestion model because of its simplicity and experimental results have shown that this model can give accurate estimations. Finally, congestion redistribution will be performed to simulate the rip-up and re-route operations of the detailed routing step by moving wires from over-congested regions to less congested regions. We have compared our new model and some existing models with the actual congestion measures obtained by global routing some placement results (using the Capo placer [3]) with a publicly available maze router [2]. Results show that our model can make significant improvement in the estimation accuracy over the other models.

This paper is organized as follows. First, an overview of our 3-step approach will be described in section 2. Details of the preliminary estimation, detailed estimation and congestion redistribution steps will be described in section 3, 4 and 5 respectively. We will also consider blockages which will be discussed in section 6. Finally, the experimental results will be shown in section 7.

2. OVERVIEW OF OUR DESIGN

We will divide the placement into a tile structure. All the nets are decomposed into a set of 2-pin nets by the minimum spanning tree method. The notations used are shown in table 1. We use a 3-step approach as follows:

- Preliminary Estimation: We estimate the congestion measure at each tile roughly according to the bounding box of each net so that we can determine which regions are likely to be over-congested.
- Detailed Estimation: Based on the information obtained from the preliminary estimation step, we estimate the congestion measure at each tile by using a diagonal-based congestion model.
- Congestion Redistribution: We will simulate the rip-up and re-route process of the routing stage by moving wires from over-congested tiles to less congested tiles.

Notation	Description
t_l	Length of a tile
c^h_{max}	Maximum horizontal wire capacity inside a tile
c^v_{max}	Maximum vertical wire capacity inside a tile
D_k	The shortest Manhattan distance between the source and sink of net k
$d_k(x,y)$	The distance from the source of net k to tile (x,y)
$P_k(x,y)$	A rough estimation of the probability of net k passing through tile (x,y)
$P(x,y)$	Congestion at tile (x,y) obtained from the preliminary estimation step
$W(x,y)$	The weight of tile (x,y)
$E_k(x,y)$	The probability of net k passing through (x,y)
$E^h_k(x,y)$	The probability of net k passing through (x,y) horizontally
$E^v_k(x,y)$	The probability of net k passing through (x,y) vertically
$E^h(x,y)$	The expected number of wires passing through (x,y) horizontally
$E^v(x,y)$	The expected number of wires passing through (x,y) vertically
$A^h(x,y)$	The actual number of wires passing through (x,y) horizontally obtained from the global router
$A^v(x,y)$	The actual number of wires passing through (x,y) vertically obtained from the global router
(s^x_k, s^y_k)	Co-ordinate of the source of net k
(t^x_k, t^y_k)	Co-ordinate of the sink of net k
T_k	The set of tiles inside the bounding box of net k
$T_k(d)$	The set of tiles inside the bounding box of net k and being d tiles away from the source
$B(x,y)$	Degree of blocking at tile (x,y)

Table 1: Notations

3. PRELIMINARY ESTIMATION

In practice, we will choose to route a net over the tiles that are less congested to prevent overflow. It means that some tiles are more attractive to net routing and some tiles are less. However, this fact is usually ignored in traditional congestion models. In our approach, a preliminary estimation of the congestion map will be performed to obtain this information. If a rough estimation of the congestion measure of a tile, $P(x,y)$, is above the maximum wire capacity, the tile (x,y) will be less attractive to net routing. On the other hand, if $P(x,y)$ is well below the maximum wire capacity, the tile (x,y) will be more attractive to net routing. We will make use of these $P(x,y)$ values to improve the accuracy of the detailed estimation step.

In this preliminary estimation step, we assume that all the tiles inside the bounding box of a net k, T_k, have the same probability, $P_k(x,y)$, of being passed through by net k. In addition, we assume that the nets can be routed in their shortest Manhattan distances. The wirelength and the area of the bounding box can be computed as $|t_k^x - s_k^x| + |t_k^y - s_k^y| + 1$ and $(|t_k^x - s_k^x| + 1) \times (|t_k^y - s_k^y| + 1)$ respectively. $P_k(x, y)$ can thus be calculated by the following equation:

$$P_k(x, y) = \frac{|t_k^x - s_k^x| + |t_k^y - s_k^y| + 1}{(|t_k^x - s_k^x| + 1) \times (|t_k^y - s_k^y| + 1)} \quad (1)$$

We can then obtain a preliminary estimation by adding up the congestion measures due to different nets:

$$P(x, y) = \sum_{\text{all } k} P_k(x, y) \quad (2)$$

4. DETAILED ESTIMATION

In our approach, we will predict the congestion measures by using a diagonal based model during the detailed estimation step. We first assume that all the nets are routed in their shortest Manhattan distances. The tiles inside the smallest bounding box of net k can be divided into $D_k - 1$ divisions where D_k is the shortest Manhattan distance between the source and the sink. An example is shown in figure 1.

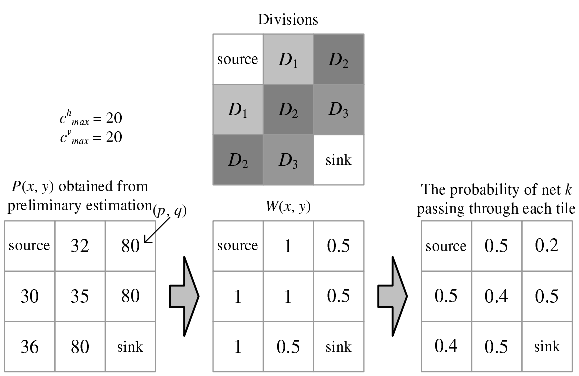

Figure 1: An example of computing the congestion measures for a two-pin net in the detailed estimation step

In this example, the tiles are divided into three divisions D_1, D_2 and D_3. Intuitively, if the net is restricted to be routed within the bounding box, the net will pass through exactly one tile in each division. We assume that the net will pass through the tiles in the same division with probabilities weighted according to $W(x, y)$ where $W(x, y)$ is computed by the following equations according to the $P(x, y)$ obtained in the preliminary estimation step.

$$W(x, y) = \begin{cases} 1 & : P(x, y) < (c_{max}^h + c_{max}^v) \\ \frac{c_{max}^h + c_{max}^v}{P(x,y)} & : \text{otherwise} \end{cases} \quad (3)$$

If $P(x, y)$ is smaller than the sum of c_{max}^h and c_{max}^v, the tile (x, y) is unlikely to be over-congested and so $W(x, y)$ is 1. If $P(x, y)$ is larger than the sum of c_{max}^h and c_{max}^v, that tile should have a smaller $W(x, y)$ when $P(x, y)$ is larger. It reflects the case in the routing stage that the nets will be routed to pass through less congested tiles. Hence, the probability of net k passing through (x, y), $E_k(x, y)$, can be calculated according to the weight of each tile, $W(x, y)$, by the following equation:

$$E_k(x, y) = \frac{W(x, y)}{\sum_{(i,j) \in T_k(d_k(x,y))} W(i, j)} \quad (4)$$

In the example of figure 1, c_{max}^h and c_{max}^v are 20. When we focus on division D_2, (p, q) the tile at the upper right corner is a congested tile according to the preliminary estimation step because $P(p, q)$ is 80, which is larger than the sum of c_{max}^h and c_{max}^v. Thus, $W(p, q)$ should be smaller than 1 and it is computed as 0.5 according to equation 3. Hence, the probability of net k passing through (p, q), $E_k(p, q)$, is 0.2. It is smaller than the others in the same division because the tile (p, q) is likely to be over-congested.

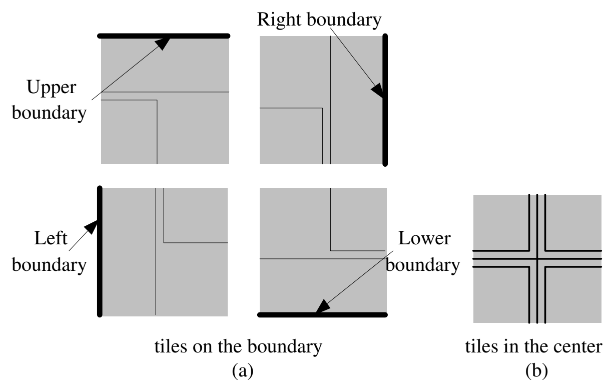

Figure 2: Possible routes inside a tile

In addition, a net may pass through a tile either horizontally or vertically. When a net is routed from the upper-left corner to the lower-right corner of the bounding box, the net may pass through a tile with a path as shown in figure 2. If the tile is on the boundary of the bounding box, the route may pass through the tile in two ways. The four different cases of the tile lying along the top, the left, the bottom and the right boundary are shown in figure 2a. If the tile is on the left or right (at the top or bottom) of the bounding box, the length of the route passing through the tile horizontally (vertically) is $0.5t_l$ and the length of the route passing through the tile vertically (horizontally) is $1.5t_l$. If a tile is not on the boundary of the bounding box, the net may pass through the tile in four different ways. They are shown in figure 2b. In this case, the length of the route passing through the tile horizontally or vertically is $3t_l$. Thus, we can calculate $E_k^h(x, y)$ and $E_k^v(x, y)$ by the following equations:

$$E_k^h(x, y) = \begin{cases} \frac{E_k(x,y)}{2} & : s_k^x < x < t_k^x \text{ and } s_k^y < y < t_k^y \\ \frac{3 \times E_k(x,y)}{4} & : y = s_k^y \text{ or } y = t_k^y \\ \frac{E_k(x,y)}{4} & : x = s_k^x \text{ or } x = t_k^x \end{cases} \quad (5a)$$

$$E_k^v(x, y) = \begin{cases} \frac{E_k(x,y)}{2} & : s_k^x < x < t_k^x \text{ and } s_k^y < y < t_k^y \\ \frac{E_k(x,y)}{4} & : y = s_k^y \text{ or } y = t_k^y \\ \frac{3 \times E_k(x,y)}{4} & : x = s_k^x \text{ or } x = t_k^x \end{cases} \quad (5b)$$

Finally, the expected number of wires passing through (x, y) horizontally and vertically, $E^h(x, y)$ $(E^v(x, y))$, can be calculated by following equations:

$$E^h(x,y) = \sum_{\text{all net } k} E_k^h(x,y) \quad (6a)$$

$$E^v(x,y) = \sum_{\text{all net } k} E_k^v(x,y) \quad (6b)$$

5. CONGESTION REDISTRIBUTION

In real routing, if some tiles are over-congested or some nets cannot be routed, rip-up and re-route will be performed. In our approach, we perform congestion redistribution to achieve the same purpose of moving wires from over-congested tiles to less congested tiles. We will only move around those congestion measures within the same diagonal (division). An example is shown in figure 3. In $T_k(1)$ of net k, the tile with congestion estimation 7.2 is the most congested in this division but it is not over-congested (less than the maximum wire capacity of a tile). Thus, no action will be taken. In $T_k(2)$, the tile with 12.4 is the most congested in this division and is over-congested. Thus, we will move 0.2 (net k's contribution to the congestion measure of this tile) from this tile to the least congested tile of the same division.

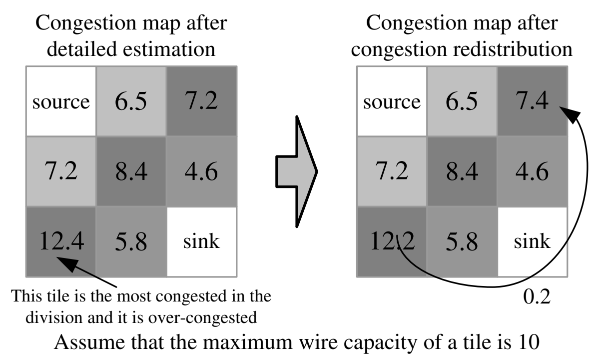

Figure 3: An example of congestion redistribution

In general, we will find the tile, (x_m, y_m), with the maximum vertical (horizontal) congestion and the tile, (x_l, y_l), with the minimum vertical (horizontal) congestion from each division of all the nets. If the tile with the maximum vertical (horizontal) congestion is over-congested, we will move $E_k^v(x_m, y_m)$ ($E_k^h(x_m, y_m)$) from (x_m, y_m) to (x_l, y_l). After redistribution, the summation of $E_k^v(x,y)$ ($E_k^h(x,y)$) in the same division still equals one. Thus, the assumption that each net will pass through exactly one tile in each division within the bounding box still holds.

6. BLOCKAGES

Blockages are regions with reduced routing resources. There are two types of routing blockages: partial or complete. Partial blockages block a certain number of layers, but there are still limited routing resources available. Complete blockages block all the layers, and no net can pass through those blockages. The degree of blocking, $B(x,y)$, at a tile (x,y) can be calculated by the following equation:

$$B(x,y) = \frac{\text{No. of blocked layers}}{\text{Total number of layers}} \quad (7)$$

Then the weight of a tile (x,y), $W(x,y)$, will be updated by the following equation:

$$W(x,y) = W(x,y) \times (1 - B(x,y)) \quad (8)$$

Note that $B(x,y)$ equals one when all the layers are blocked (complete blockage).

7. EXPERIMENTAL RESULTS

In the experiments, the test cases used are the ISPD-02 suite circuits [1]. The length of a tile, t_l, is $40\mu m$. The detailed information of the testing circuits are shown in table 2. Each circuit is first placed using a wirelength driven placer, Capo [3]. Four placement solutions are obtained for each test case. Global routing is then performed on each placement solution by a maze routing based global router [2]. During global routing, we set the value of wiring capacity to simulate two environments: more congested and less congested. In the experiments of table 3, there are about $0\% - 2\%$ tiles that are over-congested during global routing. In the experiments of table 4, there is no over-congested tile. Different congestion models are then used to estimate the congestion of the placed circuits and their estimations are then compared with the actual congestion measures obtained from the global router.

Test Cases	No. of Cells	No. of Nets	No. of 2-pin Nets	No. of Tiles
*ibm*01	12506	14111	36455	57 × 57
*ibm*02	19342	19584	61615	82 × 82
*ibm*03	22853	27401	66172	89 × 87
*ibm*04	27220	31970	73889	85 × 86
*ibm*05	28146	28446	97862	60 × 60
*ibm*06	32332	34826	93366	81 × 82
*ibm*07	45639	48117	127522	97 × 97
*ibm*08	51023	50513	154377	104 × 103
*ibm*09	53110	60902	161186	118 × 118
*ibm*10	68685	75196	222371	194 × 189
*ibm*11	70152	81454	199332	130 × 129
*ibm*12	70439	77240	240520	171 × 171
*ibm*13	83709	99666	257409	141 × 141
*ibm*14	147088	152772	394044	151 × 151
*ibm*15	161187	186606	529215	170 × 169
*ibm*16	182980	190048	588775	204 × 203
*ibm*17	184752	189581	670455	182 × 182
*ibm*18	210341	201920	617777	163 × 163

Table 2: Information of the test cases

We compare our congestion model with the models from Lou [9] and Westra [15]. We have implemented all the congestion models and compared the estimations with the results of the maze router. All programs were written in the C language and run on a machine (Sun Blade 1000) with $750MHz$ processor and $2GB$ memory. We will compare the congestion models by calculating the mean, μ, and standard deviation, s, of the estimation errors according the following equations:

Test Cases	c^h_{max} and c^v_{max}	Lou's Model		Westra's Model		Detailed Estimation only		Our 3-step Approach	
		μ (10^{-2})	s (10^{-2})	μ (10^{-2})	s (10^{-2})	μ (10^{-2})	s (10^{-2})	μ (10^{-2})	s (10^{-2})
*ibm*01	27	17.36	14.87	15.06	13.68	14.32	13.67	12.63	11.30
*ibm*02	46	16.96	18.10	14.12	17.12	11.63	15.19	10.15	13.16
*ibm*03	45	32.34	26.30	27.59	25.70	24.29	23.90	20.66	21.88
*ibm*04	140	4.51	4.90	4.37	6.17	4.12	4.65	4.10	4.62
*ibm*05	70	27.21	16.56	23.27	15.69	18.28	13.76	14.22	11.82
*ibm*06	47	23.43	20.16	21.20	20.07	17.61	18.54	14.65	16.29
*ibm*07	53	13.51	12.91	12.05	12.47	10.44	10.72	9.44	9.40
*ibm*08	400	2.17	2.83	2.07	4.18	1.86	2.71	1.85	2.64
*ibm*09	50	15.76	15.61	13.62	15.80	11.43	12.63	10.34	10.78
*ibm*10	50	18.16	15.56	11.71	14.27	9.13	10.75	8.36	9.12
*ibm*11	55	14.38	13.31	11.55	12.65	10.28	10.84	9.43	9.24
*ibm*12	60	20.79	18.98	14.86	21.27	11.42	13.28	10.16	10.84
*ibm*13	60	14.67	13.08	11.97	12.48	10.21	10.51	9.35	9.14
*ibm*14	65	13.38	11.49	10.52	10.47	9.88	9.42	9.32	8.18
*ibm*15	80	17.19	14.50	12.06	13.32	10.33	10.05	9.28	8.34
*ibm*16	60	19.54	16.01	14.47	15.18	11.58	11.70	10.58	10.29
*ibm*17	80	18.56	14.48	12.70	11.92	10.57	9.91	9.69	8.61
*ibm*18	70	17.20	14.36	14.56	13.13	12.76	11.49	11.65	10.23
Average		17.06	14.67	13.76	14.20	11.67	11.87	10.33	10.32
Comparing with Lou's		0.00%	0.00%	−19.34%	−3.20%	−31.58%	−19.05%	−39.48%	−29.61%

Table 3: Comparison of the congestion models for more congested circuits

$$\mu_h = \frac{\sum_{(x,y)\in T} \frac{|A^h(x,y)-E^h(x,y)|}{c^h_{max}}}{|T|}$$
$$\mu_v = \frac{\sum_{(x,y)\in T} \frac{|A^v(x,y)-E^v(x,y)|}{c^v_{max}}}{|T|}$$
$$\mu = \frac{\mu_h + \mu_v}{2} \quad (9a)$$

$$s_1 = \sum_{(x,y)\in T} \left(\frac{|A^h(x,y) - E^h(x,y)|}{c^h_{max}} - \mu_h\right)^2$$
$$s_2 = \sum_{(x,y)\in T} \left(\frac{|A^v(x,y) - E^v(x,y)|}{c^v_{max}} - \mu_v\right)^2$$
$$s = \sqrt{\frac{(s_1 + s_2)}{2 \times |T|}} \quad (9b)$$

where T is the set of all tiles that either their actual congestion measures or estimated congestion measures are nonzero.

The experimental results are shown in table 3 and table 4. The values are the averages of the four placement solutions for each test case. We can see that the detailed estimation step alone can give smaller means and standard deviations in most cases than the Lou's [9] and Westra's [15] models. The accuracies can be further improved when we can simulate the rip-up and re-route operations by performing the preliminary estimation and congestion redistribution steps. When the circuits are less congested, the accuracy of our 3-step approach is about 30% better than that of the Lou's model and the improvement is about 40% when the circuits are more congested. For example, the difference in the means of the estimation error between our 3-step approach and the Lou's model is 0.0183 for the data set *ibm*01 in the less congested case. It means that the error has been reduced by 0.549 net segments in each tile and the error has been reduced by $(0.549 \times |T|)$ 1882.521 net segments for the whole circuit. This shows that significant improvement on accuracy can be made using our approach and the improvement becomes even more significant when the circuits are congested.

In figure 4 and figure 5, the congestion maps obtained by different congestion models and the actual one (obtained by global routing) are shown. We can see that there are many regions that are predicted as over-congested in the Lou's and Westra's models and there are also a lot of empty regions in their models. However, the nets can be ripped up and re-routed to avoid passing through the over-congested regions. There is thus no over-congested region after global routing and most of the tiles in the placed region are used by some nets. In our modeling, we applied the preliminary estimation and congestion redistribution steps, and a similar congestion map can be obtained. Clearer comparisons can be illustrated by the error distributions of different congestion models in figure 6 and figure 7. We can see that differences occur in the surroundings of the over-congested tiles. It is because the global routing step will rip up the nets from the over-congested tiles and re-route them in the less congested tiles in the surroundings. Results show that we can improve the congestion estimation accuracy in different parts of the circuit.

In addition, we have compared the runtime of different congestion models. The results are shown in table 5. If we apply the detailed estimation step only, the runtime is faster than both the Lou's [9] and Westra's [15] models. If we also apply the preliminary estimation and congestion

Test Cases	c^h_{max} and c^v_{max}	Lou's Model		Westra's Model		Detailed Estimation only		Our 3-step Approach	
		μ (10^{-2})	s (10^{-2})	μ (10^{-2})	s (10^{-2})	μ (10^{-2})	s (10^{-2})	μ (10^{-2})	s (10^{-2})
*ibm*01	30	12.78	11.08	11.18	10.21	11.73	11.12	10.95	9.92
*ibm*02	55	9.13	9.78	8.04	9.47	7.14	8.94	6.81	8.29
*ibm*03	50	13.37	13.35	10.93	13.06	9.18	10.93	8.15	9.13
*ibm*04	150	3.90	4.08	3.78	4.90	3.54	3.85	3.54	3.83
*ibm*05	80	12.87	9.06	10.60	8.11	9.62	7.49	9.23	6.96
*ibm*06	55	10.74	9.97	10.06	9.97	9.57	9.83	8.82	8.69
*ibm*07	60	9.44	8.82	8.50	8.55	8.07	7.90	7.72	7.35
*ibm*08	500	1.66	2.07	1.56	2.92	1.42	1.96	1.42	1.95
*ibm*09	60	10.18	10.25	8.92	10.44	8.30	8.76	7.95	8.09
*ibm*10	60	12.42	10.78	8.11	9.80	6.51	7.37	6.31	6.92
*ibm*11	65	10.47	9.68	8.36	8.98	7.75	7.87	7.46	7.28
*ibm*12	70	14.94	14.13	10.22	15.42	7.72	8.75	7.38	7.68
*ibm*13	70	10.56	9.54	8.53	8.84	7.74	7.74	7.46	7.26
*ibm*14	75	10.37	8.91	8.07	8.04	7.80	7.40	7.58	6.80
*ibm*15	90	12.19	10.66	8.32	9.41	7.70	7.24	7.46	6.71
*ibm*16	70	13.04	10.73	9.51	10.19	8.07	7.69	7.80	7.21
*ibm*17	90	12.52	9.74	8.18	7.59	7.38	6.50	7.20	6.12
*ibm*18	85	8.66	7.06	7.03	6.10	6.97	5.88	6.80	5.60
Average		10.51	9.43	8.33	9.00	7.57	7.62	7.22	6.99
Comparing with Lou's		0.00%	0.00%	−20.784%	−4.52%	−28.03%	−19.13%	−31.29%	−25.87%

Table 4: Comparison of the congestion models for less congested circuits

Test Cases	Lou's Model (s)	Westra's Model (s)	Detailed Estimation only (s)	3-step Approach (s)	Global Routing [2] (s)
*ibm*01	0.24	0.14	0.13	0.31	190
*ibm*02	0.46	0.28	0.27	0.60	454
*ibm*03	0.92	0.58	0.53	1.10	987
*ibm*04	0.94	0.54	0.50	1.00	806
*ibm*05	1.03	0.60	0.55	1.20	1058
*ibm*06	0.64	0.35	0.34	0.77	642
*ibm*07	1.16	0.87	0.67	1.44	1206
*ibm*08	1.46	1.00	0.94	1.96	2021
*ibm*09	1.50	1.02	0.95	2.02	2217
*ibm*10	5.09	4.56	4.20	7.93	8820
*ibm*11	2.25	1.61	1.51	3.07	3021
*ibm*12	6.00	5.49	5.08	9.60	10543
*ibm*13	2.93	2.05	1.90	3.91	4680
*ibm*14	4.45	3.14	2.94	5.93	9480
*ibm*15	6.99	5.51	5.11	10.21	14220
*ibm*16	8.01	6.32	5.90	11.68	15684
*ibm*17	9.77	7.81	7.27	14.24	20547
*ibm*18	4.84	3.14	2.92	6.43	11235
Ave.	3.26	2.50	2.32	4.63	5990

Table 5: Comparison of the runtime of the congestion models

redistribution steps, the runtime is slower. However, it is still acceptable because a more accurate congestion model can help us to spend less time in the later routing stage.

8. CONCLUSION

To conclude, we have proposed a 3-step approach to estimate congestion. With the additional processes of preliminary estimation and congestion redistribution, we can simulate the rip-up and re-route operations in the congestion prediction stage. As a result, we will not over-estimate the number of over-congested regions and experimental results show that the accuracy of the congestion estimation can be significantly improved efficiently.

9. ACKNOWLEDGMENTS

The work described in this paper was partially supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. CUHK4231/01E).

10. REFERENCES

[1] http://vlsicad.eecs.umich.edu/bk/ispd02bench/.

[2] http://www.cs.ucla.edu/ kastner/labyrinth/.

[3] A. E. Caldwell, A. B. Kahng, and I. L. Markov. Can recursive bisection produce routable placements? In *Proceedings of the 37th ACM/IEEE Design Automation Conference*, pages 477–482, 2000.

[4] C. C. Chang, J. Cong, D. Z. Pan, and X. Yuan. Interconnect-driven floorplanning with fast global wiring planning and optimization. In *Proc. SRC Tech. Conference*, 2000.

[5] H. M. Chen, H. Zhou, F. Y. Young, D. Wong, H. H. Yang, and N. Sherwani. Integrated floorplanning and interconnect planning. In *Proceedings of IEEE Internation Cnference on Computer-Aided Design*, pages 354–357, 1999.

[6] A. B. Kahng and X. Xu. Accurate pseudo-constructive wirelength and congestion estimation. In *Proceedings of International Workshop on System-level Interconnect Prediction*, pages 61–68, 2003.

[7] Kusnadi and J. D. Carothers. A method of measuring nets routability for MCM's general area routing problems. In *Proceedings of Internation Symposium on Physical Design*, pages 186–194, 1999.

[8] S. T. W. Lai, E. F. Y. Young, and C. C. N. Chu. A new and efficient congestion evaluation model in floorplanning: Wire density control with twin binary trees. In *Proceedings of Design, Automation and Test in Europe Conference and Exhibition*, 2003.

[9] J. Lou, S. Krishnamoorthy, and H. S. Sheng. Estimating routing congestion using probabilistic analysis. In *Proceedings of Internation Symposium on Physical Design*, pages 112–117, 2001.

[10] Y. C. Ma, X. L. Hong, S. Q. Dong, S. Chen, Y. C. Cai, C. K. Cheng, and J. Gu. Dynamic global buffer planning optimization based on detail block locating and congestion analysis. In *Proceedings of ACM/IEEE Design Automation Conference*, pages 806–811, 2003.

[11] P. N. Parakh, R. B. Brown, and K. A. Sakallah. Congestion driven quadratic placement. In *Proceedings of the 37th ACM/IEEE Design Automation Conference*, pages 275–278, 1998.

[12] C. W. Sham and E. F. Y. Young. Routability-driven floorplanning with buffer planning. In *IEEE Transactions on CAD of Integrated Circuit and System*, pages 470–480, April 2003.

[13] M. Wang and M. Sarrafzadeh. Modeling and minimization of routing congestion. In *Proceedings of the ASP-ACM/IEEE Design Automation Conference*, pages 185–190, 2000.

[14] M. Wang, X. Yang, and M. Sarrafzadeh. Congestion minimization during placement. In *IEEE Transactions on CAD of Integrated Circuit and System*, pages 1140–1148, October 2000.

[15] J. Westra, C. Bartels, and P. Groeneveld. Probabilistic congestion prediction. In *Proceedings of Internation Symposium on Physical Design*, pages 204–209, 2004.

[16] X. J. Yang, R. Kastner, and M. Sarrafzadeh. Congestion estimation during top-down placement. In *IEEE Transactions on CAD of Integrated Circuit and System*, pages 72–80, January 2002.

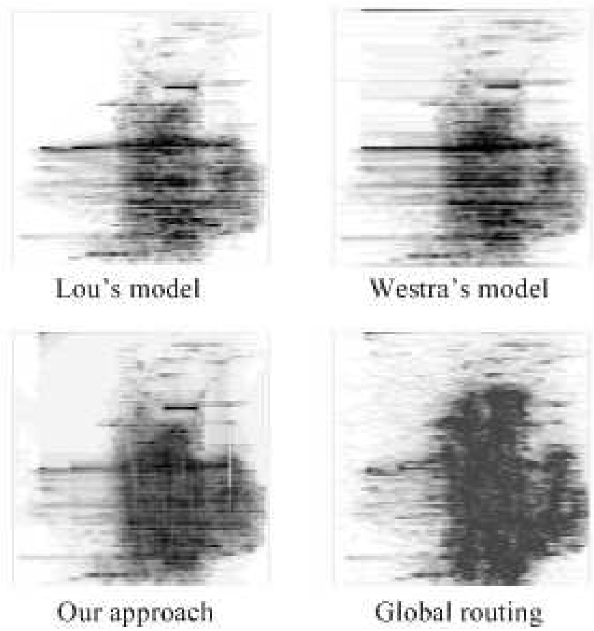

Figure 4: Congestion maps of horizontal wires (case: ibm03)

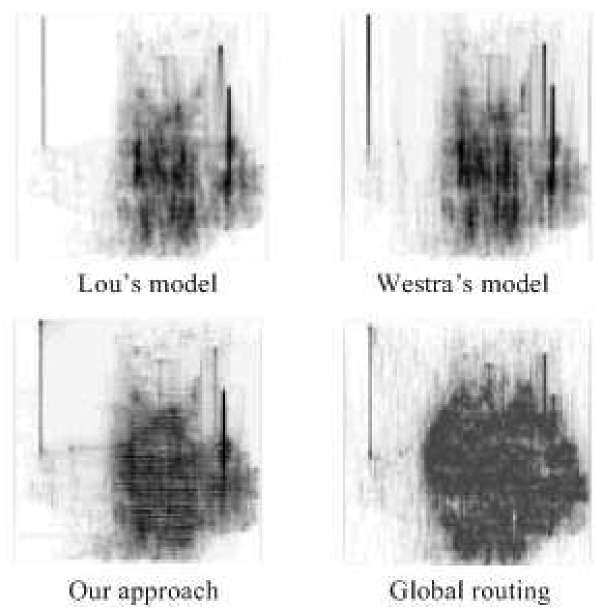

Figure 5: Congestion maps of vertical wires (case: ibm03)

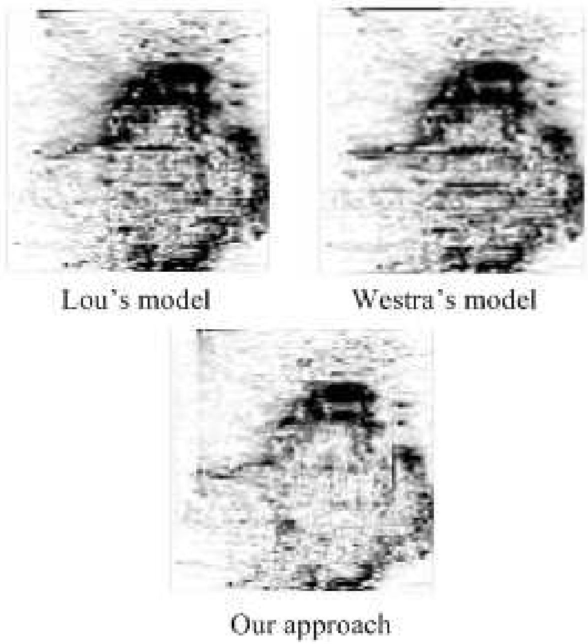

Figure 6: Error distribution of horizontal wires (case: ibm03)

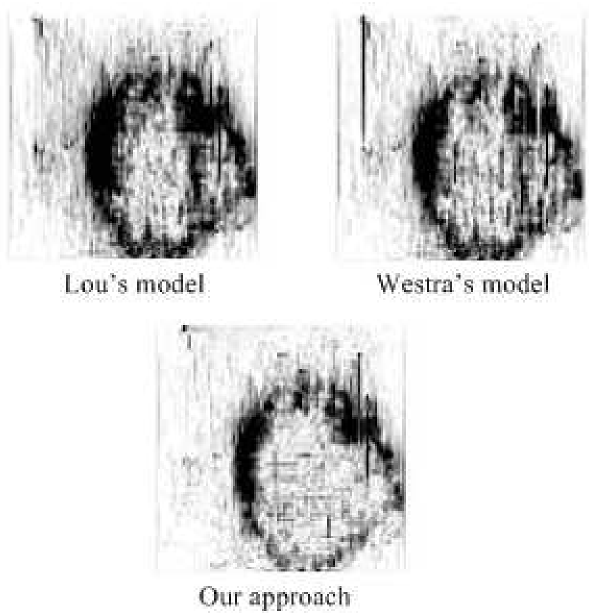

Figure 7: Error distribution of vertical wires (case: ibm03)