Accurate Pseudo-Constructive Wirelength and Congestion Estimation*

Andrew B. Kahng
UCSD, CSE and ECE Departments
La Jolla, CA 92093-0114
abk@ucsd.edu

Xu Xu
UCSD, CSE Department
La Jolla, CA 92093-0114
xuxu@cs.ucsd.edu

ABSTRACT

Accurate estimation of wirelength and congestion is one of the fundamental issues in VLSI physical design. Current probabilistic estimation methods fail to produce accurate results since they ignore congestion-related detouring and effects of the number of vias and bends. In this work, we propose a practical stochastic routing probability distribution model which includes the effects of blockage and the number of bends. The new model is tested by comparing the estimated routing probability distribution with the actual routing results of a commercial detailed router. An iterative congestion map construction algorithm based on the new probabilistic model is proposed for accurate wirelength and congestion map estimation. The results show that our proposed methods can improve the total wirelength estimation accuracy (i.e., reduce estimation error) by 90% on average with respect to the traditional RSMT wirelength estimate. Our methods also produce more accurate congestion maps than the previous congestion estimation method of [5] without significant runtime overhead.

Categories and Subject Descriptors

B.7.2 [**Hardware**]: INTEGRATED CIRCUITS—*Design Aids*; J.6 [**Computer Applications**]: COMPUTER-AIDED ENGINEERING

General Terms

Algorithms, Performance, Design.

1. Introduction and Motivation

In modern VLSI placement and routing tools, congestion is an important objective beyond wirelength, timing, and power. Routing congestion is a function of routing demand and routing supply, and measures the extent to which routing is feasible in each region, in the sense of taking no more resources than are available. It is generally believed that the total wirelength, which represents the routing demand, is closely related to congestion. Therefore, these two issues are addressed simultanously in many estimation approaches. The traditional wirelength estimation approach equates "wirelength estimation" with "estimation of the rectilinear Steiner minimal tree (RSMT) cost" [6].[1]

We distinguish two types of congestion estimations associated with placement. *Stochastic model based estimations* consider all possible paths by which a net can be routed. Every path is assigned a probability based on various assumptions. A routing probability distribution obtained by "smearing" across bounding boxes of minimum spanning tree (MST) edges is presented in [3]. Kusnadi et al. [2] propose a wire density map which distributes routing probability using Pascal's triangle: each grid is assigned a routing probability that is the sum of probabilities of its neighboring grids, a consequence of assuming that each route has the same probability of occurrence. The authors of [2] also assume that L-shapes are preferred for via minimization. However, no calibration with any actual router is given. Lou et al. [5] adopt a similar approach to estimate the routing probability. In their method, the chip is divided into rectangular regions; the probability of one region is proportional to the number of possible paths that pass through the region, and the effect of blockage is considered by adjusting the relevant bounding boxes. Recently, Brenner et al. [11] have incorporated Lou's estimation algorithm into a placer to detect routing criticalities and reduce congestion.

Empirical estimators use layout parameters to estimate wirelength and routability. In RISA [4], the routing demand is calculated by multiplying the net bounding box by a pin-count dependent net weight. The net weights are produced according to a wiring distribution map. Wang et al. [1] observe that (i) wirelength minimization is equal to minimization of average/total congestion, and (ii) wirelength minimization followed by congestion adjustment is better than simultaneous wirelength and congestion optimization. Yang et al. [10] argue that Rent's rule can be applied to estimate congestion since wirelength is a measure of routing demand. They estimate the Rent parameter through a rough min-cut based placement.

---

*This research was partially supported by the MARCO Gigascale Silicon Research Center.

[1] A side note: In this study, we focus on congestion-related wirelength estimation instead of traditional RSMT wirelength calculation. An accurate RSMT wirelength for each net is calculated (as in many previous approaches) and included in the inputs of our estimator.

SLIP'03, April 5–6, 2003, Monterey, California, USA.

Motivation

Most previous probabilistic methods are based on impractical assumptions. For example, they usually ignore the effects of vias and assume that all the paths without detouring have the same probability. They also assume that the routing demands for all line segments within the bounding box are the same. Of course, such simplifications do not reflect the reality of what routers try to optimize. For example, if we equate signal delay with cost, then the cost of a via is significant in actual routing, e.g., one via is approximately equal to 15 routing tracks in copper or 50 routing tracks in aluminum interconnect technology [9]. Therefore, paths with few bends (typically one to four bends) dominate the actual routing results.

As VLSI circuits are growing in both size and complexity, the actual routing of many nets tends to go *outside the net bounding box* due to congestion. Such detoured paths will cause additional congestion problems since they increase the routing demand. Therefore, in addition to the RSMT wirelength, the detoured wirelength needs to be considered in order to make an accurate estimation. Previous methods either ignore detoured nets or otherwise fail to solve this problem. It is therefore difficult for such methods to achieve accurate estimations.

In this work, we try to combine the probabilistic and empirical approaches to yield a probabilistic model that is informed by practical considerations. (1) We take into account the impact of the number of bends in a routing path on the probability of that path's occurrence. (2) To obtain a more accurate estimation of congestion and total wirelength, we also propose an *iterative* construction of the congestion map by fully considering the interaction between neighboring nets. Our goal is a simple, abstract scheme that is faster than a router but produces a good routing prediction. (3) To achieve desired accuracy in wirelength estimation, we take the *detouring of nets* into account so that the predicted congestion map can better fit actual routing results.

The rest of this paper is organized as follows. In Section 2, we set up a new routing probability distribution model based on the analysis of the routing probability distribution for paths with one to four bends. This model is validated by experimental results reported in Section 2.3. In Section 3, we propose the *Congestion_Factor* based iterative pseudo-constructive estimation method which uses the proposed model to estimate congestion. Experimental results in Section 3.3 show the effectiveness of the proposed method. Finally, Section 4 gives conclusions and future work.

2. New Routing Probability Distribution Model

2.1 Theoretical Analysis of Routing Probability Distribution Model

Intuitively, the number of bends b in a path is closely related to the number of vias, which increase practical "routing cost". We obtain the bend distributions after detailed routing for five industry testcases, whose characteristics are given in Table 2. As shown in Table 3, the paths with small numbers of bends have greater probabilities of occurrence than the paths with large numbers of bends. The distribution of bends is approximately Log-Normal. An empirical formula for the probability of b-bend paths is:

$$p_b = -0.05 + \frac{1.33}{\sqrt{2\pi}\eta b} e^{-\frac{\ln^2\left(\frac{b}{2.2}\right)}{2\eta^2}} \tag{1}$$

where η is a layout-dependent (and likely tool-dependent) parameter. The η values of the five testcases are 0.6, 0.36, 0.56, 0.53, and 0.62 respectively. We set $\eta = 0.6$ for all experiments below, and assume that paths without detouring *and with the same number of bends* have the same routing probability.

Notation	**Description**
N	The number of nets in the netlist.
T	The set of all nets in the netlist.
X_{max}, Y_{min}	Core layout area dimensions.
x_{max}, y_{min}	Bounding box dimensions for a net $t \in T$.
(x, y)	The point whose coordinate is (x, y).
$\overline{x, y}$	The line segment from (x, y) to $(x + 1, y)$.
b	b is the number of bends in a path.
p_b	The probability of b-bend paths.
$p_{d,b}$	The probability of b-bend detoured paths.
$BB(t)$	The bounding box of net t.
η	A parameter used for the bend distribution.
$D_b^t(x, y)$	Wire density for $\overline{x, y}$ due to b-bend paths.
w	The multi-bend factor.
V_k^p, V_k^r	Predicted value and observation at $x = k$.
N_r	Number of divided rectangular regions $r_i(i = 1...N_r)$ in the layout.
l	Detoured wirelength.
$n \times m$	Dimension of bounding box of a net.
S	A set of line segments.
$C(x, y)$	The congestion of the line segment $\overline{x, y}$.
$C'(x, y, t)$	The sum of wire density of all nets except for the net t on the line segment $\overline{x, y}$.
α	The value used to predict whether a net will detour or not.
$NP_H(n, m, b)$ $NP_V(n, m, b)$	The number of H-paths (or V-paths)with b bends for an $n \times m$ $BB(t)$.
$NP(n, m, b)$	$NP_H(n, m, b) + NP_V(n, m, b)$
$NP(n, m, b : l)$	The number of b-bend paths which pass the line $x = l$ for an $n \times m$ $BB(t)$.

Table 1: Notations used in this paper.

Testcase	# nets	# cells	# pins	# layers
A	44124	42758	113275	4
B	23521	21964	57524	4
C	7659	7016	19631	4
D	102715	96517	264310	4
E	25367	24343	60191	4

Table 2: Basic statistics of five industry testcases.

We can now develop our new practical model to be used in wirelength and congestion estimators. For a net t whose bounding box $BB(t)$ is an $n \times m$ region, set the left bottom corner of $BB(t)$ at $(0, 0)$ and the right upper corner at (n, m). The following definition applies to any unit-length horizontal (or vertical) segment in the grid induced by this coordinate system.

DEFINITION 1. *The wire density function $D^t(x, y)$ is the probability that the line segment $\overline{x, y} \in BB(t)$ will be routed (i.e., occupied) by the routing path.*

To simplify the model, we make the following assumptions.

Testcase	Number of Bends					
	1	2	3	4	5	6
A	0.24	0.36	0.19	0.11	0.05	0.03
B	0.08	0.67	0.11	0.05	0.02	0.02
C	0.23	0.42	0.11	0.05	0.03	0.02
D	0.13	0.37	0.28	0.10	0.04	0.02
E	0.17	0.34	0.20	0.12	0.06	0.04

Table 3: Bend distributions of point-to-point connections in five detail-routed industry testcases.

1. For 2-pin nets, we only specify the model for the nets whose two pins are located at the *left bottom* and *right top* corners of the bounding box. The model for the other configuration can be obtained by applying the transformation $y \rightarrow m - y$.
2. In the model, only the horizontal line segments are considered since the model for the vertical line segments can be similarly derived.
3. $n > 1$ and $m > 1$ since the wire density function for $n \leq 1$ or $m \leq 1$ is trivial.
4. Only the model for 2-pin nets is given. (3-pin (or multi-pin) nets can be decomposed into 2-pin nets.)
5. For the wire density model within $BB(t)$, only the paths without detouring are considered. (Detouring is considered in Section 3.)
6. There are no routing blockages within $BB(t)$. (The effect of blockages will be considered in Section 2.2.)

In our model, the routing probability of $\overline{x,y}$ is a function of x, y, n, and m. The following results define our model. Proofs are given in the Appendix.

THEOREM 1. *The wire density in $BB(t)$ for 2-pin nets due to one-bend paths is:*

$$D_1^t(x,y) = \begin{cases} \frac{p_1}{2} & y = 0, m \\ 0 & 0 < y < m \end{cases} \tag{2}$$

THEOREM 2. *The wire density in $BB(t)$ for 2-pin nets due to two-bend paths is:*

$$D_2^t(x,y) = \begin{cases} p_2 \frac{n-x-1}{n+m-2} & y = 0 \\ p_2 \frac{x-1}{n+m-2} & y = m \\ \frac{p_2}{n+m-2} & 0 < y < m \end{cases} \tag{3}$$

THEOREM 3. *The wire density in $BB(t)$ for 2-pin nets due to three-bend paths is:*

$$D_3^t(x,y) = \begin{cases} \frac{p_3}{2} \frac{n-x-1}{n-1} & y = 0 \\ \frac{p_3}{2} \frac{x-1}{n-1} & y = m \\ \frac{p_3}{2(m-1)} & 0 < y < m \end{cases} \tag{4}$$

THEOREM 4. *The wire density in $BB(t)$ for 2-pin nets due to four-bend paths is:*

$$D_4^t(x,y) = \begin{cases} p_4 \frac{n-x-1}{n-1} \frac{n-x-2}{n+m-4} & y = 0 \\ p_4 \frac{x-1}{n-1} \frac{x-2}{n+m-4} & y = m \\ 2p_4 \frac{(n-x-1)(m+x-y-2)+(x-1)(y-1)}{(m-1)(n-1)(n+m-4)} & 0 < y < m \end{cases} \tag{5}$$

For paths with more than four bends, we approximately consider them as four-bend paths:

$$D_{4+}^t(x,y) = \frac{p_4^{'}}{p_4} D_4^t(x,y) \quad p_4^{'} = 1 - p_1 - p_2 - p_3$$

As shown in Figure 1, each path with more than four bends can be obtained from a four-bend path by moving some line segments from the line $y = 0$ or $y = m$ into $BB(t)$. It is obvious that the line segments near $x = a$ are more likely to move and that they can not move beyond the line $y = q$. Therefore, the effect of the moving is to increase the wire densities of the line segments near the point $(\frac{n}{2}, 0)$ and $(\frac{n}{2}, m)$. The decrease of wire density is negligible at the two edges $y = 0$ and $y = m$ where the wire density due to paths with more than four bends is relatively low. We propose the empirical formula in Equation (6) for the wire density of paths with four or more bends, where $W(x,y) = w(x(n-x) - w(n-y))$ and w is a *multi-bend factor*.

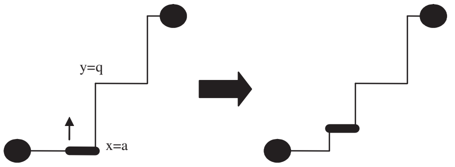


Figure 1: Example of transforming a four-bend path to a six-bend path.

$$D_{w+}^t(x,y) = \begin{cases} p_4^{'} \frac{n-x-1}{n-1} \frac{n-x-2}{n+m-4} & y=0 \\ p_4^{'} \frac{x-1}{n-1} \frac{x-2}{n+m-4} & y=m \\ p_4^{'} \frac{(n-x-1)(m+x-y-2)+(x-1)(y-1)+W(x,y)}{(m-1)(n-1)(n+m-4)/2} & 0<y<m \end{cases} \tag{6}$$

Finally, $D^t(x,y)$ can be obtained by:
$D^t(x,y) = D_1^t(x,y) + D_2^t(x,y) + D_3^t(x,y) + D_{w+}^t(x,y)$

2.2 Blockage Effect Model

The wire density map will change with blockages. Suppose among the points in the blockage whose y coordinates are the smallest, (x_0, y_0) is the rightmost point. We denote the line segment from $(x_0, 0)$ to (x_0, y_0) as $L1$. Let S be the set of line segments $\overline{x_i, y_j}$ which satisfy $(x_i, y_j) \in L1$. The wire density function for the line segments on the left side of $L1$ is as given by Theorem 5, and the wire density function on the right side of $L1$ can be similarly obtained.

THEOREM 5. *Under the assumption that the wire densities for the line segments in S are proportional to their original wire densities without the blockage,*

$$D^t(x,y) = \sum_{x_i,y_j \in S} \left(\frac{D^o(x_i,y_j)}{\sum\limits_{x_i,y_j \in S} D^o(x_i,y_j)} D^{t'}(x,y) \right) \tag{7}$$

where $D^o(x_i, y_j)$ is the original wire density of the vertical line segment $\overline{x_i, y_j}$ without the blockage, and $D^{t'}(x,y)$ is the wire density of the line segment $\overline{x,y}$ for the pseudo-net t' whose two pins are located at $(0,0)$ and (x_i, y_j). ($BB(t')$ is the region B shown in Figure 2(a))

PROOF. Since any possible routing path of t without detouring must pass exactly one line segment in S, the sum of the wire densities of all the line segments in S should be 1. Therefore, according to the assumption, the wire density for the line segment $\overline{x_i, y_j}$ is $\frac{D^o(x_i,y_j)}{\sum_{\overline{x_i,y_j} \in S} D^o(x_i,y_j)}$. For any line segment $\overline{x, y}$, the probability of a path which passes both $\overline{x, y}$ and $\overline{x_i, y_j}$ is equal to the wire density of $\overline{x_i, y_j}$ multiplied by the probability of $\overline{x, y}$ being passed in the pseudo net t'. Then $D^t(x, y)$ is obtained by adding the probabilities of passing both $\overline{x, y}$ and $\overline{x_i, y_j}$ over all $\overline{x_i, y_j} \in S$. $\square$

This updating rule can be easily extended to multiple blockages. For example, in Figure 2(b), we can update the wire density according to the lines $L1, L2$, and $L3$ in sequence.

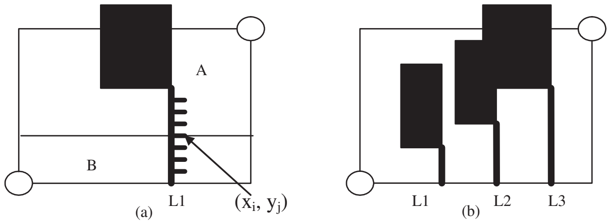


Figure 2: Examples of wire density function change due to blockages (opaque regions in figures).

2.3 Testing the Model

To verify the models, we have written a simple code to collect data from routed designs and compare it with the predicted results. The input LEF/DEF files include:

1. core layout area dimensions($X_{max}, X_{min}, Y_{max}, Y_{min}$);
2. pin locations ($x_{max}, x_{min}, y_{max}, y_{min}$); and
3. segment information.

The code normalizes the bounding boxes of all nets to a 20×20 mesh. Every unit-length grid segment in the big mesh in the horizontal or vertical routing direction serves as a bin, hence there are $20 \times 20 \times 2$ bins. Multi-pin nets are decomposed to 2-point connections. When a unit grid is passed by a routed path, we increase the count of that bin by one. We run a C++ code based on the open-source *UCLA PD Tools release* available on the web [8], and collect data from five industry testcases routed by a commercial router, Cadence WarpRoute v1.0.22. We collect the observations in the 20 bins at $y = 0, 2, ..., 20$ in the horizontal layer of the normalized mesh. Then we polynomial-fit the observations with linear or order-2 polynomials to obtain "actual functions". The predicted functions are obtained by setting $n = m = 20$, $w = 8$, and $y = 0, 2, ..., 20$ in our model for $N = 7500$ nets. The predicted and actual functions for testcase A are listed in Table 4. The predicted values are compared with observations and the values estimated by the model in Lou et al. [5]. As shown in Figure 3, the new model fits the actual observations much better. The goodness-of-fit of the proposed model is also tested by the statistical criterion, $R^2 \equiv 1 - \frac{\sum_{i=0}^{20} (V_i^p - V_i^r)^2}{\sum_{i=0}^{20} (V_i^p - \overline{V^r})^2}$, where V_i^p are the values predicted by our model, V_i^r are the actual observations, and $\overline{V^r}$ is the mean of all the observations [12]. R^2 values shown in Table 4 are all quite close to 1.

y	Actual Function	Predicted function	R^2
0	$3146 - 108x$	$3000 - 105x$	0.933
2	$205 + 13x - x^2$	$113 + 16.4x - 0.9x^2$	0.901
4	$115 + 22x - 1.2x^2$	$86 + 16.8x - 0.9x^2$	0.926
6	$144 + 18x - 1.2x^2$	$66 + 17.2x - 0.9x^2$	0.875
8	$48 + 16x - 0.9x^2$	$52 + 17.6x - 0.9x^2$	0.941
10	$17 + 14x - 0.7x^2$	$45 + 18x - 0.9x^2$	0.881
12	$33 + 16x - 0.8x^2$	$44 + 18.4x - 0.9x^2$	0.902
14	$73 + 24x - x^2$	$50 + 18.8x - 0.9x^2$	0.897
16	$41 + 20x - 0.74x^2$	$62 + 19.2x - 0.9x^2$	0.911
18	$85 + 26.5x - 0.9x^2$	$80 + 19.6x - 0.9x^2$	0.951
20	$734 + 102x$	$900 + 105x$	0.944

Table 4: Comparison of actual functions versus predicted functions inside the bounding box when $w = 8, N = 7500$ for testcase A.

3. Pseudo-Constructive Wirelength and Congestion Estimation

A fast way to estimate detouring is to assume that all nets $t \in T$ have the same wire density distribution outside $BB(t)$. Intuitively, $D^t(x, y)$ will be small when $\overline{x, y}$ is far from $BB(t)$. We propose the following empirical formula for the wire density function outside the bounding box, where $\overline{x_0, y_0}$ is the nearest line segment in $BB(t)$.

$$D^t(x, y) = \frac{D^t(x_0, y_0)}{Manhattan\ Distance((x, y), (x_0, y_0))} \tag{8}$$

Obviously, this estimation is very crude. In order to accurately estimate the effects of detoured paths (that is, the paths whose length are greater than RSMT wirelenth), we need to answer two questions: (1) Which nets will detour? (2) What will be the detoured wirelength? In Section 3.1, we investigate the relationship between congestion and detoured wirelength in order to solve the first problem. Then, in Section 3.2, we propose a pseudo-constructive estimation algorithm to estimate the detoured wirelength.

3.1 The Relationship Between Congestion and Detoured Wirelength

Figure 4 compares the actual detouring map with the actual congestion map of testcase A after detailed routing. The actual detailed routing results are generated by the Cadence WarpRoute v1.0.22 router. In the maps, the layout is divided into $N_r = 200 \times 200$ equally sized rectangular regions r_i ($i = 1..N_r$). In the actual congestion map, the value of a region is the total length of all routed tracks in the region while in the detouring map, we only count the tracks routed by detoured nets. Regions with darker colors are more congested.

As shown in Figure 4, the detoured paths occur roughly around congested regions. Therefore, we propose to use a congestion-related variable to indicate whether a net will detour.

DEFINITION 2. *The estimated congestion of the line segment $\overline{x, y}$ in the layout is $C(x, y) \equiv \sum_{t \in T} D^t(x, y)$, where T is the set of all nets in the netlist.*

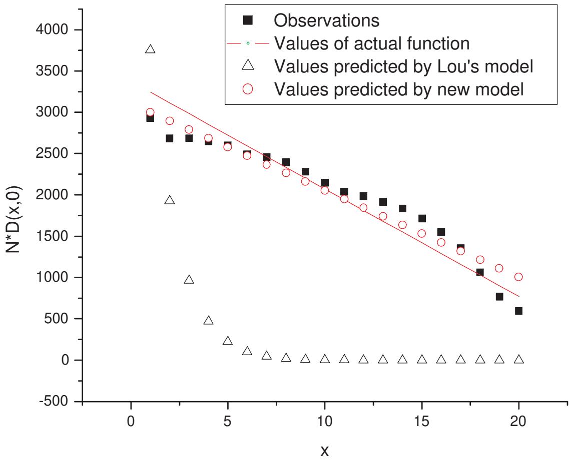


Figure 3: For the line $y = 0$, fitted actual function and values predicted by the new model, compared with the estimation by Lou's model and observations of the 20 bins.

Definition 3. *$C'(x,y,t) \equiv C(x,y) - D^t(x,y)$ represents the sum of wire density of all nets in the netlist except for the net t on the line segment $\overline{x,y}$.*

Definition 4.

$$Congestion\_Factor(t) \equiv \frac{\sum_{\overline{x,y}\in BB(t)} C'(x,y,t)D^t(x,y)}{\sum_{\overline{x,y}\in BB(t)} D^t(x,y)}$$

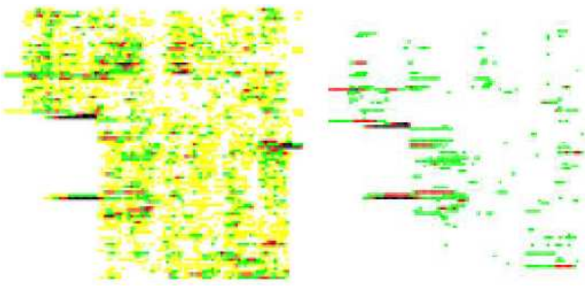

(a) (b)

Figure 4: Actual congestion map (a) and actual detouring map (b) for testcase A.

Congestion_Factor(t) is a "weighted mean" of $C'(x,y,t)$. The "weight" for every line segment $\overline{x,y}$ is $D^t(x,y)$, which can be viewed as the contribution of the line segment to the net. A large value of $D^t(x,y)$ means that the line segment $\overline{x,y}$ has a great probablity to be used in the route and that the congestion context of $\overline{x,y}$ has great impact on the net's routability. Table 5 shows the relationship between *Congestion_Factor* and detoured wirelength for testcase A, where the unit of detoured length is 1 micron. The results suggest that *Congestion_Factor* is reasonably well correlated with whether a net will detour.[2]

Detour Wirelength	# nets	Avg. Congestion_Factor
0	27627	0.36195
[0, 10]	2575	0.930869
[10, 20]	52	1.95588
[20, 30]	22	1.29852
[30, 40]	11	0.780612
[40, 50]	14	1.13645
[50, 60]	4	1.17296
[60, 300]	19	1.65263

Table 5: Relationship between *Congestion_Factor* and detoured wirelength.

3.2 Congestion_Factor Based Wirelength Estimation Algorithm

The basic idea of our "pseudo-constructive" estimation method is that if $BB(t)$ is very congested, for example, *Congestion_Factor*$(t) > \alpha$, there will be detouring. Here, α is some given value which, based on the experimental results in Table 5, we set as $\alpha = 0.6$ for our experiments. The role of detouring is actually to expand $BB(t)$ in order to include some less congested regions. According to our proposed algorithm, when $BB(t)$ expands, the wire density inside the original bounding box will drop and the wire density of the added regions will increase. As a result, *Congestion_Factor*(t) will be reduced. This process will end when *Congestion_Factor* drops below α for all nets. The four directions in which $BB(t)$ can expand are: $x > n, x < 0, y < 0$, and $y > m$. The paths which are extended in two directions can be viewed as being extended in one direction twice. Therefore, we only need to consider the case that the path is extended in one direction. For the paths extended in one direction, we can create an 1-to-1 mapping between the paths with detoured length $2l$ and the paths in an $(n+2l) \times m$ grid which must pass the line $x = n + l$, as shown in Figure 5. Theorem 6 expresses how the wire density function varies with the detoured length $2l$, where $p_{d,b}$ represents the probability of all b-bend detoured paths. The proof is given in the Appendix.

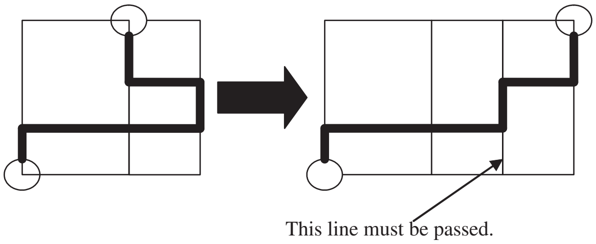


Figure 5: Mapping between the paths with the detoured length $2l$ and the paths in an $(n+2l) \times m$ grid with the line $x = n + l$ must be passed.

Theorem 6. *Under the assumption that all paths with more than four bends can be approximately considered as four-bend paths, the wire density function $D^t(x,y,l)$ for the*

---

[2] Ongoing work is evaluating other correlations, e.g., to relative rather than absolute detoured wirelength.

```
Congestion_Factor Based Estimation Algorithm
Input:
  Placed netlist with fixed pin location
  Core layout area dimensions
Output:
  Total wirelength
  Congestion map stored in a two-dimensional array
Algorithm:
For each net in the layout
    For each line segment
        calculate wire density and update its congestion
For each net $t$ in the layout
  { calculate $Congestion\_Factor(t)$
    put $t$ into a heap wherein the top element
      always has maximum $Congestion\_Factor()$
  }
While (the maximum of $Congestion\_Factor() > \alpha$)
  { expand $BB(t)$ by 1 grid in the least congested direction
    recalculate the $Congestion\_Factor()$
    put the net back into the heap and update related congestion
  }
```

Figure 6: Proposed Congestion_Factor algorithm.

detoured paths which expand only in the direction $x > n$, and whose detoured length is $2l$, is:

$$\begin{cases} p_{d,2} + \frac{p_{d,3}}{2} + \frac{2n+4l-2x-4}{2n+4l+m-6} p^{'}_{d,4} & y=0 \quad x\in[0,n+l] \\ p_{d,2} + \frac{p_{d,3}}{2} + \frac{4n+4l-2x-2}{2n+4l+m-6} p^{'}_{d,4} & y=m \quad x\in[n,n+l] \\ \frac{p_{d,3}}{2(m-1)} + \frac{2m+2x-2y-4}{(m-1)(2n+4l+m-6)} p^{'}_{d,4} & 0<y<m \quad x\in[0,n] \\ \frac{p_{d,3}}{(m-1)} + \frac{2m+4x-2n-8}{(m-1)(2n+4l+m-6)} p^{'}_{d,4} & 0<y<m \quad x\in[n,n+l] \end{cases} \tag{9}$$

where

$$p_{d,2} = \frac{\frac{p_2}{n+2l+m-2}}{\frac{p_2}{n+2l+m-2} + \frac{p_3}{n+2l-1} + \frac{p^{'}_4(2n+4l+m-6)}{(n+2l-1)(n+2l+m-4)}}$$

$$p_{d,3} = \frac{\frac{p_3}{n+2l-1}}{\frac{p_2}{n+2l+m-2} + \frac{p_3}{n+2l-1} + \frac{p^{'}_4(2n+4l+m-6)}{(n+2l-1)(n+2l+m-4)}}$$

$$p^{'}_{d,4} = 1 - p_{d,2} - p_{d,3}$$

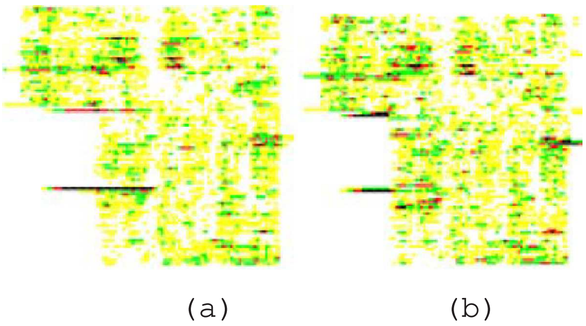
The wire density functions for the detoured paths which extend in the other three directions can be analogously derived. Our proposed Congestion_Factor Based Wirelength and Congestion Estimation Algorithm is given in Figure 6.

Time Complexity

When the detoured wirelength is increased from $2(l-1)$ to $2l$, we only need to calculate $D^t(n+l, y, l)$ for $0 \leq y \leq m$ and recalculate the coefficients. Therefore, the time to calculate the *Congestion_Factor* is at most $O(m)$. The time complexity of the Congestion_Factor based estimation algorithm is at most $O(M(E+M)N)$; where M is the maximum half-perimeter among all nets, E is the maximum detoured length and N is the number of nets. In practice, M and E are each bounded by a constant. For typical testcases, we find that only about 10% of the nets need to expand and only 3% need more than 3 expansions. The time efficiency of the *Congestion_Factor* based algorithm is confirmed by the results in Table 7.

3.3 Pseudo-Constructive Wirelength Estimation Experiments

We implemented the proposed algorithms as a C++ code and ran them over several testcases on an Intel Xeon 2.4GHz CPU. The code reads LEF/DEF files which include the netlist and placed pin locations. The outputs include the congestion map and the total wirelength for each algorithm. Wirelength estimation results in Table 6 suggest that the *Congestion_Factor* based algorithm can greatly improve estimation quality.



(a) (b)

Figure 7: Estimated Congestion Maps from Equation (8) (a) and from the Congestion_Factor based algorithm (b) for testcase A.

Testcase	RSMT	Actual	Congestion_Factor based		
			Est. WL	Improve	CPU
A	4.249	4.389	4.381	94%	15.69
B	3.214	3.317	3.307	78%	7.64
C	5.345	5.507	5.517	97%	3.37
D	2.190	2.372	2.368	93%	23.51
E	6.673	6.821	6.808	89%	7.91

Table 6: Quality of *Congestion_Factor* based wirelength estimation:
$Improve = 1 - \frac{|Actual\_Wirelength - Estimated\_Wirelength|}{|Actual\_Wirelength - RSMT\_Wirelength|}$.

To output the predicted congestion map, the layout is divided into $N_r = 200 \times 200$ equally sized rectangular regions r_i $(i = 1...N_r)$ and $\sum\limits_{x,y \in r_i} C(x,y)$ is the estimated congestion in the region r_i. The congestion maps generated by the empirical formula, Equation (8), and the Congestion_Factor based algorithm are shown in Figure 7. We also quantitatively compare the regional congestion estimates using the following metrics reported in [7].

- Mean of regional errors: $\mu \equiv \frac{\sum_{i=1}^{N_r}\left(\frac{\sum_{x,y\in r_i} C(x,y)}{\sum_{x,y\in r_i} \delta(x,y)}\right)}{N_r}$, where $\delta(x,y) = \begin{cases} 1 & if\ routed \\ 0 & if\ not\ routed \end{cases}$ and $\sum\limits_{x,y\in r_i} \delta(x,y)$ is the actual congestion value
- Standard deviation of the errors: $\sigma \equiv \sqrt{\frac{\sum_{i=1}^{N_r}\left(\frac{\sum_{x,y\in r_i} C(x,y)}{\sum_{x,y\in r_i} \delta(x,y)} - 1\right)^2}{N_r - 1}}$ and
- Runtime.

The ideal values for μ and σ are 1 and 0 respectively. In Table 7, "Lou" is the algorithm proposed in Lou et al. [5]; "EF" is the empirical formula of Equation (8); and "CF" is the Congestion_Factor based algorithm. The results confirm that our methods can greatly improve the congestion estimation accuracy.

Test case	Lou			New model+EF			New model+CF		
	μ	σ	CPU	μ	σ	CPU	μ	σ	CPU
A	0.903	1.227	15.24	0.951	1.134	6.89	0.963	0.689	15.69
B	0.878	1.352	7.56	1.037	0.621	3.12	1.031	0.566	7.64
C	0.911	1.124	4.67	0.962	0.721	1.36	0.991	0.707	3.37
D	0.923	0.946	22.65	0.981	0.545	9.27	0.987	0.323	23.51
E	1.235	1.672	8.13	1.102	1.024	3.28	1.057	0.813	7.91

Table 7: Evaluate estimation quality by actual router.

4. Conclusions and Future Work

We have developed new, accurate wirelength and congestion estimation methods appropriate throughout a top-down placement process. Our methods can give accurate total wirelength estimation with linear runtime complexity. Our contributions include: (i) developing a new practical wire density model that takes effects of bends and blockages into account; and (ii) avoiding the pitfalls of producing "one-shot" congestion maps, and instead developing iterative congestion map constructions to accurately predict detouring.

We have validated our new methods on several industry testcases. The results show that our methods improve the wirelength estimation accuracy - in the sense of reducing estimation error - by 90% on average with respect to the traditional RSMT wirelength estimate. We also produce more accurate congestion maps, while using roughly the same runtime, in comparison to the algorithm reported in [5]. Our ongoing research focuses on including our new estimator into a placer in order to reduce routing congestion.

5. Acknowledgments

We thank Dr. Bao Liu for writing the code for collecting data from routed designs.

6. REFERENCES

[1] M. Wang, X. Yang and M. Sarrafzadeh, "Congestion Minimization During Placement", *IEEE Transactions on CAD*, 19(10), 2000, pp. 1140-1148.

[2] Kusnadi and J. D. Carothers, "A Method of Measuring Nets Routability for MCMs General Area Routing Problem", *Proc. ACM/IEEE Intl. Symp. on Physical Design*, 1999, pp. 186-192.

[3] W. Huang and A. B. Kahng, "A Layout Advisor for Timing-Critical Bus Routing", *Proc. Tenth Annual IEEE International ASIC Conf. and Exhibit*, 1997, pp. 210-214.

[4] C.-L. E. Cheng, "Risa: Accurate and Efficient Placement Routability Modeling", *Proc. IEEE-ACM Intl. Conf. on Computer-Aided Design*, 1994, pp. 690-695.

[5] J. Lou, S. Krishnamoorthy and H. S. Sheng, "Estimating Routing Congestion Using Probabilistic Analysis", *Proc. ACM/IEEE Intl. Symp. on Physical Design*, 2001, pp. 112-117.

[6] A. E. Caldwell, A. B. Kahng, S. Mantik, I. L. Markov and A. Zelikovsky, "On Wirelength Estimations for Row-Based Placement", *Proc. ACM/IEEE Intl. Symp. on Physical Design*, 1998, pp. 4-11.

[7] P. Kannan, S. Balachandran and D. Bhatia, "On Metrics for Comparing Routability Estimation Methods for FPGA", *Proc. ACM/IEEE Design Automation Conf.*, 2002, pp. 70-75.

[8] http://nexus6.cs.ucla.edu/GSRC/bookshelf/Slots/Placement/Capo/.

[9] S. L. Teig, "Challenges and Principles of Physical Design", *Proc. ACM/IEEE Intl. Symp. on Physical Design*, 2002, pp. 3-4.

[10] X. Yang, R. Kastner and M. Sarrafzadeh, "Congestion Estimation During Top-down Placement", *IEEE Transactions on CAD*, 21(1), 2002, pp. 72-80.

[11] U. Brenner and A. Rohe, "An Effective Congestion Driven Placement Framework", *Proc. ACM/IEEE Intl. Symp. on Physical Design*, 2002, pp. 6-11.

[12] http://www.ssicentral.com/lisrel/column3.htm.

APPENDIX

A. Proofs of Theorems

Based on the assumption that paths with the same number of bends have the same routing probability, the wire density of the line segment $\overline{x,y}$ due to b-bend paths is: $\frac{\#\ b-bend\ Paths\ passing\ through\ \overline{x,y}}{Total\ \#\ b-bend\ Paths} p_b$. Therefore, the following proofs are all based on counting the number of possible b-bend paths which pass through $\overline{x,y}$.

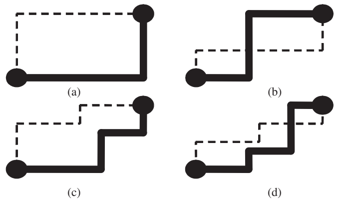


Figure 8: Paths with one (a)**, two** (b)**, three** (c) **and four** (d) **bends.**

We divide all the paths into two categories. If the bottom pin belongs to the horizontal line segment of the path, the path is called an *H-path*; otherwise, the path is called a *V-path*. In Figure 8, all H-paths are marked with bold lines while V-paths are marked with dashed lines.

There are $n-1$ vertical lines: $x=1, x=2,..., x=n-1$ and $m-1$ horizontal lines: $y=1, y=2..., y=m-1$ inside the $n \times m$ grid. It is obvious that if an H-path passes i vertical lines and j horizontal lines, then $j \le i \le j+1$.

LEMMA 1. The number of all possible H-paths (or V-paths) which pass i vertical lines and j horizontal lines is: $\left(\begin{array}{l} n-1 \\ i \end{array}\right)\left(\begin{array}{l} m-1 \\ j \end{array}\right)$

PROOF. The number of ways to choose i different integers from the set $\{1...n-1\}$ and choose j different integers from the set $\{1...m-1\}$ is $\left(\begin{array}{l} n-1 \\ i \end{array}\right)\left(\begin{array}{l} m-1 \\ j \end{array}\right)$, where the values of i, j satisfy the conditions of Lemma 2. For every H-path which passes i vertical lines and j horizontal lines, the x coordinates for all vertical lines are i different integers from the set $\{1...n-1\}$ and the y coordinates for all horizontal lines are j different integers from the set $\{1...m-1\}$. On the other hand, for i different integers from the set $\{1...n-1\}$ and j different integers from the set $\{1...m-1\}$, if the values of i, j satisfy $j \le i \le j+1$, we can order them to form two ascending sequences: $x_1...x_i$ and $y_1...y_j$. Then the H-path is uniquely defined : $(0,0) \rightarrow (x_1,0)$, $(x_1,0) \rightarrow (x_1,y_1)...,(n,y_j) \rightarrow (n,m)$ (if $i=j$) or $(x_i,m) \rightarrow (n,m)$ (if $i=j+1$). Therefore, there is a 1-to-1 mapping between these two sets. The proof for V-paths is similar. $\square$

LEMMA 2. The number of b-bend paths which pass through the horizontal line segment $\overline{x,y}$ is:

$$\sum_{b_1=0}^{b} NP(x,y,b_1)NP_H(n-x,m-y,b-b_1)$$

PROOF. For b-bend paths, the number of bends in the bottom grid defined by $(0,0)$ and (x,y), b_1, can be $0\ldots b$. The number of paths which have b_1 bends in the bottom grid is equal to the number of b_1-bend paths from the point $(0,0)$ to the point (x,y) multiplied by the number of $(b-b_1)$-bend H-paths from the point (x,y) to the point (n,m). □

Proof of Theorem 1:

PROOF. As shown in Figure 8 (a), the line segments on $y=0$ and $y=m$ are passed by exactly one path and the line segments inside $BB(t)$ are not passed. □

Proof of Theorem 2:

PROOF. For two-bend paths, each H-path passes 1 vertical line inside $BB(t)$; each V-path passes 1 horizontal line. According to Lemma 1, there are $(n-1)+(m-1)$ two-bend paths. The line segments inside $BB(t)$ are passed by exactly one V-path. The number of H-paths which pass the line segment $\overline{x,0}$ is $n-x-1$ according to Lemma 2. Therefore, the wire density of $\overline{x,0}$ is $\frac{n-x-1}{n+m-2}p_2$. The wire density function on the line $y=m$ can be derived similarly. □

Proof of Theorem 3:

PROOF. There are $2(n-1)(m-1)$ three-bend paths. For the line segment $\overline{x,0}$ on the lower edge, $NP(x,0,0)=1$ and $NP_H(n-x,y,3)=(n-x-1)(m-1)$ when $b_1=0$. The line segment $\overline{x,y}$ in the middle of $BB(t)$ is passed by $x-1$ H-paths and by $n-x$ V-paths when $b_1=1$ and 2, respectively. □

Proof of Theorem 4:

PROOF. For four-bend paths, there are $\begin{pmatrix} n-1 \\ 2 \end{pmatrix}\begin{pmatrix} m-1 \\ 1 \end{pmatrix}$ H-paths and $\begin{pmatrix} n-1 \\ 1 \end{pmatrix}\begin{pmatrix} m-1 \\ 2 \end{pmatrix}$ V-paths according to Lemma 3. Among them, there are $\begin{pmatrix} n-x-1 \\ 2 \end{pmatrix}\begin{pmatrix} m-1 \\ 1 \end{pmatrix}$ H-paths passing through the line segment $\overline{x,0}$ with $b_1=0$. For the line segment $\overline{x,y}$ in the middle of $BB(t)$, the values of $NP(x,y,b_1)NP_H(n-x,m-y,b-b_1)$ are $(n-x-1)(m-y-1)$, $(x-1)(n-x)$ and $(x-1)(y-1)$ when $b_1=1,2$ and 3 respectively. □

LEMMA 3. There is a 1-to-1 mapping between the paths with the detoured length $2l$ and the paths in an $(n+2l)\times m$ grid which pass the line $x=n+l$.

PROOF. As shown in Figure 5, we can perform the following transformation: Suppose the highest point of the path in the line $x=n+l$ is $(n+l,y_0)$, then

$$\begin{cases} \overline{x,y} \rightarrow \overline{2n+2l-x,y} & if\ y > y_0 \\ \overline{x,y} \rightarrow \overline{x,y} & if\ y \leq y_0 \end{cases}$$

This transformation is obviously 1-to-1. □

LEMMA 4. $NP(n,m,b:l) = NP(n,m,b) - NP(n-1,m,b)$

PROOF. All the paths in the grid either pass or do not pass the line $x=l$. We can view those paths which do not pass the line as being routed in an $(n-1)\times m$ grid by removing the line $x=l$ from the grid. □

Proof of Theorem 6:

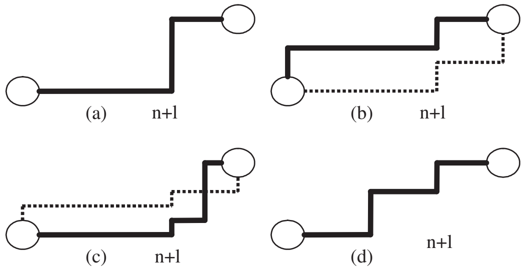


Figure 9: Paths with two (a), three (b) and four (c,d) bends which pass the line $x=n+l$.

PROOF. We assume that the probabilities of b-bend detoured paths is proportional to their probabilities in the $(n+2l)\times m$ grid. The probability of two-bend paths in the $(n+2l)\times m$ grid which pass the line $x=n+l$ is: $\frac{\#\ of\ two-bend\ paths\ passing\ x=n+l}{\#\ of\ total\ two-bend\ paths}p_2 = \frac{p_2}{n+2l+m-2}$. Similarly, we can get the probabilities of three-bend detoured paths and four-bend detoured paths. The bends distribution for detoured paths is:

$$p_{d,2} = \frac{\frac{p_2}{n+2l+m-2}}{\frac{p_2}{n+2l+m-2}+\frac{p_3}{n+2l-1}+\frac{p_4^{'}(2n+4l+m-6)}{(n+2l-1)(n+2l+m-4)}}$$

$$p_{d,3} = \frac{\frac{p_3}{n+2l-1}}{\frac{p_2}{n+2l+m-2}+\frac{p_3}{n+2l-1}+\frac{p_4^{'}(2n+4l+m-6)}{(n+2l-1)(n+2l+m-4)}}$$

$$p_{d,4}^{'} = 1 - p_{d,2} - p_{d,3}$$

The wire density function due to two-bend paths is:

$$D_2^t(x,y,l) = \begin{cases} p_{d,2} & y=0 & x\in[0,n+l] \\ p_{d,2} & y=m & x\in[n+l,n+2l] \end{cases}$$

There are $2(m-1)$ three-bend paths as shown in Figure 9(b). The wire density function due to three-bend paths is:

$$D_3^t(x,y,l) = \begin{cases} \frac{p_{d,3}}{2} & y=0 & x\in[0,n+l] \\ \frac{p_{d,3}}{2} & y=m & x\in[n+l,n+2l] \\ \frac{p_{d,3}}{2(m-1)} & 0<y<m & x\in[0,n+2l] \end{cases}$$

We distinguish three kinds of four-bend paths:

1. Paths with zero bends in the region $x<n+l$, such as the path marked with bold lines in Figure 9(c); the number of these is $(m-1)(l-1)$
2. Paths with one bend in the region $x<n+l$, such as the path marked with dashed lines in Figure 9(c); the number of these is $(m-1)(0.5m-1)$
3. Paths with two bends in the region $x<n+l$, such as the path in Figure 9(d); the number of these is $(m-1)(n+l-1)$

The wire density function due to four-bend paths is:

$$\begin{cases} \frac{2n+4l-2x-2}{2n+4l+m-6}p_{d,4}^{'} & y=0 & x\in[0,n+l] \\ \frac{2x-2}{2n+4l+m-6}p_{d,4}^{'} & y=m & x\in[n+l,n+2l] \\ \frac{2n+4l-2x+2y-4}{(m-1)(2n+4l+m-6)}p_{d,4}^{'} & 0<y<m & x\in[0,n+l] \\ \frac{2x+2m-2y-4}{(m-1)(2n+4l+m-6)}p_{d,4}^{'} & 0<y<m & x\in[n+l,n+2l] \end{cases}$$

Theorem 6 can be obtained by adding the wire density of $\overline{2n+2l-x,y}$ to the wire density of $\overline{x,y}$ for all line segments $\overline{x,y}$ which satisfy $x\in[n,n+l]$. □