A Combinatorial Congestion Estimation Approach with Generalized Detours

FEI HE
Computer Science and Technology Department
Tsinghua University, Beijing 100084, P.R. China
School of Software
Tsinghua University, Beijing, P.R. China

XIAOYU SONG
Electrical and Computer Engineering Department
Portland State University, Oregon, U.S.A.

MING GU
School of Software, Tsinghua University
Beijing, P.R. China

LERONG CHENG
Electrical Engineering Department
University of California, Los Angeles, CA U.S.A.

GUOWU YANG AND ZHIWEI TANG
Electrical and Computer Engineering Department
Portland State University, Oregon, U.S.A.

JIAGUANG SUN
School of Software
Tsinghua University, Beijing, P.R. China

(Received October 2004; revised and accepted May 2005)

Abstract—Congestion estimation plays an important role in the physical layout of VLSI design. This paper presents a new probabilistic estimation model that improves the previous estimators by relaxing the constraint on detours in a route. The model is more general and realistic for it gives the flexibility for the wires to have wider usage area to bypass the congestion regions and blockages. Given a routing grid and a set of nets to be routed, the model predicts the routing density on each edge of the grid. The routing density provides direct congestion estimation. We compare our estimation results to the actual routing results. Experimental results show the effectiveness of our estimator.

(© 2006 Elsevier Ltd. All rights reserved.)

Keywords—Routing, Congestion estimation, Probabilistic methods.
1. INTRODUCTION

As interconnection delay becomes a dominant factor in integrated circuit designs, an efficient physical design entails accurate estimations of the individual modules for area planning, optimal placement, and routability of interconnections. Estimation of interconnects before layout becomes a crucial issue for a hierarchical design process.

In classifying research in this area, three approaches can be identified: The first approach [1,2] uses empirical models, which come from the analysis of practice and usually are problem specific and problem-size specific. When the problem or the problem size changes, most of the empirical models lose accuracy. The second uses directly the global router to do the estimation, such as [3,4]. The router-based estimation is time consuming. And more importantly, the validity of the router-based estimation result cannot be guaranteed if a different router is used as the final routing tool. A third, and more recent, approach uses probabilistic models. The probabilistic approach is neither problem specific nor problem-size specific. With sizes on the order of millions of cells and connections, a probabilistic approach will play a crucial role in wire congestion estimation.

There has been some work on probabilistic estimation. Gamal [5,6] used stochastic models to estimate the interconnections area for master slice integrated circuits and custom integrated circuits respectively. Lou [7] gives out a fast probabilistic estimation for congestion. However, the estimator in [7] ignores the detour routes and the effect of bends in the route. Cheng [8] improved the method by putting detour and bends into account. Cheng [9] applied this model on 3-D routing. The model in [8] assumes that a route can only have one detour segment, thus it is still not practice. In this paper, we further improve the previous models by relaxing such restriction. Our model is more realistic, for it gives the flexibility for the wires to have more detours to bypass congestion regions and blockages.

The organization of the paper is as follows. Section 2 introduces the estimation model. Section 3 presents our combinational estimation procedure. Section 4 reports some experimental results. Finally, Section 5 concludes the paper.

2. THE ESTIMATION MODEL

Given a routing area, which is consists of a two-dimensional grid. The grid model of a routing area is shown in Figure 1a. In what follows, we consider the routing mesh model, which is the dual graph of the grid model, as shown in Figure 1b. In mesh model, each node represents a grid and the line segment connecting two adjacent nodes represents the line between two adjacent grids. To make our analysis clear, we use the coordinate on the grid.

For the routing mesh model, as shown in Figure 1, we define that a grid point is the intersection of two lines, a unit line is a line connecting two adjacent grid points, the capacity of a unit line is the number of available track in the unit line and the density of a unit line is the number of routes passing through the unit line.

Our problem is: Given a set of nets $M$ (terminals of the nets are located in the grid points), estimate the density of the each unit line in the routing area. In our analysis, the net is routed using more detours. As a result, our model is more realistic than [7,8], for it gives more flexibility for the wires to bypass congestion regions and blockages. In what follows, we restrict our discussion on two-terminal nets. Our model can be extended to handle multi-terminal cases by using rectilinear Steiner tree or minimum spanning tree, as discussed in [7].

Without loss of generality, we assume that the terminals are located at the lower left and upper right grids. The lower left terminal is the start terminal, and the upper right terminal is the end terminal of the net. For example the net shown in Figure 1b, its start terminal is $(0,0)$ and its end terminal is $(m,n)$. The direction of the route of a net is from the start terminal to the end terminal. As noted above, we do not restrict the net route with the shortest length, so routes may not be monotonic. A forward segment is a route segment that goes continuously up or right.
A reverse segment is a route segment that goes continuously down or left. The length of the route is the number of unit lines the route covers. Figure 2 illustrates a nonmonotonic path and monotonic path.

If a route contains reverse segments, it will increase the total wire length, thus increasing delay. In our estimation, we make the following assumptions for each route.

**Assumption 1.** The reverse segments are all vertical (horizontal).

Under this assumption, a vertical reverse segment (going down) and a horizontal reverse segment (going left) cannot occur in the same route. This is practical because in the real layout, the route rarely goes both left and down.

**Assumption 2.** The total length of the reverse segments is no more than $d$.

Under this assumption, we only consider the route whose reverse segment length is no more than $d$, we make this assumption because in practice, a very long route will hardly be used.

We can recognize the advantage of our model in practicability over the models in [7,8] by showing the usage area for a net respectively. As shown in Figure 3, given a net, the usage area for the model in [7] which limits the net to use the shortest route is only the rectangle $A$; the usage area for the model in [8] which allows the net to use one detour (whose length is $l$) and some bends is the rectangle $A$ plus the rectangles $B1$, $B2$, $B3$, and $B4$; however, the usage area for our model (the total length of reverses is $l$) is the whole of the octagon which is composed
of the rectangles A, B1, B2, B3, B4, and the triangles C1, C2, C3, C4. Obviously, as soon as we give the length bound of the reverse segments a sufficient large value, then the usage area for any net would cover all over the routing area. As a result, our model covers all routes of a net in theoretical meaning. The usage area of our model can be represented as,

\[ (-l \leq x \leq m + l) \land (-l \leq y \leq n + l) \land (x + y \leq m + n + l) \land (x + y \geq -l) \land (x - y \leq m + l). \]

In Sections 3 and 4, we will use the following notations for our analysis.

1. \( S_{\alpha} \) and \( E_{\alpha} \) are the start point and end point of the route \( \alpha \), respectively. For instance, for the route connecting \((0,0)\) and \((m,n)\), we have \( S_{\alpha} = (0,0), E_{\alpha} = (m,n) \).
2. \( L(\alpha) \) is the length of the route \( \alpha \).
3. \( R(\alpha) \) is the total length of the reverse segments in the route \( \alpha \).
4. \( D(\alpha) \) is the direction of the reverse segments in the route: when the reverse segments of the route are vertical, \( D(\alpha) = 1 \); when the reverse segments of the route is horizontal, \( D(\alpha) = -1 \); and when the route is monotonic (no reverse segments in the route), the value of \( D(\alpha) = 0 \).
5. \( B(\alpha) \) is the number of bends in the route \( \alpha \).
6. \( V_{x,y} \) represents the vertical unit line connecting the point \((x,y)\) and \((x, y-1)\).
7. \( H_{x,y} \) represents the horizontal unit line connecting the point \((x,y)\) and \((x-1, y)\).
8. \( U_{\alpha} = \{ u_{\alpha}^1, u_{\alpha}^2, u_{\alpha}^3, \ldots, u_{\alpha}^{L(\alpha)} \} \) is the set of unit lines the route \( \alpha \) covers, where \( u_{\alpha}^1 \) is the unit line connecting the start point, \( u_{\alpha}^4 \) is the unit line connecting \( u_{\alpha}^{i-1} \), \ldots and \( u_{\alpha}^{L(\alpha)} \) is the unit line connecting the end point.
9. \( \text{sgn}(x) \) is the sign function of \( x \) satisfying,

\[ \text{sgn}(x) = \begin{cases} -1, & x < 0, \\ 0, & x = 0, \\ 1, & x > 0. \end{cases} \]

10. \( C_a^b \) is the combination function as,

\[ C_a^b = \frac{a!}{b!(a-b)!}, \quad (b < 0) \lor (a < 0) \lor (b > a), \]

otherwise.
In the following, if we don't specially mention, we assume \( \alpha \) is a route whose \( S_\alpha = (0, 0) \), \( E_\alpha = (m, n) \).

## 3. PROBABILISTIC ESTIMATION

In this section, we introduce our probabilistic estimation. The estimation procedure is given as follows. As in [8], we first compute the density of the every unit line under each net and then sum the results to get the total density of every unite line.

**Input:** A set of two-terminal nets \( M = (N_1, N_2, \ldots, N_p) \).

**Output:** Density of routing mesh edges

1. for each \( N_i \), \( i = 1, \ldots, p \).
2. { Determine the usage area of \( N_i \); 
3. Compute the total number of routes for \( N_i \); 
4. Compute the total number of routes crossing a unit line for \( N_i \); 
5. Compute the density of each unit line under \( N_i \); 
6. Add the computed densities to the routing mesh. 
7. } 
8. Output the routing densities.

Since a net can be routed using many different routes, in Step 3 we compute the total number of routes for net \( N_i \). In Step 4, we enforce a unit line must be crossed, and then compute the number of such routes. Obviously, the enforced unit line must be in the usage area of \( N_i \). In Step 5, we compute the probability of each unit line in the usage area to be crossed by net \( N_i \). We treat this probability as the density of corresponding unit line under this net and then accumulate it in Step 6. At last, we output the accumulative densities on all unit lines.

In subsequent contents of this section, we discuss in detail Steps 3–5 in turn.

### 3.1. Total Number of Routes

#### 3.1.1. Monotonic routes

We use \( v(m, n) \) to denote the number of monotonic routes going from \((0, 0)\) to \((m, n)\). We can consider it as a function with variables to be \( m \) and \( n \), obviously \( v(m, n) = |A_1| \), where \( A_1 = \{ \alpha \mid R(\alpha) = 0 \} \).

**Theorem 1.** (See [7].)

\[
v(m, n) = C_{m+n}^{m}
\]

**Proof.** The proof of Theorem 1 is given in [7].

#### 3.1.2. Nonmonotonic routes.

We use \( p_{uv}(m, n, l) \) to denote the number of routes that have following features,

1. (1) it goes from \((0, 0)\) to \((m, n)\);
2. (2) its reverse segments are all vertical;
3. (3) the total length of its reverse segments is \( l \); that is

\[
p_{uv}(m, n, l) = |A_2|, \quad \text{where } A_2 = \{ \alpha \mid R(\alpha) = l, D(\alpha) = l \}.
\]

**Theorem 2.**

\[
p_{uv}(m, n, l) = \begin{cases} 
v(m, n), & l = 0, \\
\sum_{j=1}^{l} C_{m+1}^{l} \times C_{m+n+l-j}^{n+l} \times C_{l-1}^{l-1}, & l > 0. 
\end{cases}
\]

**Proof.**

**Case 2.1.** \( l = 0 \).

The route is monotonic, and it is obvious that the theorem holds.
CASE 2.2. $l > 0$.

In this case, the route is nonmonotonic. We know that each route will cover $m$ horizontal unit lines and $n + 2l$ vertical unit lines ($n + l$ in the forward segments and $l$ in the reverse segments). Here, we use $H$ to denote a horizontal unit line, $V$ to denote a vertical unit line in the forward segments and $R$ to denote a unit line in the reverse segments. A route is corresponding to a combination that $mH, n + lV$, and $lR$ are put into a sequence where $V$ and $R$ cannot border on each other (the vertical forward segments can not be adjacent with the vertical reverse segments). For example, the route illustrated in Figure 4 is corresponding to the sequence $HHVHRHHRHVVHHRHVVV$.

It is clear that counting the number of routes is equivalent to counting the number of sequences. To count the number of sequences, we first observe the $H$. Since $V$ and $R$ cannot border on each other, there must be $H$ between $V$ and $R$.

We first put $mH$ in a sequence, and then insert the $V$ and $R$ in it. Obviously, there are $m + 1$ places to put $V$ and $R$. From those $m + 1$ places, we pick $i$ places to put $V$ and $R$. For example, as the route illustrated above, we first put seven $H$ in a sequence, and then choose five of the eight places to put $V$ and $R$—‘$\diamond H\diamond H\diamond H\diamond H\diamond H\diamond H\diamond H$’, where ‘$\diamond$’ denotes a place and ‘$\bullet$’ denotes a chosen place.

After that, we pick $j$ places from the chosen $i$ places to put $R$ and the other $i - j$ places to put $V$. There are $\binom{i}{j}$ such choices. For the route above, we pick two of the five chosen places to put $R$ and the other three to place $V$. Such as ‘$\diamond H\diamond H\diamond H\diamond H\diamond H\diamond H\diamond H$’, where ‘$\bullet$’ denotes a place to put $R$ and ‘$\bullet$’ denotes a place to put $V$.

Later, we put $lR$ into $j$ places, this means $lR$ are partitioned into $j$ groups orderly. To compute the number of such partitions, let’s imagine the model that we insert $j - 1$ separators into $lR$. Obviously, there are $l - 1$ spaces for the separators. For the above route, 3 $R$ are divided into two groups—‘$2-1$’, such partition is correspondent to insert one separator into two spaces—‘$RR \mid R$’, where ‘$\mid$’ denotes a separator. Therefore, the number of such partitions is $\binom{i-j-1}{j}$.

Finally, we put $n + lV$ into $i - j$ places, similarly, the number of $s$ is $\binom{n+l}{i-j}$.

Moreover, it is obvious that the range of $i, j$ are $2 \leq i \leq m + 1, 1 \leq j \leq i - 1$, respectively. Therefore, the total number of routes is,

$$\sum_{i=2}^{m+1} \sum_{j=1}^{i-1} C^i_{m+1} C^{i-j-1}_j C^{n+l-j}_j = \sum_{j=1}^{l} C^j_{m+1} \times C^n_{m+n+l-j} \times C^{j-1}_{i-1}$$

The derivation of the above equation is shown in Appendix I.

Similarly, we define $p_hr(m,n,l)$ to the number of routes that have following features,

(1) it goes from $(0,0)$ to $(m,n)$;
(2) its reverse segments are all horizontal;
(3) the total length of its reverse segments is \( l \).

Obviously it is symmetric for computing \( p_{hr}(m, n, l) \) to that the reverse segment is vertical, therefore,

\[
p_{hr}(m, n, l) = p_{vr}(n, m, l).
\]

Let’s further consider the routes satisfying not only the constraints corresponding to \( p_{hr}(m, n, l) \) or \( p_{vr}(m, n, l) \), but also an additional constraint: its segment connecting the start point is in a preassigned direction. For example, we consider the routes that satisfy,

1. it goes from \((0, 0)\) to \((m, n)\);
2. its reverse segments are all vertical;
3. the total length of its reverse segments is \( l \);
4. its segment connecting the start point is horizontal forward.

We denote the number of such routes as \( q_{vr\_hf}(m, n, l) \), where the string \( vr \) is to denote that the reverse segments are vertical, and string \( hf \) is to denote the start segment is horizontal and forward. In similar way, we define \( q_{vr\_vf}(m, n, l) \), \( q_{vr\_vr}(m, n, l) \), and \( q_{hr\_vf}(m, n, l) \).

According to the definitions of these four functions, we have,

\[
q_{vr\_hf}(m, n, l) = |A_4|, \quad \text{where} \quad A_4 = \{ \alpha \mid R(\alpha) = l, \ D(\alpha) = \text{sgn}(l), \ u^e_\alpha = H_{1,0} \},
\]

\[
q_{vr\_vf}(m, n, l) = |A_5|, \quad \text{where} \quad A_5 = \{ \alpha \mid R(\alpha) = l, \ D(\alpha) = \text{sgn}(l), \ u^e_\alpha = V_{0,1} \},
\]

\[
q_{vr\_vr}(m, n, l) = |A_6|, \quad \text{where} \quad A_6 = \{ \alpha \mid R(\alpha) = l, \ D(\alpha) = \text{sgn}(l), \ u^e_\alpha = V_{0,0} \},
\]

\[
q_{hr\_vf}(m, n, l) = |A_7|, \quad \text{where} \quad A_7 = \{ \alpha \mid R(\alpha) = l, \ D(\alpha) = -\text{sgn}(l), \ u^e_\alpha = V_{0,1} \}.
\]

Notice that

\[
q_{vr\_hf}(m, n, l) = |A_4| = |A'_4|, \quad \text{where} \quad A'_4 = \{ \alpha \mid R(\alpha) = l, \ D(\alpha) = \text{sgn}(l), \ u^L_\alpha = H_{m,n} \}.
\]

This means \( q_{vr\_hf}(m, n, l) \) can also represent the number of routes satisfying,

1. it goes from \((0, 0)\) to \((m, n)\);
2. its reverse segments are all vertical;
3. the total length of its reverse segments is \( l \);
4. its segment connecting the end point is horizontal forward.

This conclusion is obvious because those two kinds of routes are symmetric to each other. Similarly, we know,

\[
q_{vr\_vf}(m, n, l) = |A'_5|, \quad \text{where} \quad A'_5 = \{ \alpha \mid R(\alpha) = l, \ D(\alpha) = \text{sgn}(l), \ u^L_\alpha = V_{m,n} \}
\]

\[
q_{vr\_vr}(m, n, l) = |A'_6|, \quad \text{where} \quad A'_6 = \{ \alpha \mid R(\alpha) = l, \ D(\alpha) = \text{sgn}(l), \ u^L_\alpha = V_{m,n+1} \}
\]

\[
q_{hr\_vf}(m, n, l) = |A'_7|, \quad \text{where} \quad A'_7 = \{ \alpha \mid R(\alpha) = l, \ D(\alpha) = -\text{sgn}(l), \ u^L_\alpha = V_{m,n} \}.
\]

Similar to the discussion of \( p_{hr}(m, n, l) \) and \( p_{vr}(n, m, l) \), we compute these functions as follows,

\[
q_{vr\_hf}(m, n, l) = \begin{cases} 
  v(m-1,n), & l = 0, \\
  \sum_{j=1}^{l} C^j_m \times C^{n+l-1}_{m+n+l-j-1} \times C^{l-1}_{l-1}, & l > 0,
\end{cases}
\]

\[
q_{vr\_vf}(m, n, l) = \begin{cases} 
  v(m,n-1), & l = 0, \\
  \sum_{j=1}^{l} C^j_m \times C^{n+l-1}_{m+n+l-j-1} \times C^{l-1}_{l-1}, & l > 0,
\end{cases}
\]

\[
q_{vr\_vr}(m, n, l) = \begin{cases} 
  0, & l = 0, \\
  \sum_{j=1}^{l} C^{l-1}_{m} \times C^{n+l}_{m+n+l-j} \times C^{l-1}_{l-1}, & l > 0,
\end{cases}
\]

\[
q_{hr\_vf}(m, n, l) = q_{vr\_hf}(n, m, l).
\]
3.1.3. Total number of routes

Using \( t_n(m, n, l) \) to denote the total number of routes satisfying,

1. it goes from \((0, 0)\) to \((m, n)\);
2. the total length of its reverse segments is \(l\), i.e.,

\[
t_n(m, n, l) = |A_8|,
\]
where \( A_8 = \{\alpha | R(\alpha) = l\} \).

Obviously, there have,

\[
t_n(m, n, l) = p_{vr}(m, n, l) + p_{hr}(m, n, l).
\]

3.2. Number of Routs Crossing a Unit Line

Denote the vertical unit line

\[
(x, y) \rightarrow (x, y - 1): -l \leq x \leq m + l, \quad -l + 1 \leq y \leq n + l,
\]
as \( V_{x,y} \), and the horizontal unit line

\[
(x, y) \rightarrow (x - 1, y): -l + 1 \leq x \leq m + l, \quad -l \leq y \leq n + l,
\]
as \( H_{x,y} \), where \( l \) is the total length of the reverse segments. We consider the number of routes satisfying,

1. it goes from \((0, 0)\) to \((m, n)\);
2. the total length of its reverse segments is \(l\);
3. it passes through \( V_{x,y} \), denoted as \( n_v(x, y, m, n, l) \), i.e.,

\[
n_v(x, y, m, n, l) = |A_9|, \quad \text{where } A_9 = \{\alpha | R(\alpha) = l, \quad V_{x,y} \in U_\alpha \}.
\]

**Theorem 3.**

\[
n_v(x, y, m, n, l) = \begin{cases}
0, & \text{if } (x < l) \lor (x > m + l) \lor (y < 1 - l) \\
\lor (y > n + l) \lor (-x + y > n + l) \\
\lor (x + y < l) \lor (x + y > m + n + l) \\
\lor (x - y \geq m + l), \\
\lor (0 \leq z \leq m) \land (0 < y \leq n) \land (l = 0), \\
v(x, y - 1) \times v(m - x, z - n - y), \\
q_{vr,vf}(m, n, l) + q_{hr,vf}(m, n, l), \\
q_{vr,vr}(m, n, l), \\
\sum_{c=0}^{l} (q_{vr,hf}(x, y - 1, c) + q_{vr,vf}(x, y - 1, c)) \\
+ (q_{vr,hf}(m - x, n - y, l - c) + q_{vr,vf}(m - x, n - y, l - c)) \\
x (q_{vr,hf}(m - x, n - y + 1, l - c - 1) + q_{vr,vr}(m - x, n - y + 1, l - c - 1)) \\
+ p_{hr}(x, y - 1, c), & \text{if } (l > 0) \land (((x = 0) \land (y = 1)) \\
\lor ((x = m) \land (y = n))), \\
\lor ((x = 0) \land (y = 0)) \lor ((x = m) \land (y = n + 1))), \\
\lor (x = o) \lor (y = 0) \lor (l = 0), \\
\lor (x = m) \lor (y = n + 1)), & \text{otherwise.}
\end{cases}
\]
PROOF.

CASE 3.1.

\[(x < -l) \lor (x > m + l) \lor (y < 1 - l) \lor (y > n + l) \lor (-x + y > n + l) \lor (x + y \leq -l) \]
\[\lor (x + y > m + n + l) \lor (x - y \geq m + l).\]

In this case, the unit line is not in the usage area. There is no route across the unit line.

CASE 3.2.

\[(0 \leq x \leq m) \land (0 < y \leq n) \land (l = 0).\]

In this case, there is no reverse segment in the route, and then the route is monotonic. The route will go monotonically from the start point to \((x, y - 1)\), pass through the unit line, and then go monotonically from \((x, y)\) to the end point. The number of monotonic routes from the start point to \((x, y - 1)\) is \(v(x, y - 1)\) and the number of monotonic routes from \((x, y)\) to the end point is \(v(m - x, n - y)\), therefore, the total number of such route is,

\[v(x, y - 1) \times v(m - x, n - y).\]

CASE 3.3.

\[(l > 0) \land ((x = 0) \land (y = 1)) \lor ((x = m) \land (y = n)).\]

In this case, the appointed unit line is connecting the start point (or end point). The route will cross the unit line only when the segment connecting the start point (or end point) is a vertical forward segment. From the definition of function \(q_{vrvf}\) and function \(q_{hrvf}\), the theorem holds under Case 3.3.

CASE 3.4.

\[(l > 0) \land ((x = 0) \land (y = 0)) \lor ((x = m) \land (y = n + 1)).\]

Similar to Case 3.3, the route will cross the unit line only when the segment connecting the start point (or end point) is a vertical reverse segment. From the definition of function \(q_{vrvf}\), we can prove the theorem.

CASE 3.5. Otherwise

Here, we discuss the problem in three subcases.

CASE 3.5.1. The reverse segments are vertical and the unit line is in a forward segment.

Let's assume the total length of the reverse segments in the route from start point to \((x, y - 1)\) is \(c\). The route will go from the \((0, 0)\) to \((x, y - 1)\), pass through the unit line and then go from \((x, y)\) to \((m, n)\). When the route goes from \((0, 0)\) to \((x, y - 1)\), the segment connecting the point \((x, y - 1)\) should not be a reverse segment, because if the segment connecting the point \((x, y - 1)\) is reverse segment, it will overlap the unit line \(V_{x,y}\). The permitted directions for the segment connecting to the point \((x, y - 1)\) are shown in Figure 5a. The number of routes from \((0, 0)\) to \((x, y - 1)\) is

\[q_{vrhf}(x, y - 1, c) + q_{vrvf}(x, y - 1, c)\]

Similarly, the number of routes from \((x, y)\) to \((m, n)\) is

\[q_{vrhf}(m - x, n - y, l - c) + q_{vrvf}(m - x, n - y, l - c)\]

Moreover, it is obvious that the range of \(c\) is \(0 \leq c \leq l\). Therefore, the total number of routes under Case 3.5.1 is,

\[\sum_{c=0}^{l} (q_{vrhf}(x, y - 1, c) + q_{vrvf}(x, y - 1, c)) \times (q_{vrhf}(m - x, n - y, l - c) + q_{vrvf}(m - x, n - y, l - c)).\]
two permitted directions for the segment connecting point \((x, y)\)

the unit line is in a forward segment

(a)

(b)

(c)

Figure 5. Permitted directions for the segments connecting to \(V_{x,y}\) in three subcases.

CASE 3.5.2. The reverse segments are vertical and the unit line is in a reverse segment.

The permitted directions for the segment connecting to the point \((x, y-1)\) and \((x, y)\) are shown in Figure 5b. Similar to Case 3.5.1, we can derive the number of routes under this case is,

\[
\sum_{t=0}^{l} \left( q_{vr,hf}(x, y, t) + q_{vr,vr}(x, y, t) \right) \times \left( q_{vr,hf}(m-x, n-y+1, l-t-1) + q_{vr,vr}(m-x, n-y+1, l-t-1) \right).
\]

CASE 3.5.3. The reverse segments are horizontal.

In this case, since the reverse segments are horizontal, the unit line can only appear in a forward segment. The permitted directions for the segment connecting to the point \((x, y-1)\) and \((x, y)\) are shown in Figure 5c. Similar to Case 3.5.1, we can derive the number of routes under this case is,

\[
\sum_{c=0}^{l} p_{hr}(x, y-1, c) \times p_{hr}(m-x, n-y, l-c).
\]

Considering the sum of Case 3.5.1 to Case 3.5.3, we prove the theorem under Case 3.5.

In similar way, we define \(n_h(x, y, m, n, l)\) to be the number of routes that pass through the horizontal unit line \(H_{x,y}\). Obviously the number of routes that pass through the horizontal unit line \(H_{x,y}\) in the \(m \times n\) routing mesh is equal to the number of routes across the vertical unit line \(V_{y,x}\) in the \(n \times m\) routing mesh. Therefore,

\[
n_h(x, y, m, n, l) = n_v(y, x, n, m, l).
\]

3.3. Density of each Unit Line

Denote \(\text{den}_v(x, y, m, n)\) (or \(\text{den}_h(x, y, m, n)\)) as the density of the vertical unit line \(V_{x,y}\) (or the horizontal unit line \(H_{x,y}\)) under a two-terminal net going from \((0, 0)\) to \((m, n)\). In practice, we attempt to have routes as short as possible. Thus, the shorter route has the higher usage
Table 1. Benchmarks.

<table>
<thead>
<tr>
<th>Benchmarks</th>
<th>Nets</th>
<th>Routing Area</th>
<th>Capacity</th>
<th>Routing Time(s)</th>
<th>Estimating Time(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>9874</td>
<td>80 x 64</td>
<td>22 x 32</td>
<td>64</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>15583</td>
<td>96 x 64</td>
<td>20 x 23</td>
<td>144</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>19386</td>
<td>128 x 64</td>
<td>20 x 33</td>
<td>238</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>28796</td>
<td>256 x 64</td>
<td>14 x 28</td>
<td>866</td>
<td>9</td>
</tr>
</tbody>
</table>

Table 2. Experimental results.

<table>
<thead>
<tr>
<th>#</th>
<th>Density Range</th>
<th>Horizontal Unit Lines</th>
<th>Vertical Unit Lines</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Global Router</td>
<td>1 Detour</td>
</tr>
<tr>
<td>1</td>
<td>Sub1</td>
<td>4788</td>
<td>4869</td>
</tr>
<tr>
<td></td>
<td>Sub2</td>
<td>373</td>
<td>253</td>
</tr>
<tr>
<td></td>
<td>Sub3</td>
<td>31</td>
<td>60</td>
</tr>
<tr>
<td></td>
<td>Sub4</td>
<td>8</td>
<td>18</td>
</tr>
<tr>
<td>2</td>
<td>Sub1</td>
<td>5369</td>
<td>5450</td>
</tr>
<tr>
<td></td>
<td>Sub2</td>
<td>736</td>
<td>650</td>
</tr>
<tr>
<td></td>
<td>Sub3</td>
<td>123</td>
<td>126</td>
</tr>
<tr>
<td></td>
<td>Sub4</td>
<td>12</td>
<td>14</td>
</tr>
<tr>
<td>3</td>
<td>Sub1</td>
<td>7688</td>
<td>6319</td>
</tr>
<tr>
<td></td>
<td>Sub2</td>
<td>572</td>
<td>1569</td>
</tr>
<tr>
<td></td>
<td>Sub3</td>
<td>56</td>
<td>35</td>
</tr>
<tr>
<td></td>
<td>Sub4</td>
<td>4</td>
<td>63</td>
</tr>
<tr>
<td>4</td>
<td>Sub1</td>
<td>14308</td>
<td>14097</td>
</tr>
<tr>
<td></td>
<td>Sub2</td>
<td>2167</td>
<td>2178</td>
</tr>
<tr>
<td></td>
<td>Sub3</td>
<td>140</td>
<td>328</td>
</tr>
<tr>
<td></td>
<td>Sub4</td>
<td>25</td>
<td>37</td>
</tr>
</tbody>
</table>

probability. We assign a weight factor to each route in terms of its length. Let \( w.l(m, n, l) \) be the weight factor for the routes from \((0, 0)\) to \((m, n)\) with \(l\)-length reverse segment. The density of each unit line is defined as,

\[
den_v(x, y, m, n) = \frac{\sum_{l=0}^{d} w.l(m, n, l) \times n.v(x, y, m, n, l)}{\sum_{l=0}^{d} w.l(m, n, l) \times t.n(m, n, l)},
\]

\[
den_h(x, y, m, n) = \frac{\sum_{l=0}^{d} w.l(m, n, l) \times n.h(x, y, m, n, l)}{\sum_{l=0}^{d} w.l(m, n, l) \times t.n(m, n, l)}.
\]
4. EXPERIMENTAL RESULTS

To test our algorithm, we implement our estimator in C language and run on a machine with INTEL® CELERON® 2.4 GHz CPU and 512M RAM. The benchmarks are obtained from UCLA [10]. We tested the benchmarks with the estimator with one detour [8] and our estimator with more detours. We compare all the estimation results to the routing results produced by the global router developed by the research group at UCLA [10].

In order to compare the results, we evenly divide the density range into several subranges, and then count the number of unit lines whose densities are in each subrange. For an estimator, the number of unit lines in each subrange is closer to the number produced by the global router, the estimator is more accurate. The experimental results are listed in Table 2. For all the benchmarks, our estimator predicts very close results to the global router.

Speed is a very important attribute for any estimator. An estimator must be very quick to make it practical. Note Table 1, the column Routing Time lists the runtime of the global router for corresponding benchmark, and the column Estimating Time lists that of our estimator with two detours. Obviously, the estimating time of our estimator is greatly shorter than the routing time.

5. CONCLUSIONS

We proposed an approach for congestion estimation. Given a routing grid and a set of nets to be routed, our model predicts the routing density on each edge of the grid. The routing density provides direct congestion estimation. Because we consider almost all the possible routes of a two-terminal net and the number of bends in the route, the model in this paper is more realistic than the previous probabilistic models. The experimental results validated our estimation.

REFERENCES


APPENDIX 1

\[ \sum_{i=2}^{m+1} \sum_{j=1}^{i-1} C_m^i C_j^{i-j-1} C_{i-1}^{j-1} = \sum_{j=1}^{i} C_m^j \times C_{m+n+i-j}^{n+i} \times C_{i-1}^{j-1}. \]
PROOF.

First, we prove that
\[
\sum_{j=1}^{i-1} C^i_j C_{n+l-1}^{i-j-1} C^{j-1}_{l-1} = \sum_{j=1}^{i} C^i_j C_{n+l-1}^{i-j-1} C^{j-1}_{l-1}
\]
because \(C^i_{l-1} = 0\), for \(j > l\), so when \(l > i-1\),
\[
\sum_{j=i}^{l} C^i_j C_{n+l-1}^{i-j-1} C^{j-1}_{l-1} = 0
\]

Then,
\[
\sum_{j=1}^{i-1} C^i_j C_{n+l-1}^{i-j-1} C^{j-1}_{l-1} = \sum_{j=1}^{i} C^i_j C_{n+l-1}^{i-j-1} C^{j-1}_{l-1} + \sum_{j=i+1}^{l} C^i_j C_{n+l-1}^{i-j-1} C^{j-1}_{l-1}
\]

because \(C^i_{n+l-1} = 0\), for \(j > i - 1\), so when \(l < i - 1\),
\[
\sum_{j=i+1}^{l} C^i_j C_{n+l-1}^{i-j-1} C^{j-1}_{l-1} = 0.
\]

Then,
\[
\sum_{j=1}^{i-1} C^i_j C_{n+l-1}^{i-j-1} C^{j-1}_{l-1} = \sum_{j=1}^{i} C^i_j C_{n+l-1}^{i-j-1} C^{j-1}_{l-1} + \sum_{j=i+1}^{l} C^i_j C_{n+l-1}^{i-j-1} C^{j-1}_{l-1}
\]

So, we have
\[
\sum_{j=1}^{i-1} C^i_j C_{n+l-1}^{i-j-1} C^{j-1}_{l-1} = \sum_{j=1}^{i} C^i_j C_{n+l-1}^{i-j-1} C^{j-1}_{l-1}.
\]

Therefore,
\[
\sum_{i=2}^{m+1} \sum_{j=1}^{i-1} C^i_{m+1} C^i_j C_{n+l-1}^{i-j-1} C^{j-1}_{l-1} = \sum_{i=2}^{m+1} \sum_{j=1}^{i} C^i_{m+1} C^i_j C_{n+l-1}^{i-j-1} C^{j-1}_{l-1} = \sum_{j=1}^{i} \sum_{i=2}^{m+1} C^i_{m+1} C^i_j C_{n+l-1}^{i-j-1},
\]

and
\[
C^i_{m+1} C^i_j C_{n+l-1}^{i-j-1} = \frac{(m+1)!}{i!(m+1-i)!} \cdot \frac{i!}{j!(i-j)!} \cdot \frac{(n+l-1)!}{(i-j-1)!(n+l+j-i)!}
\]

\[
= \frac{(m+1)!(n+l-1)!}{(m+1-i)!(i-j)!(n+l+j-i)!}
\]

\[
= \frac{(m+1-j)!}{(m+1-j)!(m+1-i)!(i-j)!} \cdot \frac{1}{(i-j-1)!(n+l+j-i)!}
\]

\[
= \frac{(m+1)!}{j!(m+1-j)!} \cdot \frac{(m+1)!(n+l-1)!}{(m+1-i)!(i-j)!} \cdot \frac{i!(n+l+j-i)!}{(i-j)!(n+l+j-i)!}
\]

\[
= C^i_{m+1} C^i_{m+1-i} C_{n+l-1}^{i-j-1}.
\]
So,

\[
\sum_{j=1}^{l} C_{i-1}^{j-1} \sum_{i=2}^{m+1} C_{m+1}^{i} C_{n+i-1}^{i-j-1} = \sum_{j=1}^{l} C_{i-1}^{j-1}
\]

\[
\times \sum_{i=2}^{m+1} C_{m+1}^{i} C_{m+1-j}^{i-j-1} C_{n+i-1}^{i-j-1} = \sum_{j=1}^{l} C_{i-1}^{j-1} \sum_{i=2}^{m+1} C_{m+1-j}^{i} C_{n+i-1}^{i-j-1}
\]

According to [11,12], we will have

\[
\sum_{i=2}^{m+1} C_{m+1-j}^{i} C_{n+i-1}^{i-j-1} = C_{m+n+i-j}^{n+l}.
\]

So, we have

\[
\sum_{i=2}^{m+1} \sum_{j=1}^{i-1} C_{m+1}^{i} C_{n+i-1}^{i-j-1} C_{l-1}^{j-1} = \sum_{j=1}^{l} C_{m+1}^{j} \times C_{m+n+i-j}^{n+l} \times C_{l-1}^{j-1}.
\]