Congestion Estimation During Top-Down Placement

Xiaojian Yang, *Student Member, IEEE*, Ryan Kastner, *Student Member, IEEE*, and Majid Sarrafzadeh, *Fellow, IEEE*

***Abstract*—Congestion is one of the fundamental issues in very large scale integration physical design. In this paper, we propose two congestion-estimation approaches for early placement stages. First, we theoretically analyze the peak-congestion value of the design and experimentally validate the estimation approach. Second, we estimate regional congestion at the early stages of top-down placement. This is done by combining the wire-length distribution model and interregion wire estimation. Both approaches are based on the well-known Rent's rule, which is previously used for wire-length estimation. This is the first attempt to predict congestion using Rent's rule. The estimation results are compared with the layout after placement and global routing. Experiments on large industry circuits show that the early congestion estimation based on Rent's rule is a promising approach.**

***Index Terms*—Congestion estimation, physical design, placement, routability.**

I. Introduction

MINIMIZING the total routed wire length is one of the fundamental goals in very large scale integration (VLSI) placement stage. In order to achieve such a challenging objective, a number of heuristics and cost functions were proposed in the past couple of decades. Half-perimeter wire length has emerged as the most typical cost function in placement because it adequately models the routed wire length, especially for two- and three-terminal nets. In general, it is believed that there is a positive correlation between half-perimeter wire length and routed wire length. Many successful placement tools are based on half-perimeter wire-length minimization [1], [2].

As VLSI circuits are growing in both size and complexity, not only the half-perimeter wire length, but also *congestion* need to be emphasized at the placement stage. Congestion is one of the main optimization objectives in global routing. However, the optimization performance is constrained because the cells are already fixed at this stage. A highly congested region in the placement often leads to routing detours around the region, in turn results in a larger routed wire length. Congested areas can also downgrade the performance of global router and, in the worst case, create an unroutable placement in the *fix-die* regime [3].

Congestion can be modeled as the summation of linear [4] or quadratic [5] function of difference between *routing demand* and *routing resource*. Existing congestion reduction techniques include incorporating congestion into cost function of simulated annealing [5], combining a regional router into placement tool [6], and performing a post placement processing step [4], [7]. While congestion reduction at late or postplacement stage is empirically effective, congestion estimate achieved at early placement stages would be equally valuable. First, a congestion-driven placement tool guided by early congestion information might be more powerful. Such a tool could use techniques like white space allocation to relieve layout congestion. Second, early congestion estimates could be utilized by combined logic and layout optimization to improve design convergence. For example, when logic designers are given a number of different net lists, they can estimate the congestion by running only several steps of placement. The net lists with bad estimated congestion will be discarded much earlier.

The main contribution of this paper is to estimate both peak congestion and congestion distribution at early top-down placement stages. Specifically, we quantitatively estimate the maximum congestion prior to placement stage. Also, we give a congestion distribution picture of the chip layout at coarse levels of hierarchical placement flow. Both estimates are made based on Rent's rule—a well-known stochastic model for real circuits. While Rent's rule has been used for wire-length estimation for a long time, to the best of our knowledge, there is no published work on congestion-estimation problem using the same basis. To evaluate our estimation approaches, the estimation results are compared with the real congestion, which is extracted from the layout after placement and global routing. It is generally believed that the global routing output correlates well with the final routing for industrial circuits [8]. Therefore, we target estimates that match the real congestion map after global routing.

Congestion is a function of routing demand and routing resource. Once the technology feature and chip characteristics (die size, number of layers, position of preplaced macros) are fixed, the routing resource is roughly determined.[1] Congestion and routing demand are so closely related that it is straightforward to convert one to the other. In this paper, we will focus on the estimation for routing demand.

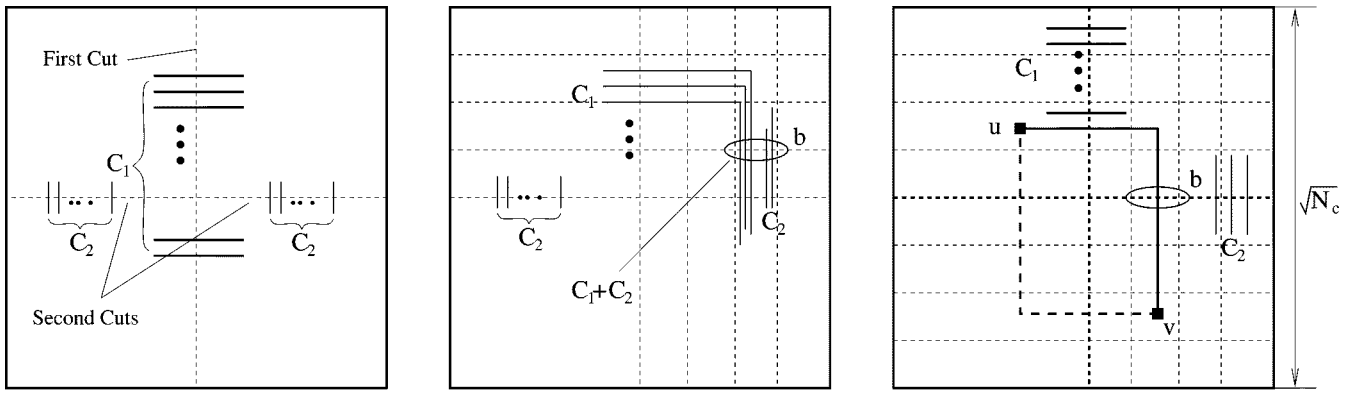
The rest of this paper is organized as follows. Section II briefly introduces the terms and definitions used in this paper. It also reviews Rent's rule, the fundamental theory upon which this paper is based. Section III analyzes the peak-congestion problem and gives a good estimate, which is validated by experiments. Section IV models the regional routing demand in a top-down placement context. Experiments in this section show the effectiveness of the proposed method. Section V gives the conclusion and the future work.

Manuscript received April 20, 2001; revised July 12, 2001. This work was supported in part by the National Science Foundation under Grant CCR-0090203. This paper was presented in part at the International Symposium on Physical Design, Sonoma, CA, April 2001. This paper was recommended by Guest Editor M. D. F. Wong.

The authors are with the Computer Science Department, University of California, Los Angeles, CA 90095 USA (e-mail: xjyang@cs.ucla.edu).

Publisher Item Identifier S 0278-0070(02)00091-X.

[1]Accurate available routing resource can only be obtained after placement and global routing with the consideration of the layer area occupied by placed cells and the number of routing layers. However, a main portion of routing resources could be predicted at this point.



(a) (b) (c)

Fig. 1. Relationship between net cut and congestion. (a) Recursive bipartitioning and cut. (b) Worst case routing demand analysis. (c) Average case routing demand analysis.

II. Preliminaries

A. Placement, Routing, and Congestion

For consistency we will use the following terms throughout the paper. A *circuit* is a hypergraph $G(V, E)$, where V is a set of cells and E is a set of hyperedges. A *hyperedge* $e \in E$ is a subset of V that contains two or more cells. A *placement* is a set of locations for all cells on a rectangular core area.

A common top-down placement flow is the min-cut approach in which the circuit is recursively bipartitioned into subcircuits. Meanwhile the layout area is also partitioned into *placement regions*, each of which contains a corresponding subcircuit.

During the global routing, the chip is divided into an array of uniform rectangular *tiles*. The tile is small enough that each placement region covers an integral number of tiles. All the nets will be routed by connecting the cells of each net using grid wires. For each *boundary* of the tiles b, the *routing demand* $d(b)$ is the number of wires that cross this boundary; the *routing supply* $s(b)$ is the number of wires that are allowed to cross the boundary. The *overflow* of a boundary $c(b)$ is $\max(d(b) - s(b), 0)$. The congestion of a placement region is the summation of the overflow over all the boundaries within this placement region. The *peak congestion* of a placement is the maximum overflow over all the tile boundaries.

B. Rent's Rule

Rent's rule is an empirical observation first described in [9]. It states the relationship between the number of elementary blocks G in a subcircuit of a partitioned design and the number of external connections T of the subcircuit. Specifically

$$T = tG^p \tag{1}$$

where t is the average number of interconnections per block and p is the Rent exponent ($0.4 < p < 0.8$ in real circuits). The Rent exponent p can be computed by plotting the T versus G relation in a log–log diagram for every value of G in a top-down partitioning process and then fitting a line on the plotted points. The slope p of this line represents the Rent exponent.

Rent's rule has been widely used to estimate interconnect wire length [10]–[12]. In general, a higher Rent exponent will result in a longer average wire length, which in turn implies a larger wiring area and more congested layout [13].

III. Peak-Congestion Analysis

A. Cut Ratio in Recursive Bipartitioning

In order to analyze peak congestion over all the tile boundaries of the layout, we assume that the circuit is an *ideal circuit*, which strictly obeys Rent's rule. This ideal circuit is placed using a hierarchical placement flow that is based on recursively bipartitioning. On each hierarchical level of the top-down placement, each subcircuit is quadrisectioned into four smaller subcircuits. A quadrisection step consists of a vertical bipartitioning followed by a horizontal one.

Let C_1 be the net cut of the first bipartitioning by a vertical cut line. Let $C_{2,1}$ and $C_{2,2}$ be the net cuts of the second horizontal bipartitioning [see Fig. 1(a)]. Similarly, the net cuts of the ith bipartitioning are $C_{i,1}, \ldots, C_{i,2^{i-1}}$. For an ideal circuit

$$C_{i,1} \approx C_{i,2} \approx C_{i,3} \approx \cdots, \quad \text{for} \quad i = 1, \ldots, 2H$$

where H is the number of hierarchical levels in the top-down recursive quadrisection placement. For simplicity, all the cuts on the same level i are denoted as C_i for $i = 1, \ldots, 2H$.

Theorem 1: In a recursive bipartitioning approach on an ideal circuit, the ratio between the net cut of the $(i+1)$th bipartitioning C_{i+1} and the net cut of the ith bipartitioning C_i is 2^{-p}, where p is the Rent exponent of the circuit.

Proof: Consider the subcircuits to be bipartitioned at each hierarchical level. Let G_i be the size of subcircuit at the ith level. Thus, the size of subcircuit at the $(i+1)$th bipartitioning is $G_{i+1} = G_i/2$. For and ideal circuit, all the subcircuits have the same Rent exponent p and Rent coefficient t. According to (1), the number of external interconnects for subcircuits at ith level is $T_i = tG_i^p$ and the number of external interconnects for subcircuits at $(i+1)$th level is $T_{i+1} = tG_{i+1}^p$. At a given level i, the interconnects between two subcircuits that are split from one of the ith bipartitionings are a subset of the external interconnects of each of these two subcircuits. Let k $(0 < k < 1)$ be the ratio between the number of cut nets of the ith bipartitioning and the number of external nets for subcircuit at ith level. We have $C_i = kT_i$. For an ideal circuit, assuming k is fixed through all the hierarchical levels, we have

$$\frac{C_{i+1}}{C_i} = \frac{kT_{i+1}}{kT_i} = \frac{kt(G_{i+1})^p}{kt(G_i)^p} = \frac{\left(\frac{G_i}{2}\right)^p}{(G_i)^p} = 2^{-p}. \qquad \blacksquare$$

B. Worst Case Analysis

The top-down placement flow terminates at the $H = \log_4 N_c$ level, where N_c is the number of cells of the circuit. In the final placement, each cell occupies one tile.[2] The global router uses L-shape routing model in which a net is routed using either the upper or the lower part of the bounding box of this net. This is not a good routing method, but it gives a general picture of wire distribution.

Now, we want to find out the maximum routing demand over all the tile boundaries without placing the circuit. First, we discuss the worst case. Let us denote the maximum routing demand of a tile boundary as $C_{\max}$.

Theorem 2: In a recursive bipartitioning approach on an ideal circuit, the maximum routing demand over all the tile boundaries $C_{\max} < C_1(1-\alpha^{2H})/(1-\alpha)$, where C_1 is the net cut of the first bipartitioning and $\alpha = C_{i+1}/C_i$ is the ratio between net cuts of two consecutive partitionings.

Proof: In Fig. 1(b), the circuit is partitioned into two parts with a net cut C_1. It follows that there are C_1 nets crossing between left half and right half. Let us look at a tile boundary located at the right half. In the worst case, all these C_1 nets pass this specific boundary. Hence, the first bipartitioning contributes C_1 to the routing demand of this boundary. Similarly, the ith bipartitioning contributes C_i to the routing demand. Thus, for any boundary, the upper bound of the routing demand is

$$\sum_{i=1}^{2H} C_i = C_1 \sum_{i=0}^{2H-1} \alpha^i = C_1 \frac{1-\alpha^{2H}}{1-\alpha}.$$

■

C. Uniform Distribution of Cut Nets

In the previous discussion, we assume that all the nets that are cut in a bipartitioning cross a particular tile boundary, which is, obviously, not the general case. However, once we construct a framework such as the model in Section III-B, we can study the congestion behavior using different cut-net distribution models.

We continue the analysis using a *uniform distribution model* in which the cut nets of a bipartitioning are uniformly distributed over all the subcircuit area. In other words, the cells in the partitioned subcircuit have equal probabilities to be connected to a cut net.

Theorem 3: In a recursive bipartitioning approach on an ideal circuit, assuming cut nets are uniformly distributed, the expected maximum routing demand over all the tile boundaries is

$$C_{\max} = \frac{C_1}{\sqrt{N_c}} \left(\frac{1}{2} + 2\alpha\right) \frac{\sqrt{N_c}\alpha^{2H} - 1}{2\alpha^2 - 1}$$

where C_1 is the net cut of the first bipartitioning and $\alpha = C_{i+1}/C_i$ is the ratio between net cuts of two consecutive partitioning operations.

Proof: In Fig. 1(c), the first bipartitioning result C_1 means that there are C_1 nets connecting the left half and right half of the design. We know that the number of final tiles on either half is $N_c/2$. Since the cut nets are uniformly distributed, for each final tile, the average number of cut nets connected to this tile is $2C_1/N_c$. Among all the horizontal tile boundaries, the ones on the center horizontal line of the chip accommodate the maximum number of net crossing caused by those C_1 nets. Note that for a specific horizontal tile boundary b at the center line, only the nets which connect to the tiles at the same column could cross b. This is because we are using the L-shape routing model. There are $\sqrt{N_c}$ tiles at the same column with boundary b. For each of them, if the connected tile in the other half is located at the different upper/lower part (half of the connections of this tile have this property), such as tile v and u in Fig. 1(c), the probability that this route crosses boundary b is 1/2 (because of L-shape routing). Therefore the first cut contributes on average $C_1\sqrt{N_c}/(2N_c)$ crossings to b. The case for the second bipartitioning is relatively simple. Since there are C_2 nets crossing $\sqrt{N_c}/2$ boundaries, on average, each boundary is crossed by $C_2/(\sqrt{N_c}/2)$ nets. Using the same approach for the third, fouth, etc. bipartitioning, we have the expected maximum number of crossings over all the boundaries

$$C_{\max} = \frac{1}{2}\frac{C_1}{\sqrt{N_c}} + 2\frac{C_2}{\sqrt{N_c}} + \frac{C_3}{\sqrt{N_c}} + 4\frac{C_4}{\sqrt{N_c}} + \cdots. \quad (2)$$

According to Theorem 1, we have $C_i = C_1\alpha^{i-1}$; plug it into (2) and we obtain

$$C_{\max} = \frac{C_1}{\sqrt{N_c}} \left(\frac{1}{2} + 2\alpha\right) \sum_{i=0}^{H-1} (2\alpha)^{2i} = \frac{C_1}{\sqrt{N_c}} \left(\frac{1}{2} + 2\alpha\right) \frac{\sqrt{N_c}\alpha^{2H} - 1}{2\alpha^2 - 1}.$$

■

D. Experimental Validation

Theorem 3 gives an estimate of peak routing demand for a circuit. If the Rent exponent of a circuit is known, we can estimate the peak routing demand prior to placement stage. The following experiments evaluate the effectiveness of the estimation method.

Given a circuit, we first compute the Rent exponent using the partitioning-tree method proposed in [13]. Specifically, we record the number of cells and the number of external nets for each partition when recursively partitioning the circuit. Then, we do a linear regression on these data points plotted on a log–log diagram. The slope of the linear regression result is the Rent exponent p. The net cut ratio α is then computed using Theorem 1. By partitioning we also obtain the net cuts of each partitioning $C_1, C_2, \ldots$, etc. Due to the existence of Region II of Rent's curve, the first several net cuts do not correlate well with Theorem 1. Therefore, we use C_h/α^{h-1} instead of C_1 in experiments ($h = 8$ in our work). Now, we can compute estimated maximum routing demand using Theorem 3.

The real peak routing demand of the design is obtained by placing and global routing the circuit. The placement algorithm used in the experiments is a recursive bisection approach using a multilevel partitioner hMetis [14]. This is a *global placement* procedure because it stops at a certain $m \times n$ level. Then, an L-shape global router is used to route the global placement output. The peak routing demand is the maximum number of crossings over all the vertical and horizontal tile boundaries.

[2]A tile has unit width and height.

TABLE I
ESTIMATION OF PEAK ROUTING DEMAND

circuit	*#cells*	*#nets*	tile grids	Rent p	est. D_p'	real D_p	*runtime(s)*
ibm01	12,036	13,056	64×64	0.48	**30.3**	31	116
ibm02	19,062	19,291	64×64	0.51	**62.7**	67	208
ibm03	21,924	26,104	64×64	0.65	47.8	62	195
ibm04	26,346	31,328	64×64	0.62	**52.1**	52	218
ibm05	28,146	29,647	64×64	0.69	**89.1**	90	289
ibm06	32,185	34,935	64×64	0.57	82.3	60	305
ibm07	45,135	46,885	64×64	0.55	**86.8**	90	422
ibm08	50,977	49,228	64×64	0.55	111.9	100	511
ibm09	57,746	59,454	64×64	0.49	93.0	75	426
ibm10	67,692	72,760	64×64	0.47	135.8	112	669
ibm11	68,119	78,843	128×128	0.51	**53.9**	50	601
ibm12	69,026	75,157	128×128	0.52	**76.1**	76	771
ibm13	81,018	97,574	128×128	0.39	85.5	108	823
ibm14	147,088	147,605	128×128	0.46	**117.6**	111	1580

Table I shows the comparison between the estimated peak routing demand D_p' and the real peak routing demand D_p. The test circuits are large industrial designs selected from the IBM-PLACE benchmark suites [15], which are derived from the ISPD98 benchmark [16]. Runtime is measured in seconds on a Sun Ultra10 workstation with a 400-MHz central processing unit (CPU). Good estimates are listed in boldface. It should be noted that the estimated peak routing demand is scaled by a factor $\sqrt{N_c/(mn)}$. This is because there are $N_c/(mn)$ cells in each tile, not one cell per tile as in the estimation model.

From Table I, one can see that eight out of 14 estimates are very close to real numbers, supporting our previous analysis. However, the estimates are not accurate for some circuits. There are a number of reasons for poor estimates. First, according to the uniform distribution model, the probability of a cut net connecting to a cell is the same from net to net. In the placement, however, the connected cells tend to be placed closer, causing the estimated value smaller than the real value. Second, in the proof of Theorem 3, we sum up the number of nets crossing boundary b at each level. However, each number is indeed an upper bound for all boundaries. Since the boundaries with maximum crossings at different levels are not identical, the summation (estimated value) tends to be larger than the real value. Other inaccuracies come from the variation of cut ratio α, the chip area aspect ratio (which is not one), etc. These factors are added up causing the uncertainty in the estimation work, including those good estimates. Nevertheless, the fact that the congestion can be predicted is promising and merits further study in this area.

IV. REGIONAL CONGESTION ESTIMATION

In Section III we have discussed the peak-congestion estimation problem. Another estimation requirement, regional congestion estimation, appears at early placement stages. In this section, we propose a routing demand estimation approach in the context of top-down placement.

Definition 1: For a given region r in a globally routed design, the routing demand $D(r)$ is the summation of the number of net crossings over all the tile boundaries within region r.

Note that the *region* in the definition is a common term in top-down placement. It is called *placement region* in [3] or *global bin* in [17]. A region contains a number of adjacent global routing tiles. Estimating the routing demand for all the regions during the placement will give us a rough congestion map, which is valuable for early design evaluation.

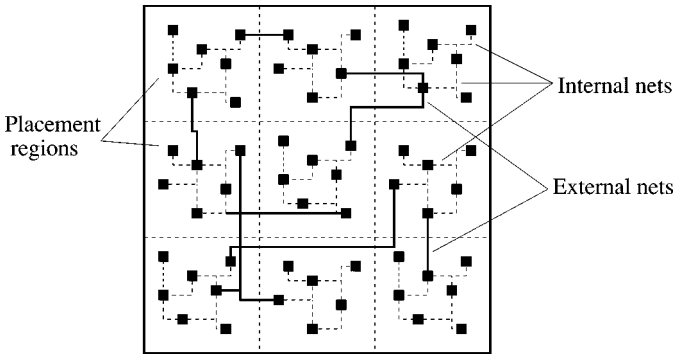


Fig. 2. Internal and external routing demand.

Given a placement region, the nets which cause the edge crossings can be classified into two types: the *internal nets*, which connect cells within this region, and the *external nets*, which span toward other regions or cross this region while connecting no cells. Thus, we give the following terms.

Definition 2: For a given region r in a design, the internal routing demand $ID(r)$ is the summation of the number of crossings caused by internal nets for all tile boundaries; the external routing demand $ED(r)$ is the summation of the number of crossings caused by external nets for all tile boundaries.

The total routing demand $D(r)$ for a region r can be calculated by

$$D(r) = ID(r) + ED(r).$$

Fig. 2 shows the concepts of internal routing demand and external routing demand at a top-down placement stage. The original circuit is divided into subcircuits and each subcircuit is assigned into a region. The dashed lines are internal nets. The thicker solid lines represent external nets. The routing demand in a region consists of two parts: net crossings caused by the internal nets and those by the external nets.

At the very coarse placement stage, the subcircuits are loosely coupled, i.e., the number of external nets is much smaller than the number of internal nets. The routing demand of a region is primarily determined by the interconnect complexity of the subcircuit which belongs to the region. As the top-down placement

flow goes into deeper levels, the routing demand of a placement region is determined by not only the internal complexity of the subcircuit in this region, but also the geometrical locations of other subcircuits and the interconnects between them.

A. Internal Routing Demand

In a typical top-down placement scheme, e.g., min-cut placement, the cells of a partitioned subcircuit will eventually be placed within the area that is assigned for this subcircuit. Therefore, estimating the internal routing demand becomes feasible. For a certain region in a top-down placement, the internal routing demand is proportional to the total routed wire length after global routing [4].[3]

The wire-length estimation problem has been studied for many years. There are several successful estimation techniques based on Rent's rule: Donath's classical method [10], its extension [11], and a more recent model [12]. In these methods the wire-length distribution of the entire design is predicted before place and route. However, in our paper, we estimate wire length for each region and take into account the locality of the Rent's rule. Since different subcircuits have different complexities and Rent parameters, wire-length estimates for subcircuits correspond to the internal routing demand.

Donath's work [10] gives the estimate of total interconnect length for a design. The total wire length of a design can be estimated as

$$L=\begin{cases}\lambda t N_c(1-4^{p-1})\frac{2}{9}\left(7\frac{4^{(p-1/2)H}-1}{4^{p-1/2}-1}-\frac{1-4^{(p-3/2)H}}{1-4^{p-3/2}}\right), & p\neq\frac{1}{2}\\ \lambda t N_c(1-4^{p-1})\frac{2}{9}\left(7H-\frac{1-4^{(p-3/2)H}}{1-4^{p-3/2}}\right), & p=\frac{1}{2}\end{cases}$$

where

N_c	number of gates of the subcircuit;
p	Rent exponent;
t	Rent coefficient;
λ	constant $\lambda \approx 1/2$;
$H=\log_4 N_c$	number of hierarchical levels.

In Davis's wire-length distribution model [12], the interconnect density function $i(l)$, which is defined as the number of interconnects with length l, is given by

$$i(l)=\frac{\lambda t}{2}\Gamma\left(\frac{l^3}{3}-2\sqrt{N_c}l^2+2N_c l\right)l^{2p-4}\quad 1\le l\le\sqrt{N_c}$$

where Γ is a function of N_c and p. When $l>\sqrt{N_c}$, $i(l)\approx 0$. The total wire length of the design is

$$L=\sum_{l=1}^{\sqrt{N_c}} i(l)\cdot l.$$

Both Donath's model and Davis's model are adopted in our internal routing demand estimation. Related results are evaluated in Section IV-E.

B. Locality of Rent's Rule

The input of wire-length estimation approach is the Rent exponent, which, in the previous work, is a fixed value for the entire design. However, a fundamental weakness of Rent's rule is that it does not reflect local fluctuations of interconnection complexity in many designs [18]. If we partition a large circuit into several parts, each part tends to have its own Rent exponent. These Rent exponents may vary a lot. This *locality* of Rent's rule means different interconnect complexities among the modules of designs. In some sense it indicates the local circuit complexity, i.e., congestion.

Fig. 3 shows local Rent curves on circuit ibm09. The curves are obtained by the following approach. We first partition the original circuit into four parts. For each part, we partition it 29 times, dividing this part to 2, 3,. . ., 30 partitions. For each partitioning, we compute the average number of cells per partition G and the average number of external connections per partition T. Then we draw a 29-point plot on a log–log coordinate, which takes $\log G$ as X axis and $\log T$ as Y axis. It is obvious from Fig. 3 that, although all of the four subcircuits are derived from the same design, the Rent curves and exponents of these subcircuits are quite different. This locality feature of large circuits motivates us to examine the relative congestion of different subcircuits.

C. Rent Exponent Extraction

In order to estimate total wire length (or the internal routing demand) of a region, we need to extract the Rent exponent of this region. A traditional way is using partitioning to get numbers of block size and external pins. Then a linear regression is performed when enough numbers are gathered. To make the Rent exponent extraction effective, a minimum number of partitioning is expected.

The part of Rent's rule curve in Region I implies true Rent exponent. Thus, locating the data points becomes a key factor in this method. The following dynamic rent exponent extraction (DREE) algorithm gradually increases the number of partitions when it partitions each subcircuit and then performs linear regression on the latest N ($N=4$ in our approach) data points. For each linear regression, we compute χ^2 probability Q, which indicates the goodness of fit [19]

$$Q=\Gamma_q\left(\frac{N-2}{2},\frac{\chi^2}{2}\right) \tag{3}$$

where N is the number of fitting data points and $\Gamma_q(a,x)$ is the incomplete gamma function which is defined by

$$\Gamma_q(a,x)=\frac{\int_x^\infty e^{-t}t^{a-1}dt}{\int_0^\infty t^{a-1}e^{-a}dt}. \tag{4}$$

Once the quality Q of linear regression is greater than a threshold value (0.9 in our experiments) for every subcircuit, we claim that these regression points are in Rent's rule Region I. The algorithm terminates and outputs Rent exponent for each subcircuit.

The total running time cost of DREE algorithm is dominated by running time of recursive bipartitioning on the circuit, which is very fast due to the recent advances of multilevel partitioning techniques. The method of extracting Rent exponent is similar to the classical approach (e.g., [13]). The algorithm is *dynamic* in the sense that unnecessary partitionings are not performed once a good linear regression is obtained. Thus, it reduces the cost of extracting Rent exponent to running time of several partitioning passes.

```
Algorithm 1: DREE
   Input: n subcircuits G_k = (V_k, E_k)  k = 1, ..., n
```

[3]We assume that the global routing tiles are square.

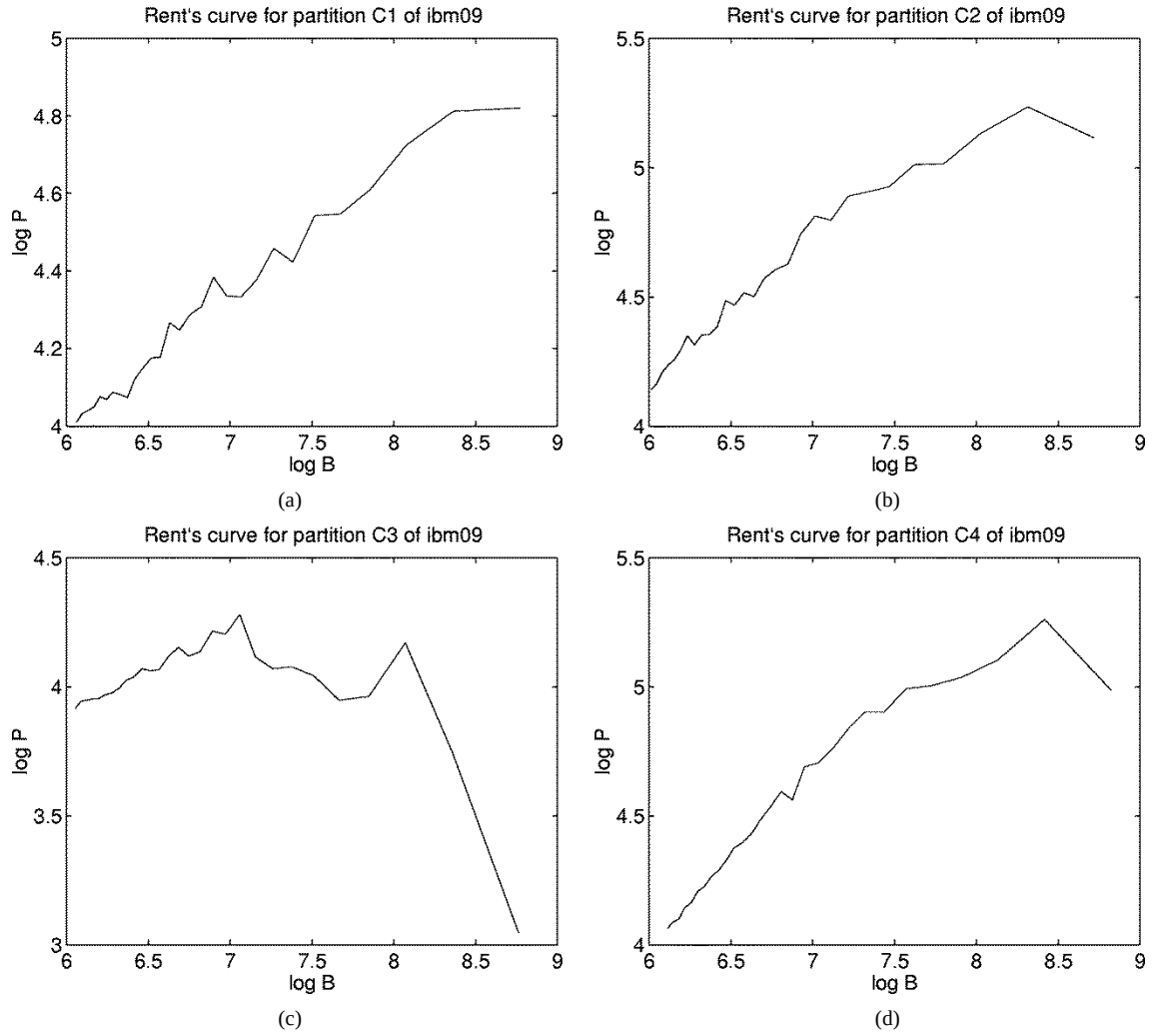


Fig. 3. Local Rent curves for different partitions of circuit ibm09, quadrisectioned into four parts. Rent curve for subcircuit (a) C_1, (b) C_2, (c) C_3, and (d) C_4.

```
Output: Rent exponent r_k of each subcircuit G_k, k = 1,...,n.
k ← 0, m ← number of data points for line fitting
repeat
   k ← k + 1
   for each subcircuit G_i(i = 1,...,n) do
      Do 2^k way partitioning on G_i
      Compute G_i, k = |V(G_i)|/2^k (average number of
      modules per partition)
      T_i, k = average number of external nets per
      partition
      Record point (x_{i,k}, y_{i,k}) = (log G_{i,k}, log T_{i,k})
   end for
   if k ≥ m then
      for each subcircuit G_i, perform a
      linear regression on m-points data set:
      (x_{i,k}, y_{i,k}), (x_{i,k-1}, y_{i,k-1}), ..., (x_{i,k-m+1},
      y_{i,k-m+1})
      get line fitting result equation f_i(x) = a_i x + b_i
      and quality of fitting Q_i
   endif
until Q_i ≥ 0.9 for i = 1,...,n
p_i = a_i, i = 1,...,n
return p_i, i = 1,...,n
```

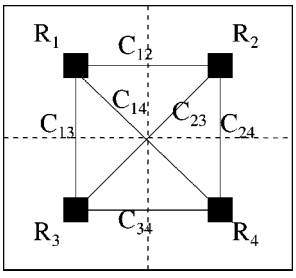


Fig. 4. External routing demand analysis.

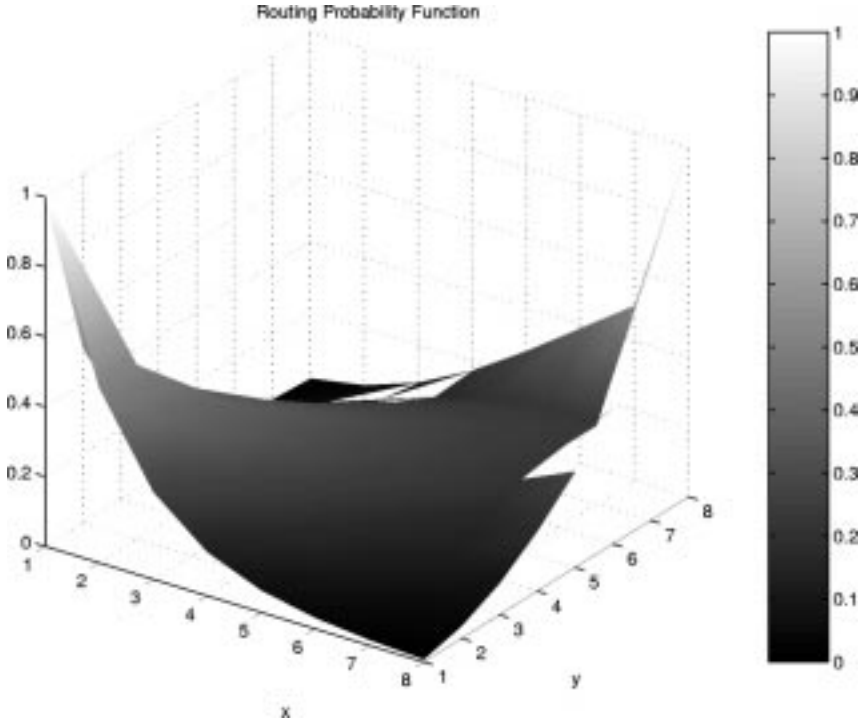
D. External Routing Demand

Internal routing demand can be estimated using Rent parameters, while estimating external routing demand requires the knowledge of interconnection between regions. We cannot simply assume a uniform distribution of interconnects between regions. However, the interconnect distribution of the current placement is known. For each region, we compute the external routing demand based on the interconnects that connect or pass through this region. Fig. 4 shows a simple example of how to estimate the external routing demand.

In Fig. 4, a design area is divided into four rectangular regions: R_1, R_2, R_3, and R_4. Let C_{ij} be the number of intercon-

1	0.5	0.25	0.125	⋯			
0.5	0.5	0.375	⋯				
0.25	0.375	⋯					
0.125	⋯						
⋯							⋯
						⋯	0.387
					⋯	0.226	0.5
				⋯	0.387	0.5	1

(a)



(b)

Fig. 5. Probability density function instance (a) on an 8 × 8 grid and (b) its 3-D representation.

nects between R_i and R_j. Assume that the distance between the center of adjacent regions is one unit. For region R_1, the wire that connects R_1 and other regions contributes one unit to its routing demand. A wire that may pass through R_1 (e.g., C_{23}) statistically contributes a half unit to R_1's routing demand because this wire has a 50% chance to be routed in this region. Then, the estimate of external routing demand (ED) of region R_1 is

$$ED_1 = C_{12} + C_{13} + C_{14} + \frac{1}{2}C_{23}.$$

Similarly, we have

$$ED_2 = C_{12} + C_{23} + C_{24} + \frac{1}{2}C_{14}$$
$$ED_3 = C_{13} + C_{23} + C_{34} + \frac{1}{2}C_{14}$$
$$ED_4 = C_{14} + C_{24} + C_{34} + \frac{1}{2}C_{23}.$$

In general, we assume that there are n rectangle regions. To estimate the external routing demand for a region, the interconnects between every pair of regions will be evaluated. Among them, the ones that connect or pass the evaluated region are counted. Therefore, the external routing demand estimate for region k is

$$ED_k = \sum_{1 \le i,j \le n, i \ne j} C_{ij}\rho_{ij}(k)$$

where C_{ij} is the number of interconnects between region i and region j. $\rho_{ij}(k)$ is the *probability density function*, which indicates the likelihood a wire from region i to region j passes a given region k. It can be calculated by equally assigning probability for interconnects passing from one region to its neighborhood regions. An example of probability density function is given in Fig. 5(a) and its three-dimensional (3-D) representation is shown in Fig. 5(b).

E. Estimation Results

We conduct experiments to estimate the routing demand for every region in the coarse placement. By summing up the wire-

length estimate and crossed routed wires of a region, we obtain the total routing demand. At an early level of top-down placement, we apply the estimation method on every region to get a routing demand distribution. This result will be compared to real routing demand map.

To get the real routing demand distribution, we continue the placement process from the point where we make estimation. The placement approach is a typical recursive bisection flow using a state-of-the-art multilevel partitioner hMetis [14]. After placement, we use a high-quality maze router to complete the global routing stage. The global routing consists of two steps: initial routing and ripup/reroute. Both the total wire length and the congestion (represented by overflow) are part of the cost function in global routing. From the result of global routing, we extract the routing demand, which is the number of net crossings on the boundaries of the global routing tiles within every region. Such a result is the real routing demand distribution, which reflects the wire requirement on a design.

For better comparison, we scale both the estimated routing demand map and the real routing demand map. The scaling process is simply dividing every routing demand value associated with a region by the average routing demand value of the whole chip area. After scaling, for each region r, we have a scaled estimated routing demand $D_e(r)$ and a scaled real routing demand $D_r(r)$. The estimation error for region r is defined by $\text{Error}(r) = |D_e(r) - D_r(r)|/D_r(r) \times 100\%$. The overall estimation error for the design is the average value of the estimation errors over all regions.

All the experiments have been done on a Sun Ultra10 workstation with a 400-MHz CPU. Table II shows the overall estimation error for three different approaches. The first approach uses Donath's wire-length estimation model and routing estimation (RE) method proposed in Section IV-D. The second approach is similar to the first one. The only difference is that it uses Davis's wire-length distribution model. The third approach estimates regional congestion based on Davis's wire-length model only, without RE. The three approaches use the same DREE algorithm to extract Rent exponent. Since the running time for external routing demand estimation can be ignored comparing with the Rent exponent extraction process, we only report the runtime for DREE algorithm, which is dominated by the partitioning approach. Experiments show that the speedup of the partitioning tool (hMetis in this paper) does not degrade the estimation. Thus, we use the fastest option of the partitioning tool.

The results in Table II show that the proposed approach is an effective way to estimate congestion. By combining either Donath's or Davis's wire-length model with the RE method, we can predict relative congestion with small errors. The comparison between approaches with or without RE shows that, in general, wire length only cannot estimate congestion well. There are some different cases for which wire length itself produces good estimates. These cases are mostly in 2×2 placement level where the RE is not as important as later levels.

It should be noted that for a given circuit, the estimation becomes harder as the number of regions increases. Circuit ibm03, ibm04, and ibm14 show the trend. A global router uses detours to avoid routing in a congested area. When regions are large, the detours for congested spots in a given region are still counted as the routing demand for this region. However, if regions are small, the detours contribute routing demand to neighboring regions, which makes estimation difficult. In general, estimates of large regions are more accurate than those of small regions. This suggests that designers may use actual global routing to get congestion information at later top-down placement stages.

TABLE II
ESTIMATION ERROR AND RUNTIME COMPARISON

circuit	*level*	Donath's + *RE*	Davis's + *RE*	Davis's only	*time* (*sec*)
ibm01	2×2	8.9%	10.4%	12.9%	39
	4×4	8.0%	11.6%	6.2%	57
	8×8	10.4%	11.1%	26.0%	128
ibm02	2×2	8.1%	11.9%	6.3%	70
	4×4	8.8%	9.6%	11.6%	147
	8×8	10.4%	11.4%	18.8%	274
ibm03	2×2	7.8%	6.8%	11.2%	79
	4×4	10.8%	13.2%	18.8%	137
	8×8	13.1%	13.3%	24.9%	249
ibm04	2×2	8.3%	6.4%	17.8%	107
	4×4	11.1%	12.1%	15.2%	146
	8×8	13.8%	13.1%	21.2%	287
ibm11	2×2	15.0%	8.8%	2.9%	258
	4×4	8.1%	7.2%	18.2%	327
	8×8	8.4%	8.4%	23.2%	880
ibm12	2×2	11.2%	8.1%	9.8%	352
	4×4	5.9%	4.2%	9.1%	447
	8×8	8.1%	6.4%	11.5%	901
ibm13	2×2	13.1%	7.4%	5.8%	393
	4×4	11.0%	11.3%	17.0%	442
	8×8	8.9%	8.8%	28.9%	962
ibm14	2×2	8.4%	6.3%	7.1%	860
	4×4	9.5%	7.0%	13.4%	721
	8×8	10.4%	10.9%	20.1%	1627
average		9.9%	9.4%	14.9%	

V. CONCLUSION

In this paper, we studied the congestion-estimation problem based on Rent's rule. Although Rent's rule has been widely used on various estimation fields for over two decades, it has not been adopted for congestion analysis. We presented a novel relationship between circuit Rent parameters and peak congestion of the layout. Experiments show that the model works well for L-shape routers. More sophisticated model could be conducted based on the analysis framework. Also we proposed a regional congestion-estimation method that takes both internal and external routing demand into account. The locality of Rent's rule is exploited by our approach. Experiments on large industry benchmarks show that the estimation error is on the average less than 10% compared with the actual congestion obtained by place and route.

Recent research work shows that the Rent exponent of a circuit can be predicted [20]. This will give the Rent's rule-based approach a bright future. The proposed method in this paper, combining with Rent exponent prediction, allows designers to fast predict congestion distribution at early placement stages. Moreover, with fast congestion estimation, a placement tool could incrementally update the congestion map during the placement process and adjust the operation accordingly, thus, achieving a congestion-free layout.

ACKNOWLEDGMENT

The authors would like to thank the reviewers for their detailed feedback.

REFERENCES

[1] C. Sechen and A. Sangiovanni-Vincentelli, "Timberwolf3.2: A new standard cell placement and global routing package," in *Proc. Design Automation Conf.*, June 1986, pp. 432–439.

[2] G. Sigl, K. Doll, and F. M. Johannes, "Analytical placement: A linear or a quadratic objective function," in *Proc. Design Automation Conference*, June 1991, pp. 427–432.

[3] A. E. Caldwell, A. B. Kahng, and I. L. Markov, "Can recursive bisection alone produce routable placements?," in *Proc. Design Automation Conf.*, June 2000, pp. 153–158.

[4] M. Wang and M. Sarrafzadeh, "Behavior of congestion minimization during placement," in *Proc. Int. Symp. Physical Design*, Apr. 1999, pp. 145–150.

[5] C. E. Cheng, "Risa: Accurate and efficient placement routability modeling," in *Proc. Int. Conf. Computer-Aided Design*, Nov. 1994, pp. 690–695.

[6] P. N. Parakh, R. B. Brown, and K. A. Sakallah, "Congestion driven quadratic placement," in *Proc. Design Automation Conf.*, June 1998, pp. 275–278.

[7] M. Wang, X. Yang, K. Eguro, and M. Sarrafzadeh, "Multi-center congestion estimation and minimization during placement," in *Proc. Int. Symp. Physical Design*, Apr. 2000, pp. 147–152.

[8] L. Scheffer and Eric Nequist, "Why interconnect prediction doesn't work," in *Proc. Int. Workshop System-Level Interconnect Prediction*, Apr. 2000, pp. 139–144.

[9] B. Landman and R. Russo, "On a pin versus block relationship for partitions of logic graphs," *IEEE Trans. Computers*, vol. C-20, pp. 1469–1479, Dec. 1971.

[10] W. E. Donath, "Placement and average interconnection lengths of computer logic," *IEEE Trans. Circuits Syst.*, vol. CAS-26, pp. 272–277, Apr. 1979.

[11] D. Stroobandt and J. V. Campenhout, "Accurate interconnection length estimations for predictions early in the design cycle," *VLSI Design*, vol. 10, no. 1, pp. 1–20, 1999.

[12] J. A. Davis, V. K. De, and J. Meindl, "A stochastic wire-length distribution for gigascale Integration(GSI)—Part I: Derivation and validation," *IEEE Trans. Electron Devices*, vol. 45, pp. 580–589, Mar. 1998.

[13] L. Hagen, A. B. Kahng, F. J. Kurdahi, and C. Ramachandran, "On the intrinsic Rent parameter and spectra-based partitioning methodologies," *IEEE Trans. Computer-Aided Design*, vol. 13, pp. 27–37, Jan. 1994.

[14] G. Karypis, R. Aggarwal, V. Kumar, and S. Shekhar, "Multilevel hypergraph partitioning: Application in VLSI domain," in *Proc. Design Automation Conf.*, June 1997, pp. 526–529.

[15] "IBM-PLACE benchmark suites," Northwestern Univ. CAD Lab, http://www.ece.nwu.edu/nucad/ibm-place.html.

[16] C. J. Alpert, "The ISPD98 circuit benchmark suite," in *Proc. Int. Symp. Physical Design*, Apr. 1998, pp. 18–25.

[17] M. Wang, X. Yang, and M. Sarrafzadeh, "Dragon2000: Fast standard-cell placement for large circuits," in *Proc. Int. Conf. Computer-Aided Design*, Nov. 2000, pp. 260–263.

[18] H. V. Marck, D. Stroobandt, and V. Campenhout, "Toward an extension of Rent's rule for describing local variations in interconnection complexity," in *Proc 4th Int. Conf. Young Computer Scientists*, Apr. 1995, pp. 136–141.

[19] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, "Numerical recipes in C," in *The Art of Scientific Computing*. New York: Cambridge Univ. Press, 1992, vol. 1, pp. 661–666.

[20] P. Christie, "Rent exponent prediction methods," *IEEE Trans. VLSI Syst.*, vol. 8, pp. 679–688, Dec. 2000.

Xiaojian Yang (S'00) received the B.S. degree in computer science from Tsinghua University, Beijing, China, in 1994 and the M.S. degree from the Chinese Academy of Sciences, Beijing, China, in 1997. He is currently working toward the Ph.D. degree in computer science at University of California, Los Angeles.

From 1998 to 2000, he was a Ph.D. Student with the Northwestern University Computer-Aided Design Laboratory. His current research interests include physical design of VLSI/CAD, with an emphasis on wire length, congestion, and timing issues in deep-submicrometer placement.

Ryan Kastner (S'00) received the B.S. degrees in electrical engineering and computer engineering and the M.S. degree in computer engineering from Northwestern University, Evanston, IA, in 1999 and 2000, respectively. He is currently working toward the Ph.D. degree in computer science at the University of California, Los Angeles.

His current research interests include physical design, reconfigurable computing, and compilers.

Majid Sarrafzadeh (S'82–M'87–SM'92–F'96) received the B.S., M.S., and Ph.D. degrees in electrical and computer engineering from the University of Illinois at Urbana-Champaign in 1982, 1984 and 1987, respectively.

He joined Northwestern University as an Assistant Professor in 1987. In 2000, he joined the Computer Science Department at University of California at Los Angeles. He has collaborated with many industries in the past ten years, including IBM and Motorola and many computer-aided design industries. He has authored or coauthored approximately 250 papers, an invited chapter in the *Encyclopedia of Electrical and Electronics Engineering* in the area of VLSI circuit layout, and *An Introduction to VLSI Physical Design* (New York: McGraw-Hill, 1996) and coedited *Algorithm Aspects of VLSI Layout* (Singapore: World Scientific, 1994). His current research interests include embedded and reconfigurable computing, VLSI and computer-aided design, and design and analysis of algorithms.

Dr. Sarrafzadeh received the National Science Foundation Engineering Initiation Award, two Distinguished Paper Awards at the International Conference on Computer-Aided Design, and the Best Paper Award at the Design Automation Conference for his work in the area of physical design. He is on the editorial board of the *VLSI Design Journal*, an Associate Editor of *ACM Transaction on Design Automation*, and an Associate Editor of IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS.He has served on the technical program committee of numerous conferences in the area of VLSI and computer-aided design. He has served as a Committee Chairs for a number of conferences, including the International Conference on Computer-Aided Design, and was the General Chair of the 1998 International Symposium on Physical Design.