# Probabilistic Congestion Model Considering Shielding for Crosstalk Reduction *

Jinjun Xiong Lei He
EE Department, University of California at Los Angeles
Los Angeles, CA, 90095

## ABSTRACT

We extend an existing probabilistic congestion model to consider shielding for crosstalk reduction. We then develop a multilevel router to study the impact of various congestion models on routing congestion by using large industrial design examples. We show that (1) when shielding is applied as a post-routing optimization for crosstalk reduction, the existing probabilistic model, when compared to a deterministic routing-order dependent congestion model, reduces routing congestion by 17.1% on average under the given routing area constraints, or reduces routing area by 9.4% on average under the given routing congestion constraints; (2) our extended probabilistic congestion model considering shielding enables shielding reservation and minimization for routing and achieves routing congestion (or area) reduction by 47.7% (or 31.0%) on average under the given routing area (or congestion) constraints, when compared to the above deterministic congestion model not able to estimate shielding and therefore not able to minimize shielding during routing.

## 1. INTRODUCTION

Efficient yet accurate routing congestion estimation is of great importance for many physical design algorithms, like placement and routing. Deterministic congestion estimation has been developed based upon net counts, net's bounding box perimeter, Rent's rule, or building routing trees via either pattern routing, Steiner tree routing or global routing followed by detailed routing [1]. Using combinatorial analysis, several recent works proposed probabilistic congestion estimation models [2, 3]. It is generally believed that the probabilistic congestion model overcomes the limit of routing-oder dependency and correlates well with real design. For example, [4] and [5] adopted such a probabilistic congestion model and successfully applied it to placement and floor-planning, respectively. However, none has employed such a model for routing.

As the clock frequency continues to increase and the minimum feature size keeps shrinking, crosstalk reduction becomes increasingly important. Shielding has been proven effective to reduce both capacitive and inductive crosstalk and has been used in modern micro-processor designs [6, 7]. Unfortunately, shielding has area overhead because it consumes extra routing resource. Therefore, congestion estimation without considering shielding for crosstalk reduction will not provide an accurate congestion picture to guide physical design tools. However, current congestion estimation models used in routing either do not consider shielding or consider shielding in an order dependent fashion.

The major contribution of this work is as follows. We extend an existing probabilistic congestion model to consider shielding for crosstalk reduction. Under such a model, a novel multilevel routing system is developed to evaluate the impact of different congestion models on routing in terms of both congestion and routing area. Using large industrial designs, we show that on average 47.7% congestion reduction or 31.0% routing area reduction can be achieved by using our extended probabilistic congestion model considering shielding for crosstalk reduction.

## 2. PRELIMINARY

The routing area is tessellated into rectangular partitions as *routing tiles*, and all cells along with their connection pins are placed at the center of routing tiles. The circuit layout can be formally modeled by an undirected graph $G(V,E)$ where each vertex $v \in V$ represents a routing tile at location $(x,y)$, and each edge $e \in E$ represents the routing area between two adjacent tiles. The right horizontal edge of $(x,y)$ is denoted as $h(x,y)$, and the upper vertical edge of $(x,y)$ is denoted as $v(x,y)$. An edge in the routing graph is also called a *routing region* $R_t$. To model the limited routing resources, we associate each edge in $G(V,E)$ with a *capacity* $C_t$, the maximum number of tracks available for routing. Same as in [8, 9], we define the congestion of $R_t$ as the difference between the required track number and the available track number, i.e.,

$$D_t = G_t - C_t. \tag{1}$$

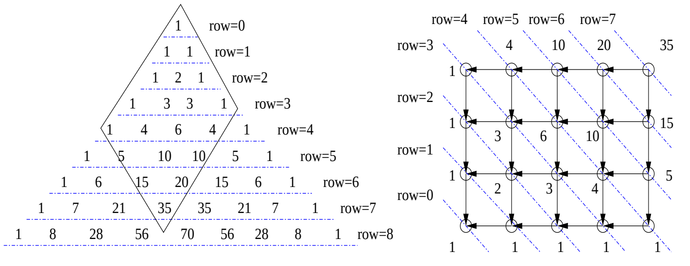
where $G_t$ is the number of signal nets routed in $R_t$. If $D_t > 0$, overflow occurs in $R_t$; otherwise, $R_t$ has no overflow.

There are many possible routing paths from the source to the sink in $G$. We call paths without detour as *valid paths* in this paper. If only valid paths are considered, the number of paths from $(u,w)$ to $(0,0)$ can be derived analytically as shown in [3]. We associate each node (x,y) in $G$ with a number $m(x,y)$ to denote the total number of paths from that node to the sink at (0,0). By the definition of valid paths, we have: $m(0,y) = 1$, $\forall y$; $m(x,0) = 1$, $\forall x$; and $m(x,y) = m(x-1,y) + m(x,y-1)$. Because of the recursive definition of $m(x,y)$, we can compute $m(x,y)$ via recursion. One example on how to compute $m(x,y)$ is shown in Fig. 1(b), where the number associated with each node is $m(x,y)$.

By drawing a series of diagonal lines in $G$ according to

---

*This research is partially supported by NSF CAREER Award CCR-0093273, SRC contract 1081, and grants from Fujitsu and Intel. We used computers donated by Intel and Sun Microsystems. Address comments to lhe@ee.ucla.edu.



(a) (b)

**Figure 1: (a) A Pascal's Triangle. (b) $m(x,y)$ in $G$.**

their Manhattan distances $(x+y)$ from the sink at (0,0), we note that $m(x,y)$ in fact form a rectangle that is part of a Pascal's Triangle [10] in Fig. 1(a). Therefore, the total number of valid paths from $(u,w)$ to $(0,0)$ is given by:

$$m(u,w) = \left( \begin{array}{c} u+w \\ w \end{array} \right) = \frac{(u+w)!}{u! \cdot w!}. \tag{2}$$

If we assume every valid path has the same probability to be chosen, then the *probability* of edge $h(x,y)$ or $v(x,y)$ being used to route the net from (u,w) to (0,0) is given by:

$$p(x,y)_h = \frac{q(x,y)_h}{m(u,w)} \tag{3}$$

$$p(x,y)_v = \frac{q(x,y)_v}{m(u,w)} \tag{4}$$

where $q(x,y)_h = m(x,y) \cdot \left( \begin{array}{c} u+w-(x+1)-y \\ w-y \end{array} \right)$, and $q(x,y)_v = m(x,y) \cdot \left( \begin{array}{c} u+w-x-(y+1) \\ w-(y+1) \end{array} \right)$.

The extension of the above computation to multi-pin nets can be carried out as follows: we decompose each multi-pin nets into two-pin nets first, then we can compute the congestion probability for each two-pin net individually.

# 3. CONGESTION WITH SHIELDING FOR CROSSTALK REDUCTION

The following crosstalk reduction model has been used successfully in industry practices for modern micro-processor designs [6, 7]: i.e., reduce the crosstalk for critical signal nets by putting shields adjacent to those critical signal nets. According to [6], signal nets are characterized into three categories according to their criticality in the timing graph: the most critical nets are shielded on both sides, which we call $s2$-nets; the next most critical nets are shielded on only one side, which we call $s1$-nets; and the rest of nets are non-critical nets and require no shielding, which we call $s0$-nets. The definition of signal nets' criticality can be obtained via either static timing analysis or noise optimization as shown in [7].

For a given routing region $R_t$, the congestion with shielding becomes $D_t = G_t + S_t - C_t$, where $S_t$ is the number of shields. Therefore, minimizing the number of shields due to shielding is very valuable for congestion reduction. This is especially true for high-end microprocessor designs because more and more nets may require shielding. We have the following Theorem:

THEOREM 1. *Given a routing region $R_t$ with $m_2$ number of s2-nets, $m_1$ number of s1-nets, and $m_0$ number of s0-nets, in order to satisfy the signal shielding requirements, the minimum shield number $S_t$ is given by:*

$$S_t = \lceil \frac{m_1}{2} \rceil + m_2. \tag{5}$$

If we denote $p_{j,t}^{(2)}$, $p_{j,t}^{(1)}$, and $p_{j,t}^{(0)}$ as the probability of s2-nets, s1-nets, and s0-nets in $R_t$, respectively, then we can approximate the total number of s2-nets, s1-nets, and s0-nets by $\sum_j p_{j,t}^{(2)}$, $\sum_j p_{j,t}^{(1)}$, and $\sum_j p_{j,t}^{(0)}$, respectively. According to the definition of congestion, we can estimate the routing congestion with shielding for crosstalk reduction as follows:

$$\begin{aligned} D_t \;\; = \;\; & \sum_j p_{j,t}^{(2)} + \sum_j p_{j,t}^{(1)} + \sum_j p_{j,t}^{(0)} \\ & + \lceil \frac{\sum_j p_{j,t}^{(1)}}{2} \rceil + \sum_j p_{j,t}^{(2)} - C_t. \end{aligned} \tag{6}$$

Because all the above quantities are obtained via closed formulae, the probabilistic routing congestion with shielding can be computed very efficiently.

# 4. MULTI-LEVEL ROUTING

In the following, we present a novel multilevel routing algorithm with shielding constraints for crosstalk reduction. A typical multilevel routing framework consists of two parts: *coarsening* and *uncoarsening*. Coarsening is to obtain a relatively accurate routing resource estimation for higher level routing. Uncoarsening is to find a global routing solution hierarchically based upon the routing resource information obtained from coarsening. Due to space limitation, we refer readers to [11, 12, 1] for more detailed discussion about multilevel routing techniques

Existing multi-level routing systems have the following drawbacks: (1) the crosstalk constraints are not considered explicitly; (2) the congestion estimation is not accurate. In this work, we employ the probabilistic congestion estimation developed in Section 3 during coarsening. By doing this, we can consider all nets' routing resource requirement (including shielding) simultaneously, because the congestion estimation is probabilistically accurate. In our current implementation, we denote the multi-level routing algorithm that employs the deterministic and routing-order dependent congestion model [1] as $MR$, the one that employs the existing probabilistic congestion model (Section 2) as $MRP$, and the one that employs our probabilistic congestion model considering shielding for crosstalk reduction as $MRPS$. As the routing framework for $MR$, $MRP$ and $MRPS$ is very similar, in the following, we only explain the overall algorithm for $MRPS$ as shown in Fig. 2.

### 4.0.1 Coarsening and Uncoarsening for MRPS

According to Fig. 2, the layout is first transfered into the lowest level grid routing graph. We then compute the path distribution probability for all nets in each routing edge at the full-chip level by using the method developed in Section 3. In the coarsening stage, the exact net number in each region is not known. But by using the probability-based planning procedure, we can have a relatively accurate congestion picture with much less effort than congestion estimation via detailed routing [1]. At each coarsening level, only $s2$- and

```
Construct the lowest level routing graph;
//Coarsening
For all nets
  Compute net path distribution probability in $\overline{G}$;
For all routing regions
  Compute probabilistic congestion estimation (6);
For($i$ = level from finest to coarsest)
  Compute coarse level routing resource;
  Global routing of $s2-$ and $s1-nets$ within current level;
//Uncoarsening
For($i$ = level from coarsest to finest)
  Global routing of un-routed nets within current level;
  Refine routing solution if necessary;
//Track assignment
For (all finest routing region)
  Track assignment considering shielding for crosstalk reduction;
```

**Figure 2: The $MRPS$ algorithm overview.**

$s1$-nets belonging to the current level are routed. The cost function for path $P_e$ is given by $cost(P_e) = \sum_{\forall t \in P_e} 2^{D_t}$, where $D_t$ is the region congestion from (6). Because in our current implementation routing is done within each net's bounding box, there is no need to include the routing length in computing the routing cost, but the extension to include routing length is straight-forward. A path is *overflow* if any edge in $P_e$ has overflow. We choose a path that minimizes the cost function without overflow. If we cannot find such a path during coarsening, we mark it as a *failed* net and it will be refined during uncoarsening. When we compute the path cost, we apply the congestion estimation equation (6) from Section 3 for each routing region. By doing this, we reserve an appropriate number of free tracks for shields for the following track assignment procedures in order to reduce crosstalk.

The uncoarsening stage refines each local failed nets and all other un-routed nets starting from the coarsest level. For better routability, the routed $s2$-nets or $s1$-nets can also be modified if such a modification results in better routability in terms of the cost function. In our current implementation, maze routing is employed to route local nets belonging to the current level during uncoarsening; and we confine the maze search scope within the tile defined by the current level and allow overflow. After uncoarsening, rip-up and reroute will be used to further improve the routing solution.

#### 4.0.2 Track Assignment Considering Shielding for Crosstalk Reduction

After multi-level routing, we then perform a post-routing optimization technique that decides the track assignment solution for both signal nets and shields within each routing region for crosstalk reduction. As track assignment is performed within each routing region, and we have reserved an appropriate number of free tracks for shields during the multi-level routing stage, therefore, we can easily find a valid track assignment solution without affecting congestion much. The optimal track assignment solution has been proved in Theorem 1. The algorithmic implementation of this step is the same as the construction based proof of Theorem 1.

## 5. EXPERIMENT RESULTS

The proposed $MR$, $MRP$ and $MRPS$ algorithms have been implemented in C++ on Linux. Nine large industrial benchmarks from the ISPD'98/IBM benchmark suite are employed to show the applicability of these algorithms to real designs. The characteristics of the benchmarks are shown in Table 1. We assume 10% $s2$-nets and 10% $s1$-nets for all benchmarks. The overall routing congestion for each benchmark is measured in terms of the total number of overflowed segments for all overflowed routing regions, i.e., $OvSeg = \sum_{D_t > 0} D_t$ for all $R_t$ with $D_t > 0$. We measure the overhead due to shielding in terms of the total number of shields inserted, i.e., $Shields = \sum_{\forall t} S_t$.

| Ckts | Net # | Pin # | Cell # | Tile # |
|---|---|---|---|---|
| ibm01 | 13056 | 44266 | 12752 | 128 × 128 |
| ibm02 | 19291 | 78171 | 19601 | 128 × 128 |
| ibm03 | 26104 | 75710 | 23136 | 128 × 128 |
| ibm04 | 31328 | 89591 | 27507 | 128 × 128 |
| ibm05 | 29647 | 124438 | 29347 | 128 × 128 |
| ibm06 | 34935 | 124399 | 32498 | 256 × 256 |
| ibm07 | 46885 | 244369 | 45926 | 256 × 256 |
| ibm08 | 49228 | 198180 | 51309 | 256 × 256 |
| ibm09 | 59454 | 187872 | 53395 | 256 × 256 |

**Table 1: Benchmark settings .**

In the first experiment setting, we give a fixed routing area for each benchmark, i.e., every benchmark has a fixed routing resource constraint, and measure the final routing quality in terms of routing congestion. The experiment results are reported in Table 2 with column 2 giving the fixed routing area for each benchmark. Column 3, 5 and 8 are the overall routing congestion for $MR$, $MRP$, and $MRPS$, respectively. Column 6 and 9 are the congestion reductions of $MRP$ and $MRPS$ over $MR$, respectively. From column 6, we find that $MRP$, under the guidance of a probabilistic congestion estimation, can improve the overall congestion in almost all cases when compared to $MR$ that is under the guidance of a deterministic and routing-order dependent congestion estimation. The congestion reduction is 17.1% on average and can be up to 92.6%. Similarly, from column 9, we find that $MRPS$ always results in better overall congestion than $MR$, and the average reduction is 47.7%. Column 10 are the congestion reduction of $MRPS$ over $MRP$. From column 10, we found that $MRPS$ with shielding estimation and reservation during routing can significantly improve the overall congestion when compared to $MRP$ that does not consider shielding during routing. On average, 36.9% reduction can be achieved for $MRPS$ over $MRP$.

We also report the runtime in column 4, 7 and 11 in Table 2. Because of the overhead associated with the probability computation, $MRP$ and $MRPS$'s runtime are longer than $MR$ in our current implementation. However, the total runtime for $MRP$ and $MRPS$ are still very reasonable for all test cases, and the worst case runtime for $MRP$ and $MRPS$ are 577.8 seconds and 574.5 seconds for IBM05, respectively. Because closed formulae are used to compute the shielding estimation, the runtime between $MRP$ and $MRPS$ are very comparable.

In the second experiment setting, we give a fixed routing congestion constraint for each benchmark and measure the final routing quality in terms of the required routing resources (area) for $MR$, $MRP$ and $MRPS$, respectively. In order to do that, we incrementally increase the routing area until every benchmark is routed with an overall congestion less than or equal to 1%. The purpose of this experiment is to see in order to achieve similar routing congestion, which routing algorithm can utilize the limited routing area better. We report the experiment results in Table 3. The normalized routing area is shown in columns 3, 5 and 8 for $MR$, $MRP$ and $MRPS$, respectively. Column 6 and 9 are routing area reduction for $MRP$ and $MRPS$ over $MR$, re-

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|
| Test Ckts | height × width | *MR* | | *MRP* | | | *MRPS* | | | |
| | | OvSeg # | Time(s) | OvSeg # | | Time(s) | OvSeg # | | | Time(s) |
| IBM01 | 1280×1280 | 787 | 5.0 | 58 | -92.6% | 17.3 | 5 | -99.4% | -91.4% | 17.3 |
| IBM02 | 1920×1920 | 17435 | 21.2 | 14025 | -19.6% | 49.4 | 10551 | -39.5% | -24.8% | 48.7 |
| IBM03 | 1920×1920 | 4238 | 17.5 | 2561 | -39.6% | 61.0 | 1083 | -74.4% | -57.7% | 60.7 |
| IBM04 | 1920×1920 | 6573 | 14.9 | 4581 | -30.3% | 49.2 | 2581 | -60.7% | -43.7% | 49.4 |
| IBM05 | 4480×4480 | 70957 | 112.5 | 546913 | -12.3% | 577.8 | 37305 | -47.4% | -40.1% | 574.5 |
| IBM06 | 2560×2560 | 5864 | 17.9 | 3633 | -38.0% | 71.9 | 1283 | -78.1% | -64.7% | 70.9 |
| IBM07 | 2560×2560 | 30318 | 33.6 | 21506 | -29.1% | 76.0 | 16866 | -44.4% | -21.6% | 75.4 |
| IBM08 | 2560×2560 | 30357 | 40.2 | 26906 | -11.4% | 82.7 | 17419 | -42.6% | -35.3% | 82.2 |
| IBM09 | 2560×2560 | 10768 | 31.6 | 11515 | 6.9% | 72.5 | 5672 | -47.3% | -50.7% | 72.1 |
| Avg | | | | | -17.1% | | | -47.7% | -36.9% | |

**Table 2: Comparison of overall routing congestion (*OvSeg*) and runtime. Column 2 shows the given routing area constraint for each benchmark.**

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|
| Test Ckts | *MR* | | *MRP* | | | *MRPS* | | | |
| | Shield # | height × width | Shield # | height × width | | Shield # | height × width | | |
| IBM01 | 256 | 1600 × 1280 | 320 | 1280 × 1280 | -20.0% | 139 | 1280 × 960 | -40.0% | -25.0% |
| IBM02 | 535 | 3200 × 2560 | 920 | 2560 × 2560 | -20.0% | 334 | 2560 × 1920 | -40.0% | -25.0% |
| IBM03 | 461 | 2240 × 2240 | 377 | 2240 × 2240 | 0.0% | 70 | 1920 × 1920 | -26.5% | -26.5% |
| IBM04 | 1502 | 2240 × 2560 | 932 | 2240 × 2240 | -12.5% | 1489 | 1920 × 2240 | -25.0% | -14.3% |
| IBM05 | 4874 | 6080 × 6400 | 3182 | 5760 × 5760 | -14.7% | 4678 | 5120 × 5440 | -28.4% | -16.0% |
| IBM06 | 698 | 3200 × 2880 | 1096 | 2880 × 2880 | -10.0% | 1028 | 2560 × 2560 | -28.9% | -21.0% |
| IBM07 | 2445 | 3520 × 4480 | 1258 | 3840 × 4160 | 1.3% | 2729 | 2560 × 3840 | -37.7% | -38.5% |
| IBM08 | 3120 | 3200 × 3840 | 1709 | 3520 × 3520 | 0.8% | 2499 | 2880 × 3200 | -25.0% | -25.6% |
| IBM09 | 3061 | 2880 × 3520 | 2816 | 2880 × 3200 | -9.1% | 2498 | 2240 × 2880 | -36.4% | -30.0% |
| Avg | | | -25.6% | | -9.4% | -10.9% | | -31.0% | -23.9% |

**Table 3: Comparison of routing area and shield number under the 1% overall congestion constraint.**

spectively. Compared to $MR$, on average, $MRP$ consumes 9.4% less routing area, while $MRPS$ consumes 31.0% less routing area. Column 10 is the routing area reduction of $MRPS$ over $MRP$. It shows that $MRPS$ results in smaller routing area in all test cases, and on average, 23.9% reduction can be achieved for $MRPS$ when compared to $MRP$. This clearly shows that a routing algorithm that employs the probabilistic congestion estimation model can significantly improve the utilization of the limited routing resource, and early shielding estimation and reservation during routing is important in order to minimize the negative impact of shielding on congestion. We also report the total number of shields used in each benchmark in column 2, 4 and 7 for $MR$, $MRP$ and $MRPS$, respectively. According to Table 3, we see that $MRP$ and $MRPS$ also use less number of shields than $MR$, and the reduction on average are 25.6% and 10.9%, respectively.

## 6. CONCLUSION AND DISCUSSION

We have presented an extended probabilistic congestion estimation model that takes into account signal nets' shielding requirements for crosstalk reduction. We have also developed a novel multi-level routing system, $MRPS$, with explicit shielding reservation and minimization during routing for crosstalk reduction, and applied the extended probabilistic congestion estimation model to $MRPS$ to improve routing congestion. Compared to the alternative multi-level routing algorithm $MR$ under a deterministic and routing-order dependent congestion model, $MRPS$ can reduce the average routing congestion by 47.7%, or reduce the average routing area by 31%. We have also shown that early shielding estimation and reservation during routing is important to minimize the negative impact of shielding on congestion. Compared to $MR$ that uses shielding as a post-routing optimization, $MRPS$ can reduce the average routing congestion by 36.9% or reduce the average routing area by 23.9% with similar runtime.

To reduce on-chip crosstalk, we have employed a high abstraction level yet effective crosstalk reduction model (shielding requirements for critical nets) same as [7, 6]. We recognize that such a model is conservative for real designs. For example, to reduce crosstalk, it is not necessary to shield critical nets from the source to the sinks. In the future, we will develop similar high abstraction level but more practical models for crosstalk reduction, and apply them to our multilevel routing framework.

## 7. REFERENCES

[1] S.-P. Lin and Y.-W. Chang, "A novel framework for multilevel routing considering routability and performance," in *ICCAD*, pp. 44–50, 2002.

[2] Kusnadi and J. D. Carothers, "A method of measuring nets routability for mcm's general area routing problems," in *Proceedings of the 1999 international symposium on Physical design*, pp. 186–192, 1999.

[3] J. Lou, S. Krishnamoorthy, and H. S. Sheng, "Estimating routing congestion using probabilistic analysis," in *Proc. Int. Symp. on Physical Design*, 2001.

[4] U. Brenner and A. Rohe, "An effective congestion-driven placement framework," in *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 22, pp. 387–394, April 2003.

[5] Y.-L. Hsieh and T.-M. Hsieh, "New effective congestion model in floorplan design," in *Design, Automation and Test in Europe Conference and Exhibition*, vol. 2, pp. 1204–1209, Feb 2004.

[6] P. Saxena and S. Gupta, "On integrating power and signal routing for shield count minimization in congested regions," *TCAD*, April 2003.

[7] B. Chappell, X. Wang, P. Patra, P. Saxena, J. Vendrell, S. Gupta, S. Varadarajan, W. Gomes, S. Hussain, H. Krishnamurthy, M. Venkateshmurthy, and S. Jain, "A system-level solution to domino synthesis with 2 GHz application," in *Proc. IEEE Int. Conf. on Computer Design*, pp. 164–171, Sept. 2002.

[8] M. Wang and M. Sarrafzadeh, "Modeling and minimization of routing congestion," in *Proc. Asia South Pacific Design Automation Conf.*, 2000.

[9] H. Su, J. Hu, S. Sapatnekar, and S. Nassif, "Congestion-driven codesin of power and signal networks," in *Proc. Design Automation Conf*, pp. 64–69, 2002.

[10] M. Aigner, *Combinatorial Theory*. Springer-Verlag, 1997.

[11] J. Cong, J. Fang, and Y. Zhang, "Multilevel approach to full-chip gridless routing," in *ICCAD*, pp. 396–403, 2001.

[12] J. Cong, M. Xie, and Y. Zhang, "An enhanced multilevel routing system," in *ICCAD*, pp. 51–58, 2002.