Tutorial on Congestion Prediction

Taraneh Taghavi **Foad Dabiri** **Ani Nahapetian** **Majid Sarrafzadeh**

Computer Science Department,
University of California, Los Angeles (UCLA)
Los Angeles, CA 90095
{taghavi, dabiri, ani, majid}@cs.ucla.edu

ABSTRACT

With the increasing sophistication of circuits and specifically in the presence of IP blocks, new estimation methods are needed in the design flow of large-scale circuits. Up to now, a number of post-placement congestion estimation techniques in the presence of IP blocks have been presented.

In this paper we present a unified approach for predicting wirelength, congestion and delay parameters early in the design flow. We also propose a methodology to integrate these prediction methods into the placement framework to handle the large complexity of the designs.

Categories and Subject Descriptors

G.3 [**Probability and Statistics**]: Statistical computing.

General Terms

Algorithms, Measurement, Design, Experimentation, Theory.

Keywords

Prediction, Congestion, Wirelength, Delay, Algorithm.

1. INTRODUCTION

In the physical layout of Integrated Circuits, a critical quantity is the amount of wire required for interconnections; quite frequently, this length is used as a measure of the quality of the placement [9, 51]. On the other hand, to achieve more efficient designs, the computer-aided design flow is following the trend of combining front-end floor-planning and physical placement. In this process a fast but accurate estimation of total wirelength is critical. The reason is that the efficiency and accuracy of front end planning depends on the performance of floor-planning, placement and partitioning tools, and the efficiency of all of these tools depend on the accuracy of the interconnect estimator [10]. With the increasing size and sophistication of circuits, and specifically in the presence of IP blocks, new wirelength estimation methods are needed in the design flow of very large-scale circuits.

The total wirelength affects three major parameters of today's chip design cycle: Chip Size, Clock Frequency, and Power Dissipation. Since the above parameters are largely affected by interconnect lengths, total wirelength is frequently used as a measure of the quality of the placement. For the placement and routing phases, the quality requirements are particularly stringent [19]. For the result of these phases to be acceptable, accurate predictions of relevant post-layout circuit properties are an absolute necessity to limit the search of the vast solution space. Hence, CAD tools use estimation tools usually based on partitioning methodologies [39, 52, 53, 54, 55, 56]. Furthermore, an integrated prediction of interconnect delay and length is needed to handle the complexity of today's designs.

Minimizing the total routed wirelength is one of the fundamental goals in the VLSI placement stage. However, as VLSI circuits are growing in complexity and more importantly in the presence of extremely large number of IP blocks, not only the wirelength but also the congestion needs to be emphasized at the placement. So far, all of the congestion estimation methods perform post-placement congestion estimation. However in the presence of IP blocks, alleviating congestion after placement may result in an abrupt increase in wirelength; therefore, congestion needs to be estimated early enough to guide placement to avoid generating highly-congested and hence un-routable designs.

In [19], the authors extracted the peak and average congestion before placement using Rent's parameter of the circuit. They have also formulated regional congestion as the summation of internal and external routing demands. They have shown that internal routing demand is well-correlated with the length of interconnects and so congestion estimation should be provided by the information of wirelength estimation.

Due to the increasing size of the circuits, many industrial integrated circuits have very high utilization factor and so congestion is very likely to happen in many areas of the chip after placement. If one area of the chip is very congested, it is very likely that most of the wires may need to have a detour around the congested area although no IP blocks are present in that area. Hence, wirelength estimation may need to know the congestion map of the circuit, and consider calculating the detour for the congested areas. As a result, congestion and wirelength estimation are dependent and should be performed concurrently, early in design flow.

The remainder of this paper is organized as follows. Section 2 presents an overview of the previous research on wirelength estimation. In Section 3, we discuss the congestion estimation problem and our methodology for concurrent congestion and

SLIP'07, Month 17–18, 2007, Austin, Texas, USA.

wirelength estimation. In Section 4, given the congestion map of the design, we show how to alleviate congestion. Section 5 represents timing analysis for delay prediction. Our methodology for integrated prediction of congestion and delay is presented in Section 6.

2. WIRELENGTH ESTIMATION

2.1 Individual Net Length Estimation

2.1.1 Half Perimeter Bounding Box

A very popular technique for estimating the length of an individual net is the half perimeter wirelength estimation which equals the half perimeter of the bounding box of a net [80]. It has been proven that this technique provides the optimal solution for 2-pin and 3-pin nets and a lower bound for nets with higher degree. However, it can significantly underestimate wirelength for higher-degree nets. Cheng [59] proposed a net weighting technique to scale up the HPWL estimation. The net weights are degree dependent constants and are experimentally determined. However, even for different nets with the same degree, the error in the HPWL estimation can be very different. It is impossible to derive a single net weight to accurately scale up the HPWL estimation for all nets.

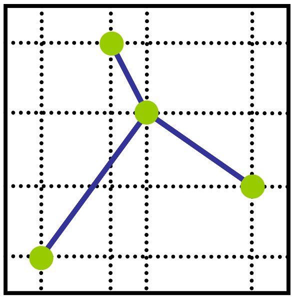

Figure 1: Minimum Spanning Tree [32]

2.1.2 Minimum Spanning Tree

Another commonly used technique for estimating wirelength is using the minimum spanning tree (MST). A spanning tree $T = \{V_T, E_T\}$ of a graph is a subgraph T of G, where T is a tree (no cycles) and such that $V_T = V$ and $E_T \subseteq E$. A minimum spanning tree is a minimum weight spanning tree over a weighted graph which over a rectilinear grid, as shown in Figure 1. This approach can produce good wirelength estimation in reasonable amount of time. The best time complexity of MST is $O(n \log(n))$. However, a simple $O(n^2)$ time implementation of Prim's algorithm is usually used in practice.

2.1.3 Steiner Tree

We can also achieve accurate estimation by constructing rectilinear Steiner minimal tree (RSMT) using either optimal algorithms or near-optimal heuristics [81, 61, 77, 79]. But these algorithms are computationally too expensive to be used in practice.

The Steiner problem is defined as follows: Given a set P of n points, find a set S of Steiner points such that $MST(P \cup S)$ has the minimum cost. Hanan pointed out that an optimal RSMT can always be constructed based on the Hanan grid. Note that the length of a horizontal (respectively, vertical) edge in the Hanan grid is equal to the distance. An example of a rectilinear Steiner tree is shown in Figure 2.

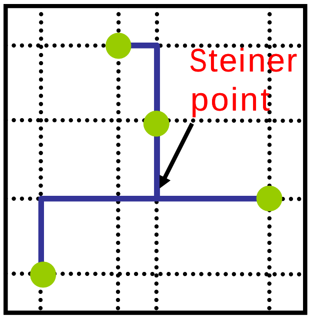

Figure 2: Rectilinear Steiner tree [32]

2.2 Total Interconnect Length Estimation

2.2.1 Rent Rule

Early work on wirelength estimation was based upon an empirical model known as Rent's Rule [31]. Rent's Rule correlates the number of signal input and output terminals T, to the number of gates C, in a random logic network by the power law relation $T=AC^P$. A is often called as Rent Coefficient, which is the average number of pins per cell. The Rent Exponent, P, is the feature parameter of the circuit [17].

This exponent is a measure of the interconnection complexity of a circuit and can vary between 0 and 1 [36]. Higher Rent exponent values correspond to a higher complexity [49]. In normal circuits, values for P have been observed to range from 0.5 for regular structures (such as RAM) to 0.75 for complex structures (such as fast VLSI circuits) [37]. Gate circuits of average complexity have A=2.5 and P=0.6 [38].

Sutherland and Oestreicher [43] developed a method for estimating track requirements for PC boards. Their method however depends on random placement and for large circuits it would yield excessively large estimates. Their method can be used as an upper bound for interconnection length.

2.2.2 Donath Wirelength Estimation Method

Using Rent's Rule, the first work on wirelength estimation is done by Donath [9]. Donath developed a wirelength estimation method based on a hierarchical placement technique which resulted in much more accurate estimations. His results have been used by several other researchers [44, 33, 46, 47, 48, 35, 40, 41, 65, 66]. Donath's technique estimates average wirelength based on a hierarchical placement of the circuit.

In [33] a new analysis of Donath's model has been presented that yields the length distribution functions of the interconnection at both the hierarchical level and the system level. With the new model of analysis, it was shown that the functions are consistent with the previously derived equation for the average interconnection length and that the distribution function accurately describes the distribution of interconnections within previously constructed computer systems.

2.2.3 Davis Wirelength Estimation Method

In [20], a rigorous derivation of a complete wire-length distribution for on-chip random logic networks was performed by recursively applying Rent's rule throughout an entire monolithic system. In this work, to predict the interconnect length, a continuous interconnect density function has been defined which determines the distribution of the number of interconnections based on the interconnect length in units of gate pitches. First the stochastic wire-length distribution of a single gate has been calculated based on Rent rule using the defined density function. Once the stochastic wirelength distribution has been determined for a single element, it is removed from the system and the same process is repeated for the other gates of the system. The wirelength distributions for individual gates are superimposed to obtain the wirelength distribution for the entire system.

2.2.4 Non-uniform Probability Distribution Methods

More recent work improves the wirelength estimation by considering non-uniform probability [19, 34, 42, 50]. These works are modifications of Donath's technique by introducing the occupancy probability as a better model for an optimal placement. The interconnection length distribution is defined as a collection of values, indicating for each length *l*, how many interconnections have this length. The sum of these values over all lengths *l* equals the total number of interconnections. The interconnection length distribution is decomposed into two distributions as $\varepsilon_l = S(l) f(l)$, where $S(l)$ is the structural distribution which depends on the physical architecture the circuit will be placed in and is determined by the enumeration of all pairs *(p, q)* in distance $L(p,q) = l$ apart in the Manhattan grid. The $f(l)$, the occupancy probability of a pair of points *(p, q)*, is the probability that the pair will effectively be connected by a wire in an optimal placement of the circuit in the physical architecture. By changing the distribution function of the interconnection points, the occupancy probability for pairs of points is changing. Since an optimal placement prefers shorter interconnections over longer ones, it thus seems acceptable to assume that most point pairs at the shortest distance will be occupied and less at the longer distance. Intuitively, the occupancy probability is expected to be a monotonic decreasing function of the wire length. In [33, 45], by using simple theoretical considerations, the normalized distribution ε_l of interconnection lengths for 2D placement is computed as $\varepsilon_l = Cl^{2r-3} \quad (1 \le l \le l_{\max})$. By using this normalized distribution and the estimating the structural distribution given by Donath as *l*, the occupancy probability is extracted as $f(l) = Cl^{2r-4}$. Then the average wirelength is calculated by making use of this occupancy probability.

2.2.5 Wirelength Estimation in Presence of IP Blocks

Most of the research done on wirelength estimation is based on regularly placed circuits such as standard cell designs. With the trend toward IP-block-based design, macro cells as *blockage* (sometimes referred to as *obstacle*), are more likely to be present in the circuit.

The blockage may be an on-chip memory, analog or RF blocks, or pre-designed hard IP. The presence of the blockage may significantly increase wirelength and cause congestion [10].

The first works on the wirelength estimation in the presence of obstacles have been done by Cheng et. al.[10, 21]. In [10], the authors identified two distinct effects of obstacles on interconnection length: (1) changes due to the redistribution of interconnect terminals and (2) detours that have to be made around the obstacles. Theoretical expressions of both effects for two-terminal nets have been derived based on geometrical characteristics of the circuits. In [21], the authors represent a more complicated analytical model using a polynomial generation technique considering the layout region aspect ratio and the presence of the blockage. Their work, however, lacks from considering the complexity of the circuits into account and hence the average wirelength obtained form these techniques overestimate the actual wirelength for circuits with large chip area.

We have proposed a methodology to estimate the average wirelength in [1, 2, 4, 6], which is based on a hierarchical placement of the circuit into a square Manhattan grid in the presence of blockages. This methodology is very similar to Donath's method for estimating total wirelength. The circuit is partitioned hierarchically into four sub-circuits. This hierarchical partitioning is continued until the number of the standard cells in all of the sub-circuits is equal or less than β, where β is a predefined constant. At each level of hierarchy, the average number n_h of interconnections and the average length L_h of interconnections between each two sub-circuits belonging to the same (*h*+1) level of hierarchy, but different *h* level of hierarchy are deduced. The total interconnection length over all hierarchical levels is then obtained from

$$L_{tot} = \sum_{h=0}^{H} n_h L_h \qquad (1)$$

where *H* is the finest level of hierarchy. The average number of interconnections between the sub-circuits in each level of hierarchy has been extracted using Rent's exponent which is experimentally proven to be a good indicator of the complexity of the circuit. The average length of interconnection between the sub-circuits is calculated in each level of the hierarchy. Then, using formula 1, the total wirelength is estimated by multiplying the average number of interconnections by the average length of interconnections for each level of hierarchy and summing these values over all the hierarchical levels.

The average number of interconnections at each hierarchical level can be calculated using Rent's rule as in [9]. Donath showed that by apply Rent rule on each level of hierarchy, the average number of interconnections can be calculated from

$$n_h = \alpha AC(1-4^{P-1})4^{L(P-1)} \qquad (2)$$

Where *C* is the total number of cells, *P* is the Rent exponent, *A* is the Rent coefficient and α is the fraction of the number of terminals for all the interconnections in one level. The value α is ½ if each net has just two terminals, and is somewhat greater but less than 1 for circuits with multi-terminal nets [9]. Parameter *L* shows the level of hierarchy.

In the presence of the Blockages, the average wirelength is represented as the summation of three parts, transparent-block and detour in X and Y directions as $\overline{L} = \overline{L}_{TB} + \overline{L}_{DT}^{h} + \overline{L}_{DT}^{v}$. The *transparent-block* wirelength, L_{TB}, is defined as the wirelength when the blockage is assumed to be transparent and wires can pass through it, while the *Detour wirelength*, L_{DT}, is the detour length needed in a routing wire because of the presence of the blockages. The probabilistic analysis to extract transparent-block and detour parts of interconnect length for horizontally, vertically and diagonally adjacent sub-circuits have been done in [1] completely and so is omitted here for brevity. For the probability distribution of wires, the assumption of uniform probability distribution for standard cells from [2, 6, 10] seems to be inaccurate since it does not consider the optimal placement tools behavior to place connected cells closer to each other. In [19], it was shown that the power-law formula $l^{-(4-2P)}$ can approximate the actual probability of occurrence of a wire with length *l* which is used by [9] for extraction of wirelength.

Having had the average wirelength for horizontally, vertically and diagonally adjacent sub-circuits, the average inter-bin wirelength can be obtained for each level of hierarchy *h* as

$$L_h = \frac{1}{6}\left(\begin{array}{l} \delta\left(L^{hr}(A,B) + L^{hr}(C,D) + L^{v}(A,C) + L^{v}(B,D)\right) \\ + (1-\delta)\left(L^{d}(A,D) + L^{d}(B,C)\right) \end{array} \right) \qquad (3)$$

where *hr*, *v*, and *d*, respectively, denote that the corresponding sub-circuits (*A*, *B*, *C*, and *D*) are horizontally, vertically, or diagonally adjacent. Moreover, δ is a parameter to capture the optimization behavior of placement algorithm, which favors horizontally and vertically adjacent sub-circuits to diagonally adjacent ones.

2.3 Open Problems

There are several research problems to consider for improving the wirelength estimation techniques such as:

- Considering the effect of vias and number of bends in the estimation
- Incorporating the effect of big IP blocks and the slivers (narrow spaces) between them
- Expanding wirelength estimation methods for 3D placement

3. CONGESTION ESTIMATION

3.1 Post-Placement Congestion Estimation

Several post-placement algorithms have been presented to analyze congestion before routing. The authors in [59] introduced an empirical parameter and used it as a weight for bounding box router. In [16, 24], the authors derived the mathematical equation to estimate the congestion using a normal distribution approximation. Some other works used a routing estimation model to predict routing congestion [23]. In [15], the authors proposed a stochastic uniform routing distribution model to compute the expected track usage. The authors in [22] calibrated their method and used it for wirelength estimation. In [11] the authors used a probabilistic approach to estimate the congestion. The authors in [12] enhanced this approach using the router's intelligence factor in minimizing the number of bends in each routed nets.

3.2 Pre-Placement Congestion Estimation

Regional congestion estimation appears in the early placement stage. In [6] the authors proposed a routing demand estimation approach in the context of hierarchical placement for regional congestion estimation. 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. 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 can be calculated by:

$$D(r) = ID(r) + ED(r) \qquad (4)$$

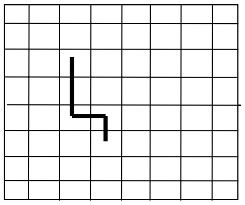

Figure 3: Internal Routing Demand is equal to wirelength

3.2.1 Internal Routing Demand

For a certain region in a top-down placement, it is shown in [18] that the internal routing demand is proportional to the total routed wirelength after global routing. As shown in Figure 3, if each bin grid edge contributes one unit in the wirelength, the number of crossings on the bin grids a wire with length *l* makes in a region equals to the *l*.

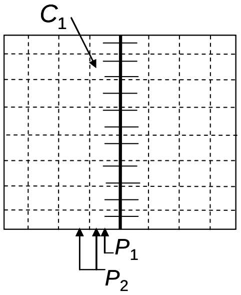

Figure 4: Cut Set C_1

The congestion estimation method needs to know the total wirelength of a region in order to compute the regional internal routing demand. In order to estimate the congestion of a circuit on every region, we need to estimate its total wirelength beforehand for estimating the internal routing demand on every level of hierarchy. The method for wirelength estimation which was discussed in detail in the previous section can be employed to perform hierarchical congestion estimation. As it can be seen, the detour happened because of IP blocks are incorporated into congestion estimation by computing the internal routing demand for each bin.

3.2.2 External Routing Demand

External routing demand for a bin is equal to the number of crossings over the edges of that bin as shown in Figure 6. Internal routing demand can be estimated using Rent parameters, while estimating external routing demand requires the knowledge of interconnection between regions.

As discussed in previous section, the power-law function $l^{-(4-2P)}$ can describe the occupancy probability for a wire of length *l* where *P* is the Rent's exponent of the circuit. Basically this function depicts the probability that a wire of length *l* can occur in the circuit. Since this is a power-law formula with negative power, it indicates that long wires can occur less as compared to short wires. This model is well-correlated with the behavior of placement tools use to place connected cells closer to each other as much as possible. Having this probability distribution, we can present the relationship between probabilities of wires of length *i* and (*i* + 1) as

$$p_{i+1} = \frac{(i+1)^{-(4-2P)}}{(i)^{-(4-2P)}} \times p_i \qquad (5)$$

where P_i is the probability of occurrence of a wire with length *i*.

Figure 4 shows a circuit divided into bins. Let us consider the first vertical cut which results in cut set C_1. One terminal of each cut from C_1 is in the right part and the other terminal is in the left part of the circuit. The minimum length of each cut is 1 (for adjacent bins on the cut border) and maximum length is $2\sqrt{NC}$ (for bins on two opposite corners), where *NC* is the total number of bins. So we have:

$$\sum_{i=1}^{2\sqrt{NC}} P_i = 1 \qquad (6)$$

By substituting formula (5) in (6), we have:

$$P_1 \sum_{i=0}^{2\sqrt{NC}-1} (i+1)^{-(4-2P)} = 1 \qquad (7)$$

To solve equation (7), we use the Riemann Zeta function which is defined as

$$\zeta(x) = \frac{1}{\Gamma(x)} \sum_{0}^{\infty} \frac{u^{x-1}}{e^{u}-1} du \qquad (8)$$

where $\Gamma(x)$ is the Gamma function. Using Zeta function by the help of Mathematica software [27], it can be shown that

$$\sum_{i=0}^{2\sqrt{NC}-1} (i+1)^{-(4-2P)} = Zeta(4-2P) - Zeta[4-2P, 2\sqrt{NC}+1] \qquad (9)$$

As can be seen in Figure 4, only the bins close to the cut boundary can contribute in terms of wires with length 1. So, the contribution of each of these bins in wires with length 1 would be equal to $C_1 \times P_1 / \sqrt{NC}$. Similarly, the contribution of each bin in terms of wires with other length can be calculated. By summing up the number of wires of different length that pass over a bin, the external demand of that bin caused by cut C_1 can be estimated. Obviously, when a bin is blocked, its contribution in any wire would be zero. The same process needs to be done for all other cuts and their external demands need to be summed to generate the total external demand for a specific bin.

3.3 CONCURRENT CONGESTION AND WIRELENGTH ESTIMATION

In this section, we propose a concurrent hierarchical wirelength and congestion estimation. We apply this estimation method on a hierarchical placement framework after each level of global and detailed placement. Our placement framework is a typical top-down hierarchical placement approach. At each level of hierarchy, the layout area is partitioned into several global bins. The bin grid for congestion is set to be the same as the partitioning bin grid. At each level of hierarchy, our methodology performs wirelength and congestion estimation for each bin.

In order to estimate the congestion of a bin we need to estimate the internal and external routing demands of that specific bin. In section 3.1 we have shown that the internal routing demand of a bin equals to the wirelength inside that bin which can be estimated using the method in section 2. For external routing demand, we apply the algorithm which was discussed in section 3.2.

In order to estimate the wirelength of a bin, we need to do a bottom-up hierarchical analysis over that specific bin considering its geometrical characteristics. This analysis can be done by using the wirelength estimation method given in section III. With this method, at each level of hierarchy each bin is partitioned into 4 sub-bins. The average number of interconnections per sub-bin can be computed using formula (2) which uses the Rent exponent of the circuit as an indicator of the complexity of the circuit. Section III described how to extract the average length of interconnections per each sub-bin. By having average number and length of interconnections per each sub-bin, the average length of interconnection for each bin can be extracted using formula (3).

However, using this wirelength estimation method, the effect of congestion is not considered. With the increasing size of the circuits, many industrial integrated circuits have very high utilization factor, which means they have a huge number of logic blocks. As a result, congestion is very likely to happen in many areas of the chip during the placement. If one area of the chip is very congested, it is very likely that all of the wires will not be able to pass through the congested areas, which means some of them may need to be detoured in order to be routed even though no IP blocks is present in that specific area. To consider calculating the detour for very congested areas, a model of an artificial IP block can be used in the wirelength estimation.

To consider the effect of congested areas of the bin on wirelength, a congestion map of the bin should be given in each level of hierarchy. In order to provide this congestion map, a congestion analysis as described on section 4 should be performed before proceeding to the next level of hierarchy for wirelength estimation. If some sub-bins are very congested, they are artificially set to be a blockage for the analysis in the next level of hierarchy. By using this technique, we can incorporate the effect of congested areas into wirelength estimation.

3.4 Open Problems

There are some open problems to consider for congestion estimation such as:

- Considering the effect of vias in the estimation
- Extending congestion estimation for 3D placement
- Extending the concurrent estimation methods for congestion and wirelength for the other estimation methods

4. CONGESTION ALLEVIATION

In order to eliminate the congestion, we need a global view of the congestion map of a design [54, 57, 58, 60, 63, 64]. Let us consider Figure 5. In this figure a small window of the congestion map of a design is shown. Here we can see two congested regions on the right and left sides of the window. In the first glance it seems that to remove the congestion, two congested regions should be expanded toward the middle part of the window, which is not congested. Now let us consider the whole congestion map of the same design in Figure 6. As we see in this figure, the above statement is does not hold any more, since the two sides of those congested regions have plenty of white space and they should be expanded in the opposite directions.

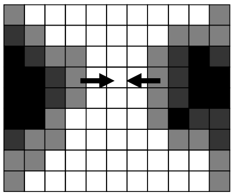

Figure 5: a window showing part of congestion map

Flow algorithms try to distribute congestion by moving the cells out of congested areas. The main problem with this category of algorithms is that they greedily try to expand congested regions, but they ignore the newly generated congested regions due to the interfering of the expanded regions. Figure 7 demonstrates this problem. As it can be seen in this figure, there are two congested regions which make a new crossing congested region when get expanded. To fix this problem in flow algorithm, new congestion estimation is needed after each round of augmenting flow inside the flow algorithm, which is very inefficient. Moreover, flow algorithms produce large changes in floorplan which cannot deteriorate the quality of wirelength optimization.

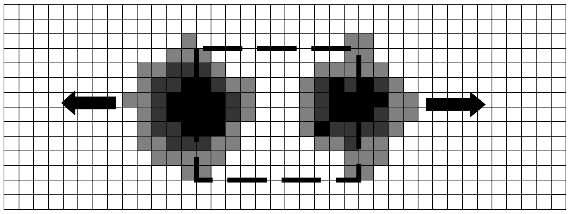

Figure 6: The whole congestion map of a design; the rectangle with dashed border is the same window in figure 5

These facts imply that flow algorithms are unable to handle congestion minimization for interfering regions and furthermore cannot preserve wirelength quality of the design. This fact motivates us to propose new approaches to address the problem of congestion reduction for the whole design in floorplanning step.

To solve the problem of congestion reduction over a bin grid**,** we detect the congested regions by plotting a contour around each of them. To form the non-interfering congested regions, we construct a graph with each vertex corresponding to a congested region. Each edge in this graph is connecting two congested regions if their contours overlap with each other. We find the non-interfering congested regions by a maximum independent set algorithm. We construct a graph for all the bins belonging to the regions of this set and apply flow algorithm to remove the congestion. For the rest of the congested regions (the interfering ones), we apply a modified ZSA algorithm [69, 72, 78].

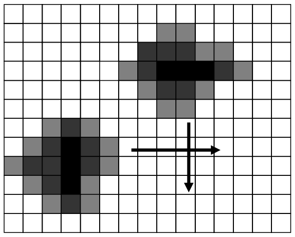

Figure 7: The interfering flow happens in the crossing of two arrows

To solve our problem with zero slack assignment we construct a directed acyclic graph (DAG) over the bins that are not included in the maximum independent set we formed in the previous section. Each node of the directed acyclic graph G_1 corresponds to one of those bins. There is a directed edge from a bin with extra white space to all the bins with overflow. An example of such constructed graph is shown in Figure 8.

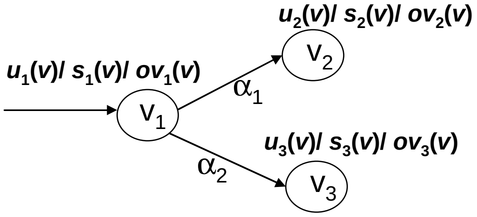
$u_2(v)/\ s_2(v)/\ ov_2(v)$
$u_1(v)/\ s_1(v)/\ ov_1(v)$
v_2
α_1
v_1
$u_3(v)/\ s_3(v)/\ ov_3(v)$
α_2
v_3

Figure 8: Constructed graph for minimum perturbation slack assignment; *u* (*v*), *s* (*v*), and *ov* (*v*) are utilized space, total space and overflow (slack), respectively.

Since we do not want to deteriorate the wirelength quality after congestion reduction, we need to incorporate perturbation minimization parameter into the modeling of our problem by slack assignment algorithm. For this purpose, each edge needs to have a weight corresponding to the Manhattan distance of its two endpoints. In the traditional zero slack assignment algorithm, the objective is to maximize the weighted sum of the slacks. To adjust our formulation with the traditional slack assignment algorithm, we define the weight of each edge *w* (*u*, *v*) proportional to 1/distance (*u*, *v*). This implies that the further two nodes (bins) are, the less the slack (white space) is traversing between them and so the less the IP blocks are moved between those nodes (bins). We normalize edge weights such that the summation of edge weights for all outgoing edges of a vertex should be equal to 1. This ensures that the portion of the white space of a bin which is going to be assigned to several other bins is distributed among them proportional to their Manhattan distance from that specific bin. So, our problem formulation can be represented as:

$$\text{Maximize} \quad \sum_{v\in Adj(u)} slack(u,v)\times\alpha(u,v), \forall u$$

$$\text{Subject to} \quad \sum_{v\in Adj(u)} \alpha(u,v)=1$$

After constructing the graph G_1, we apply the zero slack assignment algorithm which tries to assign the white space of the underutilized bins to the over utilized bins considering minimum perturbation constraint. After this assignment is done, some of the IP blocks inside each bin should move in the reverse direction from the over utilized bins to the under utilized bins. The total area of the IP blocks which are moving from one bin to another bin equals to the white space (slack) allocated from the latter bin to the former bin.

Table 1. The congestion reduction for ISPD02&ISPD05 benchmark s

Test Circuit	Overflow (%)			Wirelength		
	Before	After	Dec	Before	After	Inc (%)
Average	44.23	4.53	**39.70**	93.10	103.88	10.92

We remove congested areas by distributing IP blocks such that the total overflow becomes almost zero. To actually find the best set of IP blocks to move we formulate a 0-1 knapsack problem and apply an effective heuristic used to solve 0-1 knapsack problem [70]. We have implemented this method and have tested it using two sets of academic benchmarks ISPD02 and ISPD05 [25, 28]. The average congestion reduction can be seen in Table 1.

4.1 Open Problems

There are several open problems which should be researched in this area such as:

- Incorporating the issues regarding thermal placement and hot spot distribution in congestion alleviation algorithms.
- Considering the white space needed around the big IP blocks in the congestion removal algorithms to guarantee routability in later phases.

5. STATISTICAL TIMING ANALYSIS FOR DELAY PREDICTION

Traditionally, the variation in the underlying process parameters have been modeled in statistical timing analysis (STA) using so-called case analysis. Best-case, nominal and worst case SPICE parameters sets are constructed, and the timing analysis is performed several times. Each execution of statistical timing analysis is therefore deterministic, meaning that the analysis uses deterministic delays for the gates and any statistical variation in the underlying silicon is hidden. On the other hand, the delay of a design may statistically change because of relocation of logic elements due to congestion prediction and removal and so the design timing characteristics can be viewed as a statistical function of the cell locations. Moreover, due to the dominance of wire delay over cell delay in today's designs, the impact of the wirelength on the delay of the circuit should be investigated as well. We will model and consider this type of uncertainty in timing characteristics of circuits and its relationship with other physical design approaches such as congestion prediction, utilization control and wirelength prediction to have an early enough prediction of design delay for timing-driven placement.

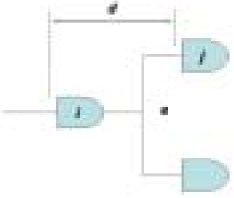

Figure 9: Delay model used in placement

Process variations are due to uncertainty in the device and interconnect characteristics, such as effective gate length, doping concentrations, oxide thickness and ILD thickness. In general, these variations can be divided into between-die variations (or inter-die variation) and within-die variations (or intra-die variations). Within die variations can have a deterministic component due to topological dependencies of device processing, such as CMP effects and lithograph distortions. In some cases, such topological dependencies can be directly accounted for in the analysis whereas in other cases such variations are treated as random. Specifically, placing different blocks at various locations result in different timing and delay behavior of the circuit.

Manufacturing process variations result in gate delay discrepancies. In order to perform an accurate timing analysis, these variations must be considered. Key parameters in device variations are channel length L, threshold voltage V_t and oxide thickness t_{ox}. Process parameter variations are assumed to be normally distributed random variables. Under these assumptions, gate delay, d, will have a probability distribution function $f_{gate}(d)$ which is a Gaussian distribution with parameters μ and σ. In statistical timing analysis, the deterministic delay assigned to a gate, D_{gate}, is chosen such that:

$$\Pr ob\left(d > D_{gate}\right) < 0.05 \qquad (10)$$

This conveys that the probability with which gate delay exceeds D_{gate} is less than 0.5 percent. D_{gate} is used for worst-case timing analysis which often acts too conservatively. The effect of wire delay within critical timing paths is increasingly becoming a problem. By comparing the large improvement of transistor performance, due to shrinking L_{eff} new technology such as silicon on insulator, versus the smaller improvements of wire delay, such as copper wires and better dielectrics, it can be seen that the wiring within an advanced microprocessor will become a more dominant portion of the critical paths. In deep sub- micron designs it is crucial to analyze and improve any wire dominated paths assuming that the transistor delay continues to improve.

Timing-driven placement is one of the most important steps to meet the circuit performance in VLSI design. The problem has been studied for more than two decades, yet it remains challenging because of the dramatically increasing circuit size and the dominance of the interconnect delay in deep sub-micron design. Traditionally, successful timing-driven placement techniques fell into three categories: path-based algorithms, net weighting algorithms and delay budgeting algorithms [67, 68, 73, 74, 75, 76]. In timing-driven placement, traditionally we use the delay model shown in Figure 9:

$$d = I_i + r\left(c_e + c_g\right) + r_e c_j \qquad (11)$$

Where I_i is the intrinsic delay of the gate i, r is the driver resistance of gate i, c_e and r_3 are the capacitance and resistance of the net e, respectively. c_j is the input pin capacitance of gate j. If gate i drives multiple gates, the summation of c_j for all fan-out gates replaces c_j [13, 14].

Previously, we used the wireload model to estimate the resistance of a net. Half the perimeter of the bounding box of the net was used to estimate wire length. We will include the intrinsic gate delays in our library, whereas before we used the result from linear regression in the data in library look-up table. As described before, many of the factors in the delay of the circuit are actually random. We will use probabilistic and statistical analysis of the delay during our timing-driven placement. Parameters defining the random variables are characterized depending on the type of technology used and its process variations. As in traditional timing analysis, finding the delay distribution of a DAG may be handled with two operators: Max and Sum.

Both of these operations are easy to handle under deterministic analysis. Unfortunately there exists no set of distributions which is closed under both sum and max. This fact makes it almost impossible to find a mathematically closed form for the total delay distribution. [29] and [30] study methodologies for finding the total delay distribution of a circuit. Their proposed techniques, however, are computationally expensive. On the other hand, reconvergent paths produce correlations between delay distributions of the paths that should be considered. Yet, if the distributions are discrete, both max and sum of two discrete random variables can be evaluated in polynomial time. Suppose *d* is a discrete random variable with values from set *Q* and probabilities from set *P* such that:

$$\begin{aligned} &prob(d = q_i) = f(q_i) = P_i, i = 1 \ldots N \\ &where\, q_i \in Q, P_i \in P\, and\, |Q| = |P| = N \end{aligned} \qquad (12)$$

where *f* represents the probability distribution function(PDF) of delay *d*. Moreover, p_i's and q_i's denote the probability of occurrence of delays and their value respectively. For further evaluation, a generating function is defined over every random variable *d* as follows:

$$G_d(x) = \sum_{i=1} P_i x^{d_i} \qquad (13)$$

It suffices to define the max operator for only two operands since the maximum of *n* random variables can be defined recursively as follows:

$$D = \max(d_1, d_2, \ldots, d_n) = \max(\max(d_1, d_2, \ldots, d_{n-1}), d_n) \qquad (14)$$

where d_i's are independent random variables.

Now the distribution of the maximum (max) and summation (sum) of two random variables can be computed. Assume we are given two discrete random variables *u* and *v*:

- $prob(u = q_i) = P_i, i = 1 \ldots N,$
 $where\, q_i \in Q, P_i \in P\, and\, |Q| = |P| = N$
- $prob(v = q'_j) = P'_j, i = 1 \ldots M,$
 $where\, q'_j \in Q', P'_j \in P'\, and\, |Q'| = |P'| = N$

Define *w* = *sum* (*u*,*v*). The distribution of *w* is computed as below:

$$prob(w = \theta = q_i + q'_j) = p_i . p'_j \qquad (15)$$

q_i and q_j' are selected from sets *Q* and *Q'* and where p_i and p_j' are their probability of occurrence respectively. From Equation 15 in can be inferred that the generating function for *w* is the convolution of the generating functions of *u* and *v*. The size of *w* (the number of its discrete values) would be at most $N \times M$. However, if the type of gate delay distribution in the graph is limited, it is easy to show that the size of the summation over *n* variables would be polynomial in terms of *n*.

The *Max* operator works in a different manner. Suppose *z* is defined to be the maximum of two random variable *u* and *v*: *z* = *max*(*u*,*v*). Therefore, *z* is only equal to either the value of *u* or *v*. In other words, if $z \in Q \cup Q'$

$$\begin{aligned} prob(z = \theta) = {} & prob(u = \theta).prob(v \le \theta) + \\ & prob(v = \theta).prob(u \le \theta) \end{aligned} \qquad (16)$$

The size of *z* is at most *N* + *M* which indicates that the size of the total distribution will be polynomial in terms of the number of nodes in the graph.

The above observations raised the need for statistical analysis of circuit delay during the placement process by considering the impact of process variation, cell relocation due to congestion utilization control and wire delay due to length of interconnection.

5.1 Open Problems

Half perimeter approximation for the length of a net is no longer accurate since branches in a net, produce statistical dependencies. From the general properties of the nets, such as length and branch types, we propose several open problems to consider:

- Obtain a probabilistic model for obtaining the topology of the nets with reconvergent fan-outs to statistically analyze delay distribution for nets during placement.
- Incorporating the probability distribution of delays into our placement algorithm for the purpose of timing-driven placement.

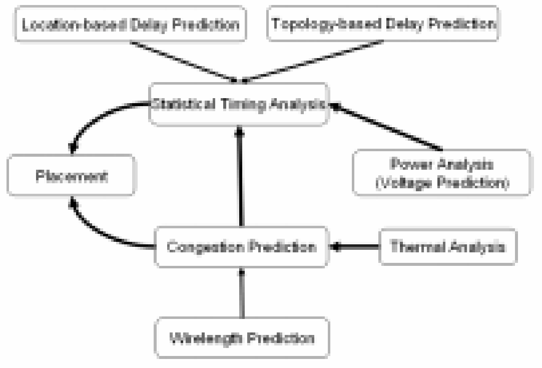

Figure 10: Integrated Prediction and Design Diagram

6. INTEGRATED PREDICTION

The growing sophistication of applications is continually pushing the design and manufacturing of integrated circuits (IC) and electronic systems to new levels of complexity.

As the complexity of IC designs increase, the traditional approaches to physical design problems will cease to work , and hence simple extensions to the existing CAD tools will not be useful. To handle the complexities, systems will need to be designed at higher levels of abstraction by having a good prediction of the effects of different issues early in the design cycle. As a result, system design will not be viable for large SoC designs without hierarchical CAD flows, which considers the effect of several parameters such as wirelength, congestion, thermal issues and power early in design flow. This is done to avoid ending up with a designs, which may severely violate one or more of these concerns, since it would be too late to tackle these problems deep in the design flow. Having a reasonable integrated estimation of all these problems is inevitable if we want to handle the large complexity of future designs. As an example, considering pure wirelength minimization without congestion estimation methods may lead to unroutable designs. As another example, applying wirelength minimization along with considering congestion removal but ignoring thermal issues may lead to designs with several hot-spots which would have low thermal reliability. As a result, integrated prediction of different influential problems would be of great importance to maintain a high quality design flow. Figure 10 shows the block diagram of our proposed approaches for integrating different prediction methods to improve the design quality in general.

As is shown in Figure 10, all these predictions are highly integrated with each other with the goal of degrading the harmful side effects of optimizing each parameter individually at the expense of the other parameters.

6.1 Open Problems

- Integrating wirelength, congestion, delay into our placement tool Dragon [3, 7, 8, 26, 62] by considering thermal and power optimization issues.

7. REFERENCES

[1] Taghavi T. and Sarrafzadeh M., System Level Estimation of Interconnect Length in the Presence of IP Blocks, in *Proc. International Symposium on Quality of Electronic Design,* 2007.

[2] Taghavi T., Amelifard B., and Sarrafzadeh M., Hierarchical wirelength estimation for large-scale circuits in the presence of IP blocks, under review *for IEEE Transaction on Very Large-Scale Integration System*, available at http://er.cs.ucla.edu/~taghavi/tech_report_0806.pdf

[3] Taghavi T., Yang X., Choi B.K., Wang M., and Sarrafzadeh M., Dragon: Congestion Minimization in Modern Placement Circuits, *To appear in ISPD Placement Contest Book, Springer*, Anticipated Publication Date: 2007.

[4] Taghavi T. and Sarrafzadeh M., Concurrent Congestion and Wirelength Estimation in the Presence of IP Blocks, *under review for IEEE Transactions on Computer Aided Design*, 2007.

[5] Yang X., Choi B-K., and Sarrafzadeh M., Routability Driven White Space Allocation for Fixed-Die Standard-Cell Placement, in *Proc. International Symposium on Physical Design, pp.* 42-47, 2002.

[6] Taghavi T. and Sarrafzadeh M., Blockage Oriented Placement, *IEEE Electronic Design Process Workshop*, 2006.

[7] Taghavi T., Yang X., Choi B.K., Wang M., and Sarrafzadeh M., Dragon2006: Blockage-Aware Congestion-Controlling Mixed-Sized Placer, in *Proc. International Symposium on Physical Design*, 2006.

[8] Taghavi T., Yang X., Choi B.K., Wang M., and Sarrafzadeh M., Dragon2005: Large Scale Mixed-Sized Placement Tool, in *Proc. International Symposium on Physical Design*, pp. 245-247, 2005.

[9] Donath W. E., Placement and average interconnection lengths of computer logic, *IEEE Transactions on Circuits and System*, vol. CAS-26, pp. 272–277, Apr. 1979.

[10] Cheng C. –K., Kahng A. B., Liau B., and Stroobandt D., Toward better wireload models in the presence of obstacles, in *Proc. Asia and South Pacific Design Automation Conference*, pp. 527–532, 2001.

[11] Yang X., Kastner R., and Sarrafzadeh M., Congestion estimation during top-down placement, *IEEE Transactions on Automatic Control*, pp. 100-108, 1999.

[12] Saeedi M., Saheb Zamani M., and Jahanian A., Prediction and Reduction of Routing Congestion, in *Proc. International Symposium on Physical Design*, pp. 72-77, 2006

[13] Amelifard B., Fallah F., and Pedram M., Low-power fanout optimization using MTCMOS and multi-Vt techniques, in *Proc. of International Symposium on Low Power Electronics and Design*, pp. 334-337, 2006.

[14] Amelifard B., Fallah F., and Pedram M., Low-power fanout optimization using multiple threshold voltage inverters, in *Proc. of International Symposium on Low Power Electronics and Design*, pp. 95-98, 2005.

[15] Lou J., Thakur S., Krishnamoorthy S., and Sheng H. S, Estimating routing congestion using probabilistic analysis, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 21. No. 1, pp. 32-41, Jan. 2002.

[16] Wang M., Yang X., Eguro K., and Sarrafzadeh M., Multicenter congestion estimation and minimization during placement, in *Proc. International Symposium on Physical Design*, pp. 147-152, 2000.

[17] Yang X., Bozorgzadeh E., and Sarrafzadeh M., Wirelength estimation based on Rent exponents of partitioning and placement, in *Proc. International Workshop on System-Level Interconnect Prediction*, pp. 25-31, 2001.

[18] Landman B. S. and Russo R. L., On a pin versus block relationship for partitions of logic graphs, *IEEE Transactions on Computer*, C-20, pp. 1469-1479, Dec. 1971.

[19] Stroobandt D. and Campenhout J. V., Accurate interconnection length estimations for predictions early in the design cycle, *IEEE Transactions on Very Large Scale Integration Systems, Special Issue on Physical Design in Deep Submicron*, pp. 1-20, 1999.

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

[21] Cheng C.-K., Kahng A. B., Liu B., and Stroobandt D., Toward better wireload models in the presence of obstacles, *IEEE Transaction on VLSI Systems*, vol. 10, pp. 177–189, Apr. 2002.

[22] Kahng A.B. and Xu X., Accurate pseudo-constructive wirelength and congestion estimation, in *Proc. International Workshop on System Level Interconnect Prediction*, pp. 61-68, 2003.

[23] Westra J., Bartel C., and Groeneveld P., Probabilistic congestion prediction, in *Proc. International Symposium on Physical Design*, pp. 204-209, 2004.

[24] Wang M. and Sarrafzadeh M., On the behavior of congestion minimization during placement, in *Proc. International Symposium on Physical Design*, pp. 145-150, 1999.

[25] http://www.sigda.org/ispd2005/contest.htm

[26] http://er.cs.ucla.edu/Dragon

[27] http://www.wolfram.com

[28] http://vlsicad.eecs.umich.edu/BK/ISPD02bench

[29] Jyu H.-F, Malik S., Devadas S., and Keutzer K.W., Statistical timing analysis of combinational logic circuits, *IEEE Transaction on Very Large Scale Integration (VLSI) Systems*, vol. 1, pp. 126 - 137, Jun. 1993.

[30] Robert Sr. and Hitchcock B., Timing verification and the timing analysis program, in *Proc. Design Automation Conference*, pp. 594 - 604. 1982.

[31] Bakoglu H. B., *Circuits, interconnections, and packaging for VLSI*, Reading, MA: Addison-Wesley, 1990.

[32] Sherwani N., *Algorithms for VLSI physical design automation*, Norwell, MA: Kluwer Academic Publisher, 1995.

[33] Cotter J. and Christie P., The analytic form of the length distribution function for computer interconnection, *IEEE Transaction on Circuits and Systems*, vol. 38, pp. 317-320, 1991.

[34] Stroobandt D., van Marck H., van Campenhout J., An accurate interconnection length estimation for computer logic, in *Proc. Great Lake Symposium on VLSI*, pp. 50-55, 1996.

[35] Hamada T., Cheng C. -K., and Chau P. M., A wire length estimation technique utilizing neighborhood density equations, in *Proc. Design Automation Conference*, pp. 57-61, 1992.

[36] Ozaktas H. M., Paradigms of connectivity for computer circuits and networks, *Optical Engineering*, vol. 31, pp. 1563-1567, 1992.

[37] Russo R. L., On the tradeoff between logic performance and circuit-to-pin ratio for LSI, *IEEE Transaction on Computers*, vol. C21, pp. 147-153, 1972.

[38] Donath W. E., Equivalence of memory to random logic, *IBM Journal of Research and Development*, vol. 18, pp 401-407, 1974.

[39] Hagen L., Kahng A. B., Kurdahi F. J., and Ramachandran C., On the intrinsic rent parameter and spectra-based partitioning methodologies, *IEEE Transaction on Computer Aided Design*, vol. 13, pp. 27-37, 1994.

[40] van March H. and van Campenhout J., Modeling and evaluating optoelectronic architectures, in *Proc of SPIE Optoelectronics II*, vol. 2153, pp. 307-314, 1994.

[41] March H. V., Stroobandt D., and Campenhout J. V., Interconnection length distributions in 3-dimentional anisotropic systems, in *Proc. International Conference on Applied Informatics*, IASTED-ACTA Press, pp. 98-101, 1995.

[42] Stroobandth D. and Campenhout J. V., Estimating interconnection lengths in three-dimentional computer systems, *IEICE Transaction on Information and Systems*, *Special Issue on Synthesis and*

Verification of Hardware Design, vol. E80-D, pp. 1024-1031, 1997.

[43] Sutherland I. E. and Oestreicher D., How big should a printed circuit board be?, *IEEE Transaction on Computers*, vol. C-22, pp. 152-155, 1981.

[44] Pedram M. and Preas B., Interconnection length estimation for optimized standard cell layouts, in *Proc. IEEE International Conference on Computer Aided Design*, pp. 390-393, 1991.

[45] Donath W. E., Wire length distribution for placements of computer logic, *IBM Journal of Research and Development*, vol. 25, pp. 152-155, 1981.

[46] Stroobandt D., March H. V., and Campenhout J. V., An accurate interconnection length estimation for computer logic, in *Proc. Great Lakes Symposium on VLSI*, pp. 50-55, 1996.

[47] Ferry D. K., Interconnection lengths and VLSI, *IEEE Circuits and Devices Magazine*, vol. 1, pp 39-42, 1985.

[48] Gura C. V. and Abraham J. A., Average interconnection length and interconnection distribution based on Rent's Rule, in *Proc. Design Automation Conference*, pp. 574-577, 1989.

[49] Christie P., Stroobandt D., The interpretation and application of Rent's Rule, *IEEE Transaction on Very Large Scale Integration Systems*, vol. 8, pp. 639-648, 2000

[50] Stroobandt D., Analytical methods for a priori wire length estimates in computer systems, PhD Thesis, Gent University, 1998.

[51] Fiduccia C. M. and Mattheyses R. M., A linear-time heuristic for improving network partitions, in *Proc. Design Automation Conference*, pp. 175-181, 1982.

[52] Kirkpatrick S., Gelatt C. D., and Vecchi M. P., Optimization by simulated annealing, *Science*, vol. 220, pp 671-680, 1983.

[53] Keyes R., *The physics of VLSI systems*, Reading, MA: Addison-Wesley, 1987.

[54] Sarrafzadeh M., Wang M., Yang X., *Modern placement techniques*, Norwell, MA: Kluwer Academic Publishers, 2003.

[55] Sechen C. and Sangiovanni-Vinecentelli A., The Timberwolf3.2: a new standard cell placement and global routing package, in *Proc. Design Automation Conference*, pp. 432-439, 1986.

[56] Sigl G., Doll K., Johannes F. M., Analytical placement: a linear or a quadratic objective function, in *Proc. Design Automation Conference*, pp. 427-432, 1991.

[57] Wang M. and Sarrafzadeh M., Behavior of congestion minimization during placement, in *Proc. International Symposium on Physical Design*, pp. 147-152, 2000.

[58] Yang X, Choi B.K, Sarrafzadeh M, Routability driven white space allocation for fixed-die standard-cell placement, in *Proc. International Symposium on Physical Design*, pp. 42-47, 2002.

[59] Cheng C. E., RISA: accurate and efficient placement routability modeling, in *Proc. International Conference on Computer-Aided Design*, pp. 690-695, 1994.

[60] Parakh P. N., Brown R. B., and Sakallah K. A., Congestion driven quadratic placement, in *Proc. Design Automation Conference*, pp. 275-278, 1998.

[61] Hwang F. K., Richards D. S., and Winter P., The Steiner tree problem. *Annals of Discrete Mathematics*, 1992. Elsevier Science Publishers.

[62] Wang M., Yang X., Sarrafzadeh M., Dragon2000: fast standard-cell placement for large circuits, in *Proc. International Conference on Computer-Aided Design*, pp. 260-263, 2000.

[63] Yang X., Kastner R., and Sarrafzadeh M., Congestion estimation during top-down placement, on *Proc. International Symposium on Physical Design*, pp. 573-576, 2001.

[64] Tsay R. S., Chang S. C., and Thorvaldson J., Early wireability checking and 2-D congestion-driven circuit placement, in *Proc. International ASIC Conference*, pp. 50-53, 1992.

[65] El Gamal A. A., and Syed Z. A, A stochastic model for interconnections in custom integrated circuits, *IEEE Transaction Circuit and Systems*, vol. 28, pp. 888-894, 1981.

[66] Cheng L. and Song X., et al., A fast congestion estimator for routing with bounded detours, in *Proc. Asia and South Pacific Design Automation Conference*, pp. 666-670, 2004.

[67] Yang X., Choi B-K., and Sarrafzadeh M., Timing-driven placement using design hierarchy guided constraint generation, in *Proc. International Conference on Computer-Aided Design*, pp. 177-180, 2002.

[68] Hamada T., Cheng C. K., and Chau P. M., Prime: a timing-driven placement tool using a piecewise linear resistive network approach, in *Proc. Design Automation Conference*, pp. 531-536, 1993.

[69] Nair R., Berman C. L., Hauge P. S., and Yoffa E. J., Generation of performance constraints for layout, *IEEE Transactions on Computer-Aided Design*, CAD-8(8), pp 860-874, Aug. 1989.

[70] Martello S., and Toth P., *Knapsack problems: algorithms and computer implementations*, Wiley-interscience series in discrete mathematics and optimization, 1990.

[71] Jackson M. A. B. and Kuh E. S., Performance-driven placement of cell based IC's, in *Proc. Design Automation Conference*, pp. 370-375, 1989.

[72] Ghiasi S., Choudhuri S., Bozorgzadeh E., Sarrafzadeh M., A unified theory for timing budget management, in *Proc. International Conference on Computer-Aided Design*, pp. 653—659, 2004.

[73] Swartz W. and Sechen C., Timing driven placement for large standard cell circuits, in *Proc. Design Automation Conference*, pp. 211-215, 1995.

[74] Burstein M. and Youssef M. N., Timing influenced layout design, in *Proc. Design Automation Conference*, pp. 124-130, 1985.

[75] Dunlop A. E., Agrawal V. D., Deutsch D. N., Jukl M. F., Kozak P., and Wiesel M., Chip layout optimization using critical path weighting, in *Proc. Design Automation Conference*, pp. 133-136, 1984.

[76] Gao T., Vaidya P. M., and Liu C. L., A new performance driven placement algorithm, in *Proc. International Conference on Computer-Aided Design*, pp. 332-335, 1991.

[77] Griffith J., Robins G., Salowe J. S., and Zhang T., Closing the gap: near-optimal Steiner trees in polynomial time, *IEEE Transaction on Computer-Aided Design*, vol. 13, pp. 1351–1365, Nov. 1994.

[78] Sarrafzadeh M., Knol D. A., and Tellez G. E., A delay budgeting algorithm ensuring maximum flexibility in placement, *IEEE Transactions on Computer Aided Design*, vol. 16, pp. 1332-1341, 1997.

[79] Zhou H., Efficient Steiner tree construction based on spanning graphs, in *Proc. International Symposium on Physical Design*, pp. 152–157, 2003.

[80] Chu C., FLUTE: fast lookup table based wirelength estimation, in *Proc. International Conference on Computer Aided Design*, pp. 696-701, 2004.

[81] Cong J., He L., Koh C-K., and Madden P. H., Performance optimization of VLSI interconnect layout, *Integration, the VLSI Journal*, 1996.