Non-Parametric Statistical Methodology for Wire-length Prediction

Jennifer L. Wong, Azadeh Davoodi, Vishal Khandelwal, Ankur Srivastava, and Miodrag Potkonjak

UCLA Technical Report #050018
May 16, 2005

***Abstract*— We address the classic wire-length estimation problem and propose a new statistical wire-length estimation approach that captures the probability distribution function of net lengths after placement and before routing. These types of models are highly instrumental in formalizing a complete and consistent probabilistic approach to design automation and design closure where along with optimizing the pertinent cost function, the associated prediction error is also considered.**

The wire-length prediction model was developed using a combination of parametric and non-parametric statistical techniques. The model predicts not only the length of the net using input parameters extracted from the floorplan of a design, but also probability distributions that a net with given characteristics after placement will have a particular length. The model is validated using the learn-and-test and resubstitution techniques.

The model can be used for a variety of purposes, including the generation of a large number of statistically sound and therefore realistic instances of designs. We applied the net models to the probabilistic buffer insertion problem and obtained substantial improvement in net delay after routing (∼20%) when compared to a traditional bounding box-based buffer insertion strategy.

I. Introduction

Wire-length has become one of the most critical metrics in physical design primarily due to the rise of the deep submicron era. Therefore, there is a strong need for early estimation and optimization of this design parameter. A large amount of research has been directed towards the development of accurate models for the estimation of this important design objective [1], [2], [3], [4], [5]. Additionally, accurate timing and routability estimation [6], [7] relies on these models.

Estimating the exact wire-length for each net in the circuit is a very challenging problem. There are a large number of different parameters and constraints, such as the bounding box of the net, number of routing grids and the grid capacity, and the total number of nets routed in the vicinity of the pertinent net, that are all potentially relevant, but typically are very hard to capture into a consistent wire-length model. Hence, estimating an exact value for wire-length is a very difficult problem. Similar difficulty in estimation has also been widely recognized for other critical metrics of deep submicron designs such as power, delay, noise immunity, and crosstalk. Therefore, synthesis optimization is typically performed in the presence of high degrees of estimation inaccuracy. The optimization decisions made in such a scenario are typically sub-optimal and often result in failure of design closure. In order to solve this problem, a new design automation paradigm is gaining steam in which unpredictable design objectives are modeled probabilistically and the overall design is also optimized probabilistically. For the success of such an approach, we need accurate models which probabilistically estimate the critical design objectives. In order to address this need, we have developed a novel statistical modeling methodology for capturing wire-length in the post placement pre-routing phase.

The model uses data that can be extracted once the placement of the designs is completed. In order to build the wire-length prediction model we used a combination of parametric and non-parametric techniques [8], [9]. Since the approach to developing the wire-length prediction model is generic and can be applied to other early estimation tasks in synthesis, we provide a detailed description of how the models were derived. Although statistical techniques have demonstrated their potential in many fields, they have rarely been used in synthesis and CAD tools. This is surprising considering their advantages. For example, they produce models that are both mathematically sound and that extract the maximal possible amount of information from the collected data. Note that non-parametric statistical techniques are applicable on any set of data with no prior assumptions about their distribution. Furthermore, statistical techniques provide a means for evaluation and validation of the obtained models as well as techniques and tools for establishing intervals of confidence on the overall model and any of its subparts. The standard and practical references for parametric and non-parametric statistical techniques that explain in detail many of the concepts, techniques, and algorithms used in this paper include [10], [11], [12]. Although our overall statistical modeling approach is new and several steps are unique, other steps are adopted from the modern statistical practice. Finally, it is important to emphasize that the developed statistical model is validated both statistically and through a driver application - buffer insertion for clock cycle optimization.

Statistical estimation and prediction methodology and models can be used in many ways. For example, one can use the prediction information to evaluate the suitability of a particular floorplan for obtaining final routing where nets satisfy a particular user specified condition. For instance, the goal can be to determine which among a number of competing floorplans is most likely to result in a final design with a few long nets or to minimize the overall sum of wirelengths. The models are also a natural component of the overall probabilistic design automation methodology. One such probabilistic algorithm is [13] which performs buffer insertion assuming wire-lengths which are estimated as distributions. We used our models in the probabilistic buffer insertion approach of [13] and obtained massive improvements in net delay (∼20%) after routing when compared with a traditional bounding box-based traditional bounding box strategies [14], [15].

It is both interesting and important to compare this work to the work of Davoodi et al [16]. In the previous work, the authors build an empirical model for estimating the probability distribution of wirelength for all nets of a design. The model is built using intuition and insight and uses the half perimeter bounding box (HPBBOX) measurements as the sole prediction property. There are five major differences between the previous paper and the proposed work. First, the previous work developed the model using intuition, while in this work we present an approach which is directly based on parametric and non-parametric statistical techniques. Secondly, only a single prediction parameter, HPBBOX, was used in the previous work, while the proposed approach uses five prediction variables taken from a set of fifteen proposed prediction variables. In this work, we validate the accuracy of the proposed statistical models using standard resubstitution techniques, while in the previous paper they used intuition-based techniques for evaluation of the model. Furthermore, the model in [16] does not detect outliers, while one of our primary goals is to accurately predict wires that have disproportional long length with respect to their prediction variables. Finally, and most importantly we have different objectives. In Davoodi et al [16] the goal is to derive a probability distribution for net lengths. Therefore, they are not concerned with the prediction of the most likely net length for each net, but only for the whole ensemble of the nets. On the other hand, in this work the primary goal is to exactly provide the probability that any given net will have a particular length. Therefore, these works are complementary in their objective as well as the used derivation and validation approaches.

The rest of this paper is organized in the following way. In order to provide a global view of the approach and make the paper self-contained we start by summarizing the probabilistic synthesis paradigm. Next, we describe our statistical modeling procedure and present the developed wire-length estimation model. After that the model is evaluated using both the learn-and-test and resubstitution validation methodologies. Finally, we present the application of our model to the task of probabilistic buffer insertion.

II. Probabilistic Synthesis Paradigms

Automation of integrated systems is marred with estimation inaccuracies which occur due to a combination of many factors. Unawareness of exact layout information such as routing, placement, and exact logic structure are among prominent reasons. In addition, as technology features shrink in deep submicron, in particular below 70nm, manufacturablity becomes an important issue that often significantly impacts performance and even the correctness of the design. Economically achievable margins of tolerance are too low and different instances of manufactured integrated circuits, even on the same wafer, can have significantly different speed or power consumption. Therefore, during all design phases it is advantageous to consider manufactuability using approaches that statistically produce designs within specified timing and power constraints often enough. In the light of such unpredictabilities, a traditional deterministic approach towards design automation often becomes incapable and obsolete. A deterministic approach assigns a fixed value to the cost function (like area, delay, power, wire-length) and does not consider the error associated with the estimation of this cost function. Hence, very little can be said about the optimality of the final design especially if the estimation was erroneous. This issue calls for the development of a probabilistic approach towards design optimization. Such an approach models the cost functions as probability distributions and optimizes the design probabilistically, hence maximizing the likelihood of satisfying design constraints. A number of researchers [17], [18], [19], [20], [21] have suggested the importance of such an approach due to the fact that estimation inaccuracies (both due to fabrication variability and layout unawareness) are becoming major bottlenecks in design closure. The main advantage of such an approach is faster design closure, better fabrication yield (since fabrication variability would be accounted for during designing) and improved robustness.

The main prerequisite for the application of a probabilistic synthesis technique which considers uncertainties, is the availability of accurate prediction techniques. Currently, these models are mainly built manually using deep insights into the design process. However, these non-statistical methods are rarely statistically tested for their accuracy. We propose the use of modern statistical techniques not only to automatize the development of models and the selection of the most accurate models, but also to provide sound mathematical estimates of their accuracy.

III. Statistical Modeling for Wire-length Prediction

In this section, we present a statistical approach for predicting the length of a given net on a specified chip that is characterized using a set of features that can be rapidly obtained after floorplanning. We begin by identifying the objectives and constraints of the problem. Next we discuss a set of net and chip features that are used as predictors to our model. The heart of the Section is the procedure used for the development of the wire-length prediction model. Additionally, the three phases of the procedure (robust linear regression [8], outlier detection, and establishment of probability distribution) are discussed. We then present a model for mapping between different designs. Finally, the evaluation of the proposed models is conducted using learn-and-test and resubstitution techniques [9], [22], [23].

A. Problem Formulation

Our primary objective is to predict the length of each net given a set of features that can be rapidly extracted from the floorplan of a chip. The goal is not only to predict the length, but also to quantitatively characterize the probability that the net will have a particular length after routing. Furthermore, the operational constraint is to only use features that can be extracted with low computational effort and can be rapidly analyzed with statistical techniques. The final major objective is to statistically validate all obtained results and to establish intervals of confidence on all deduced models and their parameters.

B. Characterization of Nets and Designs

The starting point for model development was the definition of relevant features of nets that are available after placement. We used two types of features: atomic and composite. Atomic features are ones that are directly extracted from the design. Composite features were created by combining atomic features using simple rules. Most often the composite rules were ratios of two atomic features.

We used a state of the art commercial placement and routing tool (Cadence) to collect data that is used to build our statistical models. We use the post placement information as input parameters for building the model for each net. The objective is to identify metrics that influence the post routing wire-length for each net. The basic intuition lies in the fact that the net length is inversely proportional to the amount of routing area available and directly proportional to the routing hardness. Furthermore, a net is hard to route if its available routing area is being claimed by other neighboring nets. Conceptually, a net is a neighbor of another net if their corresponding bounding box overlap. Therefore, two bounding boxes are called neighbors if they overlap. The goal is to build a statistical model using only a small set of parameters that can be easily and rapidly extracted from the placement. While computation of some features is straightforward, the computation of other parameters requires the use of several basic procedures from computational geometry [24]. For example, procedure Locate-Point-Neighbor (p, S) takes a point p and a set of rectangles S and calculates the subset of rectangles which overlap with this point. Procedure Locate-Rectangle-Neighbor(R, S) takes a rectangle R and a set of rectangles S and calculates the rectangles in S that overlap on R. Note that all used properties can be rapidly computed in low polynomial time. We have considered the following post placement properties of the nets.

π_1 **Number of Net Terminals**. The higher the number of terminals, most often the harder it is to route the net.

π_2 **Half-Perimeter Bounding Box (HPBBOX) for net** i. The HPBBOX is easy to compute and provided a lower bound on the real wire-length. However, this property does not capture the number of terminals well. More importantly, the bounding box is a function of only a small set of terminals.

π_3 **Minimal Spanning Tree (MST)**. MST is calculated using standard Kruskal's or Prim's algorithm. The property captures the best case scenario for routing while considering all terminals.

π_4 **Convex Hull (CHULL) of net** i**'s terminals**. CHULL envelopes the terminals. Many algorithms, including standard Graham's scan, can be used to calculate CHULL of a net. Runtime is $O(nlogn)$ if n is the number of net terminals. CHULL is in a sense a generalization of HPBBOX. Both MST and CHULL are often very strongly correlated with HPBBOX.

π_5 **Number of different terminals in the bounding rectangle of the net** i **(**NT_i**)**. This property aims to predict routing difficulty by analyzing the number of terminals from other nets that compete for the same routing resources - space.

π_6 **Total number of overlapping neighbors (**OV_i**)**. This property can be calculated using the procedure
Locate-Rectangle-Neighbor(R, S) (defined in the beginning of the section) and is trying to estimate the number of nets that compete with a given net for routing resources.

π_7 **White Space of net** i **(**WS_i. This predictor is a region on the design defined with respect to the bounding box (BBOX) of net i that does not overlap with the BBOX of any other nets. White space is basically the total available routing area in grids which is not potentially shared with other nets. The metric can be calculated using a strategy similar to the one used for the calculation of the previous property. The intuition is simple and clear: large white space is well correlated with higher chances for efficient routing of the net.

π_8 **Space Utilization Factor (SUF) for the net** i. Conceptually, the SUF parameter tries to calculate the amount of competition that exists for the routing resource for each net. Assuming that the net bounding box is the available area for net routing, we try to estimate the overall degree of competition that exists for this area. The bounding area of a net is divided into rectilinear regions based on the number of overlapping neighbors (nets or bounding rectangles). SUF is calculated

using the following formula:

$$SUF(Net_i) = NT_i * \sum_{R_{ij} \in R(Net_i)} (\frac{OV_{ij} * A_{ij} * P}{B_i}) \qquad (1)$$

$$\text{where } P = \sum_{\forall v} (1 - \frac{WS_k}{B_k}) \qquad (2)$$

where
v is the number of neighbors k which belong to region R_{ij}, $k \neq i$,
$R(Net_i)$ is the set of all regions for the net i,
B_i is the bounding box area for Net_i,
NT_i is the total number of terminals in the bounding box of the net, (these include the terminals for Net_i and also terminals of its neighbors that fall with the bounding area of the net)
A_{ij} is the area for region R_{ij} in the bounding area of Net_i,
OV_{ij} is the number of nets that overlap in region R_{ij} (excluding Net_i), and
WS_k is the white space of net k which is one of the nets (net k) that fall over region R_{ij}.

For a net Net_i the bounding box area is partitioned into regions by the number of bounding boxes that overlap on it (essentially regions R_{ij}). By definition, each region must have a net overlap of at-least one. The key intuition behind this metric is the fact that more overlapping regions in the bounding area of a net, implies more routing hardness. For each region R_{ij} we multiply the corresponding region area A_{ij} with the total number of nets that overlap on this region (excluding Net_i) and a parameter P. P captures the routing hardness for the nets that fall on the same region. It is calculated according to the equation above. This value is then normalized with the bounding area of Net_i and added over all regions. The value is then scaled with the total number of terminals that lie in the bounding area of Net_i. This parameter tries to capture the routing hardness for a net. Intuitively a net will be hard to route if a lot of terminals fall in its bounding region. Moreover if there are a lot of highly overlapped regions in the nets bounding area, the routing will be hard too. If the overlapping neighbors (as defined earlier) have smaller white space (which means they are themselves congested) then they will make the pertinent net congested too. This metric is calculated using the following procedure. First, we identify regions on the layout based on overlapping net bounding boxes. This is accomplished using an iterative execution of the procedure (defined in the beginning of the section) Locate-Point-Neighbor(p, S) (defined in the beginning of the section) for all grid points. Therefore, the running time of the procedure is proportional to $Grid_x Grid_y$T(Locate-Point-Neighbor), where $Grid_x$ and $Grid_y$ are the number of grid units in the x and y directions respectively. Note that, if the total number of grids is high, the procedure is relatively slow. In this case, we impose a coarser grid resulting in a faster runtime, however at the loss of accuracy. This procedure can be followed by calculating the parameter P (see the equations above) for all regions and summing them up for the net.

π_9 **Resource Competition Metric (RCM) for net** i. This is a composite property that aims to capture the congestion in regions where net i is most likely to be routed. We consider the set of regions, R, that is created after the bounding box of the net is split by considering overlaps with the bounding boxes of other nets. If we denote neighbors as $neigh$ and use the notation introduced for calculating SUF, the RCM is calculated using the following formula.

$$\sum_{R(Net_j)} \left(\frac{A_{ij}}{Area_i} - \sum_{neigh_k\ \ of R(Net_j)} \frac{A_{ij}}{Area_k}\right)) \tag{3}$$

Recall that the regions can be identified in $Grid_x Grid_y$T(Locate-Point-Neighbor) runtime and the above parameter can be calculated for each region and added up for the net. The key intuition behind this parameter is that if the value $\frac{A_{ij}}{Area_i}$ is high then the net i has a larger share of the region where as if $\frac{A_{ij}}{Area_k}$ for the neighbor k is high then that neighbor has a larger share of the region. The RCM value for a net is proportional to its share of the available routing area in a nets bounding rectangle.

π_{10} **RCM for overlapping neighbors of net** i. The property is calculated using the RCM procedure. The intuition is that if neighboring nets are very congested, they will induce higher difficulty for routing the pertinent net i.

π_{11} **Sum of RCMs for all neighbors of overlapping neighbors**. This complex measure enhances the scope of the previous metrics.

π_{12} **Amount of overlapping area with the net for all neighboring nets**. The rationale is that a higher ratio of overlap areas indicates increased hardness to route.

π_{13} **The number of common terminals of neighboring nets to net** i. This measure is positively correlated with the difficulty of routing net i.

π_{14} **Neighbor utilization factor (NUF)** is defined in the following way, where c_t is the common terminals, c_a is the common area, and n_a is the neighbor area.

$$\sum_{neigh_k\ \ of Net_i} \frac{c_t * c_a}{n_a} \tag{4}$$

π_{15} **Neighbor hardness factor (NHF)** defined in the following way.

$$\sum_{neigh_k\ \ of Net_i} c_t * c_a * RCM(k) \tag{5}$$

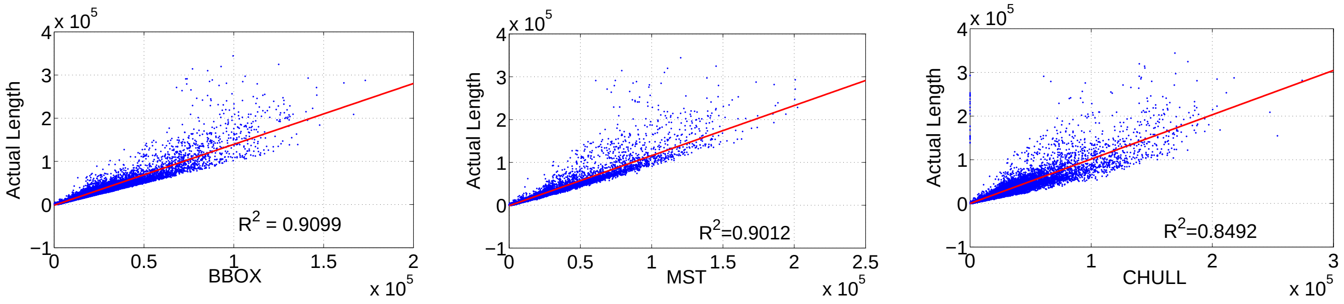
The last two properties aim to quantify the competition of neighboring nets with the net under consideration.

C. Overall Flow

In this Subsection, we present the overall flow of our statistical modeling procedure. Figure 2 summarizes the flow of the developed statistical modeling technique for prediction of the wire-lengths of the nets. The first step is the identification of relevant net properties. Two types of net properties are employed. The first group consists of properties related to the net itself. The second group consists of metrics that aim at predicting encountered congestion during routing of a given net due to the routing requirements of neighboring nets. On all properties we also applied a number of nonlinear transformations (e.g. application of logarithm function) in order to obtain better prediction abilities [25], [26]. Interestingly, while it is often reported in other fields that the use of non-linear transformations often greatly enhances accuracy of the model, for our model and our set of properties this was not the case.

The second step was data collection, or feature collection. All designs were routed using the Cadence placement and routing tool. Once the data was available, ie. actual length of the net and property values, we started with a randomly selected design and built a number of prediction models. In order to enable validation and evaluation of the statistical models, we used only 60% of the data to build the wirelength models. Note that it is important to have a set of data that is disjoint for these two tasks. Further explanation of this approach is given in [9], [11].

In Figure 1 we illustrate models built using HPBBOX, MST, and CHULL as individual prediction properties on the IBM07 design. The x-axis of each figure represents the property value, while the y-axis is the actual length of the net. It was immediately apparent that each of the following three features, bounding box (HPBBOX), minimum spanning tree (MST), and convex hull (CHULL), predicts the length of a majority of nets remarkably well, using a linear fit. We used the R^2 value to measure the accuracy of the feature's prediction ability. Specifically, the R^2 value is the square of residuals, i.e. difference between the predicted variable and its predicted value using an individual property. Each of the features (HPBBOX, MST, CHULL) had a R^2 value above 0.85 individually. The statistical t-test indicates that the probability that this correlation between the properties and actual net length is accidental is less than 10^{-16} in all three cases.



(a) HPBBOX (b) MST (c) CHULL

Fig. 1. Linear fit models for HPBBOX, MST, and CHULL properties on the IBM07 design.

TABLE I
BEST REGRESSION FITS ON IBM07 FOR VARIOUS PROPERTIES.

Property	Fit	R^2
π_2	2^{nd}	0.9099
π_3	2^{nd}	0.9012
π_4	2^{nd}	0.8492
π_5	2^{nd}	0.7024
π_6	2^{nd}	0.6984
π_7	Linear	0.0001
π_9	2^{nd}	0.3434
π_{11}	3^{rd}	0.0944
π_{14}	2^{nd}	0.5279
π_{15}	3^{rd}	0.1105

```
1. Feature Definition;
2. Feature Extraction;
3. Preliminary Data Exploration;
4. Features Evaluation and Normalization
   and Compound Feature Selection;
5. Net_Characterization {
6.        Nets Categorization;
7.        Preliminary Linear Regression on percentiles;
8.        Outliers Detection;
9.        Outliers Modeling;
10.       Final Linear Regression on percentiles; }
11. CDF and PDF model generation;
12. Chip characterization;
13. Development of Mapping Function to New Designs;
14. Evaluation and Validation;
```

Fig. 2. Modeling Approach Overall Flow.

In Table I we present the best fit regression model achieved for each property on the IBM07 design. In the first column we present the property, followed by the type of regression fit applied, and in the final column the best R^2 value achieved. Similar fits we achieved on other IBM designs. As the table shows, none of the other properties were able to predict wirelength as well as HPBBOX, MST, and CHULL.

While independently each of the property measures (HPBBOX, MST, CHULL) are strong predictors, their combination results in only marginally better prediction. Therefore, we decided to use the half-perimeter bounding box as the basis of our model because of its low computational cost.

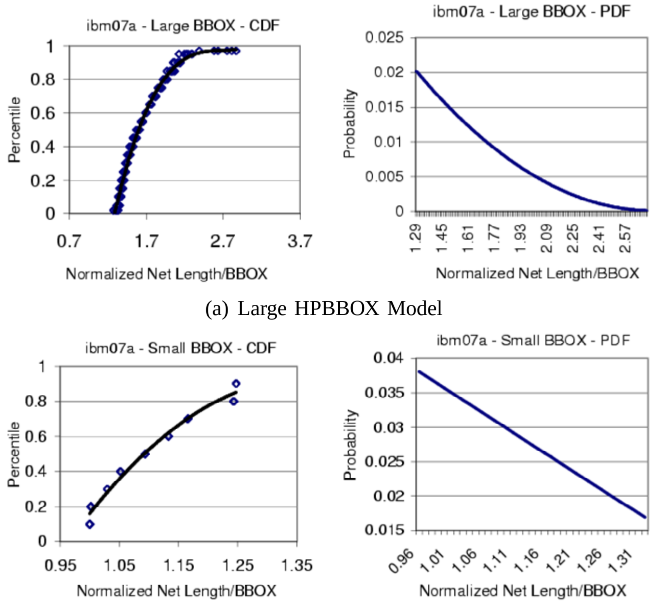
Closer examination of the data indicated that the behavior of nets with shorter length had different properties than longer nets. We performed analysis on the data set to determine a boundary value for these two groups. We partitioned the HPBBOX values using as the boundary all values between 1,000 and 9,000 at increments of 1,000. We found that the partitioning at 6,000 grid units performed superior with respect to all other boundaries. There is also strong indication that 6,000 is a good boundary because below 6,000 grid units we did not observe any outlier points. Unfortunately, we were not able to obtain convincing intuitive reason for the selected value. Additionally, we tried to partition the data into three groups but were not able to find more statistically sound models than those built on the two data sets when partitioned at 6,000 grid units. The statistical t-test indicates that correlation is significantly higher for the separated sets than for the overall set.

Once the data was divided into two sets, we conducted a linear regression-based procedure for fitting data for different percentiles. For each percentile (in the range of 10% to 90%) a separate fit is obtained and validated using the t-test. Next, to further enhances the accuracy of our model, we conduct an outliers detection procedure that identified a small subset of data that required specialized models. For this purpose we have developed a CART model [8]. Then we repeat linear regressions on the data after the outlier points were removed.

The next two steps were dedicated to the development of a probability distribution function (PDF) and cumulative distribution function (CDF) for wire-length prediction and interchip prediction. The goal of interchip prediction is to use global parameters of the chip in order to predict how features, such as global congestion and the number of nets and terminals, and the impact on PDFs for wire-length distribution. Finally, we conducted extensive model evaluation using learn-and-test and the resubstitution procedure in order to verify that the developed model is sound and no overfitting was done. In the rest of this Section, we elaborate on several key steps of the procedure.

D. Outlier detection

Outliers can be defined as nets that are not predicted well using a given set of features without significantly changing the complexity of the model. We detected the outliers using the following procedure. We begin by building our preliminary models. As candidates for outliers, we analyzed all points that differ from their prediction by more than k%. In our experimentation, we set k = 20%. Next, all the outlier candidate points are characterized according to each property. The separation value for each property is set in such a way that it



(a) Large HPBBOX Model

(b) Small HPBBOX Model

Fig. 3. Cumulative Distribution Function and Probability Distribution Function in IBM07 design for Cadence Router.

maximizes the ratio of outliers versus well predicted nets for the nets above (or below) the separation value. Note, that a linear-time sweep is sufficient to find this separation value.

All properties with their corresponding separation values are used as inputs to the non-parametric classification and regression tree (CART) software [8] to provide compact characterization of all outliers. The CART procedure resulted in the model where all nets are separated in three groups according to the number of terminals. The first group consisted of all nets with two terminals, the second with three, four, and five terminals, and the last group contained all other nets.

The final CART model used the following features: number of terminals, RCM of the net, RCM of overlapping neighbors, total number of overlapping neighbors, and the number of common terminals for a given net. The last four features were normalized against the area of the bounding box in order to achieve better separation. The overall misclassification rate for the detection of outliers was 6.7%. For the outlier nets, we build a separate linear regression fit, that had R^2 = 0.83. The t-test indicates that probability of accidental fit was less than 10^{-16}, clearly indicating the soundness of the model. It is interesting and important to emphasize that all outliers were corresponding to nets that were longer than standard predictions. This phenomenon can be easily explained by the intrinsic nature of the modeling problem. Relatively short nets for a given size of the half-perimeter bounding box (or MST or CHULL) are those that are routed using interconnect that is close to their theoretically possible minimum when no other nets cause congestion. In all designs for all values of half-perimeter bounding boxes, the number of nets with these properties was relatively large. A very high RCM was the best predictor of nets that will be routed using significantly higher length, in particular if the number of terminals was high.

One of the limitations of our model is that we did not explore systematically all possible predictors. Among intuitive potential candidates are layer assignment, that may better explain some of the outliers. This direction is one of the targets for future research efforts

E. CDF and PDF Generation

The goal of this phase is to find accurate cumulative distribution (CDF) and probability distribution functions (PDF) for the length of a net given the size of a corresponding HPBBOX. Note that partial information about the PDF and CDF is already contained in percentiles and therefore it is also contained in the percentile-based linear fit models. Therefore, the starting point for the PDF derivation was the percentile models for the ratio of the wire-length versus HPBBOX as a function of the size of the bounding box. For both small and large HPBBOX data, we used a resubstitution-based technique to obtain CDFs. Note that a PDF can be easily obtained from a CDF using either symbolic or numeric differentiation.

The PDF is built using the following procedure. First a subset of k nets are randomly selected for short nets. In our experimentation, we used value $k = 50\%$. The data is separated in bins that are dictated by HPBBOX values. The size of bin was determined in such a way that all bins contain the same number of points. The total number of bins was 10. The randomly selected subset of data is used to establish new percentile points for each bin containing data. All percentile points are normalized against the bounding box with shortest nets. The normalization is done in such a way that the average discrepancy between the values that correspond to the identical percentile is minimized. The data is fit using polynomials of low degree (three and four in our experimentation). The procedure is repeated a large number of times, the average value for each of percentile is calculated and fit using a least linear squares approach. This process was terminated once the percentile validation method indicated that we achieved user specified intervals of confidence for the PDF model. The same procedure is repeated for long nets. Figures 3(a) and 3(b) show intermediate and final results of the PDF derivation procedure.

F. Interdesign Modeling

Interchip prediction is the task where the objective is to predict properties using models that are invariant across different chips. Specifically, our goal is to predict wirelength on non-analyzed chips using properties of that chip that can be obtained after floorplanning and properties of nets and floorplans from chips that are used in the learning phases of the statistical procedure. Note that our goal is to predict the distribution of expected wire-length for nets of the design that are not used to build the statistical model. Therefore, once a model is built and validated for a single design, we must establish a means for rapid re-mapping of the wirelength model to other chips.

For this task, we considered the following atomic chip properties: (i) the area of the chip; (ii) the number of nets; (iii) the average and median of half-perimeter bounding box areas, MST, and convex hulls for all nets (iv) the average number of terminals per net; and (v) the percentage of the number of nets with a small number of terminals (two, three, or four). The composite chip metrics included ratios of all atomic chip properties and their simple statistical measures such as moments of low orders.

Table II shows the chip level characteristic of the designs. The first column denotes the name of the benchmark, followed by the number of chip layers and the number of nets in the benchmark. The fourth column denotes the total area of the chip. The overall congestion of the design is denoted in the fifth column by the total number of nets over the area of the design. The final column specifies the total number of terminals in the benchmark. Table III denotes the normalized average size of the HPBBOX, MST and CHULL for each net for each design. The statistics are normalized against the area of the chip.

In the first phase of the work, we statistically developed a wirelength prediction model for an individual design. A statistical model was feasible due to the large number of sample points (tens of thousands), however for interdesign modeling the number of available designs is limited. Therefore sound statistical practice strongly suggests not to attempt to build a statistical model on such a small dataset. It is for this reason we present an interdesign model built on intuition and consequently solely test the accuracy of the model using statistical techniques. Note, that validation of the model is possible in this case because resubstitution reuses sample points.

We denote by c_i and c_j the overall congestion of designs i and j measured by the normalized sum of convex hull area for each design divided by the total area of the design. Furthermore, we denote by NL_i and NL_j the number of layers used in designs i and j. Our model indicates that the length of the net in design i (L_i) can be calculated using the length of the net with the same HPBBOX in design j (L_j) using the following formula $L_i = L_j \frac{NL_j}{NL_i} * (\frac{C_i}{C_j})^{0.48}$. This model is built using least linear squares data fitting approach [27]. We built this model using a randomly selected subset of four designs. The model was validated against the remaining designs, as well as by using the resubstitution procedure as explained in the next Subsection.

In order to illustrate the goodness of fit for the interdesign model in Table IV we present the accuracy of prediction for the IBM07 model built using three measures (HPBBOX, MST, CHULL) on other IBM designs. The first column indicates the predicted design, while the other three columns present the R^2 error for models built using the HPBBOX, MST, CHULL properties respectively.

G. Evaluation and Validation

The last step of the modeling procedure was dedicated to the evaluation of the accuracy of the developed models. We followed two paradigms: learn-and-test and resubstitution [22], [23], [9]. In the case of the former procedure, we selected a subset of nets for building the model. This procedure was properly applicable only on modeling done on a single design, since the total number of available designs was too small statistically for sound application of this type of analysis on interchip models. Nevertheless, the application of the learn-and-test procedure on the interchip model indicates very high consistency, strongly implying that different designs follow very similar distributions of the wire-lengths for nets characterized by the selected features.

TABLE II

CHIP LEVEL CHARACTERISTICS FOR IBM DESIGNS OBTAINED USING CADENCE ROUTING AND PLACEMENT TOOL.

Bench	# layers	# nets	Area	$\frac{\#nets}{Area}$	Total Term
IBM01a	8	11507	5.89E+09	1.95E-06	44266
IBM01b	8	11507	5.72E+09	2.01E-06	44266
IBM02a	10	18429	7.65E+09	2.41E-06	78171
IBM02b	10	18429	7.31E+09	2.52E-06	78171
IBM07a	10	44394	1.63E+10	2.73E-06	164369
IBM07b	10	44394	1.55E+10	2.87E-06	164369
IBM08a	10	47944	1.76E+10	2.73E-06	198180
IBM08b	10	47944	1.67E+10	2.87E-06	198180
IBM10a	10	64227	2.97E+10	2.16E-06	269000
IBM10b	10	64227	2.82E+10	2.28E-06	269000
IBM11a	10	67016	2.31E+10	2.90E-06	231819
IBM11b	10	67016	2.19E+10	3.06E-06	231819
IBM12a	10	67739	3.44E+10	1.97E-06	284398
IBM12b	10	67739	3.26E+10	2.08E-06	284398

TABLE III

FLOORPLAN METRICS FOR IBM DESIGNS.

Bench	HPBBOX	MST	CHULL
IBM01a	1.30E-06	9.01E-07	1.08E-06
IBM01b	1.28E-06	8.85E-07	1.07E-06
IBM02a	6.44E-07	1.39E-06	2.03E-06
IBM02b	1.63E-06	6.67E-07	1.43E-06
IBM07a	5.25E-07	6.08E-07	6.39E-07
IBM07b	7.38E-07	5.46E-07	6.34E-07
IBM08a	4.79E-07	6.06E-07	6.00E-07
IBM08b	6.46E-07	4.93E-07	6.20E-07
IBM10a	3.33E-07	3.91E-07	4.13E-07
IBM10b	4.12E-07	3.50E-07	4.10E-07
IBM11a	3.35E-07	3.80E-07	3.99E-07
IBM11b	3.51E-07	3.97E-07	4.19E-07
IBM12a	4.07E-07	4.75E-07	5.07E-07
IBM12b	4.00E-07	4.70E-07	5.02E-07

We have applied the learn-and-test validation technique to both trend modeling and outlier identification. In both cases, for single chip models, we obtained predictions with 3% accuracy for more than 96% of instances.

Resubstitution is the technique that effectively resamples the available data in order to ensure that overfitting is not conducted. It was applied to modeling at both levels of abstractions: interchip and intrachip. We created 100 different subsets of data using uniform random sampling of the data. For the interchip modeling, we selected 70% of the data for each subset and built a separate model using the developed procedure. The percentile analysis indicates that for all results, the interval of confidence is less than ±3% with a probability higher than 97%. For the interchip modeling, we selected a random subset that contained between three and five designs. We repeated this procedure 100 times.

Any time when we do not know the outcome with complete certainty, there are two parameters that characterize our knowledge about the outcome. The first one is what kind of errors are possible at all to happen. That component is captured by the size of the interval of confidence that indicates the amplitude

TABLE IV

Accuracy of IBM07.a interchip model for prediction of IBM designs.

R^2- all data	HPBBOX	MST	CHULL
IBM01.a	0.86312	0.87510	0.82344
IBM01.b	0.84481	0.85810	0.80891
IBM02.a	0.77496	0.94100	0.89040
IBM02.b	0.77986	0.89390	0.88873
IBM07.b	0.86292	0.83272	0.79524
IBM08.a	0.91627	0.94573	0.86103
IBM08.b	0.92660	0.96585	0.83922
IBM10.a	0.96559	0.97846	0.88373
IBM10.b	0.95707	0.96343	0.86071
IBM11.a	0.92939	0.92134	0.83882
IBM11.b	0.92241	0.91774	0.83383
IBM12.a	0.88990	0.89081	0.80354
IBM12.b	0.92917	0.93565	0.83707

of error that is expected. Unfortunately in the majority of situations, we are not able to provide tight bounds on errors that would be of significant interest to the designer. Therefore in these situations we use a second parameter, the probability that the outcomes come out of the range specified by the interval of confidence. Obviously, if the interval of confidence is very tight (small in terms of percentage range) and the probability that the outcome will be within that range, the prediction model is highly accurate. Specifically, the interval of confidence +/- 10% indicates that we are considering the percentage of outcomes that will be within 10% of our prediction and the probability of 86% indicates that in less than 14% of the cases that will not happen. This relatively lower probability was the direct consequence of the fact that from a statistical point of view relatively few designs were available. Nevertheless, the percentile analysis [9], [11], [22] strongly validates the approach and indicates that the statistical trends have less than a one in billion chance of occuring on accident.

IV. Statistical Wire-length PDF and CDF Models

In this section, we present the obtained statistical wire-length model. We present the parameters of the model, obtained PDF and CDF, and summarize the model evaluation results. Although we present a single final model, it is important to emphasize that the procedure presented in the previous Section resulted in a large number of competitive models that differed relatively little with respect to their accuracy and interval of confidence. The model that we present was mainly selected due to its low conceptual complexity and through the use of a set of features that can be rapidly extracted from the post-placement designs.

The prediction abilities of the model are illustrated in Figures 4(a)-5(b). The demonstration example used for the development of the model is IBM07. It is important to emphasize that the model was actually developed using only 60% of randomly selected nets. Figures 4(a) and 4(b) show the normalized net length with respect to HPBBOX for different sizes of HPBBOX. The continuous lines in these two figures indicate the prediction models for small and large HPBBOX respectively. The bottom line corresponds to 10% percentile and the top line to 90% percentile value. All other lines indicate the value of expected length for percentiles that differ by 10% increments. Tables V and VI present the parameters of the models and the obtained R^2 values. They indicate that the square of residuals is consistently high. The t-test indicates that for both sets, the probability of accidental coincidence is less than 10^{-18}. Therefore, it is clear that the model is both theoretically and practically sound.

As can be seen from the table, the variability of the net lengths is well captured as indicated by the high value of the R^2 coefficient, in particular for the small HPBBOX model. There are two main reasons why it is much easier to accurately predict short nets. The first one is that there are significantly more short nets than long nets and, therefore, the statistical model can be developed using a much larger number of samples. The second reason is that short nets usually have significantly fewer terminals, simple structure, and can leverage on relatively small areas of white space in their vicinity. For longer wires, we see that the prediction of nets that are almost as short as their lower bound indicated by the HPBBOX is more accurate than nets that are long. For the long nets, the model relies on the CART model presented in the previous Section that has very high consistency. The CART model-based removal of nets that are predicted to be significantly longer than the HPBBOX-bound, improves the R^2 for all percentiles to above the 0.95 level, essentially matching the accuracy of the model for short nets. The CART model correctly identifies very long nets with accuracy better than 90%. More importantly, less than 1% of nets longer than 25% than indicated by the HPBBOX linear regression-model is not detected by the CART model. Finally, note that no short nets (with HPBBOX value less than 6,000 in either the x or y direction) were identified as outliers.

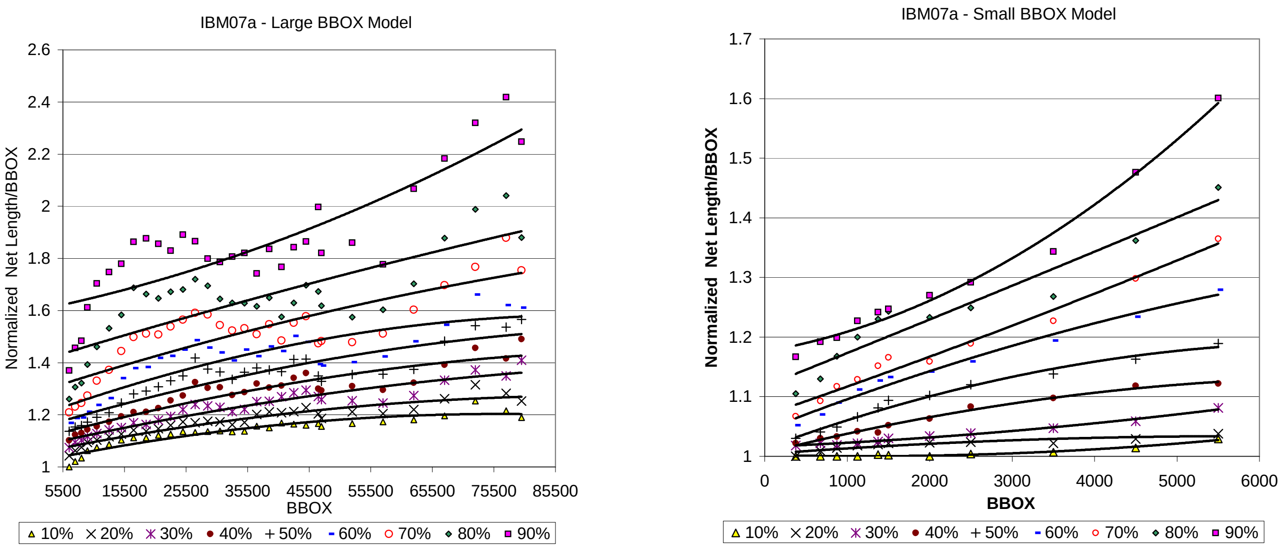
TABLE V

Linear Regression Fit Parameters and R^2 for Small HPBBOX of IBM07 design. Coefficients a, b, and c are used for the quadratic model of the form $ax^2 + bx + c$.

IBM07a - Small HPBBOX Linear Regression Models				
Percentile	a	b	c	R^2
90	9E-09	2E-05	1.1758	0.9876
80	9E-10	5E-05	1.0686	0.9762
70	2E-10	6E-05	1.1175	0.9184
60	-2E-09	5E-05	1.0439	0.9804
50	-4E-09	5E-05	1.0131	0.9849
40	-2E-09	3E-05	1.0055	0.9876
30	1E-09	6E-06	1.0160	0.9854
20	-9E-10	1E-05	1.0038	0.8049
10	1E-09	-3E-06	1.0023	0.9702

Figure 3(a) and 3(b) show a cumulative distribution function (CDF) and a probability distribution function (PDF) for short and long nets. The x-axis indicates the normalized discrepancy against the most likely values. Again, the continuous line indicates the prediction provided by the model and each plot point corresponds to the length of the nets in a particular half-perimeter bounding box bin selected by the resubstitution procedure. From the PDF figures we can conclude that the majority of nets are routed using a wire-length that is close to theoretical minimum and that longer nets are statistically rare.

We evaluated the accuracy and consistency of the PDF and



(a) Large HPBBOX Model

(b) Small HPBBOX Model

Fig. 4. Linear Regression Model for IBM07 design using Cadence Router: (a) Large HPBBOX Model (B) Small HPBBOX Model

TABLE VI

LINEAR REGRESSION FIT PARAMETERS AND R^2 FOR LARGE HPBBOX OF IBM07 DESIGN. COEFFICIENTS a, b, AND c ARE USED FOR THE QUADRATIC MODEL OF THE FORM $ax^2 + bx + c$.

IBM07a - Large HPBBOX Linear Regression Models				
%ile	a	b	c	R^2
90	5E-11	5E-06	1.5944	0.7185
80	-1E-11	7E-06	1.3948	0.6268
70	-2E-11	8E-06	1.2767	0.6890
60	-5E-11	9E-06	1.1828	0.7460
50	-3E-11	7E-06	1.1383	0.8111
40	-3E-11	6E-06	1.0981	0.8655
30	-2E-11	5E-06	1.0720	0.9109
20	-3E-11	5E-06	1.0476	0.9033
10	-3E-11	5E-06	1.0135	0.8862

CDF using the resubstitution procedure. We generated 100 different subsets that contain 60% of initial data and build the PDF and CDF wire-length model. For a hundred randomly selected points their PDF and CDF values were recorded for each of the resubstitution models. The non-parametric interval of confidence was calculated for each point and for the overall probability and cumulative distribution functions. The analysis indicates that with a probability larger than 96% the model is accurate within ±7%. It is interesting to note that the interval of confidence was sharper for the CDF than for the PDF, most likely as a consequence of the CDF integrating discrepancies of the PDF.

Finally, Figures 5(a) and 5(b) show a 3-dimensional representation of histograms that are formed by selecting bins according to their ratio of normalized net length versus HPBBOX and the size of HPBBOX on the other axis. The z-axis indicates instead of the conventional number of nets which belong to a particular bin, the logarithm of this value in order to provide better visual insight in to the distribution of wire-lengths of the net for all lengths. The data in Figure 5(a) was collected after using the Cadence routing tool. The data in Figure 5(b) is generated using the developed prediction model.

It is easy to see that there exists a close correspondence and high correlation between data in the two figures, except for a small subset of bins in the true data that have statistical anomalies due to the specifics of the actual design.

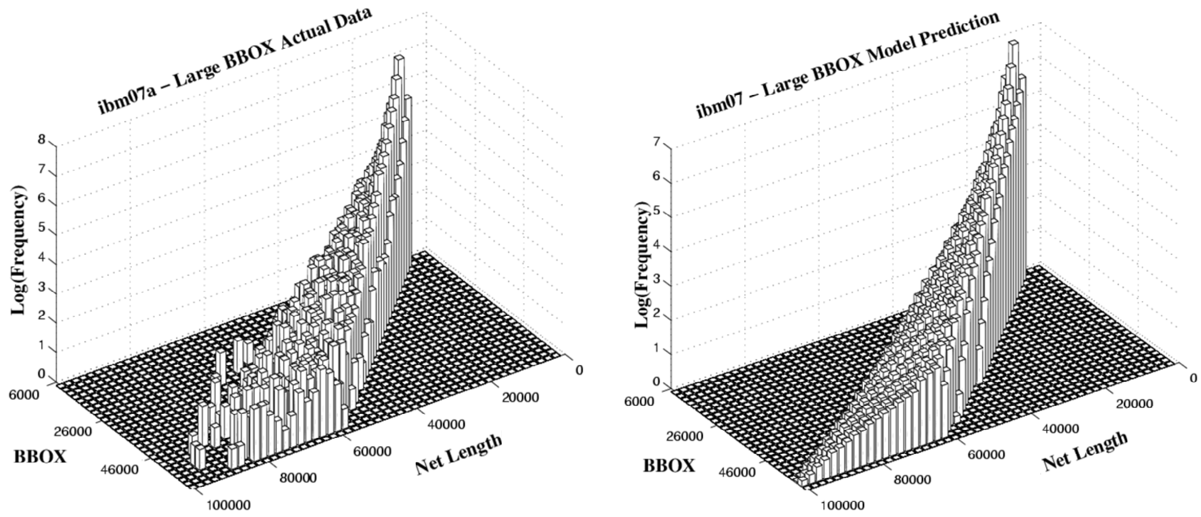
An important question is to what extent the developed models and methodology are applicable to different types of designs and different set of floorplanning and routing tools. Unfortunately, it is difficult to address this question without comprehensive statistical studies. Our expectation is that while models are not directly applicable, they can be relatively easily retargeted to other design and tool scenarios, in particular if alternative statistical methods and tools are used for derivation and validation of new models.

V. APPLICATION OF STATISTICAL WIRE-LENGTH MODEL TO PROBABILISTIC BUFFER INSERTION

In this section, we describe an application of the presented wire-length model. The common underlying idea is to demonstrate the superiority of statistical estimation and probabilistic optimization over the traditional deterministic approach to design automation. In order to accomplish this objective, we applied the developed statistical models to the probabilistic buffer insertion problem.

The buffer insertion problem can be formally stated in the following way. *Given the fan-out wiring tree with parasitic resistances and capacitances, wire-lengths, potential buffer locations, sink required times, sink capacitive loads and a delay constraint at the driving gate, the problem is to place buffers into the tree such that the required arrival time at the input of the driving gate is maximum. We also consider the optimization of the number of buffers used to satisfy the delay constraint.*

The buffer insertion problem was formalized by [28] and models the fan-out wiring tree as a set of distributed RC sections. The Elmore delay model [29] is used to compute the delay of such a wiring tree.



(a) Actual Data

(b) Length generated using Prediction Model

Fig. 5. Logarithm of Histogram of Number of Nets of given Length and given HPBBOX for IBM07 Large HPBBOX.

TABLE VII
POST ROUTING COMPARISON: PROB. VS HPBBOX BASED BUFFER INSERTION ON IBM08 DESIGN.

	Probabilistic		HPBBOX	
	Delay (ps)	# Buffer	Delay (ps)	# Buffer
Net1	1367.03	31	1546.21	24
Net2	865.32	23	983.67	19
Net3	690.46	42	1413.11	40
Net4	1563.21	19	1798.33	16
Net5	2375.49	27	2892.47	20

A detailed methodology for using this modeling effort in buffer insertion is as follows. First the design needs to be placed for generating the wire-length models. Then these models need to be used in a probabilistic buffer insertion framework. This buffering technique assumes that the placement locations of buffers have already been fixed (note that traditional Van Ginneken approach for buffer insertion makes a similar assumption). This optimization effort is following by routing.

In order to estimate the parasitics for each wire segment we need to determine the exact wire-lengths. Now let us suppose that this optimization is being performed during the *in-place mode* during which the exact wire-length is not available. The only available information is about the bounding box of the nets. Using the placement information we can generate the probability distributions of individual wire segments (through the modeling effort presented earlier) of the wiring tree and perform buffer insertion probabilistically. Khandelwal et al. [13] proposed such a probabilistic approach to buffer insertion. For brevity, we omit the details of that algorithm. We ran probabilistic buffer insertion on a placed net (placed using Cadence Qplace) and also traditional buffer insertion [28] assuming bounding box as the net length estimate. After buffer insertion, the entire circuit was routed and the net delay was computed using real wire delay values.

Table VII compares the post routing net delays from probabilistic and traditional buffer insertion. It can be seen that post routing, the probabilistic approach produces significantly better results (average of 21% reduction in delay) than a bounding box based approach indicating the effectiveness of our models and also the superiority of a probabilistic approach.

VI. CONCLUSION

We have built a compact statistical model that predicts the probability that a given net will have a particular wire-length. The model is characterized using a small set of parameters that are easily extracted from the design's floorplan. The runtime of the model is less than one second even for the largest designs. The model was validated using both learn-and-test and resubstitution evaluation techniques.

The proposed net length models have a large range of applicability in emerging probabilistic approaches to design automation that are rapidly gaining acceptance. We demonstrated the effectiveness of our model through extensive experimentation with state of the art commercial and academic tools.

REFERENCES

[1] J. Davis, V. De, and J. D. Meindl, "A stochastic wire-length distribution for gigascale integration (gsi)-part ii: Applications to clock frequency, power dissipation, and chip size estimation," *IEEE Transactions on Electron Devices*, vol. 45, no. 3, pp. 590–597, 1998.

[2] J. Dambre, P. Verplaetse, D. Stroobandt, and J. Van Campenhout, "Getting more out of donath's hierarchical model for interconnect prediction," in *International workshop on System-level Interconnect Prediction*, 2002, pp. 9–16.

[3] M. M.-S. B. Hu, "Wire length prediction based clustering and its application in placement," in *IEEE Design Automation Conference*, 2003, pp. 800–806.

[4] D. Stroobandt, "Multi-terminal nets do change conventional wire length distribution models," in *International Workshop on System Level Interconnect Prediciton*, 2001, pp. 41–48.

[5] ——, "A priori system-level interconnect prediction: Rent's rule and wire length distribution models," in *International workshop on System-level interconnect prediction*, 2001, pp. 3–21.

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

[7] R. G. Wood and R. Rutenbar, "Fpga routing and routability estimation via boolean satisfiability," *IEEE Transactions on VLSI*, vol. 6, no. 2, pp. 222–231, 1998.

[8] L. Breiman, J. Friedman, R. Olshen, and C. Stone, *Classification and Regression Trees*. Chapman and Hall, 1984.

[9] B. Efron and R. Tibshirani, *An introduction to the bootstrap*. Chapman & Hall, 1993.

[10] P. Dalgaard, *Introductory Statistics with R*. Springer-Verlag, New York, NY, 2002.

[11] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. Springer-Verlag, New York, NY, 2001.

[12] R. A. Thisted, *Elements of statistical computing*. Chapman & Hall, Ltd., 1986.

[13] V. Khandelwal, A. Davoodi, A. Nanavati, and A. Srivastava, "A probabilistic approach to buffer insertion," in *IEEE International Conference on Computer Aided Design*, Nov 2002, pp. 560–567.

[14] C. J. Alpert and A. Devgan, "Wire segmenting for improved buffer insertion," in *ACM/IEEE Design Automation Conference*, 1997, pp. 588–593.

[15] J. Lillis, C. K. Cheng, and T. T. Y. Lin, "Optimal wire sizing and buffer insertion for low power and a generalized delay model," in *IEEE International Conference on Computer Aided Design*, 1995, pp. 138–143.

[16] A. Davoodi, V. Khandelwal, and A. Srivastava, "Empirical models for net-length probability distribution and applications," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 12, no. 10, pp. 1066–1075, October 2004.

[17] A. Agarwal, D. Blaauw, V. Zolotov, and S. Vrudhula, "Computation and refinement of statistical bounds on circuit delay," in *ACM/IEEE Design Automation Conference*, 2003, pp. 348–353.

[18] C. Visweswariah, "Death, taxes and failing chips," in *ACM/IEEE Design Automation Conference*, 2003, pp. 343–347.

[19] S. Borkar, T. Karnik, S. Narendra, J. Tschanz, A. Keshavarzi, and V. De, "Parameter variations and impact on circuits and microarchitecture," in *ACM/IEEE Design Automation Conference*, 2003, pp. 338–342.

[20] A. Srivastava, E. Kursun, and M. Sarrafzadeh, "Predictability driven binding: Methodologies and tradeoffs," in *Journal of Circuits, Systems and Computers, Special Issue on Low Power IC Designs*, ser. 4, vol. 11, August 2002, pp. 223–232.

[21] A. Davoodi and A. Srivastava, "Voltage scheduling under unpredictabilities: A risk management paradigm," in *ACM/IEEE Int'l Symposium on Low Power Electronics and Design*, August 2003, pp. 302–305.

[22] B. Efron, "Bootstrap methods: another look at the jackknife," *Annals of Statistics*, vol. 7, pp. 1–26, 1979.

[23] ——, *The Jackknife, the Bootstrap, and Other Resampling Plans*. S.I.A.M., Philadelphia, 1982.

[24] F. P. Preparata and M. I. Shamos, *Computational Geometry: An Introduction*. Springer-Verlag, New York, NY, 1985.

[25] D. C. Hoaglin, F. Mosteller, and J. W. Tukey, *Understanding Robust and Exploratory Data Analysis*. Wiley, N.Y., 1983.

[26] J. W. Tukey, *Exploratory Data Analysis*. Addison Wesley, 1977.

[27] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical recipes in C (2nd ed.): the art of scientific computing*. Cambridge University Press, 1992.

[28] L. van Ginneken, "Buffer placement in distributed rc-tree networks for minimal elmore delay," in *Int'l Symposium on Circuits and Systems*, December 1990, pp. 865–868.

[29] W. Elmore, "The transient analysis of damped linear networks with particular regard to wideband amplifiers," in *Journal of Applied Physics*, ser. 1, vol. 19, 1948.