A Priori Wirelength Estimation: Statistical Models, Bounds and Applications

Jennifer L. Wong[†], Azadeh Davoodi[‡], Ankur Srivastava[‡], Miodrag Potkonjak[†]

[†] Computer Science Department,
University of California, Los Angeles, {jwong,miodrag}@cs.ucla.edu
[‡] Electrical and Computer Engineering Department,
University of Maryland, College Park, {vishwa,ankurs}@glue.umd.edu

UCLA Computer Science Department
Technical Report #060007

Abstract

A priori wirelength prediction is an important and often addressed problem in physical design with numerous and high impact applications. In the last four decades a wide spectrum of insightful and sophisticated techniques for a priori wirelength prediction have been developed. In addition, highly accurate predictions and models have been reported for average wirelength and wirelength distributions. At the same time, although it was realized that the accuracy of prediction for the wirelength of a specific net is bounded, several papers reported good predictors for individual nets.

Our goal is to address the a priori wirelength prediction problem from a new viewpoint. We use a system of rigorous statistical modeling and validation techniques to establish quantitative bounds on the accuracy of a priori wirelength prediction. For each design, we produced a large number of competitive implementations by varying the placement parameters. The data is used for the development of data-driven statistical models for an individual netlist, total wirelength, and distribution of wirelengths in terms of their quantitative predictability. The models are validated and their accuracy quantified using resubstitution. At the same time, we show that both total wirelength and the distribution of wirelength can be very accurately predicted. Using the derived statistical models, we also introduce two new applications for a priori total wirelength predictions: rapid exploration of the placement solution space and identification of placement invariant long nets. Finally, an apparent discrepancy between the previous and new results is analyzed.

1 Introduction

A priori wirelength estimation is one of the canonical problems in physical design. The objective of wirelength estimation is the prediction of various properties of an individual netlist or set of nets before placement. Typical wirelength prediction models, in addition to prediction of the wirelength of a specific netlist, which are pertinent in VLSI IC design include total wirelength and wirelength distribution. The prediction models are often crucial for numerous synthesis and analysis tasks, such as evaluation of a specific floorplan and/or placement, for routing resources prediction, design timing analysis, buffer placement, and early estimation of functional yield that characterizes the percentage of dies on a wafer that fail due to manufacturing defects [3, 5, 10, 13, 2, 9].

Wirelength estimation has been attracting a great deal of research attention [1, 26, 28, 23, 25, 29, 15, 4] since Donath's landmark papers on average wirelength distribution and statistical properties of the placement of a graph [11, 8, 28], and the publishing of Rent's rule [22]. For example, researchers at The University of Southern California developed theoretical models for wirelength distribution [23]. In particular, models for nets with two terminals, which are the most common type to many designs, have received special attention [6, 21]. Both research groups reported accurate total and, therefore, average wirelength prediction. Badapathi and Najm [1] and Wong et al [18] realize the difficulty of a priori wirelength estimation and, instead, focus on models that use floorplan and placement information to develop techniques for relatively accurate prediction of the length of individual netlists.

More recently, a number of research groups have been leveraging on Rent's rule to predict total wirelength and/or wirelength distribution [7, 26, 14, 25, 27]. Most recently, Kahng and Reda [20] proposed a predictor of wirelength for the intrinsic shortest path length (SPL) of an individual net that uses a single value related to the maximum of restricted shortest paths, i.e. paths that consist of the shortest number of hyperedges between any two nodes. Note that SPL is closely related to the number of pins [20]. They reported high accuracy of prediction over a range of designs and physical synthesis tools for average length and remarkable estimates for individual nets.

It is important to note that at the same time, there have been reports from industry that a priori wirelength estimation (APWE) is rarely, if at all, used. For example, Scheffer and Nequist [24] summarized the CAD industry and designer attitude toward a priori wirelength estimation and identified several reasons why APWE is not employed in modern design flows. Although the last two cited papers ([20, 24]) to some extent contradict each other, they are the most closely related research to our study. Specifically, our findings with respect to both total wirelength and wirelength distribution are very similar to the ones presented by Kahng and Reda: both entities are to a great extent independent from the used placement tools and their settings and, therefore, can be accurately predicted. The main difference between this work and the previous effort is that we identify significant room for optimization of the total wirelength by exploring different placements for the same design. Additionally, we characterize the corresponding design exploration space using both statistical and validation techniques. At the same time, we present evidence that contradicts the results of Kahng and Reda for APWE of an individual netlist and provide statistical support and quantification of observations of Scheffer and Nequist about the impossibility of any useful prediction along that line. Finally, it is important to emphasize that our research is sharply different from the one presented in [20] in terms of the employed concepts, and statistical techniques. For example, we examined a large number of competitive designs by exploring the flexibility of the Cadence placement tool. Most importantly, we validated our statistical models using resubstitution [12]. In addition, we identified new degrees of freedom that can be used for total wirelength optimization using placement design space exploration.

2 Methodology

In this section, we first describe the used tools and benchmarks. After that, we introduce two new regression techniques, lower bound regression (LBR) and symmetric monotonic regression (SMR).

2.1 Data Collection and Data Sets

For our experimentation we used the Cadence SOC Encounter toolset. For placement we used the *amoebaPlace* placement tool in the area-driven mode. For the placement tool the chip Aspect Ratios and Row Utilization factors (the percentage of row to be filled by gates), each of which were varied to obtain the desired data.

For the placement tool we investigated the following parameters: placement effort and cut sequence. Three options existed for the placement effort: low, medium and high, each of which performed a tradeoff between runtime and total wirelength and congestion. The medium effort is the default option of the tool which signifies the normal effort level. The cut sequence signifies the way in which the hierarchical partitioning driven placement tool proceeds with the partitioning. A cut sequence of VH signifies a vertical cut followed by a horizontal cut of the placement area while performing partitioning driven placement. A square aspect ratio will benefit from a VH cut sequence for reducing congestion. For a more rectangular aspect ratio a more unbalanced cut sequence would be better (i.e. VVVH).

For routing we used the *Wroute* tool. The routing was done in the globalAndFinal mode, which runs global and detailed routing. The router follows the global routing plan and lays down actual wires that connect the pins to their corresponding nets. The primary goal of detailed routing is to complete all of the required interconnects without creating shorts or spacing violations. While using Wroute we had to specify a global routing grid, which we set to be the existing one defined in the DEF file of each IBM benchmark. For more details we refer the reader to the associated tool documentation.

We used the IBM benchmarks in our experiments, which were provided in LEF/DEF format by [17]. These benchmarks featured an average cell count and net count of 48687 and 46827 respectively, with an average of 5 routing layers.

In addition the tools and data presented in this paper we also used the Cadence Silicon Ensemble Physically Knowledgeable Synthesis (PKS) toolset in another set experiments with the same objective. We initially create a floorplan for each design by setting the Aspect Ratio and Row Utilization factors. We generated many floorplans for each benchmark by varying the Aspect Ratio from 0.75 to 1.25 with steps of 0.01, and the Row Utilization factor to 0.6 and 0.8 with steps 0.005. We used the Qplace placement tool and the Wroute routing tool from the Cadence PKS toolset. In the placement tool, we varied the Cut Ratio from 0.8 to 1.3 with steps of 0.01. We then routed each design using Wroute in the global and final routing modes, using the global grid specified in the LEF file for each benchmark. Due to space limitation we will address only the data obtained using Cadence SOC Encounter tools.

2.2 Modeling and Validation: LBR and SMR

There is a wide variety of available regression models that predict the value of the target variable of interest (e.g. properties of a net) using an observed value of the predicting variable (e.g. wirelength of the net). They range from simple linear and polynomial regression to complex and computationally intensive techniques such as nonlinear least squares, weighted least squares, and Loess regressions [16]. Therefore, identification of the best regression technique for a given data set is a difficult and time consuming problem. In order to speed up the identification process, we have developed lower bound regression (LBR). While all other regression techniques use one data set (learning) to build a model and another (testing data set) to evaluate it, LBR is built using only a testing data set. It is easy to see that LBR can easily be calculated in linear time in terms of the available number of samples for any norm, such as L_1, L_2, and L_{inf}. For example, to calculate the optimal LBR with respect to L_1 norm is to find for each value of variable B, the median of corresponding values for variable A as the regression mapping. Or, for L_2-optimal LBR to find the average value of values of variable A that occur for each value of variable B.

LBR can be used in at least two ways. The first is evaluation of regression technique in terms of how close it is to LBR. Small percentage error between the two regressions indicate that the evaluated regression technique is performing well. The second is to terminate search for a prediction model if the LBR results are not within a error tolerance acceptable to the user.

In addition to LBR, we have also developed symmetric monotonic regression (SMR). SMR has four important properties that make it well suited for application to modeling of a priori wirelength prediction. The first is that the symmetry requirement implies simultaneous consideration of predicting wirelength from an explanatory variable and vice versa. This property is important both in order to make the regression more robust and to make it more suitable for prediction using a combination of readings from several points. Note that if the regression is not symmetric (and a great majority of regressions are not) for a majority of the predicted and explanatory variable values, when simultaneous modeling is attempted, the models will result in a process that converges toward a value that is the intersection of the two regression modeling functions. This result is because for two regression functions $y = F(x)$ and $x = G(y)$, for a majority of x values, $x \neq G(F(x)$, except for the points where functions $y = F(x)$ and $x = G(y)$ intersect.

The second advantage is monotonicity that capture common intuitions in the model, such as the fact that a net with more terminals or nets with joint terminals between many nets are statistically longer. The third advantage is that SMR often performs very close to LBR as shown in the next section. Finally, there is an easy and fast way to computer LBR using combinatorial techniques. Note that readings of digital sensors have a finite number of values. The first step for constructing an LBR or SMR model is binning where all values within a user specified range are placed in a single bin that is used to transform the regression task into the graph domain. For binning we used a procedure that places an equal number of netlists in each bin. Note that each node in the graph domain corresponds to a pair of values e_i, p_i, the explanatory variable and predicted variable respectively. For each value and therefore node, we can easily calculate an arbitrary error using the same technique as for LBR for both predicting p_i from e_i and vice versa. In order to enforce monotonicity constraints, we create edges in the graph in such a way that they are either always

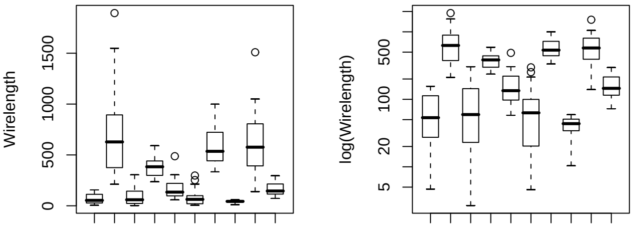

Figure 1: Linear and logarithmic boxplots of 10 randomly selected nets in IBM12 design.

horizontal or only between the nodes that satisfy monotonicity constraints. Finding the path between nodes that corresponds to the minimal value at p_i and any node that corresponds to the maximal value at p_i creates SMR. This path can be found using dynamic programming or a generic Dijkstra shortest path algorithm in (O(n log n)) time.

3 A Priori Wirelength Estimation

In this section, we analyze the achievable accuracy in predicting the wirelength of a net. In the first two subsections we focus on prediction of the expected wirelength of an individual netlist. Specifically, in section 3.1 we analyze to what extent one can predict the length of an individual specific net regardless of what specific parameters are selected for prediction and the approach for combining parameters into the prediction model. After that in the next subsection we will classify all netlists into several classes and show that prediction of wirelength of nets in these classes is also a difficult task. Both of these analyses leverage the use of lowerbound statistical models. Finally, in subsection 3.3 we analyze the properties of all nets and show that regardless of the objective, either predicting average wirelength or a probability density function (PDF) of all nets, the task can be very accurately conducted a priori.

3.1 Individual Netlist

Once adequate statistical techniques for quantifying the lower bound on expected error of prediction are in place, the collected data for each design, where a larger number of different implementations is obtained by varying parameters used by the placement tool, the analysis as to what extent one can predict the wirelength of a specific net is conducted. Figure 1 shows two boxplots in linear (left) and logarithm form (right) for 10 randomly selected netlists in the IBM12 design obtained using parameters RU60, LOW, VHVH. Each boxplot contains the following information. Recall that the boxplot has a box with lines at the lower hinge (quantile Q1 - 25%), the median (quantile Q2 - 50%), the upper hinge (quantile Q3 - 75%) and whiskers which extend to the min and max. To indicate possible outliers, we follow a convention to shorten the whiskers to a length of 1.5 times the box length. Any outliers beyond that are plotted with points.

Resubstitution is a statistical validation procedure [12] for evaluation of a statistical model. It is conducted by applying exactly the same procedure used for building the model on randomly selected subsets of data. Specifically, in our simulations, we select 60% of the available data to build a model on each resubstitution run. For each resubstitution run we create separate PDFs and CDFs. After completing r resubstitution executions (in our experimentation r was 200), we can calculate an interval of confidence for our statistical PDF and CDF models: all what is needed is to calculate the percentage of runs when a specific value of the wirelength differs by the user specified level of tolerance.

¿From Figure 2, which shows plots of the PDFs and corresponding CDFs for 20 different resubstitution rounds for a single short net from the IBM10 design. We see that while the PDFs often report a wide range of variance in probability for a specific wirelength, the CDFs are rather consistent. Specifically, we found that the largest discrepancy of CDFs in no case is larger than 15% for more than 98% of nets. Therefore, we can conclude that the prediction of wirelength prediction for an individual netlist rarely can be accomplished with precision better than +/- 30% regardless of the the type of prediction is conducted. In Table 1 the relative and absolute linear and quadratic discrepancy of a single net of short, medium, and long wirelength in the IBM10 design.

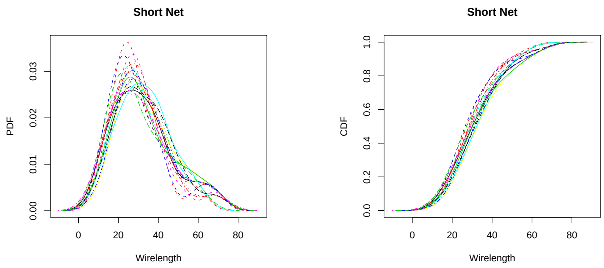

Figure 2: PDFs and CDFs obtained using resubstitution for a short net in IBM10 design.

Table 1: Relative and absolute linear and quadratic discrepancy of nets in IBM10 design.

	Rel L_1	Abs L_1	Rel L_2	Abs L_2
Short	0.37	10.68	0.48	2.71
Med.	0.31	160.50	1.62	37.87
Long	0.38	481.93	3.64	141.13

3.2 Individual Nets - Ensemble View

In the previous subsection, we analyzed and modeled a small randomly selected representative set of netlists. We repeated the same study on several hundred nets and the results were very similar. While these results are highly indicative with respect to the predictability of specific nets, it is also interesting to study nets that are invariant to a specific set of properties that can be calculated from the netlist of a targeted design. At the same time, it is important to recognize that this study is of relatively little value for a practical design flow since one can not predict individual nets accurately. Therefore, we first studied all nets of a particular average across different implementations in order to identify if there is any class of nets that are more predictable than others. Table 2(a) and Figure 3 summarizes the results. Table 2(a) shows average linear discrepancy and deviation for nets of a particular average length over all implementations. Figure 3 presents a PDF and CDF for the nets shorter than 150 units. PDFs and CDFs for other ranges have a very similar shape. Therefore, surprisingly, we can conclude that the length of the net can be predicted well using almost any model since the average linear discrepancy and deviation are just slightly higher than for individual nets.

Figure 4 shows the average linear discrepancy for all nets with 6+ terminals. Table 2(b) presents the average linear discrepancy and deviation for these type of nets. We see that the nets with two and four terminals are more predictable than ones with a larger number of terminals.

3.3 PDF of Wirelengths

In order to evaluate to what extent the PDFs and CDFs of lengths of all nets are subject to changes in different high quality placements, we analyzed CDFs of the IBM design for different placement. As high quality placements we considered only ones that are within 2% of the optimal. Figure 5 shows the CDF for IBM12 design. We see that discrepancy between two CDFs is never more than 0.17%. Our study also indicated that the total wirelength is never changing by more than 10% from the best design in more than

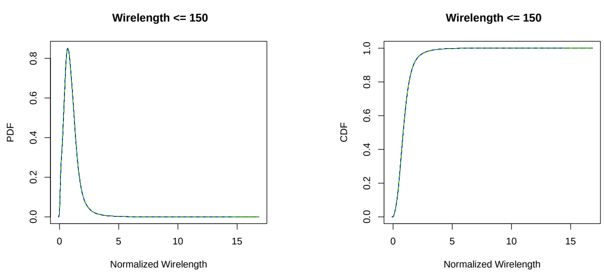

Figure 3: Normalized PDF and CDF for short nets in IBM10 designs.

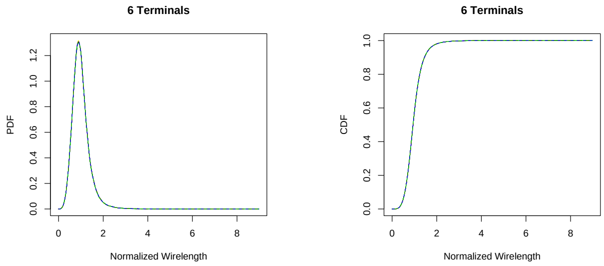

Figure 4: Normalized PDF and CDF of relative lengths for IBM10 design for nets with 6+ terminals.

Table 2: Relative discrepancy and relative standard deviation in PDFs of all nets in IBM10 design over all implementations (a) after grouping by number average wirelength; and (b) after grouping by number of terminals.

(a)

Ave WL	Linear	Dev.
0-150	0.081	0.31
150-300	0.37	0.66
300-700	0.38	0.36
700+	0.18	0.43

(b)

Terms	Linear	Dev.
2	0.15	1.16
4	0.49	0.92
6+	0.33	1.22

32% of the placements of any design.

4 Applications

In this section, we introduce two new applications that leverage on our newly developed statistical models that capture APWE predictions. The first application is the selection of placement parameters for total wirelength reduction. Note that our approach for this task can be easily retargeted in a straightforward way to several other similar application such as reduction of nets that are longer than a user specified limit, or the total number of required buffers (in order to have an implementation with a specified clock cycle). The second application is the identification of intrinsically long nets during placement design space exploration. We have developed a number of other applications such a priori wirelength statistical prediction of very long netlist that are not presented due to the space limitation.

4.1 Selection of Placement Parameters

Majority of placement tools have several parameters that can be specified (see Section 2.1). For example, two Cadence placement tools (amoebaPlace and QPlace) each have at least 3 independent parameters each of which can be assigned to more than 100 different values. Therefore, direct examination of all options even for small designs is exceptionally time consuming and completely infeasible for even medium designs. At the same time both intuition and any experimentation with the parameters strongly indicate that the selection of the parameters significantly impacts the total wirelength of the design and the number of long nets.

In order to address the problem of selecting more suitable placement parameters, we conducted the following exploratory and model building statistical analysis. Figure 6 shows a scatter plot for a small subset of solutions for IBM12 design. Two observations are immediately recognizable: (i) there is a relatively large

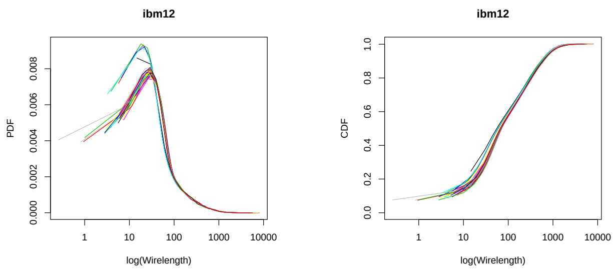

Figure 5: PDF and CDF of actual wirelength in logarithmic scale over different high quality implementations for IBM12 benchmark.

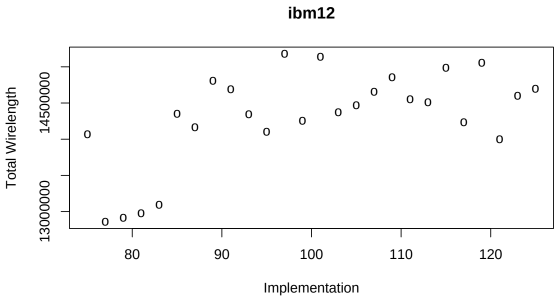

Figure 6: Total wirelength in a part of the IBM12 placement solution space.

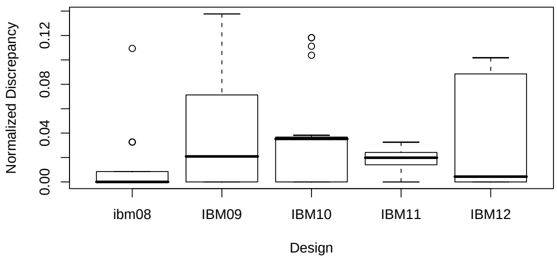

Figure 7: Search for high quality placement for 5 IBM designs using $k_s = 5$, $k_b = 1$.

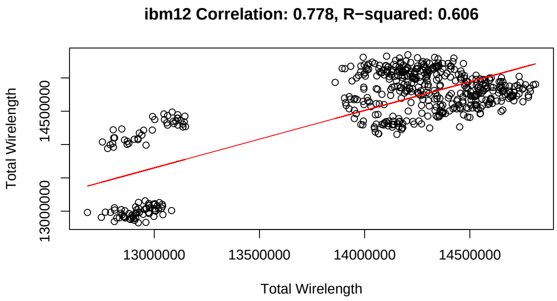

Figure 8: Correlation in terms of total wirelength of neighboring placements in the IBM12 design.

number of solutions that are close to the best; and (ii) correlation of close placement is far from perfect, but still reasonably high.

Figures 7 and 8 quantify these two findings. Figure 7 shows boxpolots of relative quality of solutions for 5 IBM designs each explored over at least several hundred different placements. The boxplot indicates the difference between the best design found over all realized implementationsss and once using the rapid search procedure. Figure 8 presents the correlation between two neighboring placements that differ by the smallest values. The graph is created by placing each pair of neighboring placement as one point in two dimensional space where each coordinate indicates the total wirelength of one of the solutions. The solution with smaller total wirelength is always used for the x-coordinate.

The pseudo-code of our approach for identifying placement parameters is the following.

Placement Parameter Selection
1. Sample the Solution Space at k_s points;
2. Identify k_b best points;
3. Search the Solution Space using steepest descent
starting from each of k_b points;

Since there is a well defined structure in the solution space, in the first step we sample the solution space. For this purpose we use a weighted random strategy, were each point is weighted proportional to its relative quality in terms of total wirelength in previously considered designs. For this purpose we use the geometric average of the relative quality with respect to the best value found for each design. We use geometric means due to its robustness [19]. Table 3 shows the quality of best found placement for various values of k_s and $k_b = 1$. The final step is a greedy search for potential improvements near the most promising

		IBM08	IBM09	IBM10	IBM11	IBM12
k_s=3	ave	0.048	0.034	0.052	0.021	0.054
	med	0.033	0.021	0.035	0.024	0.049
	max	0.136	0.136	0.134	0.033	0.129
k_s=4	ave	0.028	0.037	0.004	0.015	0.044
	med	0.008	0.021	0.003	0.014	0.009
	max	0.129	0.142	0.013	0.031	0.129
k_s=5	ave	0.012	0.040	0.038	0.018	0.034
	med	0.000	0.021	0.035	0.020	0.004
	max	0.109	0.138	0.118	0.033	0.102

Table 3: Search for high quality placement for 5 IBM designs using different k_s values and $k_b = 1$.

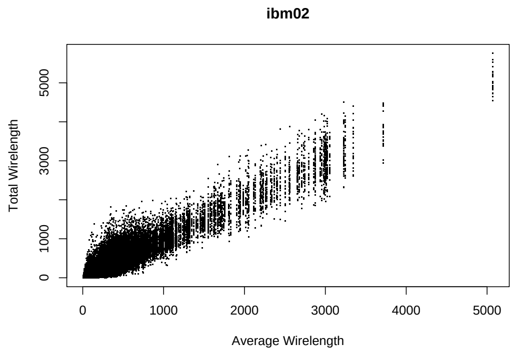

Figure 9: Scatter plot of actual and average lengths for nets in IBM02 design.

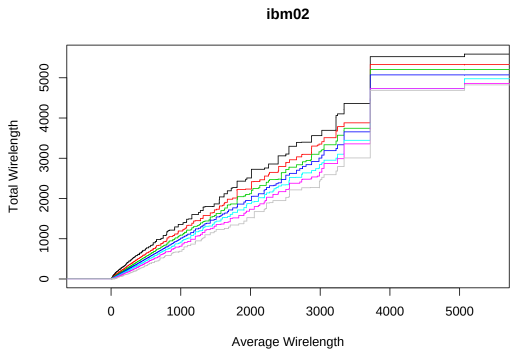

Figure 10: Density of wirelength of different implementations for IBM02 design.

current solutions. This step is motivated by relatively high correlation values of quality of the solutions that have small differences only in one parameter.

4.2 Identification of Long Netlists

As our study indicates predicting the wirelength of individual nets is impossible within useful bounds. Therefore, we decided to address a less ambitious task: identification of long nets when the total wirelength of the design is optimized using design space exploration. Design space exploration can be conducted either using the procedure presented in the previous subsection or with any other approach. We use as the definition of long nets as ones that are longer than k% of all nets in the design. In order to address this problem we conducted the following study with results shown in Figures 9 and 10. The scatter plot in Figure 9 shows the wirelength of all nets of IBM02 over all implementations plotted versus the average wirelength of each net.

Using this data we produced a probability density function using SMR. In Figure 10 the lines indicate 87.5%, 75%, 62.5%, 50%, 37.5%, 25%, and 12.5% percentile of the PDF. From the CDF we can calculate how many placements are required to correctly conclude that a particular net will be within a specified range. For example, if the net is longer than 2,000 units in 2 designs, we can expect with probability higher than 99% that it is longer than 2,000 units in any highly optimized design.

5 Global View, Limitations, Future Work, and Conclusion

Academic research resulted in numerous models with relatively high reported accuracy for both total and individual APWE. At the same time, CAD companies and designers rarely employ the developed models and results. We believe that the main reason for this discrepancy is that academic approaches and tools are implicitly overtuned to a particular placement and/or routing tool and more dangerously to a specific benchmark set(s) used to develop the models and demonstrate their effectiveness. We believe that the overtuning effect also explains the discrepancy between results of the previous publications and our research. By consistently employing statistical validation techniques (resubstitution) and increasing the number of considered instances using different setting of the used placements, we considerably alleviated the impact of overtuning.

Nevertheless, we see the main limitation of this research is that we did not use a diverse enough set of tools and benchmarks. Therefore, all negative results that we report have stronger footing than the positive synthesis and estimation results that may be restricted to the specific sets of tools, benchmarks, and concepts used for creating different implementations. At the same time, we believe that the positive results can directly benefit the user of the specific set of Cadence tools. Our future work will be, therefore, on retesting our approach and results on a more comprehensive set of tools and designs.

In summary, we studied using statistical modeling and validation techniques the a priori wirelength estimation problem. We demonstrated that the gained insights and statistical models indicate that while total wirelength and wirelength distribution are subject to prediction within less than 10%, the wirelength of individual nets can not be accurately predicted even within 50% regardless of the prediction variables and techniques. Finally, we addressed two new applications of a priori wirelength estimation that enable the creation of implementations with significantly reduced total wirelength.

References

[1] A. Agnihotri, et al. Fractional cut: Improved recursive bisection placement. In *ICCAD*, page 307, 2003.

[2] A. Khatkhate, et al. Recursive bisection based mixed block placement. In *ISPD*, pages 84–89, 2004.

[3] A. R. Agnihotri, S. Ono, and P. H. Madden. Recursive bisection placement: feng shui 5.0 implementation details. In *ISPD*, pages 230–232, 2005.

[4] H. Bakoglu. *Circuits, Interconnections, and Packaging for VLSI.* Addison-Wesley, Reading, Massachusetts, 1990.

[5] C. Alpert, et al. A semi-persistent clustering technique for VLSI circuit placement. In *ISPD*, pages 200–207, 2005.

[6] A. E. Caldwell, A. B. Kahng, and I. L. Markov. Can recursive bisection alone produce routable placements. In *DAC*, pages 477–482, June 2000.

[7] A. E. Caldwell, A. B. Kahng, and I. L. Markov. Optimal partitioners and end-case placers for standard-cell layout. *IEEE TCAD*, 19(11):1304–1314, November 2000.

[8] T. Chan, J. Cong, and K. Sze. Multilevel generalized force-directed method for circuit placement. In *ISPD*, pages 185–192, 2005.

[9] P. Christie and J. P. de Gyvez. Pre-layout prediction of interconnect manufacturability. In *SLIP*, pages 167–173, 2001.

[10] K. Doll, F. Johannes, and K. Antreich. Iterative placement improvement by network flow methods. *IEEE TCAD*, 13(10):1189–1200, 1994.

[11] W. E. Donath. Statistical properties of the placement of a graph. *SIAM Journal on Applied Mathematics*, 16(2):439–457, Mar. 1968.

[12] B. Efron and R. Tibshirani. *An Introduction to the Bootstrap.* Chapman & Hall, 1993.

[13] H. Eisenmann and F. M. Johannes. Generic global placement and floorplanning. In *DAC*, pages 269–274, 1998.

[14] S. Goto. An efficient algorithm for the two dimensional placement problem in electrical circuit layout. *IEEE Trans. on Circuit and Systems*, 28:12–18, 1981.

[15] N. Harada. LSI interconnect length prediction method using statistical mechanical placements theory. *NEC Research and Development Journal*, 72:56–63, Jan. 1984.

[16] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical learning; data mining, inference, and prediction.* Springer-Verlag, 2001.

[17] IBM Benchmarks. http://er.cs.ucla.edu/benchmarks/ibm-place2/.

[18] J. L. Wong, et al. Wire-length prediction using statistical techniques. In *ICCAD*, pages 702–705, 2004.

[19] R. Jain. *The Art of Computer Systems Performance Analysis.* John Wiley and Sons, 1991.

[20] A. B. Kahng, S. Reda, and Q. Wang. Architecture and details of a high quality, large-scale analytical placer. In *ICCAD*, page 173180, 2005.

[21] A. B. Kahng, P. Tucker, and A. Zelikovsky. Optimization of linear placements for wirelength minimization with free sites. In *ASP-DAC*, pages 241–244, 1999.

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

[23] B. Obermeier, H. Ranke, and F. M. Johannes. Kraftwerk: a versatile placement approach. In *ISPD*, pages 242–244, 2005.

[24] L. Scheffer and E. Nequist. Why interconnect prediction doesn't work. In *SLIP*, pages 139–144, 2000.

[25] C. Sechen and K. Lee. An improved simulated annealing algorithm for row-based placement. In *ICCAD*, pages 478–481, 1987.

[26] T. F. Chan, et al. Multilevel optimization for large-scale circuit placement. In *ICCAD*, pages 171–176, 2000.

[27] T. Taghavi, X. Yang, and B. Choi. Dragon2005: large-scale mixed-size placement tool. In *ISPD*, pages 245–247, 2005.

[28] T.C. Chen, et al. NTUplace: a ratio partitioning based placement algorithm for large-scale mixed-size designs. In *ISPD*, pages 236–238, 2005.

[29] N. Viswanathan, M. Pan, and C. Chu. FastPlace: an analytical placer for mixed-mode designs. In *ISPD*, pages 221–223, 2005.