Intrinsic Shortest Path Length: A New, Accurate A Priori Wirelength Estimator

Andrew B. Kahng Sherief Reda
Computer Science and Engineering Dept.
Univ. of CA, San Diego
La Jolla, CA 92093
Email:{abk@ucsd.edu, sreda@cs.ucsd.edu}

Abstract

A priori wirelength estimation is concerned with predicting various wirelength characteristics before placement. In this work we propose a novel, accurate estimator of net lengths. We observe that in "good" placements, the length of a net is very strongly correlated with the numbers of nets in the shortest paths connecting node pairs of the net, when each shortest path is computed under the restriction that the net itself does not exist. We refer to this as the net's *intrinsic shortest path length (ISPL)*. Using ISPL as a wirelength estimator has several advantages: (1) it transparently handles multi-pin nets and is a strong predictor of their length; (2) it strongly correlates with the average netlist wirelength; (3) it has a distribution that is similar to that of wirelength; and (4) it acts as a good predictor for individual net lengths. Based on ISPLs, we characterize VLSI netlists with a single value and develop an intuitive, empirical link between our proposed value and the Rent parameter. We also analytically model the relationship between ISPL and wirelength, and use ISPLs in two practical applications: *a priori* total wirelength estimation and *a priori* global interconnect prediction.

1 Introduction

A priori wirelength estimation refers to the process of estimating and predicting the wirelength characteristics of VLSI netlists without knowledge of the netlist placement or floorplanning. Wirelength estimates include, for example, average wirelength, wirelength distribution, total wirelength, and individual net length prediction. Typically, such estimation is conducted without assumptions regarding the underlying placer, since most estimator models assume that a placer will achieve a generally "good" placement. Numerous applications can benefit from *a priori* wirelength estimations, including physical-driven synthesis [16, 26, 15, 27, 21, 19, 23, 24], synthetic benchmark generation [31], field-programmable gate array routing estimation [3], and technology extrapolation [12, 5].

A great amount of established literature [14, 30, 28, 4, 36, 32, 11, 18, 32] deals with wirelength characteristics and estimations, starting from the classical paper of Landman and Russo describing Rent's rule and parameter [22], as well as the early papers of Donath on average wirelength and wirelength distributions [13, 14]. Sastry [30] characterizes wirelength distribution and develops theoretical distribution models. Hamada *et al.* [18] use the models of [30] and and the netlist topological structure to estimate the average wirelength of multi-pin nets. Recently, two wirelength estimators have been proposed for two-pin nets: mutual contraction [23, 24], and edge separability [10]. Good correlations have been observed between average wirelength and average mutual contraction over ranges of values. As for individual net length prediction, Bodapati and Najm [4] propose to estimate individual net lengths based on building a characterization of the underlying placement and routing tool. In addition to wirelength, a number of works focus on estimating congestion based on netlist characteristics [21, 24].

The last few years have seen a revival of interest in using Rent's rule to estimate netlist parameters such as average netlist wirelength [9, 32] or wirelength distribution [14, 11, 32]. However, Rent parameter-based estimates have their own limitations. For example, Rent parameter estimates typically vary depending on the partitioning or placement algorithm that is used to calculate them [17, 34]. This variation creates a source of noise in their values. Furthermore, Rent's rule-based estimates do not directly operate on hypergraphs but rather require hypergraphs to be converted to graphs [11]. More importantly, they do not answer some basic questions such as: Given only a netlist $H = (V, E)$, is the length of some net $e_i \in E$ going to be larger than that of some other net $e_j \in E$?[1] Answering this question would allow us to sort all nets based on their *a priori* estimates and flag, say, the top 10% as potentially critical early in the design process. Fundamentally, Rent-based estimates are helpful in deriving general conclusions about the netlist - e.g, average wirelength or what the wirelength distribution looks like - but are not useful in deriving *a priori* individual net length estimates.

In this paper we present a novel, accurate *a priori* wirelength estimation method. Central to our method is the concept of *intrinsic shortest path* of a net. The intrinsic shortest path length (ISPL) of a given two-pin net is the minimum number of nets required to traverse from one node of the net to another, subject to the constraint of not using the given net. Had the given net been used, the shortest path computation would certainly be trivial - the given net itself is the shortest path. Using empirical data, we show that: (1) ISPL is a strong predictor of the average and individual net length; (2) ISPL transparently handles two-pin and multi-pin nets; (3) ISPL gives succinct characterizations of individual nets, nodes and entire netlists, and (3) ISPLs are straightforward to calculate, and their values are well-defined for any given netlist. We also analytically model the relationship between wirelength and ISPL and use this in two practical applications: *a priori* total wirelength estimation and *a priori* global interconnect prediction.

The organization of this paper is as follows. Section 2

[1] This question is more difficult and relevant when the two nets have the same number of pins.

introduces the notion of intrinsic shortest path and empirically demonstrates that it is a good predictor of (i) average net length, (ii) the length of multi-pin nets, and (iii) the length of individual nets. Section 3 explores the relationship between our estimates and Rent parameter, and Section 4 demonstrates two practical applications for ISPLs: *a priori* total wirelength estimation and *a priori* global interconnect prediction. Finally, Section 5 summarizes the contributions of our work and gives a number of future research directions.

2 Intrinsic Shortest Path of a Net

We start our exposition with some basic definitions. A hypergraph is denoted by $H = (V, E)$, where V is the set of nodes and E is the set of hyperedges. A hyperedge $e_k \in E$ is a set of nodes, $e_k \subseteq V$, that indicates that these nodes are electrically connected. The cardinality of a hyperedge $e_k \in E$ is denoted by $|e_k|$. The set of hyperedges adjacent to a node v is given by $adj(v) = \{e_k|$ such that $v \in e_k\}$. The Half-Perimeter Wirelength (HPWL), or wirelength, of a hyperedge e_k is half the perimeter of the smallest bounding box including all nodes of e_k in some placement of H.[2] Central to our approach is the concept of *restricted shortest path*.

Definition 1. Given a hypergraph $H = (V, E)$ and a set of hyperedges $R \subset E$, the length of the *restricted shortest path* between two nodes u and v, denoted $d(u \rightsquigarrow v|R)$, is the minimum number of hyperedges in a path connecting u and v in $H' = (V, E - R)$.

For example, in Figure 1, the length of the restricted shortest path between u and v is three assuming $R = \{\{u, v\}\}$, and the length of the shortest path between i and j is five assuming $R = \{\{i, j\}\}$. Since hypergraphs contain multi-pin nets, we set the weight of a hyperedge e_k as $\frac{|e_k|}{2}$. Thus, when we speak of a restricted shortest path, we actually mean the minimum-weight set of hyperedges connecting u and v in $H' = (V, E - R)$.

We now extend the restricted shortest path definition to a set of nodes rather than just two nodes. For a general set of nodes $S \subset V$, the length of the restricted shortest path is $\mathcal{RSP}(S|R) = \max_{u,v \in S} d(u \rightsquigarrow v|R)$. Fundamentally, the restricted shortest path of a set of nodes is the maximum length of any restricted shortest path, taken over every pair of nodes in S. This brings us to the next crucial definition.

Definition 2. The *intrinsic restricted shortest path*, or simply *intrinsic shortest path (ISPL)*, of a hyperedge e_k is $\mathcal{RSP}(e_k|\{e_k\})$.

The intrinsic shortest path of a net and the net essentially form a cycle. Our main observation that in a final "good" placement, the wirelength of a hyperedge e_k is strongly correlated to its intrinsic shortest path length $\mathcal{RSP}(e_k|\{e_k\})$. For example, in Figure 1, it is expected that on the average the wirelength of $\{u, v\}$ is less than the wirelength of $\{i, j\}$. Intrinsic shortest path can be readily calculated *a priori* to placement using any shortest path algorithm with appropriate weighting of hyperedges. Such calculation is typically fast since most intrinsic shortest paths are comprised of very few hyperedges.

[2] In this paper, we use the term *wirelength* to denote the Half-Perimeter Wire Length (HPWL). HPWL is equivalent to the Steiner Minimal Tree (SMT) cost for two-pin and three-pin nets, and is well-correlated for multi-pin ($\geq$ 4) nets [8]. Most modern placers are HPWL-driven.

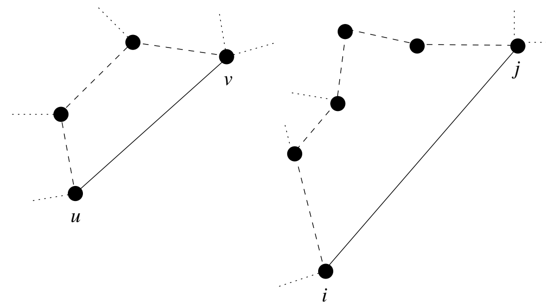


Figure 1: In a "good" placement, the wirelength of edge $\{u, v\}$ is on the average less than the wirelength of edge $\{i, j\}$. The ISPL of edge $\{u, v\}$ is equal to three, while the ISPL of edge $\{i, j\}$ is equal to five.

To validate our observation and make use of it to fulfill typical estimation applications, we pose the following questions.

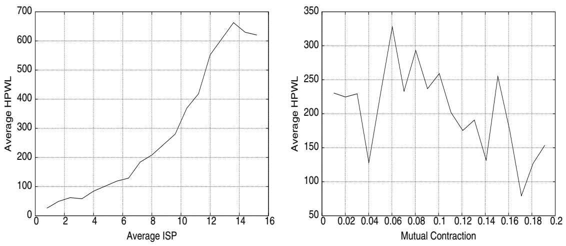
1. Is there any correlation between the placed net length and net ISPL? A strong correlation validates our observation.
2. Is there any relation between the effect of net pin count on average net length and average net ISPL?
3. Is it possible to correlate the overall average netlist wirelength and the average ISPL over a range of benchmarks? Furthermore, given two netlists with the same number of nets and nodes but different connectivity, is it possible to correlate the average ISPL to the total wirelength?
4. Given two individual nets of some netlist, can we predict which individual net will be placed with greater wirelength?
5. Is there any relationship between the distribution, or profile, of ISPL and the wirelength distribution?

These five questions can be thought of as challenges or benchmarks for any new *a priori* estimation study or claims. We answer them in each of the next five subsections.

2.1 ISPL and Net length Relationship

We start by validating our observation regarding the correlation between ISPL and net length. Given a benchmark, we first calculate the intrinsic shortest path of every hyperedge, then place the benchmark using Dragon (version 3.01) [33] and calculate the wirelength of every net. To allow comparisons to other estimates such as mutual contraction [23, 24] and edge separability [10], we use the reporting method of [23, 24]: we calculate the range of intrinsic shortest paths $[c_{\min} \dots c_{\max}]$, such that total wirelength of nets with intrinsic shortest path length $> c_{max}$ is less than 5% of the total wirelength. The calculated range is partitioned into 20 equal bins and the averages of ISPL and wirelength values in each bin are calculated. We plot the intrinsic shortest path versus average wirelength for the ibm08 in Figure 2.a. We can easily see that as ISPL increases, the average wirelength also increases. The correlation coefficient between ISPL and wirelength is 0.961 for ibm08.

To compare our estimator to existing estimators in the literature, we implement both mutual contraction (MC) [24] and



(a) ISPL vs wirelength for ibm08.

(b) Mutual contraction vs wirelength for ibm08.

Figure 2: Relation between wirelength and ISPL for two-pin nets.

edge separability (ES) [10]. A method similar to edge separability has been proposed to predict routing congestion [21]. Both mutual contraction and edge separability methods handle only two-pin nets and require hypergraphs to be transformed to graphs by replacing every hyperedge of cardinality k with a clique of edges, where each edge has the weight $\frac{2}{|e_k|(|e_k|-1)}$ in MC, and a weight of $\frac{1}{|e_k|-1}$ in ES. The *mutual contraction* between two nodes $\{u,v\}$ is defined as

$$\mathcal{MC}(u,v) = \frac{w(u,v)}{\sum_{x \in adj(u)} w(u,x)} \times \frac{w(v,u)}{\sum_{x \in adj(v)} w(v,x)}, \quad (1)$$

where the weight between two nodes $w(u,v)$ is the sum of the weights of the edges between u and v. The edge separability of an edge $\{u,v\}$ is defined as

$$\mathcal{ES}(u,v) = \frac{\lambda(u,v)}{min(|adj(u)|,|adj(v)|)}, \quad (2)$$

where $\lambda(u,v)$ is the minimum cut size among all cuts separating u and v.

In a study of two-pin nets, we give the magnitude of the correlation coefficient between wirelength and intrinsic shortest path, mutual contraction, and edge separability in Table 1 for the entire IBM (version 1.0) benchmarks. The results show that ISPL has a very strong correlation to wirelength with an average correlation coefficient of 0.914, versus 0.699 for mutual contraction and 0.579 for edge separability. We also plot mutual contraction versus wirelength for the ibm08 benchmark in Figure 2.b. Finally, the correlation coefficients between ISPL for multi-pin nets (including two-pin nets) in Table 2 again show very strong correlation to wirelength. Having validated our observation, we now address the remaining four questions from above.

2.2 Effect of Pin count on ISPL and Wirelength

The objective of this experiment is to test the effect of pin count on both average wirelength and ISPL. Initially, we calculate the average ISPL for every k-pin net, then run a placer and calculate the average placed wirelength for every k-pin net. We plot the results in Figure 3 for the ibm01 benchmark, where the x-axis gives the number of pins, the left y-axis gives the average wirelength, and the right y-axis gives the average intrinsic shortest path length. The average wirelength is shown by the solid line, while the average intrinsic shortest path is given by the dashed line. We can immediately see that both quantities are very closely related, as an increase in pin count simultaneously increases both ISPL and wirelength. A correlation test gives a very strong correlation coefficient of 0.95. We can safely conclude that the average wirelength and average ISPL of a $k+1$ net are larger than those of a k-pin net.

Bench	Estimate		
	ISPL	MC [24]	ES [10]
ibm01	0.912	0.874	0.928
ibm02	0.908	0.825	0.789
ibm03	0.914	0.705	0.520
ibm04	0.921	0.770	0.826
ibm05	0.920	0.762	0.476
ibm06	0.907	0.622	0.048
ibm07	0.881	0.781	0.744
ibm08	0.961	0.693	0.893
ibm09	0.921	0.711	0.192
ibm10	0.922	0.724	0.975
ibm11	0.955	0.805	0.551
ibm12	0.900	0.655	0.329
ibm13	0.923	0.495	0.901
ibm14	0.747	0.866	0.823
ibm15	0.946	0.777	0.368
ibm16	0.947	0.828	0.010
ibm17	0.941	0.645	0.575
ibm18	0.938	0.836	0.487
Average	0.914	0.743	0.579

Table 1: Magnitude of correlation coefficients between ISPL and wirelength for two-pin nets. *MC* is the mutual connectivity metric [24]. *ES* is edge separability [10].

bench	ISPL	bench	ISP
ibm01	0.960	ibm10	0.982
ibm02	0.909	ibm11	0.967
ibm03	0.948	ibm12	0.977
ibm04	0.971	ibm13	0.962
ibm05	0.903	ibm14	0.954
ibm06	0.916	ibm15	0.979
ibm07	0.969	ibm16	0.972
ibm08	0.944	ibm17	0.978
ibm09	0.962	ibm18	0.972

Table 2: Correlation coefficients between ISPL and wirelength for all nets whether two-pin or multi-pin. The average is 0.956. ISPL is the proposed intrinsic shortest path estimate.

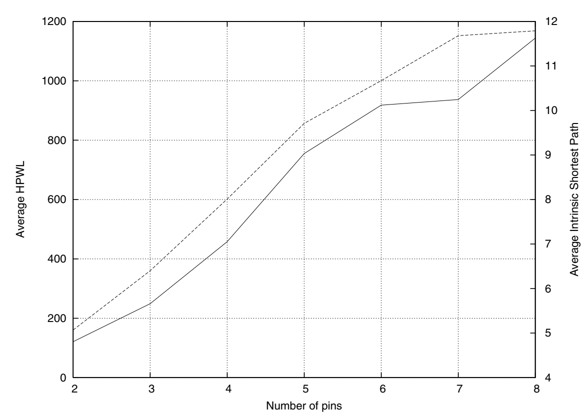
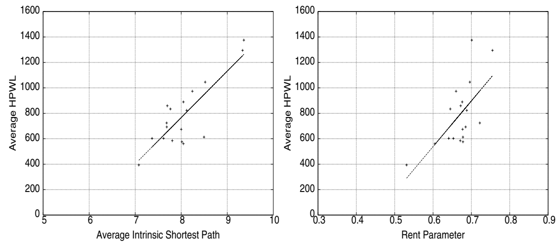


Figure 3: Relationship between number of pins of a net and (i) the average wirelength of the net (solid line) (ii) the average intrinsic shortest path (dashed line) for ibm01. Correlation coefficient between wirelength and ISPL = 0.95.



(a) Average intrinsic shortest path vs average wirelength for the 18 benchmarks of the IBM benchmark set. Standard regression error = 147.2.

(b) Rent parameter vs average wirelength for the 18 benchmarks of the IBM benchmark set. Standard regression error = 201.9.

Figure 4: Relation between average netlist wirelength and average ISPL.

2.3 Relationship Between Average Netlist Wirelength and Average ISP

The average net length of a given netlist determines the total wirelength, which in turn dictates the total required routing resources and whitespace required for the layout. For example, if two netlists have the same number of nets and nodes, then it is expected that the one with higher average net length to require more whitespace.

One of the typical applications of Rent's rule is to calculate or derive the average net length of VLSI netlists, with the common conclusion that a higher Rent parameter translates to a higher average net length. To measure the effectiveness of our estimates, we calculate the average net length and the average ISPLs for the entire set of 18 IBM benchmarks and plot these 18 points in Figure 4(a); we then execute correlation and regression tests. Our computations yield a correlation coefficient of 0.834, establishing a very strong correlation between average ISPL and average wirelength over the entire IBM benchmark set; standard regression error = 147.2. We repeat the same computations using Rent's parameter (calculated with the publicly available Rent parameter calculator RENTC [1]), and then plot the results in Figure 4(b). We find that Rent parameter correlates to the average wirelength with a coefficient of 0.654, and with a standard regression error of 201.9.

To further confirm this correlation between average ISPL and wirelength, we use the publicly available synthetic netlist generator GNL [31, 2] to generate four benchmarks with the same number of nets (9832 nets) and the same number of nodes (10032), but with different Rent parameters. We again place the four benchmarks and report their final wirelength and average ISPL in Table 3. The results confirm the strength of ISPL as an average wirelength predictor, since the average ISPL and total wirelength perfectly correlate with a coefficient of 0.999.

2.4 Individual Net length Prediction

In this experiment, we test our estimator with a basic but quite fundamental question:

• Given a netlist $H = (V, E)$, and two arbitrary nets $e_i \in E$ and $e_j \in E$, can we *a priori* decide which net will require more wirelength?

Bench	Rent	wirelength	ISPL
GNL50	0.50	53788	6.924
GNL55	0.55	56632	6.991
GNL60	0.60	65948	7.244
GNL65	0.65	73502	7.399
GNL70	0.70	77365	7.515

Table 3: Results using synthetic benchmarks with the same number of nodes (10032) and nets (9832) but with different Rent parameters. Total wirelength and average ISPL are reported. Both quantities are in perfect correlation.

Since it is expected – from Subsection 2.2 – that the average wirelength of a $k+1$-pin net to be larger than a k-pin net, we focus on the more relevant question: can we predict the length of two given different nets e_i and e_j with the same cardinality, i.e., $|e_i| = |e_j|$? As mentioned earlier, solving this question allows us to prioritize all nets early in the design process and apply possible optimizations, e.g., buffering, to the most urgent nets.

Before we test how well our estimator can answer this question, we first set an upper bound on the performance of any predictor. Assume we have an oracle O that has a good placement – ideally the optimal placement – of the netlist H. Presented with the two nets, the oracle either outputs '$<$' if it finds that e_i takes less wirelength than e_j or '$\geq$' otherwise, i.e., $O(e_i, e_j) = \{<, \geq\}$. Similarly, the predictor $\mathcal{P}(e_i, e_j) = \{<, \geq\}$ behaves according to its prediction. We say a prediction is *successful* if and only if $O(e_i, e_j) = \mathcal{P}(e_i, e_j)$; otherwise, the prediction is a *failure*.

What is the best strategy predictor $\mathcal{P}$ can use? At best, the predictor should not predict at all and just execute some placer P on the given netlist, then use the net lengths it gets from the placement in its prediction. The ratio between the number of successful predictions to the total number of predictions gives the *performance* of the best predictor. To quantify the best predictor, we execute the following experiment: We run Dragon [33] (version 3.01), FengShui (version 2.6) [35], mPL (version 4.0) [7] and Capo (version 9.0) [6] on the 18 IBM benchmarks. We found that Dragon is giving the best placement in most cases, and consequently we use its placement for the Oracle. We then use each of the other placers as the best strategy the predictor can use, and calculate the performance of the predictor.

We present the predictor with random samples of pairs of nets with the same cardinality over all net cardinalities[3]. We first notice that a helpless predictor can be correct on the average 50% of time. Surprisingly, the best performance a predictor can get - aided with some placer P - is about 70% as given in Table 4. Thus, the upper bound for performance is 70%, and the lower bound is 50%. A predictor based on mutual contraction [25] is correct on the average 52% of the time[4]. A predictor using our proposed *a priori* ISPL strategy is correct 60% of the time as given in the fifth column of Table 4. We believe this is a very strong result given that our ISPL predictor knows nothing about placement.

[3]We keep on presenting the predictor with samples of net pairs until its performance is stable with changes of less than 0.01%.

[4]Since mutual contraction is only defined to predict two-pin nets, we report its success rate with two-pin nets.

bench	Capo 9.0	mPL 4.0	Feng–Shui 2.6	MC	*A priori* ISP
ibm01	68.15	69.26	69.5	52.49	62.24
ibm02	69.13	69.20	70.08	53.20	59.48
ibm03	70.42	71.46	71.09	52.71	58.67
ibm04	71.56	71.72	72.65	52.00	57.52
ibm05	71.06	69.92	71.55	54.21	52.81
ibm06	68.44	68.14	69.17	51.14	57.89
ibm07	71.94	71.29	71.44	52.09	60.77
ibm08	69.82	71.44	71.43	53.63	62.96
ibm09	69.18	69.29	69.83	52.51	56.89
ibm10	71.91	71.84	72.07	52.71	61.88
ibm11	70.57	70.75	71.34	52.20	59.77
ibm12	71.70	72.71	72.72	51.55	60.86
ibm13	71.15	71.76	71.56	52.44	60.22
ibm14	69.45	69.53	70.34	52.25	59.48
ibm15	71.56	72.31	72.23	52.26	59.96
ibm16	70.03	70.69	70.84	52.71	60.65
ibm17	73.08	73.50	73.06	51.41	61.14
ibm18	69.72	70.41	70.43	52.45	60.89
Average	70.49	70.84	71.18	52.44	59.67

Table 4: The accuracy of predicting individual net lengths.

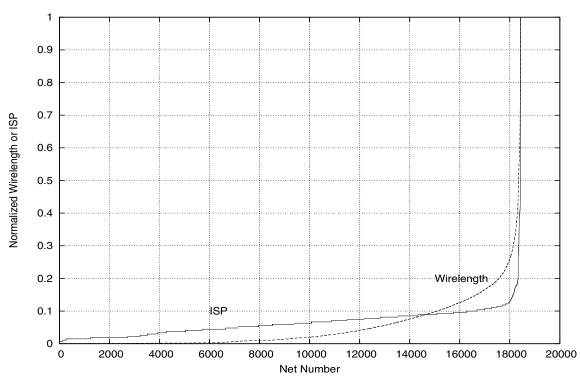


Figure 5: Wirelength and ISPL profiles or distributions.

2.5 Wirelength and ISPL Distributions

Finally, we investigate the complete *distribution* or *profile* of net lengths and ISPLs. We first rank all nets based on their length. Second, all nets are ranked with respect to their ISPL. Finally, we plot the net length and ISPL profiles for the entire 18429 nets of the ibm02 benchmark in Figure 5, where the plots are normalized with respect to their maximum values. We notice that both ISPL and wirelength share the following characteristics: (i) most of the nets have small ISPL or wirelength, and (ii) very few nets have large values of ISPL or wirelength. The ISPL profile, however, has fewer nets for small values of normalized ISPL and more nets for larger values of normalized ISPL. We have observed the same trends with other benchmarks. Overall, from the plot we can safely conclude that ISPL and wirelength profiles are similar.

3 Netlist Characterization and Relationship to Rent Parameter

Given the strong performance of ISPL as an estimator, it becomes interesting to ask if there is a relationship between Rent parameter and ISPL[5]. To uncover this relationship, we propose

[5]Rent's rule is observed in "good" placements and reflects a power-law scaling of the number of external terminals of a netlist region with the number of nodes in the region. Rent's rule is expressed as $T \propto G^p$, where p is Rent parameter, T is the number of external terminals, and G is the number of gates. A higher Rent parameter typically translates to more global communication. Rent parameters of modern circuits are in the vicinity of 0.6.

bench	Range	bench	Range
ibm01	0.976	ibm10	0.985
ibm02	0.962	ibm11	0.984
ibm03	0.985	ibm12	0.985
ibm04	0.981	ibm13	0.980
ibm05	0.892	ibm14	0.984
ibm06	0.960	ibm15	0.986
ibm07	0.985	ibm16	0.988
ibm08	0.956	ibm17	0.987
ibm09	0.974	ibm18	0.944

Table 5: Correlation coefficient between a node's range and its average wirelength. The average is 0.971.

a new characterization of VLSI netlists by a single parameter, which we call *Range parameter*, that is derived from intrinsic shortest path length calculations.

We define the *neighbors* of some node v as the set of nodes that are adjacent to v via some hyperedge incident to both. With respect to some placement, we say that node v has a larger *range* than node u if the average wirelength required to connect v to its neighbors is greater than the average wirelength required to connect u to its neighbors[6]. The average wirelength required to connect a node to its neighbors is basically the total wirelength of the hyperedges incident to the node divided by the degree of the node. An estimate $\mathcal{R}(u)$ of node range based on ISPL can be given as follows.

$$\mathcal{R}(u) = \frac{\sum_{e \in adj(u)} \mathcal{ISP}(e)}{|adj(u)|}. \qquad (3)$$

Fundamentally, the range of a node gives an estimate of the wirelength a node needs to connect to its neighbors in a good placement. A larger node range indicates that that the neighbors are more *sparse* with respect to the node. The *average range parameter*, or simply *Range parameter* of a netlist netlist is just the average node range of all its nodes $\mathcal{R} = \frac{1}{|V|}\sum_{u \in V} \mathcal{R}(v)$. A large Range parameter predicts that a netlist would require a large amount of global communication. This brings an intuitive connection to Rent parameter, since a netlist with large Rent parameter essentially means that the netlist requires more global communication in any good placement [9].

To confirm the relationship between the Range and Rent parameters, we calculate the Rent parameter [1] and the Range parameter for the full suite of the IBM benchmarks and tabulate the results in Table 6. From the table of results, the average correlation is 0.701 indicating a strong correlation between these two parameters.

On comparing the Rent and Range parameters, we find that:

- Rent parameter is calculated on entire circuit basis in a top-down fashion and is of little use on an individual net or node basis. On the other hand, the Range parameter is calculated in a bottom-up fashion on a per net or node basis. Typically, *a priori* estimation studies use Rent parameter to "reverse engineer" netlist characteristics, while the Range parameter is calculated from ISPLs and node degrees. Thus it characterizes on an individual

[6]Our Range definition should not be confused with "net range" as defined by [24]. They are entirely different concepts.

bench	Rent	Range	bench	Rent	Range
ibm01	0.531	8.840	ibm10	0.672	10.070
ibm02	0.722	11.652	ibm11	0.641	9.192
ibm03	0.671	9.525	ibm12	0.696	10.580
ibm04	0.678	10.510	ibm13	0.653	9.322
ibm05	0.755	11.870	ibm14	0.688	10.103
ibm06	0.678	10.451	ibm15	0.645	10.335
ibm07	0.685	9.710	ibm16	0.660	10.810
ibm08	0.677	11.966	ibm17	0.701	11.891
ibm09	0.605	9.968	ibm18	0.676	11.297

Table 6: Rent parameter vs. Range parameter. A strong correlation of 0.701 on average is observed between the two parameters.

	Runtime			Runtime	
bench	Rent	Range	bench	Rent	Range
ibm01	0.21	0.18	ibm10	0.34	0.77
ibm02	0.28	0.45	ibm11	0.63	0.41
ibm03	0.37	0.55	ibm12	0.60	0.86
ibm04	0.30	0.41	ibm13	0.50	0.54
ibm05	0.47	1.64	ibm14	1.00	0.61
ibm06	0.39	0.86	ibm15	1.05	1.18
ibm07	0.36	0.62	ibm16	1.25	0.56
ibm08	0.25	0.46	ibm17	1.85	1.92
ibm09	0.24	0.27	ibm18	1.51	0.89

Table 7: Runtime requirements for calculating all ISPLs of a netlist and then the Range parameter, versus Rent parameter calculation runtime, normalized to the runtime of FengShui (version 2.6).

basis – for either net or node – before characterizing the complete netlist, and hence avoids the need for reverse engineering.

- A netlist might have different Rent parameters depending on whether the parameter is calculated from partitioning or placement [17, 34]. In addition, the Rent parameter has inherent noise since it relies on partitioning heuristics, the results of which typically vary. On the other hand, our parameter is straightforward to calculate and has a constant, well-defined value for any given netlist.

Depending on the structure and size of the netlist, the ISPLs and the Range parameter might take some computational runtime to calculate; such runtime stems from the requirement to calculate the intrinsic shortest path of every hyperedge. In case there are hyperedges with large cardinality, the requirement to calculate the ISPL for every pair of nodes incurs a tremendous increase in runtime. For example, a 20-pin net requires 190 shortest path evaluations. To dramatically reduce the runtime with a negligible impact to accuracy, we can calculate the ISPL for a subset of pairs and declare the maximum value as the ISPL of the hyperedge. One possibility is to just calculate the maximum ISPL from one node to all other nodes. For each benchmark in the IBM suite, we tabulate the runtime requirements for executing ISPL-based estimation and runtime requirements for calculating the Rent parameter normalized to FengShui runtime in Table 7. We can see that the Rent and Range parameters more or less demand the same amount of runtime. Nevertheless, it is likely that runtimes will need to be further reduced for practical deploying of ISPL-based estimation in CAD applications.

4 Applications

In this section, we examine the potential of using ISPs for two practical applications: (1) *a priori* total wirelength estimation, and (2) *a priori* global interconnect prediction. To simplify our models, we modify the IBM benchmarks to have unit-size cells.

4.1 A Priori Total HPWL Estimation

To estimate wirelength *a priori* from ISPL, an analytical model between ISPL and wirelength must be devised. Using empirical data, we find an exponential relationship between ISPL and wirelength. If $w_k(s)$ is a function that gives the wirelength of a k-pin net with an ISPL value of s then

$$w_k(s) = a_k e^{g_k s}, \tag{4}$$

where a_k and g_k are constant coefficients that depend on the netlist under consideration. Figure 6 gives the actual data and the exponential fitting results of 2-pin and 3-pin nets of the IBM01 benchmark. For two-pin nets $w_2(s) = 1.56e^{0.12s}$, and three-pin nets $w_3 = 2.09e^{0.17s}$.

The coefficients a_k and g_k present a modeling dilemma since they can only be precisely determined from a fit between ISPL and wirelength. This *ideal modeling* is however unavailable in any *a priori* application. To overcome this difficulty, we propose to use fixed values for these coefficients that do not depend on the netlist being evaluated. These fixed values can be calculated based on typical values encountered in design netlists and stored in a lookup table in a manner similar to wireload lookup models [16]. We refer to modeling using fixed lookup values as *static modeling*.

Suppose we want to calculate the fixed lookup coefficients (a_2, g_2) for 2-pin nets from a given number q of actual fits[7]:

$$\begin{aligned} w_2^1(s) &= a_2^1 e^{g_2^1 s}, \\ w_2^2(s) &= a_2^2 e^{g_2^2 s}, \\ &\vdots \\ w_2^q(s) &= a_2^q e^{g_2^q s}. \end{aligned}$$

Our proposal is to seek an estimation function $w_2(s) = a_2 e^{g_2 s}$ that minimizes the total squared error from the given q fits. To realize such goal, we first linearize the given fits to

$$\begin{aligned} y^1 = \ln w_2^1(s) &= \ln a_2^1 + g_2^1 s, \\ y^2 = \ln w_2^2(s) &= \ln a_2^2 + g_2^2 s, \\ &\vdots \\ y^q = \ln w_2^q(s) &= \ln a_2^q + g_2^q s, \end{aligned}$$

and then find a linear fit $\ln w_2(s) = \ln a_2 + g_2 s$ that minimizes the total squared error using standard regression techniques. The linear fit is then readily converted to an exponential fit $w_2(s) = a_2 e^{g_2 s}$. This process of finding fits that minimizes the total squared error is repeated for all k-pin nets, and the set of values (a_k, g_k) is then stored in a lookup table for *a priori* wirelength estimation purposes.

[7] The superscripts on the coefficients give the fit number and do not indicate any power relation.

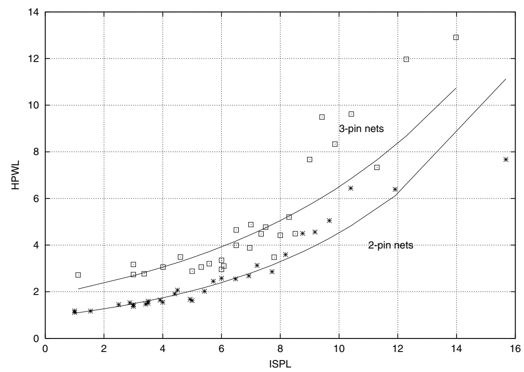


Figure 6: Exponential fitting for relationship between ISPL and wirelength for the IBM01 benchmark. For two-pin nets $w_2(s) = 1.56e^{0.12s}$, and three-pin nets $w_3 = 2.09e^{0.17s}$.

bench	static model	precise model	bench	static model	precise model
ibm01	57.93%	-5.73%	ibm10	7.36%	9.39%
ibm02	-8.87%	-0.63%	ibm11	-2.82%	-1.45%
ibm03	2.28%	-5.58%	ibm12	-13.55%	-1.56%
ibm04	28.86%	-8.94%	ibm13	-8.26%	0.22%
ibm05	-34.80%	-16.70%	ibm14	-15.09%	0.64%
ibm06	1.86%	-4.06%	ibm15	-26.23%	3.14%
ibm07	-7.02%	-1.39%	ibm16	-15.55%	0.44%
ibm08	8.69%	0.03%	ibm17	-30.94%	-0.34%
ibm09	18.04%	-3.52%	ibm18	-10.65%	-1.11%

Table 8: A priori total HPWL estimation using ISPL. Table entries give the percentage difference between the estimated wirelength according to the different models and the wirelength FengShui placements. Using the absolute values of the percentages, the ideal model gives 3.61% average difference from actual HPWL, and the approximate model gives 16.60% average difference from actual HPWL.

Our wirelength estimation results for both the precise and static models are given in Table 8. We report the percentage difference between the estimated results and the actual wirelength. The results of the precise model are not *a priori* and only serve to confirm our modeling accuracy. The results of the static model are certainly *a priori*. Using the absolute values of the percentage difference, we find that on the average the precise model results are 3.61% accurate, and the static model results are 16.60% accurate.

4.2 A Priori Global Interconnect Prediction

Another practical application for ISPL is *a priori* Global Interconnect (GI) prediction. Global interconnects are harmful to performance [20], [29] and are typically buffered once identified. In this subsection, we study the application of ISPL to identify GIs before any placement is executed. First, all nets are sorted by their ISPL and then all nets with an ISPL above a fixed *threshold* are flagged as GIs. This set of nets will be referred to as the *predicted GIs*. The success of ISPL as a GI predictor hinges on the percentage of actual GIs captured in the predicted GIs.

To empirically evaluate the success of ISPL as a GI predictor, we consider the IBM03 benchmark as follows. We first

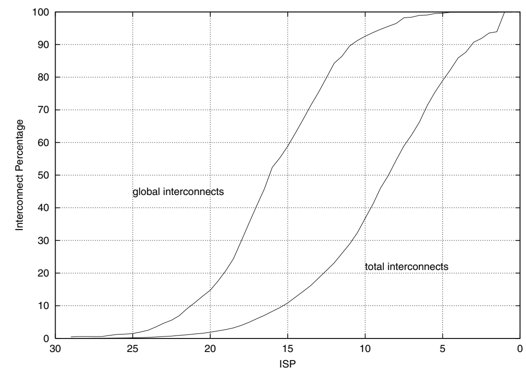


Figure 7: A Priori Global Interconnect Prediction using ISPL. The *global interconnects* curve gives the amount of actual GIs captured for each ISPL threshold value. The *total interconnects* curve gives the total amount of interconnects that are declared as predicted GIs for each ISPL threshold.

sort all nets according to their ISPLs and then place the benchmark using FengShui. From the placement, the actual GIs are extracted and compared to the predicted GIs. We consider an interconnect to be global if it belongs to the top 5% longest nets. In the plot of Figure 7, various thresholds of ISPL are on the x-axis, and for each ISPL threshold, we plot (1) the percentage of predicted GIs that turned out to be actual GIs, and (2) the total amount of interconnects that have an ISPL above that threshold.

For example, if we declare all the nets with an ISPL greater than or equal 15 as GIs, then, from the plot, we capture around 60% of the actual GIs and in the process we declare 10% of the total design nets as global.

5 Conclusions and Future Work

In this paper we have proposed a novel *a priori* wirelength estimator. Our estimator is based on intrinsic shortest path computations in the netlist, and we have shown through experimental results that our estimator is a very strong predictor of wirelength characteristics. To summarize, we have

- demonstrated strong correlation between wirelength and ISPL averages,
- established strong correlation between the effect of pin count on both net length and ISP,
- established lower and upper bounds on the performance of any individual net length predictor,
- shown that ISPL is a good predictor of individual net length,
- shown that average ISPL is very strongly correlated to average wirelength, and that its increases perfectly correlate with increases in total wirelength,
- studied the relationship between the ISPL distribution and wirelength distribution.

We have also developed the Range parameter which characterizes VLSI netlists with a succinct value, and established its relation to the Rent parameter. We have also demonstrated two successful applications of ISPL: *a priori* wirelength prediction, and *a priori* global interconnect prediction. There are still some "weaknesses" in ISPL: runtime requirements, and estimating wirelength when two nets have the same ISPL. There is also plenty of room for future work. Open directions include developing synthetic benchmarks with realistic characteristics based on ISPL profiles and Range parameters; and using ISPL in a physical-driven synthesis environment.

References

[1] "http://er.cs.ucla.edu/Dragon/download.html," *Rent parameter calculator*.

[2] "http://www.elis.ugent.be/ dstr/gnl/," *Synthetic netlist generator*.

[3] S. Balachandran and D. Bhatia, "A Priori Wirelength and Interconnect Estimation Based on Circuit Characteristics," in *Workshop on System Level Interconnect Prediction*, 2003, pp. 77–84.

[4] S. Bodapati and F. N. Najm, "Pre-Layout Estimation of Individual Wire Lengths," in *Workshop on System Level Interconnect Prediction*, 2000, pp. 91–96.

[5] A. E. Caldwell *et al.*, "GTX: The MARCO GSRC Technology Extrapolation System," in *Proc. ACM/IEEE Design Automation Conference*, 2000, pp. 693–698.

[6] A. E. Caldwell, A. B. Kahng, and I. L. Markov, "Can Recursive Bisection Alone Produce Routable Placements?" in *Proc. ACM/IEEE Design Automation Conference*, 2000, pp. 477–482.

[7] C.-C. Chang, J. Cong, D. Pan, and X. Yuan, "Multilevel Global Placement with Congestion Control," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 22(4), pp. 395–409, 2003.

[8] C. E. Cheng, "RISA: Accurate and Efficient Placement Routability Modeling," in *Proc. IEEE International Conference on Computer Aided Design*, 1994, pp. 690–695.

[9] P. Christie and D. Stroobandt, "The Interpretation and Application of Rent's Rule," *IEEE Transactions on VLSI Systems*, vol. 8(6), pp. 639–648, 2000.

[10] J. Cong and S. K. Lim, "Edge Separability-Based Circuit Clustering With Application to Multilevel Circuit Partitioning," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 23(3), pp. 346–357, 2004.

[11] J. Dambre, D. Stroobandt, and J. V. Campenhout, "Toward the Accurate Prediction of Placement Wirelength Distributions in VLSI Circuits," *IEEE Transactions on VLSI Systems*, vol. 12(4), pp. 339–348, 2004.

[12] J. A. Davis, V. K. De, and J. D. Meindl, "A Stochastic Wire-Length Distribution for Gigascale Integration (GSI) - Part I: Derivation and Validation," *IEEE Trans Electron Devices*, vol. 45, pp. 580–589, 1998.

[13] W. E. Donath, "Placement and Average Interconnection Lengths of Computer and Systems," *IEEE Trans. Circuits and Systems*, vol. 26), pp. 272–277, 1979.

[14] ——, "Wire Length Distribution for Placements of Computer Logic," *IBM J. Res. Develop*, vol. 25(3), pp. 152–155, 1981.

[15] W. Gosti, S. P. Khatri, and A. L. Sangiovanni-Vincentelli, "Addressing the Timing Closure Problem by Intgerating Logic Optimization and Placement," in *Proc. IEEE International Conference on Computer Aided Design*, 2001, pp. 224–231.

[16] W. Gosti, A. Narayan, R. K. Brayton, and A. L. Sangiovanni-Vincentelli, "Wireplanning in Logic Synthesis," in *Proc. IEEE International Conference on Computer Aided Design*, 1998, pp. 26–33.

[17] L. Hagen, A. B. Kahng, F. J. Kurdahi, and C. Ramachandran, "On the Intrinsic Rent Parameter and Spectra-Based Partitioning Methodologies," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 13(1), pp. 27–37, 1994.

[18] T. Hamada, C.-K. Cheng, and P. M. Chau, "A Wire Length Estimation Technique Utilizing Neighborhood Density Equations," in *Proc. ACM/IEEE Design Automation Conference*, 1992, pp. 57–61.

[19] M. Hrkić, J. Lillis, and G. Beraudo, "An Approach to Placement-Coupled Logic Replication," in *Proc. ACM/IEEE Design Automation Conference*, 2004, pp. 711–716.

[20] A. B. Kahng, I. L. Markov, and S.Reda, "Boosting: A Min-Cut Placement with Improved Signal Delay," in *Design, Automation and Test in Europe*, 2004, pp. 1098–1103.

[21] P. Kudva, A. Sullivan, and W. Dougherty, "Metrics for Structural Logic Synthesis," in *Proc. IEEE International Conference on Computer Aided Design*, 2002, pp. 551–556.

[22] B. Landman and R. Russo, "On a Pin versus Block Relationship for Partitions of Logical Graphs," *IEEE Transactions on Computers*, vol. 20(C), pp. 793–813, 1971.

[23] Q. Liu, B. Hu, and M. Marek-Sadowska, "Wirelength Prediction in Constraint Driven Placement," in *Workshop on System Level Interconnect Prediction*, 2003, pp. 99–105.

[24] Q. Liu and M. Marek-Sadowska, "A Study of Netlist Structure and Placement Efficiency," in *ISPD*, 2004, pp. 198–203.

[25] ——, "Pre-Layout Wire Length and Congestion Estimation," in *Proc. ACM/IEEE Design Automation Conference*, 2004, pp. 582–587.

[26] J. Lou, W. Chen, and M. Pedram, "Concurrent Logic Restructuring and Placement for Timing Closure," in *Proc. IEEE International Conference on Computer Aided Design*, 1999, pp. 31–36.

[27] D. Pandini, L. T. Pileggi, and A. J. Strojwas, "Congestion-Aware Logic Synthesis," in *Design, Automation and Test in Europe*, 2002, pp. 664–671.

[28] M. Pedram and B. Preas, "Accurate Prediction of Physical Design Characteristics for Random Logic," in *International Conference on Computer Design*, 1989, pp. 100–108.

[29] R. Otten and R. Brayton, "Planning For Performance," in *Proc. ACM/IEEE Design Automation Conference*, 1998, pp. 122–127.

[30] S. Sastry and A. Parker, "On the Relation between Wire Length Distributions and Placement of Logic on Master Slice ICs," in *Proc. ACM/IEEE Design Automation Conference*, 1984, pp. 710–711.

[31] D. Stroobandt, J. Depreitere, and J. V. Compenhout, "Generating New Benchmark Designs using a Multi-Terminal Net Model," *Integration, the VLSI Journal*, vol. 27(2), pp. 113–129, 1999.

[32] D. Stroobandt, *A Priori Wire Length Estimates for Digital Design*. Kluwer Academic Publishers, 2001.

[33] M. Wang, X. Yang, and M. Sarrafzadeh, "DRAGON2000: Standard-Cell Placement Tool for Large Industry Circuits," in *Proc. IEEE International Conference on Computer Aided Design*, 2001, pp. 260–263.

[34] X. Yang, E. Bozorgzadeh, and M. Sarrafzadeh, "Wirelength Estimation based on Rent Exponents of Partitioning and Placement," in *Workshop on System Level Interconnect Prediction*, 2001, pp. 25–31.

[35] M. Yildiz and P. Madden, "Global Objectives for Standard-Cell Placement," in *Proc. IEEE Great Lakes Symposium on VLSI*, 2001, pp. 68–72.

[36] P. Zarkesh-Ha, J. A. Davis, and J. A. Meindl, "Prediction of Net-Length Distribution for Global Interconnects in a Heterogeneous System-On-a-Chip," *IEEE Trans on VLSI Systems*, vol. 8(6), pp. 649–659, 2000.