On Wirelength Estimations for Row-Based Placement

Andrew E. Caldwell, Andrew B. Kahng, Stefanus Mantik,
Igor L. Markov and Alexander Zelikovsky
UCLA Computer Science Department, Los Angeles, CA 90095-1596, USA

Abstract

Wirelength estimation in VLSI layout is fundamental to any pre-detailed routing estimate of timing or routability. In this paper, we develop efficient wirelength estimation techniques appropriate for wirelength estimation during top-down floorplanning and placement of cell-based designs. Our methods give accurate, linear-time approaches, typically with sublinear time complexity for dynamic updating of estimates (e.g., for annealing placement). Our techniques offer advantages not only for early *on-line* wirelength estimation during top-down placement, but also for *a posteriori* estimation of routed wirelength given a final placement. In developing these new estimators, we have made several contributions, including (i) insight into the contrast between *region-based* and *bounding box-based* RStMT estimation techniques; (ii) empirical assessment of the correlations between pin placements of a multi-pin net that is contained in a block; and (iii) new wirelength estimates that are functions of a block's complexity (number of cell instances) and aspect ratio AR.

I. Introduction

Wirelength estimation in VLSI layout is fundamental to any pre-detailed routing estimate of timing or routability. Accordingly, wirelength estimation has been studied in such contexts as gate-array routability [7], hierarchical top-down layout [6] [8] [22], floorplanning [11], and growth rates of rectilinear Steiner minimal trees [20] [21] [3]. Our present work is aimed at wirelength estimation *during* the top-down placement of cell-based designs.

We distinguish three basic types of wirelength estimations associated with placement: *a priori*, *a posteriori* and *on-line*.[1]

- *A priori* estimation seeks to estimate the total wirelength of a layout design in advance, before placement. For example, a floorplanner may use such estimates to obtain rough measures of routability, RC parasitics and circuit performance; these in turn drive floorplan changes and circuit optimizations. For such estimates to provide leverage, they must be faster than the actual placement or routing constructions, at the cost of reduced accuracy. Such estimates are typified by the "wireload models" used in RTL floorplanning and logic optimization.
- *A posteriori* estimation occurs when we are given a fixed placement and want to estimate the post-routing wirelength. This is of value whenever routing requires significantly more CPU time than placement or wiring estimation. Typical applications include predicting the routability of gate

This research was supported by a contract and grant from Cadence Design Systems, Inc.

[1]We do not discuss the class of *constructive* wiring estimators, which essentially construct the layout down to global or detailed routing in order to obtain an "estimate" of the wiring. Constructive estimators can be relevant to certain design methodologies, but we are more concerned with early and fast predictions that afford the leverage essential to forward synthesis.

array layouts [7] [8], estimating channel height in standard-cell layouts [17] [18], choosing between two competing placements, etc. Again, such estimates must be faster than actual construction of the routing. (Note that accuracy need not be perfect if the estimate has good "fidelity", i.e., for any two solutions the estimator correctly predicts which one is better even if the estimate of relative solution costs may be inaccurate.)

- *On-line* estimation occurs when we want to estimate the wirelength *during* top-down hierarchical floorplanning or placement. This has many applications. For example, the estimate can be used to stop the placement process early, as soon as it becomes obvious that the placement process is leading to a bad solution. Early estimates of wirelength can also be used to shorten the feedback loops in timing- and wirelength-driven placement: clock tree synthesis, scan ordering, gate sizing, etc. may all be done earlier in the flow when good wiring estimates are available. Finally, wirelength estimates can be useful in determining the merit of local perturbations to the current solution. For example, the inner loop of a simulated annealing placer requires accurate estimation of the quality of a proposed move. Since this incremental cost estimation is one of the main contributors to annealing placement runtime, an on-line estimator must be very fast. The accuracy of on-line wirelength estimation should be between those for *a priori* and *a posteriori* regimes, reflecting the available information (more information than *a priori*, less than *a posteriori*).

Previous Wirelength Estimation Techniques

A fundamental building block for "wirelength estimation" is estimation of the rectilinear Steiner minimal tree (RStMT) cost.[2] The input to the RStMT problem is a pointset P of size $|P| = n$. In one common context, we view P as having been selected from a uniform distribution over a rectangular region R with width w_R and height h_R. Alternatively, we may know the minimum bounding box enclosing all points of P, having width w_{bb} and height h_{bb}. We now review the most relevant literature for this problem.

Growth rates of subadditive functionals in the Euclidean plane. The work of [21] [19], in a literature that stems from the seminal work of Beardwood et al. [1], shows that the expected cost (total tree length) of a Euclidean Steiner minimal tree over n points uniformly distributed within a bounded plane region R of area $w_R \cdot h_R$ is proportional to $\sqrt{w_R \cdot h_R \cdot n}$ for sufficiently large n. The constant of proportionality, denoted by β, is dependent on the functional of the pointset (e.g., the

[2] We understand that router outputs may not be the same as RStMTs (due to the routing heuristic, congestion, timing or noise constraints, and obstacles), and we understand that minimum wirelength is not perfectly correlated with minimum delay or maximum routability. Nevertheless, pure RStMT cost estimation remains a core technology within today's industry floorplanners, I/O pin optimizers, global and detailed placers, and related tools. Our ongoing work is developing extensions to model the effects of routability and performance optimization.

Euclidean minimum spanning tree cost and the Euclidean minimum traveling salesperson tour cost have similar growth rates but different constants of proportionality β_{EMSpT}, β_{ETSP}, etc.).

MSpT-based methods. Hwang [12] shows that the rectilinear Steiner ratio (worst-case ratio of rectilinear minimum spanning tree (RMSpT) cost to RStMT cost) is 3/2. Hence, 2/3 times the RMSpT cost is a lower bound on RStMT cost. Since the RMSpT cost is an upper bound on RStMT cost, one might propose, e.g., 5/6 times the RMSpT cost as an RStMT estimator that guarantees at most 16% error. Alternatively, empirical studies show that $\frac{cost(RStMT)}{cost(RMSpT)}$ averages around 0.88; see [14] for a review. While MSpT-based estimators are excellent, we avoid them because their implementation requires $\Omega(n \log n)$ runtime with fairly high constant, or else $\Omega(n^2)$ runtime.[3]

Bounding box based methods. In iterative improvement placers, the objective is typically based on the half-perimeter of the bounding box of pin locations for each net, i.e., the RStMT estimate is $w_{bb} + h_{bb}$. This is computed in linear time; given appropriate data caching, it can be updated in expected sublinear time when a cell is moved. The bounding box half-perimeter exactly gives the RStMT cost for 2- and 3-pin nets, and can be fairly accurate for larger nets if the bounding box aspect ratio becomes large (see below). However, during top-down placement or floorplanning, the pin locations are typically snapped to the centers of the *regions* in which they are located, after which the bounding box computation occurs. This can induce certain errors (see Section 2 below), and corrections for special cases have been proposed by Donath [6] and subsequent authors.

Chung and Hwang [3] study the worst-case cost of the rectilinear Steiner minimal tree (RStMT) over n points with bounding box dimensions w_{bb}, h_{bb}. The maximum value of $\frac{cost(RStMT)}{w_{bb}+h_{bb}}$ tends to $\frac{\sqrt{n}+1}{2}$ as $n \to \infty$. Several authors (e.g., Sechen) have noted that this result implies a correction factor to the bounding box half-perimeter estimate for nets with $|P| > 3$.

Hamada et al. [10] propose a purely *a priori* wirelength estimation based on local neighborhood analysis. Nets are expanded into cliques, and 2-neighborhoods of each are analyzed to obtain parameters of branching within the circuit. Multi-pin net wirelength estimates are inferred from these parameters and a physical model in which neighbors of a given cell compete for locations close to that cell.

Cheng [2] empirically estimates the probability of having a wire pass through any given point within a net bounding box when the net is routed. His methodology is equivalent to estimating the

[3]Our work emphasizes fast (hopefully, linear-time) estimators that fit a dynamic or on-line use model. We recognize that MSpT-based estimators return an actual topology, as opposed to just a cost estimate. For delay estimation, our experience with industrial deep-submicron libraries and process technologies is that with well-balanced and well-sized circuits the resistive interconnect effects are not dominant, e.g., lumped-capacitance or simple C_{eff} estimates as in [13] are adequate [15] [16]. For noise estimation, whether any routing estimate (as opposed to a detailed Steiner embedding of all nets) can be useful is yet unclear.

horizontal and vertical components of the RStMT cost, as a correction factor to the sum $(w_{bb} + h_{bb})$. The correction factor is a function of the net size n; [2] provides a table of such correction factors obtained by Monte Carlo methods.

Vaishnav and Pedram [23] suggested a priori wirelength estimations based on appropriate scaling of the net bounding box half-perimeter by the number of pins on the net according to the Chung and Hwang [3] formula. They then estimate the net bounding box assuming a uniform pin distribution.

Donath's classic wirelength estimation technique, which is based on Rent's rule and which estimates hierarchical interconnections as opposed to RStMTs, is extended in works by Stroobandt et al. and other groups. Example works are [22] and [11]. As noted in [5], classical works on wirelength estimation, e.g., those of Donath ([6]), only deal with 2-pin nets. Even though modern designs typically contain up to 60% 2-pin nets, this limitation is unacceptable because 2-pin nets rarely account for more than half of total wirelength. To fill this gap, [4] and [5] study multi-pin nets and produce accurate total wirelength estimates by again extending Donath's original technique. The Rent-based wirelength estimation literature uses different techniques than we do; our efforts are more closely related to the RStMT estimation literature.

Contributions of This Paper

Our work makes the following contributions.

- We empirically study the distribution of pin locations in blocks during top-down placement. Based on these studies, we model the set of all pins belonging to nets wholly contained in a block as being uniformly distributed within the block. Since the pins of any given net have correlated locations, we apply "shrinking" corrections to a given block's dimensions during the wirelength estimations for that block's wholly-contained and partially-contained nets.
- We develop bounding box estimators that form the basis of our new on-line wirelength estimators. Specifically, we give both an exact $O(n^2)$ algorithm and two heuristics ($O(n)$ and $O(n \log n)$) for computing the expected bounding box of n points with known distribution among k regions of a floorplan or hierarchical placement. These methods apply to arbitrary distributions. Our heuristics are considerably faster than the exact algorithm even for small values of n, which makes them useful in the placement context.
- We make new insights into the contrast between asymptotic results of Steele et al. (that expected RStMT cost is proportional to $\sqrt{w_R \cdot h_R \cdot n}$) and the accepted practice (that expected RStMT cost is proportional to $\sqrt{n} \cdot (w_{bb} + h_{bb})$). Among other implications, we find that an estimator of expected

RStMT cost should not simply be based on a single constant β. Rather, such an estimator must be based on a set of values $\beta(n, AR)$ (where AR is the aspect ratio of the region within which n points are uniformly distributed, or else the aspect ratio of the pointset's bounding box).

- Finally, we develop new wirelength estimators based on the above results.[4] Our new estimators are substantially more accurate than previous methods that are used in industry tools, including bounding box methods and the method of Cheng [2]. In the on-line regime, we achieve stable and accurate prediction of final total RStMT cost or total bounding box half-perimeter cost very early in the top-down placement process.

II. Pin Distributions in Good Layouts

Since wirelength is ultimately a function of pin locations, we must first examine the issue of creating a statistical model for pin locations in placements. This is a difficult issue, since it is dependent not only on the input netlist topology[5] but also on the quality of the placement tool used. Final pin locations are not known at higher levels of the top-down placement (which is where we require estimates) and the estimator must account for this uncertainty. In [10] such uncertainty is handled with the Weibull distribution for wirelength. Stroobandt and Kurdahi [5], [4] use statistical models of the number of "cut" and "uncut" nets on the levels of top-down placement, other empirical parameters and a series of statistical assumptions. However, we are not aware of any rigorously justifiable model of how pin locations are distributed during the placement process.

In our work, we seek to estimate wirelength of individual nets by modeling pin locations within blocks as random variables having certain statistical properties. Such an approach does not assume that all possible placements are equally likely as good final placements, but rather attempts to discern statistically significant attributes of good placements. Given any arbitrary model for the distribution of pin locations, we will embed it within a generic calculation of the expected bounding box dimensions of the placed pins for given nets. Compared to previous works, this allows our estimators to use the sizes of hierarchical blocks that contain pins of a net, as well as the number of pins of a net that each block contains.

The remainder of this section describes our approach to empirically finding simple, effective and

[4] Because much of our approach is empirical, tuned variants are easy. E..g., a given routing tool's characteristics can be captured by using the router's results, rather than the output of an RStMT heuristic, to create the lookup tables. Or, the quality of a given placer can be captured in the coefficients of our "shrinking" corrections.

[5] For example, the distribution of final locations in good placements for a given cell A's pins will not be uniform over the entire layout area, if cell A is strongly connected to a fixed pad. Another cell B's pins will also have non-uniform distributions within good placements, if B is strongly connected to A. While we cannot predict this, the main point of this section is that we can exploit the fact that the *average* of all distributions of individual pin locations will be uniform over the layout region.

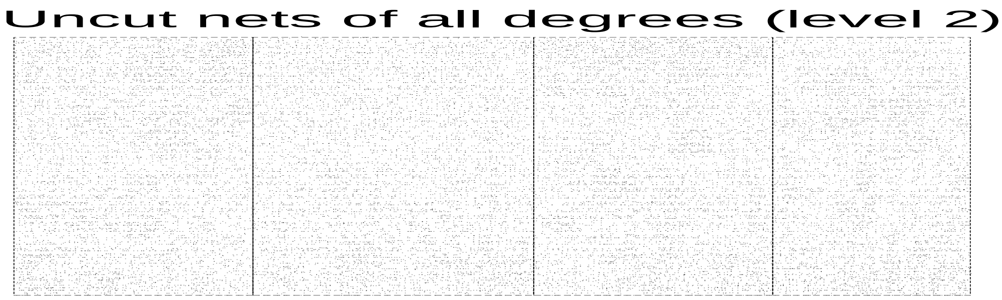

Fig. 1. Typical distribution of pin locations in a low-wirelength final solution for all "uncut" nets at the second level of top-down partitioning-based placement. The uniformity implies that individual pin distributions over any number of good solutions will *average* to a uniform distribution.

efficient models of distributions for locations of pins on the same net. Based on how these models combine with our other results to yield high-quality wirelength estimators, we are hopeful that our approach can be extended by better, more formal models – and we leave this as an open research question.

Models With Uniform Pin Distributions

Standard-cell designs in processes with four or more metal layers have very high area utilization (e.g., less than 5% whitespace for a typical synthesized block); also, the nature of logic synthesis tools and static CMOS technology tend to result in low average pin count per cell. It is natural and practical to assume that legal cell sites in utilized areas of blocks (i.e., not obstacles) of cell-based designs are *uniformly* distributed. We have studied pin distributions in placed industry designs with respect to blocks from various levels of the top-down placement. A typical plots is shown in Figure 1, which plots pin locations in a low-wirelength solution for all "uncut" nets (i.e., having pins in exactly one block) at the second level of placement. These studies suggest that locations of pins in nets of any particular degree are also distributed uniformly in good placements. Hence, when a net has pins (on cells) assigned to a given hierarchical block, we will assume that the pin locations are uniformly distributed within that block. Such an assumption is reasonable in the sense that whatever distributions individual pins may have over the space of all good placements, these distributions must *average* to a uniform distribution.[6]

On the other hand, Figure 2 shows that pins of "cut" nets on the third placement level are not uniformly distributed. The apparent non-uniformity is because wirelength minimization forces pins of cut nets toward cut lines and toward meeting points of multiple blocks. This is clearly seen

[6]This is rather a necessary condition — if the averaged distribution is not uniform, many pins will be distributed nonuniformly.

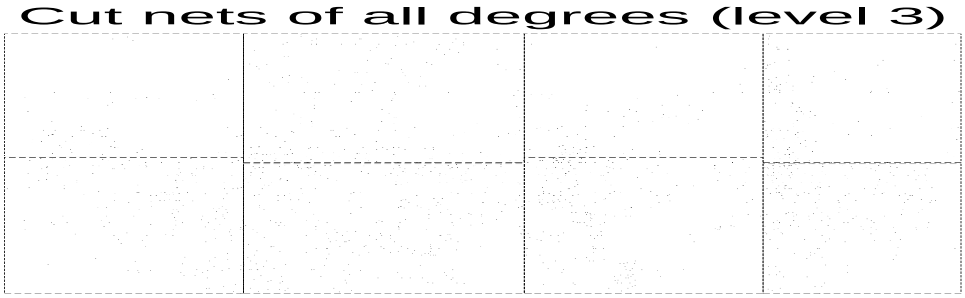

Fig. 2. Typical distribution of pin locations in a low-wirelength final placement for all "cut" nets at the third level of the top-down placement. The apparent non-uniformity is because wirelength minimization forces pins of cut nets toward cut lines and toward meeting points of multiple blocks.

in Figures 3 and 4, which respectively depict with line segments all 2-pin nets that are "cut" and "uncut" at the third placement level. The line segments in Figure 4 are substantially shorter on average than lines between independently distributed pins would be, while the line segments in Figure 3 are not, implying little correlation between pin locations. This data indicates that while uniform distribution models are reasonable for the aggregate of all pins of most types of nets, we must correct for the correlations (i.e., non-independence) of the pin locations of any given net. For the purposes of wirelength estimation, we will achieve such corrections using uniform distributions in shrunk and shifted blocks. Furthermore, following methods of [4], we find it reasonable to allow the possibility of different correlations (i) between pin locations of nets that are uncut at both the i^{th} and $(i+1)^{st}$ levels, and (ii) between pin locations of nets that are uncut at the i^{th} level, but cut at the $(i+1)^{st}$ level.

In summary, our empirical studies have led us to use independent uniform pin distributions – with separate corrections for observed correlations of cut and uncut nets – as a practical basis for wirelength estimation. This is borne out by our results in Section 6, below.[7] However, formally quantifying the correlations between locations, and using such results for wirelength estimation, remains a challenging direction for future work.

[7]Our analysis of placements has suggested two other reasons why our simple approach is successful. (1) At lower levels of the placement the small size of blocks enhances the accuracy of any estimator, and accurately modeling the location of pins in a block is less significant. At higher placement levels, there are few blocks and most nets have all their pins in one block (cf. Rent's rule). With these larger populations of nets, wirelength estimation based on the uniform distribution model appears quite accurate. (2) Exceptionally detailed models may not be necessary for half-perimeter wirelength or RStMT estimators, due to the many "don't care" regions intrinsic to the objective function (i.e., relocating cells may not significantly change the total wirelength estimate.) Again, simple models can be effective.

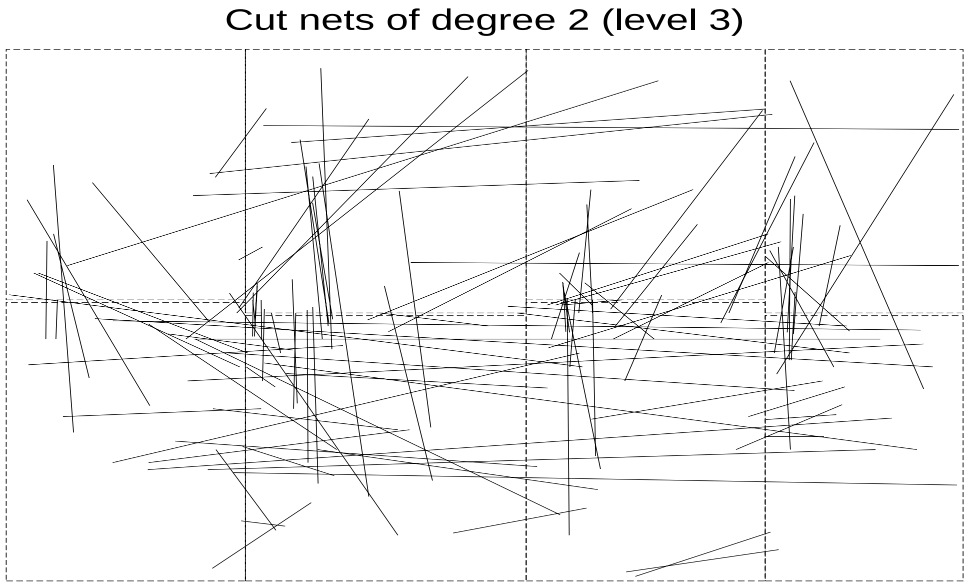

Fig. 3. "Cut" nets of degree two on the third placement level, represented by line segments.

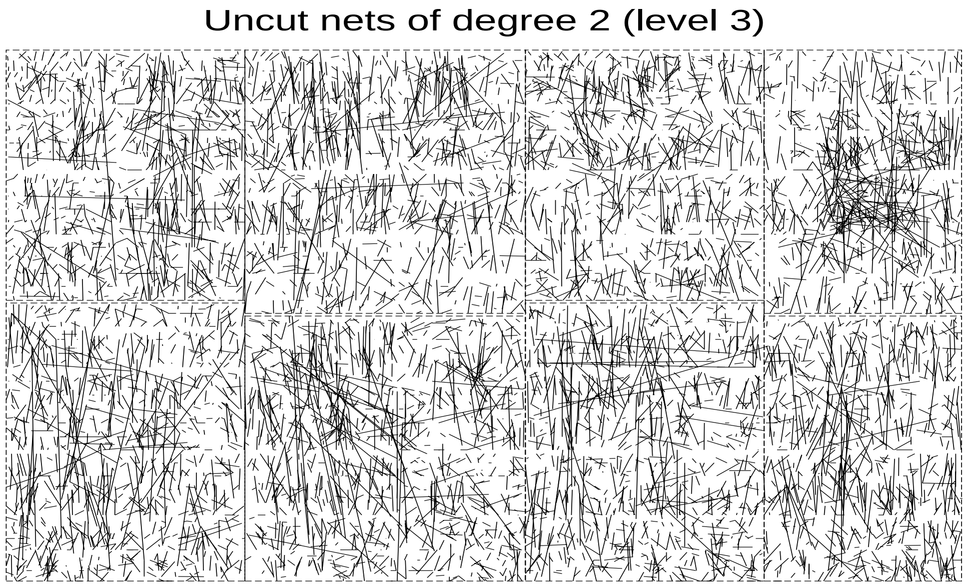

Fig. 4. "Uncut" nets of degree two on the third placement level, represented by line segments. The pin locations for a given net are strongly correlated: most nets span less than two cell rows.

III. The Bounding Box of n Random Points With Given Distribution Among k Rectangles

Hierarchical partitioning- and annealing-based placers maintain lists of rectangular regions and cells assigned to each region. Until the bottom level of the placement, cells may have no particular location, yet wirelength cost estimates are needed to drive further partitioning or annealing. Based on the conclusions of the previous section, for each net we will estimate the expected half-perimeter of its bounding box, assuming that each pin is uniformly distributed within the rectangular regions to which it are assigned.[8] Formally, we are given N rectangles R_i, $i = 1, \ldots, N$, in the rectilinear plane, and we are given that the i^{th} rectangle contains n_i uniformly distributed points. We seek estimates of the expected width and height of the bounding box of all $n = \sum_{i=1}^{N} n_i$ uniformly distributed points.[9]

Since the $x-$ and the $y-$coordinates of each pin are independent variables distributed uniformly within ranges in segments, we can estimate the two sides of the bounding box separately and add the results to obtain the expected half-perimeter. Since the expected side of the bounding box is simply the expected distance between the maximal and the minimal random point, finding these two expectations (i.e., of the maximal and minimal coordinates) will solve the problem. Let us specify a given rectangle R_i by its lower-left and upper-right corners $\{(a_i^x, a_i^y), (b_i^x, b_i^y)\}$ with $a_i^x \le b_i^x$ and $a_i^y \le b_i^y$. Then, the computation of the expected bounding box of the n points is given in Figure 5.

Computation of the Expected Bounding Box

Input: Rectangles $R_i = \{(a_i^x, a_i^y), (b_i^x, b_i^y)\}, i = 1 \ldots N$
each with n_i random points
Output: The expected width $\mathcal{E}_{width}$ and the expected height $\mathcal{E}_{height}$
of the bounding box of all points

For horizontal segments $[a_i^x, b_i^x]$ with n_i random points each:
find $\mathcal{E}_{left}$, the expected location of the leftmost point
find $\mathcal{E}_{right}$, the expected location of the rightmost point
For vertical segments $[a_i^y, b_i^y]$ with n_i random points each:
find $\mathcal{E}_{top}$, the expected location of the topmost point
find $\mathcal{E}_{bottom}$, the expected location of the bottommost point
Output $\mathcal{E}_{width} = \mathcal{E}_{right} - \mathcal{E}_{left}$ and $\mathcal{E}_{height} = \mathcal{E}_{top} - \mathcal{E}_{bottom}$

Fig. 5. Computing the expected bounding box for n points distributed over N rectangles.

In the remainder of this section, we will deal with computing the expected location of the *leftmost* point, because computing any of $\mathcal{E}_{right}$, $\mathcal{E}_{top}$, or $\mathcal{E}_{bottom}$ obviously reduces to computing $\mathcal{E}_{left}$:

The Expected Minimum Problem. *Given N segments $[a_i, b_i], i = 1, \ldots, N$ on the real line with n_i points distributed uniformly in the i^{th} segment, find the expected location of the point with minimum coordinate.*

The following subsection gives an exact $O(n^2)$ algorithm, and Subsection III-B gives $O(n)$ and

[8]The techniques that we develop below apply to the case where there is a known non-uniform probability distribution for pin locations within a given region (e.g., [11]).

[9]As explained in Subsection III-A, the straightforward and often-used heuristic – assuming that each cell is placed in the center of its rectangle (i.e., at its expected location assuming uniform distributions) – can have large error.

$O(n \log n)$ heuristics that we use as the basis of new estimators in later sections.

A. Exact Solution of the Expected Minimum Problem

We work with a random point $\mathcal{P}_i$ on a segment in terms of its *cumulative distribution function* $p_i(t) : [a_i, b_i] \to [0,1]$, which gives the probability of the point appearing to the left of t. Thus, $1 - p_i(t)$ gives the probability of the point appearing to the right of t. The uniform distribution corresponds to the cumulative distribution $p_i(t) = \frac{t-a_i}{b_i-a_i}$.

We can extend cumulative distribution functions by 0 to the left from a_i and by 1 to the right from b_i, allowing us to deal with random points *supported* on different segments (i.e., taking non-zero and non-one values of the cumulative distribution only on their respective segments).

Fact 1: For n independent random points with cumulative distributions $p_i(t)$, the distribution of the minimum is $1 - \prod_{i=1}^{n}(1 - p_i(t))$.

Proof. The probability that none of the n points appears to the left of t is the product of the probabilities that each of them does not appear to the left of t, i.e., $\prod_{i=1}^{n-1}(1 - p_i(t))$. Thus, the probability that at least one point is to the left of t is $1 - \prod_{i=1}^{n}(1 - p_i(t))$. □

Fact 2: The expected location of a random point with distribution $p(t)$ supported within $[A, B]$ is $E = B - \int_A^B p(t)\ dt$.

Proof. Let $\rho(t) = p'(t)$ be the probability density function of the distribution. Then $E = \int_A^B t\rho(t)dt = \int_A^B tp'(t)dt = (tp(t))|_A^B - \int_A^B p(t)dt = B - \int_A^B p(t)dt$ as $p(A) = 0$ and $p(B) = 1$. □

Since a point distributed on $[a_i, b_i]$ is also distributed on any containing segment (but not vice versa), one can enlarge $[a_i, b_i]$ to any $[A, B]$ when considering products in Fact 1 and the theorems below. Facts 1 and 2 imply

Fact 3: For n independent random points with cumulative distributions $p_i(t)$ supported within the segment $[A, B]$ (i.e. having its non-zero and non-one values within $[A, B]$), the expected minimum is

$$\mathcal{E}_{min} = B - \int_A^B (1 - \prod_{i=1}^{n}(1 - p_i(t)))\ dt = A + \int_A^B (\prod_{i=1}^{n}(1 - p_i(t)))\ dt \tag{1}$$

Fact 4: The expected minimum for k independent random points uniformly distributed on the segment $[0, 1]$ is $\frac{1}{k+1}$.

Proof.

$$\mathcal{E}_{min} = \int_0^1 (1-t)^k\ dt = \frac{1}{k+1}. \tag{2}$$

□

Symmetrically, the expected maximum of k independent random points uniformly distributed on the segment $[0,1]$ is $1-\frac{1}{k+1}$.

Corollary III.1: The expected difference between maximum and minimum for k independent random points uniformly distributed on the segment $[0,1]$ is $1-\frac{2}{k+1}$.

Example. If two points are uniformly distributed on $[0,1]$ the leftmost is expected at $\frac{1}{3}$, and the rightmost is expected at $\frac{2}{3}$. Consequently, the straightforward estimate for the expected distance between two random points as the distance between their expectations (0 in this case) will be wrong by $\frac{1}{3}$ of the bounding box size for the region over which the points are distributed (or 100% of the correct result).[10]

Theorem III.2: Consider n random points, each of which is independently and uniformly distributed on segment $[a_i, b_i]$, $i = 1, ..., n$. Let $A = a_1$ and $B = \min_i b_i$ and assume $a_i \le a_{i+1} \le \min_i b_i$. (Any segment with $a_k > B$ can be ignored.) Then the expected minimum $\mathcal{E}_{min}$ is

$$\mathcal{E}_{min} = A + \int_A^B \left(1 - \frac{t-a_1}{b_1-a_1}\right)\, dt - \sum_{i=2}^{n-1} \int_{a_i}^B \frac{t-a_i}{b_i-a_i} \prod_{j=1}^{i-1} \left(1 - \frac{t-a_j}{b_j-a_j}\right)\, dt \tag{3}$$

The proof is given in the Appendix.

Splitting the sum in Formula 3 into $n-2$ terms, we can view $\mathcal{E}_{min}$ as the sum of A and $n-1$ integrals of polynomials of degree at most n. To compute the integrand in term i, introduce $P_1(t) := 1$, $P_i(t) := \prod_{j=1}^{i-1}\left(1-\frac{t-a_j}{b_j-a_j}\right)$, which can be maintained in $O(n)$ time per component using multiplication by linear polynomials $P_i(t) = P_{i-1}(t)\frac{b_{i-1}-t}{b_{i-1}-a_{i-1}}$. Integrations can be performed symbolically in $O(n)$ time each, giving overall $O(n^2)$ complexity.

The Expected Minimum Algorithm
Input: segments $[a_i, b_i], i = 1..N$ each containing n_i random points **Output:** The expected position $\mathcal{E}_{min}$ of the minimum point

Set $B = \min_i b_i$, the smallest of the right segment endpoints
Discard all segments with left endpoint greater than B
Sort the segments by left endpoints, such that $a_1 \le \ldots \le a_n$
Set $A = a_1$
$\mathcal{E} = A + \int_A^B (1 - \frac{t-a_1}{b_1-a_1})dt$; $P(t) = 1$
For each $i = 2, \ldots, n-1$
    $P(t) = P(t)\frac{b_{i-1}-t}{b_{i-1}-a_{i-1}}$
    $\mathcal{E} = \mathcal{E} - \int_{a_i}^B \frac{t-a_i}{b_i-a_i} P(t) dt$ (via symbolic integration)
Output $\mathcal{E}_{min} = \mathcal{E}$

Fig. 6. An $O(n^2)$ exact algorithm for expected minimum.

[10] We are not the first to notice this error, e.g., [11] cite Donath [6] as the source for simple correction factors in the cases of $N = 1$ or 2 regions. Of course, correlations of pin locations combined with small region aspect ratios can make the center-to-center bounding box approximation an overestimate (rather than an underestimate), but $\frac{1}{3}$ of the bounding box size is indeed the correct result when only the uniform distribution (and no specific correlation) is known.

Theorem III.3: The Expected Minimum Algorithm (Figure 6) finds the expected minimum of n points uniformly distributed in segments $[a_i, b_i]$ in time $O(n^2)$.

Theorem III.2 shows that the expectation of the minimum is computed starting from the expectation of one point, with a series of apparently geometrically decreasing negative corrections. This motivates the question of designing linear or near-linear time heuristics: *if we allow for small decreasing errors to the above corrections, the cumulative error will be small.*

B. Fast Estimation of the Expected Minimum

We now present two heuristics for finding the expected minimum which are significantly faster than the exact algorithm, not only asymptotically, but also for small values of n.

The linear-time heuristic starts with the segment for the first random point and gradually shifts both endpoints of this segment to the left as it goes sequentially through the list of all segments. The midpoint of the resulting segment gives an approximation of the expected minimum (see Figure 7).

The Fast Expected Minimum Heuristic

Input: Segments $[a_i, b_i], i = 1 \ldots N$, each with one random point
Output: Approximate expected location of the leftmost point

1. $A = a_1$, $B = b_1$
2. **For each** of the $n-1$ remaining segments $[a_i, b_i]$ **do**
   if $a_i < A$, then swap the two segments $[A, B]$ and $[a_i, b_i]$
   if $a_i < B$
   then
      if $b_i \geq B$
      then $B = B - \frac{(B-a_i)^3}{3(b_i-a_i)(B-A)}$
      else $B = B - \frac{\frac{1}{3}(b_i-a_i)^2+(B-a_i)(B-b_i)}{(B-A)}$
3. **Output** $\mathcal{E}_{min} = \frac{A+B}{2}$

Fig. 7. A linear-time heuristic for expected minimum.

The second heuristic is more accurate but slower, with $O(n \log n)$ runtime. It sorts all segments in decreasing order of their left endpoints and finds the leftmost right endpoint $M = \min_i b_i$. The Fast Expected Minimum Heuristic is then applied to segments whose left endpoints are not greater than M (see Figure 8).

Step 2 of the Fast Expected Minimum Heuristic is based on the following

Proposition III.4: If $[a_1, b_1]$ and $[a_2, b_2]$ $(a_1 \leq a_2)$ are two segments each containing one random point, then the expected minimum is equal to

$$\frac{a_1 + b_1}{2} - \frac{(b_1 - a_2)^3}{6(b_1 - a_1)(b_2 - a_2)} \text{ if } \mathrm{b}_1 \leq \mathrm{b}_2$$

The Expected Minimum Heuristic
Input: Segments $[a_i, b_i], i = 1 \ldots N$ each with one random point **Output:** Approximate expected location of the leftmost point
1. Sort segments by left endpoints, such that $a_1 \geq \ldots \geq a_n$ 2. Find the leftmost right endpoint $M = \min_i b_i$ 3. Omit all segments with $a_i > M$ 4. Apply Fast Expected Minimum Heuristic to remaining segments

Fig. 8. A more accurate, $O(n \log n)$ expected minimum heuristic.

and

$$\frac{a_1 + b_1}{2} - \frac{\frac{1}{3}(b_2 - a_2)^2 - (b_2 - a_2)(b_1 - a_2) + (b_1 - a_2)^2}{2(b_1 - a_1)} \text{ if } \mathrm{b}_1 > \mathrm{b}_2$$

Proof. According to Theorem III.2

$$\begin{aligned} \mathcal{E}_{min} &= a_1 + \int_{a_1}^{b_1} (1 - \frac{t - a_1}{b_1 - a_1})\, dt - \int_{a_2}^{b_1} \frac{t - a_2}{b_2 - a_2}(1 - \frac{t - a_1}{b_1 - a_1})\, dt \\ &= a_1 + (b_1 - \frac{a_1 + b_1}{2}) - \frac{(b_1 - a_2)^3}{6(b_1 - a_1)(b_2 - a_2)} \end{aligned}$$

which completes the case for $b_1 \leq b_2$. The other case is shown using a similar computation. □

We replace a pair of segments with a new segment such that its middle approximates the expected minimum of random points in the two original segments. This can be viewed as approximating the cumulative distribution of the minimum (over the union of original segments) with a linear cumulative distribution (over the new segment). The approximation error of one such step is the difference between the original expected minimum and the middle of the new segment ("new expected minimum"). However, when the step is applied many times, additional error is incurred by our removing higher moments of the expected minimum.

To show that the sorting step (Step 1) in the Expected Minimum Heuristic improves accuracy, consider segments $[a_1, b_1] = [0, 1]$ and $[a_i, b_i] = [\frac{1}{2}, \frac{1}{2}]$, $i = 2, \ldots, n$. For this input, the Fast Expected Minimum Heuristic correctly determines the expected minimum for the random points in the first two segments as $\frac{3}{8}$, but the right endpoint of the resulting segment is placed at $\frac{3}{4}$. All other segments $[a_i, b_i]$, $i = 3, \ldots, n$, further shift the right endpoint to the left. For sufficiently large n, the right endpoint will be at $\frac{1}{2}$ and the approximate expected minimum will be at $\frac{1}{4}$. On the other hand, the exact expected minimum is still $\frac{3}{8}$.

To assess the relative error of the Fast Expected Minimum Heuristic, we compute both the approximate and the exact expected *maximum* values as well. Then evaluate the relative error between the heuristic's approximated expected distance between maximum and minimum values, and the exact expected distance between maximum and minimum values. In the above example, the relative error of the Fast Expected Minimum Heuristic is 100% since the heuristic's approximated expected distance is $\frac{3}{4} - \frac{1}{4} = \frac{1}{2}$, while the exact expected distance is $\frac{5}{8} - \frac{3}{8} = \frac{1}{4}$); we believe that this is the worst case. Our Monte-Carlo experiments indicate that the average relative error of the Fast Expected Minimum Heuristic for random input[11] is about 1.1%.

On the other hand, for the input described above, the Expected Minimum Heuristic finds the exact expected minimum. Monte-Carlo experiments also confirm the benefit of the sorting step: for 10000 random inputs for each value of $n = 3, \ldots, 30$ the expected relative error of the Expected Minimum Heuristic was always less than 0.6%, and we never encountered any instance with relative error greater than 5%. Based on the symmetry of the problem, we also believe that the maximum relative error of the Expected Minimum Heuristic occurs in the case when all segments are the same. An error bound for this case is given by the following

Fact 5: If all segments are identical, the maximum possible error of the Expected Minimum Heuristic is $\approx 5.15\%$.

Proof. Without loss of generality, assume that $A = 0$ and $B = 1$. Then by Fact 4 the expected minimum is equal to $\frac{1}{n+1}$. Each time we process another segment we shift B by $\frac{B^2}{6}$. Therefore, the sequence of right endpoints is described by the recurrence

$$B_i = B_{i-1} - \frac{(B_{i-1})^2}{3}.$$

The left endpoints of segments to the right of the midpoint M, which we use to find the expected maximum, are shifting symmetrically. Thus, the relative error equals

$$\frac{B_n - \frac{2}{n+1}}{1 - \frac{2}{n+1}}$$

A numerical computation shows that $B_{100} \approx 0.028$. Because B_n is decreasing, the relative error cannot be more than $B_{100}/0.98 \approx 0.029$ for $N > 100$. Numerical evaluation shows that the maximum relative error over the values $n = 1, \ldots, 100$ is $\approx 5.15\%$ and occurs for an instance with $n = 7$ segments.

□

This discussion suggests that the Expected Minimum Heuristic has small worst-case error. We leave determining the exact performance ratio as an open problem.[12]

[11] We generated n ($n = 3, \ldots, 30$) random segments in the $(0, 1)$-interval and found both the exact and the approximate expected distance between the maximum and minimum. We ran 10000 experiments for each n and found that the average relative error was largest (i.e., about 1.2%) for $n = 12$ while the maximum relative error was always less than 10%.

[12] As a side note: we observe that a model with discrete distributions can be applicable, particularly with standard-cell layouts wherein cells occupy fixed sites in rows. Sites can be distributed unevenly, e.g., due to obstacles or to rows of varying capacities being located unevenly. If locations have a discrete distribution, Formula 1 applies and the integral can be directly computed as a sum over allowed sites or row coordinates. The runtime needed to handle

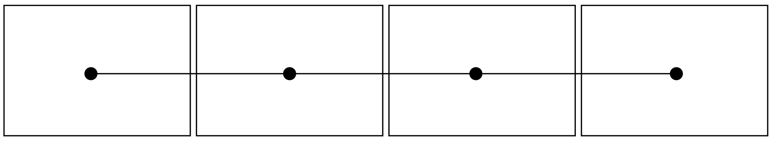

Fig. 9. A 4-pin net with pins distributed in a region with aspect ratio 4 which is a union of 4 blocks each having aspect ratio 1.

IV. Expected RStMT Cost for n Random Points Distributed in a Plane Region

A literature on growth rates of subadditive functionals of pointsets, originating with Beardwood et al. [1] and continuing through works of Steele and Snyder [20] [21], establishes bounds on the expected RStMT cost, $E[c(RStMT)]$, for n points uniformly distributed within a region R. Specifically, we know that $E[c(RStMT)] \propto \sqrt{area(R) \cdot n}$ for n sufficiently large. The constant of proportionality β does not depend on the shape of the region.[13] Empirical evidence suggests that the expected value of the ratio $\frac{c(RStMT)}{\sqrt{area(R) \cdot n}}$ converges to the constant $\beta \approx 0.76$. If R is a rectangle, which is often appropriate in layout applications, then $area(R) = w_R \cdot h_R$ and $E[c(RStMT)] \propto \sqrt{w_R \cdot h_R \cdot n}$.

Our work in this section is motivated by an apparent contradiction. If the expected RStMT cost is proportional to the square root of the area $w_R \cdot h_R$ of a given region, why are all practical estimates based on the half-perimeter $w_R + h_R$ of the region? Put another way, if the theory suggests use of a *geometric mean* estimate, why have practitioners always used an *arithmetic mean* estimate?[14] In this section, we attempt to resolve this puzzle. We show that there is very substantial deviation from the $\sqrt{w_R \cdot h_R \cdot n}$ expected RStMT cost when the pointset is small and/or when the region R is non-square (i.e., has aspect ratio > 1). As it happens, these are precisely the conditions of interest for VLSI layout applications. (Note that small aspect ratios (close to 1 and almost never bigger than 2) of layout blocks in top-down placement do not contradict the relevance of considering large aspect ratio regions. This is because a single net can often span multiple blocks during top-down placement (see Figure 9), yielding an overall region of possibly large aspect ratio.)

The results of this section, together with those of the previous section and the uniform distribution based methodology outlined in Section 2, allow us to develop new and highly accurate RStMT cost

discrete distributions depends linearly on the number of pins and on the number of sites. When the complexity is sufficiently small (e.g., if the block has 15 rows) or when the sites are distributed very nonuniformly, such an approach may give improved wirelength estimates in comparable or shorter time.

[13] The argument is simple. Tile the region with uniform small squares. Apply the known result for $R =$ the unit square to each small square, then join the "trees" in each small square together. The cost of joining is asymptotically negligible.

[14] The half-perimeter of the region is twice the arithmetic mean $\frac{w_R + h_R}{2}$. We know that many estimates scale as square-root of the number of pins, whose average is close to 3 for nets in VLSI circuits.

estimators in Section V. We first show that the convergence of the ratio $\frac{E[c(RStMT)]}{\sqrt{area(R)\cdot n}}$ to β strongly depends on the shape of the region R even though the value of β is asymptotically independent of the shape. We confine our discussion to rectangular regions, which allows the shape of the region R to be expressed as an *aspect ratio* $AR = \frac{w(R)}{h(R)}$, where we assume without loss of generality that $w(R) > h(R)$.

Theorem IV.1: For any fixed $n > 1$, let P be an n-pointset in a rectangular region R, and let $mindist(P)$ and $maxdist(P)$ be the the minimum and maximum distance between x-projections of points from P, respectively. If $mindist(P) \neq 0$, then $c(RStMT(P)) = \Theta(AR(R) \cdot \frac{maxdist(P)}{w(R)})$, where $AR(R) = \frac{w(R)}{h(R)}$ is the aspect ratio of R.

Proof. We show that scaling both the width of a rectangular region $R = R_0$ and all x-coordinates of a given n-pointset $P = P_0$ in R_0 by a sufficiently large factor will make the RStMT cost proportional to $AR(R) = \frac{w(R)}{h(R)}$. Multiply all x-coordinates in R by $\frac{h(R_0)}{mindist(P_0)}$. and denote the resulting region as R_1.

We will show that the RStMT over the scaled n-pointset P_1 intersects any vertical line at most once if the x-coordinate of the line does not coincide with the x-coordinate of any point in P_1. Indeed, suppose some vertical line intersects the RStMT at least twice (see Figure 10). We can then replace the horizontal edge e, which has length at least $\frac{h(R_0)}{mindist(R_0)} \cdot mindist(R_0) = h(R_0)$, by a shorter vertical edge (either $e1$ or $e2$) which has length at most $h(R_0)$ because the height of the region is $h(R_1) = h(R_0)$.

Since x-projections of the edges of the RStMT over P_1 do not overlap, the total cost of the RStMT is the sum of vertical edge costs and $maxdist(P_1)$. Further stretching of the region R by a factor $t > 1$ yields a region R_t and pointset P_t. The total cost of horizontal edges in the RStMT increases to $t \cdot maxdist(P_1)$ while the total cost of vertical edges is the same. Since the length of any vertical segment is at most $h(R_0)$, the total length of vertical segments is at most $n \cdot h(R_0)$. Recall that both n and $h(R_0)$ are fixed with respect to $AR(R)$, i.e., $n \cdot h(R_0) = \Theta(1)$. Then, the RStMT cost over P_t is

$$\begin{aligned} c(RStMT) &= n \cdot h(R_0) + t \cdot maxdist(P_1) \\ &= \Theta(1) + t \cdot maxdist(P_1) \\ &= \Theta\left(t \cdot maxdist(P_0) \cdot \frac{h(R_0)}{mindist(P_0)}\right) \\ &= \Theta\left(t \cdot \frac{maxdist(P_0)}{mindist(P_0)}\right) \end{aligned}$$

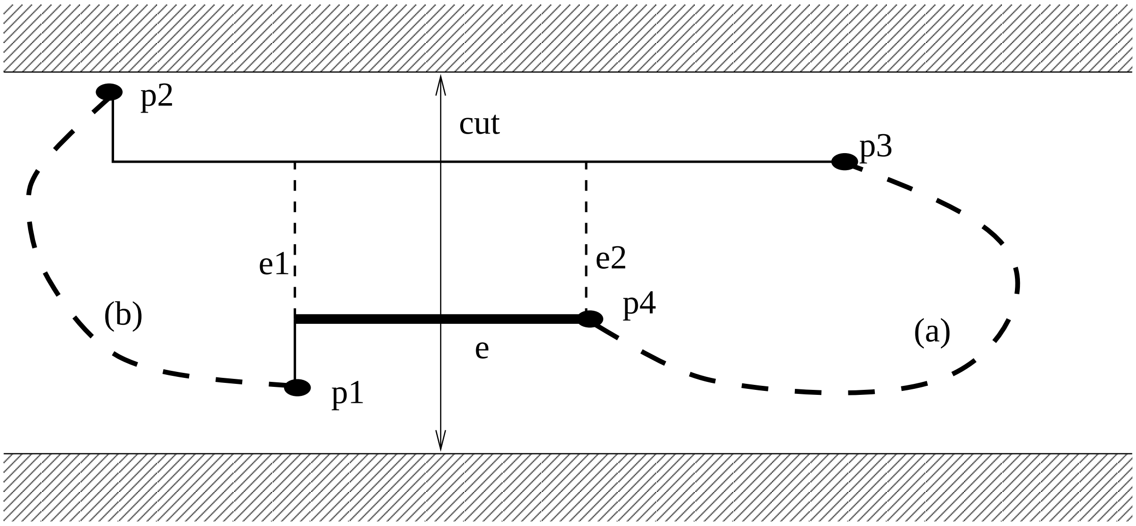

Fig. 10. Switching overlapping horizontal edges. If a vertical cut intersects two horizontal edges $(p1, p4)$ and $(p2, p3)$ of the RStMT over pointset S', we can replace the horizontal segment e (shown in dark) with a shorter vertical segment. In the RStMT, there must be a path connecting the edges $(p1, p4)$ and $(p2, p3)$. (a) If this path connects to $p4$, then e can be replaced by $e1$. (b) Otherwise, this path connects to $p1$ and e is replaced by $e2$.

On the other hand, the aspect ratio of R_t is

$$\begin{aligned} AR &= t \cdot AR(R_1) \\ &= t \cdot AR(R_0) \cdot \frac{h(R_0)}{mindist(R_0)} \\ &= \Theta\left(\frac{t \cdot w(R_0)}{mindist(P_0)}\right) \end{aligned}$$

From the previous two equalities,

$$c(RStMT) = \Theta\left(AR(R) \cdot \frac{maxdist(P_0)}{w(R_0)}\right)$$

□

For any $n > 1$, the expected minimum distance between the two closest among n points uniformly distributed within a non-zero segment is nonzero. [15] Also, the expected maximum distance is proportional to the length of the region R_0, by Corollary III.1. Thus,

Observation 1: For any $n > 1$, $E[c(RStMT)] = \Theta\left(\frac{w(R)}{h(R)}\right)$ for n points uniformly distributed within a rectangular region R.

[15] In fact, this expected minimum distance is equal to $\frac{1}{n^2}$ for n points taken from a uniform distribution in the unit interval. The proof of this folklore fact (which is an exercise in textbooks) requires about a page of discussion, and is not important to our development (all we need is that *mindist* is nonzero). We could certainly add the justification if required.

Notice that Observation 1 does not contradict the result from [1]. The result from [1] holds "for n sufficiently large", and can be formally restated as: For any region R, there exists an N_0 such that for any $n > N_0$ points uniformly distributed within R, $E[c(RStMT)]$ is proportional to $\sqrt{area(R) \cdot n}$. Observation 1 says that for any fixed n, there is a region with sufficiently large aspect ratio for which this proportionality (i.e., to $\sqrt{area(R)}$) is invalid. In other words, there is no single N_0 such that for *any* region R, $E[c(RStMT)] \propto \sqrt{area(R) \cdot n}$ for $n > N_0$. The value of N_0 *depends* on the shape of the region R, e.g., if R is a rectangle then larger aspect ratio of R requires larger N_0. We experimentally validate the theorem, as well as the original result from the literature, by the following experiment. We first generate $N = 10000$ random instances of n points, for $n = 4, 5, \ldots, 30$ (note that $n = 2, 3$ are not interesting), chosen from a uniform distribution in the rectangular region $[0, 1] \times [0, AR]$, for values of aspect ratio $AR = 1, 2, 4, 8, ..., 512$. We then find the cost of a heuristic RStMT over the generated n points using the Batched Iterated 1-Steiner implementation of Griffith et al. [9], and divide this cost by $\sqrt{area \cdot n}$.[16] Table I presents the resulting values $\beta(n, AR)$, which we know should converge to $\beta \approx 0.76$ by the theory of Beardwood, Steele et al. The plot of Figure 11 presents a portion of the Table I data in an alternate way; we give individual curves depicting the convergence of $\beta(n, AR)$ for different values of the aspect ratio AR. Notice that the convergence is slower for larger values of AR, and that the deviation of $\beta(n, AR)$ from β is larger when n is small. The wide separation of curves for small n and the slow convergence for large AR empirically confirm that the proportionality $E[c(RStMT)] \propto \sqrt{area(R) \cdot n}$ holds only for $n > N_0$, where N_0 grows with the aspect ratio of region R. Therefore, for sufficiently large aspect ratio of region R and sufficiently small n, $E[c(RStMT)]$ is not proportional to $\sqrt{area(R) \cdot n}$, i.e., there will be significant deviation of $E[c(RStMT)]$ from the expected $\approx 0.758\sqrt{area(R) \cdot n}$. The use in practice of half-perimeter based RStMT estimators is empirically justified in discussing Tables I and III in the following section.

V. Expected RStMT Cost of n Random Points Distributed in a Specified Bounding Box

The results of the previous section afford *region-based* estimates of $c(RStMT)$ for pointsets drawn from a given (rectangular) region. Such estimates can be very rough. In practice, e.g., for *a posteriori* estimation after placement, we have much more information than just a region from which points are drawn: we have the actual bounding box of the RStMT instance. Intuitively, we should be able to

[16] In what follows, we will always use this Batched Iterated 1-Steiner implementation to approximate the (NP-hard) RStMT solution. Results in [9] indicate that this will overestimate the true RStMT cost by an average of less than a quarter percent for the instance sizes that we discuss.

$c(RStMT)/\sqrt{n \cdot area}$									
	Aspect Ratio (AR)								
n	1	2	4	8	16	32	64	128	256
4	0.64	0.67	0.78	0.98	1.29	1.76	2.44	3.44	4.82
5	0.67	0.70	0.80	0.99	1.30	1.76	2.43	3.39	4.76
6	0.69	0.72	0.81	0.99	1.27	1.73	2.41	3.36	4.68
7	0.71	0.73	0.81	0.98	1.26	1.69	2.33	3.25	4.56
8	0.72	0.74	0.82	0.97	1.24	1.66	2.28	3.16	4.44
9	0.73	0.75	0.81	0.96	1.21	1.62	2.21	3.07	4.33
10	0.74	0.75	0.81	0.95	1.19	1.57	2.15	2.99	4.18
15	0.75	0.76	0.80	0.90	1.10	1.42	1.91	2.62	3.67
20	0.76	0.77	0.80	0.87	1.03	1.30	1.73	2.37	3.29
30	0.76	0.76	0.79	0.84	0.95	1.16	1.51	2.03	2.81

TABLE I

AVERAGE VALUES OF $\frac{cost(RStMT)}{\sqrt{n \cdot area}}$ OVER 10000 RANDOM n-POINT SAMPLES IN A RECTANGULAR REGION WITH ASPECT RATIO AR.

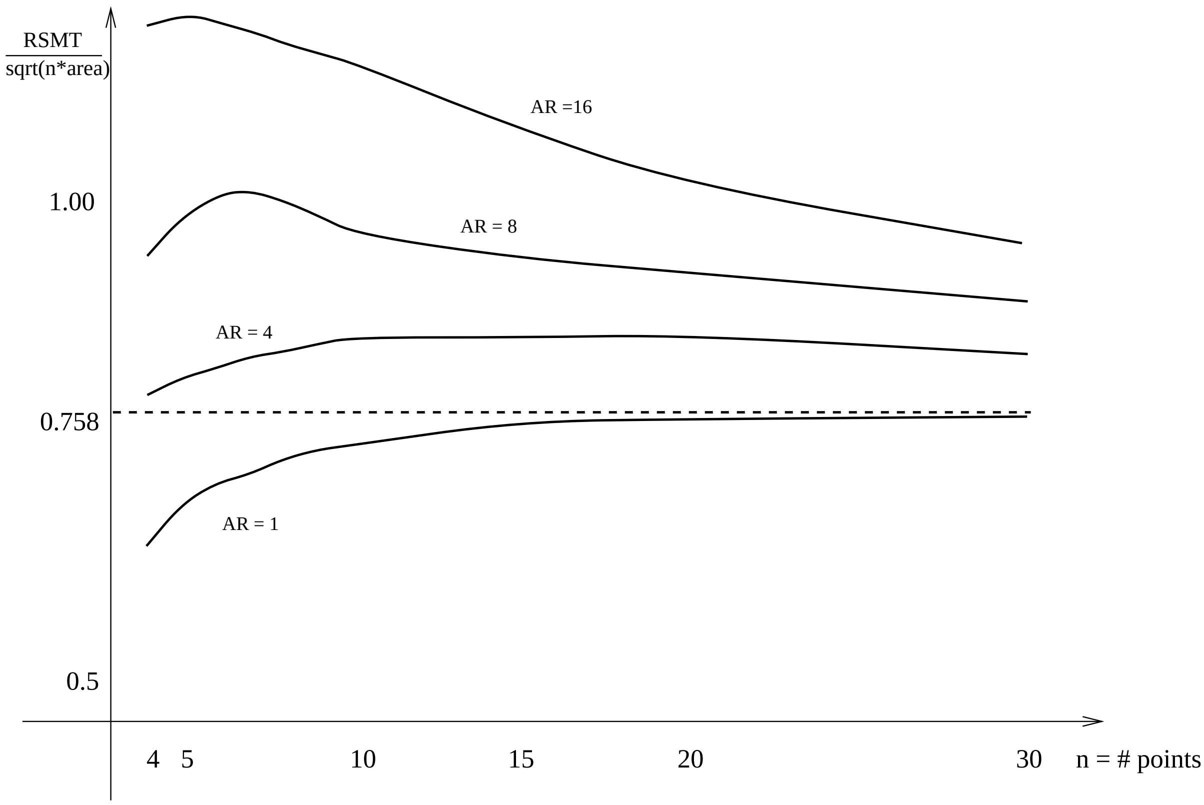

Fig. 11. Plots of $\beta(n) = E[c(RStMT)]/\sqrt{area(R) \cdot n}$ for different aspect ratios AR of the rectangular region.

exploit this extra information to achieve more accurate *bounding box-based* estimates of $c(RStMT)$. The first subsection shows that using the bounding box information allows more accurate estimates. In the second subsection, we demonstrate that for any $n > 1$, the RStMT cost is proportional to the

Deviations from Average of RStMT Cost for n Points Uniformly Distributed in Unit Square								
	#points (n)							
	4	5	6	8	10	15	20	30
RStMT	1.28	1.50	1.69	2.04	2.33	2.91	3.38	4.15
90%	40.6	34.8	31.0	25.1	20.6	16.1	13.0	10.0
95%	48.1	40.8	36.8	29.7	24.3	18.9	15.4	12.0
98%	57.2	48.0	43.2	34.8	29.3	22.6	18.4	14.5

TABLE II

AVERAGE, AND DEVIATION FROM AVERAGE, OF RSTMT COSTS OVER 10000 RANDOM n-POINT SAMPLES IN A UNIT SQUARE. MAXIMUM RELATIVE DEVIATION FROM AVERAGE (EXPRESSED AS A PERCENTAGE) IS COMPUTED FOR "BEST" 90%, 95% AND 98% OF SAMPLES.

Deviations from Average of (RStMT Cost / BBox HP) for n Points Uniformly Distributed in Unit Square								
	#points (n)							
	4	5	6	8	10	15	20	30
NRStMT	1.06	1.13	1.19	1.31	1.42	1.66	1.87	2.22
90%	10.5	11.8	14.2	14.4	13.4	11.6	10.2	8.5
95%	14.5	15.2	15.8	16.7	15.6	13.8	12.1	10.1
98%	20.0	18.3	18.7	18.9	18.6	16.4	14.4	12.2

TABLE III

AVERAGE, AND DEVIATION FROM AVERAGE, OF THE QUANTITY (RSTMT COST DIVIDED BY HALF-PERIMETER OF THE POINTSET BOUNDING BOX). THE 10000 n-POINT SAMPLES ARE TAKEN FROM A UNIFORM DISTRIBUTION IN THE UNIT SQUARE. MAXIMUM RELATIVE DEVIATION (EXPRESSED AS A PERCENTAGE) IS COMPUTED FOR "BEST" 90%, 95% AND 98% OF THE SAMPLES, RESPECTIVELY.

aspect ratio of the bounding box (this is not surprising given Theorem IV.1). The result is a new table for RStMT cost lookup based on *bounding box* aspect ratio and number of points n.

A. Bounding Box Information Helps

We begin with an empirical demonstration of the gain from knowing the bounding box. We generate $N = 10000$ random instances of n points ($n = 4, 5, \ldots, 30$) uniformly distributed in the unit (1×1) square. The first row of Table II shows average RStMT costs over the N samples for each value of n. (These correspond the first column of Table I, scaled by factors of $\sqrt{n}$.) The Table also shows the maximum relative deviation from this average (expressed as a percentage) among the 90%, 95% and 98% of the instances.[17]

[17]The exact calculation is as follows. For each of 10000 samples we find the relative deviation of $c(RStMT)$ from the average cost of RStMT for a given n. Then, we rank all relative deviations for each given value of n. To find, say, the maximum relative deviation over the 90% of samples, we determine the $90th$ percentile in the rank order. For example, we see from the table that the middle 90% of all 10000 7-point instances have $c(RStMT)$ within 27.003% of the average value of $c(RStMT)$, which is 1.8748.

Deviations from Average of (RMSpT Cost / RStMT Cost) for n Points Uniformly Distributed in Unit Square								
	#points (n)							
	4	5	6	8	10	15	20	30
MSpT	1.10	1.11	1.11	1.11	1.12	1.12	1.12	1.12
90%	9.4	9.3	8.2	6.7	6.2	4.8	4.2	3.3
95%	12.5	10.5	9.6	7.9	7.3	5.8	5.0	3.9
98%	16.5	13.2	11.8	9.3	8.7	7.0	5.8	4.7

TABLE IV

Average, and deviation from average, of the quantity (RMSpT cost divided by RStMT cost). The 10000 n-point samples are taken from a uniform distribution in the unit square. Maximum relative deviation (expressed as a percentage) is computed for "best" 90%, 95% and 98% of the samples.

Similarly, in Table III, the first row gives averages over $N = 10000$ samples of the *ratio* of the RStMT cost *divided by the half-perimeter of the pointset's bounding box.* The first row of Table III shows that when the number of points is 4 or 5, the error of the half-perimeter bounding box estimator is 6% or 13%, respectively. This is much smaller than the error of the area-based estimator (see Table I). Of course, when the number of points is 2 or 3, then the bounding box half-perimeter *exactly* gives the RStMT cost. Each column again gives maximum relative deviations in the middle 90%, 95% and 98% of the data, expressed as percentages. We see that normalizing to the bounding box half-perimeter yields a greatly improved estimate (between two and three times more accurate for $n = 4, 5, 6$).

We make a small digression to indicate how far off these estimates are from "best possible" non-constructive estimates, namely, those based on the rectilinear MSpT construction. Recall that RMSpT cost is known to average around 12% greater than RStMT cost (cf. analyses of Bern and de Carvalho, as reviewed in [14]). Table IV gives the average ratio of RMSpT cost over RStMT cost, for the same 10000 random instances for each value of n. This allows us to compare the bounding box estimator with the minimum spanning tree estimator. We conclude that the bounding box-based estimator is much better than the region-based estimator, and that the MSpT-based estimator is somewhat better still. However, as we noted earlier, the MSpT is too expensive to be used in practice – we require linear-time on-line wirelength estimators with sublinear update costs (based on reasonable storage) in the iterative placement context.

B. RStMT Cost Dependence on Bounding Box Aspect Ratio

From the results of Section IV we know there is a dependence of RStMT cost on the aspect ratio of the *region* from which points are chosen. When we also consider the previous results of this section, we are motivated to seek a dependency of RStMT cost on the aspect ratio of the *pointset bounding box* as well. Intuitively, if we can estimate this dependency, we will be able to more accurately predict the RStMT cost for uniformly distributed points with a known bounding box. The difficulty is that the $\beta(n, AR)$ values from Section IV cannot be directly applied when we have a specific pointset with a specific bounding box. We resolve this difficulty as follows.

From Fact 4, we know that if we choose n points uniformly distributed within a rectangular region with sides w and h, the expected sides of the bounding box of these points are $w' = w(1 - \frac{2}{n+1})$ and $h' = h(1 - \frac{2}{n+1})$. In other words, the expected aspect ratio of the bounding box is the same as the aspect ratio of the region in which the points were uniformly distributed.

Therefore, if we are given n uniformly distributed points having a known *bounding box* with sides w_{bb} and h_{bb}, we may predict that the expected RStMT cost is the same as the expected RStMT cost of n points uniformly distributed within the *region* having sides $w_R = \frac{n+1}{n-1} \cdot w_{bb}$ and $h_R = \frac{n+1}{n-1} \cdot h_{bb}$.

We have confirmed that wirelength predictors based on the approach of computing $w_R = \frac{n+1}{n-1} \cdot w_{bb}$, $h_R = \frac{n+1}{n-1} \cdot h_{bb}$ differ from empirically constructed (i.e., via Monte Carlo experiments) predictors by less than 1%. In the empirical approach (which is analogous to the construction of Table I, we generate a set of n points uniformly distributed within a bounding box of prescribed width w and height h in the following way:

1. generate a set of n points uniformly distributed within the 1×1 square;
2. find the bounding box of this set of points, with dimensions w' and h'; and
3. multiply all x- and y-coordinates of the points by w/w' and h/h', respectively.

We use this construction in finding RStMT costs over $N = 10000$ samples of n random points with bounding box aspect ratio $AR = 1, 2, ..., 32$, and dividing by the half-perimeter of the bounding box. Table V shows the average values for this ratio of RStMT cost to bounding box half-perimeter, as a function of n and AR. A similar empirical analysis was performed by Cheng [2], but the corresponding coefficients did not depend on aspect ratio. Additionally, the author of [2] used a worse code for finding heuristic RStMTs. As a result, the entries of the first row of Table V, which we reproduce from [2], are larger than our entries by approximately 2% even in the case of $AR = 1$. In practice, Cheng's method will produce even worse overestimates of RStMT cost because it ignores the effect of bounding box aspect ratio.[18]

[18]Readers may notice that the entries for $AR = 1$ in Table V are slightly different from the entries in the first row of Table III. This is because the Table III entries are averaging over all bounding box aspect ratios, not just $AR = 1$. The expected bounding box aspect ratio for uniformly distributed points is somewhere between 1 and 2, and this is consistent with the data in the two tables.

Average RStMT Cost for Pointsets With BBox Half-Perimeter = 1							
	#points (n)						
AR	4	5	6	8	10	15	20	30
1*	1.08	1.15	1.22	1.34	1.45	1.69	1.89	2.23
D	12.2	13.4	15.5	15.2	14.4	12.7	11.4	9.52
1	1.06	1.13	1.19	1.32	1.42	1.66	1.87	2.22
D	11.2	12.5	14.5	14.1	13.5	11.6	10.3	8.64
2	1.05	1.11	1.16	1.27	1.36	1.59	1.78	2.10
D	9.83	10.3	12.3	12.6	12.6	11.2	10.1	8.64
4	1.03	1.07	1.11	1.18	1.25	1.41	1.57	1.84
D	6.87	6.93	8.50	9.54	9.98	9.87	9.35	8.13
10	1.01	1.03	1.05	1.08	1.12	1.21	1.29	1.45
D	3.37	3.44	4.39	5.05	5.53	6.18	6.51	6.53

TABLE V

Each entry represents an average, over 10000 samples of n random points having prescribed bounding box aspect ratio, of RStMT cost divided by bounding box half-perimeter. The first row reproduces coefficients from the paper by Cheng. Each row marked with D gives the maximum relative deviation from the average in 90% of the samples, expressed as a percentage.

VI. Practical On-Line Wirelength Estimation

To see the practical value of our new estimators, we first observe that they are extremely efficient.

- In the on-line (top-down placement) context, we have n pins distributed among various regions. Instead of returning the bounding box of the region centers, we apply the linear-time heuristic of Figure 7 in Section III to obtain the expected bounding box of the pins, then perform lookup (with linear interpolation as appropriate) in Table V of Section V.
- In the *a posteriori* context, we have n pins in exact locations. Instead of returning any of the previous bounding box based estimates, we perform lookup (again with linear interpolation as appropriate) in Table V.

The time complexity of our estimates in each context is $O(n)$. Updating the estimate when a pin is moved, as long as we have exact locations and do not need to execute the heuristic of Figure 7, requires only a constant number of operations after the net bounding box has been *updated*. Thus, speedups of bounding box *updates* that are used in practice (see, e.g., TimberWolf-related papers of Sechen et al) hence transfer directly into our methods. Having established efficiency, we next show that our methods lead to improved estimation accuracy in the on-line context.

We have incorporated our new estimates, along with the previous center-based, bounding box-based, and Cheng [2] estimates, into an internal placement testbed that includes both top-down partitioning and annealing engines.[19] Specifically, we compare the following seven wirelength esti-

[19]Placements that we obtain from our internal testbed are competitive – in terms of runtime, various solution metrics, and routability by industrial routers – with placements from industry placers that we are aware of.

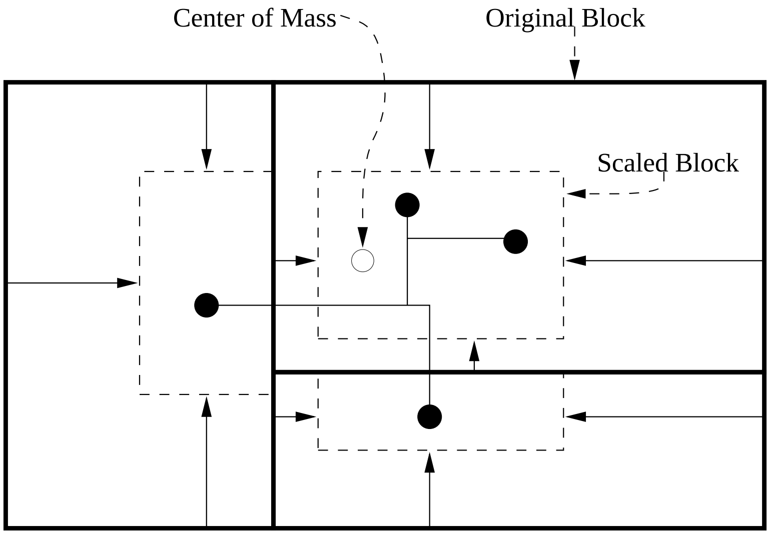

Fig. 12. Scaling of regions for a net prior to calculating expected bounding box for HBBN heuristic.

mates (the first three algorithms estimate half-perimeters of net bounding boxes and the remaining four algorithms estimate Steiner tree lengths taking in account the net sizes and the aspect ratios of bounding boxes):

- CBB : Standard bounding box estimate using the center coordinates of the region in which the given pin is located
- HBB : Heuristic bounding box estimate using the linear-time heuristic of Figure 7
- HBBN : Heuristic HBB

– When estimating the wirelength for a given net, each region the net intersects is scaled towards the net's center of mass (see Figure 12). The HBB heuristic is then applied to these scaled regions (see also the discussion below).

– For nets with all pins in the same partitioning region R, the region R is scaled by ratio $\frac{2.4}{|R|^{0.38}}$, where $|R|$ is the number of cells in R.

– For nets with pins in different partitioning regions $R_1, \ldots, R_p$, the regions $|R_i|$ are scaled towards the center of the net's bounding-box by ratio $\frac{1}{|R_i|^{0.1}}$, where $|R_i|$ is the number of cells in R_i.

- Cheng : CBB estimate, scaled by the coefficients of Cheng [2] (reproduced in the first line of Table V)
- CBBtab : CBB estimate, followed by lookup in Table V
- HBBtab : HBB estimate, followed by lookup in Table V
- HBBNtab : HBBN estimate, followed by lookup in Table V

Test Case	Number of Cells	Number of Nets	Placement Runtime
Test1	1756	1492	47.50
Test2	3286	2902	74.42
Test3	6692	6527	1274.03
Test4	12133	11828	526.20
Test5	12857	10880	446.74

TABLE VI

Parameters of five standard-cell test cases from industry, along with total runtime of the top-down partitioning based placement process for each test case (reported in CPU seconds for a 300MHz Sun Ultra-10 workstation with 128MB RAM).

As discussed in Section 2, the uniform distribution assumption tends to overestimate wirelengths during the early stages of the top-down placement process. This is because cells that are connected by a net will have correlated locations. We propose to correct for the correlation of pin locations from the same net, by shrinking partitioning regions when we estimate the net's wirelength. The scaling should be according to the size of the partitioning regions, which can be measured by the number of cells inside (there is more room for cells to "float" when the regions are large, and less room when the regions are small). Following our empirical studies (and as in [4]), our heuristic HBBN distinguishes nets which are distributed among several regions and nets which are completely contained in a single region. We empirically studied the top-down placements of a number of designs, recording wirelengths of placed nets and the nets' cut or uncut status at various levels of the placement. We then fitted, for both cut and uncut nets, the coefficients α and β for scaling of type $\alpha \cdot |R|^{\beta}$. It turns out, as expected, that the shrinking correction is larger for nets that are completely contained in a single region.

Our experiments thus far have evaluated the accuracy of on-line wirelength estimation in a top-down partitioning-based placer. We have used five standard-cell test cases Test1,...,Test5, obtained from industry; their parameters are given in Table VI.

Our results are given in Table VII. For each test case, we run the top-down partitioning based placer to completion, then measure both total net bounding box half-perimeter and total Iterated 1-Steiner heuristic RStMT cost of the result. For the bounding box estimators (the first four rows of each table), each table entry gives the relative error of the estimated sum of bounding box half-perimeters after the first $i \cdot 10\%$ ($i = 1, \ldots, 10$) levels of the top-down partitioning based placement, versus the final sum of bounding boxes. (We report data for every 10% of the levels because the number of placement levels varies according to instance size.) For the wirelength estimators (the last five lines of each table), each table entry gives the relative error of the estimated sum of Steiner tree

Average Relative Error of Wire Length Estimates										
	% of levels completed									
Estimator	10	20	30	40	50	60	70	80	90	100
CBB	.243	.185	.138	.110	.086	.059	.031	.011	.001	.000
HBB	2.36	1.59	.911	.511	.298	.169	.071	.029	.002	.000
HBBN	.112	.067	.034	.028	.015	.011	.025	.034	.026	.000
Cheng	.208	.139	.085	.051	.038	.027	.039	.057	.065	.065
CBBtab	.259	.188	.132	.104	.080	.057	.032	.016	.010	.010
HBBtab	2.37	1.61	.920	.509	.293	.164	.066	.025	.010	.010
HBBNtab	.098	.052	.032	.018	.011	.017	.021	.024	.010	.010

TABLE VII

Relative errors of estimated sums of bounding box half-perimeters and RStMT costs during the top-down placement. Data for 5 testcases are normalized and averaged.

Average CPU Time (seconds) of Wire Length Estimates							
	Estimator						
test case	CBB	HBB	HBBN	Cheng	CBBtab	HBBtab	HBBNtab
test1	.010	.010	.019	.010	.010	.010	.019
test2	.010	.019	.030	.010	.010	.020	.030
test3	.010	.050	.089	.010	.010	.050	.100
test4	.020	.060	.120	.020	.020	.060	.129
test5	.019	.050	.109	.019	.019	.059	.120

TABLE VIII

Average runtime required for total wirelength estimation at a each level of top-down placement, reported in CPU seconds for a 300MHz Sun Ultra-10 workstation with 128MB RAM. Runtimes are averaged over all placement levels for each instance.

costs, versus the final sum of I1S heuristic RStMT costs. Table VII gives the average of all the values. We see that our new estimator HBBN is *substantially* better in the on-line context than previous methods of estimating either sum of bounding box half-perimeters or sum of RStMT costs. In other words, we can obtain accurate estimates of the final values of these objectives, relatively early in the top-down placement process. This allows pruning of bad solution paths, and large potential runtime savings. At the same time, even for *a posteriori* estimation our new methods are superior to previous approaches. All our heuristics are very fast (see Table VIII), and represent a negligible amount of time relative to the total top-down placement effort (see Table VI).

VII. Conclusions and Future Work

We have developed new wirelength estimation techniques appropriate for top-down floorplanning and placement synthesis of row-based VLSI layouts. Our methods give accurate, linear-time approaches, typically with sublinear time complexity for dynamic updating of estimates (e.g., for annealing placement). The new techniques offer advantages not only for early on-line wirelength estimation during top-down placement, but also for *a posteriori* estimation of routed wirelength given a final placement. In developing these new estimators, we have made several contributions, including (i) insight into the contrast between *region-based* and *bounding box-based* RStMT estimation techniques; (ii) empirical assessment of the correlations between pin placements of a multi-pin net that is contained in a block; and (iii) new wirelength estimates that are functions of a block's complexity (number of cell instances) and aspect ratio AR.

We have validated our new techniques experimentally using test cases from industry; the HBBN and HBBNtab estimators are substantially superior to previous methods (Cheng). Our ongoing research addresses such issues as (1) confirming that our new cost estimates can successfully drive partitioning- and annealing-based placers to improved solutions, and (2) better understanding of the dependence of pin distributions on netlist topology.

VIII. Acknowledgments

We thank the anonymous reviewers for many comments which have greatly improved this work.

Appendix

Theorem 3.1 Consider n random points, each of which is independently and uniformly distributed on segment $[a_i, b_i]$, $i = 1, ..., n$. Let $A = a_1$ and $B = \min_i b_i$ and assume $a_i \leq a_{i+1} \leq \min_i b_i$. (Any segment with $a_k > B$ can be ignored.) Then the expected minimum $\mathcal{E}_{min}$ is

$$\mathcal{E}_{min} = A + \int_A^B \left(1 - \frac{t - a_1}{b_1 - a_1}\right) dt - \sum_{i=2}^{n-1} \int_{a_i}^B \frac{t - a_i}{b_i - a_i} \prod_{j=1}^{i-1} \left(1 - \frac{t - a_j}{b_j - a_j}\right) dt \tag{4}$$

Proof. First note that in the context of Formula (1)

- $p_i(t) = \frac{t - a_i}{b_i - a_i}$ for $t \in [a_i, B]$
- $p_i(t) = 0$ for $t \leq a_i$, $p_1(t) = 1$ for $t \geq B$
- For given t and i, if $p_i(t) = 0$, $p_i(t)$ can be safely omitted from $\prod_{i=1}^n (1 - p_i(t))$ in Formula (1), while $p_1(t) = 1$ zeros this product regardless of all other functions. Therefore, for $t < a_i$, $\prod_{j=1}^n (1 - p_j(t)) = \prod_{j=1}^{i-1} (1 - p_j(t))$. Also, $\prod_{i=1}^n (1 - p_i(t))$ in Formula (1) is supported on $[A, B]$, and the conditions of the Theorem apply.

Now rewrite Formula (1) as

$$A + \sum_{i=1}^{n-1} \int_{a_i}^{a_{i+1}} \left[\prod_{j=1}^n (1 - p_j(t))\right] dt + \int_{a_n}^B \left[\prod_{j=1}^n (1 - p_j(t))\right] dt$$

and transform into

$$A+\sum_{i=1}^{n-1}\int_{a_i}^{a_{i+1}}\left[\prod_{j=1}^{i}(1-p_j(t))\right]\,dt+\int_{a_n}^{B}\left[\prod_{j=1}^{n}(1-p_j(t))\right]\,dt$$

by dropping insignificant $p_i(t)$ as explained above. In order to reduce the computational complexity of this expression, we are going to merge groups of integrals over $[a_i, a_{i+1}], [a_{i+1}, a_{i+2}]...[a_n, B]$ into integrals over $[a_i, B]$ which requires equalizing the integrands. Rewrite the expression above as

$$A+\sum_{i=1}^{n-1}\int_{a_i}^{a_{i+1}}\left[(1-p_i(t))\prod_{j=1}^{i-1}(1-p_j(t))\right]\,dt+\int_{a_n}^{B}\left[(1-p_n(t))\prod_{j=1}^{n-1}1-p_j(t)\right]\,dt$$

and, using the distributive law, transform every occurrence of $(1-p_*(t))\prod$ into $\prod -p_*(t)\prod$ until all such occurrences have empty $\prod$ (i.e., apply the distributive law recursively). This yields

$$A+\sum_{i=1}^{n-1}\int_{a_i}^{a_{i+1}}\left[(1-p_1(t))-\sum_{k=2}^{i}p_k(t)\prod_{j=1}^{k-1}1-p_j(t)\right]\,dt+\int_{a_n}^{B}\left[(1-p_1(t))-\sum_{k=2}^{n-1}p_k(t)\prod_{j=1}^{k-1}1-p_j(t)\right]\,dt$$

Collect the terms with $(1-p_1(t))$ and apply $\int_{a_1}^{a_2}+\int_{a_2}^{a_3}+\ldots+\int_{a_{n-1}}^{a_n}+\int_{a_n}^{B}=\int_{a_1}^{B}$ to simplify:

$$A+\int_{a_1}^{B}(1-p_1(t))dt-\sum_{k=2}^{n-1}\sum_{i=k}^{n-1}\int_{a_i}^{a_{i+1}}p_k(t)\prod_{j=1}^{k-1}(1-p_j(t))dt-\sum_{k=2}^{n-1}\int_{a_n}^{B}p_k(t)\prod_{j=1}^{k-1}(1-p_j(t))dt$$

Similar treatment of terms with $p_k(t)\prod_{j=1}^{k-1}(1-p_j(t))$ yields

$$A+\int_{a_1}^{B}(1-p_1(t))dt-\sum_{k=2}^{n-1}\int_{a_1}^{B}p_k(t)\prod_{j=1}^{k-1}(1-p_j(t)dt$$

which gives Formula 4.

$\square$

References

[1] J. Beardwood, J. H. Halton, and J. M. Hammersley. The shortest path through many points. In *Proceedings of the Cambridge Philosophical Society 55*, pages 299–327, 1959.

[2] C.-L. E. Cheng. Risa: Accurate and efficient placement routability modeling. In *Proceedings IEEE International Conf. on Computer-Aided Design*, pages 690–695, 1994.

[3] F. R. K. Chung and F. K. Hwang. The largest minimal rectilinear steiner trees for a set of n points enclosed in a rectangle with given perimeter. *Networks*, 9(1):19–36, Spring 1979.

[4] F. J. Kurdahi D. Sroobandt. On the characterization of multi-terminal nets in computer systems. Technical Report DG 97-07, Riksuniversiteit, Gent, October 1997.

[5] F. J. Kurdahi D. Sroobandt. An interconnection length estimation fot multi-point nets. In *Proceedings of the 16th IASTED International Conference on Applied Informatics*, pages 223–225, February 1998.

[6] W. E. Donath. Placement and average interconnection lengths of computer logic. *IEEE Transactions on Circuits and Systems*, CAS-26(4):272–277, April 1979.

[7] A. A. El Gamal. Two-dimensional stochastic model for interconnections in master slice integrated circuits. *IEEE Transactions on Circuits and Systems*, CAS-28(2):127–138, February 1981.

[8] M. Feuer. Connectivity of random logic. *IEEE Transactions on Computers*, C-31(1):29–33, January 1982.

[9] J. Griffith, G. Robins, J. S. Salowe, and T. Zhang. Closing the gap: Near-optimal steiner trees in polynomial time. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 13(11):1351–1365, November 1994.

[10] T. Hamada, C.-K. Cheng, and P. M. Chau. A wire length estimation technique utilizing neighborhood density equations. In *Proc. ACM/IEEE Design Automation Conf.*, pages 57–61, 1992.

[11] W. Hebgen and G. Zimmermann. Hierarchical netlength estimation for timing prediction. In *Proceedings of the ACM/SIGDA Physical Design Workshop*, pages 118–125, 1996.

[12] F. K. Hwang. On steiner minimal trees with rectilinear distance. *Siam Journal of Applied Mathematics*, pages 104–114, January 1976.

[13] A. B. Kahng and S. Muddu. Efficient gate delay modeling for large interconnect loads. In *Proceedings of the IEEE Multi-Chip Module Conference*, pages 202–207, 1996.

[14] A. B. Kahng and G. Robins. *On Optimal Interconnections for VLSI*. Kluwer, 1994.

[15] R. H. J. M. Otten. Global wires harmful? In *Proc. International Symposium on Physical Design*, pages 104–109, 1998.

[16] R. H. J. M. Otten and R. K. Brayton. Planning for performance. In *Proc. ACM/IEEE Design Automation Conf.*, pages 122–127, 1998.

[17] M. Pedram and B. Preas. Interconnection length estimation for optimized standard cell layouts. In *Proceedings IEEE International Conf. on Computer-Aided Design*, pages 390–393, 1989.

[18] C. Sechen. Average interconnection length estimation for random and optimized placements. In *Proceedings IEEE International Conf. on Computer-Aided Design*, pages 190–193, 1987.

[19] T. L. Snyder and J. M. Steele. A priori bounds on the euclidean traveling salesman. *SIAM Journal of Computing*, 24(3):665–671, June 1995.

[20] J. M. Steele. Growth rates of euclidean minimal spanning trees with power weighted edges. *Annals of Probability*, 16(4):1767–1787, 1988.

[21] J. M. Steele and T. L. Snyder. Worst-case growth rates of some classical problems of combinatorial optimization. *SIAM Journal of Computing*, 18(2):278–287, 1989.

[22] D. Stroobandt. Improving Donath's technique for estimating the average interconnection length in computer logic. Technical report, Univ. Ghent ELIS Dept., June 1996.

[23] H. Vaishnav and Massoud Pedram. Logic extraction based on normalized netlengths. In *Proceedings of the IEEE International Conference on Computer Design*, pages 658–663, 1995.