On Partitioning vs. Placement Rent Properties

P. Verplaetse* J. Dambre D. Stroobandt† J. Van Campenhout

Department of Electronics and Information Systems
Ghent University, Belgium

{pvrplaet, jdambre, dstr, jvc}@elis.rug.ac.be

ABSTRACT

Rent's rule can be derived by direct partitioning of the circuit netlist, by indirect partitioning of the placed layout, or by averaging the number of terminals for various equally large regions of the placed circuit. It is shown that all three methods may produce different results. After investigation of the fundamental reasons for these differences, three distinct effects can be identified. The *boundary* and the *embedding effect* is present with all placement approaches, though the embedding effect may be (partly) nullified by the *grid effect* that may occur with some partitioning-based placement algorithms.

One of the main applications of Rent's rule is the estimation of wire length distributions. Both flat and hierarchical placement models can be applied, though experiments show that for the current state-of-the-art estimation techniques the latter produces better results, even for layouts that were generated using a flat placement approach. Which Rent parameters and occupation probability function should be used depends on the placement algorithm. We discuss various possibilities and present a new occupation probability function that allows better wire length estimations for partitioning-based placements.

Keywords

Rent's rule, partitioning, placement, wire length distribution, estimation.

1. INTRODUCTION

Rent's rule, and the corresponding Rent exponent, has been applied numerous times in the field of a priori wire length estimation. This empirical law states a relationship between the average number of terminals and the number of gates for a set of modules obtained by partitioning a circuit. The Rent parameters can be obtained by direct partitioning of the circuit netlist, by indirect partitioning of the placed layout, or by averaging the number of terminals for various equally large regions of the placed circuit. All three methods may produce different results. The Rent exponent obtained by direct partitioning of the netlist is usually lower than the one obtained from a layout.

In this paper we will investigate the fundamental reasons for the differences between these Rent properties. Since different Rent exponents exist, it is not evident which of them should be used for a priori wire length estimation. We will discuss the implications for wire length estimation techniques based on both hierarchical and flat placement models.

This paper is organized as follows. Section 2 begins with a review of Rent's rule, with a main focus on the definition of the partitioning, placement and average local Rent properties of a circuit. In section 3 we will focus on different placement approaches and discuss the various effects that cause the differences between the different Rent properties. In section 4 we address wire length estimation techniques, and more specifically when and how the different techniques should be applied. Finally, concluding remarks are presented in section 5.

2. RENT'S RULE

When partitioning a circuit into more or less equally sized modules constrained to a certain terminal minimization objective, there exists an empirical power-law relationship between the average number of terminals T and the number of gates in the module B:

$$T = tB^p \tag{1}$$

This is known as Rent's rule [7]; the parameters p and t are known as the *Rent exponent* and the *Rent coefficient* respectively. Small deviations to Rent's rule may occur for high and low values of the module size B; these are known as Rent region II and III [7, 10]. Sometimes it is advantageous to apply the relationship between the average number of terminals and the module size directly as a function. We will refer to this relationship

$$T = R(B) \tag{2}$$

as the *Rent characteristic*.

*Peter Verplaetse is a Research Assistant of the Fund for Scientific Research – Flanders (Belgium)(F.W.O.).

†Dirk Stroobandt is a Post-doctoral Fellow of the Fund for Scientific Research – Flanders (Belgium)(F.W.O.).

SLIP'01, March 31-April 1, 2001, Sonoma, California, USA.

The empirical relationship (1) seems valid for a large range of partitioning methods; however the exact value of the Rent parameters or, in general, the exact Rent characteristic depends on the actual partitioning method. Therefore, it is important to realize that the *Rent properties* (Rent parameters and the Rent characteristic) are a function of both the circuit and the applied partitioning method. It is meaningless to refer to Rent properties of a circuit without specifying which partitioning approach has been used.

Depending on the partitioning approach the Rent properties of a circuit can be divided into a few distinct classes. When direct netlist partitioning is applied, we refer to the *partitioning Rent properties*. Alternatively, a circuit can be placed first, and a partitioning can be obtained indirectly from this placement information. This leads to what we will call the *placement Rent properties*. A third possibility is to relax the strict definition of Rent's rule, and to measure the average number of terminals of a set of more or less equally sized modules that do not necessarily have to form a partition of the original circuit. However, when this approach is applied to modules that are obtained indirectly from a placement, quite different results can be obtained. We will refer to the latter as the *average local Rent properties*.

For the remainder of this section we will focus on these different approaches to measure the Rent properties of a circuit. First, some details on the experimental setup are provided.

2.1 Experimental setup

We conducted a number of experiments on a set of 8 synthetic circuits, which were generated by `gnl`, a research tool for synthetic benchmark generation [11]. The advantages of using synthetic benchmark circuits over real ones are threefold. First, the number of gates can be controlled exactly. This makes the comparison of actual wire length distributions with theoretical ones more straightforward, since most hierarchical wire length estimation models only work fine for circuits where the number of gates G is a power of 4. Second, the complexity of the interconnection topology can be controlled by aiming at a target Rent exponent. Third, the synthetic circuits are very homogeneous (in a statistical sense [14]), and can be generated such that there is virtually no existence of a Rent region II.

Each circuit has exactly $G = 16384$ gates and was generated with different target Rent exponents. The multi-terminal nets of the circuits were converted to 2-terminal nets by adding outputs to the gates for each additional sink of the multi-terminal nets. As such, a source-sink net model is imposed. This yields better result for the current wire length estimation models, since these do not correctly take the influence of multi-terminal nets into account.

We placed the test circuits on a 128 by 128 array using three different placement approaches. All of them try to minimize the total wire length; timing issues and routability issues such as congestion are not considered. First, a flat placement by simulated annealing (SA) was performed. We applied a very slow cooling schedule (with a temperature factor of 0.98), and annealed to quite low temperatures in order to obtain near optimal placements. Next, a hierarchical placement was obtained by Plato, a research tool that generates placements based on quadrisection. Details on this placement algorithm will be provided in the next section. Finally, the circuits were placed with QPlace (version 5.0.55, incorporated in Cadence Silicon Ensemble DSM version 5.2), which is a hierarchical placement algorithm based on quadratic placement. The average wire lengths of the test circuits placed with the different placement algorithms are reported in table 1.

Table 1: Wire lengths.

Circuit	$\overline{\ell}_{\mathrm{SA}}$	$\overline{\ell}_{\mathrm{Plato}}$	$\overline{\ell}_{\mathrm{QPlace}}$
t1	2.627	2.475	3.467
t2	2.848	3.001	3.932
t3	3.348	3.628	4.690
t4	3.781	4.428	5.210
t5	4.086	4.850	5.741
t6	5.510	6.336	7.305
t7	7.187	8.518	9.032
t8	8.669	10.37	11.52

2.2 Partitioning Rent properties

The partitioning Rent properties are obtained by directly partitioning the netlist using a net or terminal minimization criterion. An efficient approach to derive different partitions with varying module size is to recursively partition the circuit into k parts. Each level h of the resulting *partitioning tree* defines a partition of the original circuit into k^h parts of size $B = k^{-h}G$, where G is the number of gates in the circuit.

The partitioning Rent properties were measured by recursively bipartitioning the test circuits with hMetis [6]. The partitioning Rent parameters are reported in the first columns of table 2. Figure 1 shows (amongst others) the partitioning Rent characteristic for circuit `t5`.

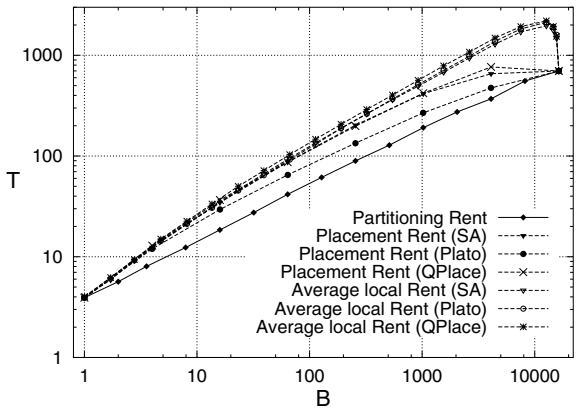

Figure 1: Rent characteristics for circuit `t5`.

2.3 Placement Rent properties

Given a placement of a circuit, one can calculate the Rent properties that correspond to this placement. The following partitioning scheme can be applied. Place a *partitioning grid* over the placed circuit. This (regular) grid induces a partition of the circuit, where gates located in the same grid cell belong to the same module. Since every grid cell has more or less the same amount of gates B, the average number of terminals for this set of modules results in a sample point of the Rent characteristic. By repeating this process

Table 2: Rent parameters for all circuits.

Circuit	Partitioning		SA				Plato				QPlace			
	p	t	p'	t'	p''	t''	p'	t'	p''	t''	p'	t'	p''	t''
t1	0.264	5.06	0.501	6.94	0.530	7.35	0.429	7.08	0.533	6.97	0.503	6.74	0.539	7.79
t2	0.379	4.39	0.504	7.21	0.536	7.63	0.436	6.95	0.545	7.17	0.533	6.66	0.554	7.87
t3	0.480	4.00	0.545	6.90	0.557	7.80	0.502	6.56	0.572	7.18	0.553	6.77	0.572	8.04
t4	0.533	3.84	0.587	6.35	0.576	7.76	0.545	6.17	0.595	7.05	0.583	6.53	0.580	8.36
t5	0.561	3.92	0.635	5.67	0.610	7.07	0.558	5.96	0.608	7.09	0.589	6.75	0.618	7.42
t6	0.661	3.49	0.689	5.30	0.665	6.56	0.642	5.38	0.677	5.95	0.682	5.60	0.658	7.14
t7	0.703	3.61	0.696	5.80	0.704	6.40	0.671	5.59	0.724	5.58	0.679	6.24	0.702	6.76
t8	0.811	2.80	0.790	4.56	0.794	4.87	0.763	4.50	0.795	4.43	0.796	4.79	0.780	5.42

for various grid sizes the complete Rent characteristic can be measured.

For a reasonably good placement of a circuit, gates that are strongly connected will be placed closely together. Hence this partitioning scheme will be favorable (though not necessarily optimal) for the minimization of the number of nets that are being cut by the grid. Consequently, the number of terminals for each module will somehow be minimized under this scheme. The resulting Rent properties are defined as the *placement Rent properties* $R'(B)$, p' and t', and depend on the applied placement algorithm. They are fundamentally different from the partitioning Rent properties of the same circuit.

We measured the placement Rent characteristics with partitioning grids of size 2^i (with i an integer from 1 to 7) and estimated the Rent parameters by fitting a line to these characteristics; the results are reported in table 2. The placement Rent characteristics of circuit `t5` are displayed in figure 1. Apparently, the placement Rent characteristics lie systematically above the partitioning Rent characteristic.

2.4 Average local Rent properties

For some applications the expected value of the number of terminals of a single module defined by the gates within a square or rectangular region of the layout is required. This can be estimated by the average terminal count for all possible locations of the region within the layout area. This average is fundamentally different from the placement Rent characteristic $R'(B)$, since the latter is the average over only a few specific locations of the regions (namely the grid cells of the regular partitioning grid). We will refer to this average over all possible locations as the *average local Rent characteristic* $R''(B)$.

The Rent characteristics of circuit `t5` are plotted in figure 1. The average local Rent characteristics are systematically higher than the corresponding placement Rent characteristics. Similar to regular Rent characteristics, a line can be fitted, which results in the *average local Rent parameters* p'' and t''. For the various placements of the test circuits, these parameters are reported in table 2.

3. IMPACT OF PLACEMENT ON RENT PROPERTIES

In this section we will investigate the reasons for the differences between the different Rent properties. Essentially, three distinct effects will be identified: the *boundary* and the *embedding* effect, which are present in all placements, and the *grid* effect, which only occurs with some partitioning-based placement algorithms. Since the different effects that may occur depend on the approach taken for placement (flat or hierarchical), we will investigate these different approaches separately, and discuss the different effects along the way.

Table 3: Rent parameters and average wire lengths for SA placements of circuit `t5`.

Runtime	p'	t'	p''	t''	$\overline{\ell}_{\mathrm{SA}}$
0%	0.967	4.34	0.828	8.12	85.15
8.5%	0.892	5.39	0.790	9.25	46.30
15%	0.815	6.64	0.749	10.31	28.14
24%	0.716	8.26	0.690	11.35	15.32
41%	0.636	9.01	0.611	12.54	8.673
61%	0.595	8.51	0.591	10.59	5.804
100%	0.635	5.67	0.610	7.07	4.086

3.1 Flat placement

The most straightforward approach for flat placement is simulated annealing [13, 8].[1] A typical property of a simulated annealing-based placement algorithm is the gradual increase of the placement quality. In order to analyze the impact of the placement effort, we measured the Rent properties and wire length distributions at different points in time during the placement of circuit `t5` (table 3). The wire length distributions are displayed in figure 2; the placement and average local Rent characteristics are shown in figure 3.

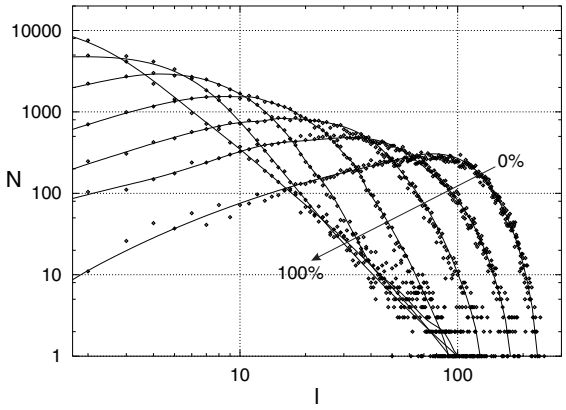

Figure 2: wire length distributions for SA placements of circuit `t5`.

[1] Simulated annealing can also be used for hierarchical placement [13]

The placement algorithm starts from an initial random placement. At this point, the Rent exponent is almost 1, as could be expected. As the placement gradually improves, the wire length distribution approaches the theoretical distribution [9] of ℓ^{2p-3}, and the Rent characteristics shift towards the partitioning Rent characteristic. At each stage of the placement, the initial slopes of the placement and the average local Rent characteristics are identical. At the end of the placement, when a near optimal wire length distribution is obtained, there still remains quite a difference between the placement and average local Rent characteristic, and also between these two and the partitioning Rent characteristic.

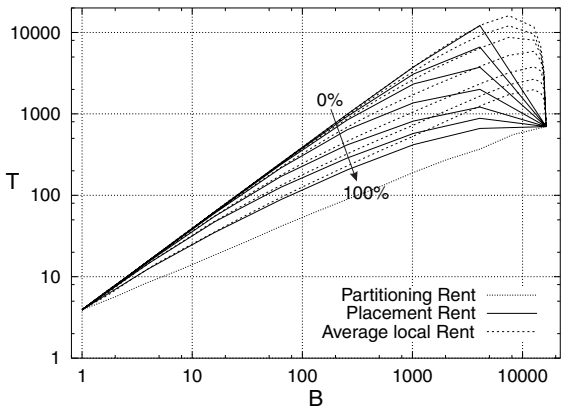

Figure 3: Rent characteristics for SA placements of circuit `t5`.

Boundary effect

For flat placements the difference between the placement and the average local Rent characteristics is due to the boundary effect: there are only a limited number of nets at the boundary of the layout area. For a module defined by a region of the placement that partly collides with the layout boundary the number of terminals will be smaller, since only a few nets will be cut at one side (or two in case of a corner location) of the region. For the placement Rent characteristic at higher values of the module size, the impact of this boundary effect is very high, since quite a lot of the grid cells abut with the layout boundary. As a result, the average terminal count of modules defined by the grid cells will be much lower than what would be expected for an infinite layout. Since the average local Rent characteristic is measured over all possible region locations, the impact of this boundary effect is much smaller.

Embedding effect

The difference between both the placement and average local Rent characteristics and the partitioning Rent characteristic can be assigned to the *embedding effect*: when partitioning a netlist into modules, any two gates that are strongly connected, will most likely be assigned to the same module, such that all strong connections reside inside the modules, and only weaker connections exist between the modules. For a good placement of this circuit, any two gates that are strongly connected, will be placed closely together, such that their connections are short. Consider a region of the layout. All gates that are located at the perimeter of the region, will not only be strongly connected to the neighboring gates within the region, but also to the neighboring gates on the outside of the perimeter. A substantial amount of the terminals of the module defined by the region under consideration, will be caused by such strong connections being cut by the perimeter. This imposes a certain lower bound on the exponent associated with the average local Rent characteristic, depending on the dimension D of the architecture in which the circuit is embedded [9]:

$$p'' \geq 1 - \frac{1}{D} \tag{3}$$

Indeed, in table 2 we have $P'' > 0.5$ for all cases. The relationship between the partitioning Rent exponent and the exponent associated with the average local Rent characteristic can be derived by using a terminal conservation technique and a model for the wire length distribution, as shown in a companion paper by Dambre et al. [3].

3.2 Hierarchical placement

Most hierarchical placement algorithms are based on partitioning [12, 1, 15]. The layout area is recursively partitioned into 2 (bisection) or 4 (quadrisection) parts. The netlist is also recursively partitioned, and at each level modules are assigned to regions of the layout. This stage yields a global placement; overlapping cells may exist. During detailed placement, the cells are assigned to a position such that all overlap is eliminated. Quite often the placement is further refined with a low temperature annealing stage [8, 16] or greedy algorithm [15] to further optimize the placement.

Such placement algorithms impose a (not necessarily regular) grid on the layout. This has an impact on the placement quality, the final wire length distribution and the Rent characteristic. To analyze the impact of the different stages in hierarchical partitioning-based placement, we developed a research tool (Plato) that implements a placement algorithm based on quadrisection. The different steps of this algorithm are:

1. *Partitioning tree generation*

   The circuit is recursively bipartitioned with hMetis [6]. The imbalance constraint is set for minimum imbalance.

2. *Global placement*

   The layout is recursively divided into 4 equal parts, which yields a regular *placement grid*. During each quadrisection step the modules at the odd levels of the partitioning tree are assigned to 4 regions of the layout. At each level, $4! = 24$ assignments of the 4 modules to the 4 regions are possible. First, these assignments are performed at random, and then a multi-pass greedy top-down *swapping* of the modules is performed to optimize the global placement for minimum wire length.

3. *Detailed placement*

   Because of slight imbalances during the partitioning of the netlist, some cell overlap may exist after global placement. This overlap is eliminated by considering a much finer grid and applying a compaction algorithm. Typically, the area is compacted with a factor of 2 in both the horizontal and vertical direction.

4. *Refinement*

   The compaction algorithm tries to maintain the relative position of the gates. However, this algorithm is not perfect, and the wire length is typically increased by 10% due to detailed placement. A quick multi-pass greedy refinement step can repair most of these deformations, and can even further improve the placement by a small amount. Experiments have shown that a low-temperature annealing step can produce slightly better results, but at a much higher cost in runtime.

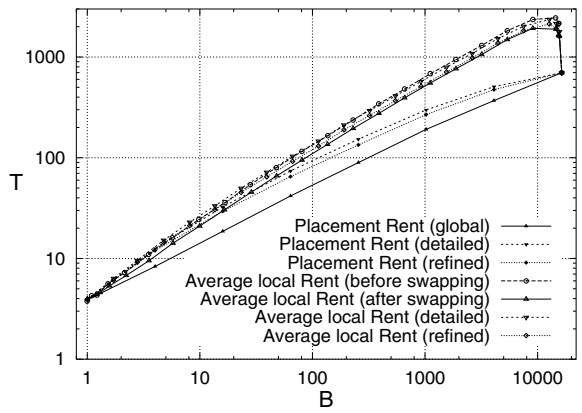

Figure 4: Rent characteristics for different stages of Plato placement of circuit `t5`.

Figure 4 shows the placement and average local Rent characteristics of circuit `t5` after the various stages of placement with Plato. The Rent parameters and average wire lengths are listed in table 4. The placement Rent characteristic after global placement collides with the partitioning Rent characteristic. This is obvious, since our approach uses regular placement grids which are identical to the partitioning grids that are being used for measurement of the placement Rent characteristic. The embedding effect seems to have disappeared. However, a slight offset in the position of the partitioning grid versus the placement grid has a huge impact on the placement Rent characteristics after global placement, as demonstrated in figure 5.

Table 4: Rent parameters and average wire lengths for different stages of Plato placement of circuit `t5`.

Stage	p'	t'	p''	t''	$\overline{\ell}_{\text{Plato}}$
Before swapping	0.565	3.87	0.655	5.95	6.677
After swapping	0.565	3.87	0.673	4.73	4.968
Detailed	0.569	6.32	0.609	7.80	5.465
Refined	0.558	5.96	0.608	7.09	4.850

The reason for this behavior becomes clear when inspecting the net-cut profile of the placed circuit, i.e. the number of nets that are cut when intersecting the layout at different positions (figure 6). This net-cut profile is very peaked as a result of the hierarchical partitioning-based approach, when comparing to the profile for the circuit placed with SA. When the partitioning grid is aligned with the placement grid, the net cuts are being sampled in the local minima of

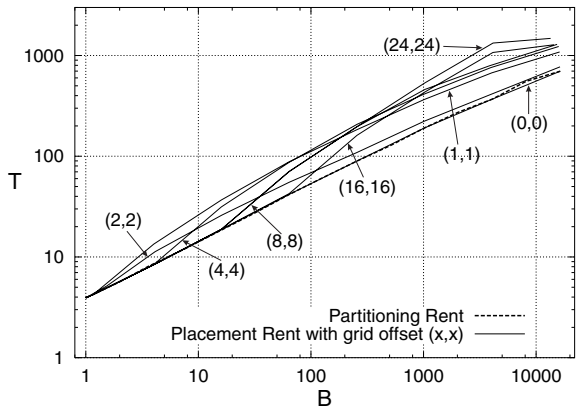

Figure 5: Placement Rent characteristics with different grid offsets of global placed circuit `t5`.

the profile. A small misalignment may cause a large increase in number of nets being cut. Detailed placement deforms the placement grid, such that perfect alignment becomes impossible. However, the grid is still partly aligned, hence the placement Rent characteristic after detailed placement is still much lower than the one for the SA-placed circuit. Refinement repairs part of the grid deformation, resulting in a small decrease of the Rent characteristics.

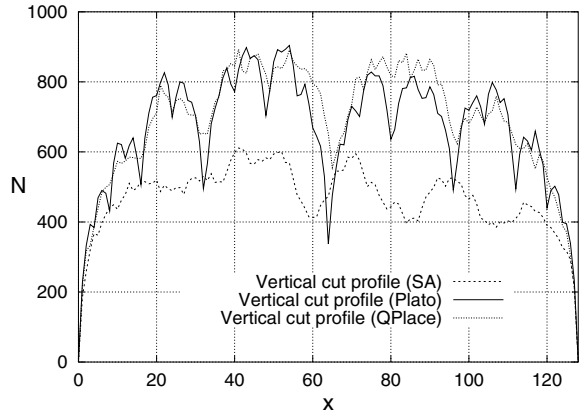

Figure 6: Vertical cut profiles for different placements of circuit `t5`.

Grid effect

The decrease of the number of terminals due to grid alignment, is called the *grid effect*. It can partly nullify the embedding effect; in case of perfect alignment the embedding effect is completely eliminated. Figure 6 also explains the difference between a partitioning-based net-cut optimization approach and a length optimization approach for placement. Note that the cut profile for QPlace reveals a slightly imbalanced initial cut. Imbalance during partitioning results in an irregular placement grid. Because the possibilities for grid alignment with the regular partitioning grid disappear, there is no noticeable grid effect, as can be seen from figure 1. Similar to the boundary effect, the grid effect can have a profound impact on the placement Rent characteristic, but has almost no impact on the average local Rent characteristic.

4. IMPLICATIONS FOR WIRE LENGTH ESTIMATION

Many a priori wire length estimation techniques make use of Rent's rule as a basis for the placement model. The Rent parameters p and t appear in the expressions for the estimated wire lengths or wire length distributions. Since three distinct types of Rent properties can be measured, it is not always clear which properties have to be used.

Depending on the placement model, there are two different approaches for wire length estimation based on Rent's rule. An overview of these different techniques is presented in [2]. With a flat placement model such as that of Davis et al. [4], the wire length distribution is estimated as the product of a *site density* function $D_b(\ell)$ and an *occupation probability* $q(\ell, p)$. The site density function is the enumeration of all possible paths in the architecture. It represents the entire collection of placement sites for the wires. The occupation probability then assigns to each of the placement sites a probability that the site is occupied by a wire.

$$N(l) = N_{\text{total}} \frac{q(\ell,p) D_b(\ell)}{\sum_{\ell=1}^{2L} q(\ell,p) D_b(\ell)} \tag{4}$$

Note that the occupation probability depends on the value of the Rent exponent p. With a hierarchical placement model, such as that of Donath [5] or Stroobandt [9], the layout bounding box is recursively quadrisectioned. At each level of the hierarchy h, the wire length distribution of the nets that would be cut during a partitioning at that level, are estimated as the product of a site function $D_c(\ell, h)$ and an occupation probability $q(\ell, p)$. The global wire length distribution is then given by the sum of the wire length distributions at each level:

$$N(\ell) = \sum_{h=1}^{H} N_{h,\text{total}}(p) \frac{q(\ell,p) D_c(\ell,h)}{\sum_{\ell=1}^{2L} q(\ell,p) D_c(\ell,h)} \tag{5}$$

The total number of nets cut at level h of the hierarchy is calculated as a fraction of the difference between the total number of terminals for all four parts at level $h-1$ and the number of terminals at the next level. The number of terminals is estimated by Rent's rule, hence $N_{h,\text{total}}$ depends on the Rent exponent p as well.

The choice of which Rent properties (partitioning, placement or average local Rent properties) and which occupation probability function should be applied, depends on the placement model used for wire length estimation and the approach used for the actual placement of the layout.

4.1 Hierarchical placement model

For a partitioning-based placement approach, the hierarchical placement model seems most appropriate. For the estimation of $N_{h,\text{total}}$ the partitioning Rent exponent p should be used. This way, a good estimate is obtained for the number of nets that are added at each hierarchical level of the actual partitioning-based placement.

When a constant occupation probability is used in the hierarchical placement model, the wire length estimation approach reduces to that of Donath [5]. This corresponds to a hierarchical partitioning-based placement without further optimization, as is the case with Plato after the global placement step, but without the swapping optimization. The wire length distribution for circuit `t5` after global placement with Plato, but before swapping is shown in figure 7. The measured average wire length and Donath's estimation is reported in the first columns of table 5. Clearly, a fairly good match is obtained.

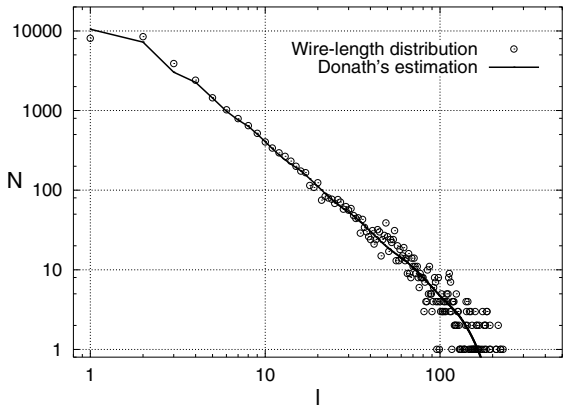

Figure 7: Measured and estimated wire length distribution for circuit `t5` after global placement with Plato, but without swapping.

For the estimation of the wire length distribution resulting from the complete placement algorithm, the impact of swapping, detailed placement and refinement must be taken into account. Assuming that the refinement step mainly compensates for the deterioration of the wire length distribution due to detailed placement, it is sufficient to incorporate the impact of swapping. This can be achieved by applying a non-constant occupation probability function. The occupation probability presented by Stroobandt [9],

$$q(\ell, p'') = C \ell^{2p''-4} \tag{6}$$

results in an underestimate of the average wire length (table 5 and figure 8). Note that the average local Rent exponent p'' should be used for the occupation probability, since (6) is a grid-less property that can be derived using terminal conservation [2]. This occupation probability assumes maximal optimization of the gate positions at each level of the hierarchy. However, a certain gate placement that may result in the optimal wire length distribution at level h, may represent a very infavorable placement for the wire length distribution at level h'. Detailed placement by swapping provides only a limited degree of freedom for wire length optimization. The unlimited degree of freedom represented by (6) results in an estimated wire length distribution with too few long wires and too many short wires. This is demonstrated in figure 8 for circuit `t5`.

During the swapping phase, the placement tool will try to place all the strongly connected regions closely together. However, some of the gates must be placed at the outer edges of the layout. If these gates connect to other regions, one will try to place the connecting gates near the inner edge of that region. Since this case may occur quite frequently, it is reasonable to assume that the optimization by swapping will be mostly effective to optimize wires that are longer than the size of the regions. At hierarchical level h, the regions are squares with side $L/2 = 2^{h-1}$. The following occupation probability function models the optimization of wires that are longer than 2^{h-1}. It approaches Stroobandt's

Table 5: Measured and estimated average wire lengths.

Circuit	No swapping		Plato			SA			
	$\overline{\ell}_{\text{Plato}}$	$\hat{\overline{\ell}}_{\text{Donath}}$	$\overline{\ell}_{\text{Plato}}$	$\hat{\overline{\ell}}_{\text{Str.}}$	$\hat{\overline{\ell}}_{\text{New}}$	$\overline{\ell}_{\text{SA}}$	$\hat{\overline{\ell}}_{\text{Str,p'}}$	$\hat{\overline{\ell}}_{\text{Str,R'(B)}}$	$\hat{\overline{\ell}}_{\text{Davis}}$
t1	3.566	3.033	2.475	1.505	2.315	2.627	2.597	3.012	2.282
t2	4.289	3.864	3.001	1.717	2.926	2.848	2.615	3.141	2.328
t3	5.278	5.036	3.628	2.051	3.802	3.348	2.882	3.546	2.499
t4	6.206	5.898	4.428	2.329	4.457	3.781	3.209	3.950	2.672
t5	6.677	6.441	4.850	2.513	4.872	4.086	3.661	4.240	3.030
t6	8.157	9.008	6.336	3.553	6.877	5.510	4.286	5.342	3.767
t7	10.75	10.44	8.518	4.303	8.035	7.187	4.378	6.791	4.434
t8	12.84	15.30	10.37	6.708	11.92	8.669	5.852	8.672	6.561

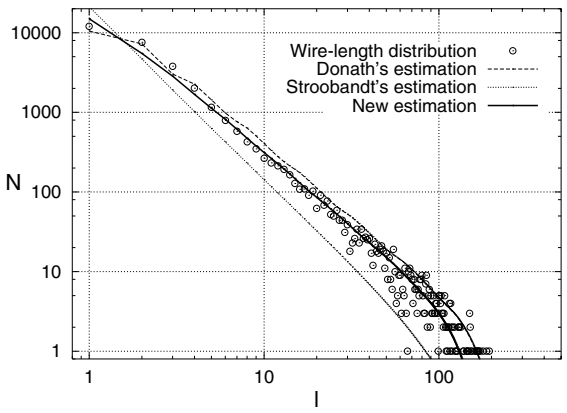

Figure 8: Measured and estimated wire length distribution for circuit `t5`, placed with Plato.

occupation probability for long wires, but represents little optimization for short wires (figure 9).

$$q(\ell, h, p'') = \frac{C'}{1 + \left(\frac{\ell}{2^{h-1}}\right)^{4-2p''}} \tag{7}$$

It is clear from figure 8 and table 5 that this new occupation probability provides much better estimates for the wire length distribution and the average wire length.

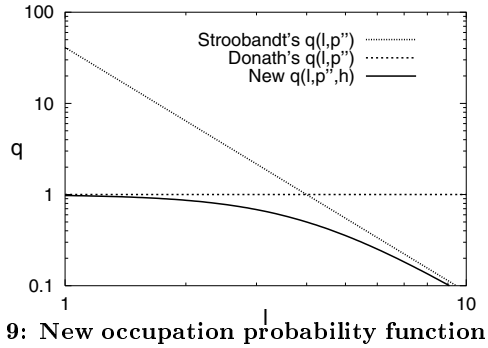

Figure 9: New occupation probability function (h=3 and p"=0.66).

When the circuit is placed with a flat placement algorithm, it seems inappropriate to use a hierarchical placement model for wire length estimation. However, one can always perform the following thought experiment: build a partitioning tree by recursive quadrisection of the placed circuit (this is similar to the measurement of the placement Rent exponent). Now place the circuit using quadrisection based on this partitioning tree. Starting from a random swapping, a greedy optimization will probably not find the original placement, but this placement does exist, and could be found using this hypothetical partitioning-based hierarchical placement approach. Therefore, it makes sense to use a hierarchical placement model. In this case, the placement Rent exponent p' should be used for the estimation of the total number of nets cut at each hierarchical level. Since the hypothetical partitioning is such that a more optimal swapping can be achieved than what would normally occur with greedy swapping, an occupation probability such as (6) should be used to reflect the true amount of optimization.

Figure 10 shows the wire length distribution of circuit `t5`, placed with SA, and the estimation using placement Rent exponent p' and Stroobandt's occupation probability function. A fairly good match is obtained, but both the number of very long and the number of very short wires is overestimated, resulting in an underestimate of the average wire length. This is due to the fact that the placement Rent characteristic cannot be fitted well by a straight line. Instead of using Rent's rule for the calculation of the number of nets being cut at each hierarchical level, one can use the placement Rent characteristic directly (for notation we refer to [2]):

$$N_{h,total} = \alpha 4^{(H-h)} \left[4R'(4^{h-1}) - R'(4^h)\right] \tag{8}$$

This yields much better estimations for the wire length distribution, as can be derived from table 5 and figure 10.

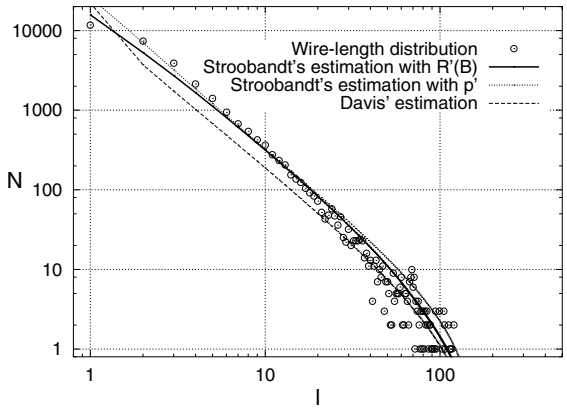

Figure 10: Measured and estimated wire length distribution for circuit `t5`, placed with SA.

4.2 Flat placement model

For flat placement approaches, it seems more appropriate to use a flat placement model for wire length estimation. However, it is perfectly possible to apply a flat placement model for hierarchical placements as well. The occupation probability used by Davis et al. [4] is based on terminal conservation. The derivation depends on Rent's rule to estimate the number of terminals for a (diamond-shaped) region, hence the Rent exponent associated with the average local Rent characteristic should be applied.

We applied the wire length estimation method by Davis et al. to estimate the average wire length and the wire length distribution of the circuits placed with the SA algorithm. The results are shown in table 5 and figure 10. Except for shortest nets, the wire length distribution is largely underestimated, resulting in an underestimate of the average wire length distribution.

5. CONCLUSIONS

Rent's rule, and the corresponding Rent exponent, has been applied numerous times in the field of a priori wire length estimation. The Rent properties can be obtained by direct partitioning of the circuit netlist (partitioning Rent properties), by indirect partitioning of the placed circuit with a partitioning grid (placement Rent properties), or by averaging the number of terminals of modules corresponding to square or rectangular regions of the placement (average local Rent properties). All three methods produce different results. The average local Rent characteristics are always above the placement Rent characteristics, which are in turn above the partitioning Rent characteristics. The difference between placement and partitioning Rent properties is due to the embedding effect, though this can by partly nullified by the grid effect for circuits placed by a partitioning-based hierarchical placement approach. The average local Rent properties mainly differ from the placement Rent properties due to the boundary effect.

Since all three methods produce different results, it is not straightforward which Rent parameters should be applied for wire length estimation. An overview is given in table 6. For a partitioning-based placement algorithm, a hierarchical placement model produces the best results when the partitioning Rent exponent is used for the calculation of the number of nets cut at each hierarchical level. In order to model the degree of optimization obtained by swapping, a new occupation probability function has been derived that is in between the no optimization model of Donath and the full optimization model of Stroobandt.

Table 6: Overview of Rent properties and occupation probabilities to be applied.

	Placement approach			
	Hierarchical	Flat		
$N_{h,\text{total}}$ $q(\ell,h)$	p p'' in (7)	p' or $R'(B)$ p'' in (6)	Hier.	Model
$q(\ell)$	p'' in an occ. prob. as in [4, 2]		Flat	

Though a flat placement model seems more appropriate for a flat placement algorithm, better results can be obtained by using a hierarchical placement model. In this case, the placement Rent parameters should be used for the calculation of the number of nets at each hierarchical level. Even better results can be obtained by using the placement Rent characteristic directly, instead of using the approximation based on Rent's rule. In either case an occupation probability function such as the model of Stroobandt should be applied to express the full optimization capabilities of the flat placement. For all evaluations of the occupation probability function (both for hierarchical as flat placement models), the average layout Rent parameters should be applied.

6. REFERENCES

[1] A. E. Caldwell, K. A. B., and M. I. L. Can recursive bisection produce routable placements? In *Proc. Design Automation Conf.*, pages 477–482, June 2000.

[2] P. Christie and D. Stroobandt. The interpretation and application of Rent's rule. *IEEE Trans. on VLSI Systems, Special Issue on System-Level Interconnect Prediction*, 8(6):639–648, December 2000.

[3] J. Dambre, P. Verplaetse, D. Stroobandt, and J. Van Campenhout. On Rent's rule for rectangular regions. In *Proc. of Intl. Workshop on System-Level Interconnect Prediction*, March 2001.

[4] 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. on Electron Devices*, 45(3):580–589, March 1998.

[5] W. E. Donath. Wire length distribution for placements of computer logic. *IBM J. of Research and Development*, 25:152–155, 1981.

[6] G. Karypis and V. Kumar. *hMetis: A Hypergraph Partitioning Package*, November 1998. Available at: http://www-users.cs.umn.edu/~karypis/metis/hmetis/main.shtml.

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

[8] M. Sarrafzadeh and M. Wang. NRG: Global and detailed placement. In *Proc. IEEE Intl. Conf. on Computer-Aided Design*, pages 532–537, November 1997.

[9] D. Stroobandt. Analytical methods for a priori wire length estimates in computer systems, November 1998. Ph.D. thesis (translated from Dutch), University of Ghent, Faculty of Applied Sciences.

[10] D. Stroobandt. On an efficient method for estimating the interconnection complexity of designs and on the existence of region III in Rent's rule. In *Proc. 9th Great Lakes Symposium on VLSI*, pages 330–331, March 1999.

[11] D. Stroobandt, P. Verplaetse, and J. Van Campenhout. Generating synthetic benchmark circuits for evaluating CAD tools. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems.*, 19(9):1011–1022, September 2000.

[12] P. Suarism and G. Kedem. An algorithm for quadrisection and its application to standard cell placement. *IEEE Trans. on Circuits & Systems*, 35(3):294–303, March 1988.

[13] W. Sun and C. Sechen. Efficient and effective placement for very large circuits. *IEEE Trans. on Computer–Aided Design of Integrated Circuits and Systems*, 14(3):349–359, March 1995.

[14] P. Verplaetse. A stochastic model for the interconnection topology of digital circuits. Technical Report PARIS 00-07, Ghent University, Belgium, ELIS Department, December 2000. Submitted to *IEEE Trans. on Very Large Scale Integration (VLSI) Systems.*

[15] M. Wang, X. Yang, and M. Sarrafzadeh. Dragon2000: Standard-cell placement tool for large industry circuits. In *Proc. of Intl. Conf. on Computer-Aided Design*, pages 260–263, November 2000.

[16] X. Yang, M. Wang, K. Eguro, and M. Sarrafzadeh. A snap-on placement tool. In *Proc. of Intl. Symp on Physical Design*, pages 153–158, April 2000.