Improved Cut Sequences for Partitioning Based Placement

Mehmet Can YILDIZ and Patrick H. Madden
State University of New York at Binghamton Computer Science Department
pmadden@cs.binghamton.edu   http://vlsicad.cs.binghamton.edu

ABSTRACT

Recursive partitioning based placement has a long history, but there has been little consensus on how cut sequences should be chosen. In this paper, we present a dynamic programming approach to cut sequence generation. If certain assumptions hold, these sequences are *optimal*. After study of these optimal sequences, we observe that an extremely simple method can be used to construct sequences that are near optimal.

Using this method, our bisection based placement tool *Feng Shui* outperforms the previously presented *Capo* tool by 11% on a large benchmark. By integrating our cut sequence method into *Capo*, we are able to improve performance by 5%, bringing the results of *Feng Shui* and *Capo* closer together.

1. INTRODUCTION

Placement by recursive partitioning is one of the oldest approaches to a fundamental problem in computer aided design. First introduced by Breuer[2], and subsequently improved by Dunlop and Kernighan[5], the approach has proven to be popular and competitive. With recent advances in hypergraph partitioning, there has been a resurgence of interest in the approach.

The work described here was motivated by a series of experiments performed with two recent partitioning based standard cell placement tools. The first tool, *Capo*[3], has been shown to obtain wire lengths that are comparable to those of a commercial placement tool, with the placements being generally routable. The second partitioning based tool, *Feng Shui*[17], was found to have comparable performance to *Capo* on the majority of MCNC benchmark circuits[10]. Both tools use modern multi-level partitioners, with *Capo* utilizing the ML partitioner[1], and *Feng Shui* using *hMetis*[8]. Both tools complete the placement process by performing branch-and-bound reordering of cells within a row.

Due to a similarity of methods, the similarity in performance on most benchmarks was not surprising. For the largest publicly available benchmark, *golem3*, however, the performance of the two tools diverged. *Feng Shui* consistently obtained wire lengths on average 11% lower than those of *Capo*, with the worst *Feng Shui* result being significantly better than the best result from *Capo*.

In this paper, we show that one factor contributing to this difference in performance is in the selection of a *cut sequence*, which appears to have increasing impact as circuit sizes grow. We will use *cut sequence* to mean the direction and order of cuts applied: for example, a horizontal cut followed by two vertical cuts of the subregions would be part of a cut sequence. We develop a mathematical foundation for wire length estimation in partitioning based placement, allowing the determination of an *optimal* cut sequence (under a simplified model, and with a number of assumptions). By studying the nature of these optimal sequences, we are able to show that an extremely simple method can be used to generate sequences that are near optimal in practice.

2. PREVIOUS WORK

Circuit placement is a well studied problem. Current approaches include linear programming or force directed methods (for example [6]), simulated annealing (for example [15]), and partitioning based methods. We focus here on partitioning based methods, derived from the approaches of Breuer or Dunlop and Kernighan.

In the classic algorithm by Breuer, repeated graph bisections were used to construct a circuit placement. With each bisection, the vertices (cells) were assigned to progressively smaller regions. Dunlop and Kernighan[5] extended this approach, through the use of an improved partitioning method[9], and also *terminal propagation*.

In these methods, and other more recent approaches, we have to choose if we are to partition a region *horizontally* or *vertically*. In Figure 1, a number of different sequences are shown. It was suggested by Breuer that a good approach might utilize alternating cut directions[2], but other strategies were considered possible.

- In the early work by Breuer, three sequences are suggested. The first is a series of cuts in alternating directions. The second employs repeated quadrisections. The third performs horizontal cuts until each region occupies a single cell row, with vertical cuts being used to complete the placement.

- In [16], sequences twelve steps long, with six horizontal cuts and six vertical cuts, were considered. There are 924 possible cut sequences; the authors only considered 64 of these, due to run time constraints. In experiments with a number of industrial circuits, no sequence studied could be considered "best," although avoidance of a long series of either horizontal or vertical cuts was beneficial. We note that the cut sequences used in this work were of uniform direction on any given "level;" sequences such as shown in Figure 1(d) were not considered.
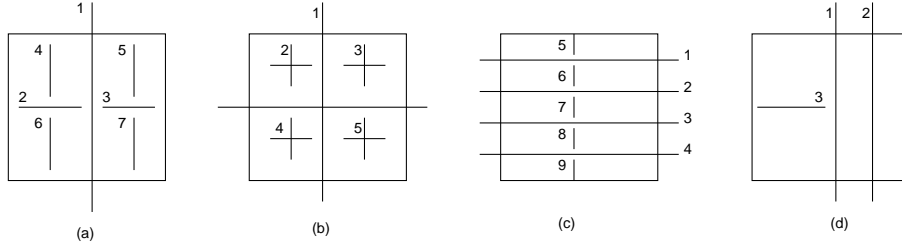
Figure 1: When using a partitioning based placement scheme, we have a number of possible cut directions to use. In (a), the sequence begins with a vertical cut, an then alternates direction at each level, producing the sequence VHHVVVV. In (b), repeated quadrisections use the sequence QQQQQ. In (c), we slice horizontally repeatedly, using the sequence HHHHVVVVV. We are not required to partition in the same direction on any given "level," as is shown in (d).

- The recent bisection based placement tool *Capo*[3] partitions a circuit based on the relative height and width of a region, with horizontal partitioning occurring if there are less than 15 cells per row.

- The quadrisection approach suggested by Breuer has been employed by a number of recent placement tools[14][7][12].

- In [13], four possible sequences are suggested, some with unbalanced partitions.

- The *Feng Shui*[17] placement tool utilizes an *aspect ratio* to determine cut direction: if the height of a region divided by its width exceeds a user-specified value, the region is bisected horizontally. If the ratio falls below the specified value, the region is bisected vertically.

Clearly, there are many approaches to the development of a cut sequence. Experiments described in [16] show that the cut sequence can have a large impact on wire length, which has motivated our work.

3. OPTIMAL GENERATION OF CUT SEQUENCES

In this section, we focus on the development of a method to obtain *optimal* cut sequences for placement by partitioning. Our sequences determine not only the *direction* of cuts to perform, but also the balance between partitions, as we are not required to bisect regions. The method can handle quadrisection directly, and can be used to determine how to mix quadrisection and bi-partitioning optimally. We also show that a relatively simple method can be used to generate a near-optimal sequence in most cases.

We follow traditional problem formulations, using nodes and hypergraph edges to model circuit elements and nets, respectively. For simplicity, we assume that we have only *two-pin* nets, and that *Rent's rule*[11] holds. We first develop a method for estimating the wire length introduced by a single partitioning, utilizing much of the approach of Breuer. Using this estimate, we can then compute the wire length of an arbitrary sequence of cuts, allowing the development of an *optimal* sequence.

Estimation of Cut Sizes and Wire Lengths

If we partition a region of area C_0 into two smaller regions having areas C_1 and C_2, we can estimate the number of nets cut in the following manner. Rent's rule states that on average, a block of C cells will have $T = k \times C^p$ propagated terminals. k is the average cardinality of cells in the netlist, while p is an experimentally derived constant (generally between 0.3 and 0.7). Utilizing this, we can determine that the number of nets split between the two subregions
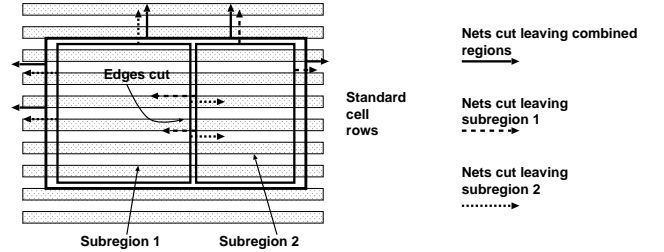

Figure 2: In any partition, the wire length introduced by the cut nets will depend on the number of cut nets, and also the size of the two new regions introduced by the partition. We can estimate the number of cut nets using Rent's rule; if we assume a Breuer-style formulation, we can also estimate the length of each cut net.

as follows. $T_0 = k \times C_0{}^p; T_1 = k \times C_1{}^p; T_2 = k \times C_2{}^p;$ cut nets $= (T_1 + T_2 - T_0)/2$. We illustrate this method in Figure 2.

The number of nets split between the two regions is the total leaving each region, *minus* the nets that would be external for the larger region. Each net passing from one subregion to the other is counted in both directions, so we must divide by 2. Clearly, the number of nets cut between a pair of regions is $(T_1 + T_2 - T_0)/2$, if we assume that Rent's rule holds.

In the development of the initial approach by Breuer, it was assumed that total wire length could be related to the cut values. Without terminal propagation, the cells which belong to a cut net will have positions that are essentially random. In Figure 3, if we partition vertically, the expected horizontal length of a cut net will be $\frac{1}{2}$ of the total x span of the initial region, while the expected vertical length will be based on the number of rows. In the following, we divide the horizontal axis by a number of *columns* (which are of arbitrary width) to simplify our formulation.

For any vertical partitioning, we split a region of r rows by c columns into two new regions which are r by c_1 and r by $c_2 = c - c_1$. If we assume random locations for the cells belonging to the cut nets, we can easily develop a simple "cut length" function $CL(r_1, c_1, r_2, c_2)$ which computes the estimated wire length for nets cut by the partitioning. A similar function can be developed for horizontal partitioning. For quadrisection, we might use $CL(r_1, c_1, r_2, c_2, r_3, c_3, r_4, c_4)$.

There is an obvious tradeoff: if we generate regions which are wide, we can expect increased horizontal wire length and decreased vertical wire length. Regions which are narrow shift expected wire lengths in the opposite direction.

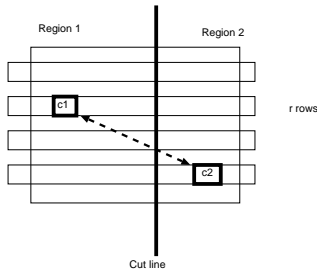Using this *CL* function, we can determine the approximate wire

Figure 3: If a net is cut, the expected length will depend on the size of the two regions to which the cells are assigned, and also on their shape.

length contributed by cut nets at any given partitioning step, *ignoring the lengths of nets cut at earlier steps*. By counting the wire length of each net the first time that it is cut, we can obtain an estimate of the wire length for any sequence of cuts. Use of terminal propagation reduces these wire lengths; we assume that this reduction will be roughly linear.

Dynamic Programming Formulation

Each partitioning splits a region into several new ones, with the total wire length depending on the contribution of nets that are cut by the partition, and also on the wire lengths introduced by the subregions. Minimization of length for an early cut may result in subregions which incur high cost. An *optimal* solution must balance these constraints; dynamic programming[4] is well suited to this problem.

Consider a region of r rows by c columns. An optimal sequence of cuts for this region will have a wire length equal to the length incurred by the first cut, plus the wire length introduced by the optimal cuts of the subregions. We use $WL(r,c)$ to indicate the *optimal* wire length for an r by c region, and can formulate the problem as follows.

$$WL(r,c) = min \begin{cases} \textbf{Horizontal cuts} \\ CL(1,c,r-1,c) + WL(1,c) + WL(r-1,c) \\ CL(2,c,r-2,c) + WL(2,c) + WL(r-2,c) \\ ... \\ CL(r-1,c,1,c) + WL(r-1,c) + WL(1,c) \\ \textbf{Vertical cuts} \\ CL(r,1,r,c-1) + WL(r,1) + WL(r,c-1) \\ CL(r,2,r,c-2) + WL(r,2) + WL(r,c-2) \\ ... \\ CL(r,c-1,r,1) + WL(r,c-1) + WL(r,1) \end{cases}$$

We wish to *minimize* the total estimated wire length required for the r by c region, and consider a variety of possible horizontal and vertical cut locations. In our dynamic programming table, we record both the estimated length, and also the direction and balance of cut which produces the minimum estimated length. Quadrisection can be integrated easily into this method, by simply modifying the computation of WL to consider cuts of this nature.

An Optimal Cut Sequence Table

With an estimated Rent parameter, and expected impact of terminal propagation for horizontal and vertical cuts, we can produce a table which defines the appropriate cut direction for any r by c region. In Figure 4, we illustrate the structure of tables we have produced; these tables have two surprising properties. First, *in all cases, optimal sequences include bisections, and not partitions of*

arbitrary balance. Second *if the ratio of rows to columns exceeds a threshold value, and optimal sequence partitions horizontally; otherwise, an optimal sequence partitions vertically*. If we adjust the wire length estimation function (to be either more or less optimistic about the effect of terminal propagation on wire length), or change the estimated Rent parameter, the general nature of the tables does not change. Similarly, adjusting the interrow spacing does not change the two observed properties.

From these observations, we can conclude that **by simply considering the ratio of region height to width, we can choose an appropriate cut direction.** This is in fact the method employed by *Feng Shui*; for an appropriately chosen *aspect ratio*, the sequence of cuts used is in fact near optimal. While we currently lack an analytic method to find an appropriate aspect ratio, we find that the value can be determined experimentally with relatively few runs of the placement tool. The impact of aspect ratio on cut sequences is somewhat granular, and relatively few values need to be considered. These properties also match Breuer's suggestion of alternating cut directions: with each partitioning, the region sizes are reduced along the "diagonal" of the optimal dynamic programming matrix, alternating between horizontal and vertical bisections.

4. EXPERIMENTAL RESULTS

As our focus in this paper has been on the development of improved cut sequences, we report results here which compare our placement approach using an aspect-ratio based sequence, the results of *Capo* (a November 1999 release), and also the results from a version of *Capo* which we have modified to use cut sequences that more closely resemble those of *Feng Shui*. Each tool was run 10 times on the benchmark *golem3*, and we report minimum, average, and maximum wire lengths obtained. *Capo* is on average three to four times faster than *Feng Shui*, but run times for both tools could be considered relatively modest. *Feng Shui* requires roughly one hour on a Pentium III 500mhz PC running Linux to complete *golem3*.

In our experiments, we find that cut sequence has greatest impact on the largest benchmarks, resulting in the most significant differences being observed on the largest available benchmark. The performance of *Feng Shui* and *Capo* on the smaller benchmarks are similar, with *Feng Shui* having slightly lower wire lengths on 8 of the 10 MCNC benchmarks. Results from runs on all MCNC benchmarks are available on the web, as well as in [17]

The experimental results are summarized in Table 1. After considering a variety of aspect ratios, we find that a value close to 2 produces the best result for our bisection based approach; observations of tables produced by our dynamic programming approach suggest that this value produces cut sequences that are near optimal. In contrast, *Capo* uses a cut sequence with aspect ratio of 1. The version of *Capo* which we have modified uses an aspect ratio of 2, similar to that of *Feng Shui*.

5. SUMMARY AND CONCLUSION

In this paper, we have resolved the problem of cut sequence generation for partitioning based placement, and have shown that *bisection* is preferable to unbalanced partitions. The cut sequences used by one recent placement approach are clearly suboptimal; a relatively minor change has resulted in a 5% improvement in the performance of *Capo* on the benchmark *golem3*, and we anticipate that similar changes can result in significant improvements in the results of other partitioning-based placement tools. We have also shown that high quality cut sequences can be generated with relatively simple methods, and that the alternating cut directions and
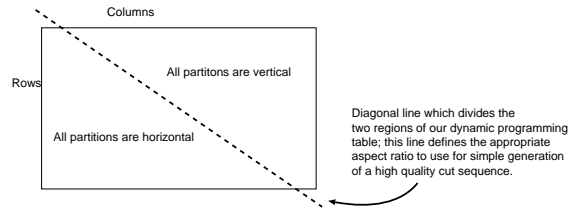
Figure 4: The dynamic programming approach produces a table with rows and columns, in which we record the direction (and balance) of an *optimal* cut to minimize total wire length. Surprisingly, the cut directions fall into two well defined regions, divided by a diagonal line. Changes in the Rent parameter estimate, cell row spacing, and estimates of horizontal and vertical wire weights only change the *slope* of this dividing line. In all cases, the optimal balance utilizes bisection.

		Feng Shui	Modified *Capo*	Original *Capo*
	min	21.900	24.100	25.246
golem3	avg	22.155	24.727	26.197
	max	23.023	25.323	27.071

Table 1: The most pronounced difference between the performance of *Capo* and *Feng Shui* was on the largest benchmark *golem3*. A minor change in the selection of cut directions for *Capo* results in a significant improvement in results.

bisections suggested by Breuer are in fact the best approach for many problems.

Our current focus is on correlation of *actual cut sizes* with those predicted by Rent's rule, and also *actual net lengths* compared to the predicted lengths. By measuring these values, we can adjust our *CL* function accordingly. We anticipate a correlation between net lengths and proximity to the center of the circuit, and can adapt our dynamic programming formulation account for this. As the accuracy of our wire length estimation improves, cut sequences should adjust to be optimal for actual circuit structures, rather than the structure predicted by relatively simple rules.

We are also integrating quadrisection and large scale multi-way partitioning methods into the *Feng Shui* placement tool. With quadrisection, we can more easily optimize the aspect ratios of regions, and we expect that optimal sequences will include both quadrisection and bisection.

The *Feng Shui* placement tool has been made available in both source code and executable form, through our research group web site. The tool parses the GSRC Bookshelf formats, and we are currently extending it to support the commercial TimberWolf formats as well. There is at least one commercial tool vendor planning support for GSRC formats, which will allow *Feng Shui* to be easily integrated into a practical design flow.

6. REFERENCES

[1] C. J. Alpert, J.-H. Huang, and A. B. Kahng. Multilevel circuit partitioning. In *Proc. Design Automation Conf*, pages 530–533, 1997.

[2] M. A. Breuer. A class of min-cut placement algorithms. In *Proc. Design Automation Conf*, pages 284–290, 1997.

[3] Andrew E. Caldwell, Andrew B. Kahng, and Igor L. Markov. Can recursive bisection alone produce routable placements? In *Proc. Design Automation Conf*, pages 477–482, 2000.

[4] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. MIT Press, 1990.

[5] A. E. Dunlop and B. W. Kernighan. A procedure for placement of standard-cell VLSI circuits. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, CAD-4(1):92–98, January 1985.

[6] H. Eisenmann and F. M. Johannes. Generic global placement and floorplanning. In *Proc. Design Automation Conf*, pages 269–274, 1998.

[7] D. J.-H. Huang and A. B. Kahng. Partitioning based standard cell global placement with an exact objective. In *Proc. Int. Symp. on Physical Design*, pages 18–25, 1997.

[8] G. Karypis, R. Aggarwal, V. Kumar, and S. Shekhar. Multilevel hypergraph partitioning: Application in VLSI domain. In *Proc. Design Automation Conf*, pages 526–529, 1997.

[9] Brian W. Kernighan and S. Lin. An efficient heuristic procedure for partitioning graphs. *Bell System Technical Journal*, 49:291–307, 1970.

[10] K. Koźmiński. Benchmarks for layout synthesis – evolution and current status. In *Proc. Design Automation Conf*, pages 265–270, 1991.

[11] B. Landman and R. Russo. On a pin versus block relationship for partitioning of logic graphs. *IEEE Trans. on Computers*, C-20:1469–1479, December 1971.

[12] S.-L. Ou and M. Pedram. Timing-driven placement based on partitioning with dynamic cut-net control. In *Proc. Design Automation Conf*, pages 472–476, 2000.

[13] N. Sherwani. *Algorithms for VLSI Physical Design Automation*. Kluwer Academic Publishers, 1995.

[14] P. R. Suaris and G. Kedem. An algorithm for quadrisection and its application to standard cell placement. *IEEE Trans. on Circuits and Systems*, 35(3):394–303, 1988.

[15] W. Swartz and C. Sechen. Timing driven placement for large standard cell circuits. In *Proc. Design Automation Conf*, pages 211–215, 1995.

[16] Kazuhiro Takahashi, Kazuo Nakajima, Masayuki Terai, and Koji Sato. Min-cut placement with global objective functions for large scale sea-of-gates arrays. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 14(4):434–446, April 1995.

[17] M. C. Yildiz and P. H. Madden. Global objectives for standard cell placement. In *GLSVLSI*, pages 68–72, 2001.