A Theorem on Partitioning a Sorted List of Numbers with an Application to VLSI Floorplanning *

Joseph R. Shinnerl
UCLA Computer Science Department
Technical Report TR040006
shinnerl@cs.ucla.edu

February 27, 2004

Abstract

Given a finite nonincreasing sequence of positive numbers $a_1 \geq a_2 \geq \ldots \geq a_n > 0$, select index j such that the absolute value of the difference between the sum of the first j terms and the sum of the remaining $n - j$ terms is minimized. An analysis shows that the ratio of these two sums is then bounded between $\gamma \equiv \max\{2, \beta\}$ and $1/\gamma$, where β is the maximum pairwise ratio of successive terms in the original sequence, $\beta \equiv \max_i a_{i+1}/a_i$. An application of this result to zero-dead-space VLSI floorplanning is described.

Key words: Sequence Partitioning, Floorplanning, Block Packing, Rectangle Packing, Aspect Ratios, Benchmarking, Dead Space, Area Partitioning

1. Balanced Partitioning of an Ordered Sequence

Given the finite, nonincreasing sequence of positive numbers

$$a_1 \geq a_2 \geq \ldots \geq a_n > 0, \tag{1.1}$$

we select a *cut index* j such that the absolute difference

$$D_k = \Big| \sum_{1}^{k} a_i - \sum_{k+1}^{n} a_i \Big| \qquad \text{is minimal for } k = j. \tag{1.2}$$

That is, we cut the ordered sequence in "half" so as to make the partial sums of the leading half sequence and trailing half sequence as nearly equal as possible. The main result of this section is that the ratio of the sums of these two half-sequences can be bounded above and below in terms of the maximum ratio of successive terms in the original ordered sequence.

Lemma 1.1. *With D_j defined as in (1.2), let $A_j = \sum_1^j a_i$, and let $\bar{A}_j = \sum_{j+1}^n a_i$. Then*

$$D_j \leq \begin{cases} a_j & \text{if} \quad A_j > \bar{A}_j \\ a_{j+1} & \text{if} \quad A_j \leq \bar{A}_j. \end{cases}$$

*Financial Support from Semiconductor Research Consortium Contract 2003-TJ-1091 is gratefully acknowledged.

Proof. Suppose $A_j > \bar{A}_j$; in this case, $D_j = A_j - \bar{A}_j$. First, observe that $D_j < 2a_j$; otherwise, $D_{j-1} < D_j$ — contradicting the minimality of D_j. Next, suppose that $D_j > a_j$, i.e.,

$$2a_j > \sum_1^j a_i - \sum_{j+1}^n a_i > a_j.$$

But then subtracting $2a_j$ gives

$$0 > \sum_1^{j-1} a_i - \sum_j^n a_i > -a_j,$$

and hence

$$0 < \sum_j^n a_i - \sum_1^{j-1} a_i < a_j,$$

another contradiction to the minimality of D_j. The case $A_j \leq \bar{A}_j$ is similar. □

Theorem 1.1. *Let $\beta = \max_i a_{i+1}/a_i$, and let $\gamma = \max\{2, \beta\}$. Then, with the notation of Lemma 1.1,*

$$\frac{1}{\gamma} \leq \frac{A_j}{\bar{A}_j} \leq \gamma.$$

Proof. Let $A = A_j + \bar{A}_j$. The conclusion can then be rephrased as

$$\frac{1}{\gamma+1} A \leq A_j \leq \frac{\gamma}{\gamma+1} A,$$

or, equivalently,

$$D_j \leq \frac{\gamma-1}{\gamma+1} A.$$

Note that for all $\gamma \geq 2$, it suffices to show that $D_j \leq A/3$, or

$$\frac{1}{3} A \leq A_j \leq \frac{2}{3} A.$$

Now if $j \geq 3$, then Lemma 1.1 ensures that $D_j \leq a_3$, and since $a_1 \geq a_2 \geq a_3 \geq \ldots \geq a_n$, it follows that $a_3 \leq A/3$. The same result also holds if $j = 2$ and $A_j < \bar{A}_j$. Therefore, we need only consider the following two cases.

Case 1: $j = 2$, and $a_1 + a_2 > \sum_3^n a_i$.

By Lemma 1.1, $a_1 + a_2 - \sum_3^n a_i < a_2$; hence,

$$a_1 \leq \sum_3^n a_i,$$

and since $a_1 \geq a_2$, we have $a_2 \leq \sum_3^n a_i$ as well. Thus, in this case,

$$\frac{1}{\gamma} \leq 1 \leq \frac{A_j}{\bar{A}_j} \equiv \frac{a_1 + a_2}{\sum_3^n a_i} \leq 2 \leq \gamma.$$

Case 2: $j = 1$.
Since $a_2 \geq a_1/\gamma$ and $\sum_3^n a_i \geq 0$,

$$\frac{A_j}{\bar{A}_j} \equiv \frac{a_1}{\sum_2^n a_i} \leq \frac{a_1}{a_1/\gamma + \sum_3^n a_i} \leq \gamma.$$

Hence, it suffices to show that $A_j/\bar{A}_j > 1/\gamma$ in this case. If $A_j > \bar{A}_j$, then we are done. Hence, we assume that $a_1 < \sum_2^n a_i$. Lemma 1.1 therefore gives $D_1 \equiv \sum_2^n a_i - a_1 < a_2$; i.e.,

$$a_1 \geq \sum_3^n a_i. \tag{1.3}$$

Now if $a_2 \leq \sum_3^n a_i$, then

$$\frac{a_1}{\sum_2^n a_i} = \frac{a_1}{a_2 + \sum_3^n a_i} \geq \frac{a_1}{a_2 + a_2} \geq \frac{a_1}{a_1 + a_1} = 1/2 \geq 1/\gamma.$$

Otherwise, if $a_2 > \sum_3^n a_i$, then (1.3) implies

$$\frac{a_1}{\sum_2^n a_i} \geq \frac{\sum_3^n a_i}{a_2 + \sum_3^n a_i} \geq \frac{1}{(a_2/\sum_3^n a_i) + 1} > \frac{1}{2} \geq \frac{1}{\gamma}.$$

□

Suppose now that a recursive sequence of balanced cuts is applied to the given ordered sequence (1.1) in order to partition it into balanced fragments. The above results can be trivially propagated through the recursion to obtain similar bounds on the ratios of subfragment sums. The next section considers an application of this idea to a novel algorithm for zero-dead-space (ZDS) floorplanning.

2. Application to Zero-Dead-Space Floorplanning

Floorplanning is the shaping and placing of n rectangular blocks of given areas in the plane, given a hypergraph-netlist specification of the connectivity among the blocks [2]. A floorplan has three main properties.

1. Estimates of the total weighted wirelength and/or timing performance of a corresponding integrated circuit represented by the floorplan.
2. Total area, i.e., the area of the smallest rectangle circumscribing the entire configuration or, equivalently, the unused area or *dead space* within this "bounding box."
3. The aspect ratios of both the blocks and the bounding box circumscribing them. We define the aspect ratio ρ of a rectangle to be the ratio of the length its longer side to that of its shorter side; hence $\rho \geq 1$ always.

In the traditional VLSICAD formulation of floorplanning, the objective is some combination of timing performance, total weighted wirelength, and total area, and the constraints are simple upper bounds (usually 2 or 3) on aspect-ratios of the blocks.

The remainder of this report considers an alternative floorplanning formulation in which block aspect ratios are implicitly minimized subject to a zero-dead-space constraint. For simplicity, all connectivity considerations, including timing performance and wirelength, are ignored. The impetus for the analysis here is to provide mathematical support for the construction of realistic floorplanning benchmarks with known optimality properties. For (i) a brief description of floorplanning's importance in the context of VLSICAD, (ii) empirical validation of this report's analysis, and (iii) comparisons of area-optimality properties of some leading academic floorplanners, the reader is referred to the companion paper [1].

2.1. A Top-Down Zero-Dead-Space Floorplanning Algorithm

The ZDS algorithm considered here is based on recursive top-down area bipartitioning. At each step, the blocks in a region are separated into two groups such that the two groups' total areas are as nearly equal as possible. The region is then cut parallel to its shorter side into two subregions such that each group fits exactly into one of the subregions. Cutting parallel to the shorter side keeps aspect ratios of subregions bounded in terms of the area variation among the blocks. Blocks are placed once they fill a sufficient fraction of their subregions; this fraction is expressed as the reciprocal of the parameter $\gamma \geq 1$.

Figure 1 shows the pseudocode for this ZDS algorithm, Algorithm 2.1. The notation for Algorithm 2.1 is as follows. Given n rectangles $r_1, \ldots, r_n$ with fixed areas $a_1 \geq a_2 \geq \cdots \geq a_n$ but variable lengths ℓ_i and widths w_i, we seek to arrange them without overlap in a given rectangle $\mathcal{R}$ of area $\mathcal{A} = \sum_1^n a_i$ such that the aspect ratios

$$\rho_i = \rho(r_i) = \max(\ell_i/w_i,\ w_i/\ell_i)$$

are bounded close to one.[1] The rectangle $\mathcal{R}$ is the *floorplanning region*. The rectangles r_i are called *blocks*.

Algorithm 2.1 is parameterized by $\rho(\mathcal{R}) \geq 1$ and $\gamma \geq 1$ and, by construction, has the following property.

Theorem 2.1. *For $\rho(\mathcal{R}) \geq 1$ and $\gamma \geq 1$, Algorithm 2.1 generates a slicing floorplan with zero dead space.* □

Although Algorithm 2.1 can accept as input any values $\rho(\mathcal{R}) \geq 1$ and $\gamma \geq 1$, the analysis below shows that the block shapes generated will be most realistic for certain choices of $\rho(\mathcal{R})$ and γ defined as follows. Let

$$\beta = \max_i a_i/a_{i+1}. \tag{2.1}$$

Then good values for γ and $\rho(\mathcal{R})$ are

$$\gamma = \max\{2,\ \beta\} \quad \text{and} \quad \rho(\mathcal{R}) \in [1,\ \gamma + 1]. \tag{2.2}$$

The utility of Algorithm 2.1 rests on the fact that for nearly all realistic circuits, all the block aspect ratios it computes are guaranteed to lie within a single small interval of the form $[1,\ \gamma + 1]$, when γ is defined as in (2.2). Hence, if the blocks are arranged in nonincreasing sorted order by area, the aspect ratios are bounded by one plus the maximum ratio of consecutive block areas, when this latter ratio exceeds 2. Otherwise, the aspect ratios are bounded above by 3. These facts are established here, under Assumptions 2.1 below.

Assumptions 2.1. *Block-locking threshold; floorplanning-region aspect-ratio bound.*

(a) *For β as defined in (2.1), $\gamma \geq \max\{2,\ \beta\}$.*

(b) *The aspect ratio of the floorplanning region satisfies $\rho(\mathcal{R}) \leq \gamma + 1$.*

Assumption 2.1(a) can be rephrased as follows: the threshold fraction of subregion area that a block must occupy in order to be shaped and locked in place is not set above 1/2. Although these assumptions are stronger than necessary to achieve zero dead space and acceptably bounded block aspect ratios, they are not very restrictive on the sets of block areas that may be considered. Further discussion of the assumptions appears at the end of the next section.

[1] By this definition, the aspect ratio of any block is always at least 1.

Algorithm 2.1. *Top-Down ZDS Floorplanning.*

input

(i) *Rectangles* $r_1, \ldots, r_n$ *with areas* $a_1 \geq \ldots \geq a_n$.

(ii) *Rectangular region* R *of area* $A = \sum_1^n a_i$ *and dimensions* $\ell \times w$, *with* $\ell / w = \rho(\mathcal{R}) \geq 1$.

(iii) $\gamma \geq 1$.

end input

if *the largest block* r_1 *satisfies* $a_1 \geq \frac{1}{\gamma} A$ **then**
 (i) make the length of one side of r_1 *equal to* w.
 (ii) place r_1 *with this shape against one side of* R.
  remark *In the analysis, let* $R(r_1)$ *denote this* R.
 (iii) **if** $n > 1$ **then**
   Replace R *by the part of* R *not used by* r_1.
   Reindex $\{r_2, \ldots, r_n\}$ *to* $\{r_1, \ldots, r_{n-1}\}$.
   if $n == 2$ **then**
    place the last block in R *and* **return**.
   else *Replace* n *by* $n - 1$.
   end if
  else return
  end if
 remark *If* R *contains more than 2 blocks, we place at most one block before repartitioning.*
end if

if $n > 1$ **then**

1. *Select* $j \in \{1, \ldots, n\}$ *such that*

$$D_j = \left| \sum_1^j a_i - \sum_{j+1}^n a_i \right| \quad \textit{is minimized.}$$

 Let $A_j = \sum_1^j a_i$ *and* $\bar{A}_j = \sum_{j+1}^n a_i \equiv A - A_j$.

2. *Cut* R *parallel to its shorter side into two rectangular subregions* R_j *and* $\bar{R}_j$ *of areas* A_j *and* $\bar{A}_j$ *respectively. Assign* $r_1, \ldots, r_j$ *to* R_j *and* $r_{j+1}, \ldots, r_n$ *to* $\bar{R}_j$.

3. *Recur on the subregions* R_j *and* $\bar{R}_j$.

end if
output
Rectangle dimensions (l_i, w_i) *and locations* (x_i, y_i) *for each* $r_i \in \{r_1, \ldots, r_n\}$.
end output

Figure 1: A Top-Down ZDS Floorplanning Algorithm.

2.2. Analysis

Throughout this section, we consider the properties of Algorithm 2.1 under Assumptions 2.1. The following lemma shows that, in order to bound the aspect ratios of the blocks, it suffices to bound the aspect ratios of the regions in which they are placed.

Lemma 2.1. *The aspect ratio ρ_i of any placed block r_i satisfies*

$$\rho_i \leq \max\{\gamma,\ \rho(R(r_i))\},$$

where $\rho(R(r_i))$ denotes the aspect ratio of the smallest subregion in which r_i is placed.

Proof. Suppose that subregion R has aspect ratio $\rho = \rho(R)$. If R contains just one block, then that block r_i will also have $\rho(r_i) = \rho$. Hence, suppose R contains more than one block. By Algorithm 2.1, the blocks $\{r_i, \ldots, r_p\}$ in R form a contiguous subsequence of the original set of blocks $\{r_1, \ldots, r_n\}$ and therefore satisfy the area decay bounds $a_k \geq a_{k+1} \geq a_k/\gamma$. Moreover, the block r_i placed in R will have one of its side lengths w equal to the shorter side length of R, as shown in Figure 2. Let ℓ_i denote the length of the other side of r_i, and let ℓ denote the length of the longer side of R.

First, suppose $\ell_i < w$. Because the algorithm requires the area a_i of r_i be at least $1/\gamma$ times the area of R, the other side ℓ_i of r_i is at least $1/\gamma$ times the length of the longer side ℓ of R. Hence,

$$\frac{w}{\ell_i} \leq \frac{w}{\ell/\gamma} = \frac{\gamma}{\rho} \leq \gamma,$$

since $\rho \geq 1$. Second, suppose $\ell_i \geq w$. Because the blocks r_k in R satisfy $a_k \geq a_{k+1} \geq a_k/\gamma$, the subregion of R containing these other blocks must occupy area at least $1/(\gamma+1)$ times the area of R, and therefore $\ell_i \leq (\gamma/(\gamma+1))\ell$. Hence,

$$\frac{\ell_i}{w} \leq \frac{\gamma}{(\gamma+1)} \frac{\ell}{w} = \frac{\gamma}{\gamma+1}\rho < \rho. \tag{2.3}$$

□

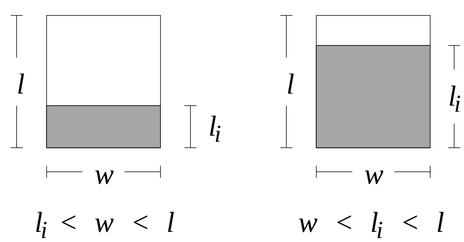

Figure 2: The aspect ratio of a block (shaded) compared to the aspect ratio of its enclosing subregion.

The following lemma bounds the aspect ratio of sibling subregions in terms of their area ratio and the aspect ratio of their common parent subregion.

Lemma 2.2. *Suppose subregion R is partitioned into subregions R_1 and R_2 with areas A_1 and A_2. Let*

$$y = \max\{A_1/A_2,\ A_2/A_1\}.$$

Then

$$\max\{\rho(R_1),\, \rho(R_2)\} = \max\left\{\frac{y+1}{\rho(R)},\ \frac{y}{y+1}\rho(R)\right\}.$$

Proof. Following the notation in Figure 3, let $A \equiv A_1$, $\rho_A = \rho(R_1)$, $a \equiv A_2$, $\rho_a = \rho(R_2)$, and assume without loss of generality that $A > a$, so that $y = A/a$. The longer side of R has length ℓ, and the shorter side has length w. Now

$$A + a = (y+1)a = \ell w,$$

and therefore

$$\ell_a = \frac{a}{w} = \frac{\ell}{(y+1)}, \quad \text{and} \quad \ell_A = y\ell_a = \frac{y}{y+1}\ell.$$

If $\rho_A \geq \rho_a$, then $\rho_A = \ell_A/w$ (otherwise, $\rho_A = w/\ell_A < w/\ell_a = \rho_a$); hence, $\rho_A = (y/(y+1))\rho(R)$. Similarly, if $\rho_a \geq \rho_A$, then $\rho_a = w/\ell_a$, and therefore $\rho_a = \dfrac{y+1}{\rho(R)}$. □

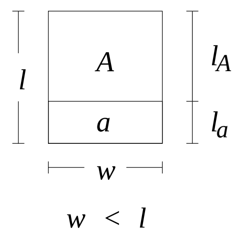

Figure 3: The aspect ratios of two sibling subregions compared to the aspect ratio of their parent subregion.

Theorem 1.1 shows that if all ratios of consecutive block areas are uniformly bounded above by $\gamma \geq 2$, then all ratios of sibling partition subregion areas are also uniformly bounded above by γ. From Lemma 2.2 and Theorem 1.1, we immediately obtain the following bound.

Corollary 2.1. *Suppose subregion R is partitioned into subregions R_1 and R_2. Then*

$$\max\{\rho(R_1),\, \rho(R_2)\} \leq \max\{\frac{\gamma+1}{\rho(R)},\, \frac{\gamma}{\gamma+1}\rho(R)\}.$$

Theorem 2.2. *Under Assumptions 2.1, the result of Algorithm 2.1 is a slicing floorplan with zero dead space and every block's aspect ratio bounded above by $\gamma + 1$.*

Proof. Follows directly from Corollary 2.1 and Assumptions 2.1, by induction. □

2.3. Interpretation and Extensions

The assumption $\gamma \geq 2$ presents no practical restriction on the sets of blocks that may be considered. It just means that the least upper bound on block aspect ratios guaranteed by the analysis here for the given algorithm is at least 3. That is, consecutive-pairwise area bounds tighter than 2 (e.g., $a_i/a_{i+1} \leq 1.5$) are not guaranteed to reduce the maximum aspect ratio below what can be attained with $a_i/a_{i+1} \leq 2$.

Similarly, a large value of γ does not necessarily indicate any large aspect ratios in the final floorplan, as Figure 4 illustrates. In the figure, one large block occupies one subregion, and several small blocks occupy another subregion. Although the area ratio of the subregions may be arbitrarily large, the presence of sufficiently many small blocks used to fill the small subregion prevents any single block's aspect ratio from becoming large.

For some designs, the presence of a few very large or very small blocks may result in a large value of γ, if γ is defined simply as $\max\{2, \max_i a_i/a_{i+1}\}$. However, a few simple preprocessing steps can usually

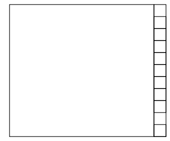

Figure 4: Block aspect ratios may all remain small, even when some subregion has a large aspect ratio.

be used to reduce this value significantly. The basic idea is simply to aggregate smaller, similarly sized blocks together until the aggregates are more comparable to larger blocks or sets of blocks. Suppose $a_m/a_{m+1} = \max_i a_i/a_{i+1}$. Define $S_m, \bar{S}_m, A_m, \bar{A}_m$ as in Section 2 above, but with $j \equiv m$ instead of j being chosen to minimize D_j. If $A_m/A_{m+1} < a_m/a_{m+1}$, then since a_{m+1} and a_m are placed in separate subregions, Theorem 1.1 ensures that γ is reduced to

$$\max\{A_m/A_{m+1},\ \max_{i \neq m} a_i/a_{i+1}\}.$$

If $\max_{i \neq m} a_i/a_{i+1} \gg A_m/A_{m+1}$, a few recursive iterations on S_m and S_{m+1} can be used to reduce the bound further, until the maximum ratio of successive r_i is comparable to the maximum ratio of sibling R_i. Although it is trivial to construct examples where this preprocessing will be useless (e.g., when $n = 2$, or when $m = n - 1$), on practical examples with large n, the reduction in γ will likely be considerable, when m is sufficiently less than n.

The above strategy will not help in the case where the smallest block's area a_n is abruptly smaller than a_{n-1}. In this case, if the areas of $r_{n-1}, r_{n-2}, r_{n-3}$, and r_{n-4} are not too dissimilar, they can be wrapped around r_n in the standard non-slicing "wheel" configuration [2]. The aggregate wheel $r_{n-4}, \ldots, r_n$ can then be treated as a single block that is reshaped along with the other blocks. If necessary, block r_{n-1} and its predecessors may first be clustered prior to wrapping around r_n.

Precise rules and analysis for the reduction of the block-locking threshold γ in the presence of mixed block sizes are left to future work. It is of course very easy to give a set of block areas for which a ZDS floorplan cannot possibly produce small aspect ratios for all blocks — e.g., $n = 2$ with $a_1 \gg a_2$. It seems clear, however, that the above ideas can be used to extend the utility of the ZDS framework to all but the most contrived block data sets.

3. Acknowledgments

The work here arose in the context of joint work with Jason Cong, Gabriele Nataneli, and Michalis Romesis on the construction of floorplanning benchmarks in VLSICAD [1]. Without their encouragement and advice, it would not have been written. A careful reading by Ashok Jagannathan also helped improve the presentation.

References

[1] J. Cong, G. Nataneli, M. Romesis, and J. Shinnerl. An area-optimality study of floorplanning. In *Proceedings of the ISPD*, Phoenix, 2004.

[2] Naveed Sherwani. *Algorithms for VLSI Physical Design Automation.* Kluwer Academic Publishers, Boston, Dordrecht, London, second edition, 1995.