Discrete Applied Mathematics 40 (1992) 303-318
North-Holland

Balanced block spacing for VLSI layout

Israel Cederbaum

Department of Electrical Engineering, Technion – Israel Institute of Technology, Haifa 32000, Israel

Israel Koren

Department of Electrical and Computer Engineering, University of Massachusetts, Amherst, MA 01003, USA

Shmuel Wimer

IBM Israel Scientific Center, Technion City, Haifa 32000, Israel

Received 15 February 1989
Revised 15 September 1989

Abstract

Cederbaum, I., I. Koren and S. Wimer, Balanced block spacing for VLSI layout, Discrete Applied Mathematics 40 (1992) 303–318.

Placement algorithms for VLSI layout tend to stick the building blocks together. This results in the need to increase the space between adjacent blocks to allow the routing of interconnecting wires. The above problem is called the *block spacing problem*. This paper presents a model for spreading the blocks uniformly over the chip area, to accommodate the routing requirements, such that the desired adjacency relations between the blocks are retained. The block spacing problem is solved via a graph model, whose vertices represent the building blocks, and its arcs represent the space between adjacent blocks. Then, the desired uniform spacing can be presented as a *space balancing problem*. In this paper the existence and uniqueness of a solution to the *one dimensional* space balancing problem are proved, and an iterative algorithm which converges rapidly to the solution is presented. It is shown that in general, the *two dimensional* problem may have no solution.

1. Introduction

The layout of VLSI chips is usually carried out in two steps: first, the building blocks are placed within the area of the chip, a step called *placement*, and then the

Correspondence to: Professor I. Koren, Department of Electrical and Computer Engineering, University of Massachusetts, Amherst, MA 01003, USA.

interconnections between them are completed, a step called *routing*. Many placement algorithms have been published in the literature and in most of those which are based on energy minimization the blocks tend to stick together (e.g. [3]), thus resulting in blockages for the routing phase. The outcome may be a chip having excessively long interconnections, and consequently, degraded performance, or even a nonfeasible layout in which the routing cannot be completed due to blockages.

Similar to placement, routing of VLSI chips have been studied intensively, and there are many well-known algorithms such as *maze routing* (e.g. [2]) and *global routing* algorithms (e.g. [1]). Both of them require some open space, sometimes called *channel*, between adjacent blocks. To satisfy this requirement the placed blocks must be spread out over the area of the chip to allow enough room for the interconnections while retaining the adjacency relationships (left–right and up–down) between blocks.

In this paper we address the problem of block spacing in VLSI layouts. The blocks within a VLSI module are interconnected by wires connected to ports located within their area. Thus, the area of a rectangular VLSI module is occupied by two types of entities: its rectangular constituting blocks and the interconnecting wires that run between the blocks. The wires running in the neighborhood of a certain block result from two origins: those that are connected to this block, and those that are passing through, on their way to other blocks. The block spacing problem does not really involve the wires that terminate in the block. The spacing for these is almost independent of the placement configuration and the routing algorithm. Therefore, the spacing for these wires can be estimated prior to the placement phase and the block can be expanded to account for this. However, the spacing for the passing through wires cannot be predicted before the placement phase since the amount of space needed depends upon the relative placement of the blocks and the particular routing algorithm which is employed later. Consequently, a reasonable way to space the blocks (which have already been expanded to account for the wires terminating in the block) is to spread them "uniformly" over the chip area. Of course, uniformity must be well defined.

The rest of the paper is organized as follows. In Section 2 we define the problem of one and two dimensional block spacing. In Section 3 we prove the existence and uniqueness of the solution for the one dimensional problem. Section 4 presents an iterative algorithm to find the one dimensional "uniformly" spaced placement and proves that the proposed algorithm converges to the unique solution of the one dimensional problem. Conclusions and problems for further research are presented in Section 5.

2. The space balancing problem

Let R_i, $1 \le i \le b$, be the rectangles corresponding to the building blocks of the layout, which are all placed within the area of the father block whose rectangular area is denoted by R_0. A placement is said to be *legal* if the building blocks do not overlap. Let $(x_i^{\rm l}, y_i^{\rm d})$ and $(x_i^{\rm r}, y_i^{\rm u})$ be the coordinates of the lower left and upper right corners of R_i, in R_0 coordinate system, respectively. A rectangle R_i is said to be *left adjacent* to the rectangle R_j if $x_i^{\rm r} \le x_j^{\rm l}$ and $[y_i^{\rm d}, y_i^{\rm u}] \cap [y_j^{\rm d}, y_j^{\rm u}] \ne \emptyset$, and if there exists no $R_k, k \ne i,j$ such that $x_i^{\rm r} \le x_k^{\rm l} \le x_k^{\rm r} \le x_j^{\rm l}$ and $[y_i^{\rm d}, y_i^{\rm u}] \cap [y_k^{\rm d}, y_k^{\rm u}] \cap [y_j^{\rm d}, y_j^{\rm u}] \ne \emptyset$. *Right adjacency* is defined similarly. In Fig. 1 the blocks R_1 and R_2 are left adjacent to R_5, while R_7 and R_8 are its right adjacent blocks, whereas R_1 and R_8, e.g., are not a pair of adjacent blocks.

The *horizontal adjacency graph* $G(U,E)$ corresponding to the placement is defined as follows: Every rectangle R_i is represented by a vertex u_i, whose weight $w(u_i)$ is defined to be the width of the rectangle R_i, i.e., $w(u_i) = x_i^{\rm r} - x_i^{\rm l}$. The vertex u_0 represents the left edge of R_0, u_{b+1} represents the right edge of R_0, and we define $w(u_0) = w(u_{b+1}) = 0$. Two vertices u_i and u_j are connected by an arc e directed from u_i to u_j if the rectangle R_i is left adjacent to the rectangle R_j. To every arc $e = (u_i, u_j)$ we assign a length $s(e)$ equal to the space (horizontal distance) between the rectangles corresponding to its end vertices, namely, $s = x_j^{\rm l} - x_i^{\rm r}$. The digraph G thus defined is acyclic and has one source u_0 and one sink u_{b+1}. The *vertical adjacency graph* $K(V,F)$ is defined similarly. Figure 2 illustrates the horizontal adjacency and vertical adjacency graphs corresponding to the placement given in Fig. 1.

Define the space along a path Ω in G, denoted by $s(\Omega)$, to be the total sum of the arc lengths (representing space between adjacent blocks) along the path. The width of the path, $w(\Omega)$, is the total sum of vertex weights (representing block widths) along Ω, including its end vertices (whose corresponding weight is zero). Finally, define the length $l(\Omega)$ of the path Ω to be the total sum of block widths and spaces between adjacent blocks along Ω, i.e., $l(\Omega) = s(\Omega) + w(\Omega)$. Obviously, all the paths connecting a pair of vertices u_i and u_j have the same length, where the length of those connecting u_0 to u_{b+1} equals the width of R_0 which is denoted by w_0.

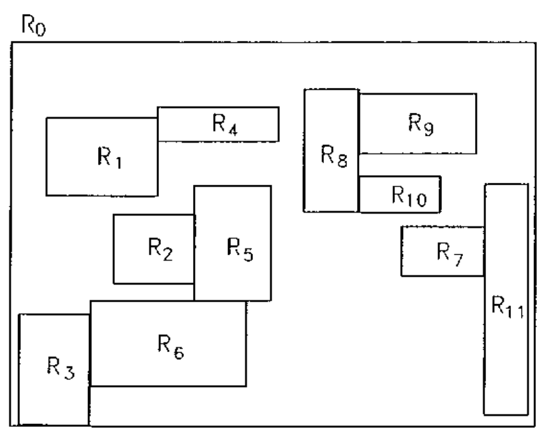

Fig. 1. Initial placement.

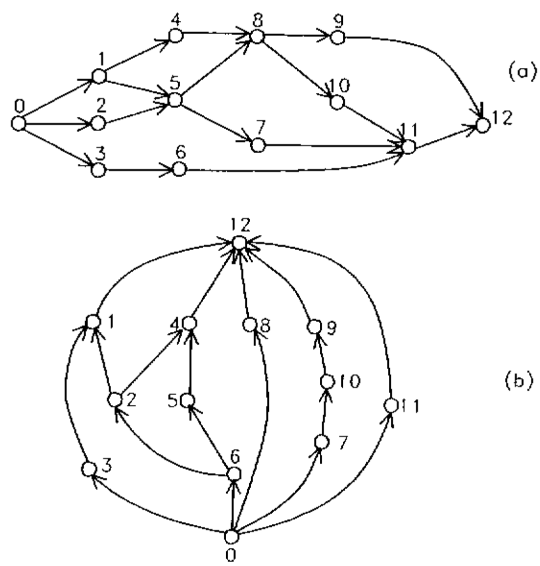

Fig. 2. Horizontal and vertical adjacency graphs.

Let Γ_i^{in} and Γ_i^{out} denote the sets of arcs entering and leaving u_i, respectively. By definition, Γ_i^{in} and Γ_i^{out} correspond to the spaces between R_i and the left adjacent and right adjacent rectangles of R_i, respectively. Let α_i and β_i denote the minimal horizontal space (distance) between R_i and any of its left adjacent and right adjacent rectangles, respectively, i.e.,

$$\alpha_i = \min\{s(e) \mid e \in \Gamma_i^{\text{in}}\}, \qquad \beta_i = \min\{s(e) \mid e \in \Gamma_i^{\text{out}}\}. \tag{1}$$

We define $\mu_i = \beta_i - \alpha_i$ to be the *horizontal imbalance* of R_i. *Vertical imbalance* is defined similarly. The placement is said to be *horizontally balanced* if

$$\mu_i = 0, \quad 1 \le i \le b. \tag{2}$$

An interesting question is whether for every given initial placement there exists a horizontal displacement of the rectangles which preserves the horizontal adjacency relations between them, and the resulting placement is horizontally balanced. This problem is called the *one dimensional space balancing problem*. Figure 3 illustrates a horizontally balanced placement obtained from the placement in Fig. 1.

Evidently, a horizontal (vertical) displacement of the rectangles does not necessarily preserve the vertical (horizontal) adjacency relations, as can be observed by comparing Fig. 3 to Fig. 1. Given an initial placement, the *two dimensional space balancing problem* is to find a horizontal and a vertical displacement of the rectangles which preserve both the horizontal and vertical adjacency relations between them, and the resulting placement is balanced in both directions. In general, this problem may have no solution as shown in Fig. 4. When the requirement to preserve

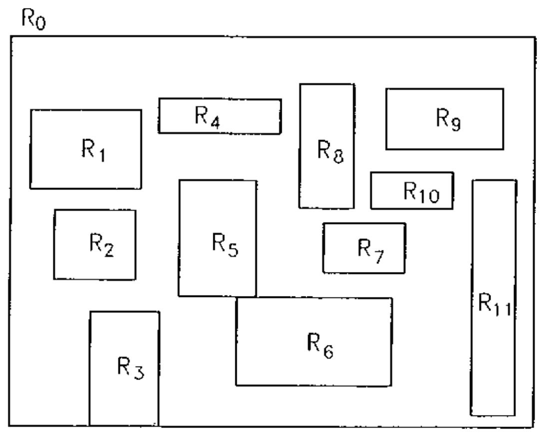

Fig. 3. Horizontally balanced placement.

the adjacency relations is relaxed, the solution might be not unique as shown in Fig. 4.

3. Existence and uniqueness of one dimensional space balancing

As will be demonstrated by construction, for every initial placement there exists

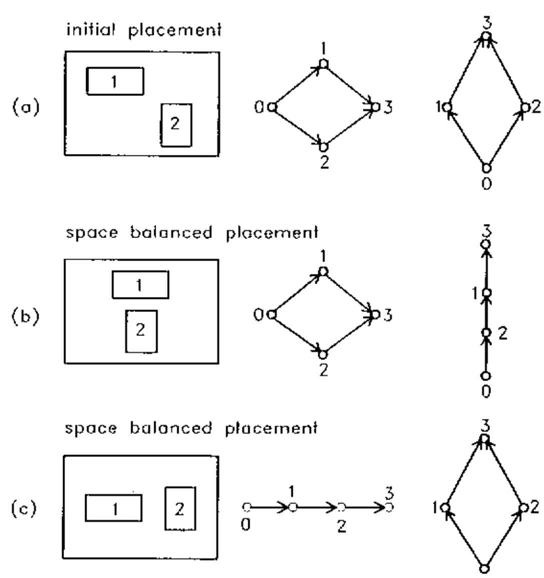

Fig. 4. Different balanced placements for the same initial placement.

a unique horizontally balanced placement. We present the existence proof first. It consists of three parts: a procedure which constructs a new weighted graph G' isomorphic to the original adjacency graph G, a proof that the new graph G' presents a feasible adjacency graph, and finally, a proof that G' presents a horizontally balanced configuration. In the following we present each part separately and then conclude by stating the existence theorem.

3.1. *Construction procedure for G'*

We construct a new graph G' isomorphic to G in an incremental manner. After an initialization step, the construction proceeds iteratively, where in every iteration some path from G is copied into G' with new arc lengths. The procedure terminates when G is completely copied into G'.

Step 0: *Initialization.* G' is empty. All the vertices and all the arcs of G are unmarked. Add the vertices u_0 and u_{b+1} to G'. Mark the vertices u_0 and u_{b+1} of G (corresponding to the left and right edges of R_0, respectively).

The following steps are repeated until G is completely copied into G'.

Step 1: *Find a new path in G.* For every path Ω between any two marked vertices of G whose remaining vertices are unmarked (and hence its arcs too) do the following: Let u_i and u_j be the tail and head vertices of the path Ω, respectively (in the first invocation of Step 1 these are the source and the sink). Let Ω^i be any path in G' from u_0 to u_i and let Ω^j be any path in G' from u_j to u_{b+1}. Notice that such paths in G' must exist since u_i and u_j are marked. Assume for the moment that we wish to augment G' with the path Ω such that the feasibility and the adjacency relations in the placement resulting from this augmentation are retained. To this end we first calculate the lengths $l(\Omega^i)$ and $l(\Omega^j)$ in G', and then calculate the desirable average space between adjacent rectangles along the path Ω in G'. This average space is given by the ratio

$$\frac{w_0 - l(\Omega^i) - l(\Omega_j) - w(\Omega) + w(u_i) + w(u_j)}{|\Omega|}, \tag{3}$$

where $|\Omega|$ is the number of arcs along Ω (in the first invocation (3) is equal to $s(\Omega)/|\Omega|$ since $l(\Omega^i) = l(\Omega^j) = 0$). The terms $w(u_i)$ and $w(u_j)$ are added to the numerator of (3) since u_i is included both in Ω^i and Ω, while u_j is included both in Ω_j and Ω. Let Ω_t be a path in G which minimizes the ratio in (3) (if there are several, choose one arbitrarily).

Step 2: *Augmentation of G'.* Add the arcs and unmarked vertices of Ω_t to G' (the two marked end vertices are already in G'). To every arc added to G' assign a length equal to the average space of an arc along Ω_t as given by (3). To every vertex added to G' assign the width of the corresponding vertex in G.

Step 3: *Updating G.* Mark the unmarked arcs and vertices along Ω_t in G (obviously, except the end vertices the entire path is unmarked in G).

Step 4: *Termination test.* If all the vertices of G are marked (and hence the arcs too) then stop, else go back to Step 1.

Notice that the way G' is augmented in Step 2, G' retains the property that all the paths between any two vertices in the horizontally adjacency graph have the same length. For the example given in Fig. 1, the first iteration of the above procedure augments G' with the path $u_0 \to u_1 \to u_4 \to u_8 \to u_{10} \to u_{11} \to u_{12}$. The resulting G' represents the portion of the placement in Fig. 3 that consists of the blocks B_1, B_4, B_8, B_{10} and B_{11} in their new locations. The second iteration augments G' with the path $u_8 \to u_9 \to u_{12}$, the third iteration with the path $u_1 \to u_5 \to u_8$, the fourth iteration with the path $u_0 \to u_2 \to u_5$, the fifth iteration with the path $u_5 \to u_7 \to u_{11}$, and the sixth (and final) iteration with the path $u_0 \to u_3 \to u_6 \to u_{11}$.

3.2. *Feasibility of the new adjacency graph*

In the following we show that the minimal average space as calculated in every iteration of the above procedure is nondecreasing. This will prove that the expression in (3) is always nonnegative. Otherwise, the assignment of arc lengths in Step 2 of the above procedure may yield negative arc lengths, which in turn will result in an illegal placement in which blocks overlap. Also, the balancing property which is proved later in Lemma 3.2, stems from the monotony of the length assigned to the arcs of G'.

Lemma 3.1. *The length assigned to the arcs of the new adjacency graph is nondecreasing.*

Proof. The proof proceeds inductively on the order of the augmentation of G'. Let Ω^n, $n = 1, 2, \ldots$, denote the path added to G' in the nth iteration of the construction procedure and let s^n be its corresponding average space (which is the length assigned to its arcs in G'). The average space s^1 calculated in Step 1 is nonnegative by definition. Let us first show that $s^2 \geq s^1$ by demonstrating that if this was not the case, then one could find a path in G from u_0 to u_{b+1} along which the average arc length is smaller than s^1. This will contradict the selection of Ω^1 as the path whose average arc length is minimal. From Step 2 of the procedure it follows that the end vertices u_i and u_j of Ω^2 must lie on Ω^1. Figure 5 illustrates the relation between Ω^1 and Ω^2. Let Ω_1^1, Ω_2^1 and Ω_3^1, be the portions of Ω^1 between the vertex pairs u_0 and u_i, u_i and u_j, and u_j and u_{b+1}, respectively. Let p_1, p_2 and p_3, be the average length of the arcs along Ω_1^1, Ω_2^1 and Ω_3^1, respectively, in G. Then, the length s^1 of every arc along Ω^1 in G' is given by:

$$s^1 = \frac{p_1|\Omega_1^1| + p_2|\Omega_2^1| + p_3|\Omega_3^1|}{|\Omega_1^1| + |\Omega_2^1| + |\Omega_3^1|}. \tag{4}$$

The average length of an arc along Ω^2 in G' is obtained from (3),

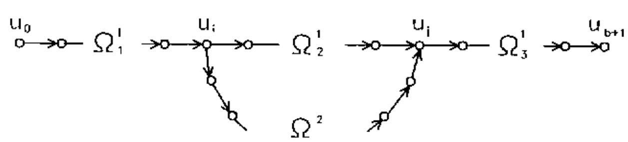

Fig. 5. Proof of Lemma 3.1: the first induction step.

$$s^2 = \frac{w_0 - l'(\Omega_1^1) - l'(\Omega_3^1) - w(\Omega^2) + w(u_i) + w(u_j)}{|\Omega^2|} \tag{5}$$

$$= \frac{w_0 - s^1|\Omega_1^1| - w(\Omega_1^1) - s^1|\Omega_3^1| - w(\Omega_3^1) - w(\Omega^2) + w(u_i) + w(u_j)}{|\Omega^2|}.$$

From the contradictory assumption that $s^2 < s^1$, and equations (4) and (5), we obtain after some algebraic operations

$$|\Omega_1^1| + |\Omega^2| + |\Omega_3^1| > (w_0 - w(\Omega_1^1) - w(\Omega^2) - w(\Omega_3^1) + w(u_i) + w(u_j)) \times \frac{|\Omega_1^1| + |\Omega_2^1| + |\Omega_3^1|}{p_1|\Omega_1^1| + p_2|\Omega_2^1| + p_3|\Omega_3^1|}. \tag{6}$$

The average length s of an arc along the path in G consisting of Ω_1^1, Ω^2 and Ω_3^1 is given by

$$s = \frac{w_0 - w(\Omega_1^1) - w(\Omega^2) - w(\Omega_3^1) + w(u_i) + w(u_j)}{|\Omega_1^1| + |\Omega^2| + |\Omega_3^1|}. \tag{7}$$

Substituting inequality (6) into (7) yields $s < s^1$ which contradicts the selection of Ω^1 among all the paths from u_0 to u_{b+1} as the one along which the average arc length is minimal.

Let $s^1 \le s^2 \le \cdots \le s^{r-1}$ and assume to the contrary that $s^r < s^{r-1}$. Let u_i^{r-1} and u_j^{r-1} be the end vertices of Ω^{r-1}, and let u_i^r and u_j^r be the end vertices of Ω^r. There are nine possibilities for the relation between Ω^{r-1} and Ω^r, three of which are illustrated in Fig. 6. Let us consider each one of them. Assume first that u_i^r and u_j^r do not lie on any path from u_0 to u_{b+1} containing Ω^{r-1}, as shown in Fig. 6(a). Then, Ω^r had to be selected prior to Ω^{r-1} in Step 2 of the iterative construction procedure, which is a contradiction. A second possibility is that u_i^r and u_j^r lie on Ω^{r-1} as shown in Fig. 6(b). Arguments similar to those used for the first induction step prove that such a situation is impossible. A third possibility is that u_i^r lies on some path from u_0 to u_i^{r-1} and that u_j^r lies on some path from u_j^{r-1} to u_{b+1}, as illustrated in Fig. 6(c). This however, results in a contradiction since Ω^{r-1} was selected as an unmarked path between two marked vertices that minimizes (3), when the vertices u_i^r and u_j^r were already marked. Therefore, there was another unmarked path between u_i^r and u_j^r (Ω^r) for which the ratio in (3) was smaller. The remaining six possibilities are combinations of the above three and similar arguments lead to contradictions. □

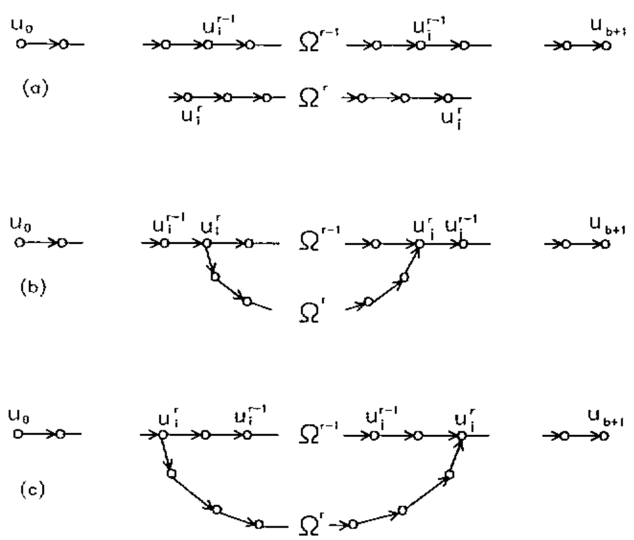

Fig. 6. Proof of Lemma 3.1: the general induction step.

From the construction procedure in Section 3.1 and from Lemma 3.1 we conclude that G' is a new horizontal adjacency graph isomorphic to the original G. We now prove that:

Lemma 3.2. *The horizontal adjacency graph G' represents a horizontally balanced placement.*

Proof. We have to show that for every vertex of G' (except u_0 and u_{b+1}) the length of the shortest entering arc equals the length of the shortest leaving arc. This follows immediately from two facts: First, whenever an unmarked vertex is added to G', one entering and one leaving arc of equal length are added too. Second, the series of arc lengths along the augmenting paths is monotonically nondecreasing as was proved in Lemma 3.1. Consequently, the equal left and right spaces determined when an unmarked vertex u is added to G' cannot be decreased by any later entering or leaving arc (cases where u can only be an end vertex of the augmenting path). □

We conclude with the following theorem:

Theorem 3.3 (existence). *Given an initial placement, its rectangles can always be horizontally displaced so that the resulting placement is legal, the horizontal adjacency relations are preserved and it is horizontally balanced.*

It occurs very often in VLSI layout that the location of some of the rectangles is predetermined so they are not movable. For example, the small rectangles along the top and the bottom boundaries of the layout in Fig. 7 are the I/O ports whose position is predetermined and cannot be changed. The above entities can be modeled as unmovable rectangles, and we say that the placement is balanced if only all its movable rectangles are balanced since we cannot require the fixed rectangles to be balanced too. The existence of some fixed rectangles does not restrict the validity of Theorem 3.3 and all the other results which follow. Let $R_{m1}, \ldots, R_{mk}$, be the unmovable rectangles. To model them we supplement G by a pair of arcs for every vertex u_{mi} corresponding to an unmovable rectangle R_{mi}, $1 \le i \le k$. One arc connects u_0 with u_{mi} and its length is equal to the distance of the left edge of R_{mi} from the left edge of R_0. The other arc connects u_{mi} with u_{b+1} and its length is defined similarly for the right edges. Then, in the initialization step of the construction procedure we add $u_{m1}, \ldots, u_{mk}$ and their associated arc pairs to G' and mark them in G, in addition to u_0 and u_{b+1}. The outcome of the construction procedure will be a configuration in which all the movable rectangles are balanced, while the unmovable ones remain in their initial location.

3.3. Uniqueness of the one dimensional balanced placement

Theorem 3.3 proves that for every given placement it is always possible to displace horizontally its rectangles to obtain a horizontally balanced placement. The question whether the horizontally balanced placement is unique is addressed in the following theorem.

Theorem 3.4 (uniqueness). *The horizontally balanced placement of a given initial placement is unique.*

Proof. Assume to the contrary that the balancing is not unique. Let G and H be two isomorphic horizontal adjacency graphs, representing two different horizontal balancings of the same initial placement. Let G be obtained by the construction procedure of Section 3.1. Consider the paths Ω^n and their corresponding arc lengths s^n, $n = 1, 2, \ldots$, in the same order as they were obtained by the construction procedure. Π^n denotes the path isomorphic to Ω^n in H. We next prove by induction on the order of Ω^n that the supposition of nonuniqueness leads to a contradiction. Recall that the lengths of all the paths from u_0 to u_{b+1} are equal to w_0 and that by definition the weights of isomorphic vertices are identical in G and H and equal to the width of the rectangle they represent.

Assume first that the arc lengths along Ω^1 are different from those along Π^1. There exist two possibilities:

(1) The arc lengths along Π^1 are not smaller than s^1, and there exists an arc f whose length in H is greater than s^1, namely,

$$s^H(e) \ge s^1, \quad \forall e \in \Pi^1; \qquad s^H(f) = p > s^1 = s^G(f). \tag{8}$$

The superscripts G and H are used to distinguish between spaces (and similarly,

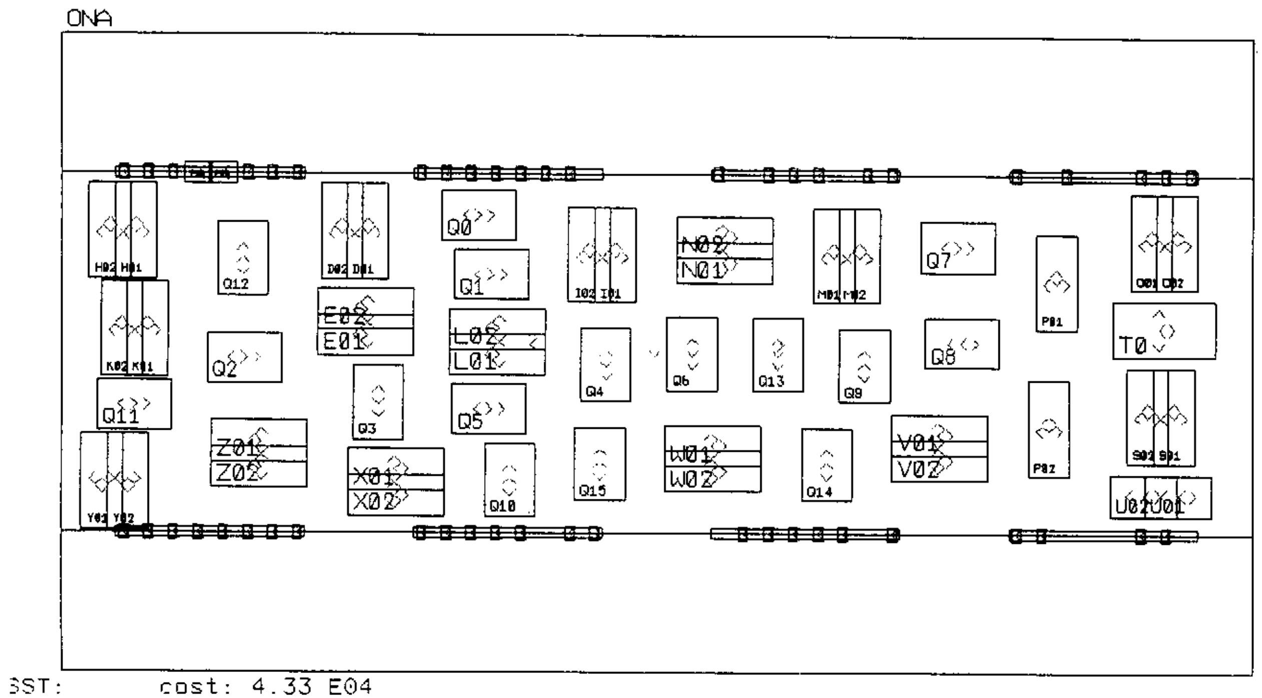

Fig. 7. Two dimensional VLSI placement with unmovable rectangles.

lengths and weights) in the G and H graphs. Calculating the length of Π^1, we obtain from (8)

$$\begin{aligned} w_0 &= l^H(\Pi^1) = s^H(\Pi^1) + w^H(\Pi^1) = s^H(\Pi^1) + w^G(\Omega^1) \\ &> s^G(\Omega^1) + w^G(\Omega^1) = l^G(\Omega^1) = w_0, \end{aligned} \tag{9}$$

which is impossible.

(2) There is an arc f along Π^1 satisfying $s^H(f) = p < s^1$. Let u_i and u_j be the end vertices of f. If $u_i \neq u_0$ then there exists an arc g entering u_i satisfying $s^H(g) \leq p$, since H represents a balanced placement. Applying this argument repetitively, we can find in H a path Π' from u_0 to u_i whose arc lengths do not exceed p. Similarly, if $u_j \neq u_{b+1}$ we can find in H a path Π'' from u_j to u_{b+1} whose arc lengths do not exceed p. All in all, we have found a path Π from u_0 to u_{b+1}, consisting of Π', f and Π'' along which the arc lengths are not greater than p and therefore, the average arc length along Π is also not greater than p. Since horizontal displacement of rectangles preserves the average arc length along any path from u_0 to u_{b+1}, the average arc length along any two isomorphic paths in G and H must be identical. This however contradicts s^1 being the minimal average arc length along any path from source to sink in the graph corresponding to the initial placement.

Assume now that Ω^n and Π^n, $1 \leq n \leq r-1$, have identical arc lengths, while Ω^r and Π^r have not. Again, there exist two possibilities:

(1) The arc lengths along Π^r are not smaller than s^r, and there exists an arc f whose length in H is greater than s^r, namely,

$$s^H(e) \geq s^r, \quad \forall e \in \Pi^r; \qquad s^H(f) = p > s^r = s^G(f). \tag{10}$$

According to the definition of Ω^r in the construction procedure, its end vertices u_i and u_j are lying on earlier paths and consequently, there exists a path Ω' from u_0 to u_i and a path Ω'' from u_j to u_{b+1} consisting of arcs belonging only to Ω^n, $1 \leq n \leq r-1$. Let Ω be the path from u_0 to u_{b+1} consisting of Ω', Ω^r and Ω'', and let Π, Π', Π^r, Π'' be their isomorphic paths in H, respectively. According to the induction hypothesis, there is:

$$l^H(\Pi') = l^G(\Omega'); \qquad l^H(\Pi'') = l^G(\Omega''). \tag{11}$$

Let us calculate the length of Π by combining (10) and (11).

$$\begin{aligned} w_0 &= l^H(\Pi) = l^H(\Pi') + l^H(\Pi^r) + l^H(\Pi'') - w^H(u_i) - w^H(u_j) \\ &= l^H(\Pi') + s^H(\Pi^r) + w^H(\Pi^r) + l^H(\Pi'') - w^H(u_i) - w^H(u_j) \\ &= l^G(\Omega') + s^H(\Pi^r) + w^G(\Omega^r) + l^G(\Omega'') - w^G(u_i) - w^G(u_j) \\ &> l^G(\Omega') + s^G(\Omega^r) + w^G(\Omega^r) + l^G(\Omega'') - w^G(u_i) - w^G(u_j) \end{aligned}$$

$$= l^G(\Omega') + l^G(\Omega^r) + l^G(\Omega'') - w^G(u_i) - w^G(u_j) = l^G(\Omega) = w_0, \tag{12}$$

which is a contradiction.

(2) There exists an arc f along Π^r satisfying $s^H(f) = p < s^r$. Since H represents a horizontally balanced placement, we can find (in the same manner as we did for the first induction step) a path Π in H whose arc lengths do not exceed p. Let Ω be the path in G isomorphic to Π. Divide the arcs along Π into two sets: E' contains the arcs belonging to Ω^n, $1 \le n \le r-1$, and E'' are the remaining arcs. According to the induction hypothesis and the definition of the paths Ω^n in the construction procedure, there is:

$$s^G(e) = s^H(e), \quad \forall e \in E'; \qquad s^G(e) \ge s^r > p \ge s^H(e), \quad \forall e \in E''. \tag{13}$$

Let us calculate the length of Π.

$$\begin{aligned} w_0 &= l^H(\Pi) = w^H(\Pi) + s^H(\Pi) = w^H(\Pi) + \sum_{e \in E'} s^H(e) + \sum_{e \in E''} s^H(e) \\ &\le w^G(\Omega) + \sum_{e \in E'} s^G(e) + p|E''| \\ &< w^G(\Omega) + \sum_{e \in E'} s^G(e) + \sum_{e \in E''} s^G(e) = l^G(\Omega) = w_0, \end{aligned} \tag{14}$$

which is a contradiction.

In conclusion, the contradiction originated from the assumption that the arc lengths along Π^r are not identical to those along Ω^r. □

4. Iterative algorithm for one dimensional space balancing

Given a placement, the construction procedure in Section 3.1 does not provide a practical way to find its corresponding horizontally balanced configuration. In the following we suggest an iterative algorithm which converges rapidly to the desired balanced placement and involves very simple calculations. Let q be the maximal number of vertices along a path in G (excluding u_0 and u_{b+1}). As shown below, the imbalance of any vertex after n iterations is bounded by $w_0 \gamma^n$, where w_0 is the width of B_0 and γ is a constant factor satisfying $\gamma \le 1 - (\frac{1}{2})^q$.

Given a placement, let us displace horizontally a rectangle R to the right in $\frac{1}{2}\mu$ distance if $\mu \ge 0$ and to the left in $\frac{1}{2}\mu$ distance if $\mu < 0$, where μ denotes the imbalance of R. We apply this displacement transformation to all rectangles one by one and call this procedure a *balancing cycle*. Without loss of generality assume that the rectangles are displaced in the order of their indices. Usually, a balancing cycle does not result in a balanced placement since a balanced rectangle R_i may become unbalanced when an adjacent rectangle R_j, $i<j$, is displaced. However, by applying the balancing cycle iteratively, the resulting placements converge to the (unique) balanced placement, as stated in the following theorem.

Theorem 4.1. *The series of placements resulting from the iterative application of balancing cycles converges to the balanced placement.*

Proof. Let μ_i^n denote the imbalance of R_i at the end of the nth balancing cycle, $1 \le i \le b$, $n = 0, 1, 2, \ldots$. Define

$$\mu^n = \max\{|\mu_i^n| \mid 1 \le i \le b\}. \tag{15}$$

We show next that there exists a real nonnegative number $0 \le \gamma \le 1 - (\frac{1}{2})^q$ such that

$$\mu^{n+1} \le \gamma \mu^n, \quad n = 0, 1, 2, \ldots. \tag{16}$$

If (16) is true then Theorem 4.1 is proved since $\mu^{n+1} \le \gamma^n \mu^0$, implying that the imbalance of each rectangle uniformly converges to zero.

To prove (16) recall that in the horizontal adjacency graph, the displacing of a rectangle equally shortens (lengthens) the length of every arc entering its corresponding vertex, and equally lengthens (shortens) the length of every leaving arc. Also, recall that during a balancing cycle the imbalance of every rectangle is reset to zero once, and later on in this cycle it may become unbalanced when its adjacent rectangles are balanced. In principle, the displacing of a rectangle R_i may affect only the imbalance of its adjacent rectangles, which in the worst case may increase by the magnitude of the displacement, i.e., by half of R_i's imbalance. Let R_j be adjacent to R_i. Then, the imbalance of R_j immediately after the balancing of R_i takes place, is increased by at most $\frac{1}{2}\mu^n$, i.e., its imbalance is bounded by $\mu^n + \frac{1}{2}\mu^n = 1\frac{1}{2}\mu^n$. Let the rectangle R_k be adjacent to R_j. Then, the imbalance of R_k immediately after the balancing of R_j takes place, is increased by at most $\frac{3}{4}\mu^n$, i.e., it is bounded by $\mu^n + \frac{1}{2}(\mu^n + \frac{1}{2}\mu^n) = 1\frac{3}{4}\mu^n$. The effect of balancing a rectangle on the remaining rectangles propagates along the paths passing through its corresponding vertex in the adjacency graph. Consequently, only those rectangles corresponding to vertices lying on paths passing through u_i (the vertex corresponding to R_i) may be affected by the displacement of R_i. Moreover, this effect is decreased in integral powers of $\frac{1}{2}$ with the arc distance from u_i.

When the imbalance of a rectangle R is considered, one entering and one leaving arc are determined (see equation (1)). Let q be the maximal number of vertices along a path from u_0 to u_{b+1} (excluding u_0 and u_{b+1}) and suppose that they are numbered $u_1, u_2, \ldots, u_q$. Then, the maximal number of balancing operations during a balancing cycle that may affect the imbalance of u_q is $q-1$. Therefore, the maximal quantity that can be added to the imbalance of u_q during cycle $n+1$ is

$$\mu^n(\tfrac{1}{2} + \tfrac{1}{4} + \cdots + (\tfrac{1}{2})^{q-1}) = \mu^n(1 - (\tfrac{1}{2})^{q-1}),$$

and the total imbalance of u_q prior to the $(n+1)$th displacement of its corresponding rectangle is bounded by $\mu^n(2 - (\frac{1}{2})^{q-1})$. Thus, after the imbalance of R_q was reset to zero in this cycle, the imbalance of u_{q-1} is bounded by $(1 - (\frac{1}{2})^q)\mu^n$. Setting $\gamma = (1 - (\frac{1}{2})^q)$, we get (16). $\square$

A direct consequence from the proof of Theorem 4.1 is:

Corollary 4.2. *The series of adjacency graphs resulting from the balancing cycles converges to the space balanced adjacency graph, independent of the order of balancing steps during a cycle (this order could vary from cycle to cycle), as long as each rectangle is balanced once in every cycle.*

In general, convergence is guaranteed for an arbitrary balancing sequence, as long as the period between two consecutive treatments of a rectangle is bounded. A simple, but illustrative, example is depicted in Fig. 8. There, the balancing during a cycle proceeded in the order of the rectangle indices. Notice that a faster convergence could be obtained if the order would be reversed.

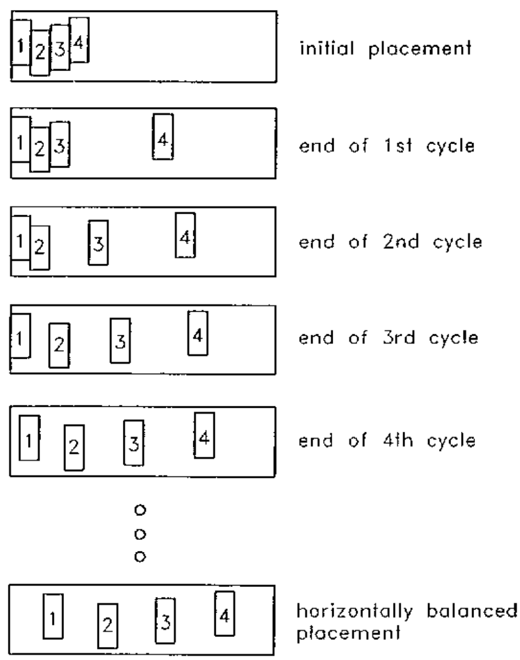

Fig. 8. An example illustrating the convergence of the balancing cycles.

5. Conclusions and further research

This paper addressed the block spacing problem whose objective is to provide enough room between the building blocks in VLSI layouts, so that the interconnecting wires can be routed successfully. We proposed a model for spreading the blocks uniformly over the chip area, to accommodate the routing requirements, while retaining their adjacency relations. The block spacing problem was solved via a weighted digraph model, on which a space balancing problem was defined. The existence and uniqueness of a solution to the *one dimensional* problem was proved, and an iterative algorithm which converges rapidly to the solution was presented.

Two alternatives for the solution of the dimensional space balancing problem were discussed. One is a byproduct of the existence proof, but as pointed out formerly, is impractical. The second solution is an efficient iterative algorithm which results in an infinite, but rapidly converging series. Still, we may look for a finite and efficient (polynomial) combinatorial solution to the space balancing problem and an algorithm for finding the path between two vertices along which the average arc length is minimized.

As we have already seen, the two dimensional space balancing problem may have no solution, but if the requirement to retain the isomorphism of the adjacency graphs is relaxed, solutions may exist (see Fig. 4). Since the two dimensional space balancing and the preservation of the isomorphism in both directions are sometimes conflicting requirements, we have in some instances to compromise. The question of how to trade off the conflicting requirements is a matter of further research.

Acknowledgement

Discussions with E. Solel from IBM Israel Scientific Center are gratefully acknowledged. The authors would like also to thank an anonymous reviewer for his helpful comments and suggestions.

References

[1] E.S. Kuh and M. Marek-Sadowska, Global routing, in: T. Ohtsuki, ed., Layout Design and Verification (North-Holland, Amsterdam, 1986) 169–198.

[2] T. Ohtsuki, Maze-running and line-search algorithms, in: T. Ohtsuki, ed., Layout Design and Verification (North-Holland, Amsterdam, 1986) 99–131.

[3] S. Wimer and I. Koren, Analysis of strategies for constructive general block placement, IEEE Trans. CAD of Integrated Circuits and Systems 7 (1988) 371–377.