# Nonrectangular Shaping and Sizing of Soft Modules for Floorplan-Design Improvement

Chris C. N. Chu and Evangeline F. Y. Young

***Abstract*—Many previous works on floorplanning with nonrectangular modules [1]–[12] assume that the modules are predesignated to have particular nonrectangular shapes, e.g., L-shaped, T-shaped, etc. However, this is not common in practice because rectangular shapes are more preferable in many designing steps. Those nonrectangular shapes are actually generated during floorplanning in order to further optimize the solution. In this paper, we study this problem of changing the shapes and dimensions of the flexible modules to fill up the unused area of a preliminary floorplan, while keeping the relative positions between the modules unchanged. This feature will also be useful in fixing small incremental changes during engineering change order modifications. We formulate the problem as a mathematical program. The formulation is such that the dimensions of all of the rectangular and nonrectangular modules can be computed by closed-form equations in $O(m)$ time in each corresponding Lagrangian relaxation subproblem (LRS) where $m$ is the total number of edges in the constraint graphs. As a result, the total time for the whole shaping and sizing process is $O(k \times m)$, where $k$ is the number of iterations on the LRS. Experimental results show that the amount of area reused is 3.7% on average, while the total wirelength can be reduced by 0.43% on average because of the more compacted result packing.**

***Index Terms*—Floorplanning, Lagrangian relaxation, physical design, rectilinear modules, shaping, sizing.**

## I. Introduction

A LOT OF previous works have reported on floorplanning with nonrectangular blocks [2]–[13]. The papers [6], [14] extend the Polish expression representation for slicing floorplans to handle L- and T-shaped modules. The works on nonslicing floorplans are mostly based on the bounded sliceline grid (BSG) structure [3], [4], [7]–[9] or the sequence pair (SP) representation [5], [10]–[12]. Most of these works explore the rules to restrict the placement of the rectangular subblocks of a rectilinear module, so that these subblocks will be placed adjacent to one another in an appropriate way to get back to its original rectilinear shape in the final packing. However, in all of these previous works, it is assumed that some modules are predesignated to have particular nonrectangular shapes, e.g., T-shaped or L-shaped, etc., but this is not common in practice since rectangular shapes are more preferable in many designing steps. They are easier to be managed not only in floorplanning, but also in downstream pin assignment, placement, routing and timing analysis. Nonrectangular shapes are often considered only when their shapes can improve the floorplan solution.

We can perform a preliminary floorplan design with all of the soft-blocks in rectangular shapes at the beginning. After a preliminary floorplan is obtained based on the important criteria like interconnect delay, routing congestion and area, etc., we can allow the flexible modules to change in shapes and dimensions slightly to further improve the floorplan solution as a postprocessing step, while keeping the relative spatial relationships between the modules unchanged. By keeping the adjacency and closeness relationship between the modules unchanged, the effect of this step on the original optimization in interconnect is little, while the area usage can be improved by allowing the flexible modules to change in shapes in the best fit way to fill up the unused area. The total interconnect length can usually be reduced by this postprocessing step because of the more compacted result packing. It is true that reusing all of the empty space may not be good sometimes because some empty space will be useful for buffer insertion or for other routing purposes. However, the selection of soft modules and empty space can be made by the users according to their needs and our method allows the users to further optimize a floorplan solution in a flexible way. This technique will also be useful in fixing small and incremental changes during engineering change order (ECO) modifications. For example, during the later stages of the design process, the implementations of some modules may have changed and new empty space is created. We can make use of this method to slightly modify the shapes of the soft modules in the surrounding of the new empty space to further optimize the design while maintaining the original relative positions between the modules unchanged.

In this paper, we formulate the problem as a mathematical program. Moh *et al.* [15] has also formulated the floorplanning problem as a geometric program and found the global minimum using some standard convex optimization techniques. Murata *et al.* [16] extend the work of [15] to nonslicing floorplan with soft and preplaced blocks, and try to reduce the number of variables and functions in the formulation to improve the efficiency. However, the execution time of their method to find an exact solution is still quite long, and they consider rectangular modules only. In our formulation, all of the flexible modules can change in dimension under their area and *aspect ratio* (width to height ratio) constraints. Those lying in the neighborhood of an empty space can change in shape (to become nonrectangular) to fill up the unused area in the best fit way. We use the Lagrangian relaxation technique [17], [18] to solve the problem. The formulation is such that the dimensions of all of the rectangular and nonrectangular modules can be computed by closed-form equations in $O(m)$ time in each of the corresponding Lagrangian

Manuscript received July 31, 2002; revised March 20, 2003. This paper was recommended by Associate Editor T. Yoshimura.

C. C. N. Chu is with the Department of Electrical and Computer Engineering, Iowa State University, Ames, IA 50011-3060 USA.

E. F. Y. Young is with the Department of Computer Science and Engineering, Chinese University of Hong Kong, Shatin, N.T., Hong Kong (e-mail: fyyoung@cse.cuhk.edu.hk).

Digital Object Identifier 10.1109/TCAD.2003.819896

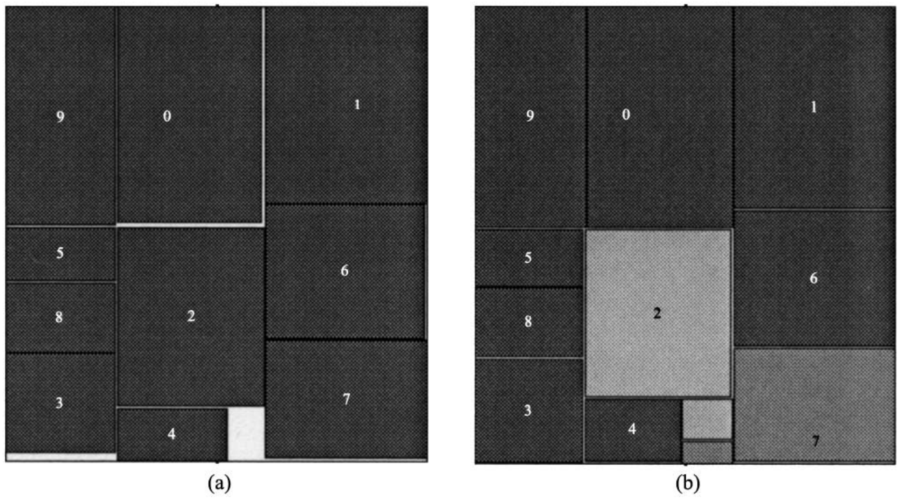



Fig. 1. Simple example of changing the shapes and dimensions of flexible modules to fill up empty space.

relaxation subproblem (LRS), where $m$ is the total number of edges in the constraint graphs. As a result, the total time for the whole shaping and sizing process is $O(k \times m)$, where $k$ is the number of iterations on the LRS.

We tested our method using some Microelectronics Center of North Carolina (MCNC) benchmarks. For each data set, a preliminary floorplan is first generated with an objective to optimize both the interconnect cost and total chip area. We then apply our mathematical programming technique to change the flexible modules in shapes and dimensions to fill up the empty space.

The rest of this paper is organized as follows. We will define the problem in the next section. Section III will give an overview of our approach and our formulation of the problem as a mathematical program. We will explain in detail of the Lagrangian relaxation technique and the optimality conditions to help solving the problem efficiently in Section IV. Experimental results will be shown in Section V, and remarks and the conclusion will be given in the last section.

## II. Problem Definition

In this problem, we are given a preliminary floorplan design, and our goal is to change the shapes and dimensions of some flexible modules to fill up the empty space, while keeping the module areas constant and the original spatial relationships between the modules unchanged. A simple example is shown in Fig. 1. In this example, the packing on the left is a given preliminary floorplan and our goal is to change the shapes and dimensions of the flexible modules to fill up the empty space. One possible final packing is shown on the right in which modules 2 and 7 become L-shaped after this postprocessing step. There are two kinds of input modules: *hard* and *soft* modules. A hard module is a module whose dimension is fixed. A soft module is one whose area is fixed, but its shape and dimensions can be changed as long as its aspect ratio (and the aspect ratios of its subblocks, if there is any) is within a given range. We are given $n$ modules of areas $A_1, A_2, \ldots, A_n$, their aspect ratio bounds $[r_1, s_1], [r_2, s_2], \ldots, [r_n, s_n]$ and their initial dimensions. (In case of a hard module, its minimum and maximum aspect ratio will be the same.) We are also given the netlist information: $\mathrm{net}_1, \mathrm{net}_2, \ldots, \mathrm{net}_m$ and the relative positions of the I/O pins $p_1, p_2, \ldots, p_q$ along the boundary of the chip. For each net $\mathrm{net}_i$ where $1 \leq i \leq m$, we are given its weight, the I/O pins, and the set of modules it is connected to.

A *packing* of a set of modules is a nonoverlap placement of the modules. We measure the area of a packing as the area of the smallest rectangle enclosing all of the modules. A *feasible packing* is a packing in which the widths and heights of all of the modules and their subblocks (if there is any) satisfy their aspect ratio constraints and their area constraints. For example, if a soft module $i$ is L-shaped, the dimensions of its two subblocks can be changed as long as their aspect ratios are within the given bounds and their total area is equal to $A_i$. A preliminary floorplan is given in the form of a pair of vertical and horizontal constraint graphs. Our objective is to change the shapes and dimensions of the soft modules to fill up the unused area, while keeping the relative positions between the modules as described by the constraint graphs unchanged. (Notice that if a module becomes nonrectangular in shape, its spatial relationship with the other modules will be measured with respect to its *main block*, i.e., its largest subblock.) The problem is defined formally as follows.

> **Problem Floorplanning With Shaping and Sizing (FP/SS)** Given a preliminary floorplan design in the form of a pair of horizontal and vertical constraint graphs, and a set of hard and soft modules with their initial dimensions and their area and aspect ratio constraints, change the shapes (from rectangular to nonrectangular) and dimensions of the soft modules to reduce the total area of the floorplan such that the relative positions between the modules (as described by the constraint graphs) are maintained and all of the area and aspect ratio constraints are satisfied.

## III. Overview of Our Approach

We are given a preliminary floorplan of a set $R$ of $n$ modules $M_1, M_2, \ldots, M_n$ with areas $A_1, A_2, \ldots, A_n$, respectively.

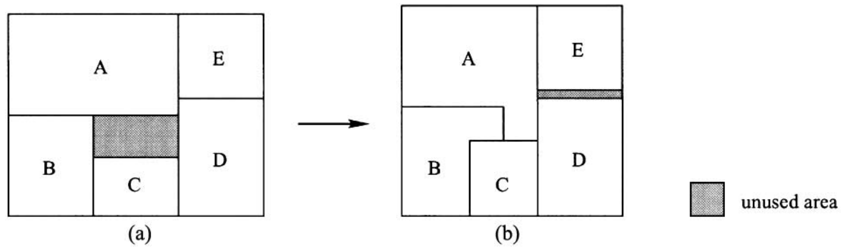



Fig. 2. Modules $A$ and $B$ are selected to be eligible to become nonrectangular in shape to fill up the empty space.

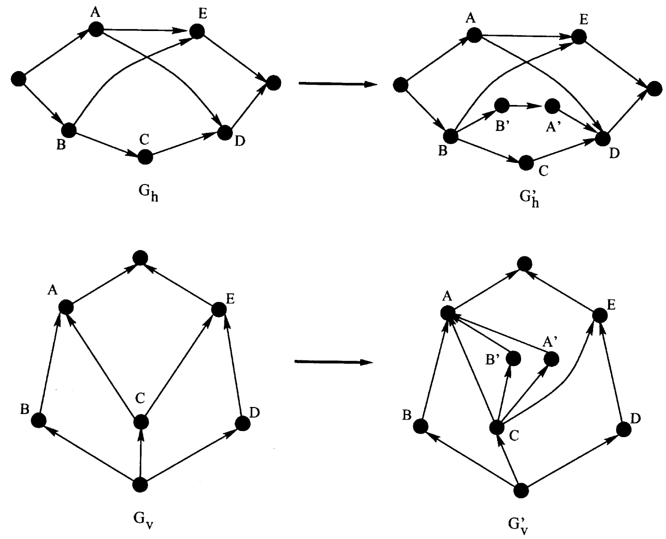



Fig. 3. Modify the constraint graphs to include the new subblocks and their associated edges.

For each module $M_i \in R$, the minimum and maximum aspect ratios are $r_i$ and $s_i$, respectively. The preliminary floorplan is given as a pair of constraint graphs $G_h$ and $G_v$, together with the initial dimensions of the modules. From this information, we can determine the packing, the positions of the unused area and the positions of the modules. We will then select some soft modules that lie in the neighborhood of some empty space into a set $S$. These selected modules are eligible to become nonrectangular in shape. An example is shown in Fig. 2. In this example, modules $A$ and $B$ are selected, and they can be changed to nonrectangular in shape to fill up the unused space [Fig. 2(b)]. Every module in this set $S$ will have one additional subblock of variable size. The constraint graphs $G_h$ and $G_v$ will also be updated (becoming $G'_h$ and $G'_v$) to include these new subblocks. New edges will be added to the constraint graphs to restrict the positions of these subblocks so that they will fill up the empty space and will abut with their corresponding main blocks. Fig. 3 shows the changes made to the constraint graphs for the example in Fig. 2. Notice that every selected module $M_k \in S$ will have one subblock $M'_k$, but the area of the subblock may become zero at the end and the selected module will then remain rectangular if this can optimize the design better.

After selecting a set of modules into $S$ and modifying the constraint graphs correspondingly, we can treat the subblocks as individual soft modules. (They will automatically abut with their main blocks because of the constraint edges added into the constraint graphs.) Let the size of $S$ be $p$, i.e., $p$ modules are selected to be possibly changed to nonrectangular in shape. Without loss of generality, we assume that module $M_1, M_2, \ldots, M_p$ are in $S$ and their corresponding subblocks are $M_{n+1}, M_{n+2}, \ldots, M_{n+p}$. Let $S'$ denote this set of subblocks. Now, we have a new set $R'$ of total $n' = n + p$ modules $M_1, M_2, \ldots, M_{n'}$. Consider the packing topology described by the constraint graphs $G'_h$ and $G'_v$. Let $x_i$ denote the smallest $x$ position of the lower left corner of module $M_i$ satisfying all of the horizontal constraints in the horizontal constraint graph $G'_h$. Similarly, $y_i$ denotes the smallest $y$ position of the lower left corner of module $M_i$ satisfying all of the vertical constraints in the vertical constraint graph $G'_v$. Then, for each edge $e(i, j)$ from $M_i$ to $M_j$ in $G'_h$, we have the following constraint:

$$x_i + w_i \leq x_j$$

where $w_i$ is the width of $M_i$. Similarly, for each edge $e(i, j)$ from $M_i$ to $M_j$ in $G'_v$, we have the following constraint:

$$y_i + h_i \leq y_j.$$

For each module $M_i \in R - S$, i.e., a rectangular module, the following relationship between $w_i$ and $h_i$ holds:

$$h_i = \frac{A_i}{w_i}.$$

For each module $M_i \in S$, i.e., a nonrectangular module, we have a constraint on the total area of $M_i$ and its subblock $M_{n+i}$

$$w_i h_i + w_{n+i} h_{n+i} = A_i.$$

In the horizontal constraint graph $G'_h$, we denote the set of sources and sinks by $s_h$ and $t_h$, respectively, where a source is a vertex without any incoming edge and a sink is a vertex without any outgoing edge. Similarly, we use $s_v$ and $t_v$ to denote the set of sources and sinks in $G'_v$, respectively. Then, for each $M_i$ in $s_h$ and $M_j$ in $s_v$

$$x_i = 0$$
$$y_j = 0.$$

For simplicity, we add one dummy vertex labeled $n' + 1$ to each of $G'_h$ and $G'_v$. The dummy vertices in $G'_h$ and $G'_v$ represent the right-most and top-most boundary of the chip, respectively. Edge $e(i, n' + 1)$ with weight $w_i$ is added to $G'_h$, for each $M_i \in t_h$ because the right-most chip boundary should be at a distance of at least $w_i$ from each module $M_i \in t_h$. Similarly, $e(i, n' + 1)$ with weight $h_i$ is added to $G'_v$ for each $M_i \in t_v$. From now on, we assume that the constraint graphs $G'_h$ and $G'_v$ contain these additional vertices and edges. The problem can be formulated as the following mathematical program *Primal Problem* (*PP*):

$$\begin{aligned}
\text{Minimize :}\quad & x_{n'+1}y_{n'+1} & \\
\text{Subject to :}\quad & x_i + w_i \le x_j \quad \forall e(i,j) \in G'_h & \text{(A)}\\
& y_i + h_i \le y_j \quad \forall e(i,j) \in G'_v & \text{(B)}\\
& w_i h_i = A_i \qquad \forall p+1 \le i \le n & \text{(C)}\\
& w_i h_i + w_{n+i}h_{n+i} = A_i \quad \forall 1 \le i \le p & \text{(D)}\\
& r_i h_i \le w_i \quad \forall 1 \le i \le n+p & \text{(E)}\\
& w_i \le s_i h_i \quad \forall 1 \le i \le n+p. & \text{(F)}
\end{aligned}$$

## IV. Solving the Problem by Lagrangian Relaxation

We will apply the Lagrangian relaxation technique [17] to solve the *PP*. Lagrangian relaxation is a general technique for solving constrained optimization problems. Constraints that are difficult to handle are "relaxed" and incorporated into the objective function by multiplying each constraint with a constant called Lagrange multipler. To solve the problem *PP*, we relax the constraints (A) and (B). Let $\lambda_{i,j}$ denote the multiplier for the constraint $x_i + w_i \le x_j$ in (A), and $\mu_{i,j}$ denotes the multiplier for the constraint $y_i + h_i \le y_j$ in (B). Let $\vec{\lambda}$ and $\vec{\mu}$ be vectors of all of the Lagrange multipliers introduced. Then, the LRS associated with the multipliers $\vec{\lambda}$ and $\vec{\mu}$, denoted by $LRS/(\vec{\lambda},\vec{\mu})$, becomes:

$$\begin{aligned}
\text{Minimize :}\quad & x_{n'+1}y_{n'+1} + \sum_{e(i,j)\in G'_h} \lambda_{i,j}(x_i + w_i - x_j)\\
& + \sum_{e(i,j)\in G'_v} \mu_{i,j}(y_i + h_i - y_j)\\
\text{Subject to :}\quad & w_i h_i = A_i \quad \forall p+1 \le i \le n\\
& w_i h_i + w_{n+i}h_{n+i} = A_i \quad \forall 1 \le i \le p\\
& r_i h_i \le w_i \quad \forall 1 \le i \le n+p\\
& w_i \le s_i h_i \quad \forall 1 \le i \le n+p.
\end{aligned}$$

Let $Q(\vec{\lambda},\vec{\mu})$ denote the optimal value of the problem $LRS/(\vec{\lambda},\vec{\mu})$. We define the Lagrangian dual problem *LDP* of *PP* as follows:

$$\begin{aligned}
\text{Minimize :}\quad & Q(\vec{\lambda},\vec{\mu})\\
\text{Subject to :}\quad & \vec{\lambda} \ge 0 \text{ and } \vec{\mu} \ge 0.
\end{aligned}$$

### A. Simplification of the LRS

$LRS/(\vec{\lambda},\vec{\mu})$ can be greatly simplified by the Kuhn-Tucker conditions [17], [18]. Consider the Lagrangian $\zeta$ of *PP* [17]

$$\begin{aligned}
\zeta = x_{n'+1}y_{n'+1} &+ \sum_{e(i,j)\in G'_h} \lambda_{i,j}(x_i + w_i - x_j)\\
&+ \sum_{e(i,j)\in G'_v} \mu_{i,j}(y_i + h_i - y_j)\\
&+ \text{terms independent of } x'_i s \text{ and } y'_i s.
\end{aligned}$$

The Kuhn-Tucker conditions imply that $\partial\zeta/\partial x_i = 0$ and $\partial\zeta/\partial y_i = 0$ for all $1 \le i \le n'+1$ at the optimal solution. Therefore, in searching for the multipliers to optimize *LDP*, we only need to consider those multipliers, such that $\partial\zeta/\partial x_i = 0$ and $\partial\zeta/\partial y_i = 0$ hold for all $1 \le i \le n'+1$. We obtain the following conditions by rearranging the terms in $\zeta$ and taking derivatives:

$$\sum_{e(j,i)\in G'_h} \lambda_{j,i} = \sum_{e(i,j)\in G'_h} \lambda_{i,j} \tag{1}$$

$$\sum_{e(j,i)\in G'_v} \mu_{j,i} = \sum_{e(i,j)\in G'_v} \mu_{i,j} \tag{2}$$

for all $1 \le i \le n'$, and

$$y_{n'+1} = \sum_{e(i,n'+1)\in G'_h} \lambda_{i,n'+1} \tag{3}$$

$$x_{n'+1} = \sum_{e(i,n'+1)\in G'_v} \mu_{i,n'+1}. \tag{4}$$

We use $\Omega$ to denote the set of $(\vec{\lambda},\vec{\mu})$ satisfying the above relationships (1)–(4) for the given pair of horizontal and vertical constraint graphs $G'_h$ and $G'_v$. When $(\vec{\lambda},\vec{\mu}) \in \Omega$, the objective function $F$ of $LRS/(\vec{\lambda},\vec{\mu})$ becomes

$$F = k + \sum_{1\le i\le n+p}\left(\left(\sum_{e(i,j)\in G'_h}\lambda_{i,j}\right)w_i + \left(\sum_{e(i,j)\in G'_v}\mu_{i,j}\right)h_i\right)$$

where $k = -(\sum_{e(i,n'+1)\in G'_h}\lambda_{i,n'+1})(\sum_{e(i,n'+1)\in G'_v}\mu_{i,n'+1})$ is a constant for a fixed pair of $\vec{\lambda}$ and $\vec{\mu}$. Let $\lambda_i = \sum_{e(i,j)\in G'_h}\lambda_{i,j}$ and $\mu_i = \sum_{e(i,j)\in G'_v}\mu_{i,j}$, for $1 \le i \le n+p$. Then $LRS/(\vec{\lambda},\vec{\mu})$ can be simplified to

$$\begin{aligned}
\text{Minimize :}\quad & \sum_{1\le i\le n+p}(\lambda_i w_i + \mu_i h_i)\\
\text{Subject to :}\quad & w_i h_i = A_i \quad \forall p+1 \le i \le n\\
& w_i h_i + w_{n+i}h_{n+i} = A_i \quad \forall 1 \le i \le p\\
& r_i h_i \le w_i \quad \forall 1 \le i \le n+p\\
& w_i \le s_i h_i \quad \forall 1 \le i \le n+p.
\end{aligned}$$

To solve this simplified LRS, we first write down its Lagrangian $\xi$

$$\begin{aligned}
\xi = \sum_{1\le i\le n+p}(\lambda_i w_i + \mu_i h_i) &+ \sum_{p+1\le i\le n}\theta_i(w_i h_i - A_i)\\
&+ \sum_{1\le i\le p}\sigma_i(w_i h_i + w_{n+i}h_{n+i} - A_i)\\
&+ \sum_{1\le i\le n+p}\alpha_i(r_i h_i - w_i) + \sum_{1\le i\le n+p}\beta_i(w_i - s_i h_i)
\end{aligned}$$

where $\theta_i$, $\sigma_i$, $\alpha_i$, $\beta_i \in \Re$, $\alpha_i \ge 0$, and $\beta_i \ge 0$ denote the Lagrangian multipliers for the constraints in (C)–(F), respectively. According to the Kuhn-Tucker conditions [17], the first-order optimality conditions for $LRS/(\vec{\lambda},\vec{\mu})$ are as follows:

$$\frac{\partial\xi}{\partial w_i} = 0 \quad \text{for all } 1 \le i \le n+p \tag{5}$$

$$\frac{\partial\xi}{\partial h_i} = 0 \quad \text{for all } 1 \le i \le n+p \tag{6}$$

$$\alpha_i(r_i h_i - w_i) = 0 \quad \text{for all } 1 \le i \le n+p \tag{7}$$

$$\beta_i(w_i - s_i h_i) = 0 \quad \text{for all } 1 \le i \le n+p \tag{8}$$

$$w_i h_i = A_i \quad \text{for all } p+1 \le i \le n \tag{9}$$

$$w_i h_i + w_{n+i}h_{n+i} = A_i \quad \text{for all } 1 \le i \le p. \tag{10}$$

### B. Solutions for Rectangular Blocks

Consider a module $M_i$ where $p+1 \leq i \leq n$, i.e., a rectangular module. Conditions (5)-(10) can be written as

$$\lambda_i + \theta_i h_i - \alpha_i + \beta_i = 0 \quad (11)$$

$$\mu_i + \theta_i w_i + r_i \alpha_i - s_i \beta_i = 0 \quad (12)$$

$$\alpha_i (r_i h_i - w_i) = 0 \quad (13)$$

$$\beta_i (w_i - s_i h_i) = 0 \quad (14)$$

$$w_i h_i = A_i. \quad (15)$$

There are three cases for the values of $h_i$ and $w_i$, according to the values of $\alpha_i$ and $\beta_i$.

Case 1) $\alpha_i = 0$ and $\beta_i = 0$. From (11), $h_i = -\lambda_i/\theta_i$. From (12), $w_i = -\mu_i/\theta_i$. Eliminating $\theta_i$, $h_i = w_i \lambda_i/\mu_i$. Substituting $h_i$ into (15), $w_i^2 \lambda_i/\mu_i = A_i$. Therefore, $w_i = \sqrt{A_i \mu_i/\lambda_i}$.

Case 2) $\alpha_i \neq 0$ and $\beta_i = 0$. Equation (13) implies that $w_i = r_i h_i$. Substituting $h_i$ into (15), $w_i = \sqrt{A_i r_i}$.

Case 3) $\alpha_i = 0$ and $\beta_i \neq 0$. Equation (14) implies that $w_i = s_i h_i$. Substituting $h_i$ into (15), $w_i = \sqrt{A_i s_i}$.

Note that $\alpha_i \neq 0$ and $\beta_i \neq 0$ is impossible since (13) and (14) cannot be satisfied simultaneously.

By combining the three cases, it is not difficult to see that

$$w_i = \min\left\{\sqrt{A_i s_i}, \max\left\{\sqrt{A_i r_i}, \sqrt{\frac{A_i \mu_i}{\lambda_i}}\right\}\right\}.$$

Once $w_i$ is found, $h_i$ is given by $A_i/w_i$.

### C. Solutions for Nonrectangular Modules

Consider a module $M_i$ where $1 \leq i \leq p$, i.e., a module that can possibly become nonrectangular in shape. Conditions (5)-(10) can be written as

$$\lambda_i + \sigma_i h_i - \alpha_i + \beta_i = 0 \quad (16)$$

$$\mu_i + \sigma_i w_i + r_i \alpha_i - s_i \beta_i = 0 \quad (17)$$

$$\lambda_{n+i} + \sigma_i h_{n+i} - \alpha_{n+i} + \beta_{n+i} = 0 \quad (18)$$

$$\mu_{n+i} + \sigma_i w_{n+i} + r_{n+i} \alpha_{n+i} - s_{n+i} \beta_{n+i} = 0 \quad (19)$$

$$\alpha_i (r_i h_i - w_i) = 0 \quad (20)$$

$$\beta_i (w_i - s_i h_i) = 0 \quad (21)$$

$$\alpha_{n+i} (r_{n+i} h_{n+i} - w_{n+i}) = 0 \quad (22)$$

$$\beta_{n+i} (w_{n+i} - s_{n+i} h_{n+i}) = 0 \quad (23)$$

$$w_i h_i + w_{n+i} h_{n+i} = A_i. \quad (24)$$

For a given pair of $\vec{\lambda}$ and $\vec{\mu}$, $\lambda_i, \mu_i, \lambda_{n+i}$, and $\mu_{n+i}$ are known. Therefore, we need to solve a system $\Gamma$ of nine nonlinear equations with nine unknowns ($h_i, w_i, h_{n+i}, w_{n+i}, \sigma_i, \alpha_i, \beta_i, \alpha_{n+i}$, and $\beta_{n+i}$). Fortunately, they can be solved by closed-form equations as described in the following.

There are three cases for the values of $h_i$ and $w_i$ according to the values of $\alpha_i$ and $\beta_i$.

Case 1) $\alpha_i = 0$ and $\beta_i = 0$. This case occurs when $r_i \leq (\mu_i/\lambda_i) \leq s_i$. From (16) and (17)

$$h_i = \frac{-\lambda_i}{\sigma_i} \quad w_i = \frac{-\mu_i}{\sigma_i}.$$

Case 2) $\alpha_i \neq 0$ and $\beta_i = 0$. This case occurs when $\mu_i/\lambda_i \leq r_i$. From (16), (17), and (20)

$$h_i = \frac{-\lambda_i r_i - \mu_i}{2\sigma_i r_i} \quad w_i = \frac{-\lambda_i r_i - \mu_i}{2\sigma_i}.$$

Case 3) $\alpha_i = 0$ and $\beta_i \neq 0$. This case occurs when $\mu_i/\lambda_i \geq s_i$. From (16), (17), and (21)

$$h_i = \frac{-\lambda_i s_i - \mu_i}{2\sigma_i s_i} \quad w_i = \frac{-\lambda_i s_i - \mu_i}{2\sigma_i}.$$

Note that $\alpha_i \neq 0$ and $\beta_i \neq 0$ is impossible since (20) and (21) cannot be satisfied simultaneously. Similarly, we can write $w_{n+i}$ and $h_{n+i}$ in terms of $\lambda_{n+i}, \mu_{n+i}, r_{n+i}, s_{n+i}$, and $\sigma_i$ according to the values of $\alpha_{n+i}$ and $\beta_{n+i}$.

Case 1) $\alpha_{n+i} = 0$ and $\beta_{n+i} = 0$. This case occurs when $r_{n+i} \leq (\mu_{n+i}/\lambda_{n+i}) \leq s_{n+i}$. From (18) and (19)

$$h_{n+i} = \frac{-\lambda_{n+i}}{\sigma_i} \quad w_{n+i} = \frac{-\mu_{n+i}}{\sigma_i}.$$

Case 2) $\alpha_{n+i} \neq 0$ and $\beta_{n+i} = 0$. This case occurs when $\mu_{n+i}/\lambda_{n+i} \leq r_{n+i}$. From (18), (19), and (22).

$$h_{n+i} = \frac{-\lambda_{n+i} r_{n+i} - \mu_{n+i}}{2\sigma_i r_{n+i}} \quad w_{n+i} = \frac{-\lambda_{n+i} r_{n+i} - \mu_{n+i}}{2\sigma_i}.$$

Case 3) $\alpha_{n+i} = 0$ and $\beta_{n+i} \neq 0$. This case occurs when $\mu_{n+i}/\lambda_{n+i} \geq s_{n+i}$. From (18), (19), and (23)

$$h_{n+i} = \frac{-\lambda_{n+i} s_{n+i} - \mu_{n+i}}{2\sigma_i s_{n+i}} \quad w_{n+i} = \frac{-\lambda_{n+i} s_{n+i} - \mu_{n+i}}{2\sigma_i}.$$

Similarly, $\alpha_{n+i} \neq 0$ and $\beta_{n+i} \neq 0$ is impossible, since (22) and (23) cannot be satisfied simultaneously. Therefore, in any combination of the above cases, we can write $h_i$, $w_i$, $h_{n+i}$, and $w_{n+i}$ in terms of $\sigma_i$. (Note that $\lambda_i, \lambda_{n+i}, \mu_i, \mu_{n+i}, r_i, s_i, r_{n+i}$, and $s_{n+i}$ are known.) We can substitute these expressions into (24) and solve $\sigma_i$. Finally, we will substitute back the value of $\sigma_i$ into the expressions for $h_i$, $w_i$, $h_{n+i}$, and $w_{n+i}$ and compute their values.

### D. Solving LRS

The algorithm *LRS* below outlines the steps to solve the LRS $LRS/(\vec{\lambda}, \vec{\mu})$ given a pair of $\vec{\lambda}$ and $\vec{\mu}$ satisfying the optimality condition (1)–(4).

```
Algorithm LRS
  /* This algorithm solves LRS/(λ⃗, μ⃗) given
a pair of (λ⃗, μ⃗) ∈ Ω */
    Input: Areas A_1, A_2, ..., A_n
      Lower bounds on aspect ratios
      r_1, r_2, ..., r_n'
      Upper bounds of aspect ratios
      s_1, s_2, ..., s_n'
      Constraint graphs G'_v and G'_h
      Lagrange multipliers (λ⃗, μ⃗) ∈ Ω
    Output: w_1, w_2, ..., w_n', h_1, h_2, ..., h_n'
      1) For i = p + 1 to n
      2)  Compute L_i = √(A_i r_i) and U_i = √(A_i s_i)
      3)  Compute λ_i = Σ_{e(i,j)∈G'_h} λ_{i,j}
      4)  Compute μ_i = Σ_{e(i,j)∈G'_v} μ_{i,j}
      5)  If (λ_i ≠ 0) and (μ_i/λ_i ≥ 0)
```

6) *Compute* $w^* = \sqrt{(A_i \mu_i)/\lambda_i}$
7) $w_i = \min\{U_i, \max\{L_i, w^*\}\}$, $h_i = A_i/w_i$
8) *For* $i = 1$ *to* $p$
9) *Compute* $\lambda_i = \sum_{e(i,j) \in G'_h} \lambda_{i,j}$
10) *Compute* $\mu_i = \sum_{e(i,j) \in G'_v} \mu_{i,j}$
11) *Compute* $\lambda_{n+i} = \sum_{e(n+i,j) \in G'_h} \lambda_{n+i,j}$
12) *Compute* $\mu_{n+i} = \sum_{e(n+i,j) \in G'_v} \mu_{n+i,j}$
13) *Compute* $\sigma_i$, $h_i$, $w_i$, $h_{n+i}$ and $w_{n+i}$ from the values of $\lambda_i$, $\mu_i$, $\lambda_{n+i}$ and $\mu_{n+i}$ according to the cases described in Section IV-C.

### E. Solving LDP

As explained above, we will only consider those $(\vec{\lambda}, \vec{\mu}) \in \Omega$ in the *LDP* problem. We used a subgradient optimization method to search for these pairs of $\vec{\lambda}$ and $\vec{\mu}$. Starting from an arbitrary $(\vec{\lambda}, \vec{\mu}) \in \Omega$ in step $k$, we will move to a new pair $(\vec{\lambda'}, \vec{\mu'})$ by following the subgradient direction:

$$\lambda'_{i,j} = [\lambda_{i,j} + \rho_k(x_i + w_i - x_j)]^+$$
$$\mu'_{i,j} = [\mu_{i,j} + \rho_k(y_i + h_i - y_j)]^+$$

where

$$[x]^+ = \begin{cases} x, & \text{if } x > 0 \\ 0, & \text{if } x \le 0 \end{cases}$$

and $\rho_k$ is a step size such that $\lim_{k \to \infty} \rho_k = 0$ and $\sum_{k=1}^{\infty} \rho_k = \infty$. After updating $\vec{\lambda}$ and $\vec{\mu}$, we will project $(\vec{\lambda'}, \vec{\mu'})$ back to the nearest point $(\vec{\lambda^*}, \vec{\mu^*})$ in $\Omega$ using a 2-norm measure and solve the LRS $LRS/(\vec{\lambda^*}, \vec{\mu^*})$ again using the algorithm *LRS*. These steps are repeated until the solution converges. The following algorithm summarizes the steps to solve the *LDP* problem.

Algorithm LDP
/* This algorithm solves the *LDP* problem optimally. */
Input:
Areas $A_1, A_2, \ldots, A_n$
Lower bounds on aspect ratios $r_1, r_2, \ldots, r_{n'}$
Upper bounds on aspect ratios $s_1, s_2, \ldots, s_{n'}$
Constraint graphs $G'_v$ and $G'_h$
Output: $w_1, w_2, \ldots, w_{n'}$, $h_1, h_2, \ldots, h_{n'}$
1) Initialize $(\vec{\lambda}, \vec{\mu})$ *and* $\rho_1$
2) Project $(\vec{\lambda}, \vec{\mu})$ to $(\vec{\lambda^*}, \vec{\mu^*})$ such that $(\vec{\lambda^*}, \vec{\mu^*}) \in \Omega$
3) $k = 1$
4) $(\vec{\lambda}, \vec{\mu}) = (\vec{\lambda^*}, \vec{\mu^*})$
5) Repeat
6) Call $LRS()$ with $(\vec{\lambda}, \vec{\mu})$
7) Compute $(x_i, y_i)$ $\forall 1 \le i \le n' + 1$ from $G'_v$ and $G'_h$ using the longest path algorithm
8) Compute $\lambda'_{i,j} = [\lambda_{i,j} + \rho_k(x_i + w_i - x_j)]^+$ $\forall e(i,j) \in G'_h$
9) Compute $\mu'_{i,j} = [\mu_{i,j} + \rho_k(y_i + h_i - y_j)]^+$ $\forall e(i,j) \in G'_v$
10) Project $(\vec{\lambda'}, \vec{\mu'})$ to $(\vec{\lambda^*}, \vec{\mu^*})$ s.t. $(\vec{\lambda^*}, \vec{\mu^*}) \in \Omega$
11) $k = k + 1$
12) $(\vec{\lambda}, \vec{\mu}) = (\vec{\lambda^*}, \vec{\mu^*})$
13) Until $h_i$ and $w_i$ $\forall 1 \le i \le n' + 1$ converge.

### F. Time Complexity

In each iteration of *LRS*, we need to look at $\lambda_{i,j}$ and $\mu_{i,j}$ a constant number of times for each edge $e(i,j)$ in $G'_h$ and $G'_v$. Therefore, the runtime for each *LRS* is $O(m)$, where $m$ is total number of edges in the constraint graphs. Let $k$ be the number of iterations on *LRS* invoked by the *LDP* algorithm. The total runtime of the whole process is $O(k \times m)$.

## V. Experimental Results

We tested our method using the MCNC benchmarks. The size of each benchmark data is shown in Table I. In each experiment, we first generated a preliminary floorplan using a simulated annealing method. In this initial floorplanning step, equal weighting were given to the area term and the wirelength term in the cost function, where the wirelength is computed using the half-perimeter estimation method assuming that the pins are located at the center of the module. After obtaining a preliminary floorplan, we selected some soft modules lying in the neighborhood of some large empty space into the set $S$. These were the modules that would possibly become nonrectangular in shape to fill up the space. In our current implementation, we would select those modules which were also lying on the critical paths of the constraint graphs. The constraint graphs were then modified to include the new subblocks of the selected modules and to restrict their positions so that they would fill up the empty space and abut with their corresponding main blocks. After these preprocessing steps, we performed shaping and sizing on the modules using the Lagrangian relaxation technique as described in Section IV.

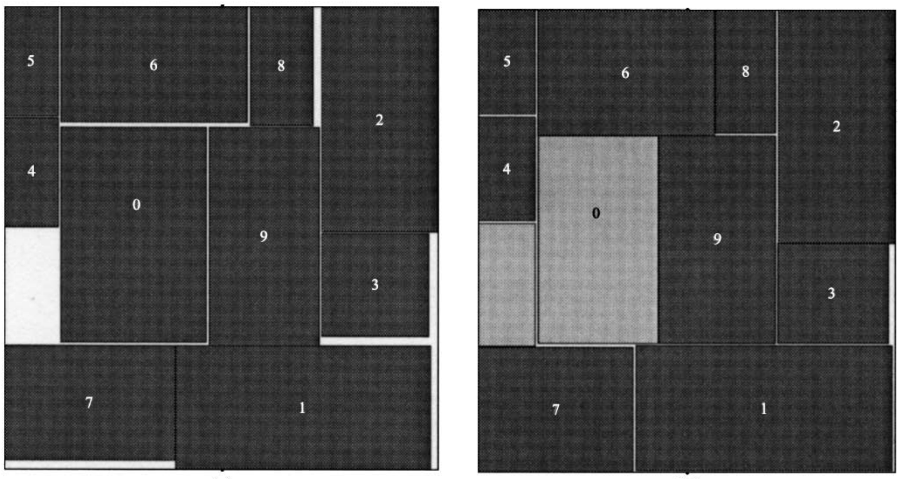
In all of the experiments, the minimum and maximum aspect ratios of the soft modules were 0.5 and 2.0, respectively, while those for the subblocks were 0.1 and 10.0, respectively. We lim-

TABLE I
TESTING DATA SETS

| Data Set | Number of Modules | Number of Nets |
|---|---|---|
| xerox | 10 | 203 |
| hp | 11 | 83 |
| ami33 | 33 | 123 |
| ami49 | 49 | 408 |

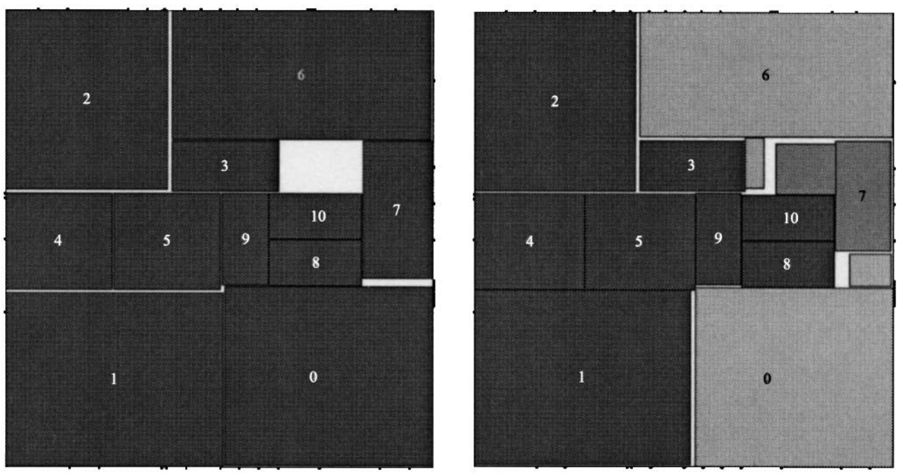
TABLE II
SHAPING AND SIZING RESULTS

| Benchmark | Deadspace in preliminary floorplan (%) | Area re-used (%) | Change in total wirelength (%) | Time (sec) |
|---|---|---|---|---|
| xerox | 3.51 | 3.01 | -0.87 | 0.16 |
| hp | 3.87 | 2.55 | -0.88 | 0.17 |
| ami33 | 7.41 | 5.01 | -0.97 | 1.44 |
| ami49 | 9.07 | 4.08 | +1.00 | 4.88 |



(a) (b)

Fig. 4. Data set xerox (3). (a) Preliminary floorplan. $\mathrm{Deadspace} = 6.38\%$. (b) After sizing and shaping. $\mathrm{Deadspace} = 0.60\%$.



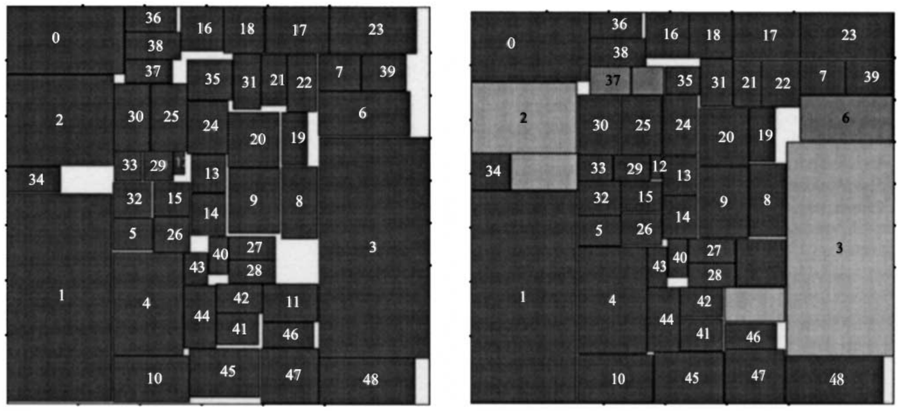
(a) (b)

Fig. 5. Data set hp (1). (a) Preliminary floorplan. Deadspace $= 2.80\%$. (b) After sizing and shaping. Deadspace $= 1.50\%$.

(a) (b)

Fig. 6. Data set ami33 (3). (a) Preliminary floorplan. Deadspace $= 5.21\%$. (b) After sizing and shaping. Deadspace $= 2.40\%$.

ited the aspect ratio of the final packing to the range of 0.9 to 1.1. All of the results were generated using a 600 MHz Pentium III processor, and were shown in Table II. Notice that each result is obtained by taking the average of performing the experiment five times. Experimental results show that our shaping and sizing technique is useful in reusing empty space by changing some soft modules to nonrectangular in shape, while keeping the relative positions between the modules unchanged. We can



(a) (b)

Fig. 7. Data set ami49 (1). (a) Preliminary floorplan. Deadspace $= 6.48\%$. (b) After sizing and shaping. Deadspace $= 1.72\%$.

(a) (b)

Fig. 8. Data set ami33 with hard modules and boundary constraints. (a) Preliminary floorplan. Deadspace $= 8.99\%$. Module 3, 15, 23 and 25 are hard blocks with boundary constraint. (b) After sizing and shaping. Deadspace $= 5.77\%$.

see that the total wirelength is reduced on average because of the more compacted result packing. Notice that after shaping and sizing, the pins of an L-shaped module are assumed to be located at the center of the main subblock. Figs. 4–7 show the preliminary floorplans and the floorplans after shaping and sizing for some of the experiments. Fig. 8 shows an example in which modules 3, 15, 23, and 25 are hard modules with boundary constraint. They are, thus, placed along the boundary in the preliminary floorplan. During the shaping and sizing process, the dimensions of the hard modules remain unchanged and those soft modules lying close to the boundary can be changed to rectilinear shape to fill up the empty space formed between the hard modules.

## VI. Remarks

In this paper, we only handle the case when the nonrectangular modules have, at most, two rectangular subblocks. However, our approach can be extended to more than two subblocks directly. If each nonrectangular module has up to $k$ rectangular subblocks, the system of equations $\Gamma$ will have $4k+1$ unknowns in $4k+1$ nonlinear equations, and can still be solved by closed-form equations by considering three possible cases for the size of each rectangular subblock.

## VII. Conclusion

We presented an efficient method to postprocess a floorplan solution to further optimize its area usage by changing some soft modules to nonrectangular in shape to fill up the empty space. The total wirelength can also be reduced because of the more compacted result packing. This technique will also be useful in fixing small incremental changes during ECO modifications. Our approach is based on an elegant closed-form solution to a mathematical program using the Lagrangian relaxation technique. Experimental results on the MCNC benchmarks have demonstrated its effectiveness in postprocessing a floorplan solution in a very flexible way.

## References


[1] T. C. Lee, "A bounded 2-D contour searching algorithm for floorplan design with arbitrarily shaped rectilinear and soft modules," in *Proc. 30th ACM/IEEE Design Automation Conf.*, 1993, pp. 525–530.

[2] D. F. Wong and C. L. Liu, "A new algorithm for floorplan design," in *Proc. 23rd ACM/IEEE Design Automation Conf.*, 1986, pp. 101–107.

[3] M. Kang and W. W. M. Dai, "General floorplanning with L-shaped, T-shaped, and soft blocks based on bounded slicing grid structure," in *Proc. IEEE Asia South Pacific Design Automation Conf.*, 1997, pp. 265–270.

[4] M. Z. W. Kang and W. W. M. Dai, "Topology rectilinear block packing," in *Proc. Int. Symp. Physical Design*, 1998, pp. 179–186.

[5] M. Z. Kang and W. W.-M. Dai, “Arbitrary rectilinear block packing based on sequence pair,” in *Proc. IEEE Int. Conf. Computer-Aided Design*, 1998, pp. 259–266.

[6] D. F. Wong, F. Y. Young, and H. H. Yang, “On extending slicing floorplans to handle L/T-shaped modules and abutment constraints,” *IEEE Trans. Computer-Aided Design*, vol. 20, pp. 800–807, June 2001.

[7] S. Nakatake, K. Fujiyoushi, H. Murata, and Y. Kajitani, “Module placement on BSG-structure and IC layout applications,” in *Proc. IEEE Int. Conf. Computer-Aided Design*, 1996, pp. 484–491.

[8] S. Nakatake, M. Furuya, and Y. Kajitani, “Module placement on BSG-structure with preplaced modules and rectilinear modules,” in *Proc. IEEE Asia South Pacific Design Automation Conf.*, 1998, pp. 571–576.

[9] K. Sakanushi, S. Nakatake, and Y. Kajitani, “The multi-BSG: Stochastic approach to an optimal packing of convex-rectilinear blocks,” in *Proc. IEEE Int. Conf. Computer-Aided Design*, 1998, pp. 267–274.

[10] J. Dufour, R. McBride, P. Zhang, and C. K. Cheng, “A building block placement tool,” in *Proc. IEEE Asia South Pacific Design Automation Conf.*, 1997, pp. 271–276.

[11] J. Xu, P.-N. Guo, and C.-K. Cheng, “Rectilinear block placement using sequence-pair,” in *Proc. Int. Symp. Physical Design*, 1998, pp. 173–178.

[12] K. Fujiyoshi and H. Murata, “Arbitrary convex and concave rectilinear block packing using sequence-pair,” in *Proc. Int. Symp. Physical Design*, 1999, pp. 103–110.

[13] C.-H. Lee, Y.-C. Lin, W.-Y. Fu, C.-C. Chang, and T.-M. Hsieh, “A new formulation for SOC floorplan area minimization problem,” in *Proc. Design, Automation, Test Eur. Conf. Exhibit.*, 2002, p. 1100.

[14] D. F. Wong and C. L. Liu, “Floorplan design for rectangular and L-shaped modules,” in *Proc. IEEE Int. Conf. Computer-Aided Design*, 1987, pp. 520–523.

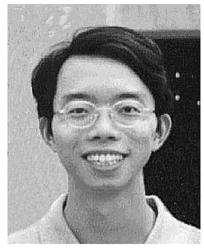
[15] T.-S. Moh, T.-S. Chang, and S. L. Hakimi, “Globally optimal floorplanning for a layout problem,” *IEEE Trans. Circuit Syst. I*, vol. 43, pp. 713–720, Sept. 1996.

[16] H. Murata and E. S. Kuh, “Sequence-pair-based placement method for hard/soft/pre-placed modules,” in *Proc. Int. Symp. Physical Design*, 1998, pp. 167–172.

[17] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, *Nonlinear Programming: Theory and Algorithms*, 2nd ed. New York: Wiley, 1997.

[18] F. Y. Young, C. C. N. Chu, W. S. Luk, and Y. C. Wong, “Floorplan area minimization using Lagrangian relaxation,” in *Proc. Int. Symp. Physical Design*, 2000, pp. 174–179.

[19] R. J. Duffin, E. L. Peterson, and C. Zener, *Geometric Programming—Theory and Application*, 1st ed. New York: Wiley, 1967.



**Chris C. N. Chu** received the B.S. degree in computer science from the University of Hong Kong, Hong Kong, in 1993, and the M.S. and Ph.D. degrees in computer science from the University of Texas, Austin, in 1994 and 1999, respectively.

He is currently an Assistant Professor in the Department of Electrical and Computer Engineering, Iowa State University, Ames. His research interests include design and analysis of algorithms, computer-aided design of very large scale integration physical design, and performance-driven interconnect optimization.

Prof. Chu has served on the Technical Program Committees of the ACM International Symposium on Physical Design since 2001. He has also served as an organizer for the ACM SIGDA Ph.D. Forum since 2002. He received the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN Best Paper Award in 1999 for his work on performance-driven interconnect optimization and the 1998–1999 Bert Kay Best Dissertation Award from the Department of Computer Sciences, University of Texas, Austin.

**Evangeline F. Y. Young** received the B.Sc. and M.Phil. degrees in computer science from the Chinese University of Hong Kong, Shatin, N.T., Hong Kong, in 1991 and 1993, respectively, and the Ph.D. degree from the University of Texas, Austin, in 1999.

Currently, she is an Assistant Professor in the Department of Computer Science and Engineering, Chinese University of Hong Kong. She is now working actively on floorplan design optimization, circuit partitioning, circuit retiming, and packing representation. Her research interests include algorithms and computer-aided design of very large scale integration circuits.