以繞線擁擠度最佳化為導向之平面規劃與模組形狀變化
Congestion-Driven Floorplanning by Adaptive Modular Shaping

黃信雄 [1] 張仲喬 [2] 林志遠 [2] 謝財明 [*2] 李志宏 [3]
H. H. Huang[1], C. C. Chang[2], C. Y. Lin[2], T. M. Hsieh[*2], C. H. Lee[3]

摘要：本論文使用兩階段的方法來同時改善平面規劃之擁擠度和連線總長度。首先使用模擬退火法的方式來找到一組兼具繞線長度、面積和繞線擁擠度的平面規劃解答，然後針對平面規劃中最擁擠的區域進行改善。每個被選到的模組將會被分割成一個小矩形集合，藉此來延長相鄰模組間的公用邊長度。我們將採用非線性規劃的方法以決定模組最後的形狀，目標是在不增加面積的條件下，改善局部擁擠度高的區域。和傳統的方法相比較，模組變形的方法將可降低 22%的繞線擁擠度和 1.54%連線總長度。

***Abstract*: In this paper, we implement a two-stage process to simultaneously minimize wire congestion and total wire-length at floorplanning stage. We first use a simulated-annealing approach with sequential-pair representation to find a floorplan with minimal wire congestion, total wire-length and area. Each of the two selected adjacent soft modules in congested region is then divided into a set of connected sub-rectangles to increase the common boundary between the adjacent modules. The longer common boundary actually reduces total wire-length between the pins of two modules and minimizes the wire congestion. A nonlinear programming method is used for modular shaping mentioned above to further minimize the wire congestion without the penalty of area. Compared to the traditional method without consideration of the modular shaping, we show experimentally that our algorithm achieves an average reduction rate of 22% and 1.54% in wire congestion and total wire-length, respectively.**

關鍵字：平面規劃，模組變形，非線性程式規劃，擁擠度
***Keywords*: floorplanning, modular shaping, nonlinear programming, congestion.**

I. Introduction

The floorplanning stage determines the locations and shapes of a set of modules in a chip. Traditional floorplanners mostly focus on the minimum of area and total wire-length, but the topics of the congestion and routability are usually ignored [1]-[3], [5]-[20], [22]-[25]. As the complexity of advanced VLSI technology is growing rapidly, the interconnections between modules become longer and denser to consume more routing resource. Because of the lack of routing resource in congested region, the old floorplanners produce an unroutable design. In order to speedup the convergence of future router and achieve one-pass design flow, it is necessary to pay attention to the minimum of congestion as early as possible.

Some congestion models are developed to estimate the congestion of a floorplan. The authors [19] proposed an empirical model with the idea of bounding box to estimate wire congestion and presented a new objective function to reduce wire congestion. The novel approaches first divide the chip into a set of fixed-size grids and then estimate the probability of each net which passes through each boundary of the grids [16][21]. The authors [4] use a two-stage simulated annealing to optimize the congestion by intersection-to- intersection model. Hsieh *et al.* [26] propose a new congestion model based on the probabilistic analysis and a new concept of irregular-size (IR) grids instead of previous models with fixed-size grids. In fact, there's a tradeoff between the accuracy and efficiency. If the numbers of grids are small, the result is inaccuracy. If the numbers of grids are large, the computation is time consuming.

Recently, most floorplanning algorithms can deal with rectangles and some proposed methods can handle arbitrarily shaped rectilinear modules. Wong *et al.* [7] extend the Polish expression representation to handle L- and T-shaped modules. In paper [17], the authors present an approach extending the sequence-pair approach for rectangular block to arbitrarily sized and shaped rectilinear blocks. They first explore the properties of L-shaped and decompose rectilinear blocks into a set of L-shaped blocks. Lee *et al.* [3] introduce the concept of block partition that the shape of modules can be automatically determined according to the optimization objectives. Chu and Young [5] formulate and adjust the shaping of nonrectangular by Lagrange relaxation. However, most of them can handle fixed-shaped modules and they assume that all of the rectilinear modules are not flexible in their shapes.

In this paper, we propose a congestion-driven approach that consists of a congestion-driven floorplanner followed by the modular shaping. The former is simulated-annealing way to obtain a congestion-driven floorplan and the latter further improves the local congested region of the former by the nonlinear programming (NLP). Our contributions are as follows:

- A simulated-annealing with sequential pair representation is to find a floorplan with minimum wire congestion, total wire-length and chip area.
- The algorithm divides the congested region between the adjacent soft modules into a set of sub-rectangles to increase the common boundary which reduces total wire-length and minimizes the wire congestion.

[1] 中原大學電子所資訊組
[2] 中原大學資訊工程學系
[3] 嶺東科技大學資管學系
[*] Corresponding author. E-mail: hsieh@cycu.edu.tw
[1] Institute of Electronic Engineering, Chung Yuan Christian University, Chung-Li, Taiwan, R.O.C.
[2] Department of Information and Computer Engineering, Chung Yuan Christian University, Chung-Li, Taiwan, R.O.C.
[3] Department of Information Management, Ling Tung College, Taichung, Taiwan, R.O.C.

Manuscript received 17 February 2006; revised 28 April 2006; accepted 11 May 2006

- Our approach which can handle both fixed-shaped modules and flexible rectilinear modules is essential to optimize area utilization and congestion.
- An accurate and efficient congestion model is embedded into our floorplanner to estimate the congestion. Compared to the previous floorplanners, our model with the concept of irregular-regular grid (IR-grid) works efficiently.

The remainder of this paper is organized as follows. Section II describes our motivation to find a congestion-driven floorplan, terminology and problem definition. Section III illustrates a new congestion-driven floorplanner, followed by the nonlinear programming approach in Section IV. Our algorithm contains two-stage process in Section V. Experimental results are shown in Section VI and conclusions are attached in Section VII.

II. Preliminary

1. Motivation

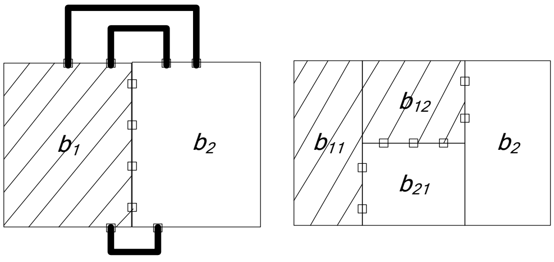
Previous papers mostly focus on the minimum of area and total wire-length without consideration of congestion. However, there is always no routing resource in congested regions and some nets must make detour routes which increase the wire-length and even timing violation. In Figure 1(a), there are seven nets connected two soft modules. Originally, the length of common boundary is twelve (unit length) and the sum of wire width and wire spacing is three (unit length). Therefore the routing capacity is 4(=12/3) and three nets go outside the boundary box spanning by two selected adjacent soft modules.

We know that detour routes increase total wire-length and consume the routing resource of the neighborhood regions, (i.e. make the neighborhood regions more congested). Our main idea is to divide the adjacent soft modules (b_1 and b_2) into a set of connected sub-rectangles such as b_{11} and b_{12} (b_{21} and b_{22}) to enlarge the common boundary. In Figure 1(b), the length of common boundary enlarges to twenty-one (unit length). The routing capacity becomes 7(=21/3) and there is no detour route. It is obviously that the local congestion and total wire-length decrease without the penalty of area.

By observation of Figure 1, we find that different topological relationships among modules make different congestion. How to determine the shaping between soft modules with the minimum congestion? It motivates us to apply the modular shaping for the boundary between soft modules before pin-assignment stage. It is deserved to mention that the method will reduce the congestion and total wire-length without the penalty of area.

2. Terminology and Problem Definition

In this sub-section, we first discuss the terminologies (connected module and adjacent module), formulas for area, wire-length and congestion and define the problem. In Figure 1, b_1 and b_2 are regarded as adjacent modules because two modules have a common boundary. Moreover, b_1 (b_2) is divided into a set of sub-rectangles b_{11} and b_{12} (b_{21} and b_{22}). Modular shaping can automatically transfer a soft module into a set of sub-rectangles and these sub-rectangles are called connected modules.



(a) (b)

Fig. 1 Under the traditional model in (a), the routing capacity of the boundary between two adjacent soft modules is four pins to be placed. For our model in (b), the capacity is seven pins and total wire-length decreases.

For n rectangular soft modules and m multi-pin nets, we will find a non-overlap floorplan. For each soft module b_i has bottom-left coordinate (x_i, y_i) and top-right coordinate $(\hat{x}_i, \hat{y}_i)$, respectively. Let

$$BW_i = (\max\{\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_n\} - \min\{x_1, x_2, \ldots, x_n\}) \quad \text{and}$$

$$BH_i = (\max\{\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_n\} - \min\{y_1, y_2, \ldots, y_n\}) \cdot$$

The area that is measured as the smallest rectangle enclosing all soft modules is computed as:

$$BW_i \times BH_i \tag{1}$$

where $\max\{\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_n\}$ ($\max\{\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_n\}$) is the maximum value of top-right x (y) -coordinates and $\min\{x_1, x_2, \ldots, x_n\}$ ($\min\{y_1, y_2, \ldots, y_n\}$) is the minimum value of bottom-left x (y) -coordinates.

In the paper, the pin locations aren't fixed and we regard the center of the soft module as the pin locations. For a multi-pin net, we first decompose it into a set of two-pin nets by minimal spanning tree and then the half-perimeter approach is applied to estimate the total wire-length, see Figure 2.

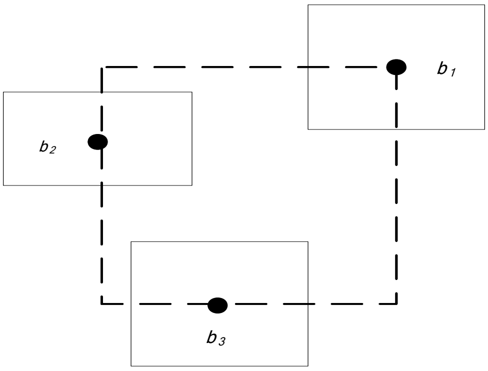


Fig. 2 Half-perimeters approach to estimate wire-length.

The congestion cost of the floorplan is estimated by the concept of global bins in Figure 3. First we divide the whole chip into a set of fixed-size global bins which have routing supply (s_{gi}) and routing demand (d_{gi}). Then a simple global router is applied to determine the routing paths of global edges. Hence we obtain the number of the nets passing through each global edge, named routing demand (d_{gi}). Besides, the routing supply is determined by the length of global edge and technique parameters. Furthermore, we define the overflow of the global bin as formula (2):

$$overflow_{g_i} = \max(d_{g_i} - s_{g_i}, 0) \tag{2}$$

In order to measure fairly congestion of a floorplan, we take "*OF*" as metric [19] and the congestion cost of a floorplan is defined as:

$$OF = \sum_{i=1}^{u} overflow_{g_i} \tag{3}$$

where u is the numbers of global bins. A floorplan with higher *OF* value has more congested global bins.

Based on the above analysis, we propose a new congestion-driven floorplanning by adaptive modular shaping (CFMS) which can automatically transfer a soft module into a rectilinear shaped module to minimize the congestion and total wire-length. The problem is defined as follows:

Given a set of n rectangular soft modules $\{b_1, b_2,\ldots,b_n\}$ and a set of multi-pin $\{n_1,n_2,\ldots,n_m\}$ nets. The module b_i has area a_i. The module b_i can be divided into a set of sub-rectangles to optimize the total wire-length, chip area and congestion. The bounding box of these connected sub-rectangles has height h_i and width w_i The aspect ratio of the bounding box is defined as h_i / w_i and can be varied within a aspect ratio $[r_{i,min}, r_{i,max}]$. A floorplan is said to be feasible if and only if no two modules overlap with each other, and the aspect ratio of each module after shaping if necessary must vary within the given range. Each multi-pin net can be divided into a set of two-pin nets. Based on above, our objective is to find a feasible floorplan solution with minimal congestion and total wire-length.

III. A Congestion-Driven Floorplanner

During the process of perturbing soft modules of a floorplan, we take our new congestion model as congestion metric. Besides, we use *OF* as measurement to choose a floorplan with the lowest congestion[19]. In this section, our new congestion model is described. In subsection 1, we first analyze the congestion distribution according to during the floorplanning stage. Then the congestion formula is shown in subsection 2. Finally, time complexity is shown in subsection 3.

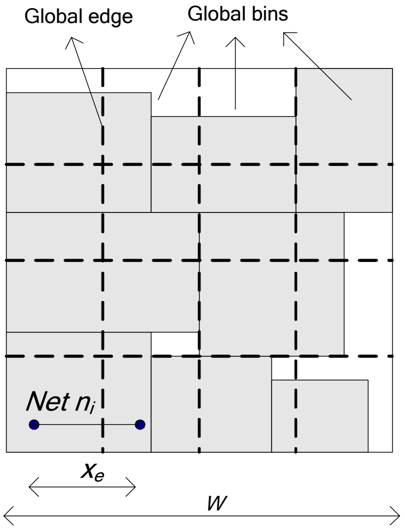


Fig. 3 The global bins of a floorplan

1. Analyze the Congestion Distribution

Inspire of the paper which study the behavior of wire congestion during placement stage [19], we analyze the wire congestion at the floorplanning stage. Assume there are t two-pin nets n_1, $n_2,\ldots,n_t$ with their corresponding wire-length l_1, $l_2,\ldots,l_t$ in a chip which contains width W and height H. Besides, we assume that all nets are distributed over the chip randomly. For a net n_i in Figure 3, the starting and ending pin locations are x_i and x_{i+1}, respectively. The starting pin location of the net n_i with length of l_i is uniformly distributed between (0, W-l_i) because the range (W-l_i, W) is not allowed to place starting pin. To summarize, we can get the expected value of the number of nets passing through global edge and x_e is defined as the x position of global edge e. We know that n_i intersects e if and only if $x_i < x_e$ and $x_i + l_i > x_e$. Then we derive the equation for the probability $p_i(x_e)$ which net n_i intersects global edge e:

$$\begin{cases} p_i(x_e) = \dfrac{x_e}{W - l_i} & ,if \quad 0 \le x_e \le l_i; \\ p_i(x_e) = \dfrac{l_i}{W - l_i} & ,if \;\; l_i \le x_e \le W - l_i; \\ p_i(x_e) = \dfrac{W - x_e}{W - l_i} & ,if \;\; W - l_i \le x_e \le W; \end{cases} \tag{4}$$

We take a function $g(l)$ that represents the numbers of the nets with wire-length of l to estimate the expected value of the numbers of total crossing nets. For a floorplan, we can estimate the expected numbers of nets crossing the global edge e by the following formula:

$$\sum_t p_t(x_e) g(l_t) \tag{5}$$

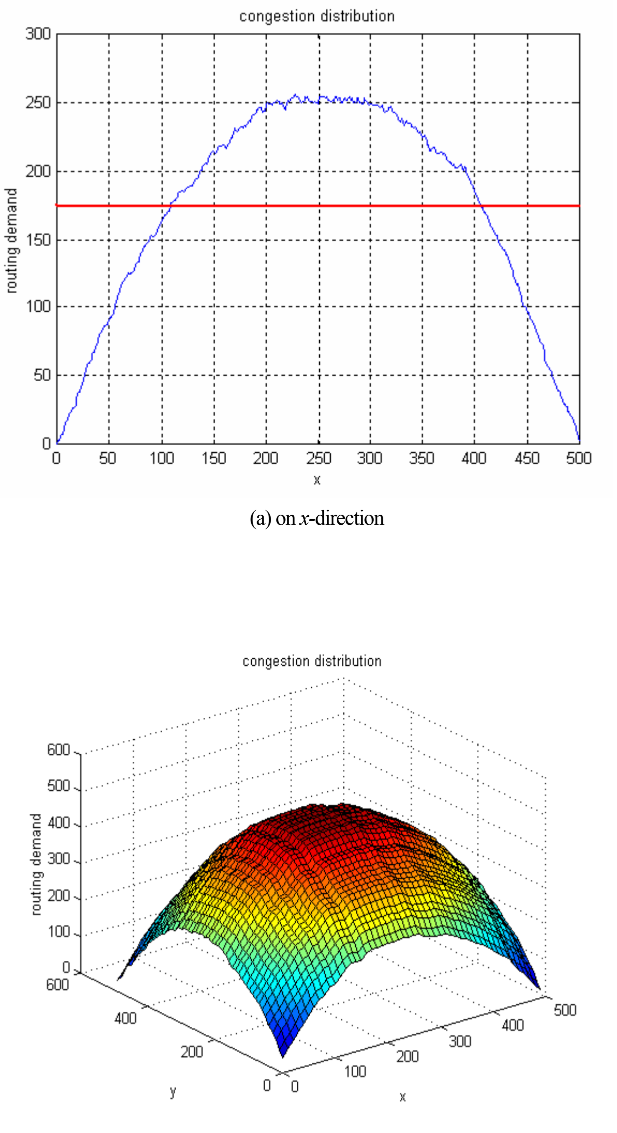
For example, we take the largest benchmark, ami49 to explain how to analyze congestion distribution and demonstrate the accuracy in experimental result. We first assign randomly 1000 two-pin nets over a chip size of $500 \times 500 \mu m^2$. Then we generate randomly 30 floorplan for ami49 and add up the total congestion of 30 floorplan solutions. By the formula (5), we find the congestion distribution of x-direction in Figure 4(a) that the x-axis and y-axis represent the x position of a floorplan and the numbers of nets crossing the global edge e at position x, respectively.

After analyzing the data, we make two important conclusions at the floorplanning stage. One is that congestion mostly appears near the center of bounding box spanning by two connected modules. The other is the one-dimension (x-direction) region from $W/5$ to $4 \times W/5$ has over average routing demand, where W is the width of the chip. In similar way, we derive the wire distribution of y-direction. Finally, we get the xy-direction congestion distribution in Figure 4(b).

2. A New Congestion Model

Traditional probabilistic-based algorithm takes bounding box as estimated wire congestion, see Figure 5(a). But the approach takes much computation time for lower congested regions. The intersection-to-intersection way first draws a line from the center of a module to another and takes the smallest rectangle which contains the two intersection points on two modules, see Figure 5(b). But it over-estimated the wire congestion for the congested region. In order to accurately estimate the congestion of a floorplan, we propose an IR-grid model instead of a fixed-size model, see Figure 5(c).

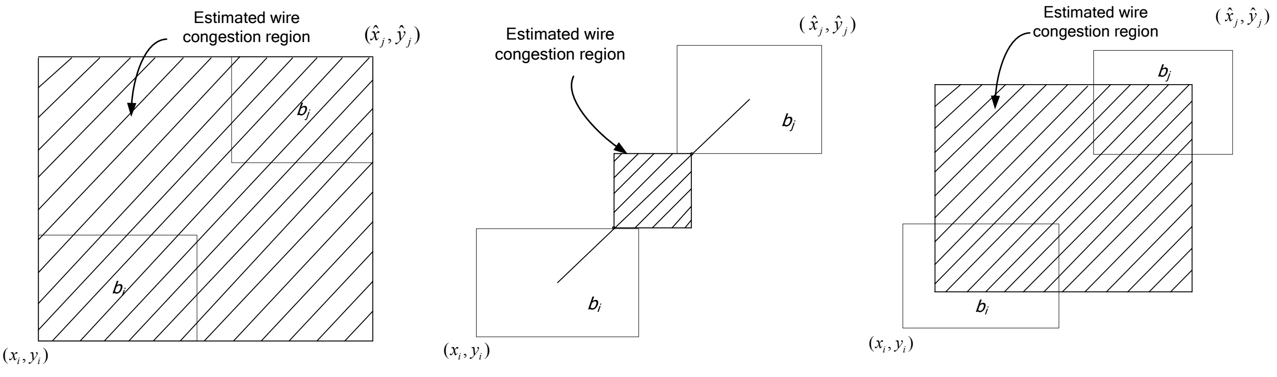
A region which a set of nets between connected modules have probabilities to pass through is called *routing region*. Assume that each net goes outside the bounding box if there is no enough routing resource. According to the analysis in subsection A, the *estimated wire congestion region* is located near the center of routing region. Therefore, the *estimated wire congestion region* is given as follows:



(a) on x-direction

(b) on xy-dimension

Fig. 4 Congestion Distributions of ami49.



(a)

(b)

(c)

Fig. 5 Three methods to determine routing regions. (a) bounding box routing region. (b) intersection-to- intersection routing region. (c) IR-grid routing region on the congestion distribution.

For a soft module b_i (b_j) in Figure 6(a), it has bottom-left corner (x_i, y_i) ((x_i, y_i)) and top-right corner $(\hat{x}_i, \hat{y}_i)$ ($(\hat{x}_j, \hat{y}_j)$). Besides, there are n_{ij} two-pin nets between two connected modules b_i and b_j. The estimated wire congestion region, called r_{ij}, is the rectangle that its bottom-left and top-right coordinates are

$(\max(\hat{x}_i, \hat{x}_j) - \delta_x, \max(\hat{y}_i, \hat{y}_j) - \delta_y)$ and

$(\min(x_i, x_j) + \delta_x, \min(y_i, y_j) + \delta_y)$ respectively,

where the values of $\delta_x = (\max(\hat{x}_i, \hat{x}_j) - \min(x_i, x_j)) / usr1$

and $\delta_y = (\max(\hat{y}_i, \hat{y}_j) - \min(y_i, y_j)) / usr2$ that *usr1* and *usr2* are user specified parameters. In our implementation, we take the routing region spanning by *x*-direction range from 1/5*W* to 4/5*W* and *y*-direction range from 1/5*H* to 4/5*H* as the *estimated wire congestion region*, where *W* and *H* are the width and height of bounding box spanning by two modules of Figure 6(b), respectively.

To estimate the congestion for the area of r_{ij}, we define the congestion cost, C_{ij}, based on the analysis of supply and demand:

$$C_{ij} = \begin{cases} \dfrac{n_{ij} - p_{ij}}{A_{ij}} & \text{if } n_{ij} - p_{ij} > 0 \ ; \\ 0 & \text{otherwise;} \end{cases} \tag{6}$$

where A_{ij} is the area of r_{ij}, n_{ij} is the numbers of two-pin nets between two adjacent modules b_i and b_j, and p_{ij} is the numbers of common pins along the common boundary.

In order to estimate precisely the congestion of a floorplan, we must concern the overlap of estimated wire congestion regions. To simplify the complexity, only overlap of two estimated wire congestion region is considered, shown in Figure 7. The congestion cost of the overlap of the two estimated wire congestion regions, named r_{ij} and r_{st}, is defined as follows:

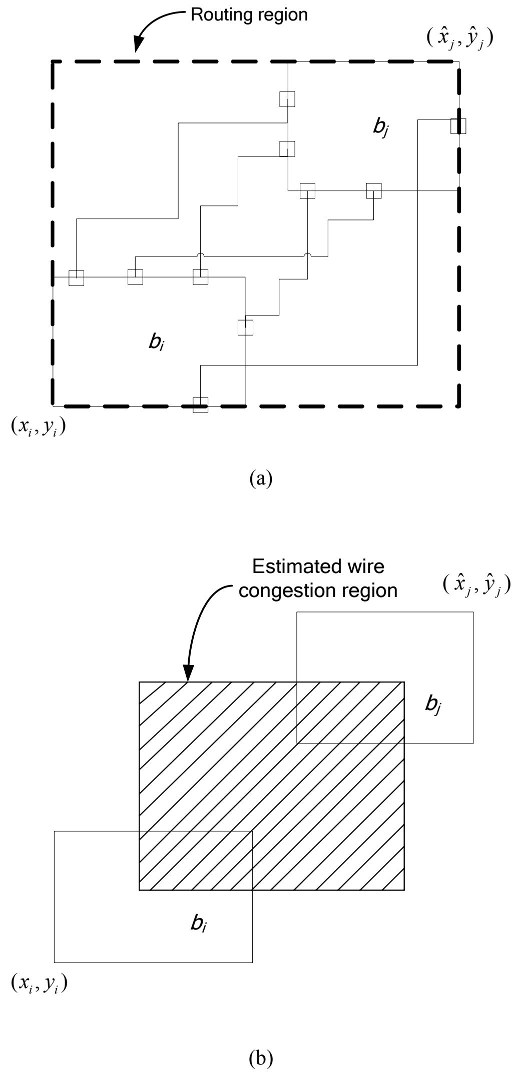
$$K_{ij,st} = \frac{O_{ij,st}}{A_{ij}} \times C_{ij} + \frac{O_{ij,st}}{A_{st}} \times C_{st} \tag{7}$$

where $O_{ij,st}$ is the overlap of r_{ij} and r_{st}. A_{ij} (A_{st}) represents area of r_{ij} (r_{st}) and C_{ij} (C_{st}) denotes the estimated wire congestion of r_{ij} (r_{st}).

The estimated wire congestion cost of a floorplan, *CGT*, is defined as follows:

$$CGT = \sum_{i=1}^{n} \sum_{j=i+1}^{n} C_{ij} + \sum\nolimits_{i,j,s,t} K_{ij,st} \tag{8}$$

where *n* is the numbers of soft modules.



(a)

(b)

Fig. 6 (a) Routing region (b)the estimated wire congestion region of our congestion model

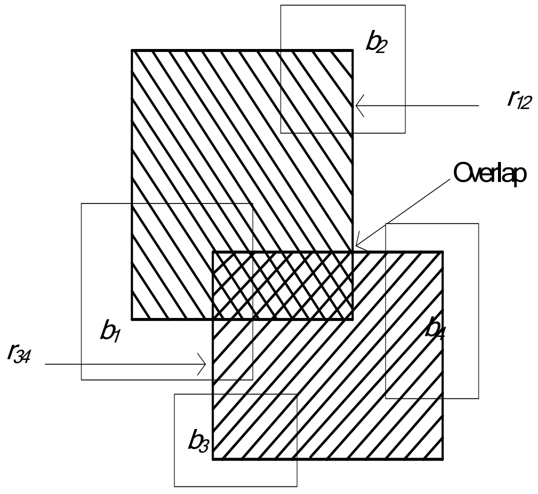


Fig. 7 The overlap of two estimated wire congestion regions

3. Time Complexity

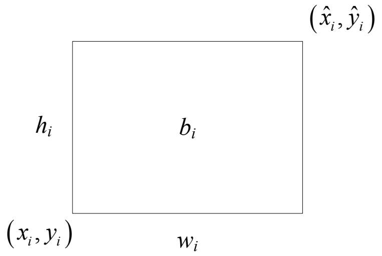
From formula (8), we found that two terms affect total congestion of a floorplan. In the first term, it takes $n \times (n\text{-}1)/2$ times to compute the congestion between two connected soft modules and the time complexity is O(n^2). In the second term, we find experimentally that only the top n congestion costs are considered as significant factors. Its complexity is also O(n^2). In other words, the complexity of our algorithm is O(n^2).

IV. Nonlinear Programming

In sub-section 1, we'll discuss the relationship of the topology and its corresponding mathematical constraints. To simplify, we only consider two pre-defined topologies for the horizontal (or vertical) relationship which are discussed in sub-section 2. For each relationship, the NLP-based approach is to enlarge the common boundary between two adjacent modules to increase the capacity of pins, see sub-section 3. In this section, we will discuss how a NLP-based approach minimizes the wire congestion by adaptive modular shaping. The mathematical model, including five constraints is described in subsection 1 and the modular shaping is listed in subsection 2. In the final subsection, the procedure that transfers the relationship into nonlinear programming is shown.

1. The Mathematical Model

In this subsection, we'll discuss the topological relationship and their corresponding constrains. The constraints are given to determine the modular shaping and they are described in the latter sub-sections.



(a)

$$y_i + h_i = \hat{y}_i \quad (9)$$

$$x_i + w_i = \hat{x}_i \quad (10)$$

$$a_i = h_i \times w_i \quad (11)$$

$$r_i = h_i / w_i \quad (12)$$

$$L_i \le r_i \le U_i \quad (13)$$

(b)

Fig. 8 (a) Basic soft module (b) the basic constraints

1-1. Basic Constraints

For a soft module b_i in Figure 8 (a), the basic constraints are described in Figure 8(b). (x_i, y_i) and $(\hat{x}_i, \hat{y}_i)$ are the bottom-left and top-right coordinate, respectively. Besides, h_i and w_i are the height and width of the module b_i, respectively. In order to avoid the snake shaping of the soft module, the ratio (r_i) of height and width must be within the range of upper bound U_i and lower bound L_i.

1-2. Area Constraints

Each module b_i of the selected adjacent soft modules in congested region is divided into a set of k connected sub-rectangles b_{i1}, b_{i2}, ... ,b_{ik} to increase the common boundary between the adjacent modules. The area constraint is defined

$$a_i = a_{i1} + a_{i2} + ... + a_{ik} \quad (14)$$

where a_i and a_{ik} are the area of the soft module b_i and a set of connected sub-rectangles b_{ik}.

1-3. Border Constraints

To simplify the problem, we only perform modular shaping for the adjacent soft modules. Of course, the non-overlapping constraints for each modules and a set of connected sub-rectangles must be satisfied. In Figure 9 (a), the bounding box forming by module b_i and b_j is named border and the borer constraints are still satisfied after performing modular shaping. Figure 9(b) and Figure 9(c) are the invalid and valid modular shaping, respectively.

For sub-modules, we develop four alignments to guide the sub-modules to meet the border constraints. Figure 10 illustrates the four alignments and Table 1 shows the corresponding constraints, including the constants of z_t, z_b, z_l and z_r. The value of z_r (z_l) is the maximum (minimum) x-coordinate and the value of z_t (z_b) the maximum (minimum) y-coordinate, respectively.

1-4. Connective Constraints

In order to avoid fragment of the sub-modules, we must pay attention to the connective constraints. In Figure 11, b_{ip} and b_{iq} are two sub-modules of soft modules b_i which is divided into a set of k sub-rectangles (p=1, 2, …, k; q =1, 2,…, k, $p \neq q$). In Figure 11(a), the sub-module b_{ip}

Table 1 Four alignments and their constraints

Type	Constraints
Top alignment	$\hat{y}_i = z_t$
Bottom alignment	$y_i = z_b$
Left alignment	$x_i = z_l$
Right alignment	$\hat{x}_i = z_r$

is bellow of b_{iq} and their corresponding constraint is as follows:

$$\hat{y}_{ip} = y_{iq} \tag{15}$$

where $\hat{y}_{ip}$ is the top-right y-coordinate of sub-module b_{ip}, and y_{iq} is the bottom-left y-coordinate of sub-module b_{iq}. In Figure 11(b) the module b_{iq} is locate the right side of b_{ip} and their corresponding constraints is

$$\hat{x}_{ip} = x_{iq} \tag{16}$$

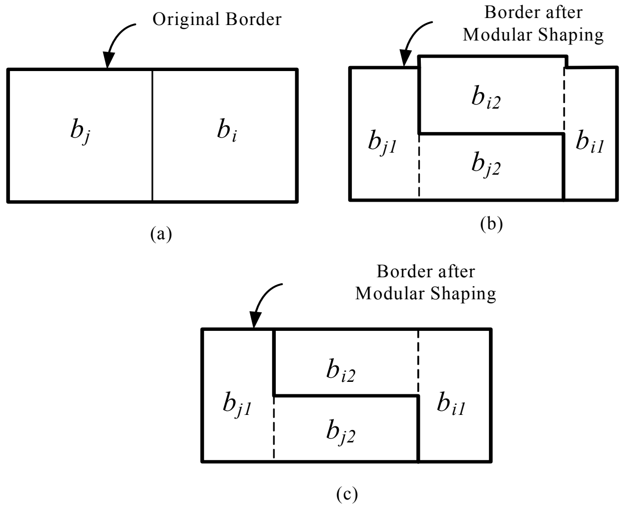
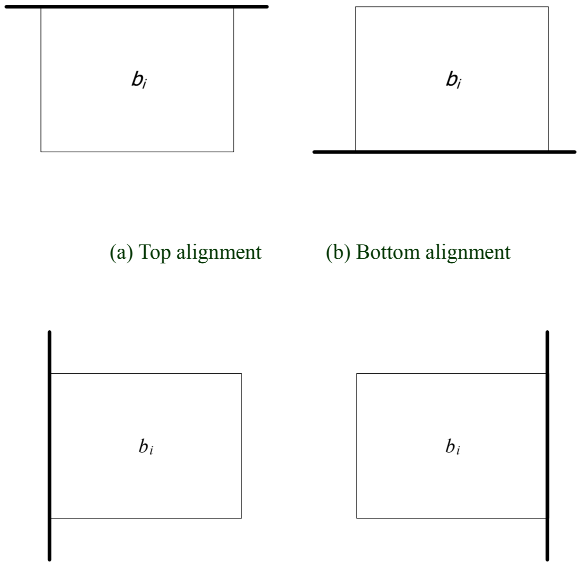


Fig. 9 Illustrations of border constraints



(a) Top alignment (b) Bottom alignment

(c) Left alignment (d) Right alignment

Fig. 10 Four alignments for border constraints

where $\hat{x}_{ip}$ is the top-right x-coordinate of sub-module b_{ip} and x_{iq} is the bottom-left x-coordinate of sub-module b_{iq}

1-5. Topological Constraints

The soft-module b_i of Figure 12 has topological relationship with the adjacent module b_{it} (t=1, 2, …, 8). Table 2 shows the corresponding constraints of eight topological relationships.

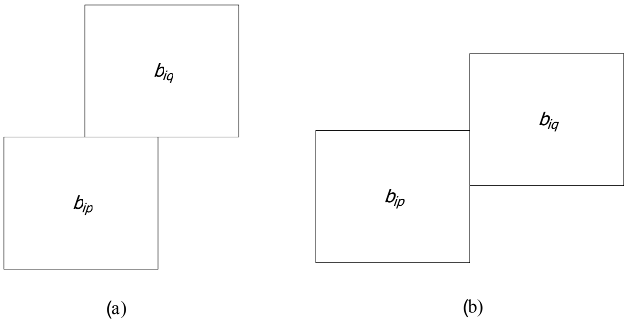


Fig. 11 Illustrations of connective constraints. (a) the vertical and (b) horizontal relationships.

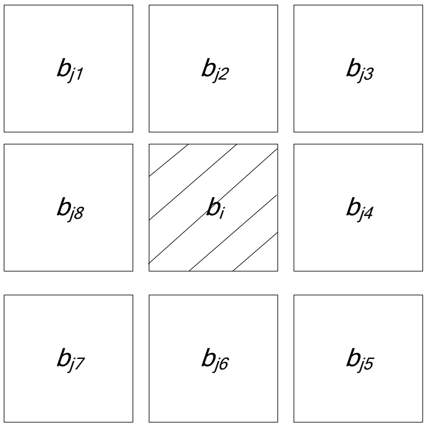


Fig. 12 Eight topologies of two adjacent modules.

Table 2 The corresponding topological constraints.

Module		Topological Constraints
i	$j1$	$\hat{x}_{j1} \le x_i$; $\hat{y}_i \le y_{j1}$
i	$j2$	$\hat{y}_i \le y_{j2}$
i	$j3$	$\hat{x}_i \le x_{j3}$; $\hat{y}_i \le y_{j3}$
i	$j4$	$\hat{x}_i \le x_{j4}$
i	$j5$	$\hat{x}_i \le x_{j5}$; $\hat{y}_{j5} \le y_i$
i	$j6$	$\hat{y}_{j6} \le y_i$
i	$j7$	$\hat{x}_{j7} \le x_i$; $\hat{y}_{j7} \le y_i$
i	$j8$	$\hat{x}_{j8} \le x_i$

2. Modular Shaping

We know that the shaping of the modules has deeply impact on the wire congestion and total wire-length. The idea of block partition is to apply modular shaping properly to achieve area minimization [3]. It inspires us to exploit pre-defined topologies of non-rectangular shapes by nonlinear programming. To simplify, we only design two pre-defined topologies for each modular relationship, including horizontal and vertical relationships. In Figure 13, here are 18 topological types, including 9 horizontal and 9 vertical relationships. Each topological type contains two pre-defined types of partitions, i.e. modular shaping. The 9 vertical relationships are listed here and we take type 1 for example. In similar way, we can draw for the other relationships of the modules. Each type contains two pre-defined modular shaping listed in Figure 14.

Each one of the two selected adjacent soft modules in congested region is divided into a set of connected sub-rectangles to increase the common boundary between the adjacent modules. Let L_{com} and v be the length of common boundary and sum of minimum wire width and spacing, respectively. Then the routing capacity is computed as L_{com}/v. From the analysis, we conclude that the longer common boundary actually reduces the total wire length between the pins of the two modules and minimizes the wire congestion.

Each type contains two pre-defined modular shaping. In Figure 14(a), there's a vertical relationship of two soft modules with two types of modular shaping. L_{com}, that denotes the original length of common boundary between module b_p and b_q, is equal to $\{d\}$, where d is the length of common boundary. In Figure 14(b), the length of L_{com} becomes $\{d + d_1 + d_2\}$, where d_1 is the height of module b_{p2} (or b_{q2}) and d_2 is the height of module b_{p4}. By the similar way, we can compute the length of L_{com} in Figure 14(c) is also $\{d + d_1 + d_2\}$.

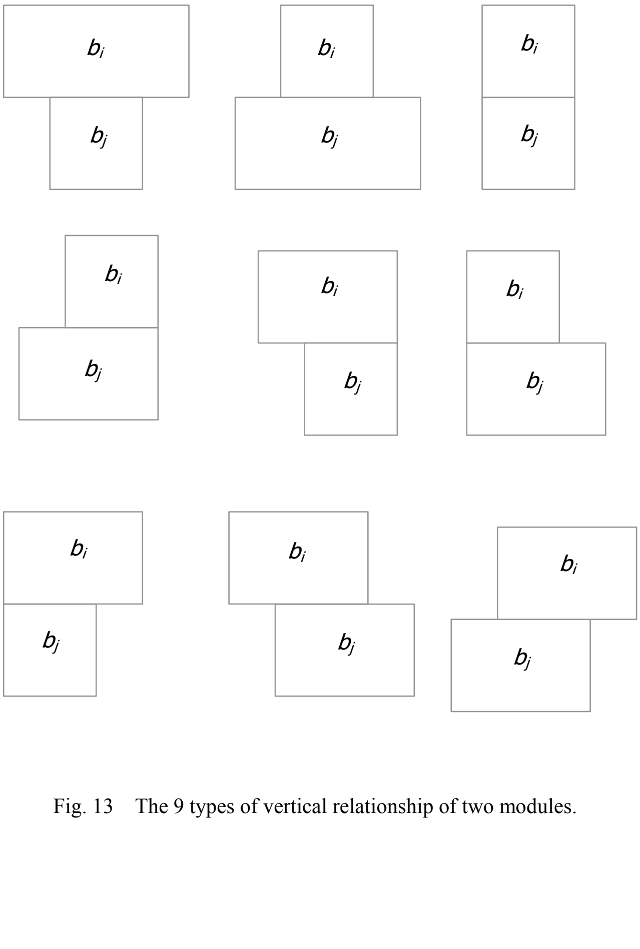


Fig. 13 The 9 types of vertical relationship of two modules.

3. The NLP-based approach to optimize the wire congestion by modular shaping

Based on the analysis of the two sub-sections, we will discuss how a NLP-based algorithm further improves wire congestion of the initial floorplan. We read the information of the two-pin nets and the modules of an initial floorplan which is obtained from the congestion-driven floorplanner. At the step 1 of Figure 15, a new congestion model in Figure 5(c) is used to estimate the wire congestion between the adjacent modules and we'll select the most congested region to reduce wire congestion. We then formulate the five topological relationships of the most congested region into nonlinear programming, see step 2 in Figure 15. At step 3, we solve the NLP problem by LINGO to enlarge the common boundary to increase the capacity of pins between the two adjacent modules by using LINGO 7.0. By doing so, the wire congestion is further improved and total wire-length is minimized. At step 4, we propose a fixed-size grid model to estimate the congestion of a floorplan. The formula (2) and (3) are applied to estimate the congestion of each fixed-size grid and the total congestion of a floorplan, respectively. We will perform modular shaping for the current most congested region iteratively and the procedure will terminate if there is no improvement any more, see step 5.

To deserve to mention, we obtain a congestion-driven floorplan by our irregular-size congestion model at stage 1 and a *OF* congestion model with grid-size $10 \times 10\text{um}^2$ is used to fairly evaluate the congestion cost of floorplan at stage 4.

In other words, the NLP-based approach automatically transforms the selected module into a set of connected sub-rectangles to enlarge the common boundary between adjacent modules. The longer common boundary actually reduces the total wire-length between the pins of the two modules and minimizes the congestion.

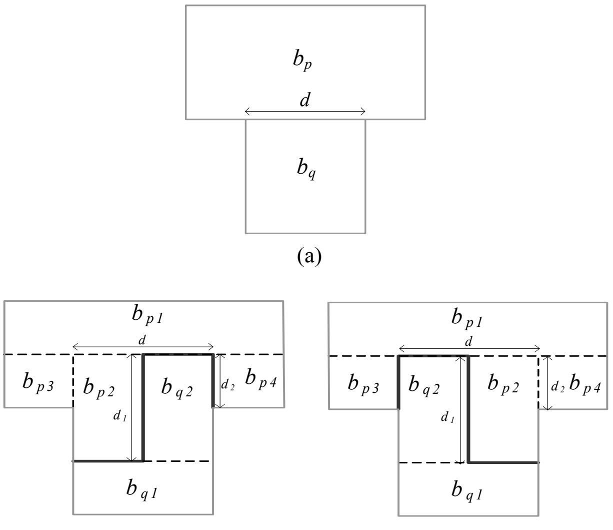
V. Our Algorithm

Our algorithm is a two-stage process, including the SA-based approach and then the nonlinear programming method. A congestion-driven floorplan is obtained in subsection 1 and we further minimize the wire congestion by adjust the modular shaping in subsection 2. Finally, the congestion- driven floorplanning by adaptive modular shaping is described in next two subsections.

1. The SA-based Floorplanner

We use simulated annealing algorithm with sequence-pair (SP) representation for non-slicing floorplans to obtain a floorplan with minimum of wire congestion and total wire-length. Figure 16 shows the flow chat of congestion-driven SA-based floorplanner. In the annealing process, we simultaneously consider the area, total wire-length and the congestion for searching a feasible solution.

The objective of simulated annealing process is to-minimize wire congestion. For a floorplan, we use half-perimeter method to obtain the total wire-length and take the center points of module as the pins location.



(a)

(b) (c)

Fig. 14 (a)Vertical relationship of two soft modules, (b) and (c) are two pre-defined types of modular shaping.

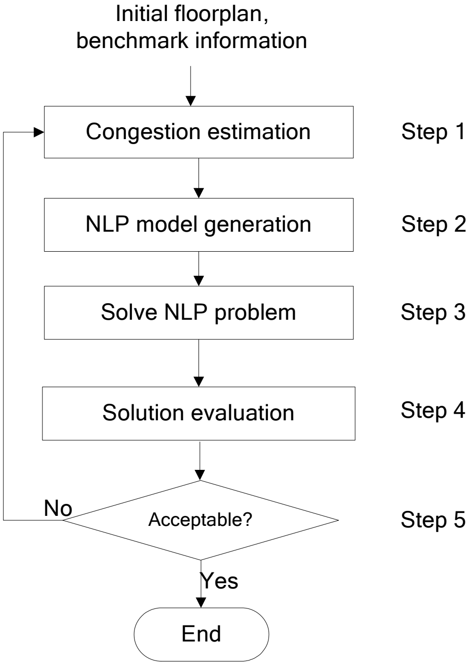
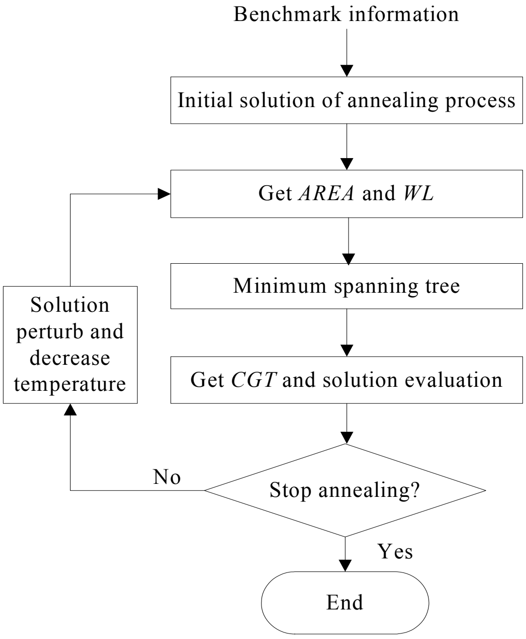


Fig. 15 The NLP-based post-processing for further improving the local congested region between the adjacent modules.


Benchmark information
Initial solution of annealing process
Get *AREA* and *WL*
Minimum spanning tree
Get *CGT* and solution evaluation
Solution perturb and decrease temperature
Stop annealing?
No
Yes
End

(a)

Algorithm Simulated_Annealing floorplanning
Input: a set of *n* modules $B = \{b_1, b_2, \ldots, b_n\}$ and a set of *m* multi-pin nets $N = \{n_1, n_2, \ldots, n_m\}$.
Output: an intermediate floorplan for post-processing
Method:

```
{  struct feasible_floorplan f, g;
   float T;
   f ← initial_floorplan();
  do{
       do{
            g ← perturb(f);
            cost(g) ← $\alpha \times AREA + \beta \times WL + \gamma \times CGT$ ;
            $\Delta c$ ← cost(g)-cost(f);
            If (min($e^{-\Delta c/T}$,1) > random(1))
                f ← g;
       }while((downhill_move<$C_1$)||(total_move<$C_2$));
            T ← new_temperature(T);
  }while((accept_ratio>$C_3$) &&(downhill_move_ratio>$C_4$) && ($T$>$T_e$));
}
```

(b)

Fig. 16 (a) The congestion-driven SA-based floorplanner (b) the algorithm of SA-based floorplaner

A heuristic minimum spanning tree algorithm is used to divide each multi-pin net into a set of two-pin nets. Moreover, the cost function is defined as:

$$cost = \alpha \times AREA + \beta \times WL + \gamma \times CGT \tag{17}$$

where *AREA* is the area of the floorplan, *WL* represents the estimated total wire length, *CGT* denotes the estimated wire congestion cost of a floorplan, and *α, β* and *γ* are user defined parameters.

2. The NLP-based Modular Shaping Algorithm

At the previous stage, we obtain a congestion-driven floorplan and the congestion of the local congested region will be further improved at the stage. For two adjacent soft modules, a new congestion model is applied to estimate the current most congested region. We then apply nonlinear programming method to determine the modular shaping. Furthermore, the algorithm stops until the total congestion doesn't reduce any more.

The objective of this post-process is to maximize the common length between adjacent modules to increase the numbers of pins, p_{ij}, see Figure 1. By formula (3), we can derive the congestion will reduce when the number of p_{ij} increases under the numbers of a_{ij} and n_{ij} *are fixed.* The following is the pseudo code for module b_i:

$$\textit{maximize} \quad L_{com}$$

$$\textit{subject to} \quad x_i + w_i = \hat{x}_i \; ; \tag{18}$$

$$y_i + h_i = \hat{y}_i \; ; \tag{19}$$

$$a_i = h_i \times w_i \; ; \tag{20}$$

$$r_{i,min} \le r_i = h_i / w_i \le r_{i,max} \; ; \tag{21}$$

$$a_i = a_{i1} + a_{i2} + a_{i3} + \cdots + a_{ik} \; ; \tag{22}$$

$$\hat{y}_i = z_t \; ; \; y_i = z_b \; ; \; x_i = z_l \; ; \; \hat{x}_i = z_r ; \tag{23}$$

Constraints (18)-(21) are defined as basic geometry condition, and constraint (22) is applied to meet the area constraint. Besides, the border condition is kept by four corners of constraint (23). Of course, the solution could not only satisfy many objectives, such as connection and topology constraints, but also avoid the snake shape [2]. By integrating the description of Figure 15 and Figure 16, our algorithm is illustrated as bellows:

Algorithm CFMS
Input: *n* soft modules and *m* multiple-pin nets.
Output: A feasible floorplan with minimum total congestion and total wire-length.

Method:

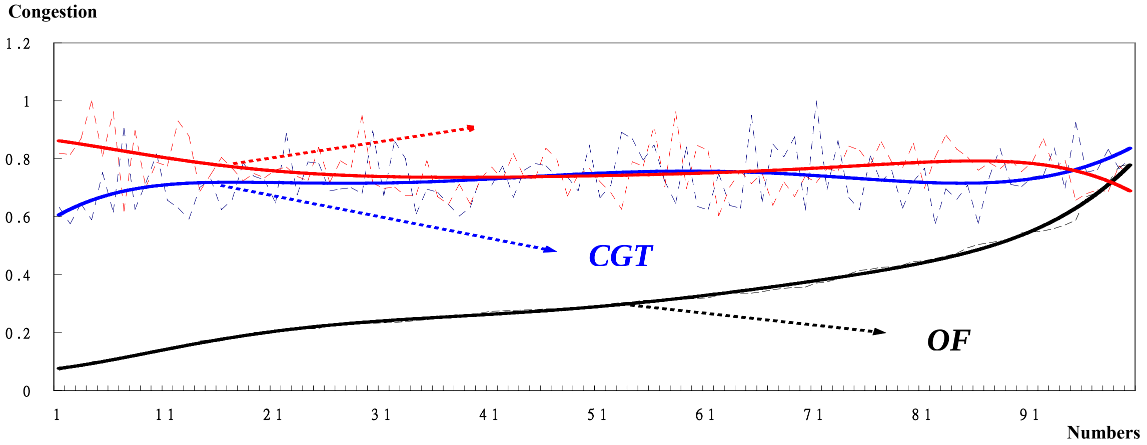
```
//Stage 1: Get a congestion-driven floorplan by SA
1.  Generate an initial floorplan.
2.  Decompose each multiple-pin net into a set of
    two-pin nets.
3.  Estimate area, wire length and congestion for a
    floorplan.
4.  do{
5.       Perturb the relationship of modules.
6.       Estimate area, wire length and congestion.
7.       }while(the constraints don't meet);
8.  for a best congestion-driven floorplan

//Stage 2: Apply modular shaping by NLP
9.  Apply modular shaping for current most
    congested region of the floorplan iteratively.
10. Output a floorplan until the congestion will
    not improve any more.
```

VI. Experimental Results

In this section, we test our congestion-driven floorplanner on MCNC benchmarks and all experiments were performed using an Intel 2.4GHz processor with 256MB memory. We use the linear/nonlinear program solver, LINGO 7 to solve the associated nonlinear problem for modular shaping. The objectives are to minimize wire congestion and total wire-length. Three sets of experiments were provided to show the accuracy and effect of our algorithm.

1. Accuracy of our new Congestion Model

In order to demonstrate the accuracy of our congestion model, we take *OF* as the congestion metric. We first design three floorplanner as follows: One is the grid-based congestion model with the grid size of $200 \times 200 \mu m^2$ [4], the other is the congestion model with the grid size of $10 \times 10 \mu m^2$ [19] and another is our new congestion model with the concept of IR-grids. Then we generate 100 floor plan solutions randomly. Thirdly, each floorplan is estimated by three congestion models individually. Finally, we



OF : A grid-based congestion model with size of 10 × 10 um^2.
CGT: Our irregular-regular grid based congestion model.
Grid_CGT: A grid-based congestion model with size of 200 × 200um^2.
Fig. 17 The congestion distribution of three congestion models.

compute the total congestion of 100 floorplan solutions. Figure 17 lists the congestion distribution curves, and the *x*-axis (*y*-axis) denotes the accumulated number of the floorplan (the accumulated normalized congestion). For three smooth fitting curves, we find both the *OF* and *CGT* are increasing while *grid_CGT* curve is decreasing during the range from 1 to 40 floorplans. Compared to *grid_CGT*, our algorithm is more accurate and efficient congestion model.

2. Effect of Our New Congestion Model

We design three floorplanners with different objectives. Table 3 summarizes the comparison of three congestion models. F1 is a floorplanner without considering congestion, F2 is based on the probabilistic analysis with considering congestion, and F3 is our new floorplanner. Each floorplanner is applied to generate 30 floorplan solutions for all benchmarks randomly. For fair comparison, the congestions of 90 floorplans are calculated by the same congestion model. The first column represents the names of three floorplanners and area (wire-length) is recorded in the second (third) column, respectively. The total congestion cost and improvement rate of a floorplan are shown in the fourth and fifth column, respectively. Moreover, the sixth column denotes the run time. The area is the product of width and height of feasible solution and the total wire-length is the sum of estimated wire-length of all 2-pin nets. Congestion cost is the *OF* value of the grid-based congestion model with grid-size of $10\times 10\text{um}^2$. We take apte circuit to show how the congestion improvement is computed. The percentage of congestion improvement of F2 (F3) is computed by the formula of (140.89-83.68) / 140.89 ((140.89-47.87) /140.89). We can find that our floorplanner (F3) obtains the lowest total congestion.

3. Post-process by Modular Shaping

We take the lowest congestion solution among 30 floorplans to show the effect of modular shaping. Table 4 gives us the comparisons of the approach without and with modular shaping. The first and second columns list the name and area of each benchmark, respectively. The third (fourth) column denotes the approach with (without) the modular shaping. In each sub-column, G and H (G′ and H′) represent total wire-length and total congestion cost of a floorplan before shaping (after shaping). From the fifth column, we conclude that our new model achieves an average 22.49% and 1.54% reduction in congestion and total wire-legth, respectively. This shows the effectiveness of the modular shaping. Because the largest amount improvement of congestion is ami33, we only show the effect of the modular shaping. Of course, we have the similar results for the other benchmarks. Figure 18(a) and 18(b) are the floorplan before and after applying the modular shaping, respectively. The white lines in Figure 18(c) and Figure 18(d) are the two-pin nets. We find that the conges tion in Figure 18(d) is better than Figure 18(c). The detailed procedure of modular shaping is shown in Figure 19 (a)-(g).

Table 3 Comparisons of three floorplanners.

TestCase	A	B	C	D	E
apte					
F1	48.66	253307	140.89	—	19.28
F2	50.93	268696	83.68	41%	25.53
F3	49.75	271054	47.87	66%	20.06
xerox					
F1	21.13	623679	5311	—	33.91
F2	21.53	644602	4639.6	13%	76.54
F3	21.28	652129	4392.19	17%	37.89
hp					
F1	11.07	138993	133.91	—	14.54
F2	13.15	158070	99.92	25%	22.7
F3	11.49	148779	88.09	34%	16.27
ami33					
F1	1.29	73563	1106.4	—	1812.34
F2	1.3	72018	1020.08	8%	1923.58
F3	1.31	71529	951.32	14%	1864.52
ami49					
F1	41.25	1072824	1123.91	—	1120.73
F2	41.42	1079264	994.03	12%	1590.76
F3	41.57	1095447	892.67	21%	1211.47

A=the area of a floorplan(mm^2), B= total wire-length (μm),
C= the congestion of a floorplan,
D=the percentage of congestion improvement, E=the run time(s).

F1: A traditional floorplanner without consideration of congestion.
F2: A grid-based congestion-driven floorplanner with consideration of congestion.
F3: Our IR-grid base congestion-driven floorplanner with consideration of congestion.

Table 4 Show the effective of the modular shaping for congestion.

	F	Before shaping		After shaping		Improvement (%)	
		G	H	G′	H′	G″	H″
apte	48.28	250374	10.08	250374	10.08	0.00 %	0.00 %
xerox	20.47	631638	3662.33	599395	2265.53	5.10 %	38.14 %
hp	10.21	140917	70.11	140861	64.76	0.04 %	7.63 %
ami33	1.29	66049	642.75	64674	268	2.08 %	58.30 %
ami49	41.27	1119793	818.5	1114746	749.87	0.45 %	8.38 %
average	-	-		-		1.54 %	22.49 %

F= the area of a floorplan, G = total wire-length of a floorplan before shaping, H = the congestion of a floorplan before shaping, G′ = total wire-length of a floorplan after shaping, H′ = the congestion of a floorplan after shaping, G″= the improvement of total wire-length, H″ = the improvement of congestion, *i.e.* G″= (G-G′)/G and H″= (H-H′)/H.

VII. Conclusion

We have developed a new congestion model which is embedded into simulated-annealing flow to obtain a high-quality floorplan. Then a nonlinear programming method is used to enlarge the length of common boundary to further minimize the wire congestion. The proposed method is accurate and efficient in reducing the congestion of a floorplan. Compared to the traditional method without consideration of the modular shaping, we show experimentally that our algorithm achieves an average reduction rate of 22.49% and 1.54% in total congestion and wire-length, respectively.

References

[1] A. B. Kahng, “Classical Floorplanning Harmful?,” in *Proc. ACM International Symposium on Physical Design*, pp. 207-213, 2000.

[2] C.H. Lee, W. Y. Fu, C. C. Chang and T. M. Hsieh, ”An Efficient Hierarchical Approach for General Floorplan Area Minimization,” in *Proc. ACM/IEEE Asia Pacific conference on Circuits and Systems*, pp.347-352, 2002.

[3] 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. ACM/IEEE Design Automation and Test in Europe Conference and Exhibition*, pp. 1100-, 2002.

[4] C. W. Sham and F. Y. Young, ”Routability-Driven Floorplanner with Buffer Block Planning,” in *Proc. ACM International Symposium on Physical Design*, pp.50-55, 2002.

[5] C. N. Chu and F. Y. Young, “Nonrectangular Shaping and Sizing of Soft Modules for Floorplan Design Improvement,” *IEEE Transactions on Computer-Aided D esign of Integrated Circuits and Systems*, pp.71-79, 2004.

[6] D. F. Wong and C. L. Liu, “A New Algorithm for Floorplan Design,” in *Proc. ACM/IEEE Design Automation Conference*, pp.261-267, 1986.

[7] D. F. Wong, F. Y. Young and H. H. Yang, ”On Extending Slicing Floorplans to Handle L/T-shpaed Modules and Abutment Constraints,” *IEEE Transaction on Computer Aided Design of Integrated Circuits and Systems*, pp.800-807, 2001.

[8] D. Mehta and N. Sherwani, “On the Use of Flexible, Rectilinear Blocks to Obtain Minimum-Area Floorplans in Mixed Block and Cell Designs,“ *ACM Transactions on Design Automation of Electronic Systems*, vol. 5, pp. 82-97, 2000.

[9] F. Y. Young, C. N. Chu, W. S. Luk and Y. C. Wong,” Handling Soft Modules in General Nonslicing Floorplan Using Lagrangian Relaxation,” *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, vol. 20, no. 5, pp. 687-692, 2001.

[10] G. M. Wu, Y. C. Chang and Y. W. Chang, “Rectilinear Block Placement Using B*-Trees,” *ACM Transactions on Design Automation of Electronic Systems,* vol. 8, no. 2, pp. 188-202, 2003.

[11] H. C. Lee, Y. W. Chang, J. M. Hsu and H. Yang, ”Multilevel Floorplanning/Placement for Large-Scale Modules Using B*-Trees,” in *Proc. ACM Design Automation Conference*, pp. 812-817, 2003.

[12] H. M. Chen, H. Zhou, F. Y. Young, D.F. Wong, H. H. Yang and N. Sherwani, “Integrated Floorplanning and Interconnect Planning,” in *Proc. IEEE/ACM International Conference on Computer Aided Design*, pp.354-357, 1999.

[13] H. Murata and E. S. Kuh, “Sequence-Pair Based Placement Method for Hard/Soft/Pre-placed Modules,” in *Proc. ACM International Symposium on Physical Design*, pp. 167–172, 1998.

[14] H. Murata, K. Fujiyoshi, S. Nakatake and Y. Kajitani, “Rectangle-Packing-Based Module Placement,” in *Proc. IEEE/ACM International Conference on Computer Aided Design*, pp. 472-479, 1995.

[15] J. Lou, S. Krishnamoorthy and H. S. Sheng, ”Estimating Routing Congestion Using Probabilistic Analysis,” in *Proc. ACM International Symposium on Physical Design*, pp. 112-117, 2001.

[16] J. M. Lin and Y. W. Chang, ”TCG: A Transitive Closure Graph-Based Representation for Non-Slicing Floorplans,” in *Proc. ACM/IEEE Design Automation Conference*, pp.764-769, 2001.

[17] J. Xu, P. N. Guo and C. K. Cheng, “Rectilinear Block Placement Using Sequence-Pair,” in *Proc. ACM International Symposium on Physical Design*, pp.173-178, 1998.

[18] M. Kang, and W. M. Dai, “General Floorplanning with L-shaped, T-shaped and Soft Blocks Based on Bounded Slicing Grid Structure,” in *Proc. ACM/IEEE Asia and South Pacific Design Automation Conference*, pp. 265-270, 1997.

[19] M. Wang and M. Sarrafzadeh, “On the Behavior of Congestion Minimization during Placement,” in *Proc. ACM International Symposium on Physical Design*, pp. 145-150, 1999.

[20] Takahashi T., C. K. Cheng and Yoshimura T, “Floorplanning Using a Tree Representation,” *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, vol.20, no.2, pp. 281-289, 2001.

[21] P. Sarkar and C. K. Koh, “Routability-Driven Repeater Block Planning for Interconnect-Centric Floorplanning,” in *Proc. ACM International Symposium on Physical Design*, pp.186-191, 2000.

[22] T. C. Wang and D. F. Wong, “Efficient Shape Curve Construction in Floorplan Design,” in *Proc. European Design Automation Conference*, pp. 356-360, 1991.

[23] X. Hong, S. Dong, G. Huang, Y. Ma, Y. Cai, C. K. Cheng and J. Gu, “A Non-Slicing Floorplanning Algorithm Using Corner Block List Topological Representation,” in *Proc. ACM/IEEE Asia Pacific Conference on Circuits and Systems*, pp. 833-836, 2000.

[24] Y. C. Chang, Y. W. Chang, G. M. Wu, and S. W. Wu, “B*-trees: A New Representation for Non-Slicing Floorplans,” in *Proc. ACM/IEEE Design Automation Conference*, pp. 458-463, 2000.

[25] Y. Feng, D. P. Mehta and H. Yang, ”Constrained “Modern” Floorplanning,” in *Proc. ACM International Symposium on Physical Design*, pp. 128-133, 2003.

[26] Y. L. Hsieh and T. M. Hsieh, “A New Effective Congestion Model in Floorplan Design,” in *Proc. ACM/IEEE Design Automation and Test in Europe Conference and Exhibition*, pp. 1204-1209, 2004.

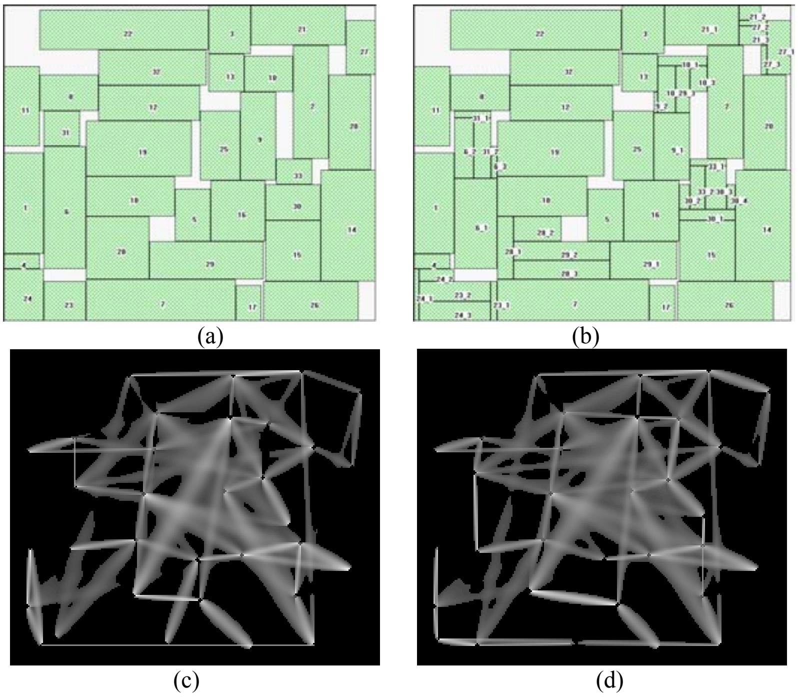
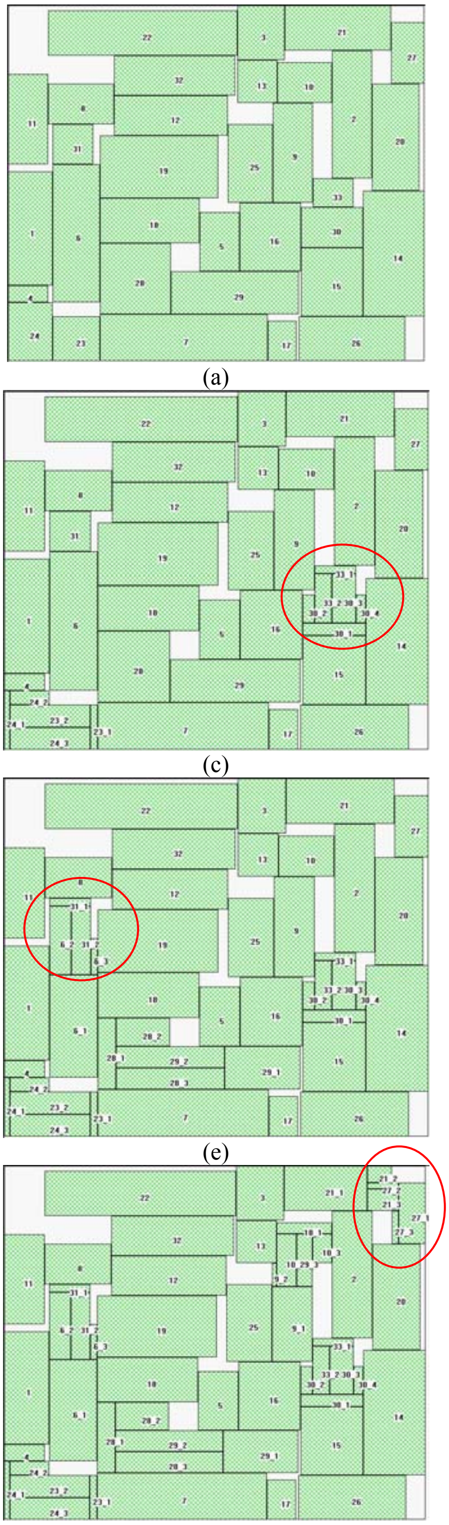

(a)
(b)
(c)
(d)

Fig. 18 The floorplan of benchmark ami33 (a) before and (b) after the modular shaping. The congestion distributions of benchmark ami33 (c) before and (d) after the modular shaping.



(a)

(b)

(c)

(d)

(e)

(f)

(g)

Fig. 19 Illustrations of performing modular shaping for ami33. (a) Original circuit. (b) We discover the common boundary between module 23 and 24 is the current most congested region by our congestion model. Therefore module 23 is divided into the set of sub-rectangles 23_1 and 23_2 and module 24 divided into the set of sub-rectangles 24_1, 24_2 and 24_3, respectively. In the similar way, we estimate the current most congested region and perform modular shaping for (c) Module 30 and 33 (d) Module 28 and 29 (e) Module 6 and 31(f) Module 9 and 10 (g) Module 21 and 27.