

An Enhanced Congestion-Driven Floorplanner

Yu-Cheng Lin¹ Shin-Jia Chen¹ Ping-Liang Chen¹ Hsin-Hsiung Huang²

¹Dept. of Information and Electronic Commerce, Kainan University, Taoyuan, Taiwan

²Dept. of EE. , Lunghwa Univ. of Science and Technology, Taoyuan, Taiwan

liny@mail.knu.edu.tw shinjia@mail.knu.edu.tw pingliang@mail.knu.edu.tw pp002@mail.lhu.edu.tw

Abstract: - In this paper, a rectilinear-based congestion-driven floorplanning algorithm is presented to enhance the wire congestion and the CPU runtime. The proposed algorithm contains two stages, including the simulated-annealing (SA) based approach with the concept of ant algorithm (SANTA) and the nonlinear programming based method. The objective of the first stage and the second stage are to minimize the multiple objectives, such as the area, wire length and wire congestion, and to further improve the wire congestion of the local congested region without the area overhead, respectively. First, the effective concept of the ant algorithm is integrated into the multiple objectives floorplanner, which simultaneously minimizes area, wire congestion and the total wire length, to speedup the runtime. Besides, the nonlinear programming (NLP) based formulations are provided to perform the module reshaping, which maximizes the common length between two adjacent congested modules. For the floorplanner, the sequential-pair (SP) presentation is utilized to deal with the floorplan data at every iteration. For each iteration of the floorplanner, we first use SANTA to improve the neighbor searching and reduce the runtime. After performing the first state, we will obtain a floorplan with the objectives of the area, wire congestion and total length. For the intermediate floorplan, we select the two adjacent soft modules located at the most congested regions and divide the two soft modules into a set of connected sub-rectangles. Hence, we further reduce the congestion by enlarging the common boundary between the selected adjacent modules. Of course, the modular reshaping technique significantly increases the common length and the capacity of pins to reduce the wire congestion by utilizing the nonlinear programming based approach. To deserve to be mentioned that there is no area overhead after we perform the modular reshaping for the selected adjacent modules. Compared to the results of the traditional SA, SANTA achieves an average improvement on the area, the total wire length and CPU runtime by 3% and 8.1% and 23.0%, respectively. To show the superior to our approach, 30 floorplan samples are randomly selected from Microelectronics Center of North Carolina (MCNC) benchmarks. The experimental result shows that the reshaping method improves the wire congestion and total wire length by 22% and 1.54%, respectively.

Key-Words: Simulated annealing, Ant algorithm, SANTA, NLP, Module reshaping, Floorplan

1 Introduction

As the technology enter the deep submicron era, the floorplanning, which determines the locations and shapes for a set of hard/soft modules in a chip, becomes more and more important to handle the serious issues, such as the congestion and the power[1][2], of the physical design. For the study of the floorplanner, some researches also explore the three dimension floorplanning [3] instead of the traditional two dimension floorplanning. Traditional methods mostly focus on the minimum of the area and the total wire-length, and the issues of the congestion, power and routability are usually not considered. In fact, as the wires between hard (soft) modules become longer and denser, to avoid the rough and unroutable floorplan, it is necessary to pay attention to the congestion as early as possible (the

floorplanning stage) to achieve the one-pass design without timing violations.

Simulated annealing, which is a popular and effective technique to get the optimal or sub-optimal solutions, is widely used in many applications [4][5][6][7]. [4] utilized the mixed methods, including the simulated annealing and Tabu Search, to obtain the global optimal solution. The simulated annealing is used to handle the multi-mode project scheduling in [5]. Sadegheih [6] applied the simulated annealing to deal with the large scale circuit. Sadegheih further integrated the simulated annealing and evolutionary algorithm to construct the spanning tree [7]. However, the concept of simulated annealing may search the solution with the redundant try-and-error within the user-defined constraints.

Congestion issues draw more and more attentions, especially the huge amount of interconnects are

appeared. Hence, the probability-based or grid-based estimators are provided to precisely predict the congestion distribution. Huang *et al.* presented the graph theory-based algorithms to even distribute the buffer congestion to maximize the success rate and inserted buffers [8]. Sham *et al.* [9] utilized the intersection-to-intersection model to optimize the congestion and the two-stage simulated annealing is integrated into the floorplanner. Kang *et al.* [10] provided the concept of the bounding box to estimate the wire congestion (an empirical model) and presented a modified objective function to improve the congestion. Hsieh *et al.* [11] provided the novel probability-based congestion model with the irregular-size grids to improve the disadvantages of the previous concept by using the fixed-size grids. It is obviously that there is a tradeoff between the efficiency and accuracy. It means that if we want to quickly estimate the congestion, the numbers of grids should be small. On the contrary, if we need an accurate, the numbers of grids in the estimator must be large, i.e. it is time consuming. So far, most methods spend the large time to estimate the wire congestion.

The soft modules are getting more and more important, especially when the SOC trend widely utilizes the pre-defined soft IP blocks. Most existing methods focus on the hard modules and some recent proposed algorithms try to handle the arbitrarily shaped rectilinear blocks. Mehta and Sherwani [12] proposed the area-driven floorplanner, which exploited the flexibility of the blocks, integrated the traditional method properly to minimize the area. Wong [13] *et al.* extended the Polish expression representation and their approach can handle both of the L-/T-shaped blocks effectively. Lee *et al.* [14] provided the novel concept of partition for blocks, which automatically determined the shape of blocks according to the user-defined optimized objectives. Chu and Young [15] also formulated the problem of the block reshaping and utilize the Lagrange relaxation to adjust the shaping of nonrectangular blocks. Xu *et al.* [16] proposed the approach to deal with the arbitrarily sized and shaped rectilinear blocks by using the sequence-pair (SP) representation. In this work, the properties of L-shaped blocks are first explored and the rectilinear blocks are divided into a set of L-shaped blocks. [17] *et al.* proposed the modular shaping and enlarged the common boundaries between the adjacent blocks to minimize the congestion. However, most of them just handle fixed-size blocks and the rectilinear modules are not handled easily during the iteration in their floorplanner.

The main contributions of this paper are as follows.

(1) Differ from the exhaustive searching of SA, our SANTA proposed the modified mechanism of searching the neighbor solution. Therefore, SANTA can run more efficiently and effectively than traditional SA. (2) For the intermediate floorplan, the module reshaping method is applied to improve the wire congestion. The congested regions between the two adjacent modules are divided into a set of sub-rectangles to enlarge the common boundary. Therefore, the total wire-length and wire congestion are further improved.

The rest of this paper is organized as follows. Section 2 describes the motivation of SANTA, the ant algorithm, a new congestion model and problem definition. Section 3 discuss the nonlinear programming-based constraints for the congestion improvement in detail. Section 4 presents the congestion-driven floorplanner. Experimental result and the conclusion are attached in Section 5 and 6, respectively.

2 Preliminary

In this section, we will discuss the motivation of SANTA, the ant algorithm, our congestion model and problem definition.

2.1 Motivation

We know that SA searches optimal the neighbor solution and accepts the sub-optimal solution according to the boltzmann's function, i.e. a certain probability. However, the SA algorithm obtains the optimization solution with the redundant searching. In order to make SA more efficient, we integrate SA with the concept of the ant algorithm. We call the proposed approach SANTA (SA with the concept of the ant algorithm), which searches the neighbor solutions more efficiently. It shows experimentally that the superior of SANTA lists in Section 4 and it obviously improves the results of the traditional SA.

2.2 The Ant Algorithm

Dorigo et al. presented the ant colony optimization (ACO) to solve the traveling salesman problem (TSP), which is defined as the problem of finding a simple cycle containing all nodes [18]. Alupoaei and Katkooari proposed the macro-cell overlap removal method which uses the concept of ACO [19]. The proposed method applies the graphs to obtain an overlap-free placement and a local optimization

procedure is used [19]. Wang *et al.* [20] presented MAX-MIN ant system optimization method to minimize the completion time and utilize effectively the computation resource. Hu *et al.* [21] proposed the practical Steiner tree construction, which applies the fast-ant strategy to speedup the runtime, to obtain the short total wire length. We know that the ACO search the solution space more efficient and effective because the algorithm can avoid the redundant solutions.

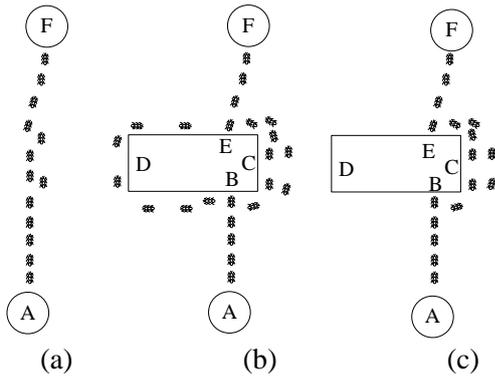


Fig.1 An example of ant walking.

Consider for example shown in Fig. 1. There is a path along which ants are walking from food source F to the nest A and an obstacle appears on the path. So at position B the ants walking from A to F (or at position E those walking in the opposite direction) have to decide whether to turn right or left. The ants select fairly the path A-B-C-E-F and A-B-D-E-F. As time goes by, the phomone of A-B-C-E-F becomes stronger due to the shorter distance. Finally, almost all ants walk through A-B-C-E-F and it is the better path. Dorigo *et al.*[18] presented solve the classic traveling salesman problem (TSP), which is defined as the problem of finding a simple cycle containing all nodes, by the ant colony optimization. The main steps of the ant algorithm are as follows.

2.2.1 Initialize

There are *m* ants are placed in randomly selected nodes and the phomone of each edge *e_{ij}* is set as follows.

$$\tau_{ij} = C; \quad \forall i, j \tag{1}$$

where *C* is constant and is the phomone intensity of the *e_{ij}*.

The visibility is defined as follows.

$$\eta_{ij} = 1/d_{ij}; \tag{2}$$

where *d_{ij}* is the distance of *e_{ij}*.

2.2.2 Probability of selecting path

We know that the large values τ_{ij} and η_{ij} have the larger priority. Therefore the probability from *i* to *j* is defined as the formula (3),

$$P_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha \times [\eta_{ij}(t)]^\beta}{\sum_{k \in allowed_k} [\tau_{ik}(t)]^\alpha \times [\eta_{ik}(t)]^\beta} & \text{if } j \in allowed_k \\ 0 & \text{others} \end{cases}$$

where *i* and *j* denotes the starting and ending points, *allowed_k* denotes the set of the unvisited nodes.

2.2.3 Update the phomone

In order to make the shortest path with the most phomone, the phomones of each path must be updated. The updating mechanism is divided into two kinds, including local and global update.

2.3 Review of the Congestion Model

The IR-grid congestion model [12] is applied to analyze the congestion of the floorplan of Fig. 2(a) and the following is the brief. For a pair of soft modules *b_i* and *b_j*, the routing region spanning by x-direction range from 1/5W to 4/5W and y-direction range form 1/5H to 4/5H is defined as the estimated wire congestion region, where W and H are the width and height of bounding box spanning by two connected modules of Fig.2(b), respectively.

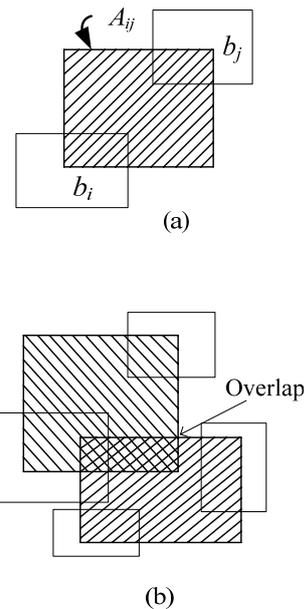


Fig. 2 (a) Routing region. (b) The estimated wire congestion region.

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 as follows,

$$C_{ij} = \begin{cases} \frac{n_{ij} - p_{ij}}{A_{ij}} & \text{if } n_{ij} - p_{ij} > 0 ; \\ 0 & \text{otherwise;} \end{cases} \quad (4)$$

where A_{ij} is the area of r_{ij} , n_{ij} is the number of two-pin nets between two connected modules b_i and b_j , and p_{ij} is the number of common pins along the common boundary. Besides, the overlapping of estimated wire congestion regions is also considered. Finally, the congestion of a floorplan is estimated by,

$$Cong = \sum_{i=1}^n \sum_{j=i+1}^n C_{ij} + \sum_{i,j,s,t} K_{ij,st} \quad (5)$$

where n denotes the number of soft modules, the first term and the second term denote the congestion of each two modules and the overlap region, respectively.

2.4 Problem Definition

According to the above discussion of the concept of ant algorithm, we define the problem, which is to simultaneously minimize the area, the wire congestion and the total wire length, as follows,

Given a set of the rectangular soft modules and a set of the multi-pin nets, for the current congested region, each soft module b_i is divided into a set of the connected sub-rectangles to optimize the total wire congestion and the wire length. Two symbols, h_i and w_i , denote the height and weight of the bounding box for these set of the connected sub-rectangles and the aspect ratio of the bounding box is defined as h_i/w_i , which is varied within the user-defined range $[r_{i_min}, r_{i_max}]$. A floorplan is regarded to be feasible if any two adjacent modules do not overlap with each other. Besides, the aspect ratio of each module without (and with) performing the modular reshaping must be under the user-defined range. Our objective is to get the feasible solution with minimal area, the congestion and total wire length.

After discussing the novel congestion model, we will describe how a NLP-based approach minimizes the wire congestion by adaptive modular shaping for two adjacent modules with the most congested value.

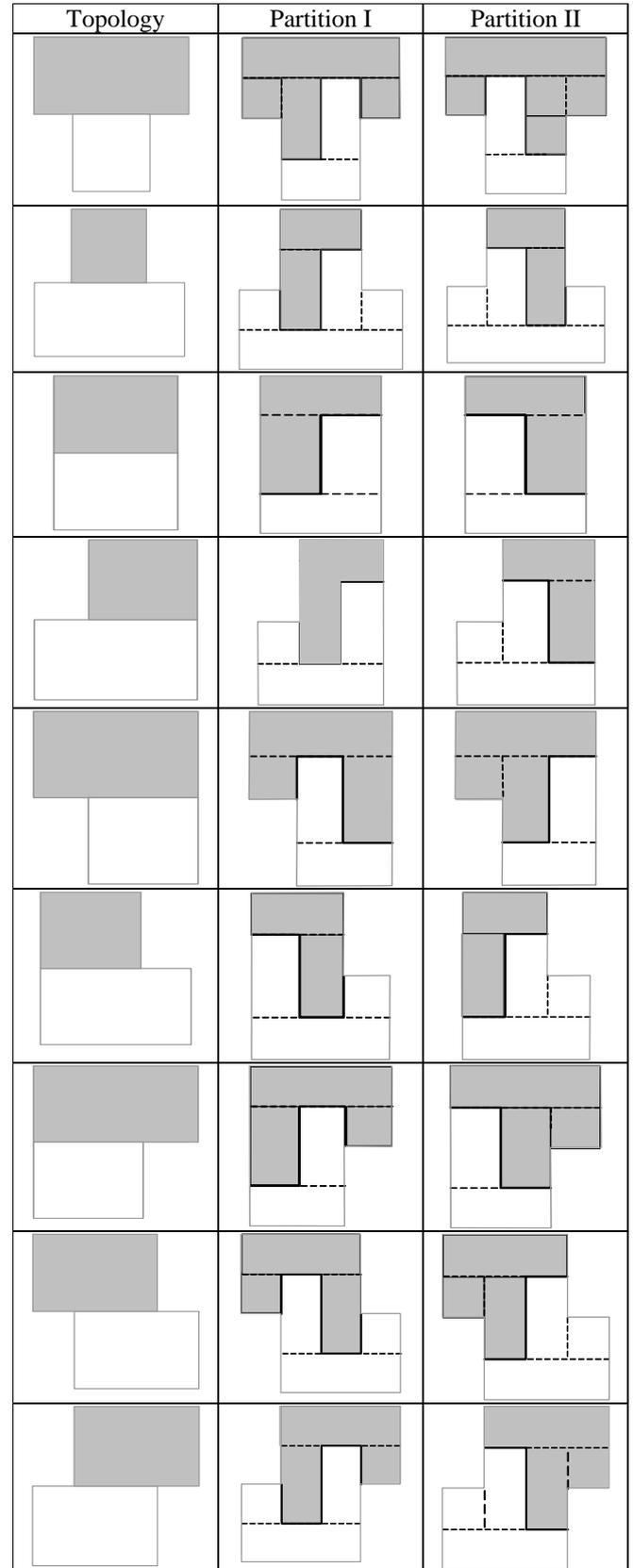


Fig. 3 Nine topologies and the pre-defined partitions.

3 Review of the Nonlinear Programming-based Module Reshaping

In this section, we first describe the objective function of our problem by using the nonlinear programming formulations. Then the basic constraints, area constraints, border constraints, connective constraints and topological constraints are described in detail. Finally, we will give a simple example to illustrate the procedure.

[17] formulated the wire congestion problem by using the NLP-based formulations and solve these constraints by the NLP solver, Lingo. To simplify, only eighteen pre-defined topologies, including the nine kinds of the horizontal and the nine kinds of the vertical relationships, are implemented in Fig. 3. At each iteration, we maximize the common length between two adjacent modules which are the current most congested region, i.e. our proposed NLP-based method is utilized to enlarge the common length between two adjacent selected modules and increase the capacity of terminals.

3.1 Objective Function

In this subsection, we'll discuss the objective function for the NLP-based modular reshaping. According to the above discussion, we enlarge the common length to further improve the local congested regions with the area penalty. The objective function is as follows,

$$\text{minimize } L_{com} \tag{6}$$

where L_{com} denote the common length between two selected modules which are the current most congested regions.

3.2 Basic Constraints

For two adjacent soft modules b_1 and b_2 in Fig. 4, the basic constraints for the modules width, height and the x -/ y -coordinates are described formulas (7)-(16). In this paper, we let (x_i, y_i) and $(\tilde{x}_i, \tilde{y}_i)$ denote the bottom-left and top-right coordinates of a set of soft modules. Furthermore, the values, h_i and w_i , denote are the height and width of the soft module b_i , respectively. In order to avoid the snake shaping, we also define the area ratio, which is determined by the height and width of the modules, and it must be within the given range, i.e. the upper bound U_i and lower bound L_i .

To simplify, we only use two modules to explain the constraints. For Fig. 4, the constraints (7)-(10) denote the relationship between the top-right point, left-bottom point, the width and the height of a soft module. Constraints (11)-(12) are the relationship between the module width, height and the module area. Constraints (13)-(16) represent the area ratio to avoid the snake shaping.

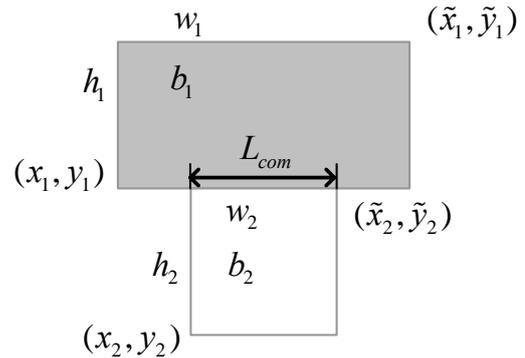


Fig. 4 The geometry of the adjacent modules.

$$y_1 + h_1 = \tilde{y}_1; \tag{7}$$

$$x_1 + w_1 = \tilde{x}_1; \tag{8}$$

$$y_2 + h_2 = \tilde{y}_2; \tag{9}$$

$$x_2 + w_2 = \tilde{x}_2; \tag{10}$$

$$a_1 = h_1 \times w_1; \tag{11}$$

$$a_2 = h_2 \times w_2; \tag{12}$$

$$r_1 = h_1 / w_1; \tag{13}$$

$$r_2 = h_2 / w_2; \tag{14}$$

$$L_1 \leq r_1 \leq U_1; \tag{15}$$

$$L_2 \leq r_2 \leq U_2; \tag{16}$$

3.3 Border Constraints

To simplify, only two adjacent modules are taken to reshape their module shape. Hence, to avoid overlapping the other neighbor modules, the reshaping results are not allowed to over the original border, which are the boundary for the rectilinear modules or original modules, see Fig. 5.

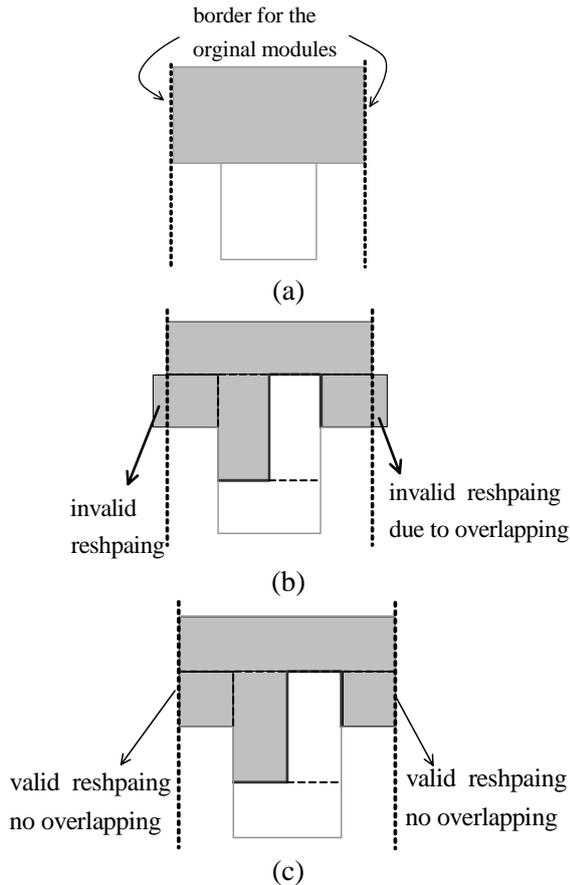


Fig. 5 Meet the border constraints without overlapping the neighbor modules.

3.4 Connective Constraints

To avoid the fragment for the set of connected sub-modules, we must make the connective constraints for the set of connected sub-modules after performing module reshaping. The connective constraints are made according the paper [16].

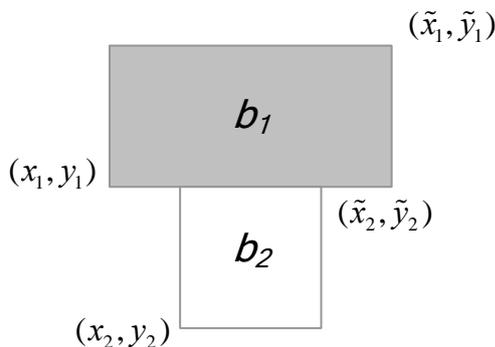


Fig. 6 Illustrations of connective constraints for the two adjacent modules with the vertical relationship.

$$\tilde{y}_2 = y_1; \tag{17}$$

where \hat{y}_2 and y_1 is the right-top and left-bottom y-coordinates of the blocks b_2 and b_1 .

3.5 Topological Constraints

To keep the locations between two adjacent modules, we must make the topological constraints. Generally speaking, the relationships between two adjacent modules can be described by the mathematical constraints. The topological constraints for two adjacent modules in Fig. 6 is formulated as follows,

$$\tilde{y}_2 \leq y_1 ; \tag{18}$$

where \hat{y}_2 and y_1 is the right-top and left-bottom y-coordinates of the blocks b_2 and b_1 .

3.6 Area Constraints

After performing the modular shaping, the total area of the set of connected sub-rectangular modules is equal to the original area. For the modules b_1 and b_2 in Fig. 5(c), which are the selected adjacent soft modules in congested region, are divided into a set of four connected sub-rectangles (b_{11}, b_{12}, b_{13} and b_{14}) and a set of two connected sub-rectangles (b_{21} and b_{22}). Two area constraints are as follows,

$$a_2 = a_{21} + a_{22}; \tag{19}$$

and

$$a_1 = a_{11} + a_{12} + a_{13} + a_{14}; \tag{20}$$

where a_2, a_{21} and a_{22} are the area of the soft module b_2 and a set of two connected sub-rectangles.

Summary, we formulate the wire congestion problem by using the NLP-based formulations. The wire congestion of the local congested region will be improved by solving the corresponding constraints to maximize the L_{com} without the area penalty.

4 The Congestion-Driven Floorplanner

The proposed congestion-driven floorplanner contains two parts, including a SANTA-based method and the NLP-based reshaping method. An area, wire length and wire congestion orientation floorplan is obtained in subsection 4.1 and the NLP-based reshaping discussed in subsection 4.2 is used to further improve the congestion.

4.1 The SANTA-based Floorplanner

For the non-slicing floorplan, the SANTA is applied to obtain a feasible floorplan with minimal area and the total wire length by using sequential-pair representation. Fig.7 shows the algorithm in detailed.

In subroutine `init_pheromone()`, we initialize parameters and pheromone trails. Pheromone trails will be updated at each stage of thermal equilibrium. For each floorplan, a heuristic minimal spanning tree is applied to divide a multi-pin net into a set of two-pin net. In annealing process, we simultaneously consider the area and total wire length. Hence the cost function of each floorplan is defined as.

$$cost = \alpha \times AREA + \beta \times WL + \gamma \times Cong; \quad (21)$$

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

4.2 Module Reshaping

Lee *et al.* [14] presented the concept of the block partition to reshape the modules. The method can minimize the wire congestion and total wire length simultaneously. It inspires us to develop the pre-defined topologies of non-rectangular shapes by the nonlinear programming. To simplify, we only design two pre-defined topologies for each modular relationship, including horizontal and vertical relation.

Each of the two selected adjacent soft modules at the congested region is divided into a set of connected sub-rectangles to increase the length of the common boundary between the adjacent modules. l and s are defined as the length of common boundary and sum of minimum wire width and spacing, respectively. Therefore, the routing capacity is computed as l/s . From the analysis, we know that the longer common boundary actually reduces the total wire length between the two modules and minimizes the wire congestion. The nonlinear programming method is used for module reshaping further minimize the wire congestion.

For each modular relationship, there are two types of module reshaping (Fig. 3 and Fig. 8(a)) which the length of original boundary between module b_j and b_i is $\{l\}$, where L_{com} is the length of common boundary. In Fig. 8(b), the length enlarges to $\{l + l_1 + l_2\}$, where l_1 is the height of module b_{j_2} (or b_{i_2}) and l_2 is the height of module b_{j_3} (or b_{i_4}). By the similar way, the length of the common boundary is obtained in Fig. 8(c). The total congestion of each floorplan is estimated and the solution with lower the wire congestion solution is accepted. Finally, the

reshaping method is iteratively performed for the current most congested boundary and the algorithm stops until the total congestion does not reduce any more.

```

Algorithm SANTA
typedef Sol class feasible_solution
bool accept(Sol f,Sol g) {
    double deltaC;
    deltaC=Cost(g)-Cost(f);
    if(deltaC<=0)
        return true;
    else
        return (exp(-deltaC/T))>rand(0,1);
}

Sol Gen_Neighbor(Sol f) {
    set<Sol> solSet = Neighbor(f);
    Sol s=Max_Pheromone(solSet);
    return s;
}

void SANTA() {
    Sol f,g;
    init_pheromone();
    double T=init_temperature();
    f=init_solution();
    do {
        do {
            g=Gen_Neighbor(f);
            if(accept(f,g))
                f=g;
        }while(!thermal_equilibrium());
        update_pheromone();
        T=T*r;
    }while(!stop());
    show_result();
}

```

Fig. 7 The pseudo code of SANTA.

5 Experimental Result

In this section, the two-stage congestion-driven floorplanner is tested on Microelectronics Center of North Carolina (MCNC) benchmarks and all experiments are performed using an Intel 2.4GHz processor with 256MB memory. The SANTA is implemented by C++ language. The linear/nonlinear program solver, LINGO 7, is used to solve the associated nonlinear problem for module reshaping. The objectives are to minimize wire congestion and total wire-length. The experiment result was provided to show the accuracy and effect of the proposed algorithm.

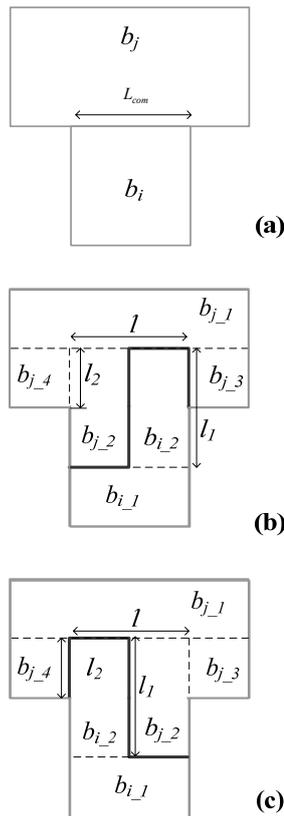


Fig. 8 (a) Vertical position of two connected soft modules. (b) one type of modular shaping. (c) the other type of modular shaping.

5.1 The Effect of SANTA Algorithm

In order to prove that SANTA could search the neighbor solution more efficiently than traditional SA, the same parameters as shown in Table 1 are applied to test the algorithm. Tables 2 and 3 show the improvement on the area, the total wire length and the runtime. The area, and total wire length and runtime is improved by 3%, 8.1% and 23.0%, respectively.

Table 1 The parameter.

Parameter	Value
Initial Temperature	0.9
Terminating Temperature	0.01
Temperature Decreasing Rate	0.95
Converge Rate	0.95
Times of Perturbation	50

5.2 The Result of Module Reshaping

Table 4 shows the comparisons of the approach without and with modular reshaping. In Table 4, the first and second columns denote the name and area of each benchmark, respectively. The third (fourth) column indicates the approach with (without) the modular reshaping. In the sub-columns, “G” and “H” represent the total wire-length and total congestion cost of a floorplan. We take the benchmark ami49 to show the result of reshaping, as shown in Fig. 9. From the table, it shows experimentally that the proposed new method achieves an average 22.49% and 1.54% reduction in wire congestion and the total wire-length, respectively. The runtime of module reshaping is very fast (less than 0.1 sec). Therefore, we skip the runtime column to save space. The result shows the effectiveness of the module reshaping.

Table 2 Runtime of SA and SANTA.

	runtime		
	A	B	I
apte	20.06	18	10.2
xerox	37.89	15	60.4
hp	36.67	26	29.1
ami33	1864.5	1572	15.7
ami49	1054	1058	-0.4
Avg.	602.6	537.8	23.0

A = SA, B= SANTA, I = the improvement (%)

6 Conclusion

The paper presents a two-stage congestion- driven floorplanner, which is the SANTA followed by the NLP-based module reshaping. SANTA is the approach, which integrates SA and the concept of the ant algorithm; it can obtain a high-quality floorplan. Besides, the nonlinear programming is used to further minimize the wire congestion by the module reshaping. Compared with the traditional SA-based algorithm, the SANTA-based method improves an average reduction of the area, the total wire length and runtime by 3% and 8.1% and 23.0% respectively. Besides, a NLP-based method is used to enlarge the length of common boundary to further minimize wire congestion. Compared to the traditional method without the module reshaping, it shows experimentally that the post-process result achieves an average reduction rate of 22% and 1.54% in total congestion and wire-length, respectively.

References:

- [1] Q. Ma and Evangeline F.Y. Young, Voltage Island-Driven Floorplanning, in *Proc. of ACM/IEEE International Conference on Computer Aided Design*, 2007, pp.644-649.
- [2] W.K. Mak and J.W. Chen, Voltage Island Generation under Performance Designs, in *Proc. of ACM/IEEE Asia and South Pacific Design Automation Conference*, 2007, pp.798-803.
- [3] R. Wang, F.Y. Young, Y. Zhu, F. Chung, G.R. Graham and C.K. Cheng, 3-D Floorplanning Using Labeled Tree and Dual Sequences, in *Proc. of ACM International Symposium on Physical Design*, 2008, pp. 54-59.
- [4] A. Sadegheih, Global Optimization Using Evolutionary Algorithms, Simulated Annealing and Tabu Search, in *WSEAS Transaction on Information Science and Applications*, 2004, pp 1700-1706.
- [5] H.Q. Pan and C.H. Yeh, Simulated Annealing for Multi-Mode Project Scheduling, in *WSEAS Transactions on Systems*, 2003, pp 681-687.
- [6] A. Sadegheih, Simulated Annealing in Large Scale Systems, in *WSEAS Transactions on Circuits and Systems*, 2004, pp 116-122.
- [7] A. Sadegheih, Evolutionary Algorithms and Simulated Annealing in the Topological Configuration of the Spanning Tree, in *WSEAS Transactions on Systems*, 2007, pp. 114-124.
- [8] H.H. Huang, Y.C. Chen and T.M. Hsieh, A Congestion-Driven Buffer Planner with Space Reservation, in *Proc. of IEEE International Symposium on Circuits and Systems*, 2006, pp. 5435-, 5438.
- [9] C.W. Sham and F.Y. Young, Routability Driven Floorplanner with Buffer Block Planning, in *Proc. of ACM International Symposium on Physical Design*, 2002, pp.190-195.
- [10] M. Kang and W.M. Dai, General Floorplanning with L-Shaped, T-Shaped and Soft Blocks based on Bounded Slicing Grid Structure, in *Proc. of ACM/IEEE Asia and South Pacific Design Automation Conference*, 1997, pp. 265-270.
- [11] Y.L. Hsieh and T.M. Hsieh, A New Effective Congestion Model in Floorplan Design, in *Proc. of Design, Automation and Test in Europe Conference and Exhibition*, 2004, pp. 1204-1209.
- [12] 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. 2000, pp. 82-97.
- [13] D.F. Wong, F.Y. Young, On Extending Slicing Floorplans to handle L/T-shaped Modules and Abutment Constraints, *IEEE Transactions on Computer-Aided Design*, 2001, pp.800-807.
- [14] C.H. Lee, W.Y. Fu, C.C. Chang and T.M. Hsieh, An Efficient Hierarchical Approach for General Floorplan Area Minimization, in *Proc. of ACM/IEEE Asia-Pacific conference on Circuits and Systems*, 2002, pp.347-352.
- [15] C.N. Chu and F.Y. Young, Nonrectangular Shaping and Sizing of Soft Modules for Floorplan-Design Improvement, *IEEE Transactions on Computer-Aided Design*, 2004, pp71-79.
- [16] J. Xu, P.N. Guo and C.K. Cheng, Rectilinear Block Placement using Sequence-Pair, in *Proc. of ACM Internal Symposium on Physical Design*, 1998, pp.173-178.
- [17] H.H. Huang, C.C. Chang, C.H. Lee. C.Y. Lin and T.M. Hsieh, Congestion-Driven Floorplanning by Adaptive Modular Shaping, in *Proc. of IEEE Midwest Symposium on Circuits and Systems*, 2005, pp. 1067-1070.
- [18] M. Dorigo, V. Maniezzo and A. Colorni, The Ant System: Optimization by A Colony of Cooperating Agents, *IEEE Transactions on Systems, Man and Cybernetics*, 1996, pp. 29-41.
- [19] S. Alupoaei and S. Katkoori, Ant Colony Optimization Technique for Macrocell Overlap Removal, in *Proc. IEEE International Conference on VLSI Design*, 2004, pp.963-968.
- [20] G. Wang, W. Gong, B. Derenzi, R. Kastner, Design Space Exploration using Time and Resource Duality with the Ant Colony Optimization, in *Proc. of ACM/IEEE Design Automation Conference*, 2006, pp. 451-454.
- [21] Y. Hu, X.L. Hong, Z. Feng, X. Hu and G. Yan, An Efficient Rectilinear Steiner Minimum Tree Algorithm based on Ant Colony Optimization, in *Proc. of IEEE International Conference on Communication, Circuits and Systems*, 2004, pp. 1276-1280.

Table 3 Comparison of SA and SANTA.

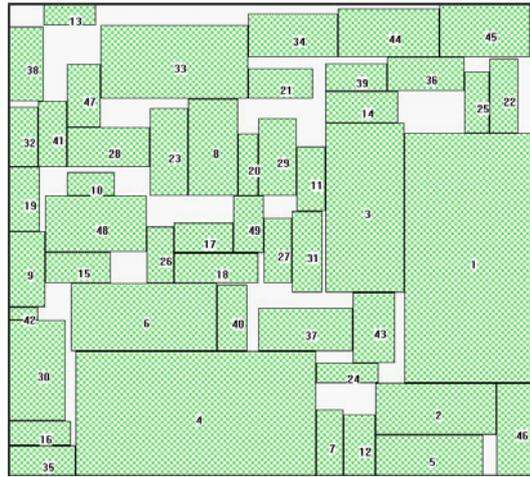
	area			total wirelength		
	A	B	I	A	B	I
apte	49.8	48.1	3.4	271054	238876	11.9
xerox	21.5	21.5	0	652429	613764	5.9
hp	10.1	9.3	7.9	140490	115766	17.6
ami33	13.1	13.0	0.8	71529	69464	2.9
ami49	41.4	40.2	2.9	1593800	1468972	7.8
Avg.	27.2	26.4	3.0	545860	501368	8.1

A = SA, B= SANTA, I = the improvement (%)

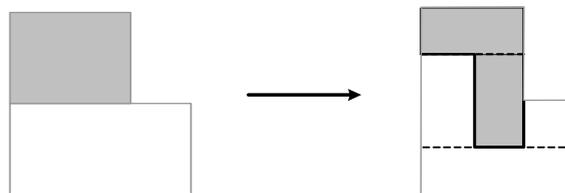
Table 4 Impact on reshaping for congestion.

Case	F	without shaping		with shaping		Imp(%)	
		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 %
Avg.						1.54 %	22.49 %

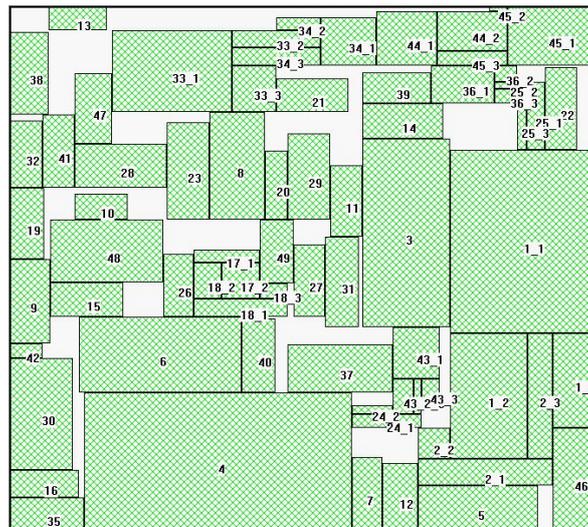
F=area(mm^2), G=wire length(μm), H=total congestion cost.



(a) The floorplan obtain without modular shaping



(b) Reshaping for the two adjacent modules 17 and 18.



(c) Floorplan with modular shaping

Fig. 9 (a) The floorplan for ami 49 without reshaping. (b) Take modules 17 and 18 for example. (c) Draw the final result.