Congestion-Driven Floorplanning with Module Reshaping

Yu-Cheng Lin  
Dept. of Information and Electronic Commerce, Kainan University, Taoyuan, Taiwan

Shin-Jia Chen  
Dept. of Information and Electronic Commerce, Kainan University, Taoyuan, Taiwan

Ping-Liang Chen  
Dept. of Information and Electronic Commerce, Kainan University, Taoyuan, Taiwan

Hsin-Hsiung Huang  
Dept. of Electronic Engineering, Chung Yuan University, Chung Li, Taiwan

Abstract: In this paper, we implement a two-stage congestion-driven floorplanner. In first stage, an area-wire length-oriented method using simulated-annealing (SA) with the conception of ant algorithm (SANTA) was introduced and a congestion-driven module reshaping was introduced in second stage. The floorplanner which simultaneously minimizes area, the wire congestion and the total wire-length generates the feasible solution. The objective of the paper is simultaneously area, the total wire length and the wire congestion. With the sequential-pair (SP) presentation, we first apply the SANTA improves the neighbor searching of traditional simulated-annealing (SA), to increase the solution quality and decrease the runtime. For the intermediate floorplan, the two adjacent soft modules located at the congested regions are selected and divided into a set of connected sub-rectangles to enlarge the common boundary between the adjacent modules. The longer common boundary actually reduces the total wire-length between the pins of two modules and minimizes the wire congestion. A nonlinear programming (NLP) method is applied for module reshaping mentioned above to further improve the congestion without the area penalty. Compared to traditional SA, SANTA can achieve an average improvement on area, total wire length and runtime by 3% and 8.1% and 23.0%, respectively. For the randomly selected 30 floorplan samples from Microelectronics Center of North Carolina (MCNC) benchmarks, the result also shows experimentally that the reshaping method obtains the improvement on the wire congestion and total wire length by 22% and 1.54%, respectively.

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

1 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. As the complexity of advanced VLSI technology is growing rapidly, the interconnections between modules become longer and denser. The previous floorplanners will produce easily an unroutable design because there is no routing resource in some congested regions at later stage. Hence, it is necessary to pay attention to the congestion minimum as early as possible to achieve the one-pass design flow without timing violation.

Recently, the congestion issue plays an important role in the nanometer technique and many methods are proposed to improve the congestion of a floorplan. Kang et al. [5] proposed an empirical model with the idea of bounding box to estimate wire congestion and present a new objective function to improve the wire congestion. Sham et al. [6] used a two-stage simulated annealing to optimize the congestion by using intersection-to-intersection model. Hsieh et al. [7] proposed a new irregular-size congestion model which is based on the probabilistic analysis, instead of the previous concept of the fixed-size grids. We know that there is a tradeoff between the accuracy and efficiency. When the numbers of grids are small, the result is inaccuracy. If the numbers of grids are large, the method is time consuming.

Most existing floorplanning method focus on the hard modules and some recent proposed algorithms can handle arbitrarily shaped rectilinear modules. Wong [8] et al. extended the Polish expression representation to handle both of the L- and T-shaped modules. Xu et al. [9] presented an approach, which extended the sequence-pair representation, deals with arbitrarily sized and shaped rectilinear blocks. The properties of L-shaped are first explored and the rectilinear blocks are divided into a set of L-shaped blocks. Lee et al.[10] introduced the concept of block partition that the shape of modules can be automatically determined according to the optimization objectives. Later, the modular shaping technique is presented to enlarge the common boundaries between two adjacent modules to minimize the congestion at the floorplanning stage [11]. Chu and Young [12] formulated the shaping problem and adjusted the shaping of nonrectangular by using Lagrange relaxation. Mehta and Shervani [13] presented some minimal-area floorplanners which exploited the flexibility, and integrated the
traditional method properly. However, most of them handle fixed-shaped modules and they assume that the rectilinear modules are not flexible in their shapes.

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 II describes the motivation of SANTA, the ant algorithm, a new congestion model and problem definition. Then the congestion-driven floorplanner is discussed in Section III. Experimental result is shown in Section IV and the conclusion is attached in Section V.

2 Preliminary

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

2.1 Motivation

SA searches the neighbor randomly and selects the sub-optimal solution with the certain probability. It means that the algorithm obtains the optimization solution by exhaustively searching. In order to make SA more efficient, we try to integrate SA with the concept of the ant algorithm. The proposed approach SANTA (SA with the conception of ant Algorithm) searches the neighbor solutions more efficiently. We will show experimentally the superior of SANTA in Section IV, and it obviously obtains the better result than 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[1]. Alupoei and Katkoori proposed the macro-cell overlap removal method which uses the concept of ACO. The proposed method applies the graphs to obtain an overlap-free placement and a local optimization procedure is used [2]. Wang et al. [3] presented MAX-MIN ant system optimization method to minimize the completion time and utilize effectively the computation resource. Hu et al.[4] 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.

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 pheromone 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. 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 pheromone of each edge \( e_{ij} \) is set as follows.

\[
\tau_{ij} = C, \quad \forall i, j
\]

where C is constant and is the pheromone intensity of the \( e_{ij} \).

The visibility is defined as follows.

\[
\eta_{ij} = 1/d_{ij}
\]

where \( d_{ij} \) is the distance of \( e_{ij} \).

2.2.2 Probability of selecting path

We know that the large values \( \tau_{ij} \) and \( \eta_{ij} \) have the larger priority. Therefore the probability form \( i \) to \( j \) is defined as follows.
2.3 The Congestion Model

Following is the brief. For a pair of soft modules $bi$ and $bj$, 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.

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

$$C_{ij} = \begin{cases} \frac{n_i - p_i}{A_{ij}} & \text{if } n_i - p_i > 0; \\ 0 & \text{otherwise}; \end{cases}$$

where $A_{ij}$ is the area of $rij$, $n_{ij}$ is the number of two-pin nets between two connected modules $bi$ and $bj$, 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:

$$CGT = \sum_{i=1}^{n} \sum_{j=1}^{i} C_{ij} + \sum_{a,j=1}^{n} K_{y,a}$$

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

The problem which is solved in this paper is to minimize the area, the congestion and total wire-length. It is defined as follows.

Given a set of the n rectangular soft modules $\{bi, bj, ..., bn\}$ and a set of the multi-pin nets. The area of the module $bi$ is $Ai$. The soft module $bi$ can be divided into a set of the sub-rectangles to optimize the total wire-length and the wire congestion. $hi$ and $wi$ are the height and weight of the bounding box of these connected sub-rectangles. The aspect ratio of bounding box is $h_i/w_i$ and can be varied within the aspect ratio $[ri_{min}, ri_{max}]$. A floorplan is feasible if and only if no two adjacent modules overlap with each other, and the aspect ratio of each module after reshaping if necessary must be under the given range. Our objective is to obtain a feasible solution with minimal area, the wire congestion and the total wire length.

3 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 and wire length orientation floorplan is obtained in subsection 3.1 and the NLP-based reshaping discussed in subsection 3.2 is used to further improve the congestion.

3.1 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.3 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 will be updated at each stage of thermal equilibrium. Parameters and pheromone trails are applied to obtain a feasible floorplan with minimal area, the wire congestion and the total wire length. Hence the cost function of each floorplan is defined as:

$$cost = \alpha \cdot AREA + \beta \cdot WL$$

where $AREA$ is the area of the floorplan, $WL$ represents the estimated total wire length, $\alpha$ and $\beta$ are the user defined parameters.

3.2 Module Reshaping

Lee et al. [10] 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.

![Algorithm SANTA](image)

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 \( \frac{l}{s} \). 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.4(a)) which the length of original boundary between module \( b_j \) and \( b_i \) is \( \{l\} \), where \( l \) is the length of common boundary. In Fig.4(b), the length enlarges to \( \{l_1 + l_2\} \), where \( l_1 \) is the height of module \( b_{j_1} \) (or \( b_{i_1} \)) and \( l_2 \) is the height of module \( b_{j_2} \) (or \( b_{i_2} \)). By the similar way, the length of the common boundary is obtained in Fig.4(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.

![Fig.4](image)

4 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.

4.1 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. Table 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>
<thead>
<tr>
<th>Table 1 The parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter</td>
</tr>
<tr>
<td>Initial Temperature</td>
</tr>
<tr>
<td>Terminating Temperature</td>
</tr>
<tr>
<td>Temperature Decreasing Rate</td>
</tr>
<tr>
<td>Converge Rate</td>
</tr>
<tr>
<td>Times of Perturbation</td>
</tr>
</tbody>
</table>

4.2 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 each subcolumn, “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. 4. 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 Comparison of SA and SANTA.

<table>
<thead>
<tr>
<th></th>
<th>area (mm^2)</th>
<th>total wirelength (μm)</th>
<th>G</th>
<th>H</th>
<th>G’</th>
<th>H’</th>
<th>G''</th>
<th>H''</th>
</tr>
</thead>
<tbody>
<tr>
<td>apte</td>
<td>49.8</td>
<td>271054</td>
<td>238876</td>
<td>11.9</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>xerox</td>
<td>21.5</td>
<td>652429</td>
<td>613764</td>
<td>5.9</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>hp</td>
<td>10.1</td>
<td>140490</td>
<td>115766</td>
<td>17.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ami33</td>
<td>13.1</td>
<td>71529</td>
<td>69464</td>
<td>2.9</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ami49</td>
<td>41.4</td>
<td>1593800</td>
<td>1468972</td>
<td>7.8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Avg.</td>
<td>27.2</td>
<td>545860</td>
<td>501368</td>
<td>8.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3 Runtime of SA and SANTA

<table>
<thead>
<tr>
<th></th>
<th>runtime (sec)</th>
<th>A</th>
<th>B</th>
<th>I</th>
</tr>
</thead>
<tbody>
<tr>
<td>apte</td>
<td>20.06</td>
<td>18</td>
<td>10.2</td>
<td></td>
</tr>
<tr>
<td>xerox</td>
<td>37.89</td>
<td>15</td>
<td>60.4</td>
<td></td>
</tr>
<tr>
<td>hp</td>
<td>36.67</td>
<td>26</td>
<td>29.1</td>
<td></td>
</tr>
<tr>
<td>ami33</td>
<td>1864.5</td>
<td>1572</td>
<td>15.7</td>
<td></td>
</tr>
<tr>
<td>ami49</td>
<td>1054</td>
<td>1058</td>
<td>-0.4</td>
<td></td>
</tr>
<tr>
<td>Avg.</td>
<td>602.6</td>
<td>537.8</td>
<td>23.0</td>
<td></td>
</tr>
</tbody>
</table>

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

Table 4 Impact on reshaping for congestion

<table>
<thead>
<tr>
<th>Case</th>
<th>F</th>
<th>Before shaping</th>
<th>After shaping</th>
<th>Imp(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>apte</td>
<td>48.28</td>
<td>250374</td>
<td>10.08</td>
<td>250374</td>
</tr>
<tr>
<td>xerox</td>
<td>20.47</td>
<td>631638</td>
<td>3622.33</td>
<td>599395</td>
</tr>
<tr>
<td>hp</td>
<td>10.21</td>
<td>140861</td>
<td>70.11</td>
<td>140917</td>
</tr>
<tr>
<td>ami33</td>
<td>13.29</td>
<td>6427.54</td>
<td>734.07</td>
<td>6465.58</td>
</tr>
<tr>
<td>ami49</td>
<td>41.27</td>
<td>1119793</td>
<td>818.5</td>
<td>1114746</td>
</tr>
<tr>
<td>Avg.</td>
<td>27.2</td>
<td>545860</td>
<td>501368</td>
<td>8.1</td>
</tr>
</tbody>
</table>

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

5 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:


