Global Density Smoothing Technique for Analytical Placement algorithm

Dawei Liu, Qiang Zhou, Jinian Bian, Yanming Jia
Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China
Email: liu-dw07@mails.tsinghua.edu.cn

Abstract

Cell migration has been widely used in global placement for the highly efficiency in smoothing cells overlap. The current cell migration methods only locally or globally smooth the density without considering the relation between the local and the global density. Furthermore, the cells generally are treated as points with area. In this paper we present a new cell migration technique called CSAGO to even the overlap. CSAGO obtains the movement distance while considering the local and the global density distribution simultaneously. By separating the standard cells and the macro blocks crossing multi-bin, a more smooth migration speed can be obtained. CSAGO has been embedded into the global placement process, experimental result shows that average HPWL have reduced 3% and the total runtime is 1.78 times faster than DPlace [1].

1. Introduction

Generally we start with a overlap placement that has been optimized or initial, by the cell migration a more even distribution can be get, then the optimization process is performed to get a new distribution and then continue the next optimization. This process runs iteratively until reaches the target overlap. Given the different locations of the cells from cell migration, the different optimization result can be achieved, so the technique of density smoothing has a great impact on the global placement. Two kinds of cell migration techniques are used in the relevant placements.

Fix-points Insertion Placement

Three similar cell migration techniques [2]–[4] have been implemented in placement. By adding fix-points, they prevent the cells collapsing back when using the quadratic equation to optimize the wire length. Fastplace [2] first proposes cell migration(cell shifting) techniques by even densities between neighboring bins. Diffusion [3] is another form of cell migration while being used in the global placement. The process of diffusion is driven by the density gradient, and evens out the densities between neighboring bins by using a discrete solver to solve the diffusion equation. The integrity of the original placement must be preserved and do less damage to the relative locations of the cells [3] in the cell migration process. In Fastplace by averaging the density of bin\(_i\) and bin\(_{i+1}\), the new virtual boundaries of the bins is constructed, then maps the cell location in a unequal bin structure in order to move the cells from a high density bin to a low one.

In diffusion the cells in the same bin are assigned to the same velocity computed by local density gradient after the bin density of step \(n+1\) is smoothed by the Forward Time Centered Space(FTCS). By velocity interpolating, diffusion could generate more smooth velocities because the cells next to each other but in different bins can be assigned to quite different velocities, which could change their relative order.

Diffusion is used in RQL [5] and DPlace [1] to even out the density in global placement. RQL is an advanced version of Fastplace, after performing a series diffusion to get a more even density DPlace solves the quadratic program to optimize the wire length. In mFar [4] the virtual bin boundary is determined by the the global cell flow crossing the bin boundary. The cells are also shifted by mapping the present locations of the cells according to the virtual bin boundary.

The cell migration techniques in [2], [3] have the common drawback that they locally smooth the density (only related with the neighboring bins) without considering the relation between the local and the global density. In many cases the local movement direction of the cell may be not consistent with the global optimized direction.

As shown in Figure 1, according to diffusion the cell migration direction of bin(4,4) and bin(3,4) is to bin(5,3), but from whole distribution of the cells their migration direction should be to bin(3,3). If the cells of bin(4,4) and bin(3,4) is shifted to bin(5,3), in the subsequent iterations we have to
relocate these cells back to the low density bins. This kind
of cell migration not only partly loses the original relative
order but also increases the placement time in the view of
the whole global placement process.

In mFar the global migration of the cell does not
consider the relation between the cell migration speed and
the local density distribution. Only global migration the cell
may damage the integrity of original placement because the
uneven density distribution could cause the uneven speeds
especially between neighboring bins.

**Force Directed Placement**

In [10], Eisenmann and Johannes introduces the forces
to push cell from the high density to the low regions. The
additional vector of forces are derived from the density
distribution of cells. Based on [10], FDP [9] calculates
the spreading forces by a Barnes-Hut quad-tree for n-body
force calculation. In kraftwerk [8], the force is computed
by Poisson potential and separates into a hold force and a
move force. These technique of force can be regarded as
another form of cell migration. This migration based on
the force during placement is calculated by the whole cell
distribution, but it is harder to control the force than the
migration methods in Fixpoints Insertion Placement.

In this paper, we present a new cell migration algorithm
called CMAGO(Cell Migration Aware of the Global Over-
lap). The main contributions of CSAGO are as follows:

**Global module migration direction and local migration
distance.**

Combining the advantages the cell migration methods of
the Fixpoints Insertion and Force Directed Placement, We
give module migration direction by global density distribu-
tion. The difference value of the average densities of bins
between two sides of the bin determines the migration
direction of the boundary, but the actual migration distance
is generated by the local density gradient.

**Discretized cell by bin boundary.**

Most of the cell migration techniques suppose that the
standard cells be movable points with area in global place-
ment and it will cause some errors because many cells may
cross multi-bin in placement but their migration speed is
obtained only by the density gradient between neighboring
bins.

Furthermore, the modern large scale placement always
involves a lot of movable macro cells [6], [7] and how to
shift these macros will have much effects on the placement
objective.

We separate the macro blocks and the standard cells by the
crossing boundaries of the bins. These discrete sections are
processed as common cells at first, and the actual migration
distance of the separated cell is obtained by summing the
migration vectors of its sections at last.

The remainder of this paper is organized as follows:
Section 2 describes the detail of CSAGO. Section 3 and

---

**Algorithm 1: CSAGO algorithm**

1. Discrete the cells and update the density of the bin
2. Update the average densities of the bin sides
3. Determine migration direction of the modules in bin,
4. Compute local boundary migration distance
5. if the cell has been discreted
6. Sum the vectors of its modules multiplied by area ratio

Figure 2. Algorithm for CSAGO

4 give the experiment of CSAGO. Finally, conclusion is
presented in Section 5.

**2. Cell Migration Aware of the Global Overlap**

Initially, the placement region is divided into equal sized
bins. Each bin can accommodate about 3 ~ 4 cells [2]. The
cells whose width or height is less than the average width
or height are set the average width or height. For x and y
dimensions of the placement is independent, the following
only describe the case in the x dimension. Let N,M be the
row and column size, A be the area of bin, Let A be the
area of cellk and Ψ be the set of cells to placement. The
total area of the modules inside bin, can be given by

\[
U_i = \sum_{k \in \Psi} (A_k \cap A_i)
\]

The density of bin, is given by

\[
D_i = U_i / A_i
\]

Many cells (especially the macro) may cross multi-bin during
the global placement, and in the following we have separate
these cells into several sections by the boundaries of bins,
otherwise if we only treat such cells as points with area some
information of the distribution would be lost. (The section
and the unseparated cell is called module in the following.)
The algorithm 1 illustrates the process of CSAGO.

**2.1. Module Migration Direction**

First of all, CSAGO require the modules be assigned a
definite migration direction. In [4], the cell flow crossing
the bin boundary is given by the difference value between
the partial row capacity on the left and right sides of the bin
boundary. The volume of the cell flow is transformed into the
migration distance of the bin boundary and the cell location
is mapped according to the changed virtual bin boundary and
the positive and negative volume of the cell flow determine
the direction of the cell migration.

In CSAGO the module migration direction is determined
by the difference value between the average densities of the
bins on two sides of column or row that bin, belongs.
Let $D_{li}$ be the average density of the bins in the left column $i (i > 1)$ and $D_{ri}$ be the average density of the bins in the right column $i (i < N)$. They are given by

$$
D_{li} = \gamma \cdot \frac{\sum_{r=1, r \neq i}^{M} \sum_{c=1}^{i-1} D_{rc} + (1 - \gamma) \sum_{c=1}^{i} D_{rc}}{(M-1)(i-1)} + \frac{(1 - \gamma) \sum_{c=1}^{i} D_{rc}}{(i-1)}
$$

$$
D_{ri} = \gamma \cdot \frac{\sum_{r=1, r \neq i}^{M} \sum_{c=i+1}^{N} D_{rc} + (1 - \gamma) \sum_{c=i+1}^{N} D_{rc}}{(M-1)(N-i)} + \frac{(1 - \gamma) \sum_{c=i+1}^{N} D_{rc}}{(N-i)}
$$

The model of migration direction is illustrated in Figure 3. The parameter $\gamma$ is used to adjust the ratio in the up and down of row $j$, in our experiments $\gamma$ is set to be 0.4. We set the $+X$ axis is the positive migration direction and the $-X$ axis is negative (to left).

Here, the cell migration can even out overlap and is favorable for density distribution of the whole placement. Perhaps, the opposite neighboring bin may has a lower density than the bin, migration the modules toward this bin could also probably reduce the overlap locally, but for the global distribution this movement would worsen the overlap because of the difference of density $|D_{li} - D_{ri}|$ becoming larger.

### 2.2. Local Overflow Distance

Figure 4 illustrates the three cases of the density relation between $bin_i$ and his neighbor in the direction of movement. In Figure 4, $D_i > D_{i+1}$, if we assume that the modules of $bin_{i+1}$ is still and only the modules of $bin_i$ are permitted to shift, the boundary migration distance is given by:

$$
\vec{F}_i = \frac{U_i - U_{i+1}}{2 \cdot w} \cdot \alpha 
$$

$w$ denotes the width of the bins. $\alpha$ is used for controlling the movement speed. When the migration direction is to left and $D_i > D_{i-1}$, $\vec{F}_{i-1}$ is given by:

$$
\vec{F}_{i-1} = -\frac{U_i - U_{i-1}}{2 \cdot w} \cdot \alpha
$$

Here we locally optimize the overlap. In Figure 4, if $D_i \leq D_{i+1}$, we set the cells still ($\vec{F} = 0$) because their movement cannot even out the overlap locally, conversely will worsen the local overlap. This overflow rule can also prevent two neighboring bins move cells toward each other.

We also change precondition of (1)/(2) to get another version of more rapid local migration distance. Supposing the modules of $bin_{i+1}$ has been shifted instead of still the new equation be given by

$$
\vec{F}_i = \frac{U_i - U'_{i+1}}{2 \cdot w} \cdot \alpha 
$$

$U'_i$ can be given by the following recursive equation:

$$
U'_i = \frac{U_i + U'_{i+1}}{2} 
$$

From the still bins or enforced setting the bin to be still we can get the initial $U'_i$.

By integrating local boundary migration and global module migration direction, the local density smoothing is consistent with the global optimized objective.

In order to lessen the damage to the wire length the cell migration speed must be “smooth”. Especially during initial placement the density difference between bins may be very large, which can cause $|\vec{F}_i| \gg w$ and the larger migration distance can damage the wire length objective. Therefore, besides using $\alpha$ to adjust the speed we must limit the module overflow maximum speed according to the speed of placement.

### 2.3. Cell Mapping

Let $B_i$ be the right boundary location of $bin_i$ and $B'_i$ be the boundary location of $bin_i$ after boundary migration. Let $x$ and $x'$ be the location of the module before and after boundary migration. We map the location by the bin boundary and virtual bin boundary.

$$
\frac{x - B_{i-1}}{B'_i - x'} = \frac{B_i - x}{B'_i - x'}
$$
Because can preserve the integrity of the original placement greatly.

In one pass instead of more than once in [1] for CSAGO

5 CPlace.In algorithm 2 we only perform lines

From equation (1, 2, 4) or (3, 4) we can get the new location:

\[ x' = \frac{x(w + \vec{F}_1 - \vec{F}_{i-1} + B_i \cdot \vec{F}_{i-1} - \vec{F}_1 \cdot B_{i-1})}{w} \]

The vector of the migration distance is given by:

\[ \vec{f}_k = x' - x \quad (x' > x) \]
\[ \vec{f}_k = x' - x \quad (x' < x) \]

If the cell \( k \) crosses multi-bin its actual migration distance is given by

\[ \sum_{i \in B} \frac{A_k \cap A_i}{A_i} \vec{f}_k \]

B is the set of bins.

3. CSAGO based placement

CSAGO is another form of cell-migration technique [2], and given some change CSAGO can be used in many cases(e.g. [1], [2], [4]). For testing the efficiency of CSAGO, we embed it into our placer named CPlace.

FastPlace,RQL,mFar and DPlace has the common character that they add or fix a number of cells or points to prevent cells collapsing back due to the quadratic program. FastPlace,RQL and mFar add fixed cells to keep the shifted cells under force equilibrium. DPlace fixes the virtual cells which are in the center of gravity of all the pins of the state-of-art academic placers(from [15])on the ISPD 2005 placement contest.

We develope the legalization and detailed placement base on FDP [9]. Initially the macros are located legally using the sequence pair analysis [13] and then perform greedy cell swapping between rows in a sliding-windows to fit the cell in an optimized position.

4. Experiments

We test CPlace on ISPD 2005 [6] benchmark. Table 1 give the ratio of the time of CSAGO to the total run time. The average CSAGO takes only 3.91% of the total runtime and it demonstrates the high efficiency of the CSAGO technique.

Table 1. Ratio of the time of CSAGO to the total run time

<table>
<thead>
<tr>
<th>Circuit</th>
<th>RQL</th>
<th>CPlace</th>
<th>DPlace</th>
</tr>
</thead>
<tbody>
<tr>
<td>adaptec1</td>
<td>1.82</td>
<td>1.01</td>
<td>1.01</td>
</tr>
<tr>
<td>adaptec2</td>
<td>2.21</td>
<td>1.04</td>
<td>1.01</td>
</tr>
<tr>
<td>adaptec3</td>
<td>2.79</td>
<td>1.06</td>
<td>1.02</td>
</tr>
<tr>
<td>adaptec4</td>
<td>4.02</td>
<td>1.07</td>
<td>1.04</td>
</tr>
<tr>
<td>bigblue1</td>
<td>3.28</td>
<td>1.06</td>
<td>1.05</td>
</tr>
<tr>
<td>bigblue2</td>
<td>3.56</td>
<td>1.09</td>
<td>1.05</td>
</tr>
<tr>
<td>bigblue3</td>
<td>2.78</td>
<td>1.11</td>
<td>1.05</td>
</tr>
<tr>
<td>bigblue4</td>
<td>4.37</td>
<td>1.13</td>
<td>1.05</td>
</tr>
<tr>
<td>average</td>
<td>3.91</td>
<td>1.07</td>
<td>1.03</td>
</tr>
</tbody>
</table>

In lines 8 ~ 11 we use Iterative Local Refinement (ITL) [2] and BoxPlace [9] [12] to improve the cell order and wire length and it similar to DPlace.

We develop the legalization and detailed placement base on FDP [9]. Initially the macros are located legally using the sequence pair analysis [13] and then perform greedy cell swapping between rows in a sliding-windows to fit the cell in an optimized position.

Table 2. Wirelength comparison to placer in ISPD2005 placement contest

<table>
<thead>
<tr>
<th>Circuit</th>
<th>RQL</th>
<th>CPlace</th>
<th>DPlace</th>
</tr>
</thead>
<tbody>
<tr>
<td>adaptec1</td>
<td>81.68</td>
<td>0.95</td>
<td>1.01</td>
</tr>
<tr>
<td>adaptec2</td>
<td>90.53</td>
<td>1.03</td>
<td>1.02</td>
</tr>
<tr>
<td>adaptec3</td>
<td>224.60</td>
<td>0.95</td>
<td>1.04</td>
</tr>
<tr>
<td>adaptec4</td>
<td>192.44</td>
<td>1.02</td>
<td>1.04</td>
</tr>
<tr>
<td>bigblue1</td>
<td>95.07</td>
<td>1.00</td>
<td>1.06</td>
</tr>
<tr>
<td>bigblue2</td>
<td>165.24</td>
<td>0.93</td>
<td>1.05</td>
</tr>
<tr>
<td>bigblue3</td>
<td>376.32</td>
<td>0.96</td>
<td>1.02</td>
</tr>
<tr>
<td>bigblue4</td>
<td>910.20</td>
<td>0.91</td>
<td>1.02</td>
</tr>
<tr>
<td>average</td>
<td>1.07</td>
<td>1.09</td>
<td>1.03</td>
</tr>
</tbody>
</table>

boundary migration distance parameter \( \alpha = 1 \) and the max boundary migration distance less than \( w \).

In lines 8 ~ 11 we use Iterative Local Refinement (ITL) [2] and BoxPlace [9] [12] to improve the cell order and wire length and it similar to DPlace.

We use the hybrid model for the multi-pin nets in CPlace.In algorithm 2 we only perform lines 5 ~ 6 once in one pass instead of more than once in [1] for CSAGO can preserve the integrity of the original placement greatly. Because CPlace needs the cell rapidly spreading we set the boundary migration distance parameter \( \alpha = 1 \) and the max boundary migration distance less than \( w \).
Table 3. Runtime comparison to placer in ISPD2005 placement contest

<table>
<thead>
<tr>
<th></th>
<th>CPlace</th>
<th>FastPlace</th>
<th>DPlace</th>
</tr>
</thead>
<tbody>
<tr>
<td>adaptec1</td>
<td>808</td>
<td>1.24</td>
<td>1.79</td>
</tr>
<tr>
<td>adaptec2</td>
<td>1190</td>
<td>0.83</td>
<td>1.12</td>
</tr>
<tr>
<td>adaptec3</td>
<td>2803</td>
<td>0.76</td>
<td>0.92</td>
</tr>
<tr>
<td>adaptec4</td>
<td>2830</td>
<td>1.90</td>
<td>2.30</td>
</tr>
<tr>
<td>bigblue1</td>
<td>1132</td>
<td>1.52</td>
<td>2.02</td>
</tr>
<tr>
<td>bigblue2</td>
<td>6072</td>
<td>0.82</td>
<td>1.90</td>
</tr>
<tr>
<td>bigblue3</td>
<td>7861</td>
<td>1.75</td>
<td>2.46</td>
</tr>
<tr>
<td>bigblue4</td>
<td>14322</td>
<td>0.74</td>
<td>1.8</td>
</tr>
<tr>
<td>average</td>
<td>1</td>
<td>1.20</td>
<td>1.78</td>
</tr>
</tbody>
</table>

The run time is 1.78 times faster DPlace, 1.2 faster than FastPlace. From the results we can conclude the CSAGO have the obvious advantage of accelerating the speed of converging.

5. Conclusions

We have presented a novel cell migration technique, CSAGO can globally control the cell migration from the high density to the lower. The experimental result shows that average HPWL have reduced 3% and the total runtime is 1.78 times faster than DPlace [1].

References


