Sample Solutions to Homework #3

1. (10)

The estimation results are shown in Figure 1:

\[ m = (21 - 3) + (24 - 2) = 40, \]
\[ n = 45 \text{ (Figure 1(a))}, \]
\[ p = 40 \text{ (Figure 1(b))}, \]
\[ q = 2035, \]
\[ r \approx 67.22, \]
\[ s \approx 21.87. \]

![Figure 1: Wirelength estimation for different methods.](image)

2. (15)

(a) Let \((x, y)\) be the position of cell \(i\). The force-directed method can be computed by:

\[ 1 \times ((x, y) - (2, 21)) + 3 \times ((x, y) - (25, 1)) + 5 \times ((x, y) - (3, 18)) + 7 \times ((x, y) - (16, 8)) = 0 \]
\[ \Rightarrow \quad (x, y) = (12.75, 10.625) \approx (13, 11) \]

(b) We can search by window in nearby area to find a legal location.
3. (15)

(a) (5) The result is shown in Figure 2.

(b) (5) As the B*-tree packing, the x-coordinates of nodes in MP-tree can be directly derived; therefore we focus on the y-coordinates computing. To compute y-coordinates of macros in MP-tree, we keep two horizontal contours, the bottom contour and the top contour, which are initialized according to the bottom side and the top side of the given rectilinear region, respectively.

The bottom horizontal contour is for BL- and BR-packing to record the current maximum y-coordinates, and the top one is for TL- and TR-packing to record the minimum y-coordinates. The packing subtrees that use the same contour data structure always generate overlap-free placement results since the contour reserves the spaces of the traversed blocks. Figure 3 gives a packing example for an MP-tree with a BL-packing subtree and a BR-packing subtree. The packing order is \(b_1, b_2, b_3, b_4, b_{10}, \text{and} b_{11}\). Adding a new block \(b_{11}\) to the placement, we search the contour and update it with the top boundary of the new block. It is clear that no overlap will occur when we process BL- and BR-packing subtrees.

BL-/BR-packing subtrees, however, may overlap with TL-/TR-packing subtrees, and thus we should discard this kind of infeasible solutions by adding a *vertical overlap penalty cost* in the evaluation function of SA.

Similar to B*-tree, the computing for y-coordinate of one macro requires amortized \(O(1)\) time. Therefore, the time complexity is amortized \(O(n)\) time, where \(n\) denotes the number of macros.

(c) Strength: The white space in the center of placement region is more complete for standard cell placement. Weakness: The implementation are complicated, where two horizontal contours need to be handled.
4. (10) Let \( \hat{z}_1, \hat{z}_2, \ldots, \hat{z}_n \) denote the sequence \( z_1, z_2, \ldots, z_n \) sorted by non-decreasing order. Then

\[
\text{err}_\gamma(z_1, z_2, \ldots, z_n) = LSE_\gamma(\hat{z}_1, \hat{z}_2, \ldots, \hat{z}_n) - \hat{z}_n
\]

\[
= \gamma (\ln(e^{\hat{z}_n}/\gamma) + \sum_{i=1}^{n-1} (e^{\hat{z}_i}/\gamma)) - \hat{z}_n.
\]

Since \( \sum_{i=1}^{n-1} (e^{\hat{z}_i}/\gamma) \) is non-negative and approaches 0 when \( \hat{z}_i \to -\infty, \forall i \in [1, n-1] \), therefore the lower bound is derived as follows:

\[
\text{err}_\gamma(z_1, z_2, \ldots, z_n) \geq \gamma (\ln(e^{\hat{z}_n}/\gamma + 0)) - \hat{z}_n
\]

\[
= \hat{z}_n - \hat{z}_n = 0.
\]

Also, since \( \ln \) and \( \exp \) are non-decreasing functions and we have \( \hat{z}_i \leq \hat{z}_n, \forall i \in [1, n-1] \), therefore the upper bound is derived as follows:

\[
\text{err}_\gamma(z_1, z_2, \ldots, z_n) \leq \gamma (\ln(n \times e^{\hat{z}_n}/\gamma)) - \hat{z}_n
\]

\[
= \gamma \ln(n) + \hat{z}_n - \hat{z}_n = \gamma \ln(n).
\]

5. (10) The quadratic wirelength is minimized when \( Q(x_1, x_2, x_3)^T + d_x = 0 \), and \( Q(y_1, y_2, y_3)^T + d_y = 0 \).

From the problem, \( Q, d_x, \) and \( d_y \) are calculated as:

\[
Q = \begin{pmatrix} 3 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 2 \end{pmatrix} - \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 3 & -1 & -1 \\ -1 & 3 & 0 \\ -1 & 0 & 2 \end{pmatrix}
\]

\[
d_x = (-25, -25, -30)^T,
\]

\[
d_y = (-5, -45, -30)^T.
\]

Therefore, the wirelength is minimized when \( (x_1, x_2, x_3) \approx (22.31, 15.77, 26.15) \), and \( (y_1, y_2, y_3) \approx (16.15, 20.38, 23.08) \).

6. (10)

(a) We first use MP-tree to do macro placement in each domain. Then, analytical placement is performed in each domain to place standard cells. Finally, legalization and detail placement are performed in each domain.

(b) The analytical approach has good quality in both sparse and dense designs, and it is easy to perform standard cell legalization. However, it may encounter some difficulties while tackling legalization for mix-sized cells and macros.

7. (10)

(a) \( a = 20, b = 20, \) error rate = 0%.

(b) \( a = 20, b = 24, \) error rate = 16.67%.

(c) \( a = 20, b = 28, \) error rate = 28.57%

(d) As the degree of a net increases, the HPWL wirelength approximation would underestimate the wirelength. Scaled-HPWL can be used to approximate high-degree net; that is, we scale the HPWL by a factor \( c \):

\[
\text{scaled-HPWL} = c \times \text{HPWL},
\]

where \( c \) is proportional to the degree of a net.

8. (20)

We can utilize Abacus to solve the problem.

Algorithm for legalization with PlaceRow as Abacus algorithm:
To consider double-row-height cells and white-space issues, the quadratic cost function in PlaceRow should be modified. Note that the original problem formulation of Abacus is:

$$\min \sum_{i=1}^{N_r} e(i) \ (x(i) - x'(i))^2$$

such that 

$$x(i) - x(i - 1) \geq w(i - 1), \ 2 \leq i \leq N_r,$$

where $N_r$ is the number of cells in row $r$, $x(i)$ is the $x$ coordinate of cell $i$, $x'(i)$ is the original $x$ coordinate of cell $i$, $e(i)$ is the weight of cell $i$, and $w(i)$ is the width of cell $i$.

We modify the objective function of the above problem formulation to the following new objective function:

$$\min \sum_{i=1}^{N_r} e(i) \ (x(i) - x'(i))^2 + \alpha \text{ (white space area)}$$

, where $\alpha$ is the user-defined parameter, and white space area is to consider the white space induced by double-row-height cell legalization. Note that a double-row-height cell $j$ might be blocked by other cells in another row $r'$ such that cell $j$ can not be attached to the boundary of cell $j - 1$, which is the cell before cell $j$ in row $r$.

9. (50) DIY Problem