Sample Solutions to Homework #2

1. (10)

To apply simulated annealing, we must define 4 items: the solution space, the neighborhood structure, the cost function, and the annealing schedule.

   (1) Solution space: We divide all cells into two set $C_1$ and $C_2$, where $C_1 \cap C_2 = \emptyset$ and both meets balance criteria.

   (2) Neighborhood structure: We randomly select a cell, move it to the opposite side, and then get two new cell sets $C'_1$ and $C'_2$, which is the neighborhood of the original solution.

   (3) Cost function: We define $\phi$ representing the cutsize between $C_1$ and $C_2$ as our cost function. The objective of our simulated annealing is to minimize the cost function.

   (4) Annealing schedule: See the following procedure:

   1. Get a randomly initial partition $C_1, C_2$;
   2. Set the initial temperature $T > 0$;
   3. while not yet “frozen” do
      4. for $1 \leq i \leq P$ do
         5. Pick a random cell $c$ in $C_1 \cup C_2$;
         6. $\Delta \leftarrow \text{cost}(\phi') - \text{cost}(\phi)$;
         7. if $\Delta \leq 0$ then $C_1, C_2 \leftarrow C'_1, C'_2$;
         8. if $\Delta > 0$ then $C_1, C_2 \leftarrow C'_1, C'_2$ with probability $e^{-\Delta T}$;
      9. $T \leftarrow rT$;
   10. return $C_1, C_2$

2. (20)

   (a) (5) No, the balloting property is not satisfied because the number of operands is equal to the number of operators for subexpression $E_8 = 12V34HVH$.

   (b) (5) No, $E$ is not a normalized Polish expression because the balloting property is not satisfied.

      We apply the $M_3$ operation to make $E$ be a normalized Polish expression $E'$:

      $$ E = 12V34HVH5 \xrightarrow{M_3} E' = 12V34HV5H. $$

   (c) (5) Note that $E$ is not a Polish expression because the balloting property is not satisfied. We can only show the corresponding tree for $E'$. The corresponding tree is shown in Figure 1.

   (d) (5) The area of the smallest bound rectangle is 42. The steps are shown in Figure 2.
3. (10) Because the y coordinate of an element in list \( C \) equals \( \max\{q_i, y_j\} \), \( \forall 1 \leq i \leq m, 1 \leq j \leq n \), there are at most \( m + n - 1 \) different kinds of y coordinate in \( C \). (At least one element with the smallest y coordinate will be deleted.)

And since the element \((c_j, d_j)\) would be deleted if there exists another element \((c_i, d_i)\) in \( C \) with \( c_i \leq c_j \) and \( d_i \leq d_j \), for those elements with the same y coordinate in \( C \), only one element (which has the smallest x coordinate) can survive from deletion.

Therefore, the resulting list \( C \) has at most \( m + n - 1 \) elements.

4. (35)

(a) (5) See Figure 3.

\[ \Gamma_+ = eca db \]
\[ \Gamma_- = acbde \]
(b) (5) See Figure 4.

Figure 4: The horizontal and vertical constraint graphs $G_H$ and $G_V$ for $S$.

(c) (5) We can derive the length of the longest path in $G_H (s \rightarrow a \rightarrow d \rightarrow t)$ is 7, and in $G_V (s \rightarrow b \rightarrow d \rightarrow e \rightarrow t)$ is 8. Hence, the cost is $7 \times 8 = 56$.

(d) (5) To derive a B*-tree, we first compact the floorplan to left and down as shown in Figure 5(a). Then, we can get the B*-tree (Figure 5(b)).

(e) (5) Module $a$: $(0,0)$,
Module $b$: $(0 + 3, 0) = (3, 0)$,
Module $c$: $(0, 0 + 2) = (0, 2)$,
Module $d$: $(0 + 2, 0 + 3) = (2, 3)$,
Module $e$: $(0, 3 + 2) = (0, 5)$.
Area cost = $6 \times 8 = 48$.

(f) (5) See Figure 6.

(g) (5) $\Gamma_S = acbde$ ($C_H$ and $C_V$ are the same with Figure 6.)

5. (15)

(a) (5) We first decompose the L-shaped modules into rectangular ones (Figure 7(a)). Then, the B*-tree can be constructed (Figure 7(b)).

(b) (5) Let $W_i$ and $H_i$ be the width and height of module $i$, respectively.
$1a : (0,0)$
Figure 6: The TCG for Problem 4.

Figure 7: (a) The placement after the decomposition. (b) The corresponding B*-tree for the left placement.

2 : (0 + W_{1a}, 0) = (W_{1a}, 0)
3 : (W_{1a} + W_2, 0)
4a : (W_{1a} + W_2, H_3)
5 : (W_{1a} + W_2 + W_{4a}, H_3)
6 : (W_{1a} + W_2 + W_{4a} + W_7, H_3)
4b : (W_{1a} + W_2, H_3 + H_{4a})
9a : (W_{1a} + W_2, H_3 + H_{4a} + H_{4b})
10 : (W_{1a} + W_2 + W_9a, H_3 + H_{4a} + H_{4b})
9b : (W_{1a} + W_2, H_3 + H_{4a} + H_{4b} + H_{9a})
1b : (0, H_{1a})
7a : (0, H_{1a} + H_{1b})
7b : (0 + W_{7a}, H_{1a} + H_{1b}) = (W_{7a}, H_{1a} + H_{1b})
8 : (W_{7a}, H_{1a} + H_{1b} + H_{7b})

(c) (5) For the nodes corresponding to the modules along the bottom boundary of a floorplan, the nodes are on the left-most branch of B*-tree. For the nodes corresponding to the modules along the left boundary of a floorplan, the nodes are on the right-most branch of B*-tree.

6. (10) In general, the advantages of the Λ-shaped multilevel framework are the drawbacks of V-shaped multilevel framework and vice versa. The Λ-shaped framework has the view of local configuration at earlier stages and therefore good for local effects. However, such a framework lacks of global view. For the V-shaped framework, it has the view of the global configuration at earlier stages and good for global effects. Take a full-chip routing problem as an example. To achieve high routability, it is desirable to use a Λ-shaped routing framework since the routability
is a local effect. For a good circuit performance, using a V-shaped routing framework may be desirable.

7. (10) Let \((x_i, y_i)\) be the coordinate of the lower left corner of module \(m_i\), and \(y\) the top boundary of the floorplan. For the topological relationship of modules, we introduce the following binary variables:

\[ r_i : m_i \text{ is rotated.} \]

\[ p_{ij}, q_{ij} : \text{Specifier of nonoverlap constraints.} \]

The constraints of the mixed-ILP are described as follows:

\[
\begin{align*}
\text{minimize} & \quad y, \\
\text{subject to} & \quad x_i, y_i \geq 0, i = 1, 2, 3 \\
& \quad p_{ij}, q_{ij}, r_i \in \{0, 1\}, i = 1, 2, 3, j = \text{mod}(i, 3) + 1 \\
& \quad x_1 + 2(1 - r_1) + 3r_1 \leq x_2 + 10(p_{12} + q_{12}) \\
& \quad y_1 + 3(1 - r_1) + 2r_1 \leq y_2 + 10(1 + p_{12} - q_{12}) \\
& \quad x_2 + 5(1 - r_2) + 4r_2 \leq x_1 + 10(1 - p_{12} + q_{12}) \\
& \quad y_2 + 4(1 - r_2) + 5r_2 \leq y_1 + 10(2 - p_{12} - q_{12}) \\
& \quad x_2 + 5(1 - r_2) + 4r_2 \leq x_3 + 10(p_{23} + q_{23}) \\
& \quad y_2 + 4(1 - r_2) + 5r_2 \leq y_3 + 10(1 + p_{23} - q_{23}) \\
& \quad x_3 + 3(1 - r_3) + 6r_3 \leq x_2 + 10(1 - p_{23} + q_{23}) \\
& \quad y_3 + 6(1 - r_3) + 3r_3 \leq y_2 + 10(2 - p_{23} - q_{23}) \\
& \quad x_3 + 3(1 - r_3) + 6r_3 \leq x_1 + 10(p_{31} + q_{31}) \\
& \quad y_3 + 6(1 - r_3) + 3r_3 \leq y_1 + 10(1 + p_{31} - q_{31}) \\
& \quad x_1 + 2(1 - r_1) + 3r_1 \leq x_3 + 10(1 - p_{31} + q_{31}) \\
& \quad y_1 + 3(1 - r_1) + 2r_1 \leq y_3 + 10(2 - p_{31} - q_{31}) \\
& \quad x_1 + 2(1 - r_1) + 3r_1 \leq 10 \\
& \quad y_1 + 3(1 - r_1) + 2r_1 \leq y \\
& \quad x_2 + 5(1 - r_2) + 4r_2 \leq 10 \\
& \quad y_2 + 4(1 - r_2) + 5r_2 \leq y \\
& \quad x_3 + 3(1 - r_3) + 6r_3 \leq 10 \\
& \quad y_3 + 6(1 - r_3) + 3r_3 \leq y
\end{align*}
\]

\(y\) is the objective such that the floorplan area can be shrink. Equations 2 and 3 specify the types of variables and trivial constraints. Equations 4–7 are the nonoverlap constraints for \(m_1\) and \(m_2\). Equations 8–11 are the nonoverlap constraints for \(m_2\) and \(m_3\). Equations 12–15 are the nonoverlap constraints for \(m_3\) and \(m_1\). Equations 16–21 are the constraints to fit modules into the outline.

8. (15)

(a) (5) Adding a new operator \(Z\) representing Z-dimension slicing.

(b) (5) Similar to MP tree, we use several B*-tree to represent layers of modules. During packing, B*-trees in lower layers are packed before those in higher layers, maintaining Z-dimension contour.

(c) (5) We use two sequence pairs for 3D floorplanning one for X-Y plane loci, and one for X-Z (or Y-Z) plane loci. Packing with these sequence pairs can get the correct Z-dimension location for each module.

9. (25) DIY problem