10.
I would apply a 2-stage algorithm to address this problem. The first stage determines the rough location of each module, while the second stage determines the actual outline of the soft blocks.

The first stage is implemented by force driven analytical method referenced to [1].
We use a point representing each block. There are 2 types of force involved. The wire force and repel force.

$$min\ \widehat{W}(\boldsymbol{x},\boldsymbol{y}) + \hat{R}(\boldsymbol{x},\boldsymbol{y}) + \hat{B}(\boldsymbol{x},\boldsymbol{y})$$

Where $\widehat{W}(\boldsymbol{x},\boldsymbol{y})$ is the smoothed wirelength [1], and

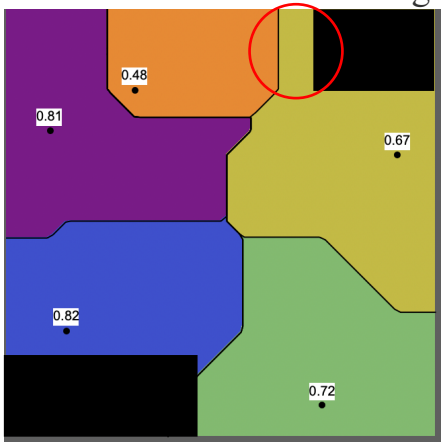
$$\hat{R}(\boldsymbol{x},\boldsymbol{y}) = \Sigma_{vi,vj}\ \frac{\alpha * area(vi) * area(vj)}{euclidean_{dist(vi,vj)}{}^{2}}$$

$$\hat{B}(\boldsymbol{x},\boldsymbol{y}) = \Sigma_{vi}\ (\frac{\beta * area(v_i)}{{x_i}^2} + \frac{\beta * area(v_i)}{(X-x_i)^2} + \frac{\beta * area(v_i)}{(Y-yi)^2} + \frac{\beta * area(v_i)}{{y_i}^2})$$

The minimization problem is solved by iteratively move the current solution towards the direction of gradient.
Note that the fixed block is included here to calculate the repel force, but the position is not variable so that the position is fixed.

In second stage, we first construct a weighted L1 Voronoi diagram of the soft block nodes with weight being the area of the block. Then place the fixed block.

Demo with normalized weight [2]:



After the initial floorplan is done, the following steps are to refine the borders.

1. Remove the spikes:

The spikes on the borders that has too large aspect ratio, as shown in the red circle, are trimmed, leave white spaces.

2. Straighten the border:
   For the z shaped border, calculate the proportion of two blocks and draw a straight border with the proportion retained.

(a) Before straighten (b) After straighten

3. Integer rounding
   The vertical and horizontal borders are rounded to the nearest integer lines and the 45 and 135 -degree border are converted to stair-shaped border and the turning points are integer valued.
4. Area calculation:
   Calculate the area of all blocks, if the block is smaller than it's given area, push it to a stack. The blocks with larger area are labeled.
   Then pop a block from the stack. If there's any whitespace or labeled block adjacent, push the whole border out without creating new turning points until the area requirement is satisfied, so that the removed spike won't grow back in this stage.
   If no labeled blocks or whitespace at adjacency, randomly select a border and push out until the requirement is satisfied, and push the affected blocks to the stack.
   Do this step iteratively until the stack is empty. Then, the floorplan is done!

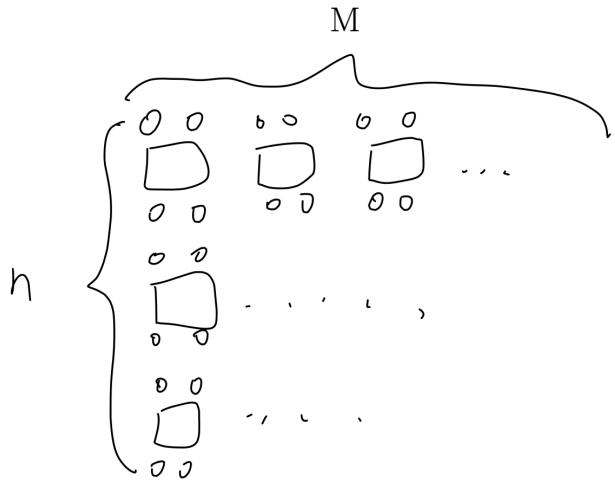
There are some obvious problems in the border refinement procedure that could be the direction of future work. For example, step 4 would probably result in other violations in other constraints, and may worsen the whole floorplan each iteration and the loop goes on forever.

[1] M. -K. Hsu and Y. -W. Chang, "Unified Analytical Global Placement for Large-Scale Mixed-Size Circuit Designs,"

[2]: https://demonstrations.wolfram.com/WeightedVoronoiDiagrams/

DIY

John is the manager of a small restaurant that have N row, M column of identical tables. Each table has 4 seats.



The restaurant accepts both booking online and walk-in costumer. Thus, a set S = {g1, g2, …, gn} representing the booked customer is given in the beginning. Also, for the quality of service, the turnover rate is 1, means one table would server at most once every night.

One important job of a manager is to arrange tables for the customers. From the over 10 years of career experience, he has come up with some rules for assigning tables.

Rule 1: Merging table makes customer feel uncomfortable, so no merging table. Tables take up by all groups would be $\sum\lceil gi.num/4\rceil$

Rule 2: For the customers with num>4 that separated to different tables, they want to stay as close as possible, but the distance means nothing when it goes too far. We can use a cost function to formulate the interest in distance between tables

$$satisfaction = \sum_{1\le k\le n} \sum_{ti,tj\ \epsilon gk,\ \ i<j} \frac{\alpha}{(ti.x - tj.x)^2 + (ti.y - tj.y)^2}$$

The value of x and y is indexed by row and column, (0, 0) at top left corner.

Rule 3: Sometimes little accident would turn into a huge conflict between two groups and ruin experiences of all costumers, and the reputation of your restaurant. The probability of such incident is higher between large groups. The formulation of the loss is as follow

$$loss = \sum_{1 \le i < j \le n} \beta * Overlap(\mathrm{i,j}) * (\mathrm{gi.num} * \mathrm{gj.num})$$

$$Overlap(i,j) = overlap\ area\ of\ bounding\ box\ of\ gi\ and\ g$$

To find the best assignment of tables, the objective functions is

$$\max (satisfaction - loss)$$

please help him to find a good algorithm to assign the tables.

Solution:
We can first calculate the table need for each group, and for each table, we use a block to represent it.
The compact form is not good for this problem due to the existence of the repelling force, so I just use a n*m array to keep the position information, by recording which group the occupying customer belongs to. If the table is empty, record 0. For each group, apart from the bounding box, the sum of satisfaction of all tables belongs to that group is stored.
Also, we keep a loss table: loss[(n-1+1)*(n-1)/2] recording the losses i, j instead of summing them up.
Then just apply the simulated annealing framework. The only neighborhood structure in this problem is swapping the location of two tables. Also, because only the involved tables' location information is modified in each iteration, we only need to re-calculate the satisfaction of the two involved group, and the losses that related to the any of them. Based on this method, the swapped table should belong to different groups, since all tables belongs to the same group has no difference when calculating the cost.

After the first round of simulated annealing, all booked table's positions are determined. For each group of walk-in customers, just do the simulated annealing again and those table that are already occupied will involve in the calculation of loss, but can't be selected as the swap target.