

CS6135 VLSI Physical Design Automation
Homework 3: Fixed-outline Slicing Floorplan Design
Student ID: 107065507 Name: Yun-Fan Lu

1. Goal

Implement the algorithm “A New Algorithm for Floorplan Design” by Wong and Liu, published in DAC, 1986.

2. Design Concept

實作 DAC-86 的 Simulated Annealing (SA) for floorplan 的演算法

Define Data Structure:

```
vector<string> NPE;
```

Normalized Polish Expression (NPE)為一個 `vector<string>` 結構，因為 STL 的 `vector` 操作很方便，速度也夠快。而 SA 演算法的所有操作只需要用 NPE 即可，不需要建 tree。

```
Algorithm: Simulated_Annealing_Floorplanning( $P, \epsilon, r, k$ )
1 begin
2  $E \leftarrow 12V3V4V \dots nV$ ; /* initial solution */
3  $Best \leftarrow E$ ;  $T_0 \leftarrow \frac{\Delta_{max}}{\ln(P)}$ ;  $M \leftarrow MT \leftarrow uphill \leftarrow 0$ ;  $N = kn$ ;
4 repeat
5    $MT \leftarrow uphill \leftarrow reject \leftarrow 0$ ;
6   repeat
7     SelectMove( $M$ );
8     Case  $M$  of
9        $M_1$ : Select two adjacent operands  $e_i$  and  $e_j$ ;  $NE \leftarrow Swap(E, e_i, e_j)$ ;
10       $M_2$ : Select a nonzero length chain  $C$ ;  $NE \leftarrow Complement(E, C)$ ;
11       $M_3$ :  $done \leftarrow FALSE$ ;
12      while not ( $done$ ) do
13        Select two adjacent operand  $e_i$  and operator  $e_{i+1}$ ;
14        if ( $e_{i-1} \neq e_{i+1}$ ) and ( $2N_{i+1} < i$ ) then  $done \leftarrow TRUE$ ;
15         $NE \leftarrow Swap(E, e_i, e_{i+1})$ ;
16         $MT \leftarrow MT + 1$ ;  $\Delta cost \leftarrow cost(NE) - cost(E)$ ;
17        if ( $\Delta cost \leq 0$ ) or ( $Random < e^{-\frac{\Delta cost}{T}}$ )
18          then
19            if ( $\Delta cost > 0$ ) then  $uphill \leftarrow uphill + 1$ ;
20             $E \leftarrow NE$ ;
21            if  $cost(E) < cost(best)$  then  $best \leftarrow E$ ;
22            else  $reject \leftarrow reject + 1$ ;
23        until ( $uphill > N$ ) or ( $MT > 2N$ );
24       $T = rT$ ; /* reduce temperature */
25    until ( $\frac{reject}{MT} > 0.95$ ) or ( $T < \epsilon$ ) or OutOfTime;
26 end
```

Unit 4

Perturbation:

Move 1: swap 2 adjacent operands.

Move 2: complement a chain.

Move 3: swap 2 adjacent operator and operand.

Move 4: rotate a block.

Cost function:

Cost = area + wirelength

3. Discussion

The details of your implementation. If there is anything different between your implementation and the algorithm in the DAC-86 paper, please reveal the difference(s) and explain the reasons.

本次作業是要用 DAC-86 的演算法去實作 fixed-outline slicing floorplan，但是還有要加上 rotate 問題。比較正確的做法是使用 Stockmeyer 演算法去 Optimal 決定旋轉方向，但是我的程式實作出來結果有誤，最後因為時間關係只好採用 random 旋轉的策略。即原來 M1, M2, M3 再加上一個 M4:

Rotate。

另外我的 SA 演算法會在是否接受 Best solution 時檢查是否符合 Fixed-outline 的規定，如果不符合就不接受，即使 cost 比原本還低，也不會接受這個答案。

另外我發現 M1 和 M4 比較有用，所以給 M1 和 M4 較大的機率。推測是因為 initial 排法就已經很 compact，太大的 Move 會讓 cost 降低的緣故。

What tricks did you do to speed up your program or to enhance your solution quality? Also plot the effects of those different settings like the ones shown below.

Simulated Annealing 是一種最佳化演算法，如果時間給夠長，應該會得到更好的結果。

一開始的 initial 方式是照著 Paper 上將 blocks 排成一排(12V3V...nV)，但這樣排其實 M3 會完全動不了，因為違反 Balloting 和 Skew 規則。之後又改成比較隨機排列的方式，但這樣可能過了 20 分鐘也還排不出一個符合 fixed-outline 的 solution。

所以我的策略是將 initial 改成 heuristic 的排法：我們的目標是將 Blocks 放在一個正方匡(outline)中，所以我會希望盡量將 width/height 相近的 blocks 排在一起以利減少浪費空間。我先將 blocks 的 height 排序，再排成一列，並且不超過 outline，一列排好後往上疊。這樣一開始就會是個比較好的 floorplan，然後再去做 SA 的最佳化。

另外也有像是用 unordered_map 來加速 map 結構的使用。

Please compare your results with the top 5 students' results last year for the

case where the dead space ratio is set to 0.15, and show your advantage either in runtime or in solution quality. Are your results better than theirs?

最後決定讓 SA 跑滿 20 分鐘，看看能優化的結果如何。

但是跟去年比的結果都不太好。

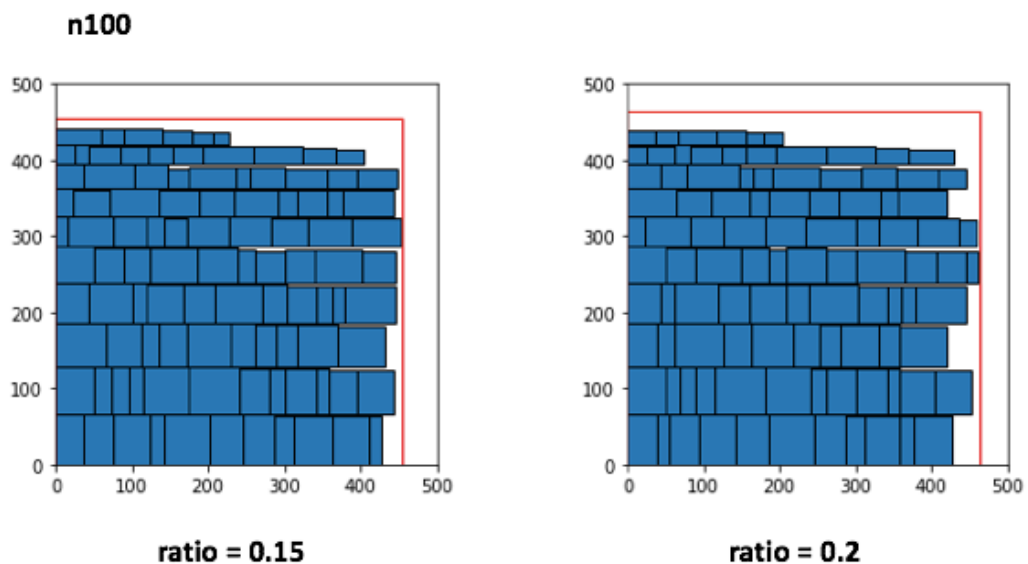
white space ratio = 0.15

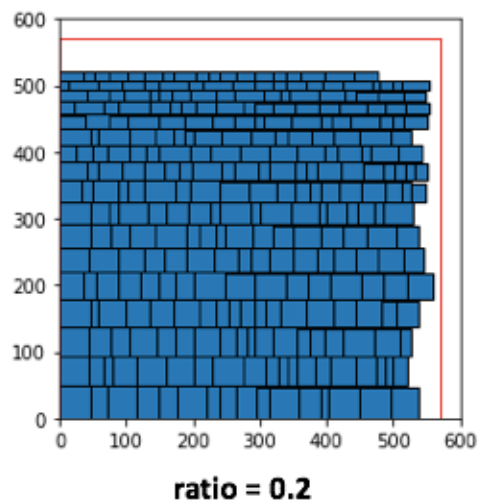
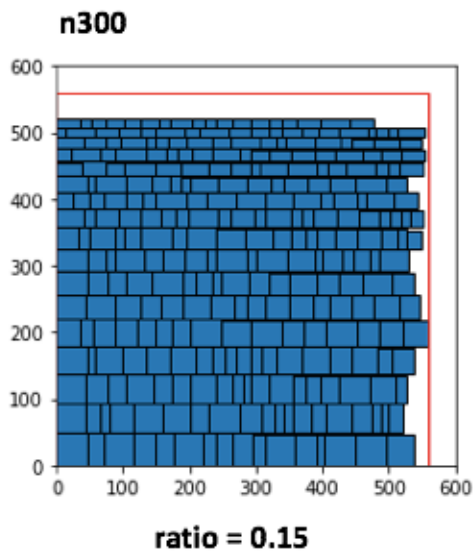
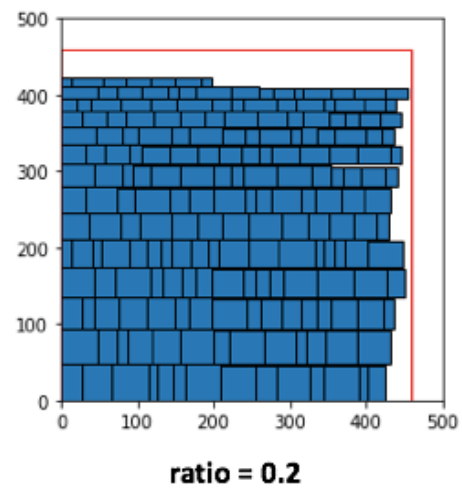
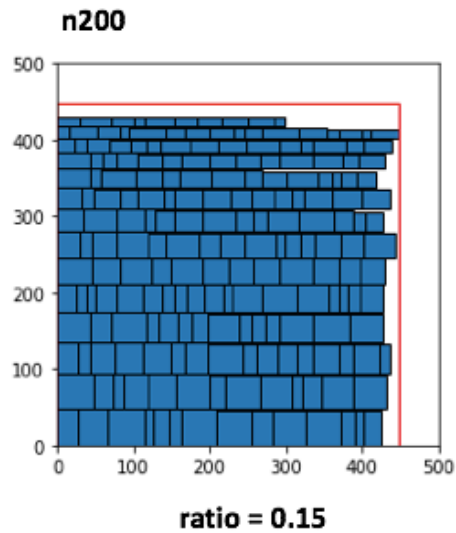
	n100	n200	n300
run time (s)	1200	1200	1200
wirelength	302068	5500357	845571

white space ratio = 0.2

	n100	n200	n300
run time (s)	1200	1200	1200
wirelength	297970	548740	834349

4. Results





5. Homework Review

原本應該要實作 Stockmeyer，無奈程式一直有 bug，最後只好退而求其次用 random 的方式 rotate。這次作業比較困難的應該是在計算 block 座標，花比較多時間。而 initial 的方式也是很重要。主要學到了 SA 的最佳化方法，資料結構的應用(stack, DFS)，以及要活用 call by reference 不要都放在 global。