Homework 3: Fixed-outline Floorplan Design

109062556 李濬安

- **How to compile and execute your program ?**

可以根據以下資料來去 compile & Run. (src/README)

src > $ README

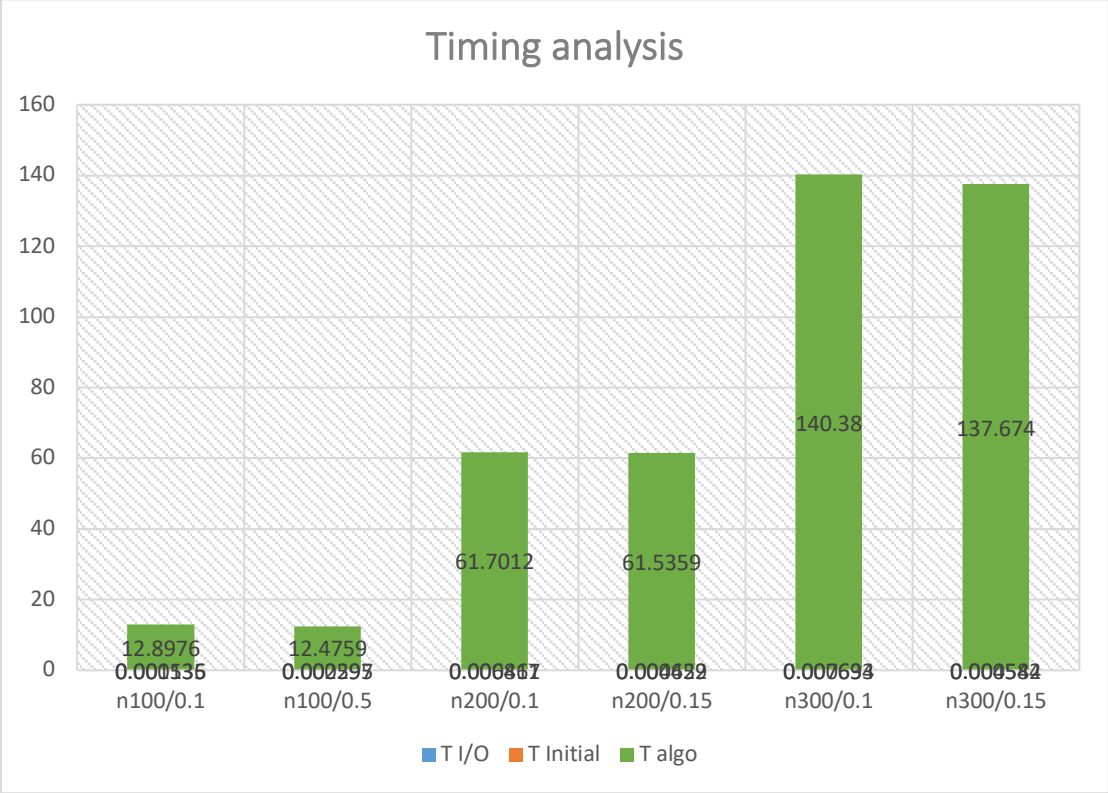
```
1   --  How to Compile the
2       In this directory, enter the following command:
3       $ make
4       It will generate the executable file "hw3" in "HW3/bin".
5
6       if you want to remove it, please enter the following command:
7       $ make clean
8
9   --  How to Run
10      In this directory, enter the following command:
11      Usage: ../bin/<exe> <hardblocks file> <nets file> <pl file> <outout file> <dead_space_ratio>
12      e.g.:
13      $ ../bin/hw3 ../testcases/n100.hardblocks ../testcases/n100.nets ../testcases/n100.pl ../output/n100.floorplan 0.1
14
15      In "HW3/bin/", enter the following command:
16      Usage: ./<exe> <hardblocks file> <nets file> <pl file> <outout file> <dead_space_ratio>
17      e.g.:
18      $ ./hw3 ../testcases/n100.hardblocks ../testcases/n100.nets ../testcases/n100.pl ../output/n100.floorplan 0.1
```

- **The wirelength and the runtime of each testcase.**

Wirelength			
ratio	n100	n200	n300
0.1	212975	379107	549241
0.15	198067	368495	522473

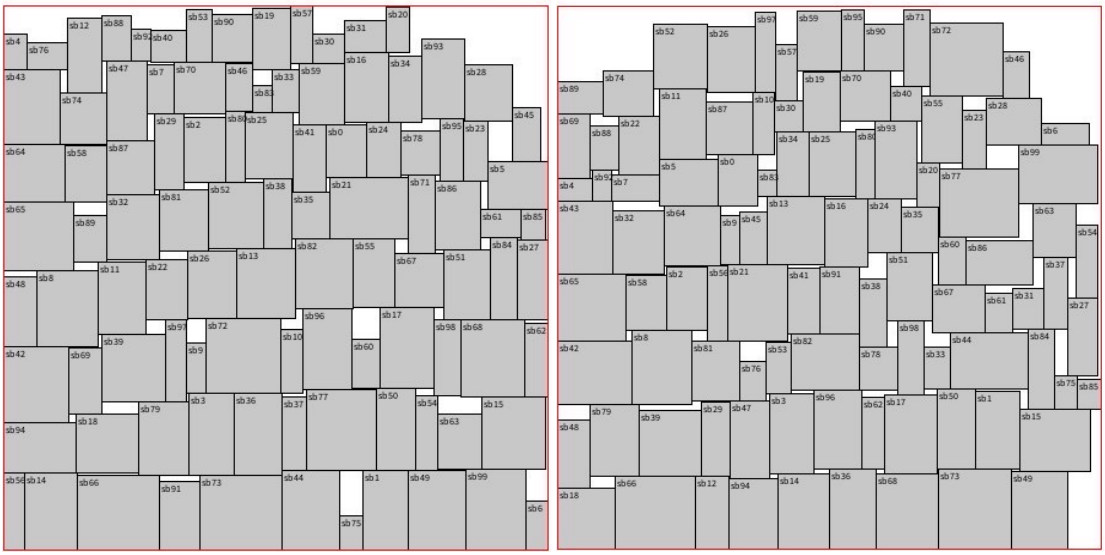
*T I/O: read input time, T Initial: initial floorplan time, T algo: algorithm time

Runtime(s)					
Testcase	ratio	T I/O	T Initial	T algo	Total
n100	0.1	0.001536	0.000135	12.8976	12.93
	0.15	0.002595	0.000297	12.4759	12.51
n200	0.1	0.006861	0.000417	61.7012	61.78
	0.15	0.004452	0.000629	61.5359	61.62
n300	0.1	0.007633	0.000694	140.38	140.53
	0.15	0.004544	0.000582	137.674	137.8

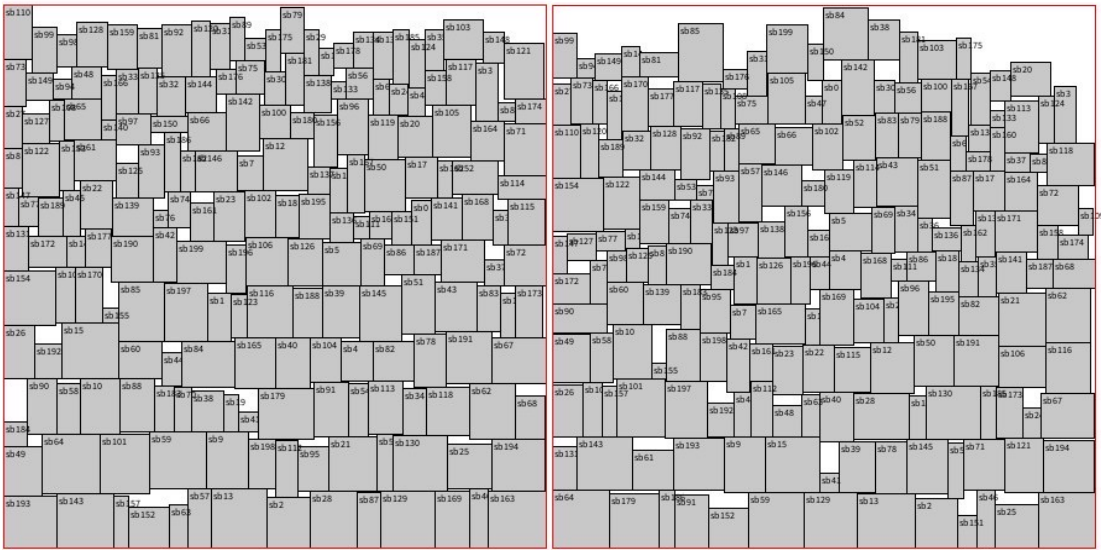


可以根據上圖發現我的 input time 跟 initial time 幾乎是看不到多，因此我主要花的時間都是在 SA 上面。

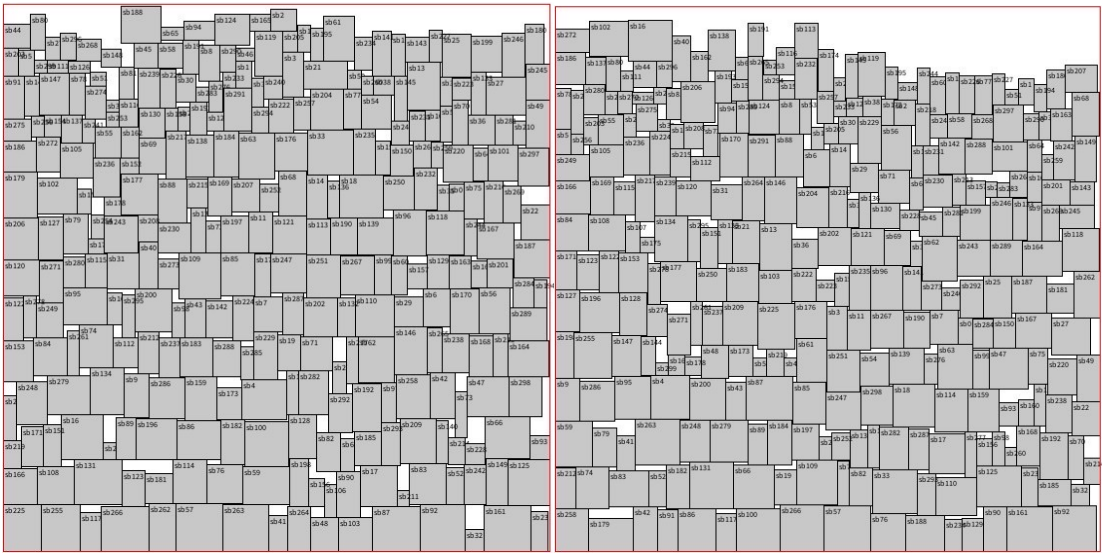
n100(0.1 / 0.15):



n200(0.1 / 0.15):



n300(0.1 / 0.15):



- **Please show that how small the dead space ratio could be.**

For testcase n100: 最小 dead space ratio 為 0.05，並 verify 成功

```
[g109062556@ic51 src]$ ../verifier/./verifier ../testcases/n100.hardblocks ../testcases/n100.nets ../testcases/n100.pl ../output/n100.floorplan 0.05
Total block area: 179501
Width/Height of the floorplan region: 434
Wirelength: 225593
Checking fixed-outline and non-overlapping constraints of the blocks locating ...
WL computed by verifier: 225593 <---> WL reported in .floorplan: 225593
OK!! Your output file satisfies our basic requirements.
```

For testcase n200: 最小 dead space ratio 為 0.06，並 verify 成功

```
[g10906255@ic51 src]$ ../verifier/./verifier ../testcases/n200.hardblocks ../testcases/n200.nets ../testcases/n200.pl ../output/n200.floorplan 0.06
Total block area: 175696
Width/Height of the floorplan region: 431
Wirelength: 509903
Checking fixed-outline and non-overlapping constraints of the blocks locating ...
WL computed by verifier: 509903 <---> WL reported in .floorplan: 509903
OK!! Your output file satisfies our basic requirements.
```

For testcase n300: 最小 dead space ratio 為 0.06，並 verify 成功

```
[g10906255@ic51 src]$ ../verifier/./verifier ../testcases/n300.hardblocks ../testcases/n300.nets ../testcases/n300.pl ../output/n300.floorplan 0.06
Total block area: 273170
Width/Height of the floorplan region: 538
Wirelength: 576470
Checking fixed-outline and non-overlapping constraints of the blocks locating ...
WL computed by verifier: 576470 <---> WL reported in .floorplan: 576470
OK!! Your output file satisfies our basic requirements.
```

- **The details of your algorithm.**

1. Initial partition:

   所使用的方法為 B star tree，那主要我會先將所有的 block 先去判斷它的高和寬，會將比較大的邊當作他的高，也就是是否要 rotate。

   再來會有一 sort_vector 將所有的 blocks 根據他的高做大小排序，也因此，較高的 block 會在 vector 的第一個。

   那根據 sort_vector 開始一個一個 pop 出來做 floorplan，先從最左下角開始往右擺，也就是開始建 left node，直到接近寬的 outline 或是剛好寬的 outline 時，再貼齊目前最左邊(最高的 block)往上擺，也就是建 right node，再繼續重複上述的往右擺，直到全部擺完為止。

   那這樣的 initial floorplan 的好處是，一開始較大的機率擺進 fixed outline。

2. Perturb:

   參考 paper 所使用的 perturb 有 rotate, move, swap。那 rotate 跟 swap 和原 paper 是一樣的方式，但 move，我有做一點修改。

   原 paper 的作法是將要被 move 的 node (from)先一直往他的 leaf 找，找到後與他交換，並最後接在目的地 node(to)的 child(隨機選擇左或右)，而原本被交換的 leaf node 則直接刪除。

   我的作法是在選擇接在 node to 的左或右中有一個判斷，是會看要被交換的 leaf node 是 parent 的左或右，如果是左的話，換到 node to 時會接在他的 left child。這樣想法是希望如果在最 left child node，可以一直在 left child，因為我是照著高度排序，也就是說被換過去如果是在 right child 可能往上疊容易超過 outline，也因此較難找到更好的 move。

3. Simulated annealing:

主要架構跟原 paper 類似，一樣先有起始溫度，接著一定比率的溫度下降，並且接受不好的 perturb 機率也隨著溫度下降而降低。

但 cost function 我所實作的方法較不一樣，原 paper 是將 HPWL 和 Area 當作考慮，而我的 cost function 是如下:

```
double COST(double area, double wirelength, pair<int, int> cur){
    double a = 1000, beta = 1;
    double widthCost = std::max((cur.first - outline) , 0);
    double heightCost = std::max((cur.second - outline) , 0);
    return a * ( widthCost + heightCost ) + beta * (wirelength);
}
```

我會根據這次 perturb 後的結果，找出他的最右邊界跟最上邊界，在各別和 outline 相減，如果在 outline 裡，會將參數為 0，接著會將相差值乘上一很大的 parameter = 1000，並加上 1 * HPWL()。

會這樣設定是因為在實驗中我發現我所做出來的 floorplan 很容易超出 outline，因次經過多次微調，發現這樣的比率實驗出來是較好的。

- **What tricks did you do to speed up your program or to enhance your solution quality?**

1. Speed

在執行速度上，我提升速度方法是會將所有的 tree node 都有 block 參數，好處是要取用 block 或對 block 做 perturb 時，不用再去從 block 容器裡搜尋並更新，這大幅提升了每次 perturb 的速度。

2. Quality

我試過多種的 initial floorplan 及 cost function，那覺得這兩個都相當重要，initial floorplan 是用上述的方法來完成。結果比用隨機分配的來得好。

那 cost function 本來是 alpha*HPWL() + (1-alpha)*Area()，發現常常會超過 fixed outline，抑或是沒辦法找到更好的 cost 的 perturb，因此經過多次調整才有了上述新的方式，著重在有沒有超過 fixed outline。

- **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**

Wirelength				Runtime(s)		
Ranks	n100	n200	n300	n100	n200	n300
1	200956	372143	516906	24.63	47.29	65.81
2	198593	368731	535257	200.25	308.06	226.42
3	194369	354107	491069	385.75	709.61	926.55
4	204001	367298	499733	330.42	576.15	793.26
5	208575	378187	567794	26.72	120.73	247.22
My	198067	368495	522473	12.51	61.62	137.8

Wirelength:

n100: 我的只比 rank 3 大一些，但都比其他的短

n200: 我比大部分人都短，比 rank3, rank4 大一些

n300: 比一半的人短，比 rank3 rank4 來得大

Runtime:

n100: 比所有人都快

n200: 只比 rank1 慢，比其他人都快

n300: 只比 rank1 慢，比其他人都快

總結來說，我認為的我實驗結果算蠻好的，比大多數的短並且也都更快。

- **What have you learned from this homework? What problem(s) have you encountered in this homework?**

在這次作業，我學到如何做出 floorplan 的所有流程，並且也實驗了用 fast SA 和 SA 的比較，而最後我選擇用 SA，因為實驗結果較好。那也學到很多演算法，並且實際 implement，所以蠻有成就感的！那遇到最大的問題是一開始很難下手，因為不到要怎樣的 initial floorplan 及怎樣的 SA 參數設定會比較好，並且也不知道 Cost function 如何設定比率會比較好，因此由於種種的變數設定，讓我這次作業很難下筆。最後，也花了不少時間在調整參數上。總體來說，這次作業很有成就感，實際 implement 一篇 citation 超多的 paper，並且結果也蠻好的。

```
grading on 109062556:
  testcase |       ratio | wirelength |     runtime | status
      n100 |        0.15 |     198067 |       12.61 | success
      n200 |        0.15 |     368495 |       61.36 | success
      n300 |        0.15 |     522473 |      141.44 | success
      n100 |         0.1 |     212975 |       12.75 | success
      n200 |         0.1 |     379107 |       62.83 | success
      n300 |         0.1 |     549241 |      143.45 | success
|-------------------------------------------------------|
|                                                       |
|    Successfully generate grades to HW3_grade.csv      |
|                                                       |
```