Homework 3: Fixed-outline Floorplan Design

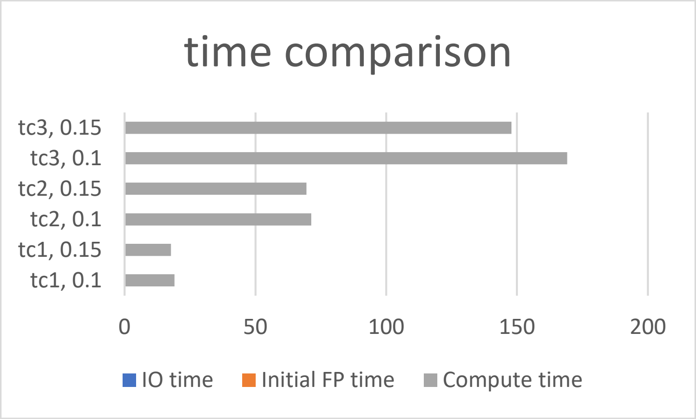
(1) 洪偉勛 109065527

(2) How to compile and execute your program and give an execution example

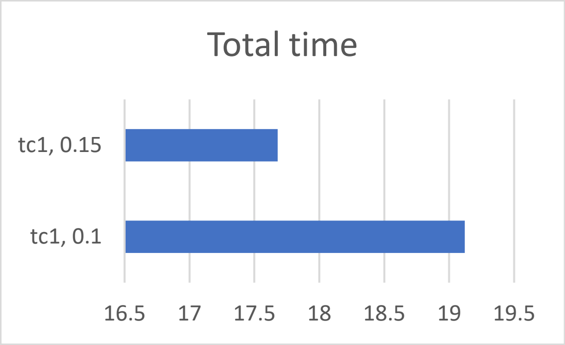
```
$ make
$ ../bin/hw3 ../testcase/n100.hardblocks ../testcase/n100.nets ../testcase/n100.pl ../output/n100.floorplan 0.1
```

(3) The wirelength and the runtime of each testcase with the dead space ratios 0.1 and 0.15, respectively

TESTCASE, DEAD RATIO	WIRELENGTH	FIXED OUTLINE	IO TIME	INITIAL FP TIME	COMPUTE TIME	TOTAL TIME
TC1, 0.1	221329	444	0.0047	9.20E-05	19.115208	19.12
TC1, 0.15	215084	454	0.004414	0.000103	17.675483	17.68
TC2, 0.1	410526	439	0.005	1.60E-04	71.40484	71.41
TC2, 0.15	396880	449	0.004	9.82E-05	69.5459018	69.55
TC3, 0.1	575355	548	0.008113	2.30E-04	169.151657	169.16
TC3, 0.15	559255	560	0.008879	0.000231	147.88089	147.89



It is clear that IO time and others count nothing in comparison to Compute time. Time between different dead space ratio doesn’ t differ much since the max time is limited by the dead space ratio.



(4) Please show that how small the dead space ratio could be for your program to produce a legal result in 20 minutes.

CASE, DEADSPACE RATIO	MY DEADSPACE RATIO
TC1, 0.1	0.0785
TC1, 0.15	0.0729
TC2, 0.1	0.08834
TC2, 0.15	0.1285
TC3, 0.1	0.0813
TC3, 0.15	0.089

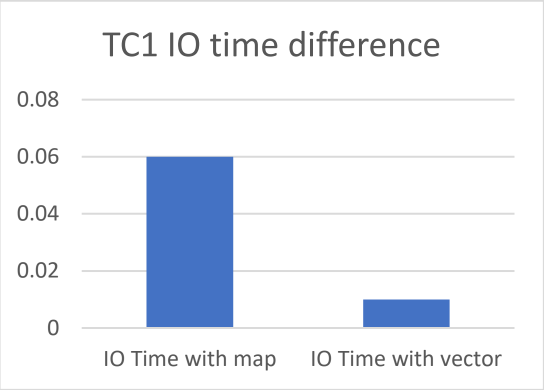
(5) The details of your algorithm.You could use flow chart(s) and/or pseudo code to help elaborate your algorithm. If your method is similar to some previous work/papers, please cite the papers and reveal your difference(s).

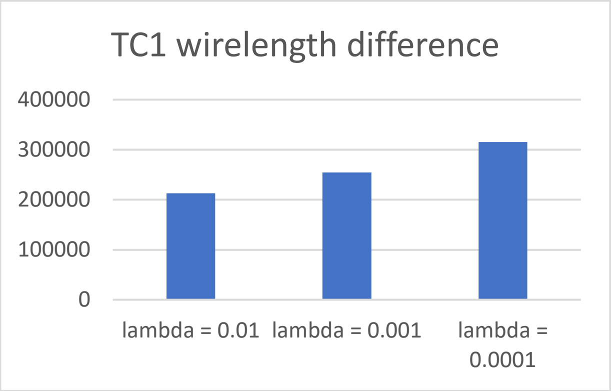
I used the b star tree algorithm with simulated annealing talked in the lecture. A minor difference is that I modified the cost function in order to implement the fix-outline constraint. My cost function includes the W and H penalty when violating FO. And also the aspect ratio of the floorplan is a cost.

(6) 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.

I had been using <map> to implement the list of Block/Terminal/Net. However this data structure is much slower than vector, which can be random accessed. So I decided to use vector finally. The IO runtime dropped from 0.06 to 0.01, so does the compute time.

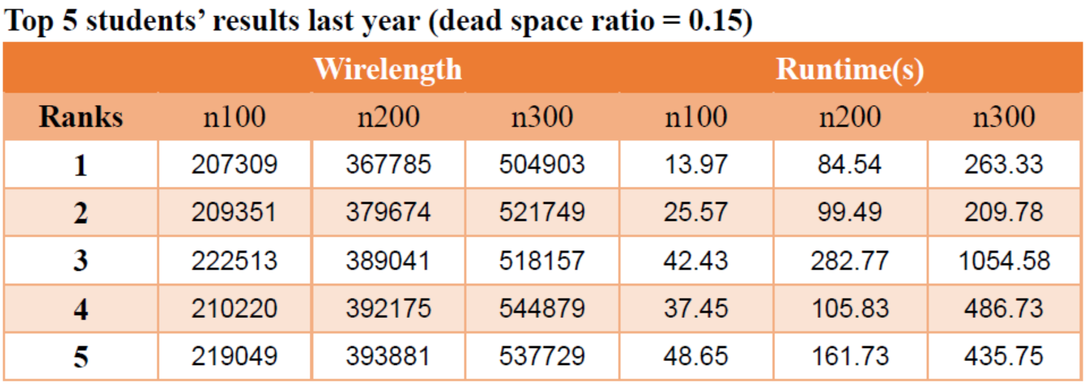
On the other hand, tuning some parameters can also enhance the solution quality such as lambda which controls the weight in cost function. WL reduces significantly when lambda increases. However the fixed outline constraint may be violated sometimes.





(7) Please compare your results with the top 5 students' results

Some results are better than 1 or 2 of them in n100. I still have some room for improvement. Such as making rotate moves only when temperature is lower than some specific temperature. This may help to further minimize the size and HPWL.



Top 5 students' results last year (dead space ratio = 0.15)

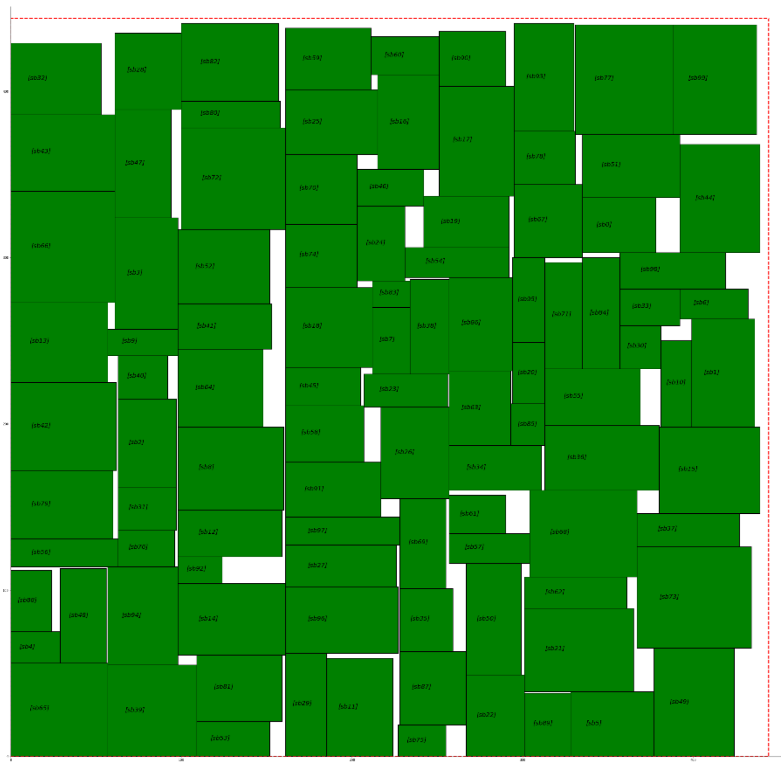
	Wirelength			Runtime(s)		
Ranks	n100	n200	n300	n100	n200	n300
1	207309	367785	504903	13.97	84.54	263.33
2	209351	379674	521749	25.57	99.49	209.78
3	222513	389041	518157	42.43	282.77	1054.58
4	210220	392175	544879	37.45	105.83	486.73
5	219049	393881	537729	48.65	161.73	435.75

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

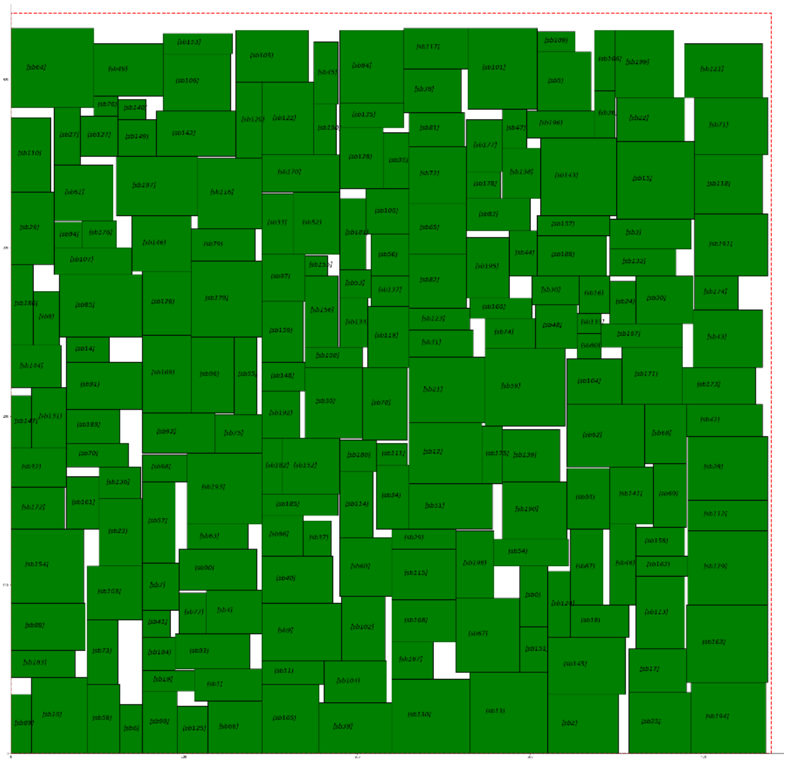
B star tree is way easier to implement than slicing tree and floorplan is also smarter. I did slicing tree in my first handout of HW3 and the performance was really bad and I had no time to write another one. I also had some difficulties when writing the perturb moves. Sometimes the parent/child relation doesn' t match. I spent lots of time to deal with move and swap. I would try to use fast SA next time.

Bonus graph(6 graphs in total)…next page

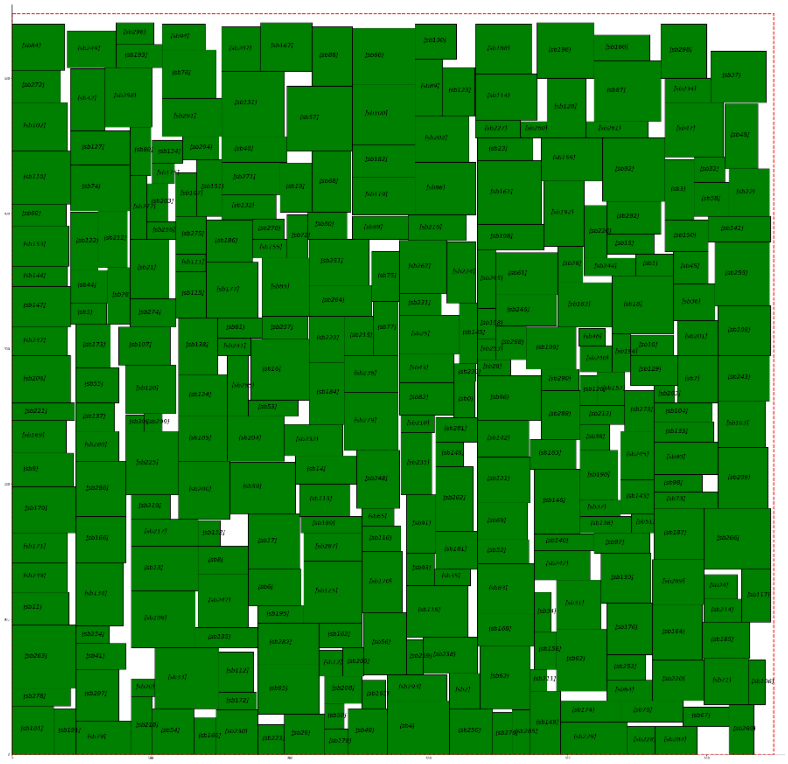
TC1, 0.1



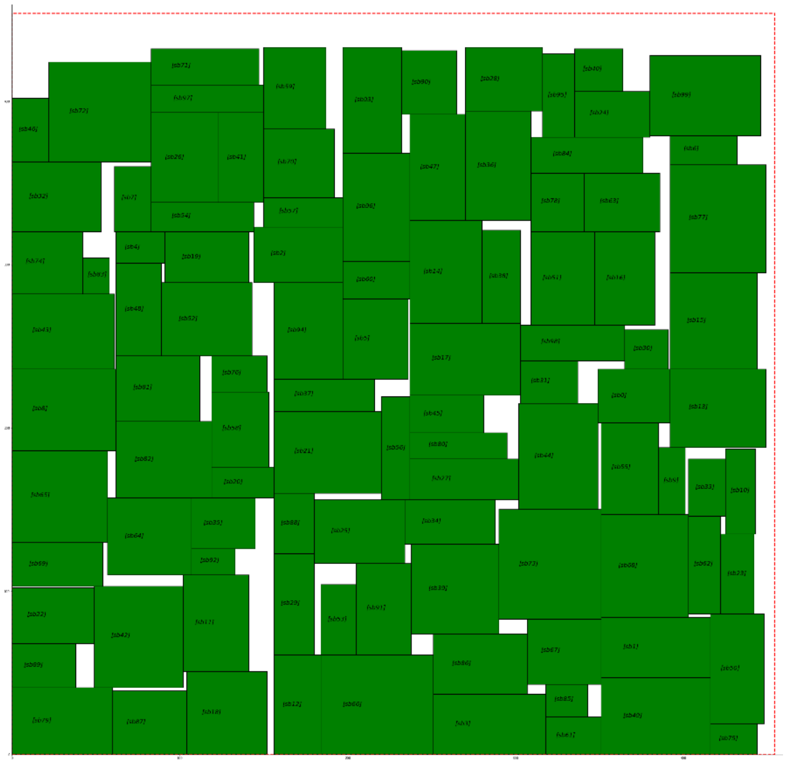
TC2, 0.1



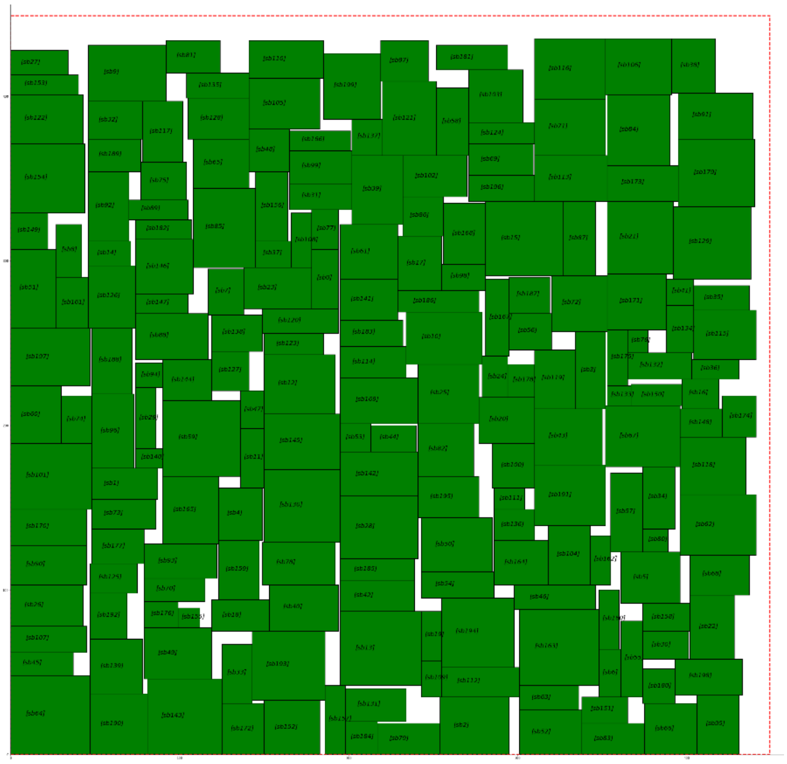
TC3, 0.1



TC1, 0.15



TC2, 0.15



TC3, 0.15