CS6135 VLSI Physical Design Automation
Homework 3: Fixed-outline Slicing Floorplan Design

111062625 蔡哲平

1. How to compile and execute my program.

- Compile: Enter ***src/*** and make, it'll generate the executable file to ***bin/***.

  ```
  $ cd src
  $ make
  ```

- Execute

  ```
  $ ./hw3 [*.hardblocks] [*.nets] [*.pl] [ *.floorplan] [dead_space_ratio]
  ```

  e.g.

  ```
  $ ./hw3 ../testcases/n100.hardblocks ../testcases/n100.nets ../testcases/n100.pl
    ../output/n100.floorplan 0.15
  ```

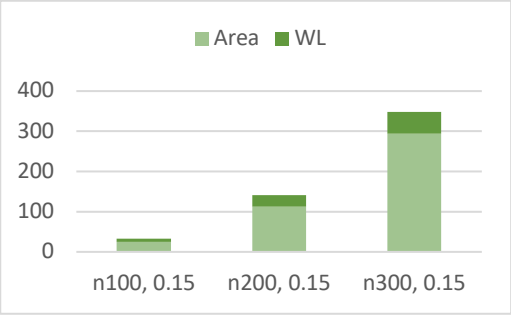
2. The wire length and the runtime of each testcase.

(Dead space ratio = 0.15)

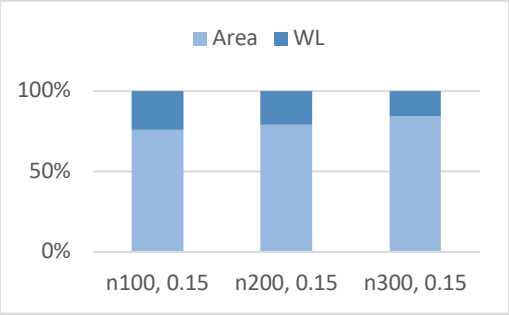
	n100	n200	n300
Wire length	204057	372608	502408
Runtime (s)	31.72	147.96	330.97

(Dead space ratio = 0.1)

	n100	n200	n300
Wire length	209466	384103	532129
Runtime (s)	30.07	140.75	330.95



Time Profile
(X: test case, Y: runtime(s))



Time Proportion
(X: test case, Y: proportion)

You can see that my program spends about 75% of its runtime calculating the area since it will also determine the hardblocks' coordinates at the same time. After each movement, it will re-compute all hardblocks' coordinates, making it very inefficient and becoming the bottleneck. Unfortunately, I didn't have enough time to optimize this part. (Note: IO time is extremely short compared to others.)

3. **The smallest dead space ratio my program can find in 10 minutes.**

	n100	n200	n300
Ratio	0.07	0.08	0.08
Wire length	224864	394374	547642
Runtime (s)	23.7476	143.422	341.493

4. **The details of my implementation:**

My program is very similar to the DAC-86 paper but with the below differences.

I. Initial solution

Instead of using *12V3V4V…* as my initial solution, which is used in the paper, I follow the below steps to generate it.

- Sort the hardblocks by their height.
- Put hardblocks from left to right in sorted order.
- If placing the current hardblock will make the floorplan out of the horizontal bound, then place it into the above new row.

Following the initial solution in the paper, it's nearly impossible to fit the floorplan into the outline even when the SA is terminated. However, following my approach, the initial floorplan can easily fit into the outline in most cases. It's because it will put the hardblocks with similar heights into the same row, making the floorplan very compact.

II. Cost function

My cost function is different from the one in the paper. The goal of the paper is to minimize both the area and the wire length. However, in this assignment, the goal is to fit the floorplan into the fixed outline and minimize the wire length. Minimizing the area and don't care about the wire length doesn't guarantee it can fit into the outline. Hence, the cost function should be adjusted to consider the outline too. Below is my cost function.

- $\boldsymbol{If}(width < outline\_width)\ \boldsymbol{then}\ width = outline\_width$ $\boldsymbol{else}\ penalty \mathrel{+}= (width - outline\_width)$
- $\boldsymbol{If}(height < outline\_height)\ \boldsymbol{then}\ height = outline\_height$ $\boldsymbol{else}\ penalty \mathrel{+}= (height - outline\_height)$
- $cost = 128 * (width * height + penalty) + wire\_length$

III. Movements

There are three types of movement to generate a neighbor in the paper, denoted as M1, M2, and M3. Although it's proved that each normalized polish expression can be obtained from any other one through a finite set of moves of

these three types, only M1 is useful in this assignment since both M2 and M3 will easily make the floorplan out of bounds, making it an illegal movement. Besides, I also apply an additional movement called *swap_random_operands*, denoted as M4. Different from M1 only swapping adjacent operands, M4 will randomly pick 2 operands and swap them. I found that applying both M1 and M4 can significantly shorten the wire length compared with applying M1, M2, and M3.

5. What tricks did you do to speed up your program or enhance your solution quality?

As mentioned above, the trick I used to improve the wire length is to apply both M1 and M4 with equal probability and abandon M2 and M3. This way, the wire length will be shorter. Below is a comparison table between them. You can see that although applying M1 and M4 has a little longer runtime, the wire length it produces is much shorter than applying M1, M2, and M3.

(Test case: n100, 0.15)

	M1, M4	M1, M2, M3
Wire length	**204057**	248459
Runtime (s)	30.72	**28.38**

6. Please compare your results with the previous top 3 students' results for the case where the dead space ratio is set to 0.15 and show your advantage either in runtime or in solution quality.

Wire length			
Ranks	n100	n200	n300
1	207309	**367785**	504903
2	209351	379674	521749
3	210220	392175	544879
Mine	**204057**	372608	**502408**

Considering only the wire length, you can see that my wire length is ranked first in n100 and n300 and ranked second in n200.

Runtime (s)			
Ranks	n100	n200	n300
1	**13.97**	**84.54**	263.33
2	25.57	99.49	**209.78**
3	37.45	105.83	486.73
Mine	31.72	147.96	330.97

Considering only the runtime, you can see that my runtime is ranked third in n100 and n200 and ranked fourth in n300.

Overall, I think my result may be ranked second compared with the previous top 3 students' results since my wire length is shorter than the first student's in two test cases, and the runtime is a little longer than the second student's.

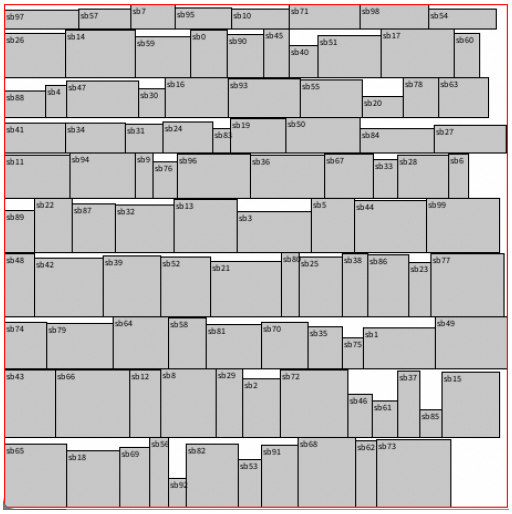
7. If you implement parallelization (for the algorithm itself), please describe the implementation details, and provide some experimental results.

Sorry, I didn't implement parallelization.

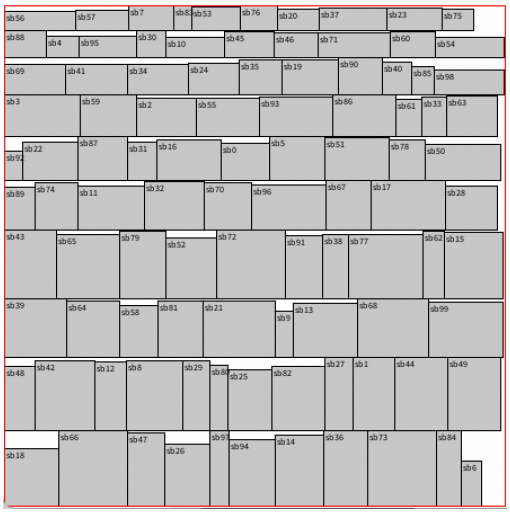
8. What have you learned from this homework? What problem(s) have you encountered in this homework?

The most difficult algorithm to implement in this assignment is Stockmeyer's algorithm. I spent lots of time thinking about how to calculate the correct coordinate of each hardblock. The other annoying problem is that the paper doesn't aim to fit the floorplan into the outline but to minimize the total area. Hence, you need to adjust the initial solution and the cost function to meet the outline constraint. Furthermore, there are too many parameters to be tuned in this assignment such as the initial temperature, the temperature decaying ratio, the probability of each movement, etc., even the random seed has a lot to do with your result, therefore making it difficult to come up with a feasible solution. All you can do is keep tuning your parameters, making this assignment isn't as interesting as the previous one.

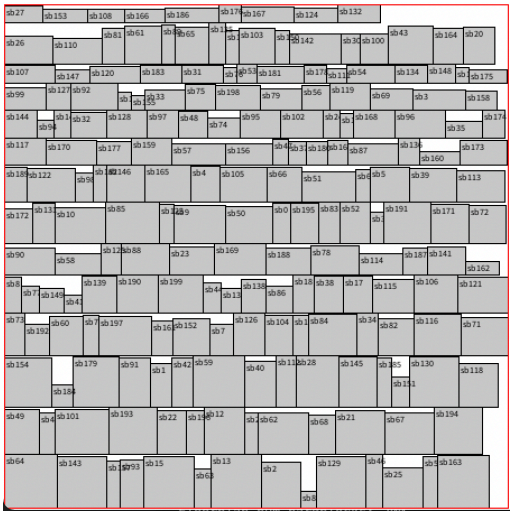
9. Floorplan figures



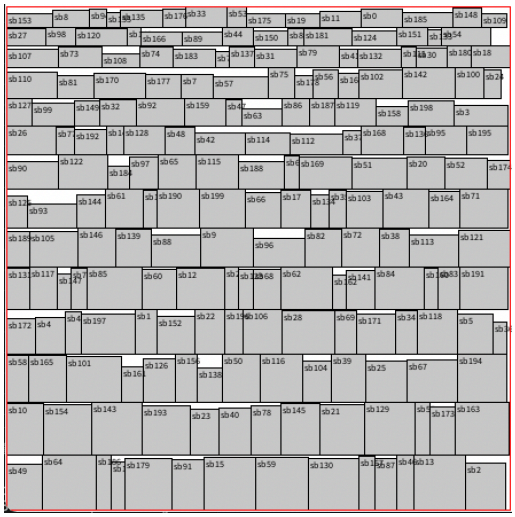
n100, ratio = 0.15



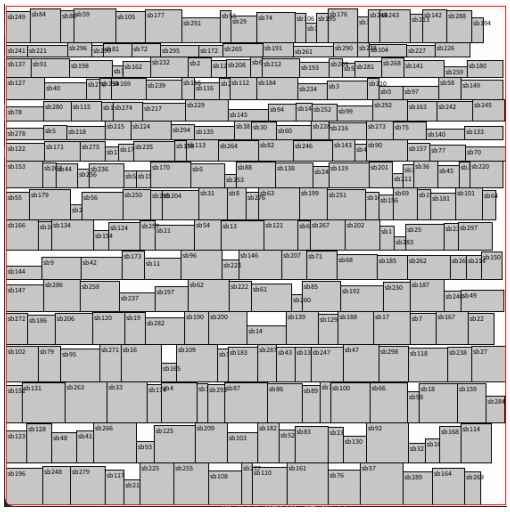
n100, ratio = 0.1



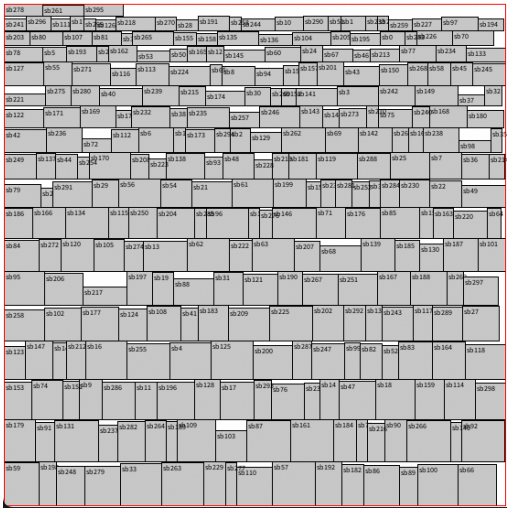
n200, ratio = 0.15



n200, ratio = 0.1



n300, ratio = 0.15



n300, ratio = 0.1