COL215 Software Assignment 2: Wire Aware Gate Positioning

Vanshika (2023CS10746)
Laksh Goel (2023CS10848)

October 5, 2024

1 Task

We are given gates, which contain pins. Gates were assumed to be rectangular and could not be re-oriented. The objective is to minimize the total wire length of the circuit.

2 Design Decisions

We begin by implementing basic algorithms.

1. We began by implementing this task using the software 1 assignment with the difference that we placed the blocks in order of their connectivity instead of maxside.
2. But this does not consider the coordinates of ports in it's implementation it used just the information about which gates are connected and thus was not much optimised.
3. Secondly, we implemented simulated annealing to find the most optimised placement.
4. Simulated annealing alone was not enough. So, in the base case of annealing we placed the blocks so that they occupy minimum bounding box.
5. The issue with the above implementation is that the blocks are placed too close to undergo any perturbation while doing simulations.
6. So, we need to think of a way such that the gates which are highly connected are close to each other but not much close.

2.1 Final Implementation

1. We, made a connectivity list that is the list which has connected gates adjacent to each other.
2. Next, we splitted this list into smaller clusters of size 10.

3. The gates in each cluster are arranged tightly using minimum bounding box concept from previous assignment.
4. The clusters are also then arranged using minimum bounding box concept with respect to each other.
5. Then we apply simulated annealing on clusters.
6. Further we apply simulated annealing on gates as a whole without considering any clusters.

Note: The main reason for applying simulated annealing on clusters is because when we were not doing so, while applying simulated annealing to individual gates the gates were automatically forming clusters in 2-D plane in the initial simulations and once, they form such clusters the gates won't move from those positions, since the wire length would increase if I move one gate. I need to move the whole cluster including that gate for optimisation.

2.1.1 Time complexity analysis

For every iteration, we choose random gate which takes logn amount of time, and we check the overlap for the perturbation and do other updations in coordinates of gates etc. which takes O(n) amount of time;where n is the number of gates. Calculating wire length in each iteration takes O(m) amount of time , where m is the number of pin_groups, but maximum value m takes is the number of wires. So, finally,
time complexity = O(number_of_iterations * (n + logn + m))

3 Test Cases

3.1 Some random Test Cases

Note: These are also submitted as custom_tc.txt and custom_tc_output.txt in gradescope submission.

1. Test Case 1
   wire_length 42

Example Test Case:

```
g1 20 10
pins g1 20 5
g2 20 30
pins g2 0 0
g3 70 10
pins g3 0 0
g4 20 40
pins g4 20 30 0 0 0 12
g5 30 0
pins g5 0 0 30 0
g6 30 20
pins g6 30 0
g7 10 50
pins g7 10 50
wire g3.p1 g2.p1
wire g2.p1 g5.p1
wire g4.p1 g6.p1
wire g4.p2 g1.p1
wire g4.p2 g4.p3
wire g7.p1 g5.p2
```

Output:

```
bounding_box 99 102
g4 79 42
g6 69 82
g1 59 37
g3 0 50
g7 20 0
g2 0 60
g5 0 50
```

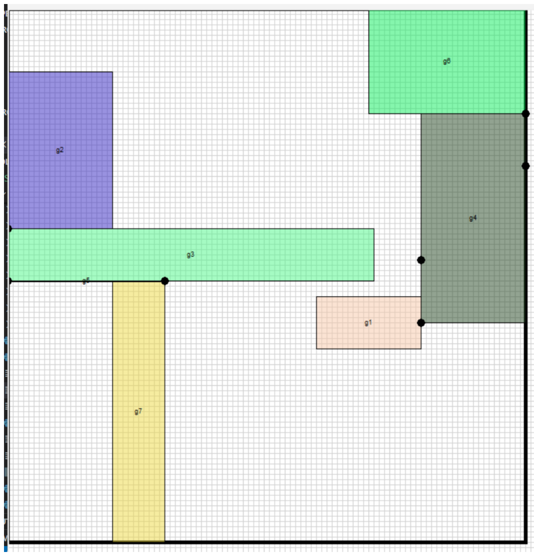

Figure 1: Our code output for above test case

2. Test Case 2
   wire_length 10

Example Test Case:

```
g1 30 2
pins g1 30 2
g2 40 30
pins g2 0 0 40 30
g3 20 2
pins g3 0 2
g4 10 42
pins g4 0 10 0 20
wire g1.p1 g2.p1
wire g2.p2 g4.p2
wire g4.p1 g3.p1
```

Output:

```
bounding_box 100 54
g4 70 12
g2 30 2
g1 0 0
g3 80 20
```

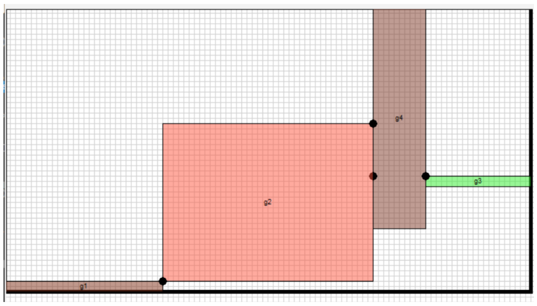

Figure 2: Our code output for above test case

3. Test Case 3

   wire_length 26

   Example Test Case:

```
g1 10 30
pins g1 10 25 10 8
g2 30 20
pins g2 30 10 30 15
g3 20 20
pins g3 0 15
g4 10 5
pins g4 0 2 10 4
g5 20 10
pins g5 20 5
wire g1.p1 g4.p1
wire g4.p2 g5.p1
wire g1.p2 g2.p1
wire g3.p1 g2.p2
```

   Output:

```
bounding_box 50 60
g1 20 20
g2 0 0
g3 30 0
g5 20 50
g4 30 43
```

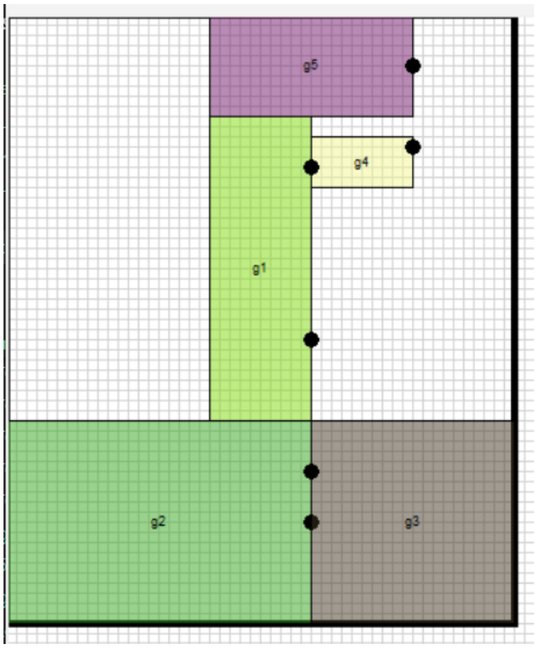

Figure 3: Our code output for above test case

3.2 Given Test Cases on moodle

1. Test Case 1
   Wire length: 26

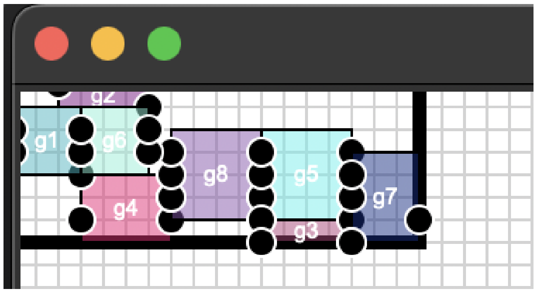

Figure 4: Our code output for above test case

2. Test Case 2
   Wire length: 26

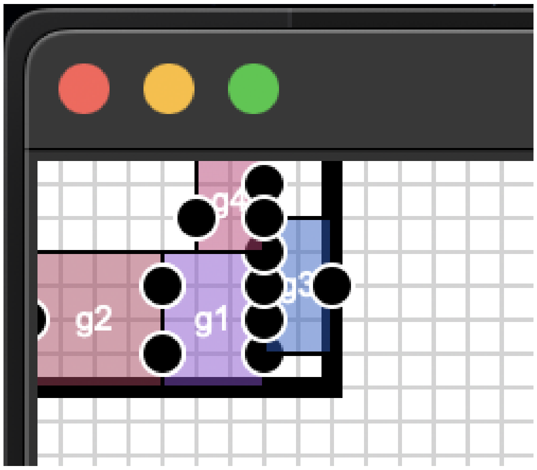

Figure 5: Our code output for above test case

3. Test Case 3
   Wire length: 51

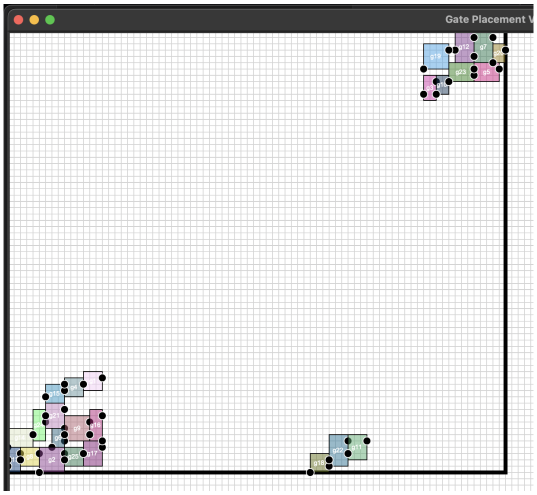

Figure 6: Our code output for above test case

4. Test Case 4
   Wire length: 29

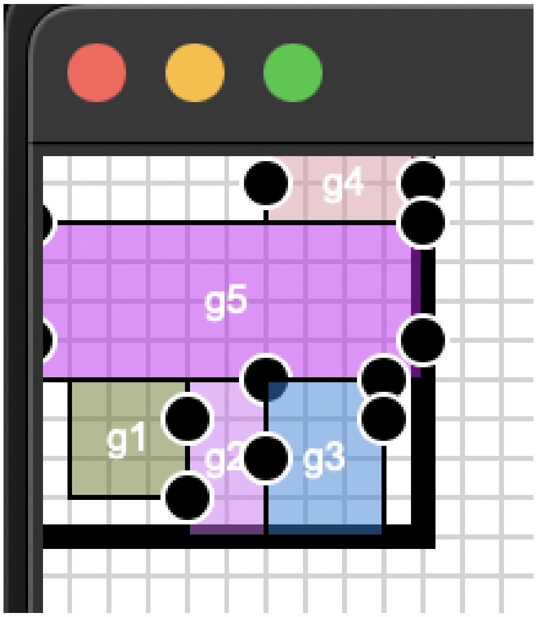

Figure 7: Our code output for above test case

4 Conclusion

Our code is highly optimised and takes time 1-2 mins. for 1000 gates scenario and within few seconds for less number of gates.