CS6135 VLSI Physical Design Automation

Homework 3: Fixed-outline Floorplanning

Due: 23:59, November 20, 2018

1. Introduction

In this homework, you are asked to implement an existing algorithm or develop your own algorithm to solve the fixed-outline floorplanning problem with a set of hard (or soft) modules.

2. Problem Description

(1) Input:

- Hard modules : A set *B* of *m* hard blocks, where each block b_i in *B* has a fixed width and height, w_i and h_i
- Soft modules : A set *B* of *m* soft blocks , where each block b_i in *B* has a fixed A_i and aspect ratio constraints L_i U_i. Each block's aspect ratio (R_i) = width/height, $L_i \leq R_i \leq U_i$. If width and height are not interger, please round it.
- A netlist *E*
- The percentage of white space ratio is predefined and passed by the argument, and the aspect ratio is 1. Then, you can calculate the width w_{fl} and height h_{fl} of the floorplanning region as follows:

$$w_{fl} = h_{fl} = \sqrt{(total\ block\ area) * \big(1 + (white\ space\ ratio)\big)}$$

For example, if the total block area is 1100000 and the white space ratio is 0.1, the width w_{fl} and height h_{fl} of the floorplanning region are as follows:

$$w_{fl} = h_{fl} = \sqrt{1100000 * 1.1} = 1100$$

Then, the coordinates of the lower-left corner and upper-right corner of the floorplanning region are (0, 0) and (w_{fl}, h_{fl}), respectively.

(2) Output:

- The total wirelength of all nets, where the wirelength for each net is defined as the half-perimeter wirelength (HPWL) of the minimum bounding box of pins of the net, and the pin for each block is located at its center.
- The coordinate (x_i, y_i) of the lower-left corner of each block b_i, where x_i, y_i are real numbers.

(3) Objective:

The total wirelength of the floorplanning result and runtime are minimized and the following constraints must be satisfied, where (only hard)blocks can be considered to rotate by 90 degrees.

1. Fixed-outline constraint: $0 \le x_i \le w_{fl} - w_i$ and $0 \le y_i \le h_{fl} - h_i$, $\forall b_i \in B$, which means that each block b_i, whose lower-left corner is located at (x_i, y_i) and width and height are w_i and h_i, respectively, must be entirely inside the following floorplanning region:
$$\boldsymbol{R} = \{ (x, y) \mid 0 \le x \le w_{fl}, 0 \le y \le h_{fl} \}$$
2. Non-overlapping constraint: No two blocks overlap with each other.

3. Input File

(1) The *.hardblocks* file:

The .hardblocks file specifies the name and other information about each block/terminal node in the floorplan. Each line specifies a single block/terminal node.

```
NumHardRectilinearBlocks : 10
// NumHardRectilinearBlocks : number of hard rectilinear block nodes
NumTerminals : 69
// NumTerminals : number of terminal (pad etc.) nodes
sb0 hardrectilinear 4 (0, 0) (0, 82) (199, 82) (199, 0)
//nodeName hardrectilinear vertexNumber vertex1,vertex2,..., vertexN
p1 terminal
// nodeName terminal
```

- *nodeName* is an arbitrary-length alpha-numeric string, and is case-sensitive.
- *hardrectilinear* is a literal which declares that the node is a hard rectilinear block.
- *vertexNumber* is the number of vertices of the corresponding hard rectilinear block.
- *vertex1, vertex2, ..., vertexN* are a list of all vertices of the corresponding hard rectilinear block in a clockwise order, *vertex1* != *vertexN*. Each vertex is a pair of parentheses-enclosed and comma-separated doubles indicating the X-, then the Y- coordinate of the vertex, relative to the lower-left corner of the corresponding hard rectilinear block's bounding box.
- *terminal* is a literal which indicates that the node is a terminal.

(2) The *.softblocks* file:

The .softblocks file specifies the name and other information about each block/terminal node in the floorplan. Each line specifies a single

block/terminal node.

```
NumSoftBlocks : 10
// NumSoftBlocks : number of soft block nodes
NumTerminals : 69
// NumTerminals : number of terminal (pad etc.) nodes
sb0 softrectilinear 16318 0.5 2
//nodeName softrectilinear area lower-bound of aspect ratio upper-bound of aspect ratio
p1 terminal
// nodeName terminal
```

- *nodeName* is an arbitrary-length alpha-numeric string, and is case-sensitive.
- *softrectilinear* is a literal which declares that the node is a soft rectilinear block.
- *area* is the area of the corresponding soft rectilinear block.
- *lower-bound of aspect ratio* is the lower bound of aspect ratio of a soft rectilinear block.
- *upper-bound of aspect ratio* is the upper bound of aspect ratio of a soft rectilinear block.
- *terminal* is a literal which indicates that the node is a terminal.

(3) The *.nets* file:

The .nets file specifies the netlist.

```
NumNets : 118
// NumNets : number of nets
NumPins : 248
// NumPins : number of net-node connections
NetDegree : 2
// NetDegree : number of pins on the net
p1
// nodeName
sb6
```

(4) The *.pl* file:

The .pl file specifies the pin coordinate of each terminal node in the floorplan.

```
p1   0     0
// nodeName XY-coordiante
```

4. Output File

(1) The *.floorplan* file:

The .floorplan file specifies the floorplanning result including the total wirelength of all nets and the coordinate, width and height of the lower-left corner of each block with/without rotating (in softblock testcases rotating

information will be ignored).

```
Wirelength 75563
Blocks
sb0  152  284 100 50 1
// nodeName, lower-left corner coordinate (x,y), width, height, Rotated
sb1  126  179 100 60 0
// nodeName, lower-left corner coordinate (x,y), width, height, Unrotated
```

5. Language/Platform

(1) Language: C/C++

(2) Platform: Unix/Linux

6. Report

Your report should contain the following content, and you can add more as you wish. The more the better.

(1) A cover page containing the title, your name, and your student ID

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

(3) The wirelength and the runtime of each testcase in white space ratios 0.1 and 0.15, respectively.

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

(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 works/papers, please cite the papers and reveal your difference(s).

(6) The details of your implementation. What tricks did you do to speed up your program or to enhance your solution quality?

(7) Please compare your hardblock testcases' results with the top 5 students' results from last year and show your advantage either in runtime or in solution quality. Are your results better than theirs?

- ✓ If so, please express your advantages to beat them.
- ✓ If not, it's fine. If your program is too slow, then what could be the bottleneck of your program? If your solution quality is inferior, what do you think that you could do to improve the result in the future?

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

7. Required Items

Please compress `HW3/` (using tar) into one with the name `CS6135_HW3_${StudentID}.tar.gz` before uploading it to iLMS.

(1) `src/` contains all your sources code, your `Makefile` and `README`.

- `README` must contain how to compile and execute your program. An example is like the one shown in HW2.

(2) `output/` contains all your outputs of testcases for comparison.

(3) `bin/` contains your compiled executable file.

(4) `CS6135_HW3_${STUDENT_ID}_report.pdf` contains your report.

You can use the following command to compress your directory on a workstation:

```
$ tar -zcvf CS6135_HW3_{StudentID}.tar.gz <directory>
```

For example:

```
$ tar -zcvf CS6135_HW3_105062901.tar.gz HW3/
```

8. Grading

- ✓ 70 ~ 80%: The solution quality (wirelength) and the runtime of each testcase, hidden testcases included.
- ✓ 20 ~ 30%: The completeness of your report
- ✓ **10% Bonus**: soft-block testcases, but quality should be better than hard-block testcases.

Notes:

- The executable file name must be named as *hw3*.
- Please use the following command format to run your program.

  `$ hw3 *.hardblocks *.nets *.pl *.floorplan` *`white_space_ratio`*

  or

  `$ hw3 *.softblocks *.nets *.pl *.floorplan` *`white_space_ratio`*

  E.g.: `$ hw3 n100.hardblocks n100.nets n100.pl n100.floorplan 0.1`
- Program must be terminated within 20 minutes for each testcase.
- Grading is based on the total wirelength (primary) and runtime (secondary).

Top 5 students' results from last year (RT = runtime , WL = wirelength)

	n100 RT	n200 RT	n300 RT	n100 WL	n200 WL	n300 WL
1	12.3184	76.3368	121.519	229922	424173	624955
2	80.213	159.382	220.363	199515	374333	543763
3	59.08	170.2	264.93	201928	371590	557651
4	60.08	131.12	237.97	229209	431069	639244
5	0.58	5.28	6.38	295571	544181	819231