Programming Assignment #2: Fixed-outline Floorplanning (due 5pm, April 29, 2018 on-line)

Modified from Problem #1 of the 2003 IC/CAD Contest (Source: Springsoft/Synopsys); please see also Exercise 10.8 of the W&C&C book (pages 628-630).

Submission URL: http://eda.ee.ntu.edu.tw/~yslu/pd18/pa2_submission/
Online Resources: http://eda.ee.ntu.edu.tw/~yslu/pd18/input_pa2.tar.gz

1. Problem Statement

This programming assignment asks you to write a fixed-outline chip floorplanner that can handle hard macros. Given a set of rectangular macros and a set of nets, the floorplanner places all macros within a rectangular chip without any overlaps. We assume that the lower-left corner of this chip is the origin (0,0), and no space (channel) is needed between two different macros. The objective is to minimize the area of chip bounding box and the total net wirelength. The total wirelength W of a set N can be computed by

$$W = \sum_{n_i \in N} HPWL(n_i),$$

where n_i denotes a net in N, and $HPWL(n_i)$ denotes the half-perimeter wirelength of n_i. The objective for this problem is to minimize

$$Cost = \alpha \frac{A}{A_{norm}} + (1-\alpha)\frac{W}{W_{norm}}$$

where A denotes the bounding-box area of the floorplan, A_{norm} is the average area, W is the total wire length, W_{norm} is the average wire length, and α, $0 \leqq \alpha \leqq 1$, is a user defined ratio to balance the final area and wirelength. To compute A_{norm} and W_{norm}, we can perturb the initial solution or m times to obtain m floorplans and compute the average area A_{norm} and the average wire length W_{norm} of these floorplans. The value m is proportional to the problem size. Note that a floorplan which cannot fit into the given outline is not acceptable.

2. Input/Output Specification

Input Format

Each test case has two input files, *input.block* and *input.nets*. The first file (*input.block*) gives the outline size, the number of blocks, and the number of terminals defined in this file. Then the block dimensions are listed, followed by the terminal locations. The file format is as follows:

Outline: <outline width, outline height>
NumBlocks: <# of blocks>
NumTerminals: <# of terminals>

<macro name> <macro width> <macro height>
… More macros

<terminal name> terminal <terminal x coordinate> <terminal y coordinate>
… More terminals

The second file gives the number of nets in the floorplan, followed by the terminal information for each net. The file format is as follows:

NumNets: <# of nets>
NetDegree: <# of terminals in this net>
<terminal name>
… More terminal names

… More "NetDegree" and "terminal name"

The user-defined ratio α is given through the command-line argument. It ranges between 0 and 1.

Output Format

The output file (*output.rpt*) records the problem output. This report consists of six parts: (1) the final cost, (2) the total wirelength, (3) the chip area, (4) the chip width and height, (5) the runtime in seconds, and (6) the bounding-box coordinate for each macro (specified by the lower-left corner and upper-right corner). The report file format is shown above.

<final cost>
// $Cost = \alpha A + (1-\alpha)W$
<total wirelength>
// $W = \sum_{n_i \in N} HPWL(n_i)$
<chip_area>
// area = (chip_width) * (chip_height)
<chip_width> <chip_height>
//resulting chip width and height
<program_runtime>
//report the runtime in seconds
<macro_name> <x1> <y1> <x2> <y2>
<macro_name> <x1> <y1> <x2> <y2>
// (x1, y1): lower-left corner, (x2, y2): upper-right corner
… More macros

3. Command-line Parameters

In order to test your program, you are asked to add the following command-line parameters to your program:

[executable file name] [α value] [input.block name] [input.net name] [output file name]

For example, "Floorplan.exe 0.5 input.block input.nets output.rpt".

4. Example

Figure 1 illustrates an example of the IO files (assume α = 0.5):

Input files (input.block):

Outline: 120 120
NumBlocks: 4
NumTerminals: 0
A 40 50
B 60 50
C 60 50
D 40 50

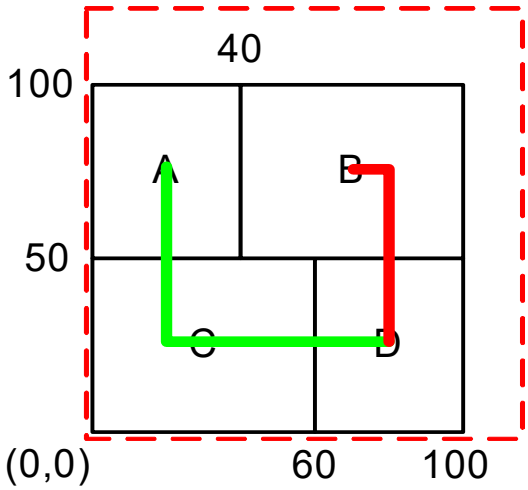

Fig. 1: A floorplanning problem and its solution

(input .nets)

```
NumNets: 2
NetDegree: 3
A
C
D
NetDegree: 2
B
D
```

```
5085
170
10000
100  100
0.24
A    0    50   40   100
B    40   50   100  100
C    0    0    60   50
D    60   0    100  50
```

Output files (output.rpt)

5. Language/Platform

- Language: C, C++, or Java
- Platform: Linux. **Please also develop your programs on the servers in the EDA Union Lab.**

6. Advanced Features

Provide a GUI (Graphic User Interface) to show the floorplanning result.

7. Submission:

You need to submit the following materials in a .tar or a .zip file (e.g., r06943001-p2.zip) at the course website by the deadline: (1) source codes, (2) executable binaries, (3) a text readme file (readme.txt), stating how to build and use your programs, (4) a report (report.doc) on the data structures used in your program.

8. Evaluation

This programming assignment will be graded based on the (1) correctness of the program, (2) solution quality, (3) running time, (4) report.doc, and (5) readme.txt. Please check these items before your submission. *Note that a floorplanning result will not be graded if the running time exceeds one hour on the machines in the EDA Union Lab.*

9. Online Resources

Sample input files (ami33.block/ami33.nets), readme.txt, and report.doc can be found at the course website through the submission link.