Bangladesh University of Engineering & Technology
Department of Computer Science & Engineering
CSE 482: VLSI Design Sessional
Session January 2017

Assignment 2
Slicing Floorplan with Simulated Annealing

---

1 Introduction

In this assignment, you have to implement Slicing Floorplan with Simulated Annealing (SA). In a slicing floorplan, there are vertical and horizontal slices generally denoted by V and H, respectively. In this problem , the blocks are fixed and denoted by the symbols of the set, $S = \{a, b, \ldots, z, 0, 1, \ldots, 9\}$. Each block has a width w and a height h. Note that, width and height are relative, which means the blocks can be rotated to make a floorplan.

A slicing tree can be generated from a slicing floorplan, where the internal nodes of the tree are of type V or H, denoting vertical and horizontal slicing, respectively and the leaf nodes denote the blocks. We only consider skewed slicing trees that means no right child is same as its parent node. This is a binary tree and we can get the Polish expression of the floorplan from the postfix traversal of the tree. So, the postfix expression is a string which only consists symbols of the set S and symbols V and H. We need to find the expression which denotes the best floorplan in this assignment using Simulated Annealing.

Algorithm 1 solves the problem, which is to be implemented. The algorithm takes an initial expression E_0 of a floorplan, initial temperature T_0 for SA, rate of temperature decrease r, and threshold value for temperature limit ϵ as inputs. Output of the algorithm is the best floorplan found during the process. In SA, we can take a bad move to a worse state depending on some probability. In this problem, we will keep a best variable, which will store the best floorplan found in the whole process regardless of the bad and good moves. E_{Best} is used for that purpose. The tasks you need to implement are given in section 4.

2 Neighbor

Neighbor of a state can be generated in three ways. (i) swap two adjacent operands (ii) complement a chain of the expression (interchange V and H in a chain) (iii) swap two adjacent operator and operand. Randomly select any of the three ways and generate a neighbor for a state in each iteration of the algorithm. Keep

in mind that cases may arrive where the way-(iii) cannot generate a neighbor in order to preserve the properties of Polish expression. You need to handle this issue, otherwise the algorithm may get stuck trying to generate a neighbor using way-(iii) sometimes. There are two choices for the way-(iii) shown in the Algorithm 1, which need to be chosen randomly.

3 Cost

For cost calculation, we will only calculate the whole area the blocks take for a floorplan. We can calculate the cost using the bottom-up manner. Note that, blocks that are not square can be rotated. So you need to consider both orientation for area calculation. Keep in mind that during the cost calculation you should keep necessary information to track the orientation of the blocks responsible for a specific cost. As we need to select the floorplan that gives minimum area in the root node, the blocks orientation for that specific floorplan is to be showed in the output. During calculation, you should check if any floorplan dominates others which will speed up the whole calculation process. Follow the reference pdf document for further information.

4 Tasks

1. Randomly generate a valid postfix expression. It denotes the initial state of the slicing floorplan with the blocks. This random generation can be done in different ways. Two ways are given here:

   (a) Generate a binary tree randomly of $2n-1$ nodes, where n is the number of blocks. Then label the nodes of the tree so that the postfix expression is a Polish expression. The leaf nodes are to be labeled randomly with the blocks. The internal nodes are to be randomly labeled with either V or H. Maintain the condition so that the tree is skewed.

   (b) Another way is to generate the binary tree with the ruleset given in Appendix A. The rule set is defined in such a way so that the generated tree is skewed. Continue generation until there are $2n-1$ nodes for n number of blocks.

2. Follow Algorithm 1. Keep the best floorplan during the whole process. There are three operations for neighbor generation. Select any of the three operations randomly during each iteration. Keep in mind the properties of the Polish expression are to be preserved.

3. After neighbor generation, calculate cost using the method given in Section 3. You should store enough information about the blocks so that the final orientation can be stated with the respective cost, which you will be required to show in the output.

4. Select move depending on the cost and probability. Run iteration until the temperature is less than the threshold limit. The algorithm outputs the best floorplan during the whole process.

5. Follow Sections 5 and 6 for instructions on input and output of the problem.

Algorithm 1 SA-Floorplanning

input: E_0, T_0, r, ϵ
output: E_{Best}

$T = T_0$
$E_{Best} = E$
$E = E_0$
while $(T > \epsilon)$ **do**
    $Op = SelectOperation()$
    case Op
    Op_1: Select two adjacent operands e_i and e_j; $E' = Swap(E, e_i, e_j)$
    Op_1: Select a non-zero chain C; $E' = Complement(E, C)$
    Op_3: $done$ =FALSE
        while $(!done)$
            Choice 1: Select two adjacent operand e_i and operator e_{i+1};
                if $(e_{i-1} \neq e_{i+1})$ and $(2N_{i+1} < i)$ **then** $done$ = TRUE
            Choice 2: Select two adjacent operator e_i and operand e_{i+1};
                if $(e_i \neq e_{i+2})$ **then** $done$ = TRUE
        end while
        $E' = Swap(E, e_i, e_j)$
    end case
    $\Delta cost = cost(E') - cost(E)$
    if $(\Delta cost \leq 0)$ or $e^{\frac{-\Delta cost}{T}}$ **then**
        $E = E'$
        if$cost(E) < cost(E_{Best})$ **then** $E_{Best} = E$
    end if
    $T = rT$
end while
return E_{Best}

5 Input

- Input is to be taken from a text file.
- Each line of the input file contains a block symbol followed by its width and height which are integers.

See example input file for clarification.

6 Output

- Run the algorithm 10 times with different initial states. That means, each time generate a random initial expression as stated in task 1 of Section 4 and run the algorithm.
- In output, show each initial expression and respective cost followed by the best expression and respective cost for that run.
- Find the best result from the 10 runs and show as final output.
- Show the initial floorplan and best floorplan in GUI.

See the example output file for clarification.

7 Rules

7.1 General Rules

1. Plagiarism is strongly prohibited. Any kind of plagiarism will result into NEGATIVE marking.
2. NO submission will be granted after deadline.
3. Input must be taken from file.
4. Input format must be same as specified.

7.2 Submission Rules

Submission will be taken in moodle. Steps to be followed are given below.

1. Create a folder named as your 7-digit student ID (i.e. 1205001) in your local machine.
2. Put only the source files (.java or .cpp) in the folder created. (Do NOT copy-paste the whole project)
3. Compress the folder in zip format which must also be named as 7-digit student ID.
4. Submit the zip file in moodle before deadline.

8 Submission Deadline

Submission deadline is **8.00 am on 15 May, 2017 (Monday)**.

9 Conclusion

Good luck for the assignment! For any query please feel free to email me at *ahmaadsabbir@gmail.com*.

Prepared by-
Sabbir Ahmad
Lecturer,
Department of CSE, BUET
ahmadsabbir@cse.buet.ac.bd
ahmaadsabbir@gmail.com
web: teacher.buet.ac.bd/ahmadsabbir

A Ruleset

$$E := EVF|EHG|I$$
$$F := EHG|I$$
$$G := EVF|I$$
$$I := a|b|c|d|e$$