

ECE 1387F - CAD for Digital Circuit Synthesis and Layout Handout #10

Assignment #2 - Branch and Bound Solution to the Partitioning Problem

October 1999

J. Rose

Assignment Date: October 20

Due Date: November 3 (before lecture begins)

Late Penalty: -1 mark per day late, with total marks available = 10

You are to write an implementation of the branch-and-bound partitioning algorithm described in class. It is to take netlist of (equal-sized) blocks and divide it in half so as to minimize the number of nets that connect blocks in the two partitions.

You should write *two* implementations of the branch-and-bound algorithm:

1. An exact one that guarantees to get the correct answer.
2. An approximate one that uses an approximate bound, one of your own devising.

Your program should display its progress and results using X11-based graphics, as in Assignment #1, using the same graphics package. You should design your graphics carefully, so as to illustrate as best you can, the present state of the tree, and the pruning as it progresses.

Your program should use the following netlist file input format, where each line has the following form:

blocknum $netnum_1$ $netnum_2$ $netnum_3$... $netnum_n$ -1

Where **blocknum** is a positive integer giving the number of the cell, and the $netnum_i$ are the numbers of the nets that are attached to that block. Every block that has the same $netnum_i$ on its description line is attached. Note that each block may have a different number of nets attached to it. While each line is terminated by a -1, the whole list is terminated by a blocknum of -1.

Example input file:

```
1 2 3 4 -1
2 5 4 -1
3 5 6 2 -1
4 6 3 -1
-1
```

(over)

In this example, block 1 is connected to nets 2, 3 and 4. Note that each net may be connected to more than two blocks.

Test your Program on the following three testfiles found in the directory ~jayar/1387/a2 on both ECF and eecg: netlist1, netlist2 and netlist3. Do not allow unequal partition sizes. Note that netlist3 may be too large to do using the exact approach.

What to hand in:

1. **The location of the executable**, which must be on an eecg or ecf or ugsparc machine. Make sure that I will have permission to get to the directory and to execute the program, so I can observe it.
2. A short description of the flow of your program. Assuming that I already have a basic knowledge of the branch and bound method, you should describe:
 - i. The branching structure. (i.e. how you designed your decision tree)
 - ii. How you determine the initial “best” solution
 - iii. The bounding function(s) that you use.
 - iv. How you traverse the tree and prune it with the bound.
3. For each version of the program (exact and approximate) give the following:
 - i. A paper plot of the results from the two testfiles,
 - ii. A count of the number of nodes that are actually visited. There will be a prize for the smallest number traversed on netlist2, (tuning to the algorithm to the exact problem is not allowed).
 - iii. The running time for each of the two test cases.
 - iv. The value of the crossing count that you achieved on each netlist.
4. Summarize and discuss your results. How close are the approximate answers to the exact ones?