

Branch and Bound Partitioning `partitioning.py` contains all methods related to the branch and bound partitioning algorithm. First, before running the branch and bound partitioning algorithm, some random valid partitions are applied to the circuit, and get the best cut size so far. Then, assume the number of cells in the circuit is n , create two variables

```
left_remain = ⌊n/2⌋  
right_remain = n - left_remain,
```

which indicate the number of empty spaces in the left and right partitions (If n is an odd number, the right partition will have one more cell than the left partition). Then run recursive branch and bound partitioning,

```
--partition(circuit, assigned, nid, label, left_remain, right_remain),
```

where `assigned` is the current assignment, and `nid` is the next node to assign.

For pruning the search space:

1. Whenever the current label is greater than or equal to the best result, it would be pruned.
2. If the `left_remain` is 0, it will not continue to search the left subtree, and it is the same argument for the `right_remain`. It implies that it prunes left subtree or right subtree.

`self.pruned` is used to record the number of pruned leaves.

Efficiency In order to make my implementation as efficient as possible, instead of calculating the label of the new assignment totally again, only calculate the label changes of nets include the current node to assign (`nid`). Thus,

```
new_label = old_label + (post_label - pre_label)
```

Source Code All of source code and the report are hosted on GitHub (<https://github.com/ZhenpengWu/CAD-Partitioning.git>). A README file containing instructions on how to run the program is also included in the repository.

Code Quality My code is well structured and designed by the object-oriented principle. All methods and class are well documented.

Graphics `tkinter` is employed to implement the GUI. Some basic information of the input file will be listed in the right.

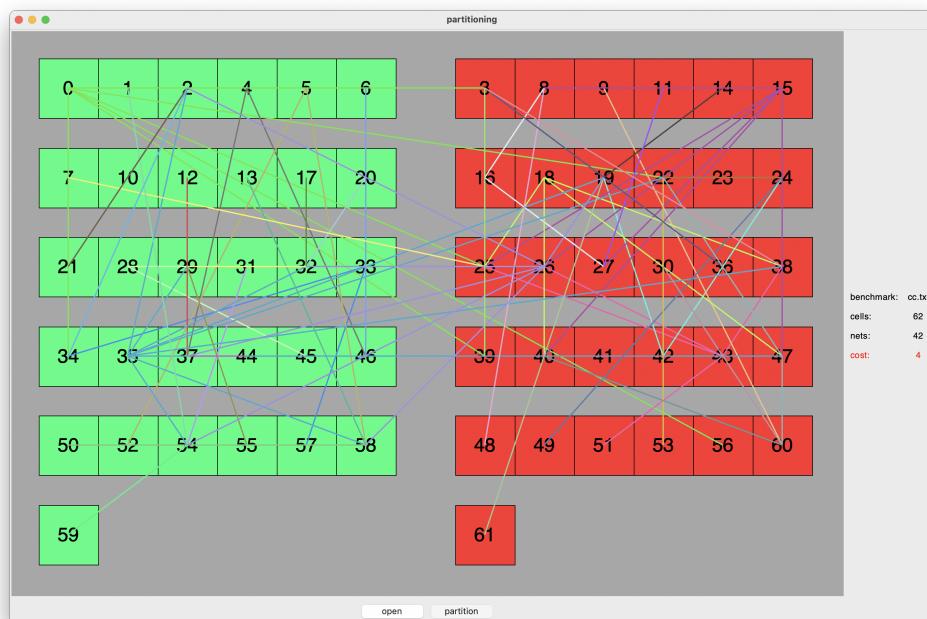


Figure 1: cc

Table of Results

Benchmark	Cost	Run Time
cc.txt	4	about 57 minutes
cm82a.txt	1	less than 1 second
cm138a.txt	4	about 1 second
cm150a.txt	6	about 9 minutes
cm162a.txt	6	about 2 minutes
con1.txt	4	less than 1 second
ugly8.txt	8	less than 1 second
ugly16.txt	16	about 2 seconds
z4ml.txt	3	about 2 seconds
twocm.txt	N/A	N/A

All graphic results are shown in the `docs/images` directory.