ECE 556 PA 1 Report
Json Zhou

1. Code Usage

There is only 1 source code file "fm.cpp", along with 7 provided input files, and a checker.

Compile the code using command "g++ fm.cpp -o fm -O3", this will generate an executable file called "fm".

The fm takes 3 arguments: "Usage: ./fm [input file name] [history size] [debug_output_step]"

[input file name]: the name of the input file, such as "input_1.dat"
[history size]: the program has a terminal mechanism, it looks for the costs of the past [history size] steps, if the costs plateaued or are going back up, the program early terminates the current pass.
[debug_output_step]: the program prints out debug information for each certain number of steps.

To give a start, use 1000 for both number, so give a command such as:

```
./fm input_5.dat 1000 1000
```

The program does 5 runs with different initial cut. The program will generate a file named "output.dat".

2. Code Results:

Timing conducted for [history_size = 1000] and [debug_output_step = 1000], on CAE machine, i5-7500 CPU, using the linux "time" command

Input File Number	Run Time on CAE Machine (might vary due to spec)	Cut Size
0	N/A(very long, way exceed time limit, but no infinite loop and the program can finish)	101474 (I terminated while the program is running as it took hours, no output file provided for this case)
1	0.088s	1510
2	0.167s	2625
3	3.811s	31630

4	9.442s	54268
5	42.356s	168564
6	0.005s	1

Checker Print Out:

"Input_1.dat"
```
tux-133:~/Desktop/workspace/JZ-projectlets/ECE556$ ./checker_linux input_1.dat output_1.dat
[Check] Cut size = 1510 matched!
[Check] Balance passed:: 1485(min) < 1505(G1), 1495(G2) < 1515(max)
==================================
Congratulations! Legal Solution!!
==================================
```

"Input_2.dat"
```
tux-133:~/Desktop/workspace/JZ-projectlets/ECE556$ ./checker_linux input_2.dat output_2.dat
[Check] Cut size = 2625 matched!
[Check] Balance passed:: 3430(min) < 3462(G1), 3538(G2) < 3570(max)
==================================
Congratulations! Legal Solution!!
==================================
```

"Input_3.dat"
```
tux-133:~/Desktop/workspace/JZ-projectlets/ECE556$ ./checker_linux input_3.dat output_3.dat
[Check] Cut size = 31630 matched!
[Check] Balance passed:: 29999.7(min) < 34622(G1), 32044(G2) < 36666.3(max)
==================================
Congratulations! Legal Solution!!
==================================
```

"Input_4.dat"
```
tux-133:~/Desktop/workspace/JZ-projectlets/ECE556$ ./checker_linux input_4.dat output_4.dat
[Check] Cut size = 54268 matched!
[Check] Balance passed:: 74621.2(min) < 75873(G1), 74877(G2) < 76128.8(max)
==================================
Congratulations! Legal Solution!!
==================================
```

"Input_5.dat":
```
tux-133:~/Desktop/workspace/JZ-projectlets/ECE556$ ./checker_linux input_5.dat output_5.dat
[Check] Cut size = 168564 matched!
[Check] Balance passed:: 189332(min) < 191354(G1), 191135(G2) < 193157(max)
==================================
Congratulations! Legal Solution!!
==================================
```

```
tux-133:~/Desktop/workspace/JZ-projectlets/ECE556$ ./checker_linux input_6.dat output_6.dat
[Check] Cut size = 1 matched!
[Check] Balance passed:: 3e-08(min) < 1(G1), 5(G2) < 6(max)
==================================
Congratulations! Legal Solution!!
==================================
```

3. Known Problem and Troubles

1. The file IO handling process can be significantly optimized, but due to time constraints, I couldn't perform much optimization. The current reader does two passes - the first pass is to create all cells and nets; the second pass is to link cells and nets. This entire process can be done using one pass to save system IO time for large files.

2. The runtime for input_0.dat is extremely long. This seems to be related to a self-loop, cells connected to themselves. My program runs, and does give out results, and does not end up in an infinite loop. I have tried different methods, such as ignoring the cell being moved (the cell itself), though I think some of the fixes I performed should have fixed the issue, but eventually I had no luck.