112062559 邱仁緯

(2) How to compile and execute your program, and give an execution example.

Ans:

In HW2/src enter “make” to compile and and the command in make is:

$ g++ -O3 main.cpp -o ../bin/hw2

After compile change the directory to HW2/bin, enter the following command:

$./hw2

For example if we want to test public1.txt then enter the command:

$ ./hw2 ../testcase/public1.txt ../output/public1.out

(3) The final cut size and the runtime of each testcase

	Public1	Public2	Pulic3	Public4	Public5	Public6
Cut size	178	3063	7395	1922	14001	18456
Runtime(s)	0.03	16.19	132.28	0.44	142.86	6.13

(4) The details of your algorithm. You could use flow chart(s) and/or pseudo code to help elaborate your algorithm. If your algorithm is similar to some previous work, please cite the corresponding paper(s) and reveal your difference(s).

Ans:

My algorithm is basically Fiduccia-Mattheyses (FM) Algorithm. Fiduccia and Mattheyses. “A linear time heuristic for improving network partitions,” 19th Design Automation Conf.,1982.

In the paper, there was no explanation of how to perform the initial partition. Therefore, I conducted additional work on the initial partitioning process, such as partitioning by file input order or randomly shuffling the cells and equally partitioning them into two groups.

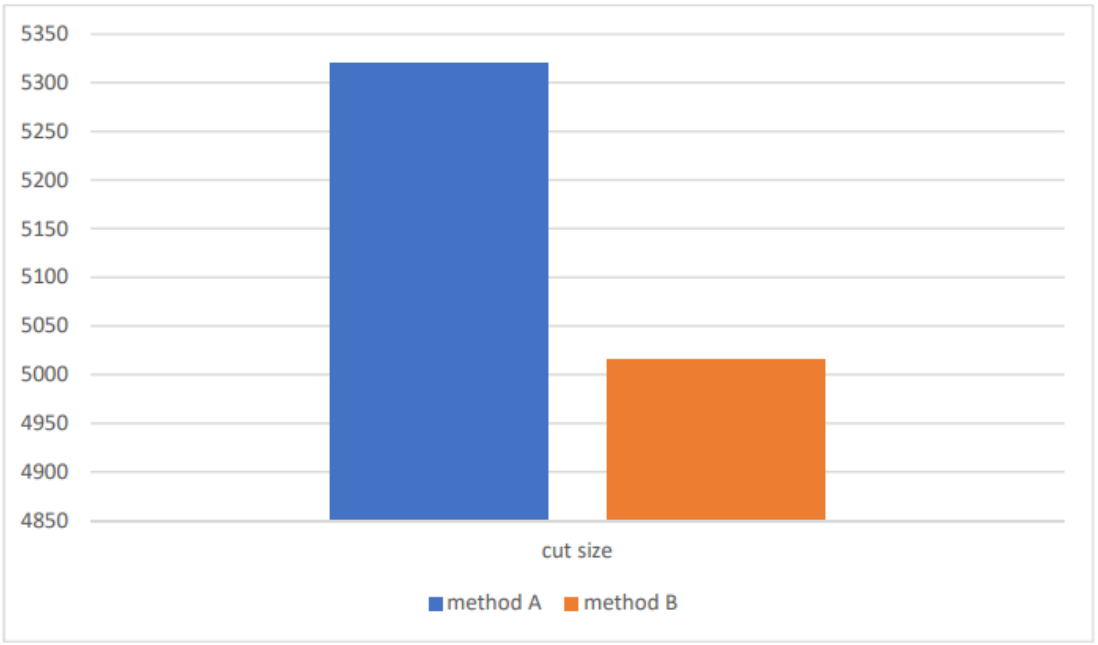
The key difference from the original FM lies in the iteration process. In the original FM, we choose the cell that provides the maximum gain and maintains 'balance.' Instead, I decided not to constrain it to be 'balanced'; I allowed the process to pursue the maximum gain, whether it achieves balance or not. I then perform the area constraint validation later in the process of finding the maximum partial part to ensure that the final partition is legal.

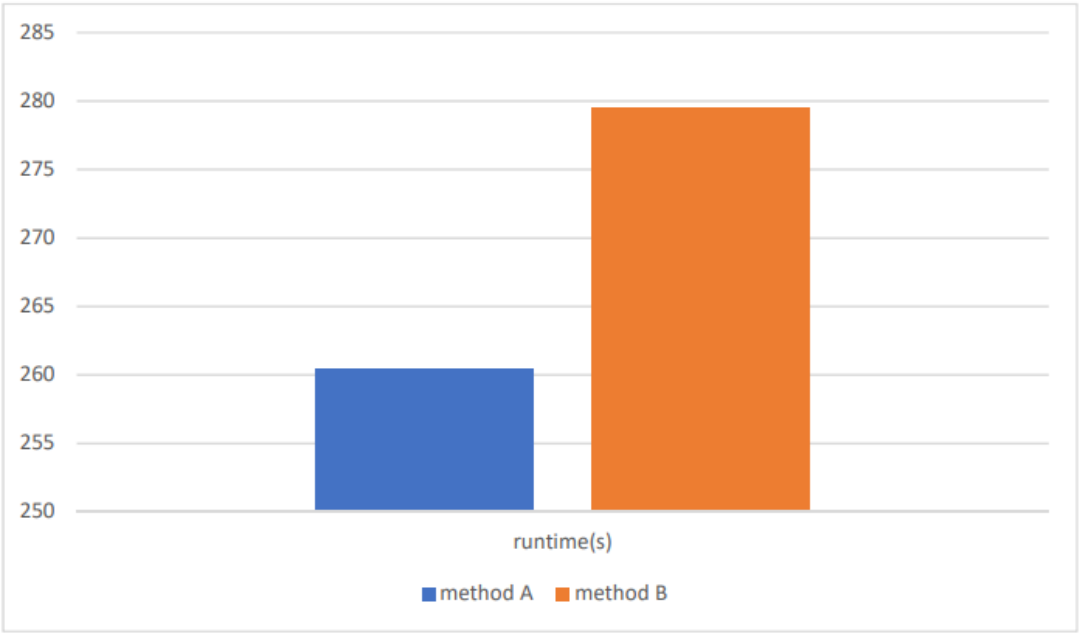
(5) Try your best to enhance your solution quality. What tricks did you use to enhance your solution quality? Also plot the effects of those different settings like the ones shown below.

Ans:

MethodA: by defult input order

MethodB: random shuffle





(6) If you implement parallelization (for algorithm itself), please describe the implementation details and provide some experimental results.

Ans: I didn't implement parallelization.

(7) What have you learned from this homework? What problem(s) have you encountered in this homework?

Ans:

Since I haven't use C++ for awhile, I have a hard time figuring my data structures to implement FM algorithm. I learned a lot from this homework it helped me familiar with C++ STL, makefile and some algorithms.