CS6135_HW5 Report

109062648 柳孟芸

1. **How to compile and execute your program, and give an execution example.**

   ➤ How to compile
   - In "src/" directory, type the command:
     $ make
     It will generate the executable file "hw5" in "bin\" directory.
   - If you want to remove it please type the command:
     $ make clean

   ➤ How to execute
   - In "src/" directory, enter the following command:
     Format:
     $ ..bin/hw5 *.modified.txt *.result
     e.g.:
     $ ../bin/hw5 ../testcase/ibm01.modified.txt ../output/ibm01.result
     --Note: output file will generate in "output\" directory.
   - In "bin/" directory, enter the following command:
     Format:
     $ ./hw5 *.modified.txt *.result
     e.g.:
     $ ./hw5 ../testcase/ibm01.modified.txt ../output/ibm01.result
     --Note: output file will generate in "output\" directory.

```
liumengyunde-MacBook-Pro:src newmileou$ make
g++ -std=c++11 -O3 -lm main.cpp -o ../bin/hw5
liumengyunde-MacBook-Pro:src newmileou$ ../bin/hw5 ../testcase/ibm01.modified.txt ../output/ibm01.result
[ Testcase ibm01 ]
Reading Input File
Start Routing
Writing Output File
[ Total Overflow ] : 0
[ Total Wirelength ] : 62529
[ Total Run time ]: 98.88 sec
```

2. **The total overflow, the total wirelength and the runtime of each testcase.**

	The Total Overflow	The Total Wirelength	Runtime(s)
ibm01	0	62449	104.68
ibm04	158	163724	590.46

3. **The details of your algorithm.**

一開始的initial routing，我是用Lee algorithm去做wave propagation的maze routing，並且參考到NTHU-Route 2.0中的Main stage裡採用monotonic routing的方式和搭配history based cost function，因monotonic限制我們的path只能往上或往右，希望藉由這種方式能讓我們在initial routing中為每條net找出最短路徑。

在做完initial routing後，就去檢查total overflow是否大於0，若是則需Rip-up & Reroute；若否，則直接產生我們最後global routing的結果。

而做Rip-up & Reroute之前，我們必須選出我們需要rip-up的net，而這裡我創了一個priority_queue叫ripup_queue，並將只要有overflow的edge都存進去，再去看哪些net有經過這些edge，之後依序去做reroute，而reroute的部分則是採用非monotonic的maze routing；只要total overflow=0且queue內非空上述部分就是要一直循環做，否則結束global routing產生output結果。

整個演算法的流程為Fig 1所示。

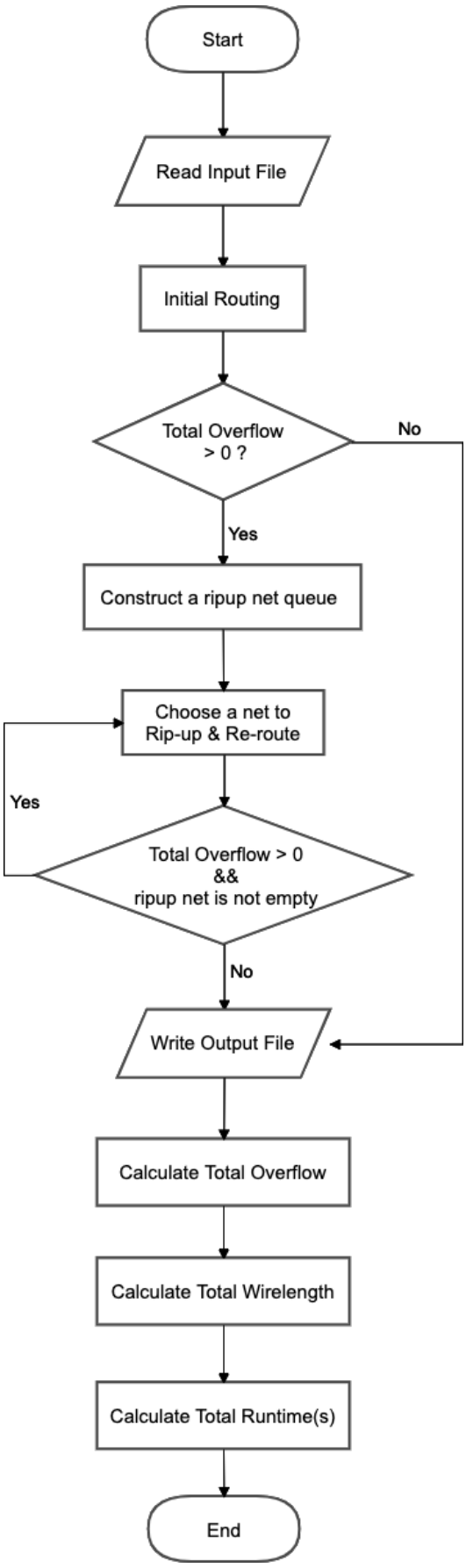
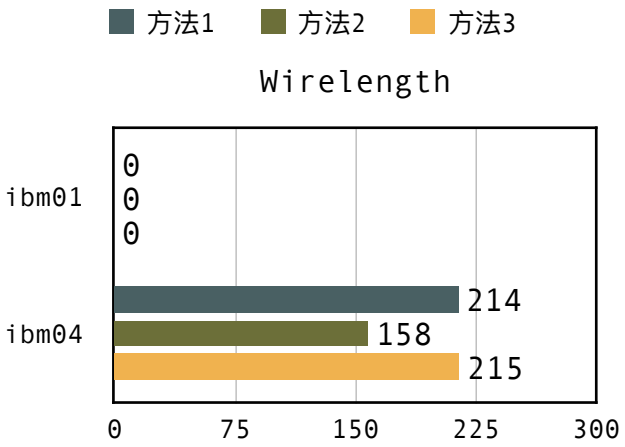


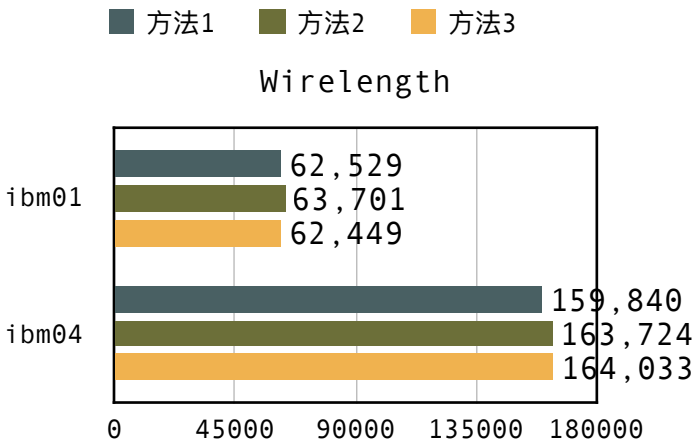
Fig 1. Algorithm flow chart

4. **What tricks did you do to speed up your program or to enhance your solution quality? Also plot the effects of those different settings like the ones shown below.**

- 在runtime方面:我並沒有採用Parallelization或其他方式去加速我的program
- 在quality方面:我嘗試調整adaptive base cost function的β與γ值去enhance solution quality
  - A. 方法1 : β = 5 ; γ = 0.1
  - B. 方法2 : β = 5 ; γ = 0.4
  - C. 方法3 : β = 10; γ = 0.2

$$b_e = 1 - e^{-\beta e^{-\gamma i}}$$





5. **What have you learned from this homework? What problem(s) have you encountered in this homework?**

在此次作業中，我學會如何用Lee演算法去做global routing，並且也更深刻了解整個global routing的流程，但是我覺得cost function的不確定因素太多，導致可能在不同的parameter中會有很多不同的結果，必須記錄一下哪些參數才能導致出最好的結果。在設定cost function上測試了好幾種方法調整我的cost，例如：給予參數不同的值，看哪種cost function才能有更好的答案，因為這一切沒有固定答案，只能透過不停嘗試去找，有時甚至會導致結果變得更差，就只能自己花很多時間去嘗試;還有testcase ibm04的net數量非常多，若沒有設置時間的停損點必定會超過作業要求的時間，這部分將來可能要將程式做平行化或是其他方法才能去解決，讓我不得不佩服現今工業用router的強大。