

HW5 Global Routing Report

107062637 黃博浩

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

compile

在有 Makefile 的路徑底下執行 make 指令，即會產生執行檔 route，若要刪除該執行檔則執行 make clean。

execute

在有執行檔的目錄底下執行：./route <input file path> <output file path> 即可執行指令。

2. The total overflow, wirelength and run time for each test case

61107	10	0	160780	30	0	143484	16	158	161436	106.749
-------	----	---	--------	----	---	--------	----	-----	--------	---------

3. The details of your algorithm

Pseudo Code:

```
Global Routing(){  
    Random_Shuffle()           // 隨機選擇要繞的 net  
    initial_routing()           // 2-direction  
    while( overflow!=0 or round<50 ){  
        ripup()                 // ripup overflow net  
        reroute()               // reroute ripup net  
        count_overflow()  
    }  
    write_file()  
}
```

4. The details of your implementation. What tricks did you do to speed up your program or to enhance your solution quality?

Random Shuffle

繞線的順序會引響程式的結果，因此我在一開始讀完檔後，隨機的將 net vector array 做亂序的排序並記錄當下的 seed，再從中選一個最好的 seed 當作最後執行的結果。

Initial Routing

在第一次繞線時，只選擇 2 個 direction，在 source 與 target 所形成的 bounding box 內進行繞線。

Reroute

在每次 Reroute 時，我會優先選擇 overflow 較高的 net 先 Reroute。

Cost function when wave propagate

在 cost function 的部分，我是參考 NTHU-Route 2.0[1] 這篇 paper，除了考慮當前通過的 edge congestion 外，也考慮進去 history term，整體的 cost function 為 $\text{edge_congestion} * \text{history_cost}$ ，edge congestion 的公式為 $(\text{demand} + 1) / \text{capacity}$ ^5，而只要 edge 有 overflow 則將目前通過的 grid 的 history cost 加 1。

[1] NTHU-Route 2.0: A Robust Global Router for Modern Designs

5. Please compare your results with the top 3 students' results from last year and show your advantage either in runtime or in solution quality

ibm01		ibm02			ibm03			ibm04		
WL	CPU	OF	WL	CPU	OF	WL	CPU	OF	WL	CPU
59371	8.887	0	156556	14.417	0	142680	3.567	71	160342	158.518
59519	3.911	0	157394	6.398	0	143000	2.911	63	159136	55.571
63495	6.57	0	166148	8.832	0	143894	13.214	101	162460	122.495
61107	10	0	160780	30	0	143484	16	158	161436	106.749

不論在 wire length 或 overflow，我都輸去年的前 3 名，只有在 execution time 比別人快了一些，在這次優化的 constraint 中沒有將 wire length 考慮進去，覺得自己在效能上還有很多該補強的地方。

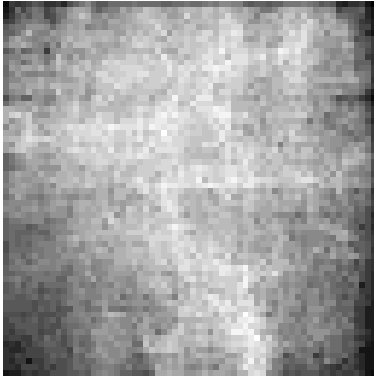
6. What have you learned from this homework? What problem(s) have you encountered in this homework?

我覺得這次作業是 5 個作業中花最多時間的一次，之間花了很多時間重新更改資料結構，光是要如何記錄 overflow 的 edge 或需要 reroute 的 net 就嘗試了 2 種資料結構，從原本只考慮 source、target 間最短距離的 A* search，到更改為每次都要先做 wave propagate 的 Maze Routing，每次的修改都讓自己開始懷疑人生，但隨著 overflow 在 3000~4000 上上下下，到最後能將 overflow 降為 0，真的帶給了自己很大的成就感，除了更了一些 Debug 技巧，也對於自己在寫 code 的邏輯上有了更好的幫助。

7. Bonus

白色為 congestion 越大，黑色為較少 overflow

Testcase01



Testcase02



Testcase03



Testcase04

