

CS6135 HW2 Report

107033137 謝郁楷

1. How to compile and execute the program:

In "HW2/src", enter the following command:

```
$ make
```

The executable file hw2 will be generated in "HW2/bin/".

Using command:

```
$ make clean
```

to remove %.o files and the executable file

--How to Run

In this directory, enter the following command:

Usage: `../bin/<exe> <net file> <cell file> <output file>`

For example:

```
$ ../bin/hw2 ../testcases/p2-1.nets ../testcases/p2-1.cells ../output/p2-1.out
```

or in "HW2/bin", enter the following command:

Usage: `./<exe> <net file> <cell file> <output file>`

For example:

```
$ ./hw2 ../testcases/p2-1.nets ../testcases/p2-1.cells ../output/p2-1.out
```

The result showed in terminal:

```
[u107033137@ic51 bin]$ time ./hw2 ../testcases/p2-5.nets ../testcases/p2-5.cells ../output/p2-5.out
cut size 125811
cut size 123081
cut size 122577
cut size 122546
cut size 122503
cut size 122497
cut size 122495
cut size 122493
cut size 122493
IO time is 0.7 seconds
execution time is 10.57 seconds
best cut is 122493
```

2. The final cut size and the runtime of each testcase:

Testcases	Min cut size	Run time		
		real	user	sys
p2-1	25	0.01	0.00	0.00
p2-2	656	0.05	0.03	0.01
p2-3	9547	7.79	7.57	0.19
p2-4	50297	11.34	11.10	0.20
p2-5	122493	11.71	11.42	0.24

3. I/O time and execution time:

Testcases	I/O time	Execution time
p2-1	0	0
p2-2	0.02	0.02
p2-3	0.26	7.48
p2-4	0.43	10.86
p2-5	0.74	10.9

Input 檔案的 size 越大，需要的 I/O time 就越長。而在 execution 的地方，case 4 跟 5 的 cell 跟 net 數量一樣多，但差在 5 的每個 net 連到的 cell 多很多，可能因為 net 連到太多，所以大部分的 net 都不在 critical，不會帶來 gain，所以比較容易達成終止條件，所以速度跟 p2-4 差不多。

4. The details of your implementation containing explanations of the following questions:

I. Where is the difference between your algorithm and FM Algorithm described in class?
Are they exactly the same?

大部分都是一樣的，不過我的程式多跑了很多次FM的演算法。我將第一次終止時的結果，當成下一次initial的partition狀態，再下去跑一次FM演算法，直到結果都沒有再進步為止。這樣做法會比只算一次的結果好非常多。還有終止條件的地方，slide是當 $G \leq 0$ 時結束，不過我發現終止條件改成 $G \leq -10$ 可以再讓結果好一些，不過繼續減小終止條件並不會一直讓結果再更好。

II. Did you implement the bucket list data structure? If so, is it exactly the same as that described in the slide? How many are they?

我就是使用slide裡的那種方式implement的，除了bucket list外，也有implement一個指標的array可以指到想要刪掉的那個cell那，直接將他從list中移除。

III. How did you find the maximum partial sum and restore the result?

我使用一個array假設叫a，來記每個cell最一開始的位置，再用一個vector，將每次移動的cell一個一個push_back進去。在整個過程中，我會記下我在移動第幾次時拿到最好的結果，假設10次，我就將a中在vector前10個的cell的位置改變，就可以restore最好的結果了。

IV. What else did you do to enhance your solution quality or to speed up your program?

我不確定是不是enhance，不過slide中更新bucket list時，應該是記下每個cell最後各移動了多少，再一次搬動。不過我的作法是找到critical net後，會有一些cell要的gain要加1或減1，這時我就直接移動那些gain被加1或減1的cells在bucket的位置。這個做法比較簡單可以也不用再用額外的資料結構去記哪個cell被移動了多少。還有上面有提到的我改了FM演算法的終止條件為 $G \leq -10$ ，以及我做了很多次FM，並將上個結果當成下個的初始條件。

5. Please compare your results with the top 5 students' results from last year and show your advantage either in runtime or in solution quality. Are your results better than them?

我的cut size結果在input小的時候差他們的差滿多的，能跟他們比較的只有p2-4跟p2-5的結果，特別是p2-5結果很不錯。速度上看起來還算是水準之上，不算花很多時間。我覺得他們很可能在implement時並不是完全照slide中的選擇每次要搬動的cell的方式，或是可能有再將整個initial的partition重分，重算過好幾次找出最好的結果，因為可以看到他們有些花非常長的時間來計算。未來如果要improve的話，我想也會從這方面去嘗試。

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

因為是電機系的大學部學生，對C++不是非常熟，一開始花非常多時間才熟悉。遇到最大的問題可能還是code沒有辦法整理的很好，其實我在完成作業後有去稍微查過網路上開源的程式碼，發現自己對於C的寫法還有很大的進步空間。另外一個問題是C在編譯多個檔案的問題，我想嘗試把程式碼整理成像別人一樣，不過編譯時一直遇到一些沒找到function reference或是重複define的問題，花了不少時間找原因。最後就是我一直找不到可以把結果再提升的方法，特別是p2-1，很想知道該如何才可以切到只剩下5，真的非常厲害。