1 王竑智 109062542

2 在 HW2/src make

在 HW2/bin ./hw2 *.nets *.cells *.out

*代表任意檔案名，

3 前後有稍微修改程式，所以跑出的結果未必會和表格相同

Case	p2-1	p2-2	p2-3	p2-4	p2-5
Run time	6.14	9.19	293.68	537.01	442.20
Cut size	6	526	9881	48073	123805

4

Case	p2-1	p2-1	p2-3	p2-4	p2-5
I/O time	0.01	0.01	0.16	0.27	0.26
Computation time	6.13	9.18	293.52	536.74	441.94

我想資料集越大，則越小的比例是花在 I/O 上，畢竟 I/O 頂多 linear time，Algorithm 有可能到 2 次或更高，而資料越多則越多需要計算的。

5

5.1 大部分皆相同，我依照上課的演算法流程去寫我的程式，唯獨在 critical nets 那部分比較沒有照著判斷，大致結構流程相同。

5.2 有，我建立兩個 bucket list，分別是屬於 A 區跟 B 區，並全程只使用這兩個 bucket list 做調整。

5.3 用一條 linked link 紀錄選 max gain 的過程，並記錄 partial sum，同時記錄最佳的 partial sum，linked list 建立過程由後往前，restore 時按照順序從頭往下依序搬回即可。

5.4 我的結果並不理想，不論是時間還是結果的品質，在資料越大的情況下，我的結果是可以接近 top five 的，但時間上就差很多，我覺得瓶頸主要在我為了確保正確性，所以在所有可以修正數值的的地方做重算，並且避免邏輯錯誤，我也避開了很多可以做減少運算的判斷，還有若是我重算的部分使用平行化，也許時間可以減少許多，而在結果的品質上，因為程式裡使用到了許多的指標，我不確定我的邏輯有充分的考慮到可能的情況，我想若是加入其他可以減少時間的判斷，或許也能夠同時幫助到結果的品質。

5.5 若是要說協助結果的品質的話，我想能做修正的地方我直接重算，可以加上平行化，比起修正數值所要考慮其他的互斥存取變數，重算可能會有比較好的結果

5.6 應該是指標型態的運用更熟了，為了減少變數的使用，我都盡可能的使用最一開始紀錄的 cell, net 資訊，這加重了資料維護的負擔，可以減少資料傳遞不同步的狀況。

5.7 No.

6 Hidden case

Case	h2-1	h2-2	h2-3
Runtime	332.01	389.12	571.96
Cut size	74580	99443	139888
I/O time	0.20	0.24	0.32
Computation time	331.80	388.77	571.50