Fixed-outline Floorplanning

姓名：李宇哲
學號：B10732040
校系：國立台灣科技大學 四資工四乙

/**
助教好，我的程式有用到 GNUplot 畫出結果圖，環境需要先安裝 GNUplot
若安裝後依舊無法畫出，或是程式無法成功執行完畢，可以註解掉 src/main.cpp 裡第 47 行 fp->plot(BEST_FLOORPLAN)，以停用繪製功能。

```
46      fp->writeOutputFile(tEnd);
47      fp->plot(BEST_FLOORPLAN);
48      return 0;
```

**/

一、 SA 參數設定

m1 = Block 數量 + Net 數量	// 事先擾動 floorplanning 的次數，計算出 area、wirelength 和 aspect ratio 大略的平均數，已進行後續標準化的動作。
m2 = Block 數量 + Net 數量	// 承上，繼續擾動 floorplanning 的次數，計算出平均 cost 的變化量。
M = 100 * Block 數量	// Uphill 次數的上限
N = 2 * M	// 同一溫度擾動次數的上限
T = abs(avgCostDelta / ln0.99)	// 初始溫度
TemLimit = 0.00001	// 溫度下限
r = 0.93	// 冷卻權重
rejectLimit = 0.95	// Reject 的比例上限

二、 資料結構

Class Terminal

儲存 terminal 和 block 中 pin 點(block 的中心點)的資訊。

Class Block

儲存 block 資訊，包含長、寬和最終結果擺放的位置。

Class Net

儲存連線資訊，包含 degree 和連接的所有 terminal 點。

Class Node

B*-Tree 中的節點資訊，包含節點所代表的 block、節點的 left/right child、paren 和節點的座標位置

Class B_Tree

包含 b*-tree、contour line 和 recover 等相關的資料結構和函數，如下：

(1) B*-Tree 相關：

root: 儲存 b*-tree 的 root

Perturbation 函數: 包含 rotate、swap 和 delete&insert

computeCoord_dfs(): 用深度優先將 b*-tree 轉成實體擺置。

(2) Contour line 相關:

_ctl: 用 std::list 來儲存 contour line 的線段

updateContourLine(): 當有 block 擺入時，更新 contour line 的線段

(3) Recover 相關：

_rec: 當執行 Perturbation 函數時，會同時用 std::list 儲存 node 的所有變動紀錄，將來需要恢復到上一個的 b*-tree 時，只需要把_rec 儲存的紀錄用 Last-In-First-Out(LIFO)的方式復原就可以了。如此一來，rotate 和 swap 復原只需要 O(1)的時間複雜度，delete&insert 只需要 O(logN)的時間複雜度。

Class Floorplanner

儲存 floorplanning 相關資訊、實作 SA 和繪製結果圖。

三、 我的發現

1. 把程式碼長度短且頻繁被呼叫的函數變成內嵌函數（在函數前加上 inline），可讓程式的 runtime 變小。
2. 執行 SA 時，若擾動遭 reject 的話，需要恢復成前一步的 b*-tree，因此需要隨時記錄當前和前一步的 b*-tree，該操作若是對整顆 b*-tree 做的話將十分耗時。因此我利用 recover 資料結構，記錄擾動時對 node 造成的所有變動，如此一來，將來需要復原 b*-tree 時，只需將 recover 的紀錄回推就行了。Swap 和 rotate 還原可從原本的 O(N)複雜度降為 O(1)，而 delete&insert 則是從 O(N)降為 O(logN)。
3. 把 b*-tree 轉成實體位置的函數，我實驗了用深度優先(DFS)和廣度優先(BFS)的方法實作，DFS 在大部分測資可以獲得較好的結果。

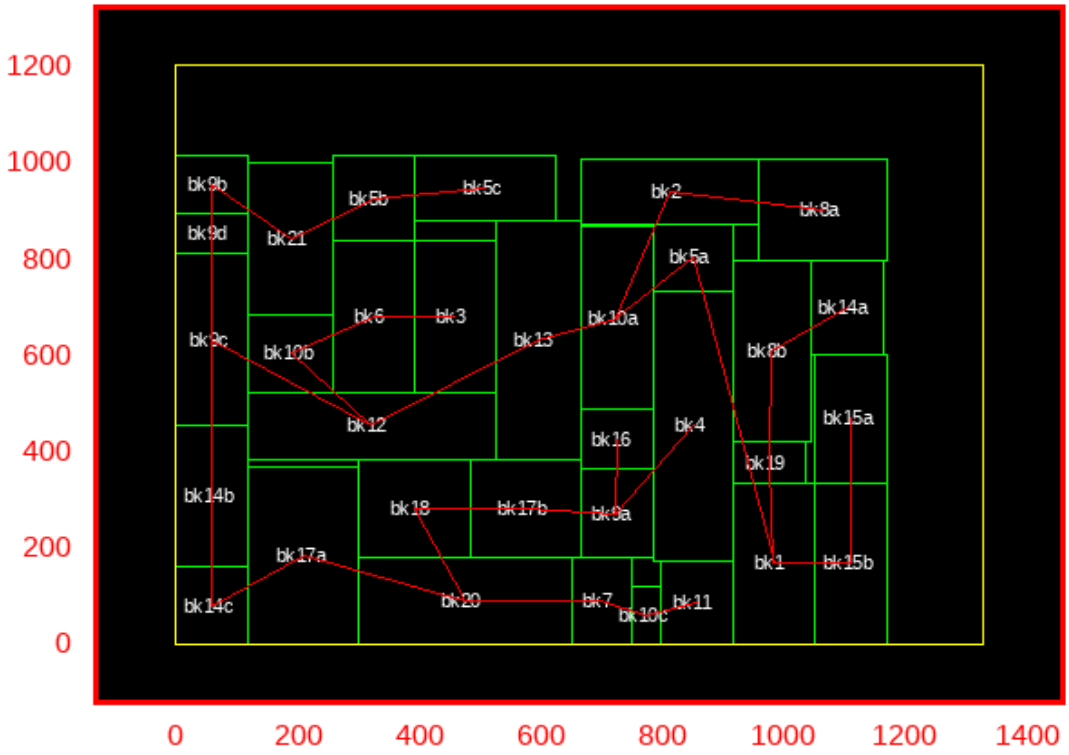
四、 實驗結果

1. 數據

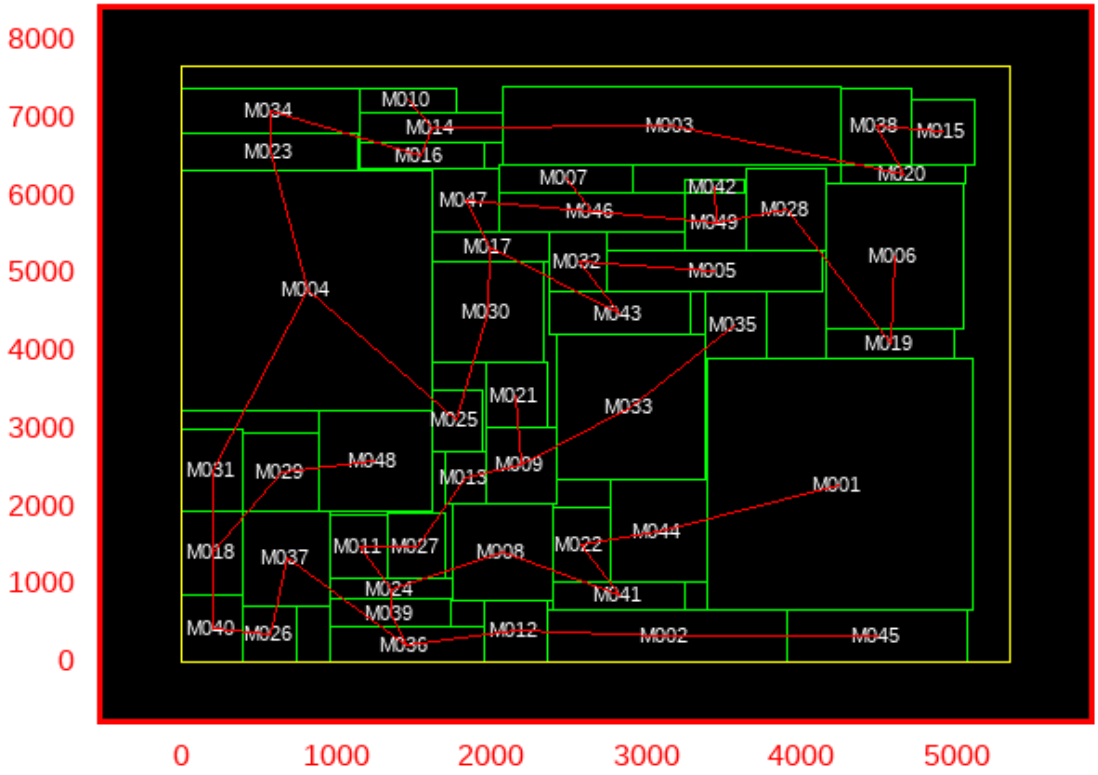
令 α = 0.5

Test Case	Area	Wirelength	Runtime(s)	Final cost
ami33	1186535	114859.5	2.1882	650697.25
ami49	37844660	1910825	5.386446	19877742.5
apte	47313280	869674	0.348551	24091477
hp	9169076	337224	0.340667	4753150
xerox	19831084	607578	0.560284	10219331

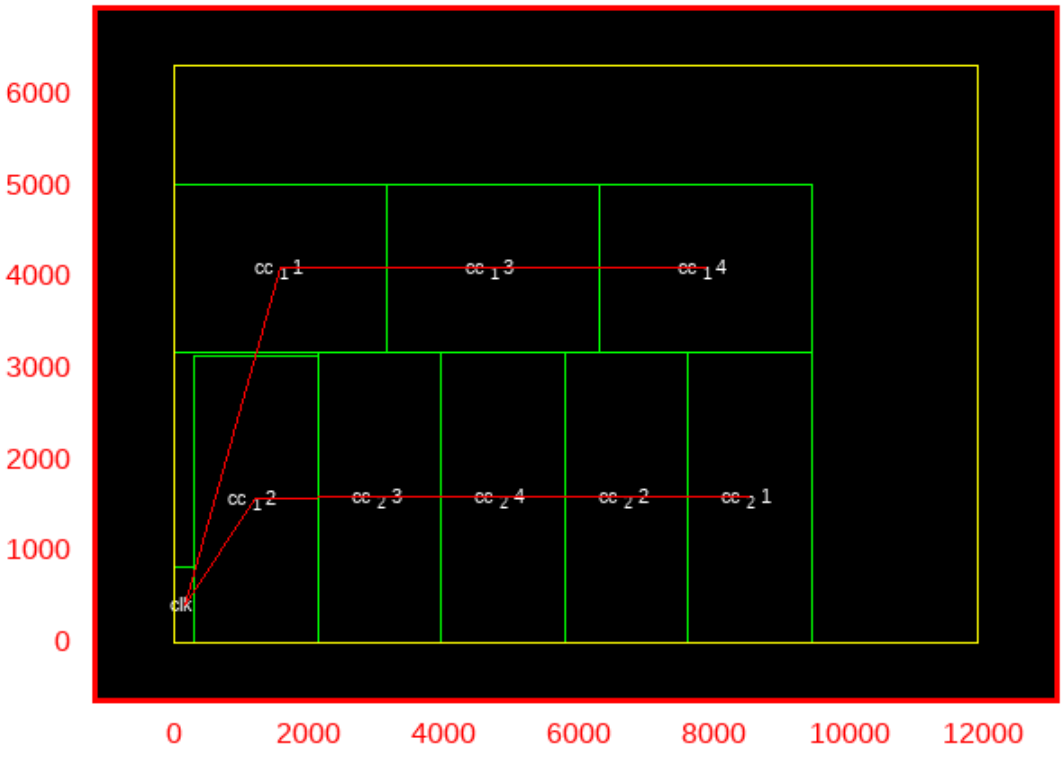
2. 結果圖（使用 GNUplot 繪製）



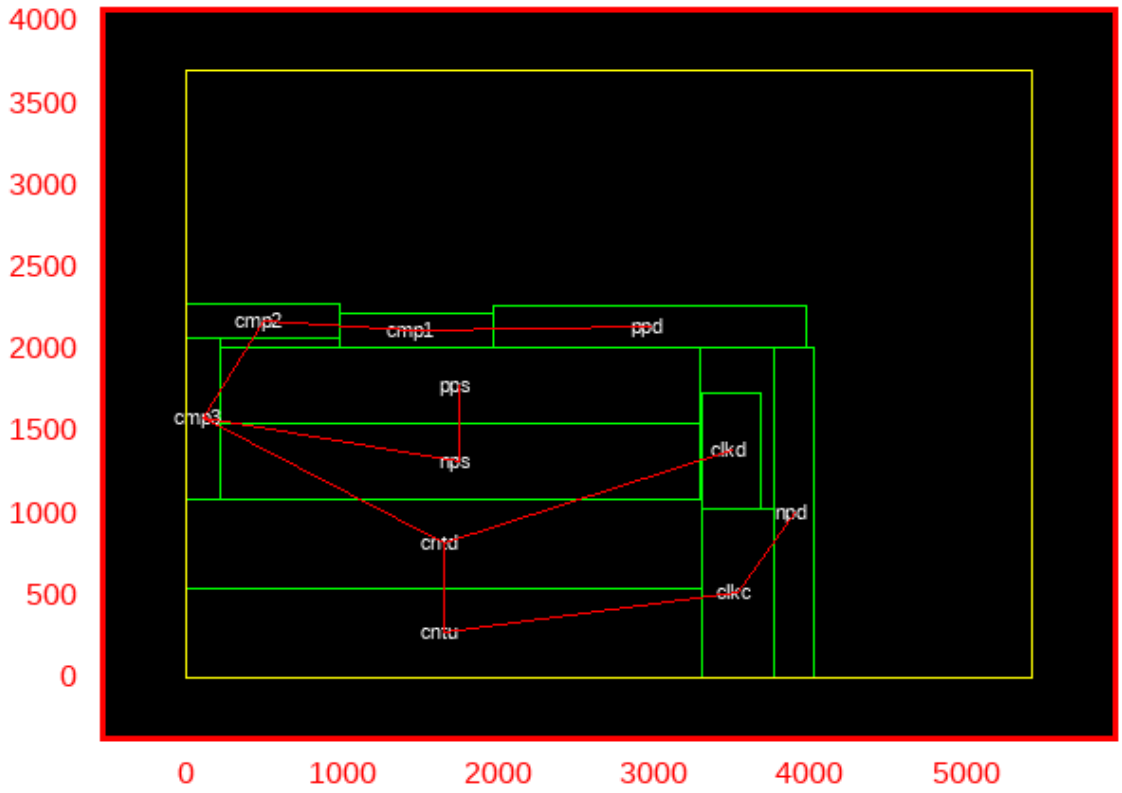
圖(一) ami33



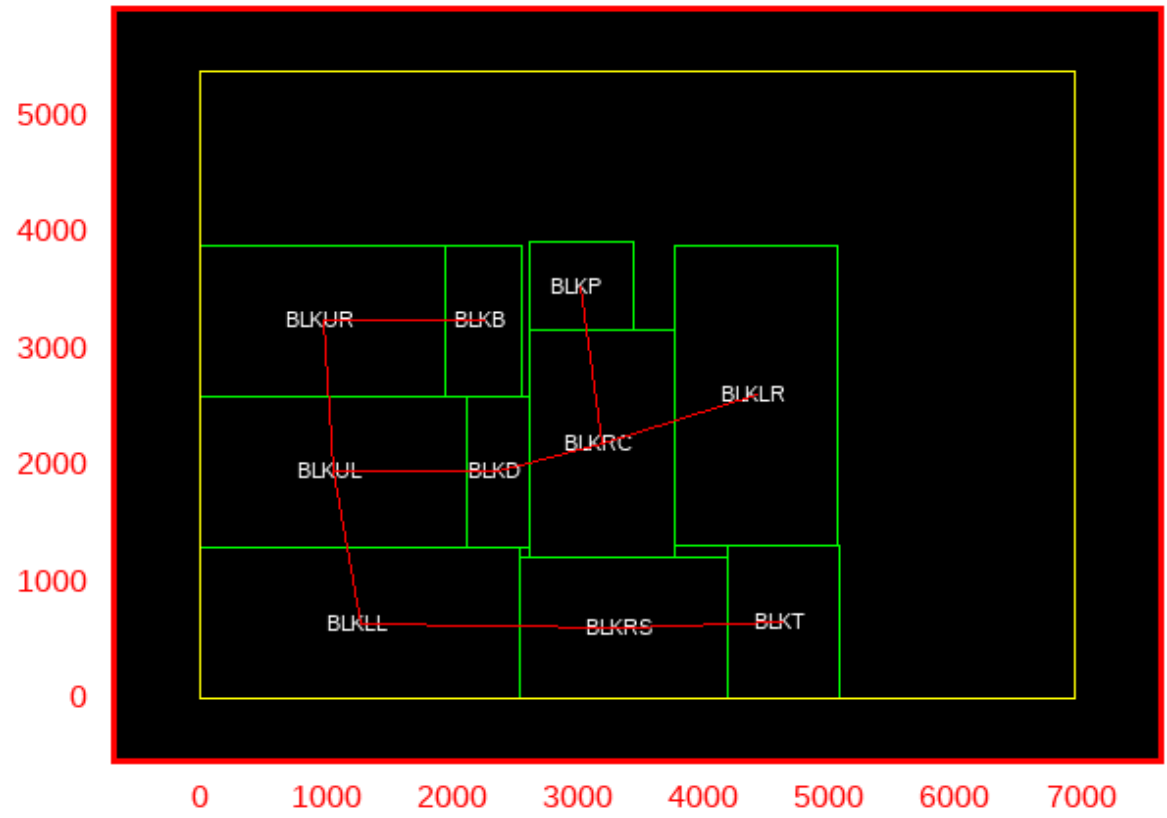
圖(二) ami49



圖(三) apte



圖(四) hp



圖(五) xerox