Fixed-outline Floorplanning

姓名：李宇哲

學號：B10732040

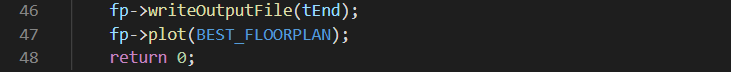
校系：國立台灣科技大學 四資工四乙

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

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。



\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

1. **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

1. **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。

1. 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

1. 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

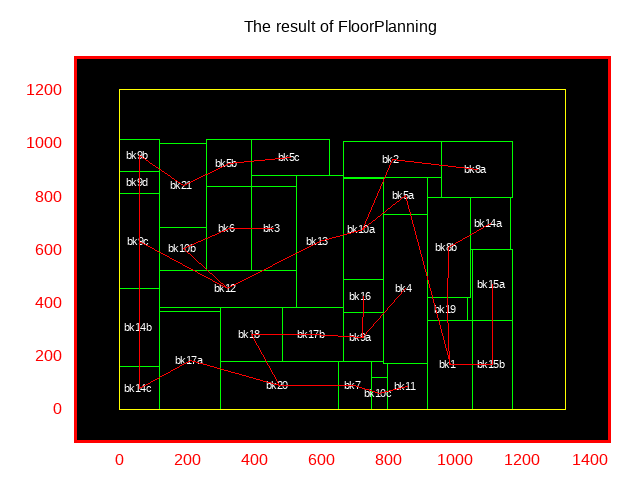
儲存floorplanning相關資訊、實作SA和繪製結果圖。

1. **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。
2. **實驗結果**
   1. **數據**

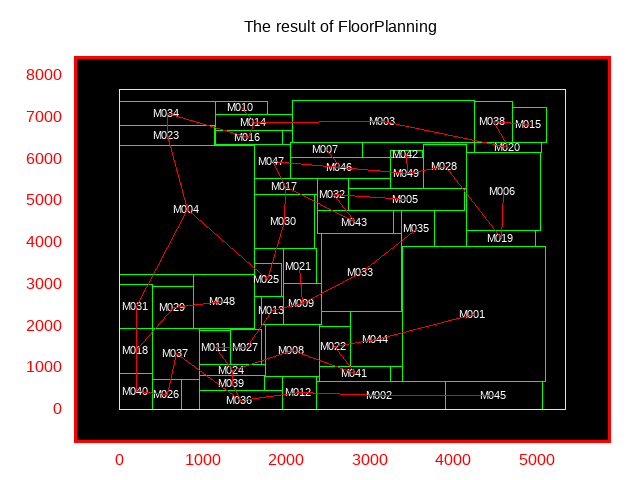
令α= 0.5

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 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 |

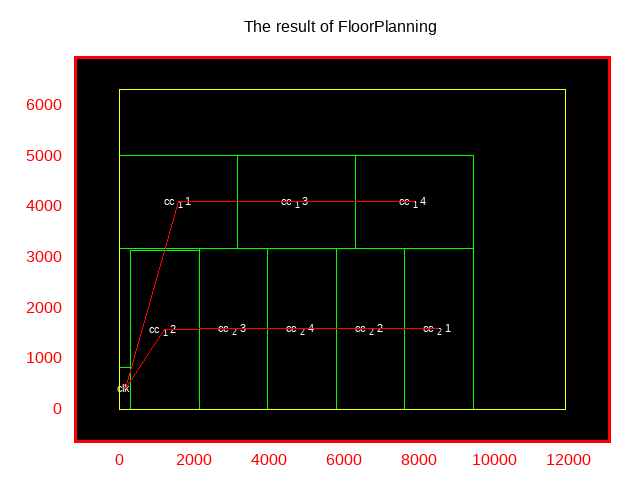
* 1. **結果圖（使用GNUplot繪製）**



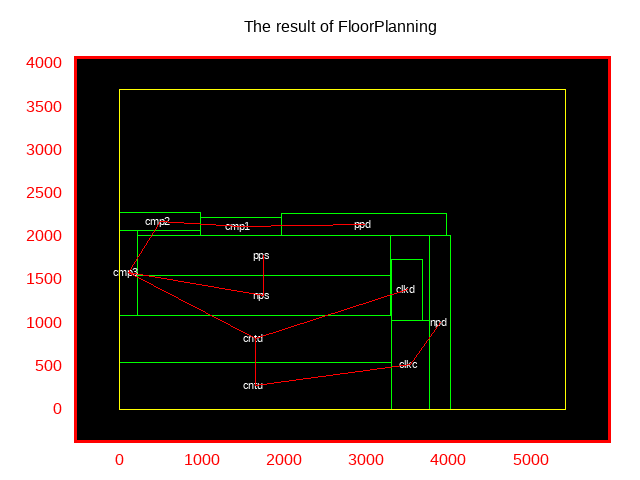
圖(一) ami33



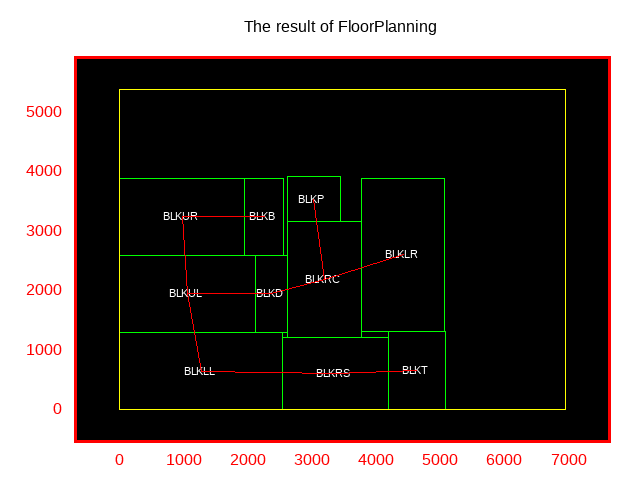
圖(二) ami49



圖(三) apte



圖(四) hp



圖(五) xerox