Programming Assignment Report

B04502139 電機四 戴瑋辰

1. 演算法流程
   1. **讀檔**

將整個Block和Net讀進來，並記錄所有資訊

* 1. **建立B\*Tree**

初始化一個代表整個Floor Plan的B\*Tree，為了讓其有很多種不同的變化，初始化的時候以左右交替的方式建立B\*Tree

* 1. **利用Simulated Annealing先將所有的Block放到Outline裡面**

由於放到Outline之中並不等價於minimize面積，所以在cost function的設計上我選擇講義第4章68頁的實作方式並先把wire length的部分抽調。

由於這樣的設計考慮了整個Outline的形狀，最後比較容易將所有的Block放進Outline中

* 1. **利用Simulated Annealing將已經塞入Outline中的Floor Plan做Wire Length的優化**

由於已經將Block全部塞入Outline中，因此此步驟目的僅是為了減少wire length。引此cost function直接定義如下

* 1. **做完步驟III和IV後產生的Floor Plan為最終結果**

1. 資料結構

* B\*Tree

比較特殊的地方只有在記錄horizontal contour的部分。不同於講義上double linked list的作法，我將contour以map的形式儲存。，由於map為RBTree的資料結構，其時間複雜度為O(log n)，相較linked list的O(n)低。

1. 問題討論

由於本次作業目標是同時optimize Area和Wire Length，因此一開始我只有將cost function定義與作業說明上相同。

但是不管我如何調整Simulated Annealing的參數，永遠都沒辦法fit進outline裡面。因此我最後決定將兩部分分開做。

因為要放入Outline中才算是一個legal的解，所以第一步是將所有的Block塞進去。因此參考講義的作法，我將這一個phase的cost function訂為

由於此cost的第一項和第二項相差非常大，為了讓其對cost有相同的影響力我再分別乘了和，此時cost就會介在0到1之間。

Simulated Annealing的部分則是從10000度開始以0.9的比率decay到1度，每一個溫度重複做50000次。其作法與講義上的幾乎一樣唯有算uphill move threshold的地方做了一些改變，由於cost的部分指介於0到1之間，因此最後將threshold訂為

這樣一來threshold就會介於正常的範圍之間。

至於minimize wire length的部分就非常trivial，直接把wire length當作cost，再重新跑一次Simulated Annealing並reject所有無法放入outline的解就可以完成。

1. 結果

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Cost | Wire Length | Area | Width | Height | Run Time |
| ami33 | 692336.75 | 69905.5 | 1314768 | 1204 | 1092 | 29.52s |
| ami49 | 20580570 | 963893 | 40197248 | 5278 | 7616 | 63.02s |
| apte | 24110738 | 660873 | 47560604 | 9478 | 5018 | 6.5s |
| hp | 5077005.5 | 189175 | 9964836 | 3766 | 2646 | 5.54s |
| xerox | 10452806 | 535822 | 20369790 | 5215 | 3906 | 10.42s |