**Physical Design PA2-Floorplan**

**GIEE**

**學號：R05943092**

**姓名：曾育為**

* 1. **演算法流程 (Algorithm Flow)**

Simulated

Annealing

演算法流程圖

配合流程圖，將程式運作流程再加上文字敘述如下：

1. 讀入檔案，將 Block和Terminal各自建立資料結構，並將Net連結所有連結的Terminal
2. 將所有的Block依照讀入順序建立一個Complete binary tree
3. 進入Simulate Annealing流程中，依據cost function判斷是否接納這個解

(a). 隨機選取三個operation調整tree ( Rotate、Swap、Delete and Insert )

(b). 重新packing計算contour

(c). 判斷是否能將block放入Outline內，並計算HPWL求得cost function

(iv). Output the optimal solution

* 1. **資料結構 (Data Structure)**

下面將介紹程式中主要的資料結構：

1. Block: 代表輸入檔的block

class Block

{

private:

//parser.cpp 將Input file資訊讀入所記錄的名字、寬、高

string \_name;

int \_width;

int \_height;

//Btree.cpp. 紀錄此Ｂlock在Btree中的位置資訊

int \_id;

int \_id\_p;//父點ID

int \_id\_l;//左子點ID

int \_id\_r;//右子點ID

//contour.cpp 紀錄Floorplan的位置資訊（此Block左下方點的(x,y)）

int \_x;

int \_y;

};

1. Terminal: 代表輸入檔( .block)的terminal

class Terminal

{

public:

string \_name;//Terminal的名稱

int \_x;//Terminal X座標

int \_y;//Terminal Y座標

Block \*block;//所依附的Block

};

1. Net: 代表輸入檔( .nets)

class Net

{

private :

int \_degree;//此條net含有的terminal個數

vector<Terminal\*> net\_of\_terminals; //儲存此條net的terminal

};

（iv）Btree : 代表所建的tree 資訊

class Btree

{

private :

fstream fp;//讀檔與輸出檔

int \_outline\_w;//outline的width

int \_outline\_h;// outline的height

int \_block\_num;//block總個數

int \_terminal\_num;//terminal總個數

int \_net\_num;//net總條數

bool \_fit\_outline;//是否可以放入outline中

int \_root\_id;//紀錄root的ID

vector<Block> blocks;//儲存所有的block

vector<Terminal> terminals;// 儲存所有的terminal

vector<Net> nets;// 儲存所有的net

};

* 1. **問題與討論：**

**(3-1). 比較Cost Function的結果：**

* **固定參數：**
* **參數列a value = 0.5**
* **初始溫度T0 =100000**
* **T=r\*T (r=0.98)**
* **比較Function :**

**[Function 1]. (0.5\*max\_X\*max\_Y)+(0.5)\*HPWL()\*100+100\*\_penalty**

**[Function 2]. (0.5\*max\_X\*max\_Y)+100\*\_penalty**

* **實驗數據：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Case** | **Function** | **HPWL** | **Chip Area** | **Time** |
| **apte** | **1** | **740359** | **2.42278e+07** | **2.54576 sec.** |
| **2** | **894900** | **2.49684e+07** | **0.345457 sec.** |
| **xerox** | **1** | **220197** | **1.12315e+07** | **2.56397 sec.** |
| **2** | **686697** | **1.07271e+07** | **1.37166 sec.** |
| **hp** | **1** | **178339** | **4.9925e+06** | **1.86742 sec.** |
| **2** | **251363** | **5.03891e+06** | **1.75623 sec.** |
| **ami33** | **1** | **71333** | **677444** | **6.8353 sec.** |
| **2** | **131201** | **679718** | **4.51843 sec.** |
| **ami49** | **1** | **879242** | **2.04901e+07** | **9.52 sec.** |
| **2** | **1.76943e+06** | **2.09502e+07** | **4.2779 sec.** |

* **結論：**

**Cost Function.1 和.2 差別於 (0.5)\*HPWL()\*100， Case (如hp、ami33、ami49)，block數量大於十個之Function.1的Runtime會比Function.2快二倍關係，但所得的HPWL也較佳，結果大約為Function.2的2/3。**

**(3-2). 提升初始溫度討論：**

* **固定參數：**
  + **以ami49作為比較對象**
  + **參數列a value=0.5**
  + **T=r\*T (r=0.98)**
  + **Cost Function: (0.5\*max\_X\*max\_Y)+100\*\_penalty**
* **實驗數據：**

|  |  |  |  |
| --- | --- | --- | --- |
| **T0初使溫度** | **HPWL** | **Chip Area** | **Time** |
| **10000** | **1.75083e+06** | **2.13366e+07** | **3.60529 sec.** |
| **100000** | **1.76943e+06** | **2.09502e+07** | **4.2779 sec.** |
| **1000000** | **1.7957e+06** | **2.07905e+07** | **4.52929 sec.** |
| **10000000** | **1.68839e+06** | **2.04232e+07** | **7.80109 sec.** |

* **結論：**

**初始溫度越高所需要時間越久，但所得的HPWL 和Chip Area不一定能得到較好結果，HPWL 和Chip Area結果取決於Cost Function。**