CS6135 VLSI Physical Design Automation

Homework 4: Global Placement

徐祈 112065504

A. The wirelength and the runtime of each testcase

testcase	Wirelength	Runtime
Public1	83827612	56.46
Public2	16697557	127.17
Public3	633868222	207.51

B. The details of your algorithm. You could use flow chart(s) and/or pseudo code to help elaborate your algorithm. If your method is similar to some previous work/papers, please cite the papers and reveal your difference(s).

Initial placement 我用的是 center initial 的方法，就是把所有 modules 都放在中間。後面在慢慢優化他。

從老師的 slides 中發現 objective function 為

$$\text{minimize} \sum_{e \in E} c_e \times \text{WL}_e(x,y) + \lambda \sum_{b \in B} (D_b(x,y) - T_b)^2$$

而它的 High level 的觀念可以用以下 pseudo code 描述

```
void objective(){
    f = 0.0;
    // Log-Sum-Exponential
    for (net in _placement)
        for (module in net)
            get sum of xMax: xMaxSum;
            get sum of xMin: xMinSum;
            get sum of yMax: yMaxSum;
            get sum of yMin: yMinSum;

        f += lambda * (
                log(xMaxSum) +
                log(xMinSum) +
                log(yMaxSum) +
                log(yMinSum)
            );

    // density (Bell-shape)
    binDensity = [0...0];

    for (b in Bins)
        for (module in _placement)
            if (module is not fixed)
                b += bellShapeFunc(module, b);
        f += lambda * (b - tarDensity)^2;
}
```

其中 bellShapeFunc 為估計 density 的 function:

$$D_b(x,y) = \sum_{i \in V} C_i \times \widetilde{\Theta}_x(b,i) \times \widetilde{\Theta}_y(b,i)$$

$\widetilde{\Theta}_x(b,i) =$

- $a \times d_x^2 \; if \leq d_x \leq w_b/2 + w_i/2$
- $b \times (d_x - w_b - w_i/2)^2 if w_b/2 + w_i/2 \leq d_x \leq w_b + w_i/2$
- $0 \; if \; w_b + w_i/2 \leq d_x$

where $a = \frac{4}{(w_b+w_i)(2w_b+w_i)}, b = \frac{4}{w_b(2w_b+w_i)}$

$D_b(x,y)$ 的架構可用以下 pseudo code 描述

```
bellShapeFunc(module, b)
{
    mW = module.width();
    mH = module.height();

    aX = 4 / ((binW + mW) * (2 * binW + mW));
    bX = 4 / (binW * (2 * binW + mW));
    aY = 4 / ((binH + mH) * (2 * binH + mH));
    bY = 4 / (binH * (2 * binH + mH));

    a = binIdx % binCut;
    b = binIdx / binCut;
    c = module.area() / binArea;
    // c: normalization factor so that the total potential
    // of a block equals its area.

    dX = oldX - ((a + 0.5) * binW + _placement.boundryLeft());
    ABSdX = abs(dX);
    dY = oldY - ((b + 0.5) * binH + _placement.boundryBottom());
    ABSdY = abs(dY);

    thetaX = thetaByConstrs(ABSdX, mW, binW, aX, bX);
    thetaY = thetaByConstrs(ABSdY, mH, binH, aY, bY);
    return c * thetaX * thetaY;
}
```

$\Theta_x(b,i)$可以用以下 pseudo code 描述

```
thetaByConstrs(ABSdX, mW, binW, aX, bX)
{
    if (ABSdX <= mW * 0.5 + binW * 0.5)
        return 1 - aX * (ABSdX^2);
    else if (ABSdX <= mW * 0.5 + binW)
        return bX * (ABSdX - binW - 0.5 * mW)^2;
    else
        return 0;
}
```

$\Theta_x(b,i)$的 gradient 可以用以下 pseudo code 描述

```
thetaGradByConstrs(ABSdX, mW, binW, aX, bX, dX, thetaY, c)
{
    signX = dX's sign;
    if (ABSdX <= mW * 0.5 + binW * 0.5)
        return c * (-2 * aX * signX * ABSdX) * thetaY;
    else if (ABSdX <= mW * 0.5 + binW)
        return c * 2 * bX * signX * (ABSdX - (binW + 0.5 * mW)) * thetaY;
    else
        return 0;
}
```

reference:

1. Teacher's slides unit6
2. NTUplace3's intro slides

C. Try your best to enhance your solution quality. What tricks did you do to enhance your solution quality?

我覺得這次作業的實作上，大家會差不多，頂多是評估 wirelength 的 function 從 log-sum-exponential 換成 weighted average 可以讓線長估的更趨近 HPWL。所以比較難從實作上去優化 (如果 runtime 不算在優化範圍內的話)因此優化重點變成是找到好的參數，包括 gamma, stepsize, epoch, numIter, lambda 的初始值和每次要 increase 的量等等。

在我寫好一份確定為正確的版本後，給了一組 hyperparameter

- numIter = 50;
- no.setStepSizeBound(10000);
- EPOCH = 2;
- ef.lambda += 1000 (initial=1000)
- gamma = 10;

結果如下

testcase	Wirelength	Runtime
Public1	324193780	20.86
Public2	13014416	26.21
Public3	2644255425	91.96

和我最後的結果比較可以發現真的差很多 (尤其是 public1 & 3)，它是用以下參數跑出來的

- numIter = (epoch == 0) ? 100 : 50;
- no.setStepSizeBound((bTop - bBottom) * 2);
- EPOCH = 3;
- ef.lambda += 2000 (initial=4000)
- gamma = chipH / 700;

testcase	Wirelength	Runtime
Public1	83827612	56.46
Public2	16697557	127.17
Public3	633868222	207.51

D. Please compare your results with the previous top 5 students' results and show your advantage in solution quality. Are your results better than theirs?

- **If so, please express your advantages to beat them.**

- **If not, it's fine. If your solution quality is inferior, what do you think that you could do to improve the result in the future?**
  Public 1 & 3 還可以，雖然沒有前五名但應該也不差。
  Public 2 遠遠落後前五名，猜測可能是用 WA 估線長才能做到好的表現，因為我發現 public2 的晶片比較小，HPWL 特別容易受 gamma 大小影響，越小的 gamma 越能減少 HPWL，但 LSE 受限於 numerical issue 無法估計很準確的 HPWL，因此十分可惜的沒能盡到當年前五的行列。也許之後有時間可以實作 WA 來比較結果是否如果所猜測會比較好!