VLSI Physical Design HW4 Report　111062678 卓榮祥

➢ The wirelength and the runtime of each testcase.

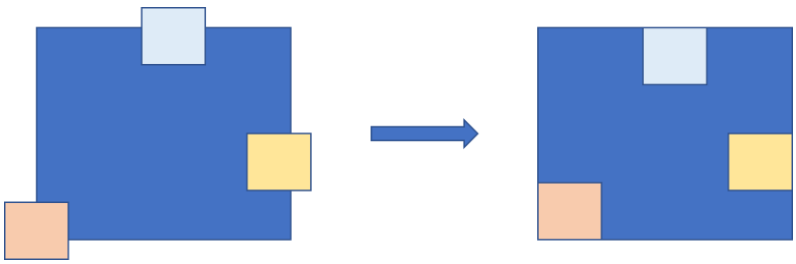
	Public1	Public2	Public3
Global HPWL	56220821	15078586	467278573
Legal HPWL	106068960	17319337	741201829
Detail HPWL	78682692	14444642	580169927

➢ The details of the algorithm.

```
/* @@@ TODO
 * 1. Understand above example and modify ExampleFunction.cpp to implement the analytical placement
 * 2. You can choose LSE or WA as the wirelength model, the former is easier to calculate the gradient
 * 3. For the bin density model, you could refer to the lecture notes
 * 4. You should first calculate the form of wirelength model and bin density model and the forms of their gradients ON YOUR OWN
 * 5. Replace the value of f in evaluateF() by the form like "f = alpha*WL() + beta*BinDensity()"
 * 6. Replace the form of g[] in evaluateG() by the form like "g = grad(WL()) + grad(BinDensity())"
 * 7. Set the initial vector x in place(), set step size, set #iteration, and call the solver like above example
 * */
```

程式的架構主要是遵照助教給的 TODO list 實作，wirelength model 採用的是 LSE，bin density model 我則是採用投影片上的 Bell-Shaped Function。在計算 gradient 時參考了線上微分網站：WolframAlpha(https://www.wolframalpha.com/)來驗證結果是否正確。

值得一提的是，在 solver 解完後，有些 module 有可能會超出 chip boundary，因此我將 module 沿著 x 方向或 y 方向平移進 module 中，試圖產生出合法的解，如下圖。



- **Enhance the solution quality**

這次作業如果要增進 solution quality，我覺得不斷的調整參數是必經過程，參數能夠決定 solver 的收斂速度、module 的分佈等，因此在程式架構完成後，我就不斷地在反覆調整參數，希望透過調整 seed、step size 或者是 number of iteration 等參數，讓產生出的解品質更好些。

- **Compare the results with the previous top 5 students' results and show your advantage in solution quality. Are your results better than theirs?**

wirelength			
Rank	**Public1**	**Public2**	**Public3**
1	68783367	8778312	456197589
2	79542407	9155930	478967869
3	83860394	9783498	463664700
4	73804994	11105419	413161709
5	80176617	10921603	539864787
My	**78682692**	**14444642**	**580169927**

我一開始做的結果和 top 5 有些落差，在實作一開始我將 objective function

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

中的 Tb 當作 average density，在這之下不管怎麼調整其他參數都使跑出來的解遠遠落後去年 top 5，最後仔細看發現其實 Db 這個 function 目的是要近似 module 所占該 bin 的面積，因此為了方便估算，我將一開始 random 擺放時

的 bin density 的平均作為 Tb，我希望在以 random 能夠合法化的前提下，將 random 所產生的 bin density 作為參考值，更改過後發現產生出的 solution quality 有變得更好。另外，在調整參數時，發現 bin 的個數也會影響到 solution quality，例如在所有參數固定之下，在 bin 個數為 49 時可以讓 public1 的 wirelength 減少至 78682692。因此我會依據每個 case 在哪個 bin 數目表現比較優異而個別設定。將來如果要再試著進一步做優化的話，我覺得可以在一開始決定每個 module 的位置時讓每個 net 所連到的 module 越靠近越好，換句話說，盡量在最一開始就能夠先壓低 wirelength，再做 Analytical Approach，感覺會再更進一步的增加 solution quality。另外，這次的做法中參數的組合有非常多種，在 LSE 中有 smoothing parameter、objective function 中有 bin density 的比重、初始化擺放 module 的 random seed 等，這些參數都需要多花時間慢慢調才可以找到更好的解。