CS6135_HW4 Report

109062648 柳孟芸

1. **The wirelength and the runtime of each testcase.**

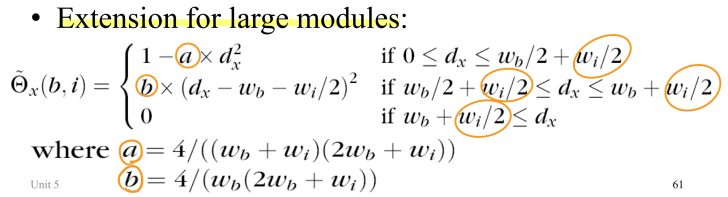
	ibm01	ibm05	ibm09
Wirelength	208981186	20606340	1608405308
Runtime(s)	46.0 sec (0.8 min)	108.0 sec (1.8 min)	195.0 sec (3.2 min)

2. **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).**

```
/* @@@ 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 main(), set step size, set #iteration, and call the solver like above example
 * */
```

演算法我大致上按照助教給的TODO指示流程去修改範例code

1. 我採用上課教的Log-Sum-Exponential Function來當作Wirelength Function，並且將前面係數設為gamma = (width + height) / 100.
2. Bin Density Function我則是參考lecture notes的p.59~61，並且採用Smooth bell-shaped function(以下圖示)來作為bin density model



3. 在不斷嘗試各種數據之後：
   iteration方面我總共做三大回合，第一大回合先做150個小回合，之後都做35個小回合；
   max_step_size 則設為 (boundary_right - boundary_left) * 5。
   並且給各個testcase固定的seed去穩定solution quality:
   ibm01 -> seed = 1545382299
   ibm05 -> seed = 1545407982
   ibm09 -> seed = 1608223220

3. **What tricks did you do to speed up your program or to enhance your solution quality ?**

- 在runtime方面：我並沒有採用Parallelization或其他方式去加速我的program，但因為我的code並不冗長，iterations也設的不大，所以時間也不會跑很久。
- 在quality方面：
  1. 我跑過好幾次code去找能使各個testcase solution最好的seed。
  2. 嘗試過稍微小修改bin density function的內容，但跑出來的結果都沒有原來lecture notes裡的好，所以還是按照原來課堂所教的function。
  3. 也試著將iterations加大，看看讓時間跑久一點是否能改善solution quality，但最後卻發現solution雖然有改善，但都是很些微的改善不然就是反而增加了WL，runtime卻增加了快要兩倍之多，所以有點不太平衡。

4. **Please compare your results with the previous top 5 students' results and show your advantage either in runtime or in solution quality. Are your results better than theirs?**

   我的runtime跟去年的top5比起來是可以排在第三名左右的，可能是因為我的code算是精簡所以時間方面不會跑太久；而solution quality算是排在top5的尾巴，可能是因為我單純的就用上課教的方式去做global placement，並沒有想其他更好的演算法去精進答案，或是要試試看其他更好的objective function，或許在有限的時間內我的solution quality跟runtime就能更好。