CS6135_HW3 Report

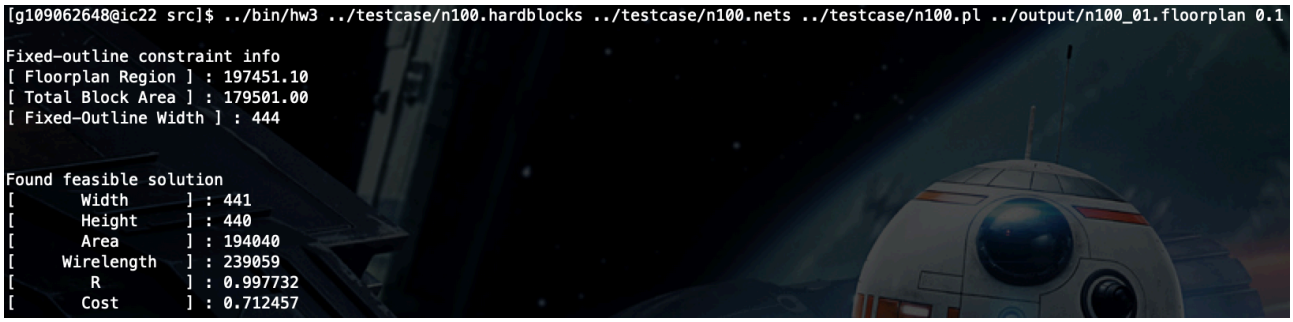
109062648 柳孟芸

1. **How to compile and execute your program, and give an execution example.**

   - How to compile
     - In "src/" directory, type the command:
       $ make
       It will generate the executable file "hw3" in "bin\" directory.
     - If you want to remove it please type the command:
       $ make clean

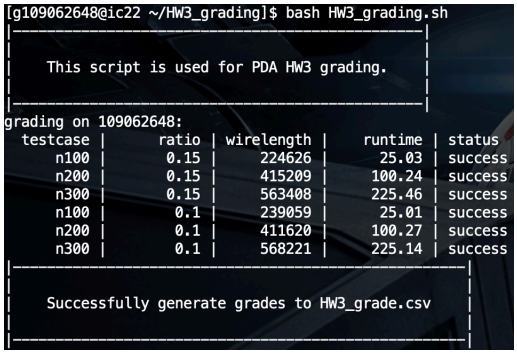
   - How to execute
     - In "src/" directory, enter the following command:
       Format:
       $ ..bin/<exe> <hardblocks file> <nets file> <pl file> <output file> <dead_space_ratio>
       e.g.:
       $ ../bin/hw3 ../testcase/n100.hardblocks ../testcase/n100.nets ../testcase/n100.pl ../output/n100_01.floorplan 0.1
       --Note: output file will generate in "output\" directory.
     - In "bin/" directory, enter the following command:
       Format:
       $ ./<exe> <hardblocks file> <nets file> <pl file> <output file> <dead_space_ratio>
       e.g.:
       $ ./hw3 ../testcase/n100.hardblocks ../testcase/n100.nets ../testcase/n100.pl ../output/n100_01.floorplan 0.1
       --Note: output file will generate in "output\" directory.

   - Execution example



2. **The wirelength and the runtime of each testcase with the dead space ratios 0.1 and 0.15, respectively**

	Wirelength			Runtime(s)		
	n100	n200	n300	n100	n200	n300
0.1	239059	411620	568221	25.01	100.27	225.14
0.15	224626	415209	563408	25.03	100.24	225.46



3. **Please show that how small the dead space ratio could be for your program to produce a legal result in 20 minutes.**

	n100	n200	n300
the smallest dead space ratio	0.06	0.075	0.08

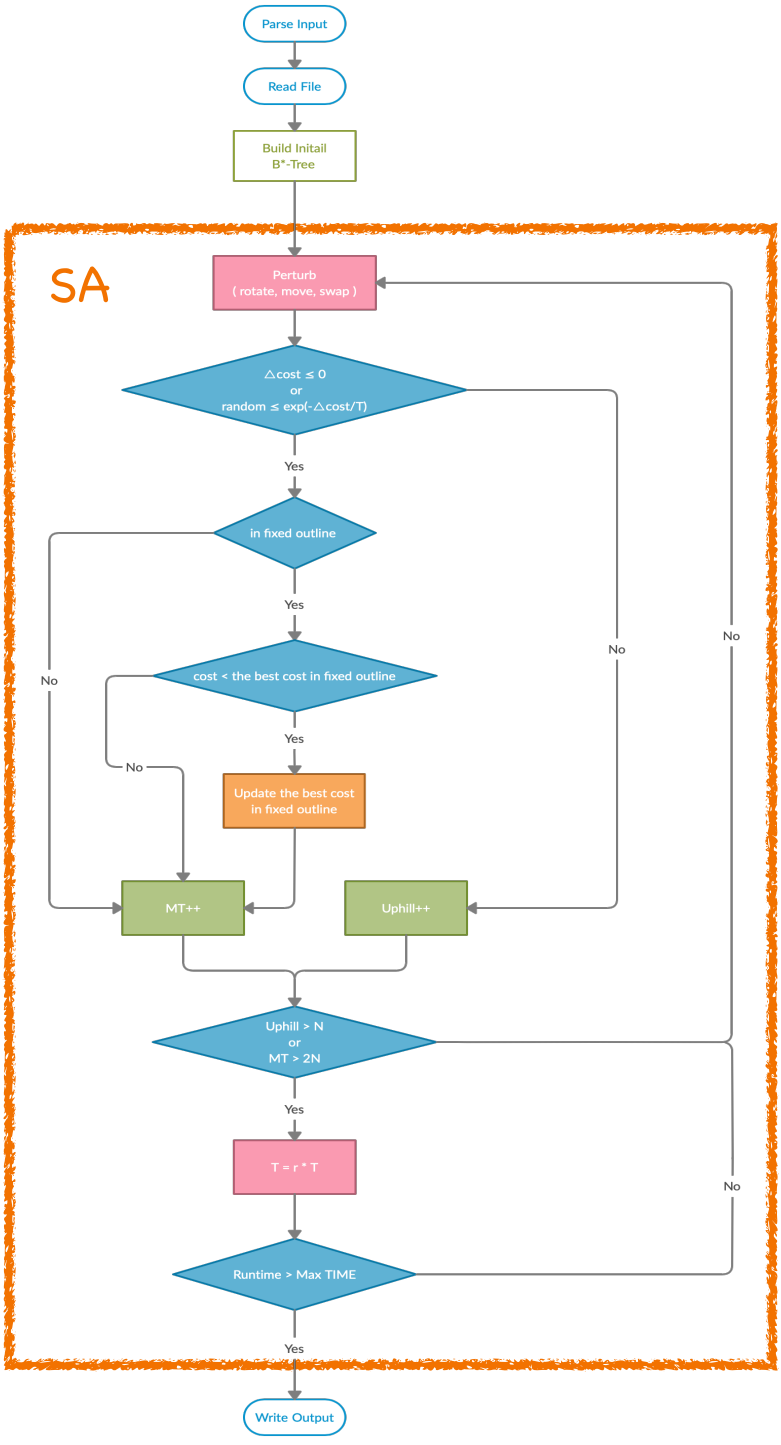
4. **The details of your algorithm. You could use flow chart(s) and/or pseudo code to help elaborate your algorithm. If your method is similar to some previous work/papers, please cite the papers and reveal your difference(s).**

我是用上課教的B*-tree data structure搭配模擬退火法去做floorplanning，因為老師在上課時有提過O-tree以及B*-tree，並且講到兩者相比下B*-tree是優於O-tree的。

一開始，我randomly的用fully binary tree去建立initial B*-tree。B*-tree的座標計算相對簡單，所以可以比較輕易地轉換成floorplan，left child的x座標＝parent的x座標＋parent的寬，而right child的x座標＝parent的x座標；y座標則用horizontal contour 去 maintain，但我並非用老師上課所說的double link list的方式去記錄contour，而是用vector去存每個x座標之中最高的y座標。

在SA中，其中重要的一個因素就是cost function，我聽取老師上課的意見，除了area, wirelength, aspect ratio之外，還多增加了寬高是否超出fixed-outline的penalty，也 deemphasize aspect ratio 重要性，將它的權重設為0.6，並且有將area跟wirelength都做normalization，其餘SA的流程大致都照Unit4 ppt p.21的演算法去實做。

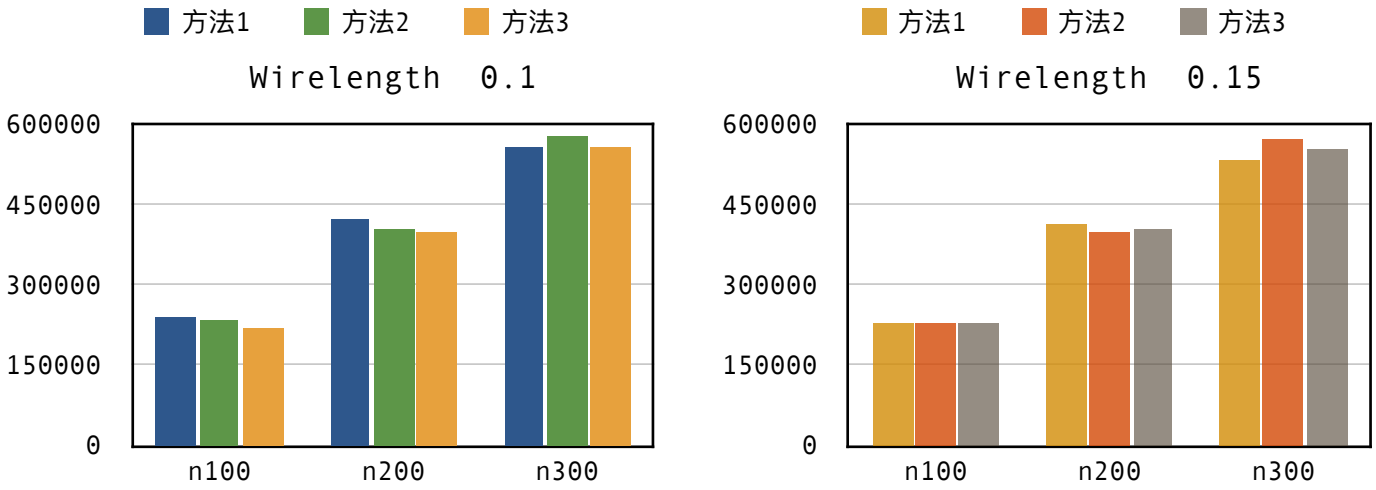
右圖為我的flow chart————>



5. **What tricks did you do to speed up your program or to enhance your solution quality? Also plot the effects of those different settings like the ones shown below.**

- 在runtime方面：我並沒有採用Parallelization或其他方式去加速我的program
- 在quality方面：我嘗試用不同的cost function去enhance solution quality
  - A. 方法1 ： cost = area + wirelength + R + width & height penalty
  - B. 方法2 ： cost = area + wirelength + width & height penalty
  - C. 方法3 ： cost = area + wirelength + 0.6*R + width & height penalty

  最後是方法3的quality最好。



6. **Please compare your results with the top 5 students' results last year for the case where the dead space ratio is set to 0.15, 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，而solution quality跟top5相比是完全排在他們之後，可能是因為我單純的就用上課教的方式去做floorplannig，並沒有想其他更好的floorplan演算法去精進解答，或是要用其他方式更進我的SA演算法，或許在有限的時間內我的solution quality跟runtime就能更好。

7. **What have you learned from this homework? What problem(s) have you encountered in this homework?**

   在此次作業中，我學會如何用B*-tree的結構去做floorplanning，並且也學到如何使用模擬退火法，但是我覺得SA此種方法的不確定因素太多，太多選擇是用random的方式去選，導致可能在不同case中會有很多不同的parameter，必須記錄一下哪些參數才能導致出最好的結果。在設定cost function上測試了好幾種方法調整我的cost，例如：給予權重或是增加不同的penalty，看哪種cost function才能有更好的答案，因為這一切沒有固定答案，只能透過不停嘗試去找，有時甚至會導致結果變得更差，所以在這上面花了我不少時間。