

# CS6135 VLSI Physical Design Automation

## Homework 3: Fixed-outline Floorplanning

107062536 李岳容

### (1) How to compile and execute your program, and give an execution example.

Usage: ./<exe> <hardblocks file> <pl file> <nets file>  
<output file> <space ratio>

e.g.:

./main ../testcases/n200.hardblocks ../testcases/n200.pl ../testcases/n200.nets ../output/n200.output 0.15

Make: make

Clean: make clean

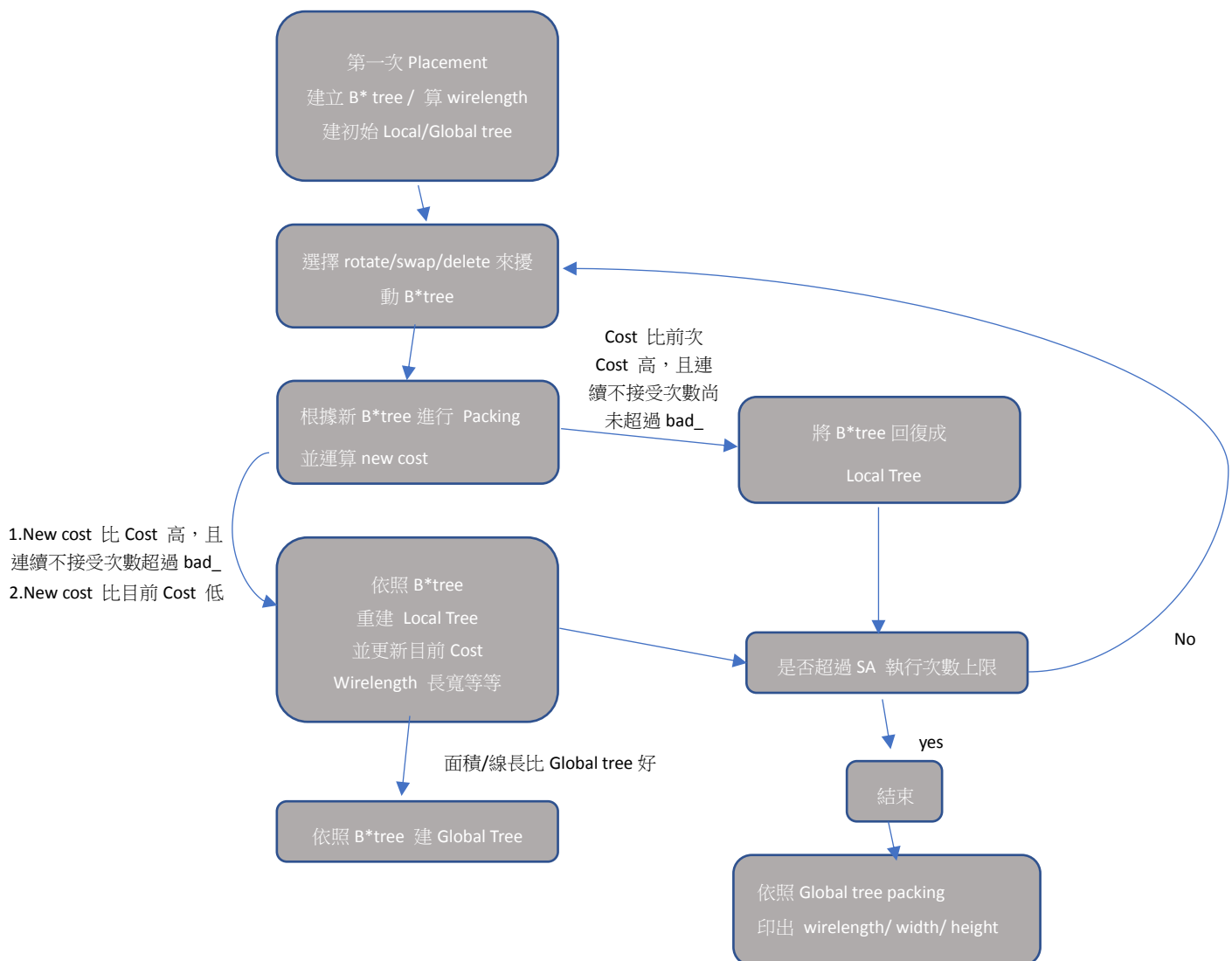
### (2) The wirelength and the runtime of each testcase in white space ratios 0.1 and 0.15, respectively.

	100 hardblocks		200 hardblocks		300 hardblocks	
Space Ratio	0.1	0.15	0.1	0.15	0.1	0.15
HPWL	221933	209584	397640	375829	588912	522036
RUNTIME	34.86	36.04	72.1	72.25	104.87	105.63

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

	<b>100 hardblocks</b>	<b>200 hardblocks</b>	<b>300 hardblocks</b>
<b>Min Space Ratio</b>	0.06	0.05	0.06

**(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 works/papers, please cite the papers and reveal your difference(s).**



這次的方法的 cost function 有參考 102062139 裡的 cost function，原先我的方法，無法同時考慮 wire length 也沒有考慮超過邊際多少，導致 0.1

的 space ratio 我都無法放進去。

**(5) The details of your implementation. What tricks did you do to speed up your program or to enhance your solution quality?**

為了加速程式，程式裡每次擾動一次 tree 都要重新 placement，而 placement 為了 packing 要確定目前擺放位置的 horizontal contour，必須要先掃過一次 block 寬度的 horizontal，確定目前最高後，再把寬度以內的 horizontal 改成最高高度加上 block 本身高度，也就是說在 packing 的過程，每擺放一排，要掃過 2 次 bounding box 寬度大小的 array。為了加速這個過程，我將在 packing 的過程每擺上一個 block 就把這個 block 右上角的 X Y 座標記錄下來，並用 double link list 紀錄。(如果有個 block 擺上去後其右上座標，蓋過其下方 block 右上角的 X 座標，那其下方 block 右上座標就會從 double linked list 裡消除。)怎麼一來，當我要確認某個 block 在 packing 時，高需不需要根據 horizontal 往上調或往下，可以藉由這個方法從掃整整 bounding box 的寬度，變成掃一排幾個 block 就掃幾個點就好。時間上可以大幅地降低。為美中不足的是，我是建 double linked list，其實在本次作業中，這個方法用單向 linked list 即可。時間可以更快。

**(6) Please compare your hardblock testcases' results with the top 5 students' results from last year and show your advantage either in runtime or in solution quality. Are your results better than theirs?**

我的 runtime 大該都是 1-2 名，然後 performance 也有前三名，所以我應該是 runtime 減低，並同時維持不錯的 performance

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

先撇除這次的作業有很多要注意指標的部分。包刮 swap 如果交換到有相連的 block 在指標上會很難寫，或是在建 B\*tree 時很容易左右子沒有處理好，會有無限迴圈。又或著我的程式為了記住目前最好的 B\*tree 結構，跟要回復到上一步驟的 B\*tree 結構，都得要另外創造一個資料型態，把指標的狀態存起來，這部分非常容易出錯。但都比不上最震撼的我，是我第一次遇到，真的完全沒轍的狀況，不管我怎麼改變我的退火法，我的設計都不夠好到可以引導我的程式慢慢走向一個非常好的結果。這次的作業是我第一次遇到，因為要配合諸多限制，所以找不到正確解的狀況。它必須靠好的觀察力，去觀察，怎麼樣的條件限制，可以既同時讓 block 的擺放大小壓縮，又可以同時掌握好長寬比例不要失衡，且要同時慢慢下降，同時

也要考慮線長。稍稍沒有設計好，或是細節沒有注意到，就會導致整個結果往不對的方向走。像我這次就是觀察不夠入微，沒有想過，同樣都是超過寬的 bounding box，也有分，有多少 blocks 超過多少的 bounding box，超過總共多少，這些都應該要納入我的 cost function。但我因為僥倖，都一直以超過 bounding box 的最多的點超過多少，想作為 cost function 的依據結果是，0.15 可以擺得進去。但 0.1 無法。這大概是我這次學到最大教訓吧。