

HW 3: Fixed-outline Floorplanning

105062600 Yi-Cheng, Chao 20:38, May 7, 2017

Homework Concepts

使用 C++ 實作完成固定區域的 floorplanning。

Compile & Execute

How to compile:

Go to src/ directory and type "make " command

```
[vlsipda14@ic29 src] make
```

How to execute:

Go to src/ directory and type the following command

```
[vlsipda14@ic29 src] ./fp -b ../testcase/n100.blocks -n ../testcase/n100.nets -p ../testcase/n100.pl  
-o n100.output -r 0.1 -s
```

Option :

-d: show the detail summary

-t : show the time report

-s: engage assignment of the best performance seed

-h: show the usage of executable binary

-c <custom seed>: assign a custom seed

```
[vlsipda14@ic29 src]$ ./fp -h  
Usage: ./fp -b <blocks_file> -n <nets_file> -p <terminal_file> -o <output_file> -r <dead_space_ratio>  
-d: Show detail message [Option]  
-h: Show the command usage [Option]  
-t: Show the time report [Option]  
-s: Engage assignment of the best performance seed [Option]  
-c <custom seed>: Assign a custom seed [Option]
```

跑測資時請助教輸入 -s 指令使用 tune 好的 seed 予以 rand() 使用。

Time Measurement

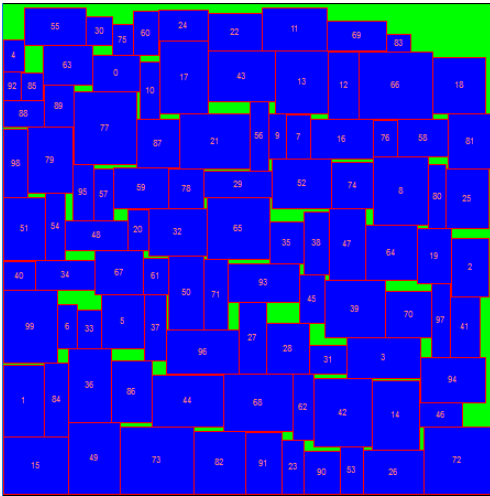
指令使用 <sys/time.h> library 的 gettimeofday 函數來取得時間，精度可達微秒。

Case	n100		n200		n300	
WSR	0.1	0.15	0.1	0.15	0.1	0.15
I/O time (sec)	0.0051	0.0022	0.0065	0.0115	0.0298	0.0071
Computing time (sec)	7.5830	9.0103	19.9113	24.1822	46.7324	31.6325
Execution time (sec)	7.5881	9.0125	19.9178	24.1937	46.7622	31.6396

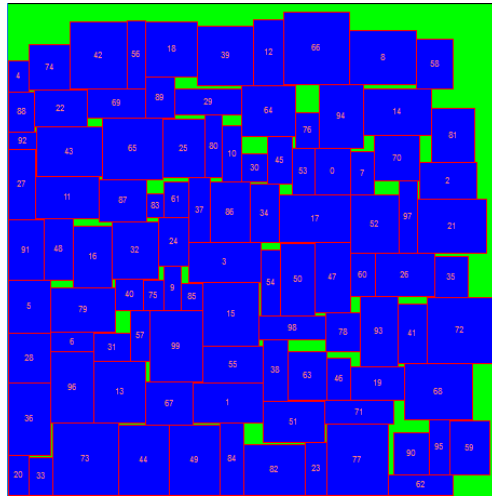
主要的 bottleneck 都在 simulated annealing 的 computation，I/O time 的讀寫檔案如上次作業二一樣，並沒有花費太多時間。

Result Display

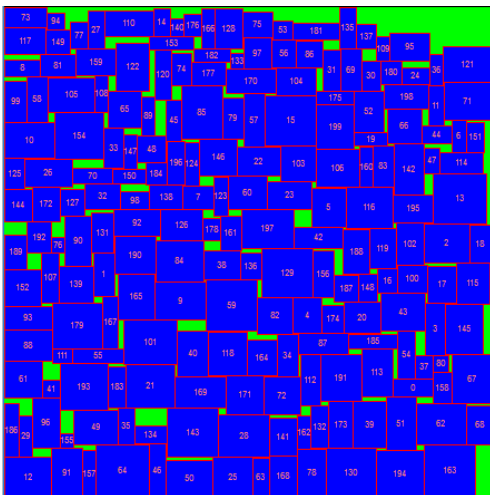
N100 white space ratio = 0.1



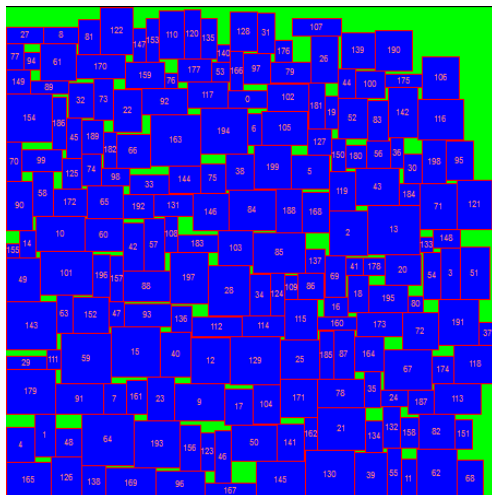
N100 white space ratio = 0.15



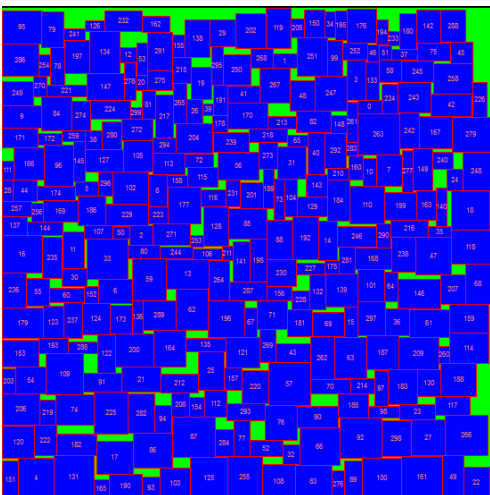
N200 white space ratio = 0.1



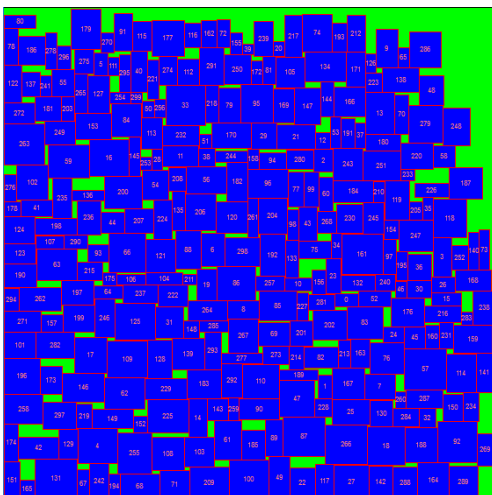
N200 white space ratio = 0.15



N300 white space ratio = 0.1



N300 white space ratio = 0.15



Experiment Analysis & Question/Answer

Q1: The wirelength and the runtime of each testcase in white space ratios 0.1 and 0.15, respectively.

Case	n100		n200		n300	
WSR	0.1	0.15	0.1	0.15	0.1	0.15
TWL	212229	220563	393496	388163	556736	544783
CPU	7.5881	9.0125	19.9178	24.1937	46.7622	31.6396
WSR: white space ratio TWL: total wirelength CPU: runtime in seconds						

Q2: Show that how small the white space ratio could be for your program to produce a legal result in 10 minutes.

助教可以使用 -c + 附上的seed編號產生同樣的結果進行驗證

n100 white space ratio: 0.35; seed: 1494169160

```
[vlsipda14@ic29 src]$ ./verifier ../testcase/n100.blocks ../testcase/n100.nets ../testcase/n100.pl n100.output 0.035
Total block area: 179501
Width/Height of the floorplan region: 431
Wirelength: 271130
Checking fixed-outline and non-overlapping constraints of the blocks locating ...
WL computed by verifier: 271130 <---> WL reported in .floorplan: 271130
OK!! Your output file satisfies our basic requirements.
```

n200 white space ratio: 0.04; seed: 1490174177

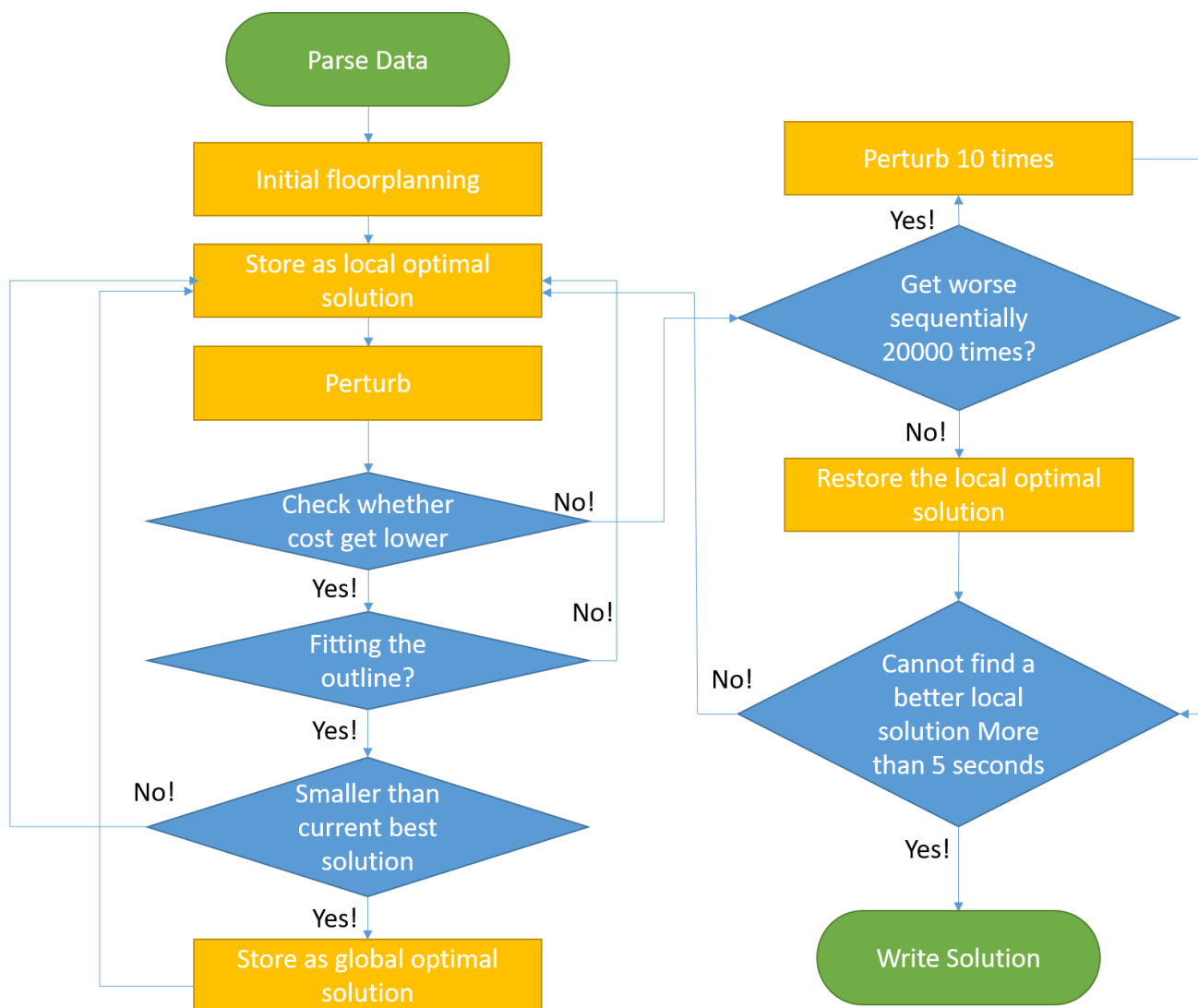
```
[vlsipda14@ic29 src]$ ./verifier ../testcase/n200.blocks ../testcase/n200.nets ../testcase/n200.pl n200.output 0.04
Total block area: 175696
Width/Height of the floorplan region: 427
Wirelength: 498510
Checking fixed-outline and non-overlapping constraints of the blocks locating ...
WL computed by verifier: 498510 <---> WL reported in .floorplan: 498510
OK!! Your output file satisfies our basic requirements.
```

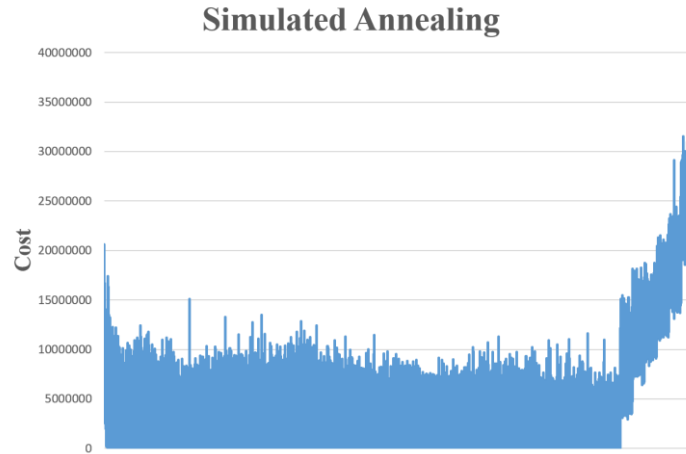
n300 white space ratio: 0.045; seed: 1494171671

```
[vlsipda14@ic29 src]$ ./verifier ../testcase/n300.blocks ../testcase/n300.nets ../testcase/n300.pl n300.output 0.045
Total block area: 273170
Width/Height of the floorplan region: 534
Wirelength: 700192
Checking fixed-outline and non-overlapping constraints of the blocks locating ...
WL computed by verifier: 700192 <---> WL reported in .floorplan: 700192
OK!! Your output file satisfies our basic requirements.
```

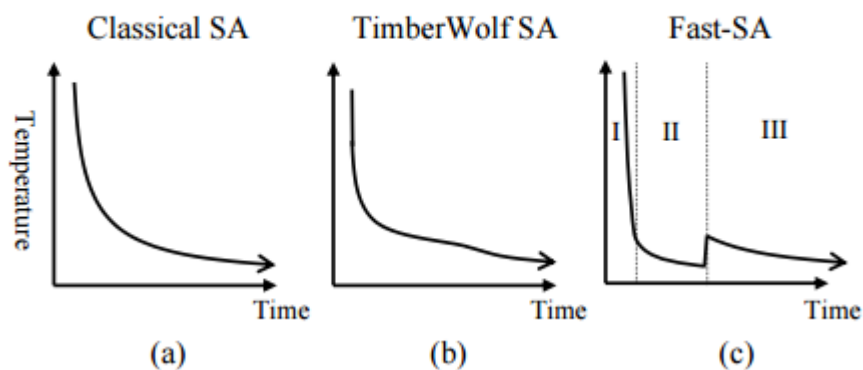
Q3: The details of your algorithm. You could use flow chart(s) and/or pseudo code to help elaborate your algorithm. If your method is similar to some previous works/papers, please cite the papers and reveal your difference(s).

整體程式流程圖如下：





上圖為SA的cost對時間作圖，整體雖然波動很大但有逐漸穩定的趨勢，直到不能壓縮後會發散，發散五秒即認為無法接受這個溫度，因此進行寫檔，與paper中的下圖幅度波動表示可能因為參數及實作方式不同而有不同結果。

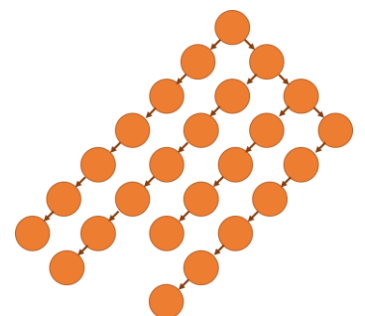


此外因為SA algorithm使用了大量的rand()來進行選擇，每次的seed或多或少都會對結果有影響，因此我設定了可以直接存取tune好的seed來使用的“-s”以及可以手動進行seed的設定的“-c”指令，如此便可以利用seed的控制來掌握每次的擺放結果。

Reference: *Modern Floorplanning Based on Fast Simulated Annealing, ISPD05, Tung-Chieh Chen*

Q4: The details of your implementation. What tricks did you do to speed up your program or to enhance your solution quality?

初始擺放的方式為先根據net盡可能地把同一條net上的cell擺放在一起，從root開始由左子樹先開始長(即往水平方向擺放)，擺到超過outline所圍成的方框後便回最上一個有空的右子點的node開始，再依序由左子樹開始擺放直到結束，大致上會形成很歪斜的B* Tree，詳細如左圖所示：



Simulated Annealing部分我終止條件設定為如果找到outline方框內的解並且五秒內無法再找到更好的解，便認為已經被壓縮到極致便跳離迴圈，因為Simulated Annealing Algorithm會需要到很多的資料存取(如packing, calculate cost, calculate wire length...等)，因此我使用陣列來進行存取資料的優化。

此外因為SA algorithm使用了大量的rand()來進行選擇，每次的seed或多或少都會對結果有影響，因此我設定了可以直接存取tune好的seed來使用的“-s”以及可以手動進行seed的設定的“-c”指令。

Q5: Please compare your results with the top 3 students' results from last year and show your advantage either in runtime or in solution quality. Are your results better than theirs?

Case	n100				n200				n300			
WSR	0.1		0.15		0.1		0.15		0.1		0.15	
Sol.	TWL	CPU	TWL	CPU	TWL	CPU	TWL	CPU	TWL	CPU	TWL	CPU
1 st	238099	8.33	226047	9.46	437794	33.20	408833	35.55	591858	58.92	553331	75.17
2 nd	209323	82.49	197203	84.88	373657	178.57	363289	167.92	560417	244.10	520531	233.22
3 rd	212518	19.79	206510	18.86	396010	141.96	382250	73.01	556285	197.87	534071	225.53
Me	212229	7.58	220563	9.01	393496	19.91	388163	24.19	556736	46.76	544783	31.63

因為當初在debug我的SA函式時以為是因為資料結構存取太慢導致一直做不出可以不超出邊界的floorplanning解，最後才發現是SA演算法有誤導致，但過程中不斷改良資料結構以及存取方式，致使做43萬次perturb以及計算好cost只要2~3秒的時間，因此最後在SA運行上幫到很大的忙，使得我的run time可以遠低於去年的Top3，然而因為我終止條件為5秒內沒有找到更好的解(即無論怎麼perturb都會走向發散)便跳出SA迴圈進行寫檔，因此可能其實還能找到更好的解，但因為時間限制便終止，因此在結果上會比不上去年的二三名，去年第二名在線長品質上非常好，實在令人敬佩。

Q6: What have you learned from this homework? What problem(s) have you encountered in this homework?

這次在資料結構的使用上因為有大量的容器存取，因此為了把原本的程式中使用的vector資料結構換成array花費了不少時間處理會遇到的segmentation fault，此外還有在嘉諄學弟與宇翔討論中學到了不少，像是在自訂義結構陣列中應該要避免直接的使用vector，否則會因為記憶體擺放方式發生一些不可預期的錯誤。

Homework Review

這次速度上令我非常滿意，但實在無法兼顧 floorplanning 品質與速度，希望下次能有更好的結果，這次前前後後寫了一個禮拜，農出程式碼到最後能打報告實在令人高興，此外自己人生中最長程式碼紀錄也刷新了，希望

自己能更精煉自己的程式碼，因為這不是值得光榮的事。

此外特別感謝羅嘉諄學弟在我撰寫程式上給了我很多幫助與意見，處理了數多天的 SA 演算法有了學弟的在旁給予意見才得以看到完成的曙光，整份程式可以說沒有學弟的幫忙我是不可能完成，如果可以的話，請把我的分數兩分給他吧 XD。

