

2024 PDA Lab 3 Report  
Optimizer and Legalizer Co-optimization

313591010 丁慧慈

1. 實作想法和 Pseudocode:

這次的作業花了比較多時間和同學討論想架構，最後推動block的想法未成功實作，只做了插空的兩個function。每次都挑較好的結果更新，以達到較好的 performance。

我的簡易流程圖和Pseudocode如下所示，由於未推動任何block，因此對於插空演算法，我實作了水平找空和全局找空，利用開多線程加快執行時間(上次lab2看到實驗室同學有使用，因此這次就來玩玩看!)，並根據回傳之結果選擇cost較低者更新。

<pre>MAIN: Parse input files Initialize PlacementRow and all_blocks FOR each operation:     Parse REMOVE and ADD commands     IF overlap:         Call window_search() or greedy_search()         Choose best result from W/G     Update PlacementRow and write output  FUNCTION window_search(temp block):     Calculate possible positions in search range     FOR each candidate position:         IF position is legal:             Place block at position             RETURN true         RETURN false  FUNCTION greedy_search(greedy block, range):     Initialize search range     FOR each position in range:         IF position is legal:             Place block at position             RETURN true     RETURN false</pre>	<pre>graph TD; ParserInfo[/Parser info/] --&gt; ParserOpt[/Parser opt/]; ParserOpt --&gt; RemoveBankingFF[Remove banking FF]; RemoveBankingFF --&gt; WindowSearch[Window search]; RemoveBankingFF --&gt; GreedySearch[Greedy search]; WindowSearch --&gt; ChooseResult[Choose better result and update placementrow and blocks]; GreedySearch --&gt; ChooseResult; ChooseResult --&gt; InsertMBFF[Insert MBFF]; InsertMBFF --&gt; OutputData[output data]</pre>
Fig 1. Pseudocode	Fig 2. Flow chart

## 2. Time complexity analysis:

關於這個演算法的時間複雜度，根據總共的 block 個數 B, Placementrow 數量 R, 平均每行的 block 數 n 和 window\_search 中可能的位置集合大小 k。

$$O(B \cdot m \cdot (k \cdot (h \cdot \log n)))$$

## 3. Special features of your program:

我認為在我的 code 中比較特殊的作法是我會將找到的擺置解利用 set 和目標位置的距離作排序，在從最近的空位開始檢查是否為合法解，因此較能找到離目標點較近的解！

關於推動 block 我的想法如下，之後會繼續嘗試，希望能成功推動！

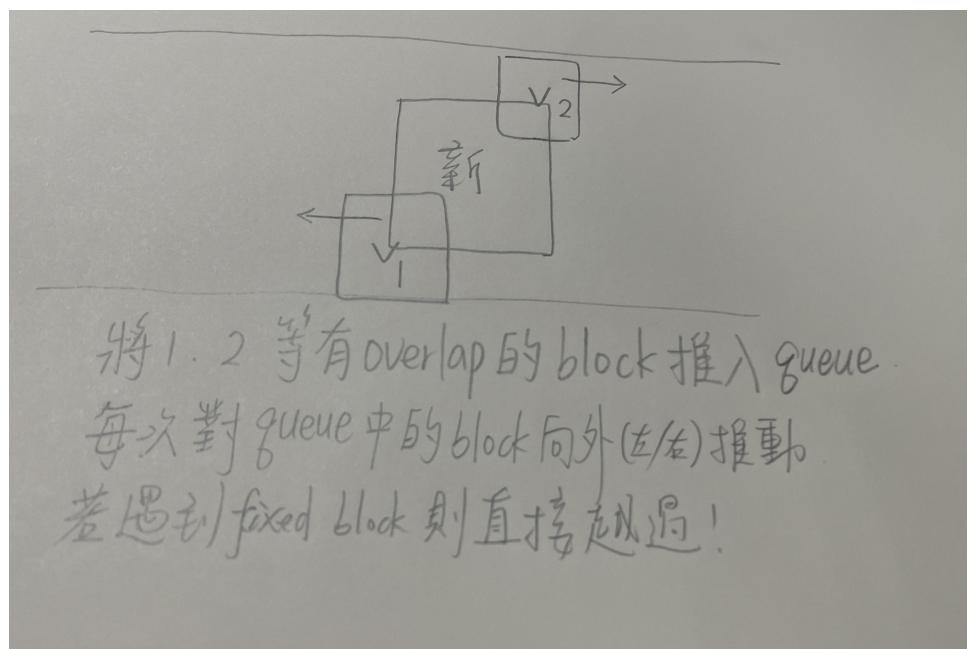


Fig 3. Move idea

## 4. Feedback:

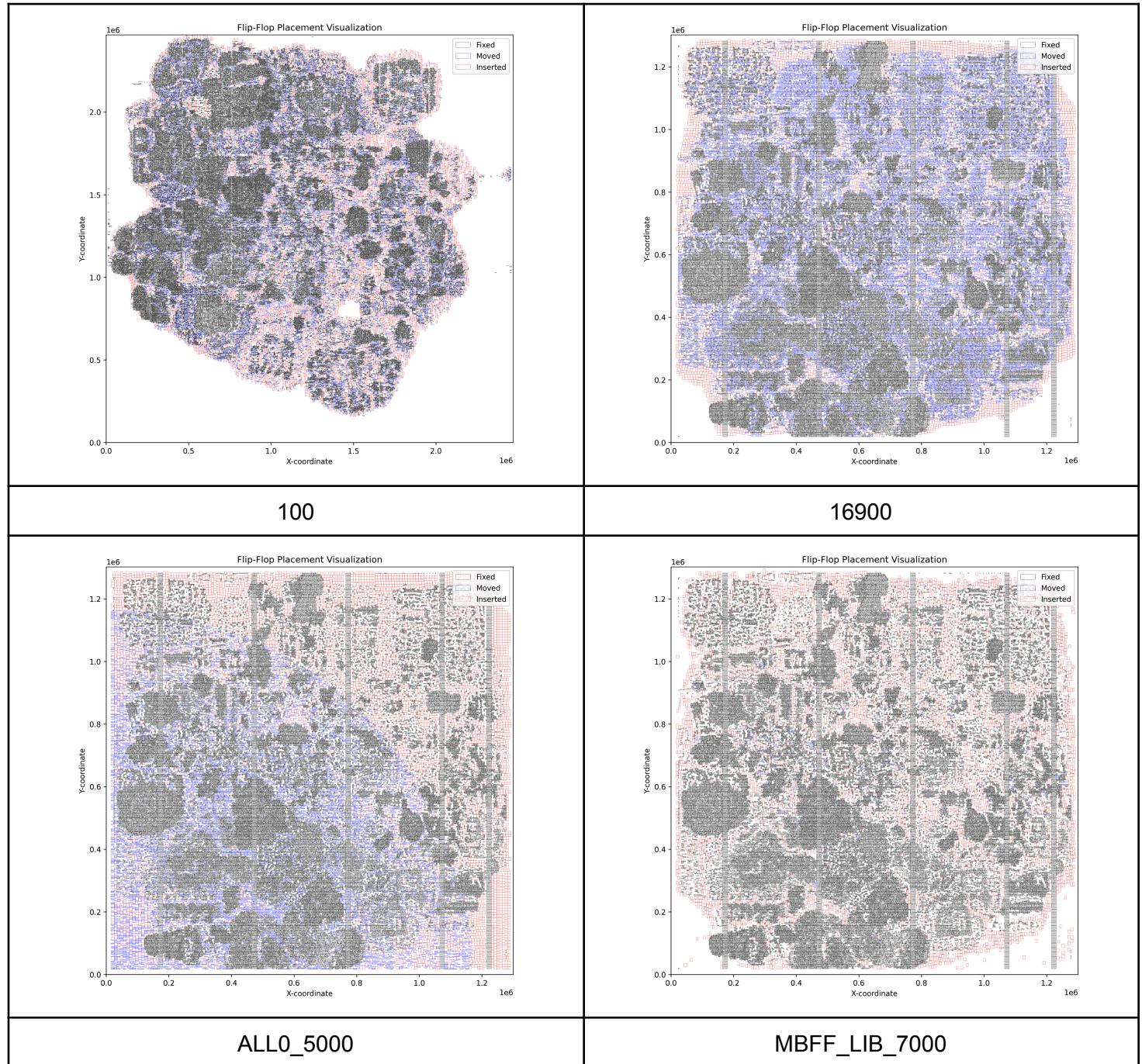
這次的作業挺有挑戰性的，但若不追求 performance 單純找合法解可以很快實現。自己親自刻完這次的作業，對於 Legalizer 有很深刻的了解，喜歡和同學憶起在實驗室想演算法的過程，也從中學到了多進程和寫程式的特殊技巧。

## 5. Conclusion:

這次考量到要在資料數量級很大的狀況下找解，原本測資都可以在兩分鐘跑完，但為了拚一下 performance，有改用全局的方式尋找，使得時間有點吃緊，聽到同學們的時間都跑得特別快，感覺程式還有許多可以再改進的地方，也有嘗試助教們提到的可以玩 gif，可惜畫得有點久跑不動，但最終已知的四個公開測資皆有通過，非常開心！

## Appendix A

### Visualizer



## Appendix B

### Final cost

	100	16900	5000	7000
PASS	V	V	V	V
Cost	22,1788,5000	2,6611,1670	411,4640,7000	2649,6271,8000

```
[2024PDA022@mseda01 ~/Lab3]$ ./Evaluator testcase/testcase2_100.lg testcase/
+-----+-----+-----+-----+
| Cost | - | Weight | Value | Percentage(%) |
+-----+-----+-----+-----+
| Move Times | 0.00 | 50000.00 | 0.00 | 0.00(%) |
| Total Distance | 110894250.00 | 20.00 | 2217885000.00 | 100.00(%) |
| Total | - | - | 2217885000.00 | 100.00(%) |
+-----+-----+-----+-----+
[2024PDA022@mseda01 ~/Lab3]$
[2024PDA022@mseda01 ~/Lab3]$ ./Evaluator testcase/testcase1_16900.lg testcase/
+-----+-----+-----+-----+
| Cost | - | Weight | Value | Percentage(%) |
+-----+-----+-----+-----+
| Move Times | 0.00 | 10000.00 | 0.00 | 0.00(%) |
| Total Distance | 266111670.00 | 1.00 | 266111670.00 | 100.00(%) |
| Total | - | - | 266111670.00 | 100.00(%) |
+-----+-----+-----+-----+
[2024PDA022@mseda01 ~/Lab3]$
[2024PDA022@mseda01 ~/Lab3]$ ./Evaluator testcase/testcase1_ALL0_5000.lg testcase/
+-----+-----+-----+-----+
| Cost | - | Weight | Value | Percentage(%) |
+-----+-----+-----+-----+
| Move Times | 0.00 | 50000.00 | 0.00 | 0.00(%) |
| Total Distance | 2057320350.00 | 20.00 | 41146407000.00 | 100.00(%) |
| Total | - | - | 41146407000.00 | 100.00(%) |
+-----+-----+-----+-----+
[2024PDA022@mseda01 ~/Lab3]$
[2024PDA022@mseda01 ~/Lab3]$ ./Evaluator testcase/testcase1_MBFF_LIB_7000.lg testcase/
+-----+-----+-----+-----+
| Cost | - | Weight | Value | Percentage(%) |
+-----+-----+-----+-----+
| Move Times | 0.00 | 100.00 | 0.00 | 0.00(%) |
| Total Distance | 1324813590.00 | 200.00 | 264962718000.00 | 100.00(%) |
| Total | - | - | 264962718000.00 | 100.00(%) |
+-----+-----+-----+-----+
```