

一、排程方法介紹

我們怎麼解決 JSSP?

JSSP 為 NP 問題，適合以元啟發式演算法以較高效的方式近似找出較優解，而我們從這學期修習的基因演算法與管理科學中獲得靈感，決定以基因演算法（Genetic Algorithm）來解決這次的問題。

GA 概念介紹

基因演算法是一種優化解的演算法，通過模擬遺傳、進化的過程，在一組隨機解中找到最佳解。並以基因編碼表示解，通過選擇、交叉和變異等操作來進行搜索和改進。

本組特色 1: 超參數優化

我們使用貝葉斯（Bayesian Optimization）優化我們在 GA 中的參數，優化的超參數有交配率、突變率、以及群體個數。

Algorithm 1 Sequential Model-Based Optimization

```

Input:  $f, \mathcal{X}, S, \mathcal{M}$ 
 $\mathcal{D} \leftarrow \text{INITSAMPLES}(f, \mathcal{X})$ 
for  $i \leftarrow |\mathcal{D}|$  to  $T$  do
   $p(y | \mathbf{x}, \mathcal{D}) \leftarrow \text{FITMODEL}(\mathcal{M}, \mathcal{D})$ 
   $\mathbf{x}_i \leftarrow \arg \max_{\mathbf{x} \in \mathcal{X}} S(\mathbf{x}, p(y | \mathbf{x}, \mathcal{D}))$ 
   $y_i \leftarrow f(\mathbf{x}_i)$   $\triangleright$  Expensive step
   $\mathcal{D} \leftarrow \mathcal{D} \cup (\mathbf{x}_i, y_i)$ 
end for
    
```

本組特色 2: 自動化

我們將整個 GA 的結果透過自動化的方式銜接到 Flexsim 中來展現我們的成果，並模擬 JSSP 問題，也就是將問題自動讀檔、將貝式優化演算法的最優解輸出成 Excel，匯入至 Flexsim 的 Source、Route、Arrival Order 等表來模擬問題。若是 Flexsim 有 API，我們便能一鍵解決 JSSP 問題了。

備註：預設檔案為 rcmax_20_15_8，如果要使用預設以外的題目，請在 flexsim 的匯入 excel 時選擇正確的 solution 檔名。目前我們只實做 20_15 機台作為交叉驗證。

Flexsim:

Global Table 名稱	功用
Time	紀錄每個 Job 最後完成時間。
Route	紀錄每個 Job 的機台加工先後順序。
Rank	紀錄每個 Job 在不同 Operation 的優先順序數值。
time	紀錄各個 Job 進入各個機台的時間和離開的時間。
EachmachinePtime	紀錄每個 Operation 在機台中運作的時間，方便統整分析結果。
State	紀錄每個機台各自已經做完幾個 Job。
Sequence	紀錄每個機台各自加工 Job 的順序。

流程概念：

1. 建立基本模型，拉出符合模擬的情境模型。
2. 將已知的 Rank, Route 以及 Sequence 的相關資料自動化匯入 Flexsim 對應的表格中。
3. 開始模擬，機台初始設定為將 input 全數關閉，避免先進入 Buffer，但還沒輪到順序的 Job 進入機台（因為派工法則不是 FIFO，是按照演算法得到的 Rank 進行排序）。
4. Flow control 更新該輪的 operation label，進而使 Job 能在此領到該輪相對應的 Rank。
5. Job 進入該輪 Buffer，觸發排序系統，在該 Buffer 中的 Job 會利用 Rank 排序。
6. 當 Rank 順位最前面的 Job 編號和 Sequence 內所示之該機台當時輪到的 Job 編號吻合時，Buffer 傳訊息給機台，要求其打開入口讓 Job 進入運作，若當時沒有吻合的 Job 出現，則位於 Buffer 中的所有 Job 必須全數排隊等待，不能因為先到就先進入。
7. 以上少考慮一種情況：當該輪順序 2 的 Job 進入 Buffer 時，順序 1 的 Job 正在排隊等待。直到順序 1 的 Job 離開機台時，都沒有新進入 Buffer 的 Job 觸發檢查機制，那麼順序 2 的 Job 會在 Buffer 中乾等。為了解決此問題，我們在機台的 OnExit 中也設置了檢查功能，檢查完工時有無符合順位的 Job 在 Buffer 等待，避免因沒檢查到而使 Makespan time 莫名增加。
8. 當 Job 進入機台時，機台入口會關上，直到下一順位的 Job 進入 Buffer 時，入口才會打開。
9. 每個 Job 在進入機台、離開機台的時間，會分別被記錄到 global table time 中。
10. Job 回 Flow Control 時，會領新的 Rank 並重複動作直到結束，得到最終 Makespan time。

二、排程結果與討論

結果如下：

Name	Size	LB	UB	Solution	Solution-UB
Dmu01_rcmax_20_15_8	20x15	2601	2669	2944	275
Dmu01_rcmax_20_15_5	20x15	2731	2731	2972	241
Dmu01_rcmax_20_15_4	20x15	2501	2563	2966	403
Dmu01_rcmax_20_15_1	20x15	2749	2749	3178	429
Dmu44_cscmax_20_15_7	20x15	3283	3475	3917	442
Dmu44_cscmax_20_15_6	20x20	3575	4035	4517	482

註解&相關資料：

1. 超參數優化：<https://zhuanlan.zhihu.com/p/53826787>
2. GA 中設定的 iteration 要跑完才能有 sequence。若尚未跑出 best solution 就想先停止演算法，做 flexsim 測試，可利用已求得現有 solution 以及 route，透過 'ADD_SEQUENCE_TO_RANK' 求得 sequence 做 flexsim 模擬。
3. Python 部分：code 中有註解介紹函式，請確保有安裝 numpy, matplotlib, pandas, bayes_opt 以及 openpyxl。若是缺少任一函式庫可至 cmd 或是 bash 中按照下列方式安裝：`pip install numpy matplotlib pandas bayesian-optimization`