強化學習於多站點排程系統之應用

-以 TFTLCD 為例

Applying Q-learning to Multi-Site Scheduling System - A Case Study on TFTLCD

指導教授：王宏鍇
專題成員：何宇婕
開發工具：Python
測試環境：windows11

一、簡介：

由於此為一個多站點多機台排程最佳化問題，若以傳統排程方法無法有效率且系統性找出有效降低總完工時間的最佳解，故此專題設計採用強化學習中的 Double Q-learning 方法以逐步收斂出最佳解。

每一個工作單位(job)須經五個站點(stage)，每個站點限制條件皆不完全相同，因此每一站點各自使用自己的 Q table。Q table 中的 state 設定為按照到達時間(arrival time)排序的所有工作單位，action 設定為可選擇之機台，而 reward 則設定為每個工作單位於當前站點完成時間之倒數。

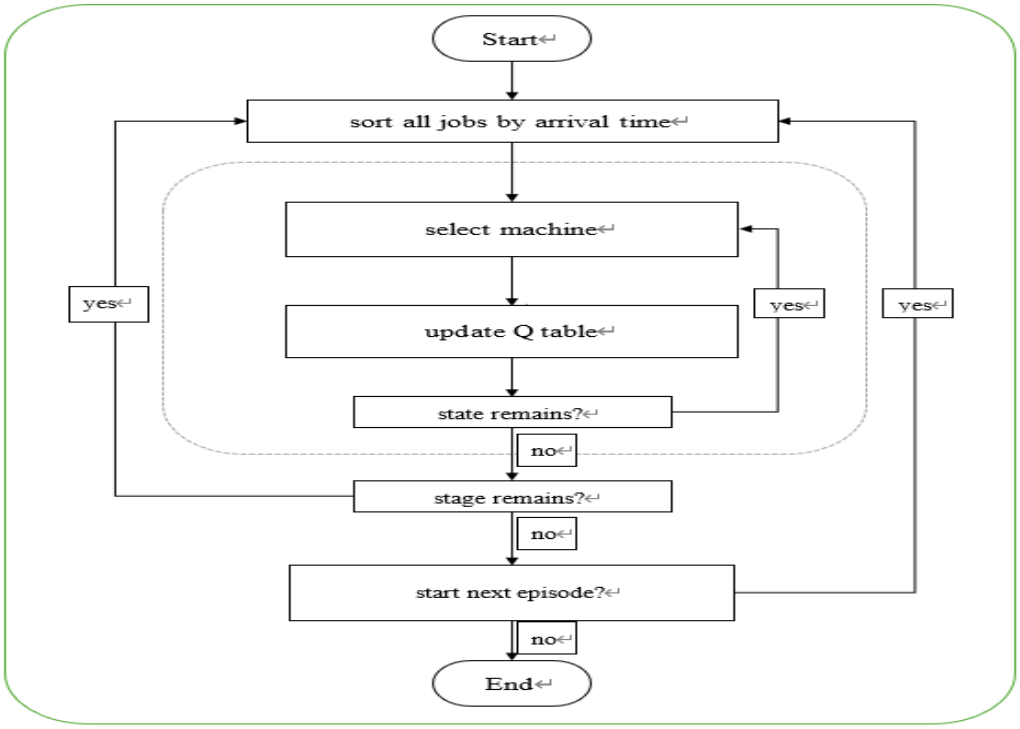
程式執行時每次依到達當前站點的 arrival time 排序所有工作單位，接著依序根據 Q table 中 Q value 最大者選擇機台。選擇機台時加入 epsilon decay 機制以增加初期探索可能。另外，由於使用單一 Q table 做訓練易傾向過度高估 Q value，從而導致可能造成總是選擇到 Q value 被高估的 action 來更新 Q table。為避免此問題故設計採用兩個 Q table 相互訓練，即每次選擇 action 之 Q table 跟計算更新 Q value 之 Q table 並非同一個，如此便能達到使兩個 Q table 相互制衡。

此方法中可將每個站點視為一個環境，而上述更新每站點 Q table 即為與環境互動的過程。每次與環境互動後我們獲得一新的狀態，也就是當前每個機台的使用情況以及每個工作單位到下一站點之 arrival time，並繼而使用此狀態與新的環境(stage)互動。當與五個站點皆互動完成即稱之為一 episode，其後使用此方法迭代多次以收斂出最佳解。

以下為用於更新 Q Table 之公式：

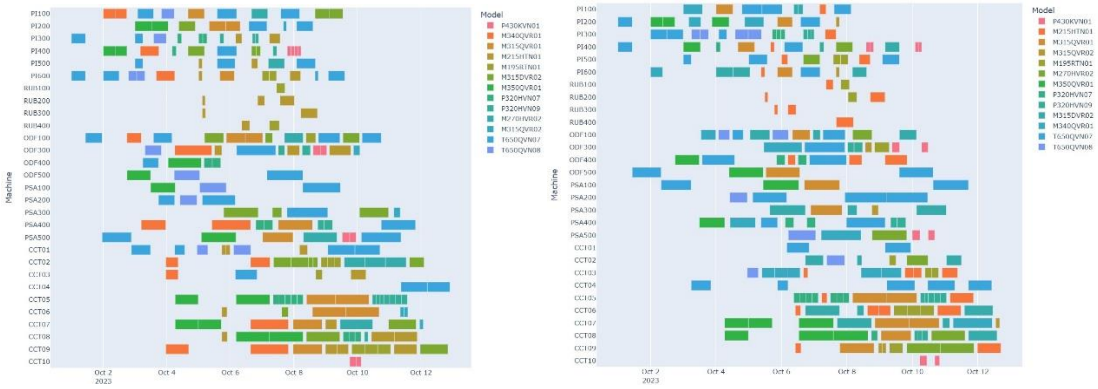
$$\text{Define } a^* = \arg\max_a Q^A(s', a)$$
$$Q^A(s,a) \leftarrow Q^A(s,a) + \alpha(s,a)\left(r + \gamma Q^B(s', a^*) - Q^A(s,a)\right)$$

以下為流程圖：



圖一：flow chart

二、測試結果：



圖二：排程策略甘特圖輸出結果

圖二以甘特圖呈現兩組分別各執行6000 episode 後獲得之最佳排程策略。最佳完工時間(make span)分別由 414小時及 350小時收斂至 285小時及280小時。系統亦同時記錄每個站點之最佳 setup time。

```
MakeSpan: 280.9570175438596
PI setupTime: 39
RUB setupTime: 4
ODF setupTime: 23
PSA setupTime: 16
CCT setupTime: 36
Episode 0, best makespan = 350.239, current makespan = 350.239, current epsilon = 0.9
...
Episode 5999, best makespan = 280.957, current makespan = 335.685, current epsilon = 0.366
```