模擬學期末專題 Group20

1. 動機：

我們選擇了Dmu01(rcmax)及Dmu44(cscmax)來做這次的題目，首先先使用最簡單的FIFO及LIFO，再使用上課曾教過的RemainTime來進行模擬，發現結果都與UB有一段不小的差距，因此我們認為要先試著以數學模型的方式來解Job Shop問題，我們找到了OR-tools這個套件來解決這樣的問題，並且很好入手。在認識到Job Shop的MIP模型後，並透過OR-tools求解，我們才發現原來NP hard的題目是可能我們有生之年都跑不出最佳解的，所以才開始著手研究Genetic Algorithm，試著以基因編碼的方式，進行選擇、交配、突變來接近最佳解。

1. 排程方法介紹：
2. FIFO (First In First Out):

先進先出法，依照Job進入Buffer的順序，越早進入的優先進入Workstation。

1. LIFO (Last In First Out):

後進先出法，依照Job進入Buffer的順序，越晚進入的優先進入Workstation。

1. RemainTime:

計算每個Job剩餘的加工時間，同一個Buffer內依照剩餘加工時間進行降冪排序，越大RemainTime的Job優先進入Workstation。

1. OR-Tools:

OR-Tools是Google的開源求解器，能通過設定好變數、限制式及目標式的方式來求解，效率高，將Job Shop的MIP model轉換成code後，丟給CPsolver進行求解，但會因為運行時間很長求不出答案，所以可以設定wall time來看限制時間內的最佳解。（細節與介紹寫在code檔案OR\_Tools.ipynb內）

1. Genetic Algorithm (GA,基因演算法):

將Job Shop的解轉換成基因編碼的方式，隨機生成多組解，進行選擇交配對象、交配生成子代、子代突變，經過很多世代後能得出收斂的最優解。（細節與介紹寫在code檔案GA.ipynb內）

1. 排程結果與討論：
2. Dmu01\_rcmax\_20\_15\_4:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 方法 | FIFO | LIFO | RemainTime | GA | OR-Tools | UB |
| Makespan | **3511** | **3367** | **3421** | **3173** | **2587** | **2563** |

首先使用最簡單的FIFO及LIFO，發現都與UB有不小的差距，RemainTime的方法也沒有改善很多，接著使用Or-Tools做出來的解與UB僅有不到30的差距，而GA跑出來的解和UB相差600多一點，效果也還算不錯。

1. Dmu44\_cscmax\_20\_15\_4:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 方法 | FIFO | LIFO | RemainTime | GA | Or-Tools | UB |
| Makespan | **4914** | **4638** | **4545** | **4503** | **3564** | **3475** |

在這個cscmax的題目中，RemainTime及LIFO相較FIFO減少不少，使用Or-Tools求出來的解依舊相當接近UB，而使用GA跑出來的Makespan和UB落差較大，僅比使用RemainTime好一點而已，或許在參數的設定上可以有所改進。

1. 參考資料：

1. https://developers.google.com/optimization/scheduling/

2. https://github.com/wurmen/Genetic-Algorithm-for-Job-Shop-Scheduling-and-NSGA-II/blob/master/implementation%20with%20python/GA-jobshop/GA\_For\_Jobshop.md

3. 基因演算法之課程內容 by 林春成 教授