

利用兩點交配之 Inver-over 演算法解決單機排程與動態抵達時間的問題

Use Inver-over algorithms of two-point intersection to solve the problems of single machine and arrival time

陳世興^a 王佳文^b 李易霖^c

^a正修科技大學 資訊管理學系

^b南華大學 電子商務管理學系

^c南華大學 資訊管理學系

^aworldstar.chen@gmail.com

摘要

單機排程(Single machine)指的是，所有工件的加工均透過此機器來做加工處理。余宣慧[3]於該研究，在解決單機排程的相關問題時，都是以靜態來假設。例如 n 個工件在開工時就已經抵達了，因此並不會有動態抵達時間(Arrival time)的問題。但在實際的狀況卻存在著工件是動態抵達的情形。因此本研究將會以在解決路徑規劃及順序擾動方面，效果相當的 Inver-Over 演算法為新方法 Inver-over CX 演算法的基礎。且為比較對象，並結合兩點交配的 PMX (Partial Message Crossover) 改善 Inver-over 演算法於大問題陷入局部最佳解的缺失[2, 5]。最後再利用基因演算法之特性，讓工件比喻為染色體，並去交配、突變進而產生更好的順序解。實驗設計的部份，實驗本研究將會以 Rabadi et al.[7]對於單機排程所提出經調整後處理時間矩陣(Adjusted processing, AP)的不同均值範圍去分成 low、med、high 等問題類別去做測試，得以了解本研究所提出的方法是否較佳。

關鍵詞：單機排程、Inver-Over、基因演算法、共同交期、整備時間

Abstract

Single machine means that every process of the work piece is done by this machine. As for this research, 余宣慧[3] uses static state to suppose when

it comes to the problem of single machine. For example, unknown numbers of work pieces arrive before it starts, so it has nothing to do with the problem of arrival time. However, in reality, the problem of arrival time exists. For example, unknown numbers of work pieces arrive successively when it starts, so we suppose work pieces arrive successively, which could make the result much closer to reality when it comes to single machine. To solve the problem of single machine and arrival time, the order of work pieces matters and is much closer to reality. Therefore, this research will take Inver-Over as the basis of Inver-over CX to solve the problem of route planning and order disturbance and as compare object. Combining two-point intersection PMX (Partial Message Crossover) improves Inver-over algorithms. Lastly, usssork pieces to chromosome, and then have them intercourse, mutate and proceed better order resolution. As for research planning, we will test adjusted processing of low, med and high ranges proposed by Rabadi et al. [Rabadi et al.(2004)Rabadi, Mollaghasemi, and Anagnostopoulos], Chen et al.[Chen et al.(2014)Chen, Chen, and Liou]. Finally, the result shows that the new method, Inver-over CX is better than Inver-over.

1. 緒論

隨著科技時代的進步，不管在傳統製造業或高科技產業，只要有生產線的存在，就會存在生產排程的問題，且生產排程在各個產業中扮演著重要的角色，如從一開始接單，及針對不同產品的不同交期、共同交期、處理時間與整備時間，甚至於操作人員的熟練度等，都是影響生產排程的因素之一。

單機排程是所有排程問題中較單純的一種，其涵義是指所有工件的加工均透過此機器來做加工處理，例如修車場(只有一位技工)、一部電腦工作站共用等等。余宣慧[1]提到過在解決單機排程的相關問題都是以靜態來假設，例如 n 個工件在開工時就已經抵達了，因此並不會有動態抵達時間的問題，但在實際的狀況卻存在著工件是動態抵達的情形，例如 n 個工件在開工時才陸續抵達，所以假設工件為動態抵達的情形，這樣在解決單機排程的問題時，會更貼近實際的狀況。

在解決單機排程的問題時大都是以靜態假設，所以在實際真實的狀況下，以靜態假設時會顯得無法證實此假設的有效性，且在單機排程的問題中每個工件都會有各自的抵達時間、共同交期(Common due date) [4]及整備時間[3]，因此本研究希望能針對單機排程問題，考慮動態抵達時間、共同交期及整備時間等等問題，且目標使總延後時間最小，這樣就可以假設遇到突發狀況的情形也更貼切實際真實的狀況。

Rabadi et al.[7]是第一個提出共同交期與整備時間之順序相依性之問題，且在排程上的順序相依之成本問題，儘管相似Kanet [10]，Rabadi et al. [7]所提出共同交期在排程上的順序相依之成本問題是屬於NP-hard的問題，而不是多項式時間代替順序相依整備時間。

Chang et al.[12] 於該研究提出績優型基因演算法於單機排程與整備成本的問題，該研究為Rabadi et al. [7]之延伸，Chang et al.[12]於該研究中提到，單機排程問題中的 n 個工件需要經過機器的

整備時間，及工件轉換至下一個工件所經過的時間，而每個工件之間會有共同的交期，故該研究找出工件順序的中間點，及將研究目的設定為找出最佳的工件順序以減少單機排程早到及晚到的成本，在找尋工件順序方面是以局部搜尋法的績優型，為搜尋工件順序的方式，局部搜尋法的空間複雜度為 $O(n^2)$ ，最後經由目標函數分析結果顯示，此研究是不錯的。

Chen et al.[13]於該研究解決單機排程及順序相依整備時間之問題以人工染色體基因演算法2(ACGA2)為解決方法，此研究為Chang et al.[43]延伸的研究，在Chen et al.[47]的研究中提出新的ACGA，命名為ACGA2，因為原來ACGA無法在一些排程問題上有良好的表現，而ACGA2與原先的ACGA的差異是在於，該研究提出一種雙變量的機率模型加入到原先的ACGA進而產生新方法ACGA2，ACGA2的用途在於解決順序相依整備時間與共同交期之問題，最後結果表明，新方法ACGA2的平均錯誤率是ACGA一半，此外，當ACGA2是與其他啟發式方法和VNS結合應用時，是目前最好的解。

在解決單機排程與動態抵達時間的問題上，工件的順序對於整個單機排程的績效會是重要的核心。在過去之研究已利用在解決路徑規劃及順序擾動效果相當的Inver-over演算法，來去解決旅行推銷員問題TSP(Traveling Salesperson Problem) [6]，而TSP涵義是由一個旅行家要出發至多個旅遊景點，每個景點均要拜訪一次，最後回到原點，而要怎麼出發及安排才會最省時最有效率，這即是旅行銷售員問題也是路徑規劃的問題。所以TSP的問題與單機排成的問題本質是一樣的，因此該研究[6]利用Inver-over演算法去搜尋最佳路徑，此方式也是可以將其利用於解決單機排程與動態抵達時間的問題上的。

本研究在此會提出新的演算法，目的是為了解決動態抵達時間的單機排程問題，其引用基礎為Inver-over演算法，因為在解決動態規劃的單機排程問題時必定會考慮到工件順序上的問題，而解決順序上的問題即為排程相關問題，故本研究以

Inver-over 演算法為基礎，但 Inver-over 演算法在解決順序問題時收斂速度確實比傳統演算法來的快，不過卻有遇到大問題收斂速度變慢的缺點[2, 5]，因此本研究會加入 PMX 兩點交配[14]的方法去減少遇到大問題的缺點，目的在於增加擾動工件順序，所以本研究會以 Inver-over 演算法為基礎，並加入 PMX 為新方法，其命名為 Inver-over CX，並以此新方法來解決動態抵達時間的單機排程問題。

2. 研究方法

單機排程的問題，在過去文獻裡已經有人提出相關的演算法來解決，本研究會以 Inver-over 演算法為基礎，並加入 PMX 及基因演算法為新方法，其命名為 Inver-over CX，並以此新方法來解決動態抵達時間的單機排程問題，故在方法的研究上是項新的議題，在 2.1 乃動態抵達時間的單機排程問題，2.2 簡介 Inver-over 演算法之流程，2.3 為本研究之方法 Inver-over CX。

2.1 動態抵達時間的單機排程問題

本研究在解決動態抵達時間的單機排程問題時，會去考慮工件的順序，因此本研究採取在順序擾動及路徑規劃相當出色的 Inver-over 演算法為基礎，並加入 PMX 去改善 Inver-over 演算法遇到大問題陷入局部最佳解的缺陷為新方法，其命名為 Inver-over CX，以此新方法去解決本研究要解決的問題，而工件順序產生目標解時，會依基因演算法的精英政策，將總延遲時間較好的前 10 組遺傳至下一代，剩下的 90 組去兩兩進而得到抵達時間 r_i 。

亂數的範圍值 Rmax 計算公式如公式(1)，P 為處理時間，S 為各個工件的整備時間，N 為各個工件的數量。

$$Rmax = (P + S) \times (N - 1) \quad (1)$$

在計算以考慮動態時間的共同交期 K 必須先找出考慮動態抵達時間的工件順序中間點 b (Middle Position)，而計算中間點 b 的公式會依照工

件數量的奇數及偶數去區分，所以計算中間點 b 的公式如公式(2)，b 為中間點，N 為工件的數量。

$$b = \begin{cases} N \div 2 & (\text{if } n \text{ is even}) \\ (N + 1) \div 2 & (\text{if } n \text{ is odd}) \end{cases} \quad (2)$$

在產生各個工件的抵達時間 r_i 後，可以依照各個工件的處理時間 P 及抵達時間 r_i 去計算出各個工件的完工時間 C (Completion Time)，而本研究是要以考慮動態抵達時間為目的，故下列公式是先判斷抵達時間 r_i 是否大於前一個的完工時間 C，之後再去相加處理時間 P 為該工件的完工時間 C，下列公式(3)為計算完工時間 C 的公式， r_i 為抵達時間，P 為處理時間， r_i 為工件的數量，j 為工件。

$$C_j = \{\max(r_j, C_{j-1}) + P_j\} \quad (3)$$

當計算出考慮動態抵達時間的中間點 b 及考慮動態抵達時間的完工時間 C 後，可以利用完工時間 C 的中間點 b 找出該問題大小且考慮動態抵達時間的共同交期 K，而此共同交期 K 為考慮到早到及晚到而產生的，詳見公式(4)。

$$K = C_b \quad (4)$$

當完工時間及共同交期 K 計算出來後，可以以共同交期為基準點，往前為共同交期 K 減完工時間 C，往後為完工時間 C 減共同交期 K，這樣一來就會產生各個工件的總早到及總晚到的時間，下列公式(5)及公式(6)為計算各個工件總早到時間 E_{total} 及總遲到時 T_{total} ，j 為工件(job)。

$$E_{total} = \sum_{j=1}^b E_j = \sum_{j=1}^b \{\max(K, C_j) - C_j\} \quad (5)$$

$$T_{total} = \sum_{j=b+1}^n T_j = \sum_{j=b+1}^n \{C_j - \max(K, C_j)\} \quad (6)$$

2.2 Inver-over 演算法

圖 2 為方法 Inver-over 演算法的流程，主要是以 Inver-Over 搭配 Swap，Swap 為兩點交配，即為兩點交換的意思，在一開始先依單機排程問題的大小，產生限定範圍的亂數為父代，接下來的子代是利用 Inver-over 演算法去擾動其染色體的順序，其擾動的方法會以亂數 R 是否小於門檻值 P 去做判斷，小於使用 Swap，大於使用 Inver-over，再來就

是計算其共同交期、整備時間、完工時間等等的目標函數，最後本研究是以最小遲到時間為研究目標，因此當計算完目標函數後，就是讓擾動完的染色體順序解(即工件的順序解)給菁英政策去篩選並選擇，選擇完後再去判斷其解是否達到，沒達到即回到擾動順序前執行。

Guo & Michalewicz [6]，於研究中提到 Inver-over 演算法的步驟及流程，以圖 1 為例，P1 及 P2 比喻成染色體，P1' 為經過 Inver-over 演算法後產生的，下列為 Inver-over 演算法的步驟及 Inver-over 演算的邏輯式：

- 步驟一：在 P1 隨機產生起始點 c1；
- 步驟二：將 P1 產生的 c1 往右之工件設定為 Inver-over 起始點 cs；
- 步驟三：利用 P1 的 c1 找尋 P2 的 c1；
- 步驟四：利用 P2 的 c1 往右之工件設定為 Inver-over 終點 ce；
- 步驟五：利用 P2 的 ce 找尋 P1 的 ce；
- 步驟六：將 P1 的 cs 到 ce 做 Inver-over 的動作並產生 P1'。

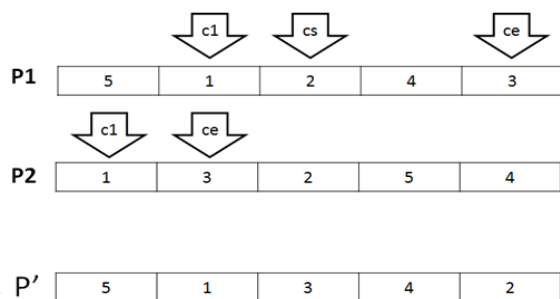


圖 1 Inver-over 流程

下列為 Inver-over 演算法的虛擬碼，R 為亂數且 R 的設置值為 U(0,1)，P 為實驗門檻值，小問題世代數設定為 1000，大問題是代數設定為 2000，族群大小設定為 100。

Inver-Over 演算法:

```

1: for i=1 to 1000 do
2:   for x = 1 to 100 do
3:     if R < P
4:       Swap ()
5:     else
6:       Inver-over ()

```

```

7:       end if
8:     end for
9: end for

```

下列邏輯是為 Inver-over 的虛擬碼，P1 及 P2 為兩組工件順序，P' 為經過 Inver-over 後產生的工件順序，亂數 R 的設定值為 U(0,1)，c1 為 P1 之起始點，c1 往右之工件設定為反轉內容之起始點即 cs，P2 的 c1 往右之工件設定為 Inver-over 終點 ce，下列為 Inver-over 的邏輯式。

Inver-over ()邏輯式：

```

1: R ← U(0,1)
2: c1 ← R
3: cs = P1[c1+1]
4: for i = 0 to P1
5:   if P1[c1+1] == P2[i]
6:     c1 = P2[i]
7:     ce = P2[i+1]
8:   end if
9: end for
10: for i = cs to ce], i--
11:   P' [cs] to P'[ce] = P1[i]
12: end for

```

2.3 Inver-over CX 演算法

圖 8 為本研究的方法 Inver-over CX 演算法的流程，主要是以 Inver-Over 演算法搭配 PMX[14] 為主軸，並結合基因演算法的特性，將其工件順序比喻為染色體，流程一開始先依單機排程問題的大小，產生限定範圍的亂數為父代，接下來的子代是利用 Inver-over CX 演算法去擾動其染色體的順序，其擾動的方法會以亂數 R 是否小於門檻值 P 去做判斷，小於使用 PMX，大於使用 Inver-over，再來就是計算其共同交期、整備時間、完工時間等等的目標函數，最後本研究是以最小遲到時間為研究目標，因此當計算完目標函數後，就是讓擾動完的染色體順序解(即工件的順序解)給菁英政策去篩選並選擇，選擇完後後，再去判斷其解是否達到，沒達到即回到擾動順序前執行。

下列為 Inver-over CX 演算法的虛擬碼，R 為亂數且 R 的設置值為 $U(0,1)$ ，P 為實驗門檻值，小問題世代數設定為 1000，大問題是代數設定為 2000，族群大小設定為 100。

Inver-Over CX 演算法:

```

1: for i = 1 to 1000 do
2:   for x = 1 to 100 do
3:     if R < P
4:       PMX ()
5:     else
6:       Inver-over ()
7:     end if
8:   end for
9: end for

```

3 實驗設計與分析

在實驗設計與分析本研究將會以 Rabadi et al.[7]所提出的問題大小 10、15、20、25 去作測試，但 Rabadi et al.[7]於該研究之求解目標，並不是解決動態抵達時間的問題，因此本研究依照公式一產生動態時間到測試問題中。

工件數 10、15、20、25 的每 1000 世代一次實驗，本研究將每個世代數設定為 100 個族群，一個世代即為一次實驗，而方法比較則是將每個世代的第一名為一次實驗的最佳解取出並比較，本次實驗有 8 種問題大小，每個大小有 3 種問題類型 (low、med、high)，而每個問題類型有 15 個重覆之測試案例，並將每個案例執行 30 次實驗，所以本研究共 180 個問題(4 × 3 × 15)。

實驗一開始先提出方法的最佳門檻值設定，之後再將本研究方法 Inver-over CX 與過去的方法 Inver-over 演算法去作比較，最後再由統計軟體 Minitab 測試分析並計算出代表本次研究的比較目標函數值，其目標函數的計算，請參照公式(1)~公式(6)。受限於篇幅限制，僅能於 3.1 中將呈現數據分析及方法的比較。

3.1 方法數據比較與分析

本研究已利用實驗設計方法篩選適合參數後，新方法 Inver-over CX 篩選的門檻值為 0.5，要比較的方法 Inver-over 演算法篩選的門檻值為 0.1，所以本節將會以這兩個方法篩選後的門檻值作為比較，方法門檻值的比較表，詳見表 1。

表 1 方法比較表

方法	門檻值 P
Inver-over CX	0.5
Inver-over	0.1

透過 Minitab 統計軟體，GLM 產生的 ANOVA 分析表由表 2 呈現，而來源的 Size 為問題大小，如 10、15、20、25，Type 為問題種類，如 low、med、high，Value 為門檻值，如 0.1、0.5、0.9，Method 為比較方法 Inver-over CX、Inver-over 演算法，自由度為樣本數減一，Seq SS 為平方合，Adj SS 為經誤差值調整後的平方合，Adj MS 為平方合除自由度的到的均方和，F 為調整後的均方合除誤差值得到的臨界值，詳見表 2。可得知表中 Method 的 P 值小於 0.05，這也代表這兩組方法的比較是有顯著的因子，

表 2 方法比較 ANOVA 表

來源	自由度	Seq SS	Adj SS	Adj MS	F	P
Method	1	34339123668	34339123668	34339123668	1667.15	0.000
Type	2	3.58960E+13	3.58960E+13	1.79480E+13	871368.01	0.000
Size	7	8.06616E+14	8.06616E+14	1.15231E+14	5594395.68	0.000
Method*Type	2	782837207	782837207	391418603	19.00	0.000
Method*Size	7	30503496211	30503496211	4357642616	211.56	0.000
Type*Size	14	7.76869E+13	7.76869E+13	5.54907E+12	269404.76	0.000

Method*Type*Size	14	4240620629	4240620629	302901473	14.71	0.000
誤差	21552	4.43918E+11	4.43918E+11	20597511		
合計	21599	9.20711E+14				

P < 0.05 為顯著

由表 2 得知方法會有因子是顯著的後，所以本研究將在事後檢定的部份將採用 Bonferroni 檢定來將兩組方法分組並分析，進而得到最為顯著的方法。表 3 為兩組方法經由事後檢定 Bonferroni 後所產生的表，在 95% 可信賴度對方法進行分組，

分組 B 在均值及控制皆顯著，也就是本研究的新方法在經由 Bonferroni 檢定得知的結果為本研究的新方法 Inver-over CX 優於過去的方法 Inver-over 演算法。

表 3 方法比較檢定表

Bonferroni 檢定			
方法	N	均值	分組
Inver-over	10800	133004.7	A
Inver-over CX(控制)	10800	130449.2	B

95% 可信賴度對方法進行分組，分組 B 在均值及控制皆顯著

3. 結論

最後實驗結果在經過統計軟體 Minitab 的分析下，新的方法 Inver-over CX 與過去的 Inver-over 演算法有相當的顯著性，這也代表新方法 Inver-over CX 優於過去的 Inver-over 演算法，由過去解決單機排程的問題都是以靜態來假設，但現實中往往都會發生動態的問題，本研究提出的新方法 Inver-over CX 是以考慮動態抵達的情形為前提構想的，且在實驗測試的結果下也能優於過去的方法，此演算法除了在學術上提供貢獻外還可以有效的讓物流業及運輸業等等行業在業務上得到實際的幫助。

4. 參考文獻

[1] 余宣慧，「具有限等候時間限制之雙機流程式動態排程」，國立雲林科技大學工業工程與管理研究所，2004。

[2] 戚玉濤、劉芳、焦李成，「求解大規模 TSP 問題的自適應及規約免疫算法」，西安電子科技大學，2008。

[3] 黃柏維，「應用時間派翠網路於考慮整備時間和可替代途程之製造系統排程研究」，國立雲林科技大學工業工程與管理系，2013。

[4] 蔡志弘，「工型工廠交期設定模式之構建」，國立交通大學工業工程研究所，1997。

[5] 鄭東亮、薛雲燦、楊啟文、李斐，「基於 Inver-over 運算的改進離散例子群優化算法」，河南大學計算機與信息學院，2010。

[6] Guo, T. and Michalewicz, Z. "Inver-over operator for the TSP". 1998.

[7] Rabadi, G. and Mollaghasemi, M. and Anagnostopoulos, G. "A branch-and-bound algo-rithm for the early/tardy machine scheduling problem with a common due-date and sequence-dependent setup time," Computers & Operations Research 1727-1751, ISSN 0305-0548, 2004.

- [8] Glass, C. A., Gupta, J. N. D. and Potts, C. N. "Lot Streaming in Three- stage Production Processes," *European Journal of Operational Research*, 75(2), pp. 378-394, 1994.
- [9] Holland, J.H "Adaptation in Natural and Artificial System" Oxford, England: U Michigan Press.1975.
- [10] Kanet, J. "Minimizing the average deviation of job completion times about a common due date," *Naval Research Logistics Quarterly*, vol. 28, no. 4, pp. 643–651, 1981.
- [11] Jun, P. and Xinbao, L. and Panos, M. and Wenjuan, F., Shanlin, Y. , "Single machine serial-batching scheduling with independent setup time and deteriorating job processing times," *Optimization Letters*, 2014.
- [12] Chang, P.C. and Chen, S.H. and Lie, T. "A Genetic algorithm enhanced by dominance properties for single machine scheduling problems with setup costs," Yuan-Ze University, 2011.
- [13] Chen, S.H. and Chen, M.C. and Liou, Y.C. " Artificial chromosomes with genetic algorithm 2 (ACGA2) for single machine scheduling problems with sequence-dependent setup times",2014
- [14] Syswerdra, G. Uniform Crossover in Genetic Algorithms, *Proceedings of the Third International Conference in Genetic Algorithms*, Morgan Kaufman, Publishers, San Mateo,1989.