

南 華 大 學

資訊管理學系

碩士論文

利用兩點交配之 Inver-over 演算法解決單機排程與動

態抵達時間問題

A Consideration of Single Machine Scheduling Problems
with Dynamic Arrival Time and Sequence Dependent Setup
by Using a New Inver-over Operator

研究生：李 易 霖

指導教授：王 佳 文、陳 世 興

中華民國 一〇三 年 六 月

南 華 大 學

資訊管理學系

碩 士 學 位 論 文

利用兩點交配之 Inver-over 演算法解決單機排程與動

態抵達時間問題

A Consideration of Single Machine Scheduling Problems
with Dynamic Arrival Time and Sequence Dependent
Setup by Using a New Inver-over Operator

研究生： 李昌霖

經考試合格特此證明

口試委員： 邱宏彬
阮金隆

指導教授： 王佐文
陳世興

系主任(所長)： 洪錦達

口試日期：中華民國 103 年 6 月 25 日

誌 謝

經過了兩年的研究，終於順利的完成此論文，在這個學習的過程中，最感謝的人莫過於指導我的陳世興老師及王佳文老師，在碩一還未正式開學前，陳世興老師就已積極的與我討論研究方向，讓我很快的進入狀況，並且提供相當多的資源，讓我擁有良好的學習環境，在接下來的兩年中，每週不間斷的都會定期與我討論研究進度，以及關心我的學習狀況，每當遇到瓶頸時也都樂於幫我解決各種問題，而在這求學的過程當中，陳世興老師於我碩一升碩二時，升遷於正修科技大學，但陳世興老師對於學生的負責，依然繼續指導我到完成論文，並替我在南華大學安排另一位優秀的指導教授，而那位指導教授正是王佳文老師。

在碩一升碩二的階段，王佳文老師定期都會叮嚀我的論文進度及給我鼓勵，也在每當遇到瓶頸時也都樂於幫我解決各種問題，回想這兩年碩士生活的點滴，的確有很多讓人值得回憶的片段，當然首先必須感謝我的家人及女朋友在背後不斷給我支持，另外就是這兩年來所有修過課的老師，最後就是恭喜與我一起從大學讀到研究所的吳奇燊同學，雖然在陳世興老師升遷時選擇更換指導教授，但在這一年來的努力下也是以順利畢業的形式結束學習的生涯。

最後當然還要感謝口試老師，感謝你們花時間閱讀我的論文，並且給我肯定及指教，讓我的論文可以更完美的呈現。

李易霖 2014年6月

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

學生：李易霖

指導教授：王佳文 博士
陳世興 博士

南 華 大 學 資 訊 管 理 學 系 碩 士 班

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(2004)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。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。Armentano & Mazzini (2000) 在研究中也有提到各個工件都會有獨自的抵達時間即整備時間(Setup time)，所以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，且會更貼近實際遇到的狀況。

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。最後再利用基因演算法之特性，目的是為了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，本研究將會以 Rabadi et al.(2004)、Chen et al.(2014)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。最後實驗結果顯示新方法 Inver-over CX 優於 Inver-over。

關鍵詞：單機排程、Inver-Over、基因演算法、共同交期、整備時間



A consideration of single machine scheduling problems with
dynamic arrival time and sequence dependent setup by using a
new inver-over operator

Student : Yi-Lin Li

Advisors : Dr. Jia-Wen Wang
Dr. Shih-Hsin Chen

Department of Information Management
The Graduated Program
Nan-Hua University

ABSTRACT

The prior research of single machine scheduling problem with common due date and sequence-dependent setup assumed the static arrival time. That is, these research did not consider the dynamic arrival time. However, it is practical for the jobs arriving dynamically. As a result, this research is the first one to solve this new problem. In this study, we proposed some testing instances and also propose a new algorithm, inver-over CX. This proposed algorithm integrates the advantage of Inver-over operator which has faster convergence ability and the Partial Message Crossover could diversify the individuals in a population. We expect this new approach could enhance the solution quality.

There are 360 testing instances in total, including the different kind of job sizes, the three types of the setup distributions, and 15 instance replications. According to our extensive experiment studies, the new algorithm outperforms the inver-over operator significantly. This research, hence, provides a new research direction in scheduling problem and also proposed a new approach to improve the inver-over operator.

Keywords: Single machine, Inver-Over, Genetic algorithms, Common due date, Setup time



目錄

口試合格證明	i
誌謝.....	ii
中文摘要.....	iii
英文摘要.....	V
目錄.....	Vii
圖目錄.....	iX
表目錄.....	X
第一章、緒論	1
第一節 研究動機	1
第二節 研究目的	4
第三節 研究流程	6
第二章、文獻探討	8
第一節 單機排程	9
第二節 共同交期	10
第三節 整備時間	14
第四節 基因演算法.....	17
第五節 旅行推銷員問題.....	19
第六節 INVER-OVER 演算法	22
第三章、研究方法	24
第一節 動態抵達時間的單機排程問題及其貢獻.....	24
第二節 INVER-OVER 演算法	27
第三節 INVER-OVER CX 演算法.....	31
第四節 單機排程演算法與例子說明	35

第四章、 實驗設計與分析	37
第一節 實驗設計	38
第二節 方法數據比較與分析	46
第五章、 結論與建議	56
第一節 結論	56
第二節 建議	57
參考文獻.....	58
中文部分	58
西文部分	61



圖目錄

圖 1 研究流程	7
圖 2 基因演算法流程	19
圖 3 TSP 範例圖	21
圖 4 TSP 最佳解範例	21
圖 5 TSP 不是最佳解範例	22
圖 6 INVER-OVER 演算法流程圖	28
圖 7 INVER-OVER 流程.....	29
圖 8 INVER-OVER CX 演算法流程圖.....	32
圖 9 PMX(PARTIAL MESSAGE CROSSOVER)	33
圖 10 總延遲時間計算	36
圖 11 INVER-OVER CX 小問題門檻值區間圖	45
圖 12 INVER-OVER CX 大問題門檻值區間圖	45
圖 13 INVER-OVER 小問題門檻值區間圖.....	46
圖 14 INVER-OVER 大問題門檻值區間圖.....	46
圖 15 方法比較區間表	49
圖 16 CPU TIME 小問題區間圖.....	52
圖 17 CPU TIME 大問題區間圖.....	52

表目錄

表 1 抵達時間及時間點之排程問題分類.....	2
表 2 工件處理時間矩陣.....	35
表 3 計算出來的表格.....	35
表 4 實驗分組表.....	38
表 5 門檻值設定.....	39
表 6 實驗組合表.....	39
表 7 INVER-OVER CX 小問題門檻值篩選 ANOVA 表.....	40
表 8 INVER-OVER CX 大問題門檻值篩選 ANOVA 表.....	40
表 9 INVER-OVER 小問題門檻值篩選 ANOVA 表.....	41
表 10 INVER-OVER 大問題門檻值篩選 ANOVA 表.....	42
表 11 INVER-OVER CX 小問題門檻值分組檢定表.....	43
表 12 INVER-OVER CX 大問題門檻值分組檢定表.....	43
表 13 INVER-OVER 小問題門檻值分組檢定表.....	44
表 14 INVER-OVER 大問題門檻值分組檢定表.....	44
表 15 方法比較表.....	47
表 16 方法比較 ANOVA 表.....	47
表 17 方法比較檢定表.....	48
表 18 方法比較小問題 CPU TIME 的 ANOVA 表.....	49
表 19 方法比較大問題 CPU TIME 的 ANOVA 表.....	50
表 20 CPU TIME 小問題 BONFERRONI 檢定表.....	51
表 21 CPU TIME 大問題 BONFERRONI 檢定表.....	51
表 22 方法比較表.....	53
表 23 問題大小 10、15、20、25 的執行時間比較表.....	54
表 24 問題大小 50、100、150、200 的執行值間比較表.....	55

第一章、緒論

本章節將敘說為何於該研究探討，單機排成於考量動態抵達時間之問題的動機與目的，在第一節將呈現研究該問題的背景及研究動機，第二節將呈現經由動機而產生的研究目的，第三節將述說做這議題研究之流程。

第一節 研究動機

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。所以要使訂單如期完成，並使總完工時間最小或是總延遲時間最小，那要符合顧客的要求與縮短加工時間，使該產業能在激烈競爭的環境下保有一席之地，可以說生產排程的問題是一項重要的議題。

一般的排程問題大致上可分為單機排程、平行機器(Parallel Machines)、流水線(Flow Shop)等數種，而單機排程問題為所有排程問題之基礎，且解決單機排程問題之結果不單單只侷限於單機排程，對於平行機台或多機台排程等更複雜之機器環境之經驗法則，更提供了一個良好的理論基礎。李日春(2007)於該研究中提到，複雜之機器環境的排程問題常被分解為處理單機情況的各個問題。

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(2004)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。下表為依照工件抵達及時間點等兩項特性，進而將排程問題的分類，詳見表 1。

表 1 抵達時間及時間點之排程問題分類

靜態	所有工件在加工前就已經到達生產系統，已經不會在有變動。
動態	各工件抵達生產系統之時間不同或訂單數目與作業內容會隨時間變化而變動之情況。
確定	工件之生產作業時間已知且固定。
隨機	工件之生產作業時間不固定，可能呈現機率分配之模式。

而陳茂生(2003)提到工作排程的目標是要以最有效的方式將 n 項工作依規定的限制安排到 m 部機器上處理，Johnson (1954)在研究中依據理論，分析並提出簡易並有效解決 m 大於 2 的排程問題，Cook (1971)於 NP-hard 演算理論發展出來後，發現當 m 大於 2 時，一般的排程問題都屬於 NP-hard 的問題，甚至在 m 等於 1 即單機排程的問題也屬於 NP-hard 的問題，所以這也意味著並沒有簡易解決排程的方法。

張昱敬(2005)於研究中提到，單機排程的解決方法可分為兩大類，第一類為最佳化方法即是透過數理規劃，其求解時間一般而言遠大於啟發式及局部式的演算法，所以會局限於較不複雜的問題，但以

最佳化方法所求得的最佳解可用來當作績效評估的準則，第二類為啟發式演算法及局部搜尋法等，第二類對於特殊問題所設計的啟發式演算法其求解績效十分理想，且求解時間也相當迅速。而在局部搜尋法方面有基因演算法及模擬退火法(Simulated Annealing)等，根據使用局部搜尋法方面的測試結果，此類求解績效也相當理想。

李日春(2007)於該研究中提到，排程慢慢趨向複雜化，且在真實的狀況中要求得最佳解須花大量的時間，因此排程的相關議題已逐漸傾向啟發式演算法，並要求近似最佳解能在短時間內完成，而在近幾年的學術研究上，求解排程相關問題大都採用搜尋法，其中又以基因演算法被廣為使用，而在基因演算法的研究方面，白建志(2010)於研究中提到基因演算法是用來解決最佳化問題的方法之一，且是利用廣大空間的搜尋方式來解決組合最佳化問題，基因遺傳顧名思義就是將好的基因遺傳至下一代，所以 GA 的整體概念就是利用染色體透過相互交配及突變後產生新的下一代，而在從新的一代中挑選較佳的染色體，淘汰不好的，透過好幾個世代的演化之後，進而從中找出最佳的染色體，所以基因演算法於其他方法相較之下也具有較佳的求解效果，且利用基因演算法多點搜尋之特性，使在大型排程問題中找出最佳近似解。

Potts & Wassenhove (1982) 於該研究在解決單機排程的問題時，是以總延後時間(total tardiness)最小為目標，並提出分解式演算法，此分解式演算法是將問題分解成數個相當小的子問題，再利用動態規劃對各個子問題求解，那如果假設單機排程會遇到動態抵達的情形且以總延後時間最小為目標，Armentano & Mazzini (2000) 各個工件都會有獨自的抵達時間及整備時間，所以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。

綜合以上文獻可得知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)及整備時間，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。

第二節 研究目的

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 Guo & Michalewicz (1998) 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)，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 Guo & Michalewicz (1998) 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。

因此本研究會以改善路徑規劃效果顯著的 Inver-over 演算法為基礎，因為 Zheng et al. (2010) 於改善離散粒子群優化演算法中有提到，Inver-over 演算法收斂速度快、精準度高，但 Qi et al. (2008) 在

求解大規模 TSP 自適應規約免疫演算法時，證實 Inver-over 演算法求解的收斂效率優於其它演算法但在對於大規模的問題時，求解的收斂速度會變得較緩慢，所以本研究會加入在局部搜尋法方面效果相當的基因演算法增加 Inver-over 演算法的收斂速度，及在 Inver-over 演算法的過程中加入 PMX(Partial Message Crossover)增加順序的擾動進而提升整體的收斂效果。

因此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，因此本研究會加入 PMX 兩點交配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。

本研究的目的是有以下三點:

1. 探討單機排程的整備時間、抵達時間、共同交期等問題。
2. 以單機排程的總延遲時間最小為目標。
3. 最後依 Inver-Over 演算法的收斂及擾動工件的順序為新方法的比較依據。

第三節 研究流程

本研究流程分為六個步驟，如圖 1 所示，首先我們先做問題的釐清，再利用文獻的回顧去了解單機排程(Single Machine)與動態時間抵達的背景，以及透過文獻了解過去的研究使用什麼的方法去解決問題，當我們問題釐清後接著文獻探討，利用過去相關文獻的探討來找出是否有更好的方法來解決單機排程與動態抵達時間的問題。

文獻探討之後，本研究方法決定以 Inver-Over 演算法及 PMX 為基礎並結合基因演算法的特性讓染色體交配、突變進而產生更好的順序解，過程為大問題 2000 個世代、小問題 1000 個世代為基準，並以總延後時間最小為目標並做為工件順序最佳解的條件，接著就是作實驗的設計，本研究的測試對象為 10 至 200 個工件，其中抵達時間是以各個問題的處理時間為範圍去隨機產生的，並將其固定，完成這樣後就可以開始作實驗，測試出各個問題最佳順序解，最佳順序解的停止條件是以總延後時間最小為目標，最後再提出結論與貢獻，並針對未來研究發展給予建議。

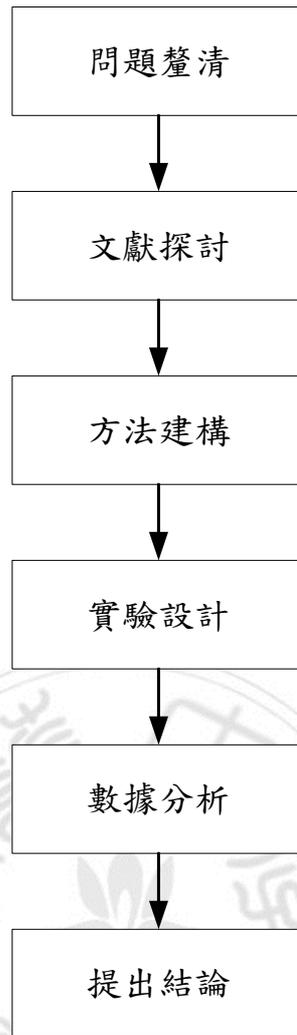


圖 1 研究流程

第二章、文獻探討

排程(Scheduling)為一種最佳化或多個目標的決策程序，必須在一定時間內分配有限資源來處理的工作，也就是將各個生產作業，根據其順序限制、作業時間以及所需的資源與設備，完成一系列的排序步驟或路徑，稱之為排程。

Pinedo(2002)將排程的機器環境分為數種，有單機排程、平行機台(Parallel Machines)、流程工廠(Flow Shop)等等。而排程的問題，多屬於 NP-hard 的問題，如果要解決排程的問題大致上分成四大類，有最佳化技術(Optimization Techniques)、區域搜尋法(Local Search)、人工智慧法(Artificial Intelligence)、派工法則(Dispatching Rules)。

單機排程是所有排程問題中最單純的一種，其涵義是指所有工件的加工均透過此機器來做加工處理，然而余宣慧(2004)提到在過去解決單機排程的問題都是以靜態來假設，但在實際的狀況卻存在著工件是動態抵達的情形，所以本研究在第二章會去針對單機排程相關的文獻去做探討，本研究在第二章會提出六個部分來做探討，首先第一節先探討單機排程的問題，利用文獻的探討做問題的釐清，接著在第二節探討的是共同交期，而在第三節是探討整備時間，第四節會針對基因演算法去做探討，在第五節探討旅行推銷員問題，第六節探討 Inver-over 演算法。

第一節 單機排程(Single Machine)

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，雖然單機排程在排程的問題中屬於較單純的一種，但也有較複雜的問題，所以考慮整個單機排程問題的完整性，如工件的早到(Earliness)、晚到(Tardiness)、成本(Cost)、時間限制(Time Constraint)、及機器的限制(Machine Constraint)等等問題，因此單機排程也是可以很複雜的。

Armentano & Mazzini (2000)於該研究針對單機排程，目標為使總延後時間最小的問題，提出基因演算法，其研究中，每件工作皆有獨自的抵達時間且整備置時間為順序相依(sequence dependent)，趙文涼(2000)探討考慮整備時間，交期絕對偏差和整備成本總和最小化的多目標單機排程問題。並藉由基因演算法的優良平行搜尋能力，搜尋最佳或是近似最佳之加工排程，針對母體大小、交配率、突變率和演化世代等參數加以分析討論，並且提出變動突變機率和固定突變機率，對於基因演算法求解結果之分析與比較，其中變動突變機率雖然會有較佳的結果出現，但是其缺點就是不容易有收斂的情況發生，反之固定變動機率的收斂效果較佳，而且求解品質也較變動突變機率為之穩定。

汪祐如(2006)提出考慮提早成本、延遲成本、運送成本下的單機排程問題，並在研究中指出，排程的問題應從工廠接獲訂單，由開始處理訂單，到完成訂單之需求，直到將所製造之完成品送至顧客手上時，才能算是結束訂單，而現實的狀況下，即使是單一顧客，可能對訂單要求之交期不同，因此對於訂單排程來說，已造成搜尋最佳排程

方式上的困難度，所以在解決訂單排程的問題是屬於 NP-hard 的問題，因此在求解過程的時間會呈現指數成長，故汪祐如(2006)在研究中發展兩階段解法以及運送演算法，然而在求解目標為提早成本、延遲成本及運送成本之總和最小化下的目標下，首先建立訂單排程模式，以找出適當的排程方式，再假設已知訂單排程之情況下，建立運送排程模式，並由最佳排程之性質中，找出規則建立運送演算法，根據結果得知，研究所提出之方法，能在較短時間內處理大量訂單的排程問題，並能兼顧求解品質。

施智懷(2003)提到，由於近幾年來，為了配合消費市場需求型態的變化，生產型態已由原先的少樣大量的生產模式轉變為多樣少量的生產模式，因此為了因應生產型態的多樣化，故牽涉了整備成本(Setup Cost)、提早完工成本(Earliness Cost)及延遲完工成本(Tardiness Cost)等問題，但如果一味追求某一項的最佳化，經常會造成另一項的損失，因此施智懷(2003)於研究中提出具動態權重之混合基因演算法(Hybrid Genetic Algorithms, HGA)，應用於考量順序相依整備時間的情境下，整備成本、提早完工成本與延遲完工成本等整體最小化的多目標單機排程問題，經由實驗證明，其結合動態權重值與排程搜尋之混合基因演算法，在求解時間與最佳化的搜尋上，均具有不錯的結果。

第二節 共同交期(common due date)

在過去工作排程的問題中，交期會是影響整個工作排程效率的因素之一，其意義蔡志弘(1997)有提到交期分為可預測及可控制兩種，可預測及為依工件去預測並規劃讓整個工作排程達到好的效率，可控制及為依整個流程去控制其工件順序進而達到整個工作排程好的效

率，Emmons(1994)考量工件早到及晚到的因素進而控制工件的交期，並發展出啟發式演算法來解決指定工作的共同交期，使加權的工作延遲及提早完成時間之和最小化。

要找出滿足訂單且較少生產成本的多廠訂單問題，一直是討論的重點，洪國庭(2007)在遇到生產排程的問題時，其特徵是以生產排程中所涉及的限制條件為主，如交期滿足、作業的先後順序、複雜的資源配置等，而限制條件的複雜會依不同的考量，所以交期規劃的重要性也跟排程的成本效率息息相關，蔡志弘(1997)而交期規劃是含意即是使所設定的交貨期有著可預測性(Predictability)和可控制性(Controllability)兩種特性。

彭武年(2006)在研究中指出，所謂交期可預測性，指的是訂單的交期符合產品在廠內實際的生產時間，但對於工廠的內外環境不可避免或無法預期的隨機變動因素，如停水、停電和機器故障等的應變能力，則為可控制性。

羅久哲(2009)考慮在具有工件釋放時限制下最小化提早與延遲成本之單機排程問題，在該研究的問題中，每一個工件有不同的釋放時間，此釋放時間代表工件進入機台等待加工的時間點，只有在此時間點或此時間點之後，工件才可進行加工，而所有的工作有一個相同的到期日，在到期日之前完工，會造成提早完工成本，而在到期日之後完工，會造成延遲成本，該研究的求解目標是找到一個提早時間與延遲時間最小的最佳解，該研究利用窮舉法與特例來驗證演算法的正確性，而效率評估的部分，窮舉法與該研究所提出之演算法進行比較，根據實驗結果平均顯示，該研究所提出演算法可有效地刪除超過

99%的點。

Liu & Min (2010) 提出延遲工件及單機準時生產機制的問題，研究中提到隨著準時生產機制的出現，準時(Just In Time, JIT)排序問題是對工件的早到、晚到都要受罰即都要付出代價的排序問題，因此 Liu & Min (2010) 指出，理想的加工排程應該要使所有工件在設定的交貨期完工即共同交期，故此研究以極小化的共同交期提出一個時間的優化演算法，並成功解決單機準時生產機制的問題。

尤金亭 & 唐恒永(2009) 提出單機排程的工件老化維護與共交期的問題，此研究中有提到，工件執行時間過常會導致效率變慢，而此工件的順序會影響整個單機排程的完工時間，故工件的處理時間決定在於工件的順序。此研究的目標函數包含交期成本及排程的總早到時間及工件無法在交期內抵達的成本，並且提出多項式動態抵達演算法，此演算法考慮到早到成本、晚到成本、受罰成本。

依據產業的不同，考量族群整備時間(family setup time)情況下，勢必有所不同，為了因應現實的改變，Chen et al.(1997)在環境為單機排程情況下，並考慮族群整備時間，設定共同到期日，且目標為最小化總延遲工作成本，但是並不是針對多目標問題進行探討，而在此研究解決該問題時，考慮兩種問題的兩種狀況並去做分析，第一種是小組處理，第二種是非小組處理，而在分析結果顯示是好的，並還發展出新的演算法來解決該問題的多項式時間。

Azizoglu & Webster (1997)在研究中提出的觀點反映多目標權重法融合的概念，設定共同交期的情況下，先將目標式使用權重法進行

合併即為設定下界，再利用分支界限法(branch-and-bound algorithm)進行求解的工作，最後結果顯示，在解決小範圍的問題時，能有效解決，但在解決大範圍的問題，在執行時間上較差。

Birgin and Ronconi (2012) 於該研究提出關於啟發式的單機排程問題及不同的整備時間與共同交期之問題，研究指出單機排程具有共同交期與不同的整備時間，而結果的計算是由早到及遲到的處罰加權與最小化，由於這個問題屬於 NP-hard 問題，故本研究利用此特點，以提高啟發式演算法的性能為研究，該研究通過 280 個測試案例，共 1000 多個的工件去做比較，最後結果顯示該研究提出之提高啟發式演算法的性能的研究是可行的。

Yin et al. (2013) 於該研究提出單機的分批排程問題及可控制的處理時間與分配共同交期，於該研究 Yin et al.(2013) 提到可控制的處理時間與分配共同交期是不同的，在一開始，該研究利用動態規劃的算法去找尋好工件的順序，及分配共同交期、資源與控制處理時間，最後將研究目的設定為，以最大的限度減少早到及晚到的作業成本，該研究證明，此研究有特殊的情況可以由低階(lower-order)多項式去解決，然後該研究找到不錯的方案，且也成功解決該問題。

第三節 整備時間(Setup time)

整備時間的涵意為在同一機台上運行兩個工件且兩個工件會因為順序的不同而產生的機台整備時間，例如紙尿布製造廠很重視整備時間，因為黃祥熙(2009)有提到一片紙尿布的生產週期僅數秒，但其生產流程的整備時間超過 20 小時。整備時間可以分為兩種，可預先進行的、不可預先進行的兩種，可預先進行的含意如零件未進加工站前，加工機台可先進入整備，例如工作排程的工件在上機台前已事先準備好，那上機台時就可避免準備的時間，黃柏維(2013)指出工作排程的工件可以預先進行加工整備時，可以避免機台過多的閒置時間，生產效能可以獲得大幅度的提升，且替代途程可以增加閒置機台的使用率又能增加系統的彈性，不可預先進行的含意如零件裝載至機台後才可進入整備，所以當機器可預先進行整備時將可大幅縮短生產所需要的時間。

Rabadi et al. (2004) 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 (1981)，Rabadi et al. (2004)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。

熊詩敏(2007)提到整備時間在未來要性，且提到整備時間在總生產時間比率占的越來越高，因此，該研究引用在 Rabadi et al.(2004)及在探討順序相依整備時間的條件下，考慮以下二種型態的排程問題，第一部份是探討共同交期，目標為最小化交期總偏差值(Tardiness & Earliness)的單機排程問題，第二部份則是考慮非等效平行機台的最

小化總時程 (Makespan) 的問題，熊詩敏(2007)於該研究所推導的 Dominance Property (DP)，是透過數學推導的方式，瞭解工件排序上的前後關係，得到一個較佳的排序，但是其容易陷入局部最佳解，因此該研究加入全域搜尋的演算法，本研究採用基因遺傳演算法，與模擬退火演算法，並將第一階段 DP 的結果當作是第二階段的初始解，成為為一種二階段的啟發式演算法，最後實驗結果發現，在單機及平行機台上在結合 DP 之後，GADP 與 SADP 的求解效果均與單純的 GA 與 SA 有顯著差異，且求解品質也更穩定。

Chang et al. (2011) 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. (2004)之延伸，Chang et al.(2011)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。

Tanaka and Araki (2013)於該研究提出，對於單機排程之總延遲加權與順序相依性的算法，該研究是作者以前對於整備時間的延伸，這是基於逐次提升動態規劃 SSDP (Successive Sublimation Dynamic Programming)的方法，該研究第一階段為結合梯度算法或列生成算法應用到拉式鬆弛的原生問題，第二階段為依約的條件讓鬆弛的上界 (Upper bounds)及下界(Lower bounds)之間的差距變為零，而鬆弛的抵銷是通過動態規劃及不必要的動態規劃去計算時間及記憶體空間的

增加，該研究的另一個方案以列生成算法去進行管理，以便解決實際的例子，該算法應用到指標性的實例中能有效解決問題。

Chen et al.(2014)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.(2011)延伸的研究，在Chen et al.(2014)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的平均誤差率是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。

Rajabinasab & Mansour (2010)在零工式排程的環境中，考量了動態的事件，例如可能到達的工作、不確定的作業時間、不可預期的機器停擺等，同時也考慮了零件的多樣途程和作業的彈性。此篇利用多層代理的方法，包括工作代理、作業代理和機台代理與五個派工法則做比較，並以模擬工具進行統計分析，結果顯示了代理的方法提供了較佳的績效和可靠性。

高清貴(2011)在考量整備時間與工件動態到達之單機延遲時間分析模型中有提到，工件到達時間的不確定性會使的整備時間及整個工件流程設定變得複雜，所以在整備時間的部份會影響所有工件到最後的完工時間，故本研究將會以控制共同交期已達到各個工件的早到遲到時間能大幅縮短已達到整個工件完工時間的縮短。

葉玉玲(2005)於具相依整備時間之等效平行機台流程型排程問題中，以總流程時間最小化之目標下，建構以模擬退火法為基之啟發式演算法 SATS，以快速尋求到一近似解。為了保持其演算法的穩定性，作者對影響演算法的參數進行分析並建議其最佳化參數設置，最後在數個小問題規模下與數學模式最佳解進行比較，在大問題下與傳統模擬退火法和禁忌搜尋法進行績效評估。

Pei et al. (2014) 於該研究提出單機串行分批排程與獨立整備時間的惡化工件處理時間問題，研究開始是以工件不斷惡化去做假設，接著就是將工件分批並連接進行，而每個分批得工件均有一個整備時間，最後該研究提出兩個演算法以最小完工時間及延遲工件數小為目標，而解決問題主要是假設兩種特殊狀況，即問完工時間最大及最小，該研究以此去產生優化的演算法，並且在這兩種狀況下設定獨立的整備時間，實驗結果顯示該研究是可行的。

第四節 基因演算法(Genetic algorithm)

基因演算法是用來解決最佳化問題的方法之一，由 Holland (1975)在 *Adaptation in Natural and Artificial System* 書中所提出，基因演算法是根據達爾文的「適者生存，不適者淘汰」的理論，效仿生物界物競天擇的自然演化法，以繁殖、突變、競爭、選擇的方式相互交配產生出較佳的後代來適應環境。過去的研究中已有很多人利用基因演算法來解決有關排程、路徑規劃、工作分配等最佳化問題。

Glass et al. (1994) 針對總完工時間最小化之目標，探討塔布搜尋法、模擬退火法及基因演算法，發現若改善鄰近解的搜尋方法，可提高基因演算法的求解品質。

Wang & Uzsoy (2002)針對工件具有動態到達之批量製程，求解最大延遲時間最小化之問題，研究中發展適合該問題之動態規劃演算法，並將該演算法與基因演算法結合並求解問題有很好的效果，Chang et al. (2002)回顧基因演算法於排程之應用，指出以基因演算法求解相當有效果。

林勇(2013)於該研究提出自行車路徑規劃於旅行銷售員問題使用基因演算法，該研究的問題為多目標點的旅行銷售員問題，而多目標點的旅行銷售員問題指的是由多個起始點出發而所完成之路徑，要求每一目標點只能通過一次，且必須回到原點。該研究目前已經可以完成目標點 10 以上的路徑規劃問題，而該研究之方法包含基因演算法之隨機初始化、交配、突變、菁英政策、基因毀滅等主要程序，當完成良好的路徑規劃，由電腦端的無線模組傳送引導循線自走車行進。每行經一個目標點時，即由自行車體發出無線射頻辨識(RFID)模組讀取目標點的射頻辨識標籤，判斷目標點資料化後，車體進行轉向，續朝下一目標點循線前進。當目標數為 10 時，設定染色體數為 200，最大世代數為 5000，突變率及交配率分別為 0.3 及 0.85，求得全域最佳解機率為 100%，該研究成功以基因演算法求解旅行銷售員問題之最短路徑，並以自行車驗證路徑規劃結果。

圖 2 為參考林豐澤(2005)的研究畫出的基因演算法流程圖，一開始先產生初始母體，再去計算適應函數，如果有達到目標就會結束，如果沒有則會進行選擇、交配、突變的動作，並將結果計算適應函數，如果有達到目標就會結束，如果再沒有就會一直執行至達到目的為止。

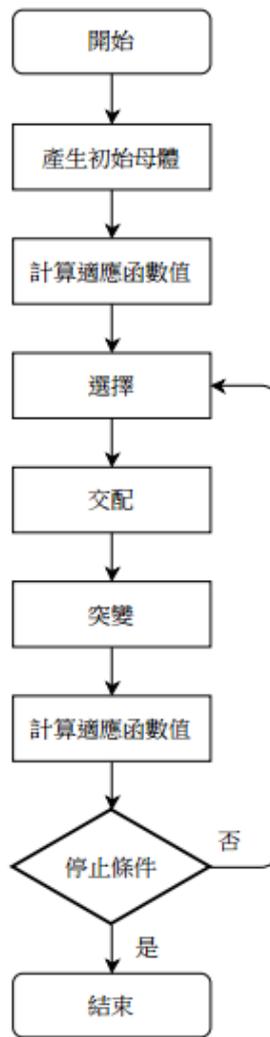


圖 2 基因演算法流程

第五節 旅行推銷員問題(Traveling Salesperson Problem)

Carter & Ragsdale(2003)於該研究定義了旅行推銷員問題是由一個旅行家要出發至多個旅遊景點，每個點均要拜訪一次，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。

白健志(2010)於該研究之文獻回顧中有提到，旅行銷售員在多點的排列組合中找出最佳化即是最短的距離、時間或成本的路徑提供旅行推銷員跑完所有的服務據點並回到出發點，此可行解為 $N!$ 的組合，

因此該問題為 NP-Complete 的問題。

張旭翔(2013)於該研究提到要如何有效率地管理生產排程，且在短時間內求取近似最佳解，於現今已經成為重要的議題，並引用陳柏志(2012)所提出的礦工基因演算法(Miner Genetic Algorithm, MGA)改良基因演算法的不足，並在求解旅行商問題上得到相當大的改善成果，因此該研究使用礦工基因演算法解決總加權完工時間最小化之單機生產排程問題，並以基因演算法做為比較基準，實驗結果顯示在一定時間內使用礦工基因演算法所求解出的值遠比基因演算法優良許多，收斂速度及收尋能力較為佳。

圖 3 為旅行推銷員問題的範例圖，圖 4 為最佳解範例圖，圖 5 為不是最佳解的範例圖。圖 3、圖 4、圖 5 的白點及為旅行銷售員要完成的旅行點，黑點為出發點，箭頭為旅行的路徑及方向，理想的路徑為圖 4，因為線條交錯的少，這樣也意味著每個旅行點是以最靠近且不重複交錯去執行的，這樣可以以最快的效率完成並回到起始點，圖 5 則是不是最佳解的範例，因為路徑不斷之交錯且旅行點到旅行點之間的距離較長，所以圖 5 的執行效率較差。

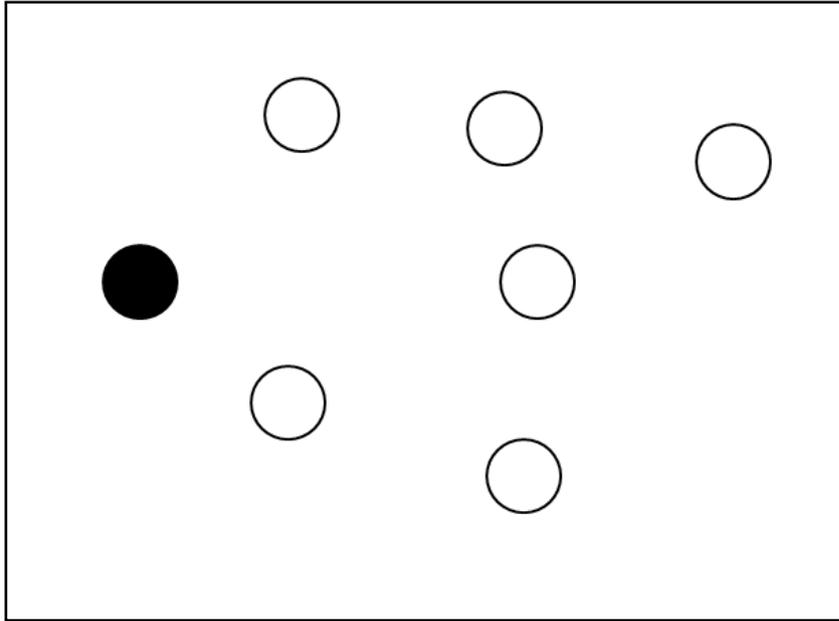


圖 3 TSP 範例圖

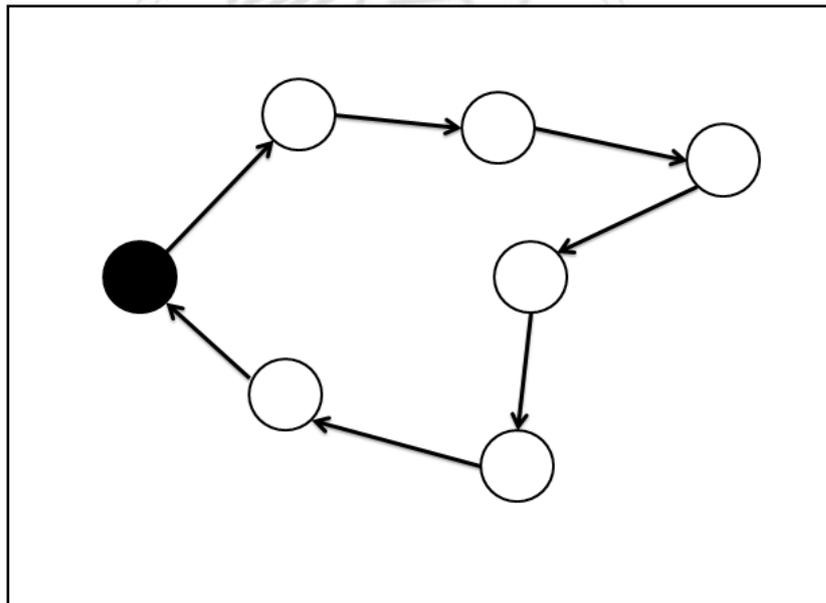


圖 4 TSP 最佳解範例

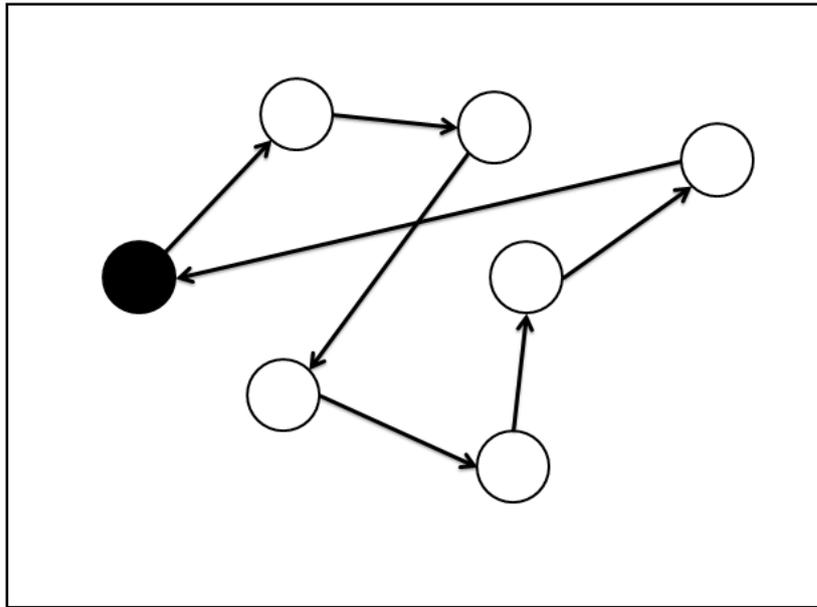


圖 5 TSP 不是最佳解範例

第六節 Inver-Over 演算法

過去的文獻中，Guo & Michalewicz (1998) 利用 Inver-over 解決旅行推銷員問題，此研究預先將 Inver-over 演算法設定參數值，並將過程設定為當值的範圍小於參數值時以兩點交配的 Swap 為執行依據，當值的範圍大於參數值時以 Inver-over 演算法為執行依據，而此演算法目的在於改善路徑的長度，故在解決排程相關問題時，常常被引用。

胡哲維 & 林葭華(2007)在以優先權選擇的混合式基因演算法解決旅行家問題一文中提到，旅行家問題及路經規劃問題，而旅行家問題也屬 NP-hard 問題，且無法在多項的時間內求得最佳解，故此研究以 2-OPT 區域搜尋法結合基因演算法提出 2-OPT 混合式基因演算法，並以 Inver-over 運算子改善路徑的長度，最後結果顯示 2-OPT 混合式基因演算法可改善傳統基因演算法搜尋最佳解的時間。

Qi et al. (2008) 在求解大規模 TSP 自適應規約免疫演算法時，證實 Inver-over 演算法求解的收斂效率優於其它演算法但在對於大規模的問題時，求解的收斂速度會變得較緩慢，所以本研究在解決單機排程的問題時，會結合兩點交配的 PMX，其目的是增加順序擾動。

Zheng et al. (2010) 於改善離散粒子群優化演算法中有提到，Inver-over 演算法收斂速度快、精準度高，離散粒子群演算法能在區域或全域搜尋得到最佳解，但收斂速度慢，故以 Inver-over 演算法收斂速度快結合離散粒子群演算法，以此改善離散粒子群的收斂速度，最後結果顯示，在與普通的離散粒子群優化演算法相較之下，改善的離散粒子群優化演算法，收斂速度優於普通的離散粒子群優化演算法。

鄭彬彬 & 王文義(2009) 以改善 Inver-over 演算法解決 TSP 旅行家問題，改善原先 Inver-over 演算法中，當值的範圍小於參數值時以兩點交換的 Swap 為執行依據，當值的範圍大於參數值時以 Inver-over 演算法為執行依據的方式，改為以鄰近點為優先的方式，實驗證實可有效縮小搜索空間、提高效率。

Tasgetiren et al. (2013) 於該研究提出貪婪演算法及 Inver-over 演算法解決旅行銷售員問題，而貪婪演算法(Greedy algorithm)主要是基於破壞及施工的算法，其意義為破壞及從中獲得解決方案，進而得到更完整的演算法，但以貪婪演算法解決旅行銷售員問題，會有成本高的問題，因為要破壞及重建，故該研究以結合 Inver-over 演算法解決旅行銷售員問題，在實驗部分，該研究以 TSPLIB 的問題去做測詢，實證結果證實，該研究是不錯的。

第三章、 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，在第一節乃動態抵達時間的單機排程問題及其貢獻，第二節簡介 Inver-over 演算法之流程，第三節為本研究之方法 Inver-over CX，第四節為測試範例。

第一節 動態抵達時間的單機排程問題及其貢獻

Armentano & Mazzini (2000)於該研究有提到，各個工件都會有獨自的抵達時間即整備時間，所以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，且會更貼近實際遇到的狀況。

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 組去兩兩比較，所以本研究研究會先去計算其總延遲時間的目標函數，這當中會計算到各個工件的共同交期 C 、抵達時間 r_i 、早到時間 E_{total} 、遲到時間 T_{total} 等等。

本研究在計算案例前必預先產生各個案例大小中，各個工件的抵達時間 r_i ，而抵達時間 r_i 的產生是利用案例中的處理時間為範圍，而抵達時間 r_i 的產生是以亂數產生的，亂數的範圍是以Rmax為範圍最大值先去計算Rmax，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。

$$R_{max} = (P + S) \times (N - 1) \quad (1)$$

下列為第四節將要使用的抵達時間，順序為因應第四節節測試所預先產生的，實際跑實驗時是以 Inver-over CX 演算法為基準去擾動產生的， r_i 為經由Rmax為範圍產生出來的。

r_i : 142、19、94、31、73

順序 : 5→1→2→4→3

在計算以考慮動態時間的共同交期 K 必預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， n 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(4)及公式(5)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)。

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

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

在計算目標函數前則是利用基因演算法的菁英政策及本研究的新方法 Inver-over CX 去進行突變及交配的動作。鄒佳臻(2009)有提到兩點交配的演算法在工作排程中即是將工作隨機交換，這樣就可以避免尋找最佳解時落入區域最佳解，本研究的新方法 Inver-over CX 即 Inver-over 演算法搭 PMX 也屬兩點交配的一種。

Zheng et al. (2010) 在解決小範圍的工作排程問題時其收斂解的速度優於其它演算法，所以在解決單機排程的問題上本研究將會以兩點交配的 Inver-over 演算法及兩點交配的 PMX 為核心，且會在 Inver-over 演算法中加入基因演算法讓其收斂效率提升並有效的去擾動單機排程中工件的順序並獲的最佳目標解，最後不會因為 Inver-over 演算法只有在小範圍的問題中才能比其他演算法來的好的缺失進而影響單機排程解決大範圍問題的效率，故此樣的結合能有效提升本研究所提出方法 Inver-over CX 的效率。

本研究會採取 Inver-over 演算法及兩點交配的 PMX 並去結合基因演算法的方式不外乎是考量到基因演算法適者生存，不適者淘汰的特性外，其中的菁英政策及在染色體選擇的時候以兩兩競爭的方式去選擇也是本研究在此方法的重要核心。

第二節 Inver-over 演算法

圖 6 為方法 Inver-over 演算法的流程，主要是以 Inver-Over 搭配 Swap，Swap 即兩點交配，為塞選出的兩點個工件交換位置的意思，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，圖 6 中的 R 為亂數的範圍，P 為實驗的門檻值，詳見圖 6。

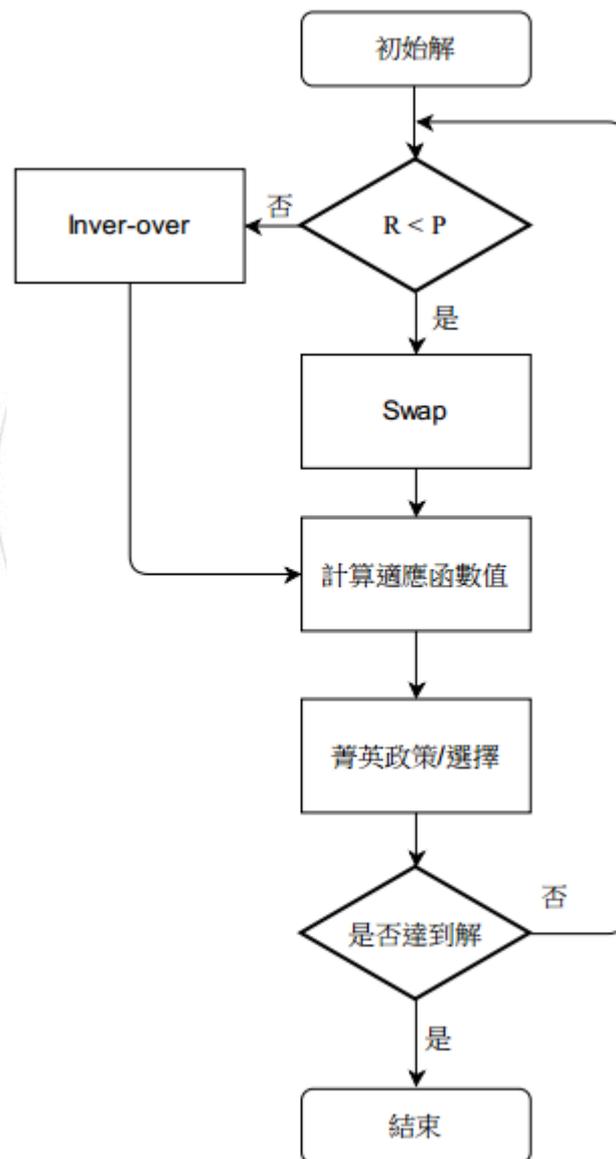


圖 6 Inver-over 演算法流程圖

Guo & Michalewicz (1998)，於研究中提到 Inver-over 演算法的步驟及流程，以圖 7 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 P'。

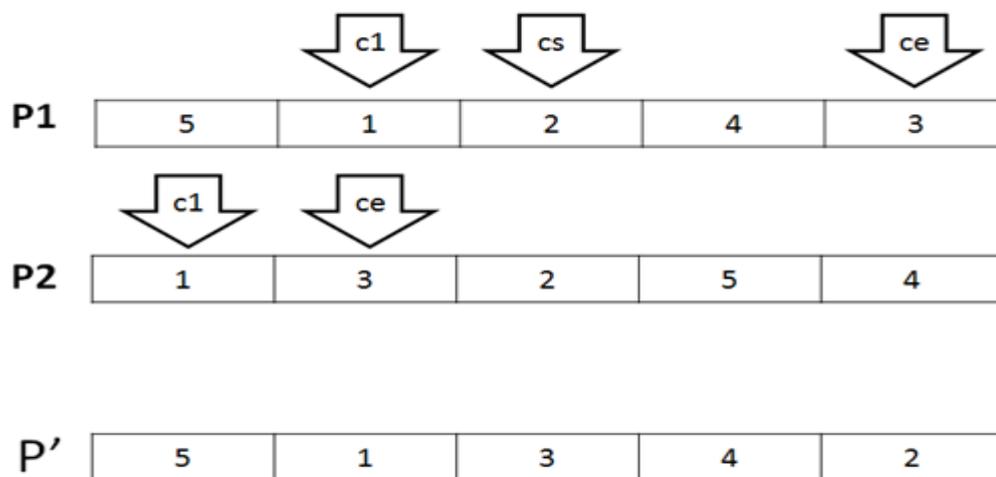


圖 7 Inver-over 流程

下列為 Inver-over 演算法的邏輯式，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 演算法邏輯式：

```
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 的邏輯式，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
```

第三節 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 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，圖 8 中的 R 為亂數的範圍， P 為實驗的門檻值，詳見圖 8。

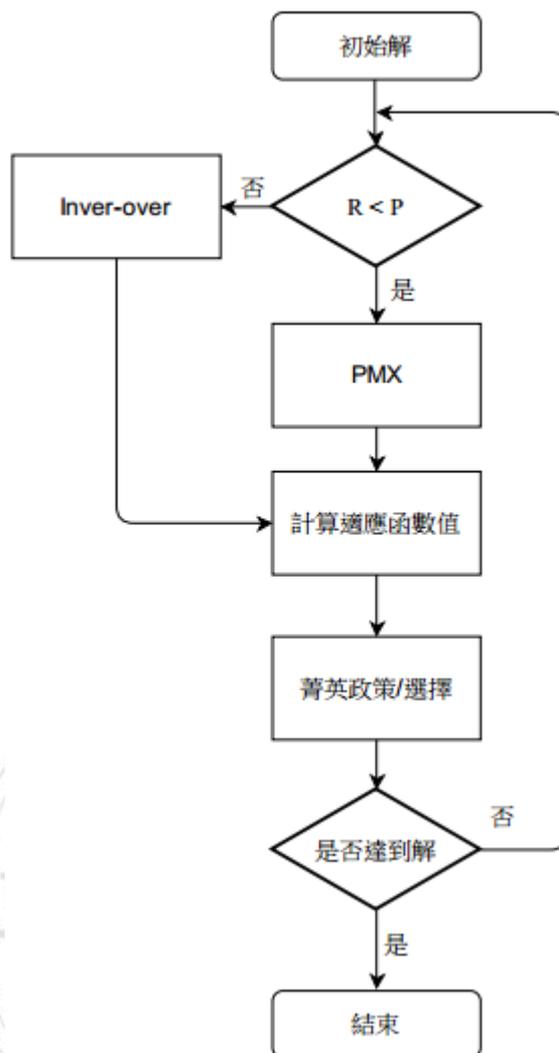


圖 8 Inver-over CX 演算法流程圖

以圖 9 為例，P1 及 P2 比喻成染色體，New P1 及 New P2 為經過兩點交配的 PMX 後產生的，下列為 PMX 的步驟及 PMX 的邏輯式：

- 步驟一：在 P1 隨機產生起始點 A 及終點 B；
- 步驟二：將 P1 產生的起點及終點設立在 P2 上；
- 步驟三：將 P1 起點到終點的值複製至 New P2；
- 步驟四：將 P2 起點到終點的值複製至 New P1；
- 步驟五：將 New P1 及 New P2 剩下的空格依序將對應的 P1 及 P2 填入且不重覆；

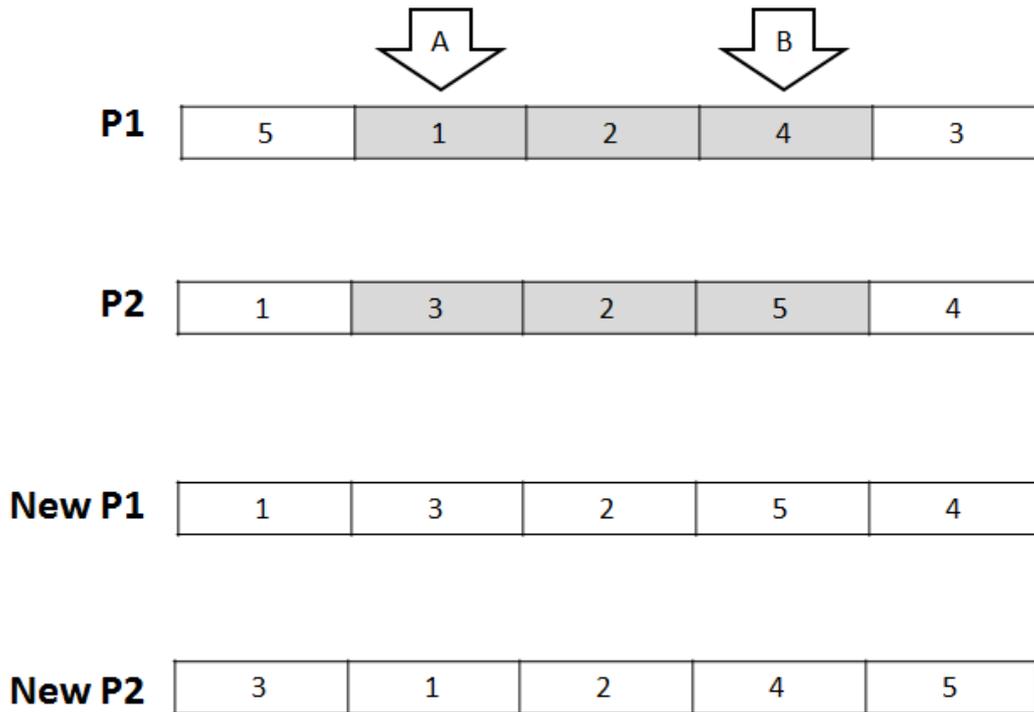


圖 9 PMX(Partial Message Crossover)

下列為 Inver-over CX 演算法的邏輯式，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 演算法邏輯式：

```

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

```

下列邏輯是為 PMX 的邏輯式，P1 及 P2 為兩組工建順序，New P1 及 New P2 為經過兩點交配的 PMX 後產生的工件順序，亂數 R 及 R1 的設定值為 U(0,1)，A 為起始點及 B 為終點，下列為 PMX 的邏輯式。

PMX()邏輯式：

```
1: R ← U(0,1)、R1 ← U(0,1)
2: A ← R、B ← R1
3: NEW P2[A] to NEW P2[B] = P1[A] to P1[B]
4: NEW P1[A] to NEW P1[B] = P2[A] to P2[B]
5: for i = 0 to P1
6:   for a = 0 to P1
7:     if P1[i] != NEW P1[a]
8:       NEW P1[a] = P1[i]
9:     end if
10:    if P2[i] != NEW P2[a]
11:      NEW P2[a] = P2[i]
12:    end if
13:  end for
14: end for
```

第四節 單機排程演算法與例子說明

本章節測試的工件範例如表 2，表中的幾個 1000 代表幾個工件數，也可由表 2 由上至下、由左至右依序為第一個工件至第五個工件，其它數字為工件的處理時間。

表 2 工件處理時間矩陣

	1	2	3	4	5
1	1000	71	71	70	33
2	26	1000	23	53	34
3	41	97	1000	35	19
4	88	64	37	1000	25
5	36	49	31	44	1000

表 3 為利用公式(1)~ 公式(6)及表 2 節範例所產生出來的表格，其中的順序於實驗時會以本研究的新方法 Inver-over CX 產出，而此表格的工件順序為假設產生的，到達時間 r_i 為公式(1)當範圍所產生出來的表格，而處理時間 P 是以工件順序的先後順序對應出來的，在完工時間 C 的部份是是利用工式(3)，去判斷到達時間的早晚，是否加上處理時間或是續加前一工件的完工時間所計算出來的，共同交期 K 是先利用公式(2)算出中間點，在利用已考慮動帶抵達時間的完工時間的中間點即公式(4)去計算出來的。

表 3 計算出來的表格

工件順序	到達時間	處理時間	完工時間	早到晚到
工件 5	73	36	109	266-109
工件 1	142	71	213	266-213
工件 2	19	53	266*	0

工件 4	31	37	303	303-266
工件 3	94	19	322	322-266
*共同交期			Obj 303	

依表 3 與圖 10 可知，K 為共同交期線值為 266，x 為完工時間線，以共同交期線為基準，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，最後將相減的值加總後可得值為 303，所以利用基因演算法加入 Inver-Over CX 的染色體順序解為 303，此解越小代表越好。

所以說順序是會影響得到的解的，因此本研究才會採取利用基因演算法加入本研究 Inver-Over CX 演算法的方式去讓工件順序解篩選進而得到的解趨近最佳解，這樣在解決單機排程的問題時也能提升收斂到最佳解的效率。

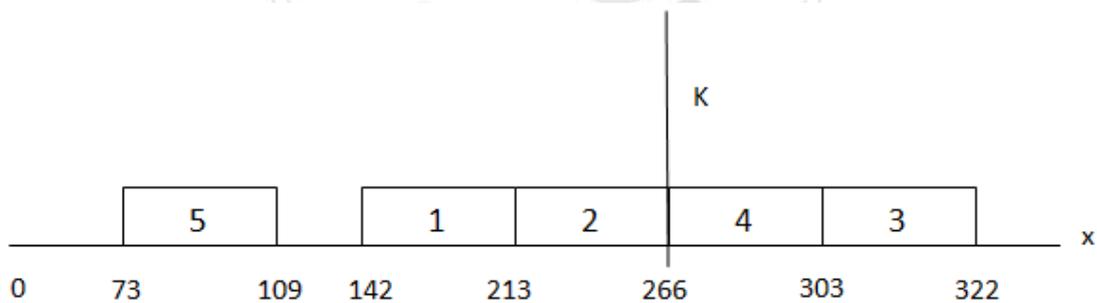


圖 10 總延遲時間計算

第四章、實驗設計與分析

在第四章本研究將會以 Rabadi et al.(2004)所提出的問題大小 10、15、20、25 去作測試，但 Rabadi et al.(2004)於該研究之求解目標，並不是解決動態抵達時間的問題，且 Rabadi et al.(2004)所提出的問題大小為小問題，因此本研究會加入 Chen et al.(2011)所提出的問題大小 50、100、150、200 問題去作測試，而這些問題會依照 Rabadi et al.(2004)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，小範圍均值(10, 60)為 low，中範圍均值(10, 110)為 med，大範圍均值(10, 160)為 high。

本研究將問題總類(low、med、high)依據過去文獻 Rabadi et al.(2004) 和 Chen et al.(2011)依問題大小(工件數)分成小問題及大問題兩大類，工件數 10、15、20、25 為小問題，工件數 50、100、150、200 為大問題，小問題的每 1000 世代一次實驗，而大問題為小問題的兩倍，故大問題設定為 2000 世代為一次實驗，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，所以本研究共 360 個問題(8×3×15)，並將每個案例執行 30 次實驗。

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)。在本章第一節中我們

將作實驗設計門檻值設定及實驗環境的介紹，及在重覆實驗次數方面的敘述，之後再利用所得到的數據找出本次實驗之最佳的門檻值。第二節中將呈現數據分析及方法的比較。

第一節 實驗設計

本次實驗目的，將研究方法中所建構出的 Inver-over 演算法搭配 PMX 為新的方法，命名為 Inver-over CX，並將此研究方法與過去的方法 Inver-over 演算法去作比較。測試問題為依處理時間區分的 low、med、high，問題大小為 Rabadi et al.(2004)及 Chen et al.(2011)所提出的問題大小 10、15、20、25 及 50、100、150、200 作實驗設計，重覆實驗時即是將這三種問題的各個問題大小各跑 30 次並均分至五台機器上去測試，測試機器的規格為亞馬遜 Xeon(R) E5-2670 V2 處理器 2.50GHz、記憶體為 244G，數據的測試軟體為 Minitab 第 16 版，方法內容比較詳見表 4。

表 4 實驗分組表

方法	運算子一	運算子二
Inver-over CX	Inver-over	PMX
Inver-over	Inver-over	Swap

本研究在本次實驗門檻值設定為 0.1、0.5、0.9，因為在兩個方法比較的實驗開始之前我們必須先設定好門檻值，由表 5 可見門檻值的設定表，表中因子的運算子一及運算子二如表 4，表中的 Level 組別為門檻值的設定，詳見表 5。

表 5 門檻值設定

因子	Level 1	Level 2	Level 3
運算子一	0.1	0.5	0.9
運算子二	0.1	0.5	0.9

由表 6 可以看到 R 為亂數，P 為實驗的門檻值，本研究的方法 Inver-over CX 在 R 小於 P 時將會選用 PMX，大於時選用 Inver-over，比較的方法 Inver-over 在 R 小於 P 時將會選用 Swap，大於時選用 Inver-over，詳見表 6。

表 6 實驗組合表

方法	R < P	R > P
Inver-over CX	PMX	Inver-over
Inver-over	Swap	Inver-over

透過 Minitab 統計軟體，GLM 產生的 ANOVA 分析表由表 7、表 8 可見，當門檻值依分組去作比較而產的 ANOVA 表可以看出表 7、表 8 在門檻值的 P 值皆小於 0.05，這也代表本研究的方法 Inver-over CX 在測試問題大小 10 至 25 及 50 至 200 門檻值的篩選會有顯著的因子。表 7、表 8 來源的 Value 為門檻值 0.1、0.5、0.9，Size 為實驗測試的問題大小(工件數)10、15、20、25、50、100、150、200，Type 為問題種類，如 low、med、high，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 7、表 8。

表 7 Inver-over CX 小問題門檻值篩選 ANOVA 表

來源	自由度	Seq SS	Adj SS	Adj MS	F	P
Value	2	108910800	108910800	54455400	991.15	0.000
Type	2	12547538884	12547538884	20480367609	1141189.93	0.000
Size	3	61441102828	61441102828	55215753121	372766.61	0.000
Value*Type	4	14890113	14890113	3722528	25.19	0.000
Value*Size	6	6151920980	6151920980	1025320163	18662.02	0.000
Type*Size	6	13772504996	13772504996	2295417499	15535.37	0.000
Value*Type*Size	12	101473302	101473302	8456108	57.23	0.000
誤差	16164	888074868	888074868	54942		
合計	16199	81137548360				

P < 0.05 為顯著

表 8 Inver-over CX 大問題門檻值篩選 ANOVA 表

來源	自由度	Seq SS	Adj SS	Adj MS	F	P
Value	2	1.50751E+11	1.50751E+11	75375405000	1680.43	0.000
Size	3	6.75417E+14	6.75417E+14	2.25139E+14	5019275.29	0.000
Type	2	1.05481E+14	1.05481E+14	5.27404E+13	1175801.52	0.000
Value*Size	6	5.11407E+12	5.11407E+12	8.52345E+11	18618.56	0.000
Value*Type	4	206405442	206405442	51601361	1.13	0.342
Size*Type	6	6.48685E+13	6.48685E+13	1.08114E+13	241031.04	0.000
Value*Type*Size	12	1023931748	1023931748	85327646	1.86	0.034
誤差	16164	7.25034E+11	7.25034E+11	44854873		
合計	16199	8.46642E+14				

P < 0.05 為顯著

透過 Minitab 統計軟體，GLM 產生的 ANOVA 分析表由表 9、表 10 可見，當門檻值依分組去作比較而產的 ANOVA 表可以看出表 9、表 10 在門檻值的 P 值皆小於 0.05，這也代表本研究的比較方法 Inver-over 在測試問題大小 10 至 25 及 50 至 200 門檻值的篩選會有顯著的因子。表 9、表 10 來源的 Value 為門檻值 0.1、0.5、0.9，Size 為實驗測試的問題大小(工件數)10、15、20、25、50、100、150、200，Type 為問題種類，如 low、med、high，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 9、表 10。

表 9 Inver-over 小問題門檻值篩選 ANOVA 表

來源	自由度	Seq SS	Adj SS	Adj MS	F	P
Value	2	12359880	12358800	61799400	1148.26	0.000
Type	2	12883495189	12883495189	6441747595	119690.24	0.000
Size	3	63320955974	63320955974	21106985325	392176	0.000
Value*Type	4	140232	140232	35058	0.25	0.909
Value*Size	6	558193	558193	93032	0.67	0.672
Type*Size	6	6210760567	6210760567	1035126761	19233.07	0.000
Value*Type*Size	12	101473302	974754	81230	0.58	0.850
誤差	16164	869949060	869949060	53820		
合計	16199	83408759590				

P < 0.05 為顯著

表 10 Inver-over 大問題門檻值篩選 ANOVA 表

來源	自由度	Seq SS	Adj SS	Adj MS	F	P
Value	2	1.09448E+11	1.09448E+11	5472408156	1462.07	0.000
Size	3	6.74688E+14	6.74688E+14	2.24896E+14	6008600.21	0.000
Type	2	1.05248E+14	1.05248E+14	5.26242E+13	1405973.73	0.000
Value*Size	6	4.45532E+12	4.45532E+12	7.42553E+11	12473.71	0.000
Value*Type	4	5.26382E+11	5.26382E+11	1.31596E+11	2210.60	0.000
Size*Type	6	6.51285E+13	6.51285E+13	1.08548E+13	290008.88	0.000
Value*Type*Size	12	5.25487E+11	5.25487E+11	43790586355	735.61	0.000
誤差	16164	6.05003E+11	6.05003E+11	37429034		
合計	16199	8.45780E+14				

P < 0.05 為顯著

在執行門檻值篩選的 ANOVA 分析之後，得知門檻值的篩選有顯著的因子後，所以本研究將在事後配對檢定的部份將採用 Bonferroni 檢定來將門檻值分組並分析，進而得到最為顯著的門檻值。當本研究的方法 Inver-over CX 在問題大小 10 至 25 及 50 至 200 時將方法進行分組並設定可置信度為 95%，透過 Bonferroni 分析檢定過後門檻值為 0.5 時為顯著的，這代表新的方法 Inver-over CX 在 Inver-over 演算法搭 PMX 這樣的組合執行機率各半時能產生最好的解，所以在比較方法時，Inver-over CX 將採用門檻值為 0.5 時去作比較，詳見表 11、表 12。

表 11 Inver-over CX 小問題門檻值分組檢定表

Bonferroni 檢定			
Value	N	均值	分組
0.9	5400	3306.8	A
0.1	5400	3230.8	B
0.5	5400	3107.8	C

95%可信賴度對方法進行分組，分組 C 在均值之間差異顯著

表 12 Inver-over CX 大問題門檻值分組檢定表

Bonferroni 檢定			
Value	N	均值	分組
0.9	5400	265026.2	A
0.1	5400	263026.2	B
0.5	5400	257791.2	C

95%可信賴度對方法進行分組，分組 C 在均值之間差異顯著

當本研究的方法 Inver-over 在問題大小 10 至 25 及 50 至 200 時將方法進行分組並設定可信賴度為 95%，透過 Bonferroni 分析檢定過後門檻值 P 值為 0.1 時為顯著的，這代表方法 Inver-over 演算法在 Inver-over 演算法搭 Swap 這樣的組合執行幾乎以 Inver-over 演算法為主時能產生最好的解，所以在比較方法時，Inver-over 演算法將採用門檻值 P 值為 0.1 時去作比較，詳見表 13、表 14。

表 13 Inver-over 小問題門檻值分組檢定表

Bonferroni 檢定			
Value	N	均值	分組
0.9	5400	3570.1	A
0.5	5400	3439.2	B
0.1	5400	3358.1	C

95%可信賴度對方法進行分組，分組 C 在均值之間差異顯著

表 14 Inver-over 大問題門檻值分組檢定表

Bonferroni 檢定			
Value	N	均值	分組
0.5	5400	268819.3	A
0.9	5400	264585.7	B
0.1	5400	262584.3	C

95%可信賴度對方法進行分組，分組 C 在均值之間差異顯著

圖 11、圖 12 為方法 Inver-over CX 在問題大小 10 至 25 及 50 至 200 門檻值區間圖，圖 1、圖 2 的 X 軸的 Value 為門檻值及問題的種類 Type，Y 軸為目標函數 Obj 的值，在圖 11、圖 12 可以很明顯的看出，不管問題的種類是 low、med、high 都是門檻值 0.5 時產生的目標函數解為本次門檻值篩選之優良解。

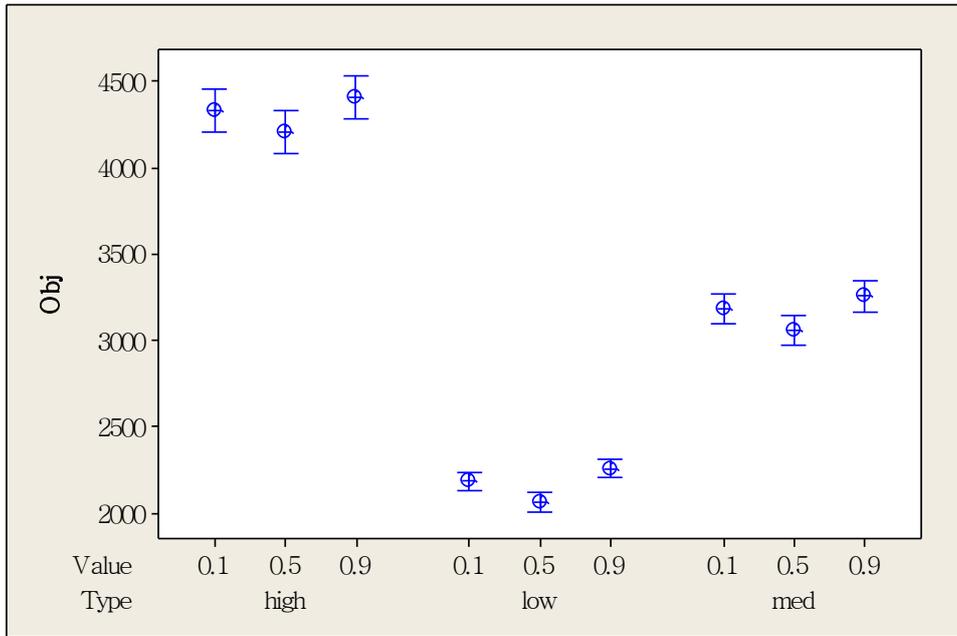


圖 11 Inver-over CX 小問題門檻值區間圖

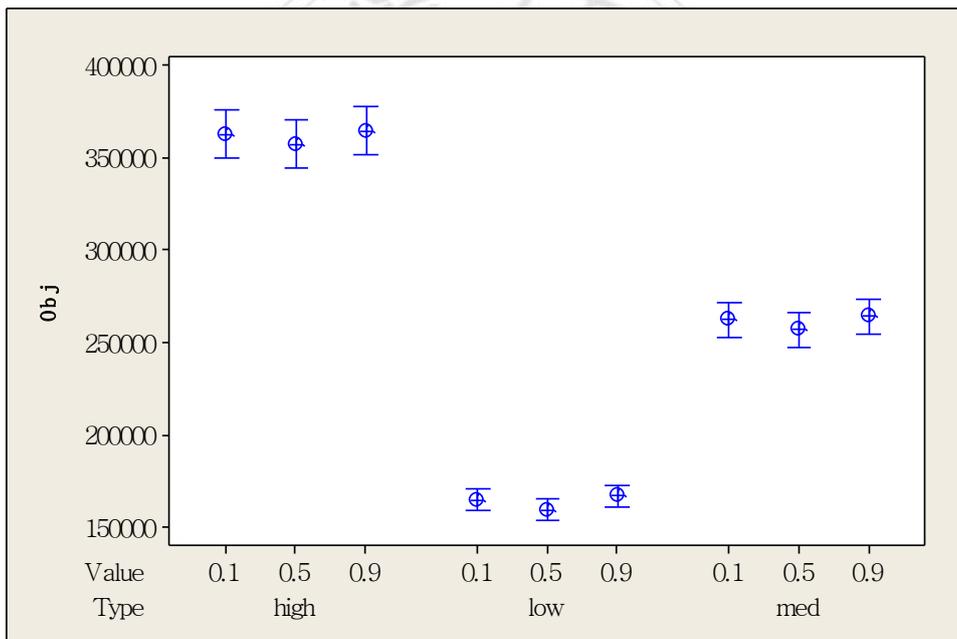


圖 12 Inver-over CX 大問題門檻值區間圖

圖 13、圖 14 為方法 Inver-over 在問題大小 10 至 25 及 50 至 200 門檻值區間圖，圖 13、圖 14 的 X 軸的 Value 為門檻值及問題的種類 Type，Y 軸為目標函數 Obj 的值，在圖 13、圖 14 可以很明顯的看出，不管問題的種類是 low、med、high 都是門檻值 0.1 時產生的目標函數 Obj 解為本次門檻值篩選之優良解。

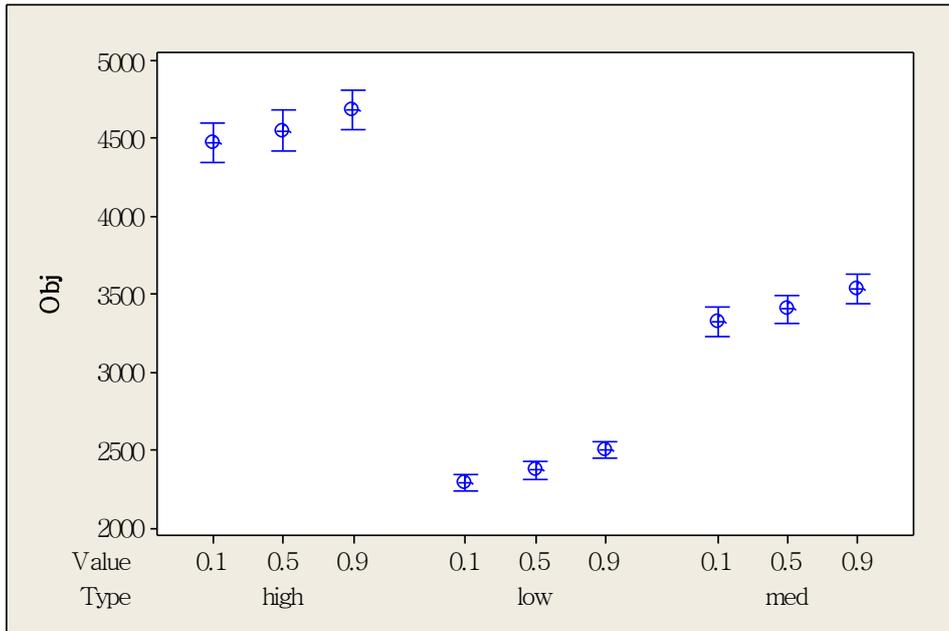


圖 13 Inver-over 小問題門檻值區間圖

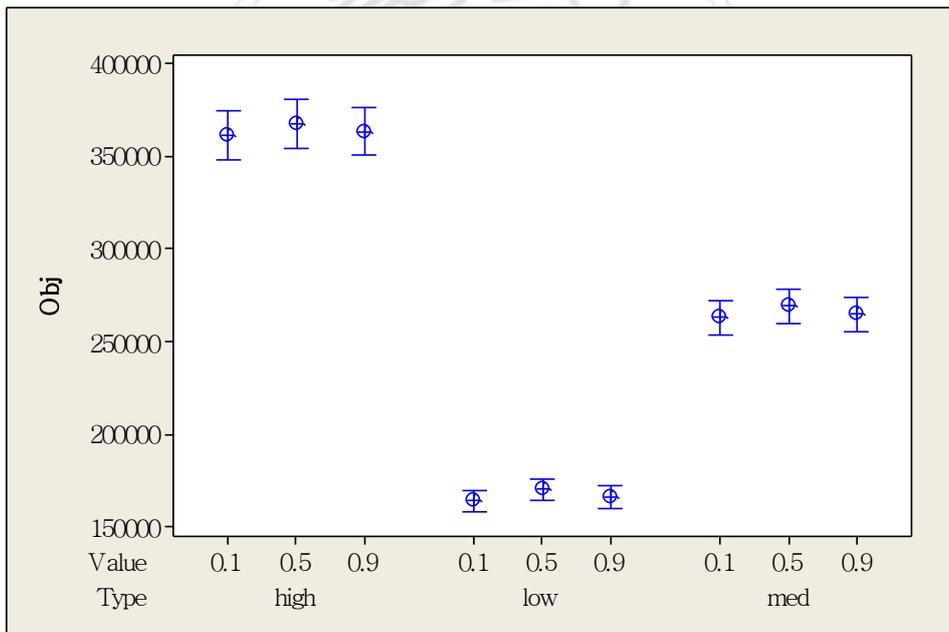


圖 14 Inver-over 大問題門檻值區間圖

第二節 方法數據比較與分析

經由上一節的數據、分析、最後篩選後，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 15。

表 15 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 15 可見，表 16 的 ANOVA 表是將兩個要比較的方法依上一節篩選的門檻值分成兩組去最比較，由表 16 的 ANOVA 表可得知表中 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 16 來源的 Size 為問題大小，如 10、15、20、25、50、100、150、200，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 16。

表 16 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				

由表 17 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 17 為兩組方法經由事後檢定 Bonferroni 後所產生的表，由表 17 可得知在 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，詳見表 17。

表 17 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

圖 15 為方法比較的區間圖，圖 15 的 X 軸的 Method 為比較的方法及問題的種類 Type，Y 軸為目標函數 Obj，由圖 15 可以很明顯的看出不管問題種類 Type 是 low、med、high 都是方法為本研究的新方法 Inver-over CX 時為最佳解，詳見圖 15。

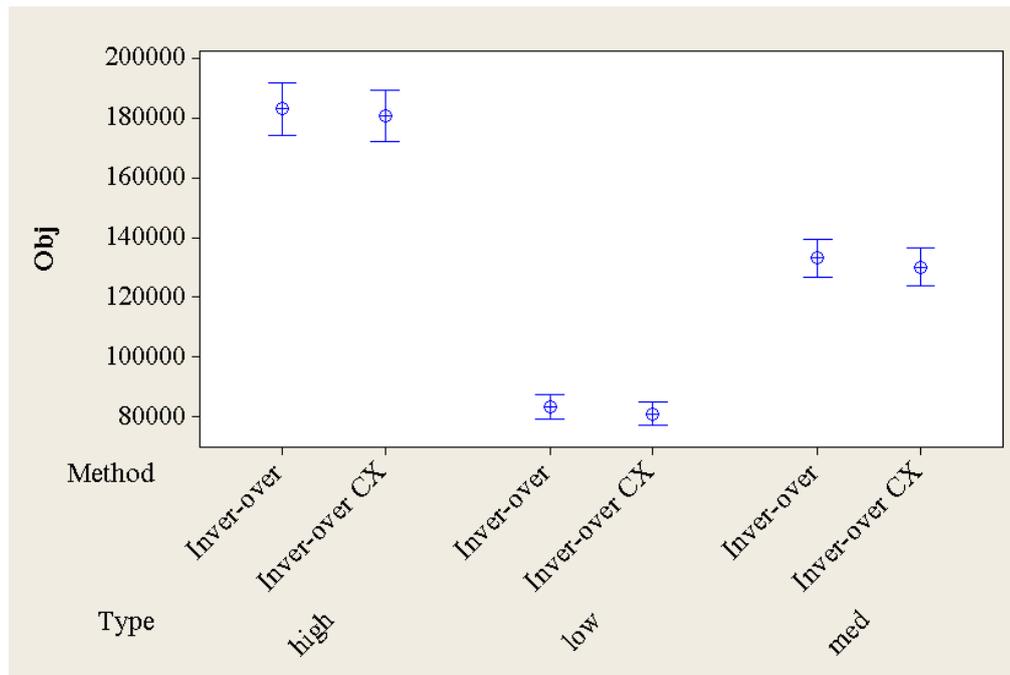


圖 15 方法比較區間表

表 18、表 19 為本研究方法及比較方法的執行時間 CPU Time ANOVA 表，由表 18、表 19 中 Method 的 P 值都小於 0.05 可得知在問題大小 10 至 25 及 50 至 200，執行時間 CPU Time 的比較會有顯著的因子，表 18、表 19 來源的 Size 為問題大小，如 10、15、20、25、50、100、150、200，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 18、表 19。

表 18 方法比較小問題 CPU Time 的 ANOVA 表

來源	自由度	Seq SS	Adj SS	Adj MS	F	P
Method	1	0.0187151	0.0187151	0.0187151	725.31	0.000
Size	3	0.007691	0.007691	0.002564	9.94	0.000
Type	2	0.00124	0.00124	0.00062	193.48	0.000
Method* Size	3	0.0000256	0.0000256	0.0000085	0.33	0.803

Method* Type	2	0.0001510	0.0001510	0.0000755	2.93	0.054
Type*Size	6	0.0075113	0.0075113	0.0012519	48.52	0.000
Method*Type*Size	6	0.0000512	0.0000512	0.0000085	0.33	0.921
誤差	10776	0.2780522	0.2780522	0.0000258		
合計	10799	0.0000512				

P < 0.05 為顯著

表 19 方法比較大問題 CPU Time 的 ANOVA 表

來源	自由度	Seq SS	Adj SS	Adj MS	F	P
Method	1	0.775954	0.775954	0.775954	15303.62	0.000
Size	3	0.790836	0.790836	0.263612	5199.04	0.000
Type	2	0.039859	0.039859	0.019930	393.06	0.000
Method* Size	3	0.000014	0.000014	0.000005	0.09	0.0965
Method* Type	2	0.000140	0.000140	0.000070	1.38	0.0251
Type*Size	6	0.028690	0.028690	0.004782	94.31	0.000
Method*Type*Size	6	0.000028	0.000028	0.000005	0.09	0.997
誤差	10776	0.546386	0.546386	0.000051		
合計	10799	2.181907				

P < 0.05 為顯著

由表 18、表 19 得知 CPU Time 的比較有因子是顯著的後，所以本研究將在事後檢定的部份將採用 Bonferroni 檢定來將兩組方法的執行時間 CPU Time 分組並分析，進而得到最為顯著的執行時間 CPU Time 即為最為顯著的方法。表 20、表 21 為兩組方法經由事後檢定 Bonferroni 後所產生的表，由表 20、表 21 可得知在問題大小 10 至 25 及 50 至 200 時將方法進行分組並設定可信赖度為 95%，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事後檢定 Bonferroni 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，詳見表 20、21。

表 20 CPU Time 小問題 Bonferroni 檢定表

Bonferroni 檢定			
方法	N	均值	分組
Inver-over	5400	0.016	A
Inver-over CX(控制)	5400	0.012	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

表 21 CPU Time 大問題 Bonferroni 檢定表

Bonferroni 檢定			
方法	N	均值	分組
Inver-over	5400	0.048	A
Inver-over CX(控制)	5400	0.031	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

圖 16、圖 17 為方法 Inver-over CX 及比較方法 Inver-over 在問題大小 10 至 25 及 50 至 200 執行時間 CPU Time 的區間圖，圖 16、圖 17 的 X 軸的 Method 為方法及問題的種類 Type，Y 軸為執行時間 Time，在圖 16、圖 17 可以很明顯的看出，不管問題的種類是 low、med、high 都是本研究之方法 Inver-over CX 優於 Inver-over，這也表示在解決單機排程之動態抵達時間的問題時，本研究之新方法 Inver-over CX 能有效縮短單機排程問題之時間，且是在考慮動態抵達時間的狀況下，而 Inver-over 屬啟發式演算法，特性為收斂速度快，但相較於加入 PMX 後，PMX 的程式執行步驟較為簡易且在門檻值篩

選時，本研究之方法 Inver-over CX 門檻值為 0.5，這意味著 Inver-over 及 PMX 出現機率各半，相較於 Inver-over 門檻值為 0.1，主要都以 Inver-over 為主，固本研究之方法的 CPU time 較快。

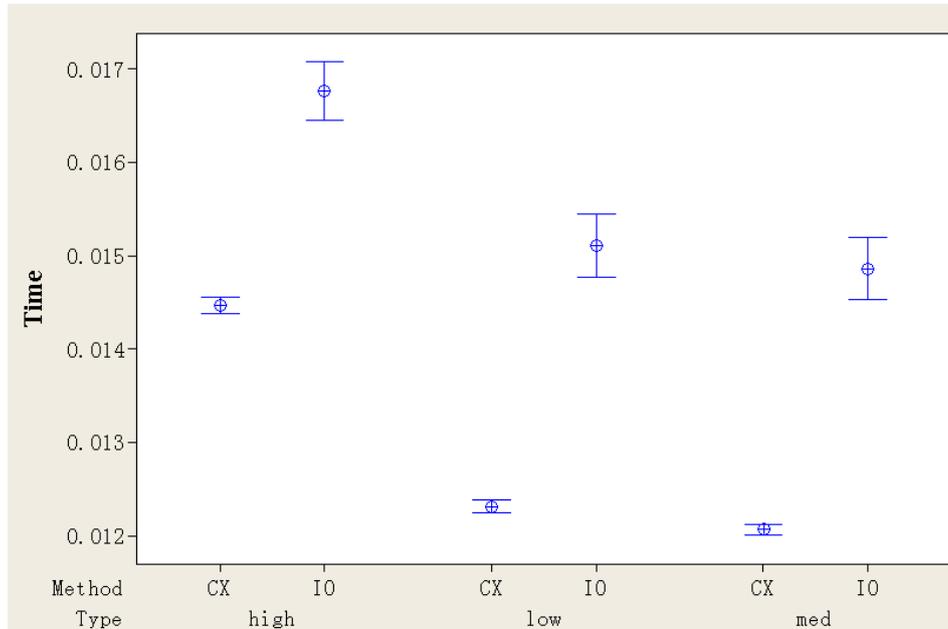


圖 16 CPU Time 小問題區間圖

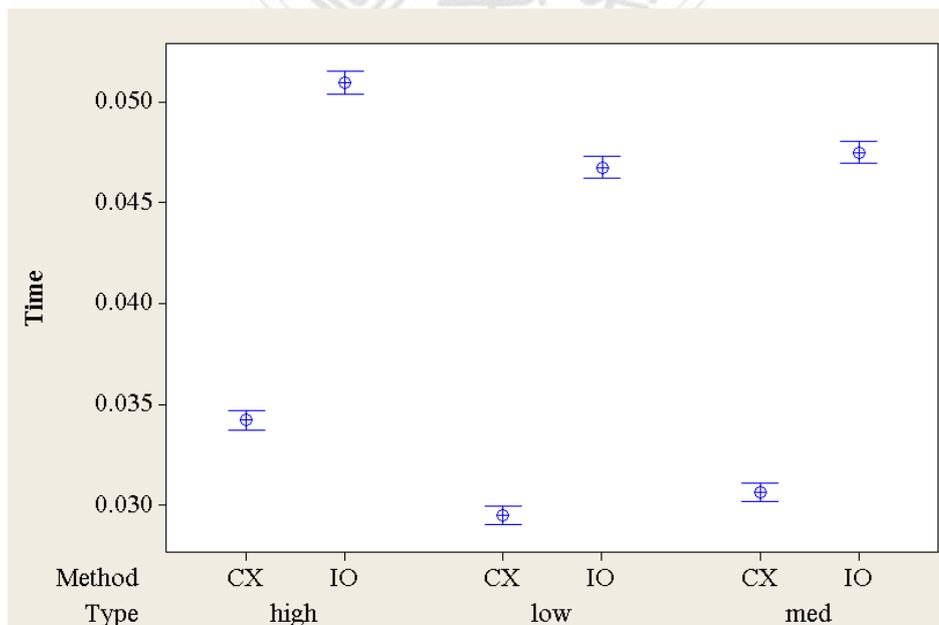


圖 17 CPU Time 大問題區間圖

經由第一節的參數篩選及第二節的演算法比較，表 22 為統整第一、第二節的方法比較表，表中的 Size 為問題大小，Type 為問題種類，Min 為目標函數的平均最小值，Avg 為目標函數的平均值，Max 為目標函數的平均最大值，Best Found 為本實驗將問題大小的小問題世代數改為 100000，大問題改為世代數 200000 的優良值，詳見表 22。

表 22 方法比較表

		Inver-over CX				Inver-over		
Size	Type	Best Found	Min	Avg	Max	Min	Avg	Max
10	low	452	592	649	708	785	852	912
	med	608	628	801	888	896	1000	1132
	high	885	924	1075	1270	1116	1259	1464
15	low	1173	1258	1345	1512	1403	1517	1685
	med	1742	1792	1874	2139	1807	2126	2363
	high	2342	2442	2517	2898	2481	2501	3062
20	low	2237	2359	2476	2535	2451	2605	2811
	med	3537	3668	3882	3948	3687	3949	4304
	high	4896	4930	5025	5384	5022	5250	6062
25	low	3617	3816	3953	4149	3917	4110	4418
	med	5710	5977	6071	6362	6041	6200	6725
	high	8101	8234	8521	8923	8624	8825	9331
50	low	13344	13756	15224	16157	19478	20306	21171
	med	22567	24064	25939	27368	29758	31462	33000
	high	29931	32908	37267	39893	38770	42172	45023
100	low	54720	76097	79656	82502	80668	84829	88213
	med	96291	122982	129132	133504	124279	135249	140038
	high	137739	175521	180916	187285	168484	186211	194466
150	low	137719	186418	192654	199777	193030	196825	200844

	med	248994	300750	310933	320150	293114	313756	329006
	high	357821	419430	428912	439554	415209	431204	44502
	low	238804	334742	350366	358778	343789	353416	363418
200	med	502544	524042	561107	583614	552953	570057	587465
	high	605732	748853	781387	795890	759919	785526	799641

表 23、表 24 為本研究方法 Inver-over CX 及比較方法 Inver-over 的執行時間比較表，表中的 Size 為問題大小，Type 為問題種類，Time 為方法執行的時間，Avg Time 為將問題大小分為大問題及小問題進而的平均，詳見表 23、表 24。

表 23 問題大小 10、15、20、25 的執行時間比較表

Inver-over CX			Inver-over		
Size	Type	Time	Size	Type	Time
10	low	0.010	10	low	0.014
	med	0.011		med	0.014
	high	0.015		high	0.017
15	low	0.011	15	low	0.015
	med	0.011		med	0.016
	high	0.015		high	0.017
20	low	0.013	20	low	0.015
	med	0.012		med	0.015
	high	0.013		high	0.019
25	low	0.013	25	low	0.016
	med	0.013		med	0.017
	high	0.013		high	0.017
Avg Time :		0.012			0.016

表 24 問題大小 50、100、150、200 的執行值間比較表

Inver-over CX			Inver-over		
Size	Type	Time	Size	Type	Time
50	low	0.016	50	low	0.033
	med	0.019		med	0.036
	high	0.027		high	0.044
100	low	0.025	100	low	0.042
	med	0.026		med	0.043
	high	0.027		high	0.044
150	low	0.033	150	low	0.05
	med	0.033		med	0.05
	high	0.035		high	0.055
200	low	0.042	200	low	0.059
	med	0.043		med	0.060
	high	0.045		high	0.062
Avg time :		0.031			0.048

第五章、 結論與建議

經過第四章的實驗設計與分析之後，在接下來的第五章將提出結論以及貢獻，第一節針對本論文做整體的結論說明，最後在第二節將對未來的研究提出一些建議。

第一節 結論

單機排程(Single Machine)已是目前大多數企業及物流業最常遇到的問題，在過去解決單機排程的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，所以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，因此工件的順序將會影響以動態抵達時間為考量的目標解，所以本研究將解決動態抵達為研究目標，並利用兩點交配之 Inver-Over 演算法為解決方法之基礎架構，並改善 Inver-Over 演算法的缺失，故在學術研究上，這是一項新的議題。

研究方法及實驗設計部分本研究以 Inver-Over 演算法結合 PMX 為本研究的新方法命名為 Inver-over CX，其中的基礎包含基因演算法的特性，目的為讓工件順序比喻成染色體，且進行交配、突變、選擇等方式，進而產生更好的順序解，本研究在過程中將小問題為 1000 個世代數為一次實驗為一次實驗，大問題為 2000 個世代數為一次實驗，並以總早到時間、總延遲時間最小為目標並做為工件順序最佳解的條件，接著就是作實驗的設計，本研究的測試問題為單機排程的 low、med、high 的三種問題去作實驗設計，其中抵達時間是以各個問題的處理時間為範圍去隨機產生的，並將其固定，完成這樣後就可以開始作實驗，而實驗的比較對象為未經改善的 Inver-over 演算法。

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。

第二節 建議

本研究所提出的是針對多單機排程(Single Machine)與動態抵達時間的解決方法，所以在 Inver-over CX 的設定上是以工件的總延遲時間最小做為目標函數值，然而要解決單機排程還有很多種方法，所以對於未來的研究，本論文建議在方法上還可以加入更多的門檻值的比較來因應現實的狀況，希望未來的研究可以提高工件的擾亂程度，建構出更好的工件順序。在實務應用方面可以運用本研究所提出的新方法 Inver-over CX 建置有關於單機排程的相關系統，希望可以有效的對物流業及運輸業等等行業帶來實際上的幫助。

參考文獻

中文部分：

1. 尤金亭、唐恒永，一類具有維護和共同交期的單機排程問題，瀋陽師範大學 數學與系統科學學院，2009。
2. 白健志，利用分組基因遺傳演算法解決多旅行推銷員的問題，南華大學資訊管理學系，2010。
3. 余宣慧，具有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。
4. 李日春，模糊-基因演算法於單機排程之研究，國立台灣科技大學工業管理系，2007。
5. 汪祐如，考慮提早成本、延遲成本、運送成本下的單機排程問題，國立雲林科技大學工業工程與管理研究所，2006。
6. 林勇，使用基因演算法求解旅行銷售員問題於自走車路徑規劃之應用實現，國立中央大學電機工程學系在職專班，2013。
7. 林豐澤，基因演算法及三種應用實例，文化大學應用數學系，2005。
8. 施智懷，具動態權重之混合基因演算法求解順序相依整備時間下單機排程問題之研究，華梵大學資訊管理研究所碩士論文，2004。
9. 洪國庭，多重代理人系統在有限資源下之多專案排程研究，東海大學工業工程與經營資訊研究所，2007。
10. 胡哲維，以優先權選擇的混合式基因演算法解決旅行家問題，國立中山大學資訊工程學系，2007。
11. 高清貴，考量整備時間與工件動態到達之單機延遲時間分析模型，國立交通大學工業工程與管理系所，2011。

12. 張旭翔，以礦工基因演算法求解單機排程問題，國立台灣科技大學工業管理系，2013。
13. 張昱敬，應用動態規劃演算法於單機排程流工廠之研究，國立台北科技大學商業自動化與管理研究所，2005。
14. 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。
15. 陳柏志，以礦工基因演算法求解旅行商問題，國立台灣科技大學工業管理系，2012。
16. 陳茂生，多群組工作之生產排程問題，國立清華大學工業工程管理學系，2003。
17. 彭武年，建構不同交期下肢先進規劃系統，南華大學管理科學研究所，2006。
18. 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。
19. 黃祥熙，排程理論，國立屏東科技大學工業管理系暨碩士班，2009。
20. 葉玉玲、許洲榮、蔡碧芳，相依整備時間考量下具等效佇行機台之多階段流程型排程問題啟發式求解模式建構，技術學刊第 20 卷第 3 期，頁297-304，2005。
21. 鄒佳臻，粒子群演算法結合低閒置策略於生產排程之應用，中原大學資訊管理研究所，2009。
22. 熊詩敏，結合優勢性質與基因遺傳演算法於具有整備時間之單機與非等效佇行機台之研究，元智大學工業工程與管理學系，2007。

23. 趙文涼，基因演算法於單機交期絕對偏差及整備成本最小化排程問題之應用，元智大學工業工程研究所碩士論文，2000。
24. 劉靜、閔嘯，一個延遲工件的單機準時生產制問題，嘉興學院數學與訊息工程學院，2010。
25. 蔡志弘，零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。
26. 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。
27. 鄭彬彬、王文義，基於Inver-over算子的並行TSP演算法研究，中原工學院，2009。
28. 羅久哲，具工件釋放時間限制下最小化提早與延遲成本之單機排程問題，義守大學工業管理研究所，2009。

西文部分：

1. Armentano, V.A. and Mazzini, R. "A genetic algorithm for scheduling on a single machine with set-up times and due dates," *Production Planning and Control*, 11, pp.713-720, 2000.
2. Azizoglu, M. and Webster S. "Scheduling job families about an unrestricted common due date on a single machine," *International Journal of Production Research*, pp.35, 1321–1330, 1997.
3. Carter, A.E. and Ragsdale, C.T. "A new approach to solving the multiple traveling salesperson problem using genetic algorithms," *College of Information Science and Technology, Radford University, Department of Business Information Technology, Pamplin College of Business, Virginia Tech, Blacksburg, 2003.*
4. Chang, P. C. and Hsieh, J. C. and Hsiao, C. H. "Application of Genetic Algorithm to the Unrelated Parallel Machine Problem Scheduling," *Journal of the Chinese Institute of Industrial Engineering*, 19(2), pp.79-95, 2002.
5. Chen, D. and Li, S. and Tang, G. "Single machine scheduling with common due date assignment in a group technology environment," *Mathematical and Computer Modelling*, pp.25, 81–90, 1997.
6. Emmons, h. "Scheduling to a common due date on parallel uniform processors," *Naval Research Logistics*, Vol.34, pp.803-810, 1994.
7. Ernesto, G. Birgin and Débora, P. "Heuristic methods for the single machine scheduling problem with different ready times and a common due date," *Engineering Optimization* Volume 44, Issue 10, pp. 1197-1208, 2012.
8. Tao, G. and Michalewicz, Z. "Inver-over operator for the TSP".1998.
9. 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.
10. 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.
 11. Holland, J.H “Adaptation in Natural and Artificial System” Oxford, England: U Michigan Press.1975.
 12. 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.
 13. 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.
 14. Fatih, M. and Buyukdagli, O. and Kızılay, D. and Karabulut, K. “A Populated Iterated Greedy Algorithm with Inver-Over Operator for Traveling Salesman Problem ,” *Lecture Notes in Computer Science Volume 8297*, 2013, pp 1-12, Pages 652–662, 2014.
 15. 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.
 16. Potts, C.N. and Wassenhove, L.N “Decomposition algorithm for the single machine total tardiness problem,” *Operation Research Letters*, 1, pp.177–181,1982.
 17. Rajabinasab, A. and Mansour, S. “Dynamic flexible job shop scheduling with alternativeprocess plans: an agent-based approach”, *Int J Adv Manuf Technol*, pp.1091-1107,2011.

18. Selmer, J.” Optimal two- and three-stage production schedules with setup times included” Naval Research Logistics Quarterly 1, 61-68, 1954.
19. 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
20. Tanaka, S. and Araki, M. “An exact algorithm for the single-machine total weighted tardiness problem with sequence-dependent setup times,” Volume 40, Issue 1, pp.344-352, 2013.
21. Cook, S.A. “The Complexity of Theorem-Proving Procedures”, ACM press, 1971.
22. Syswerdra, G. Uniform Crossover in Genetic Algorithms, Proceedings of the Third International Conference in Genetic Algorithms, Morgan Kaufman, Publishers, San Mateo,1989.
23. Wang, C. S. and Uzsoy, R. “A genetic algorithm to minimize maximum lateness on a batch processing machine,” Computers & Operations Research, 29, pp.1621-1640, 2002.
24. Yunqiang, Y. and Cheng, T.C.E. and Cheng, S.R. and Wu, C.C. “Single-machine batch delivery scheduling with an assignable common due date and controllable processing times,” Computers & Industrial Engineering Volume 65, Issue 4, August 2013, Pages 652–662, 2013.