

以改良鄰近搜尋之TABU應用於JSP問題求解之研究

余仁朋

銘傳大學資訊管理學系
助理教授
jpyu@mail.mcu.edu.tw

朱美珍

銘傳大學資訊管理學系
副教授
mcchu@mail.mcu.edu.tw

陳柏鈞

銘傳大學資訊管理學系
碩士生
aa932587@gmail.com

摘要

企業追求著多項目標而營運，對於資源分配與排程規劃常常受限於資源必須最大化利用且成本精簡至最小化。因此，管理者在執行排程規劃時必須更加謹慎，零工式排程的延伸問題正好是企業問題的延伸，更是多數學者探討的議題。

本研究提出一針對零工式排程型態為主的求解方法，使用禁忌搜尋法進行改良其鄰居搜尋的方式，來達到優異的搜尋求解功能。藉由此一改良方法使得禁忌搜尋法的搜尋空間更多樣化與更廣闊，進而適時的收斂達到全域問題求解的目的；而目標函式則透過 MTNS 這一個方法修改禁忌搜尋法在鄰居搜尋階段地操作，來使排程規劃耗費時間最小化，同時透過 job shop 的各類標竿問題進行驗證，比較本研究提出的方法與求解模式在此一問題所帶來的效益是有所貢獻的。

關鍵詞：禁忌搜尋法、零工式生產排程、鄰居結構。

1. 前言

全球化的時代來臨，企業的經營也隨著資訊科技的發展，相繼導入新的技術來提昇企業的競爭能力。例如供應鏈管理(SCM)與企業資源規劃(ERP)的導入其目的不外乎是提昇公司產品開發與縮短製程時間，進而使得公司可以在外環境變化上迅速應變，以展現出企業彈性的能力。另外，在生產環境的型態也與過去截然不同，過去製造業基本生產模式是由一個或是多個生產線同時生產單一類型的產品；現今的生產型態則因應快速變遷的環境，

由過去的大量生產型態轉變為少量生產模式，同時對過去單一產品生產的方式改變為多樣化產品的生產模式。環境的變遷，客製化且多元化的產品需求促使製造業者生產模式也隨著需求的型態有所轉變，因此，零工式生產排成問題(Job Shop Scheduling problem; JSP)更是學術與實務相當熱門探討的議題。

排程問題在過去數十年來研究方向較著重於單一排程目標為主；而現今的製造業環境中將實務需求因素考量於中，因此通常都不會僅是單一的排程目標，大都以多目標的考量為主。然而這些多重目標彼此之間都會有所干擾，這干擾將會大大影響到排程的流暢性。對企業的經營管理者而言，多目標因素考量的確提升了生產效率，但是對於公司經營方針、上游廠商之協調與顧客需求等行銷經營因素，在先前提升了效率但是不符合市場行銷，在營運上就變的不協調，因此問題零工式生產排程問題(Job Shop Scheduling problem)越顯重要。

由此可見怎樣的在有限資源下，讓工廠可以在合理的時間內，安排所有的工作與機器的運作排程，規劃出合理可行的一套生產計劃，並可以讓人期待在時間內保證可以生產出所需的產品，如此可以節省下的成本和時間都是公司經營上的目標。

學術上所探討的排程問題都與實際的工廠排程環境略有差異。譬如像是測試例子中的所有機器工作頻率都相同或是工作量都相同或是測試案例太小。然而實際的案例中，卻有某些機器是相較於其它機器來說負擔較多的工作頻率且不管其它機器處理的速度再快，依舊要等待瓶頸機器處理完才能繼續加工，我們稱這些機器

為瓶頸機器(Bottleneck Machine),或是工作
工作量不一致。因此本研究就是要藉由
MTNS 方法解決動態 JSP 這類的問題,計
算排程在合理的時間裡完成生產流程的工
作,以準時達到交貨的目的,並在求解各
種大大小小的標竿問題(benchmark
problem)可以獲得接近目前最佳解(optimal
solution)或是找出最佳解,來證明本研究改
良的方法在尋找最佳解的績效。

針對本研究目建立目標式建立起數學
模型,並用 MTNS 方法在合理的時間內找
出近似最佳解,且根據文獻中前人所研究
的標竿問題所得到的已知解答作為比較,
證明其解答的有效性,最後研究成果終將
導入實務研究實際運作,提供實際排程工
作所用為最終目的。

本研究提出的 MTNS 方法期望突破禁
忌搜尋法求解能力,以禁忌搜尋法為基
礎,在既有架構中,加入新的鄰居結構搜
尋方式而成為「改良鄰近搜尋之 TABU」,
用以求解零工式生產排程問題。在這樣
的排程加工環境中,如何安排工作之加工
順序,追求工作的完工時間最能符合最
小化總時間成本,以滿足企業降低成本與
滿足客戶需求的目標,並顧及實務應用
之時效性,以改良鄰近搜尋之 TABU 求
取近似最佳解。

2. 文獻回顧

2.1 零工式生產排程

1965 年 Conway 提到當一群工作(jobs)
其執行順序需要被決定時,所謂的排程
問題(scheduling problem)便出現了,而
排程的結果則是建議那一工作使用哪一
個資源來執行。也就是說,一個排程決
定了何時(when)及何地(when)做什
麼事(what)(Parunak, 1991)。Michael
及 Xiuli (1999)指出生產現場的排程
問題若以機器數目配合工作劃分,一段
時間內,分配有限的資源,來處理一群
工作,以達到一個或多個的最佳目標。
可將排程問題分成數種不同複雜程度
的問題,依序分類如下:

1. 單機 (Single-machine) 問題: 各
訂單只有一道加工途程,且現場只有一
台加工機器。
2. 平行機台 (Parallel-machine) 問
題: 各訂單只有一道加工途程,且現場
只有多部加工相同功能機器。平行機
台問題可細分為等效平行機台問題與
非等效平行機台問題。等效平行機台
(Identical) 是指該加工站的機器的
處理加工效能相同,非等效平行機台
(Non-Identical) 則表示處理加工效
能存有差異,同一作業會因為指派到
不同機台加工而導致加工時間不同,
或良率不同。
3. 流程式生產 (Flow Shop) 問題:
各訂單有數道加工途程,所有各訂單
遵循相同的加工途程順序。
4. 零工式生產 (Job Shop) 問題:
各訂單有數道加工途程,所有各訂單
有其各自的加工途程順序。
5. 開放式生產 (Open Shop) 問題:
各訂單有數道加工途程,但加工途程
順序不受限制。

在這些種類中以 Job Shop 最為複雜,
是屬於 NP-HARD 問題,本研究在此討
論的主軸將以多階-零工式工廠這類
型的排程問題, Garey 等學者 (1976)、
Michael 及 Xiuli (1999)、Joseph (2004)
指出在 1970 年代開始,進行排程規
劃時,當問題規模過大時,如: 訂單數
、機器數、加工途程數等變數規模的
增加,排程規劃問題被認為 NP-hard
問題。而 NP-hard 問題,表示該問
題困難度極高,無法用多項式表示其
求解空間,即無法以多項式時間演算
法 (Polynomial time algorithms) 求
解出最佳解,若是以傳統的數理規劃
方式求解,則需耗費大量的時間;在現
實環境中,由於市場競爭日趨嚴重,
企業須滿足顧客需求、快速反應市場
變化,將資訊回饋到研發、製造生產
部門,追求產品多樣化、客製化,並
使其生產規劃具有彈性來因應變化
發生,才能強化自身企業的核心競爭
力。

一般求解 JSP 技術大致可分成三大
類 (Jain 等, 1998):

1. 派工法則(dispatching rule)

派工法則是最為流行的探索法(heuristics)(亦可稱之為單一階段探索法、排程法則或優先法則)，主要將工作的某些特徵(parameters)經過計算，作為指派工作的下一個操作(operation)的基礎。一般而言是容易理解且求解速度快，但是除了在少數特殊問題型態下，其求解績效並不是很理想，原因在於派工法則通常無法考慮到全部整個排程計劃(Herrmann, 2001)，多適用於單機問題的求解。常用的派工法則有SPT(shortest Processing Time)、LPT(longest Processing Time)、EDD(Earliest Job Due Date)等。

2.最佳化方法(optimization method)

最佳化方法求解主要是利用某些數學規劃上之技術或方法，如整數規劃法(integral programming)，來求得問題之最佳解。但普遍而言，以最佳化方法求解問題除了在構建上相當困難與複雜外，求解所花費的時間也是相當驚人的，因此，大部份學者均傾向於以啟發式方法來求得問題之近似最佳解，或僅以小樣本來探討最佳化方法之各方面研究。

3.近似解方法(approximation methods)

近似解方法求解主要是希望在有限的計算時間內，求得一近似最佳解的可行解，故近似解的方法可適用於大型的排程問題。另一是區域搜尋法(local search)，如基因演算法(genetic algorithm)、蟻群最佳化(Ant Colony Optimization, ACO)、模擬退火法(simulated annealing)、禁忌搜尋法(Tabu search)演算法及不同方法合併使用的方式來獲得排程的可行解，其求解績效亦相當理想。

2.2 派工法則

在生產過程中，決定從等候線上選取誰作為下一個加工的工作，此種行為稱為派工，而選取的準則稱為派工法則；派工法則可以在複雜的問題中迅速決定出一個可行之排程結果。因此，派工法則的選取，對於整體生產績效有顯著的影響，但是並沒有一種派工法則，可以普遍在不同的生產型態與績效衡量準則上，都能將資源做

最適合的分配並求出一組最佳解。所以必須針對各種不同的生產環境、不同的績效準則，去選取適當的派工法則，過去對於零工式生產排程問題發展出許多值得參考的派工法則，而將比較常見的派工法則整理如表 1 所示，而這些派工法則還可做相加或是相除動作，以產生新的派工法則。

表 1 派工法則簡介

派工法則	說明
MWKR(Most Work Remaining) 最多剩餘加工時間法	等候線中剩餘最多加工時間的工作優先處理
SPT(Shortest Processing Time) 最短加工時間法	對於加工時間最短的工作為優先
SPT/TWKR (Shortest Processing Time/Total Work Remaining) 加工時間/總剩餘加工時間法	加工時間除以工作總剩餘加工時間最短的優先處理
SRPT(Shortest Remaining Processing Time)最短剩餘加工時間法	最短剩餘加工總時間者優先

資料來源:本研究整理

故派工法則的好壞評估都有密切關係，在選擇使用時，皆應仔細謹慎評估。此外，因為派工可改變加工順序，故派工法則的優劣會直接影響生產週期時間的績效。為了減少非預期性之因素與動態事件對系統造成之影響，就必須在生產系統中選擇適合的派工方法。派工法則若依據順序安排則會隨著以下五項條件而有很大的差異：

1. 工作數目。
2. 需經加工之機器數目。
3. 設備佈置型態。
4. 工作到達之方式。
5. 評估之標準。

由於派工法則種類繁多，因此我們挑數個來做簡單的介紹，雖然派工法則能快速的找出可行的解，但是所有的派工法則都有

共通的缺點，即是找出來的解只是個可行解，其解的排程效率往往到達不了理想的結果。因為每個工廠訂單到達型態不同，所以其適用的派工法則也不盡相同，一般會選用不同的決策準則將使派工法則之適用性改變。

2.3 禁忌搜尋法

禁忌搜尋法 (Tabu search) 是 Fred Glover 於 1986 年所提出並應用在整數規劃上，禁忌搜尋法屬於區域搜尋的一種經驗法，通常可以有效快速地收斂到最佳解，而區域最佳解可能形成的陷阱，禁忌搜尋法則是透過禁忌列表 (Tabu list) 以避免徘徊於某區域的最佳解而得以展現探索其他區域的能力，也就是說此法兼具了彈性的搜尋與搜尋效率。因為此方法簡單有效，所以目前已經被廣泛應用於各類型的問題上，例如車輛途程問題、排程與旅行銷售員問題等。

關於禁忌搜尋法的方法，一開始會先建立一個初始解，接著找尋最優的鄰近 (居) 解或是符合凌駕規則的解作為移步的依據，也就是在現行解的鄰近區域搜尋最佳解，其中有一個重要的特色就是禁忌名單 (Tabu list) 的記憶機制，目的在於記錄拜訪過的解避免重複搜尋所導致的效能遞減與浪費，一旦遍覽所有的鄰近區域之後，則會選擇一個最佳的方向來進行現行解的變更，若出現比目前最佳解還要好的解，則會更新目前最佳解，以此不斷的重複動作，直到符合結束規則才停止。

禁忌搜尋法與爬山法 (hill climbing) 最大的差別是可以接受比目前最佳解還差的解，這樣的方式可以讓搜尋跳出區域最佳解而得以擁有更寬廣的搜尋區間。基本上 Glover 的禁忌搜尋法中，指出禁忌搜尋法主要的組成要素大致上可以分為六個概念模組：起始解 (starting solution)、鄰域解 (neighbor solution)、移動路徑 (move)、記憶架構 (memory structure)、凌駕準則 (Aspiration criterion)、停止條件 (stop criterion) 分述如下：

1. 起始解：由最原始的可行解中，產生一

啟發解作為搜尋的起始點。

2. 鄰域解：不屬於記憶架構內的紀錄，並符合問題限制之可行解。而鄰域解的移動取法有兩種交換式變動 (swap move) 與插入式變動 (insert move)，因為一個插入式變動是由兩次交換式變動所造成，所以插入式變動對目前的解所造成的變動程度較交換式高，但必須視問題的型態作最佳的選取。
3. 移動路徑：在目前的鄰域解中，選取評估值最高者作為改善之移動路徑，若此鄰域解已在禁忌串列中，且又不符合凌駕準則選取次佳之鄰域解作為改善移動的路徑；若此鄰域解符合凌駕準則，則其不受禁忌串列的限制。
4. 記憶架構：記憶結構亦可稱為禁忌名單 (tabu list)，記憶結構又可分為「短期記憶」與「長期記憶」，短期的架構只記錄該次搜尋前的數次記憶，長期記憶則是記錄從開始運作以來所有的搜尋過程。記憶期數的大小必須依照問題的特性和複雜度決定，Glover (1990) 建議可使用魔術數字 7 作為禁忌串列的尺寸。
5. 凌駕準則：考慮一鄰域解，若此一移動所得之鄰域解之評估值比截至目前為止搜尋過的最佳解還要好，若其恰巧在禁忌串列的限制中，此時應破除禁忌串列的限制，接受此一移動，以使得被禁區限制住的鄰近解有第二次被選擇的機會。例如：當候選的鄰近解被禁區限制住時，但此鄰近解比目前最佳解還要好，此時禁區便會被打破，而選擇了此移步。
6. 停止條件：終止搜尋進行的條件，一般最常是以
 - (1) 預設可允許之最大搜尋次數。
 - (2) 預設目標值持續未改善之次數。
 - (3) 預設最長的 CPU 執行時間。
 - (4) 預設可接受之目標函數值。

一旦搜尋達到這些預設值則停止搜尋，取當時最好的解作為最終的最佳解。整體來說，由於禁區搜尋法是利用彈性的記憶架構來禁止某些移步，不允許再次拜

訪記錄在禁忌列表中的移步，我們稱為「禁區」，以防止又回到曾經搜尋過的解，在一定時間後這些移步便會被釋放。而有效率的禁區搜尋法必須在強化性與多樣性中取得平衡。換言之，就是要在「搜尋解的效率」與「搜尋的範圍」間取得平衡。

Wesley (1995) 以 14 種派工法則求出起始解，再利用禁忌搜尋法求解 Job shop 排程問題。Adenso (1996) 利用禁忌搜尋法結合模擬退火法求解流程式排程問題，利用模擬退火法中接受較差解的機制，避免求解過程中掉入局部最佳解的情形。Liaw (2000) 的研究中利用禁忌搜尋法結合基因演算法求解流程式排程問題，以最小化製距為目標，然後利用 Tabu 搜尋法對每一個體解其關鍵路徑 (critical path) 進行鄰域搜尋修正改善。Scrichet al. (2004) 在彈性零工式生產排程問題上，以最小延遲時間為目標，先以派工法則先求出一初始解，再以禁忌搜尋法進行鄰域搜尋，並驗證出此求解機制具有高效率的求解品質。Leyuan 及 Yunpeng (2005) 以禁忌搜尋法為基礎，針對零工式生產問題設計出更具效率的演算方法。Saidi (2007) 利用禁忌搜尋法在彈性零工式生產排程問題中，以最短製距為績效考量，成功求解出最佳的作業排序與作業加工機器指派。在 1995 年 Kolahan 以增大禁區列表長度對解的改善是否有所幫助為目標，針對求解自動化加工中心混合零件排序和工具更換問題，其研究結果發現在相同搜尋時間下，發現成效不明顯。在 1997 年 Wen 及 Yeh 研究中將禁區列表固定為 7，並以兩種初始解結合兩兩交換 (pair interchange) 及鄰近點交換 (adjacent interchange) 為移步方式進行探討，應用於以最大總完工時間最小化為目標之流程型工廠排程問題，結果都能有滿意之效果。

3. 問題定義

典型的零工式排程問題 (Job Shop Scheduling Problems, JSP) 為對 N 個工作及 M 部機器的排程問題，其中每個工作包含

特定數量的作業 (operation or activity)，且各作業間有固定的先後順序關係，每項作業的處理時間和必須要在那部機器上處理都是事先決定且固定。有 N 個工作由 M 個機器來進行操作，而每個工作皆要完成數個操作，每個操作則需要不同的機器來完成或某工作只能由某機器來完成，同一時間裡一部機器只能進行一個操作，並使成本目標函數可以在最短時間內完工，在考慮各個操作時間的優先限制下來完成最短的工作時間與消耗的資源成本。

問題假設之條件

我們在求解 JSP 問題所追求的目標函式，最小化的花費時間。min(Makespan) 求解的過程中會因不同條件限制而有不同的解答，所以 JSSP 問題有以下這些特性與條件：

1. 每個步驟在進行過程中不能被中斷。
2. 每個工作的執行順序不能被違反，對同一個工作來說，每一個步驟必須等到前面的步驟完成後才能開始。
3. 每一個工作只能去同一個機器一次。
4. 同一個工作在一個時間內，只能有一個步驟在機器上執行。
5. 同一台機器在同一個時間內只能執行一個工作步驟。
6. 每個作業所需工時都是已知且不變的正整數。

依據上述假設條件本研究建構問題的數學模型如下：

$$\text{Makespan} = \text{Min}(Z)$$

$$Z = \sum_{j=1}^n \sum_{i=1}^m R_{ij} + \sum_{i=1}^m M_i$$

其中 Z = 花費時間。

R_{ij} = 實際工作時間

M_i = 機器閒置時間

n: 工作總數

m: 機器總數

$i = 1, 2, \dots, m, \quad j = 1, 2, \dots, n$

4. MTNS 方法之建構

Modified Tabu Neighborhood Search (MTNS)，主要是修改禁忌搜尋法在鄰居搜尋階段時候的一個新方法，利用這麼一個方法來搜尋鄰居結構，產生可行的鄰居解是為了達到搜尋的範圍更加廣闊，不易於屈就在局部搜尋而只能尋得區域最佳解。MTNS 方法流程圖如圖 1 所示。

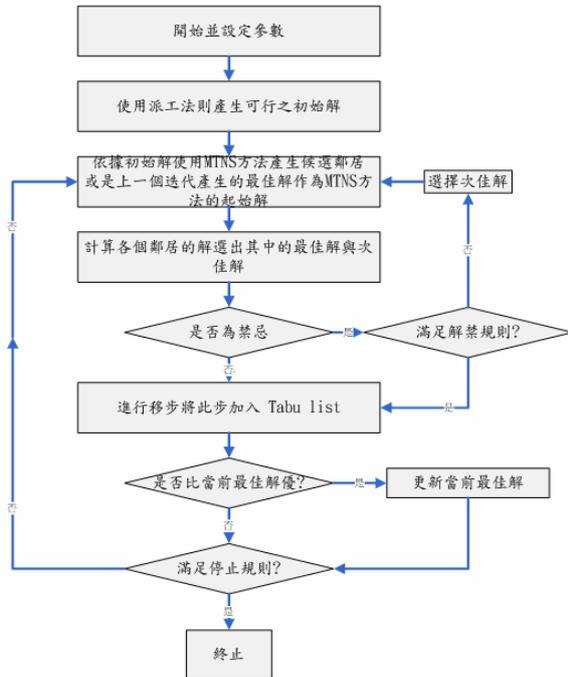


圖 1 Modified Tabu Neighborhood Search 流程圖

4.1 初始化起始解

最初的解決方案可以通過各種方法產生，如派工法則，各種不同的法則，如 MWKR、SPT/TWKR。在最初產生的解決方法一般來說影響後來演化所提供的解決方案的質量是沒有很大的差異，但是這些起始解的產生它們會影響下一階段運行時間。依據表 2 測試數據本研究採用 MWKR 這法則作為起始解。

表 2 派工法則 Makespan

派工法則	MWKR	SPT	SPT/TWKR	SRPT	隨機
MT06	*66	109	68	154	94
MT10	*1447	2648	2138	3180	1819
MT20	*1960	2762	2565	2941	2158

LA01	*878	1462	1121	1959	1204
LA06	*1066	2367	1201	3118	1549
LA11	*1522	3164	1814	4038	2070
LA16	*1333	3456	2007	3838	1846
LA21	*1671	4361	2342	6165	2518
LA26	*1887	6857	2715	8240	2853
LA31	*2753	8061	4083	11908	3677
LA36	*1898	6899	3769	9694	2937

資料來源:本研究整理

4.2 產生新鄰居的方式

鄰居結構是一個取得新的一套解決方案的一個機制，透過一個鄰居變化的應用給訂一個新的解決方案，而在本研究中的問題裡，產生一套可行的解決方案必須依據問題所給定的限制來進行解決方案。假設如圖 2 選定位置 2 的 Job 0,2(工作 0 的第 2 個操作)它可以交換產生新鄰居的範圍右邊可以選位置 3 與 4 左邊則沒有位置可以選擇，原因在於每個操作是不可逆的 Job 0,2 不可以在 Job 0,1 的前面所以只要是交換到 Job 0,1 前面的動作就不可行，另一點 Job 0,2 在向左或向右尋找交換位置時只要是任一相同的 Job 出現第二次就是找到交換的範圍了，例如位置 4 和 5 這邊 Job 1,1 與 Job 1,2 相同的 Job 出現第二次所以交換的位置就落在位置 4 這邊。

位置	0	1	2	3	4	5
Job & O	2,1	0,1	0,2	2,2	1,1	1,2

圖 2 鄰居交換合法範圍

除了使用鄰居結構，並且瀏覽整個鄰居結構，從第一個元素至最後一個，並利用上面的方法來找尋鄰居交換的合法範圍，同時結合關鍵路徑(Critical Path)來產生新的鄰居結構，找到的這些關鍵路徑在改善 Makespan 有很大的功效。

4.3 移動準則和演化

根據目前所找到的鄰近解，挑出比目前所在解更好的可行解，並且以此可行解

作為下一個迭代的起始解，若此鄰近解中沒有找到更佳的可行解，我們便退而求其次，挑選鄰近解中的次佳解作為下一代的起始解。因此判定移動與否的步驟如下：

- 步驟一：經過目標函式的計算本迭代挑出最佳。
- 步驟二：判別此鄰居解是否為禁忌鄰居解，是就前往禁忌列表，否則進行移動。
- 步驟三：更新目前最佳解，反之接受移動不更新最佳解。

4.4 禁忌列表

這個列表的目的是為了避免禁忌搜尋過程裡選擇解決方案時回頭重複選了前面已經出現過之相同的解決方案；做了禁忌列表這樣的一個動作，其主要目的是以這樣的一個動作來節省運算時的浪費，省去了可能的重覆計算節省時間與運算記憶體；這邊我們使用先進先出(FIFO, First In First Out)的方式來更新禁忌列表，因為記憶了搜尋過的解決方案可以避免掉入迴圈搜尋，而避開重複搜尋的這個時間拿來進行其他區域的搜尋這樣，如此以來便可以達到多樣性的搜尋。而本研究中所定義的禁忌列表長度為7，是依據文獻 Fred Glover 所建議而使用 7 這樣一個長度恰好在演算法的運行上較為順暢。圖 3 為禁忌列表示意圖且長度 $k=7$ 。

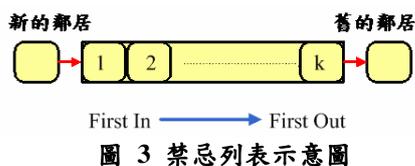


圖 3 禁忌列表示意圖

4.5 演化之終止條件

本研究將迭代次數設定再 1000 代，未改善次數達到最高時，或解決方案被證實為最佳的情況(最大完工時間小於等於已知的結果)則達到中止條件。已被證實的解決方案最佳情況如表 3 所示。

表 3 被證實已知最佳解

問題名稱	問題規模	已知最佳解
MT06	6x6	55
MT10	10x10	930
MT20	20x5	1165
LA01	10x5	666
LA06	15x5	926
LA11	20x5	1222
LA16	10x10	945
LA21	15x10	1046
LA26	20x10	1218
LA31	30x10	1784
LA36	15x15	1268

資料來源:本研究整理

4.6 敏感度分析

起始的參數設定對搜尋效率的影響與成功是有所關聯的。我們所設定的初始參數有迭代次數(iteration)、探索鄰居數(neighbor)、禁忌列表(tabu list)長度。我們將設計實驗來探討不同的參數設定值對於這些基準測試例題的影響，表 3 至表 6 是本研究用來測試參數設計不同時對不同題組的影響情況，試找出與問題組相對應的參數設計情況。

禁忌搜尋法在進行鄰居搜尋時，鄰居的個數決定其搜尋的量影響著瀏覽的效能；而禁忌列表長度用來記錄過去移動過的位置搜尋的歷史，一般而言禁忌列表越大則落入區域最佳解的機會將越低，有助於偏離這些區域跳脫局部最佳解，以及避免陷入迴圈，此長度越長記憶體空間需求也會增加，每次偵測耗費時間也多，所以我們將禁忌列表長度預設為長度 7 與文獻建議相同，迭代次數固定為 1000 次迭代。

這邊進行了數次的測試所得結果整理如下表 4 所示。

表 4 鄰居數測試

鄰居數	MT06	誤差值	MT10	誤差值	MT20	誤差值
5	55	0.00%	1042.4	10.78%	1224.4	4.85%

10	55	0.00%	1043.8	10.90%	1219.2	4.45%
20	55	0.00%	1044.4	10.95%	1223.4	4.77%
30	55	0.00%	1023.8	9.16%	1204	3.24%
鄰居數	LA01	誤差值	LA06	誤差值	LA11	誤差值
5	708	5.93%	926	0.00%	1222	0.00%
10	666	0.00%	926	0.00%	1222	0.00%
20	675	1.33%	926	0.00%	1222	0.00%
30	676	1.48%	926	0.00%	1222	0.00%
鄰居數	LA16	誤差值	LA21	誤差值	LA26	誤差值
5	991.6	4.70%	1097.4	4.68%	1227.6	0.78%
10	975	3.08%	1085.8	3.67%	1234	1.30%
20	987.6	4.31%	1084.2	3.52%	1237.6	1.58%
30	978.6	3.43%	1079.2	3.08%	1246.4	2.28%
鄰居數	LA31	誤差值	LA36	誤差值		
5	1784	0.00%	1341.8	5.50%		
10	1784	0.00%	1336.9	5.15%		
20	1784	0.00%	1358.8	6.68%		
30	1784	0.00%	1346.8	5.85%		

資料來源:本研究整理

從上表我們可以找出各個例題在各個鄰居數下所展現出來的結果，因此我們就採用各例題分析後誤差值最低的鄰居數做為我們參數設定的依據。

5. 實驗討論

關於本研究中求解的問題對象，是從文獻中挑選出來的 11 個例題分別是有不同大小規模的問題，下表 5 為本研究使用的標竿問題。

表 5 標竿問題資訊摘要

例題	SIZE	Opt	範例來源	
1	MT06	6x6	55	H. Fisher, G.L. Thompson
2	MT10	10x10	930	H. Fisher, G.L.

				Thompson
3	MT20	20x5	1165	H. Fisher, G.L. Thompson
4	LA01	10x5	666	S. Lawrence
5	LA06	15x5	926	S. Lawrence
6	LA11	20x5	1222	S. Lawrence
7	LA16	10x10	945	S. Lawrence
8	LA21	15x10	1046	S. Lawrence
9	LA26	20x10	1218	S. Lawrence
10	LA31	30x10	1784	S. Lawrence
11	LA36	15x15	1268	S. Lawrence

資料來源:本研究整理

5.1 MTNS 搜尋策略

鄰居(neighbor)數與禁忌列表(tabu list)長度皆為禁忌搜尋法的重要參數，其目的分別為增加搜尋空間與避免重複搜尋，因此其設定的策略往往關係著求解的最後結果，所以上一個章節的分析中給了我們實驗設定的一個方向。如下表 6 所示。

表 6 參數設計集合

問題名稱	迭代次數	鄰居個數	禁忌列表長度
MT06	1000	5	7
MT10	1000	30	7
MT20	1000	30	7
LA01	1000	10	7
LA06	1000	5	7
LA11	1000	5	7
LA16	1000	10	7
LA21	1000	30	7
LA26	1000	5	7
LA31	1000	5	7
LA36	1000	10	7

資料來源:本研究整理

5.2 效率比較

共有 11 個例題，並與文獻中的方法做比較像是 MGA 是由 L. Wang and D.Z. Zheng 學者提出的一個方法是一個在基因

演算法中的一個步驟加入了模擬退火法的概念，HTSACO 是由 V. P. Eswaramurthy & A. Tamilarasi 學者所提出，此方法是結合了蟻群最佳化與禁忌搜尋法的禁忌列表長度以動態方式呈現的策略。這些例題每個都進行了 100 個測試所平均出來的結果。而表 7 中有標示*則是本研究所提出方法在此例題求的最佳解的情況，標示--表示此處資料從缺。

表 7 計算結果比較表

問題名稱	問題規模	已知最佳解	MTNS 最佳解	MTNS 平均最佳解	MGA	HTSACO
MT06	6x6	55	*55	55	55	55
MT10	10x10	930	989	1051.8	930	930
MT20	20x5	1165	1178	1216	1165	--
LA01	10x5	666	*666	670.1	666	666
LA06	15x5	926	*926	926	926	926
LA11	20x5	1222	*1222	1222	1222	1222
LA16	10x10	945	947	985.4	945	945
LA21	15x10	1046	1053	1096.5	1058	1046
LA26	20x10	1218	*1218	1237.6	1218	1218
LA31	30x10	1784	*1784	1784	1784	1784
LA36	15x15	1268	1297	1336.9	1291	1268

資料來源:本研究整理

上表 7 是所有例題共 11 個的計算結果統計，在這邊所得到的求解結果可以看出我們的 MTNS 方法在 MT06、LA01、LA06、LA11、LA26 與 LA31 皆有求得例題的最佳解(解決方案)，而在其他例題上的表現則沒有預期的好。接下來看到平均最佳解是平均了 100 個測試所得結果，這邊有 MT06、LA06、LA11 與 LA31 這三個例題求的 100 個測試平均值與已知最佳解是一樣的，這表示本方法在這些例題上每次測試皆能求出最佳解。

這邊 MGA 在 LA21 與 LA36 也無法達到最佳解的情況，HTSACO 則是在所有例題上皆達成求得最佳解。

表 8 比較已知最佳解之誤差值

問題名稱	問題規模	已知最佳解	MTNS 最佳解	最佳解誤差值	MTNS 平均最佳解	平均最佳解誤差值
MT06	6x6	55	*55	0.00%	*55	0.00%
MT10	10x10	930	989	5.97%	1051.8	11.58%
MT20	20x5	1165	1178	1.10%	1216	4.19%
LA01	10x5	666	*666	0.00%	670.1	0.61%
LA06	15x5	926	*926	0.00%	*926	0.00%
LA11	20x5	1222	*1222	0.00%	*1222	0.00%
LA16	10x10	945	947	0.21%	985.4	4.10%
LA21	15x10	1046	1053	0.66%	1096.5	4.61%
LA26	20x10	1218	*1218	0.00%	1237.6	1.58%
LA31	30x10	1784	*1784	0.00%	*1784	0.00%
LA36	15x15	1268	1297	2.24%	1336.9	5.15%

資料來源:本研究整理

上表 8 中比較 MTNS 方法與已知最佳解的差距，這邊的誤差值是由%表示公式如下

$$\text{誤差值} = 1 - (\text{已知最佳解} / \text{求得最佳解})$$

在 MT06、LA01、LA06、LA11、LA26 與 LA31 中最佳解誤差皆為 0%，其中的 LA01 與 LA26 在平均的情況下誤差分別為 0.61%與 1.58%，顯示再這兩個例題上本方法還不能百分之百在每次的測試上階達到最佳的情況。而 MT 系列例題有兩個分別誤差值高達 1.1%與 5.97%，LA 系列例題有三個誤差分別為 0.21%、0.66%與 2.24%再這些有誤差的例題我們的方法仍需改進以適應這些例題並消除誤差。

MTNS、MGA 與 HTSACO 三個方法在求解上的誤差如下表 9 所示。HTSACO 在求解的精準度很高在所有的例題中每個測試例題均能達到最佳解誤差為 0%，MGA 而僅次於它，有兩個例題求解上有些許誤差，而本方法 MTNS 在求解例題只有半數能確實達到最佳解，但是誤差大部分皆在可以接受之範圍內。

表 9 效率誤差表

問題名稱	問題規模	已知最佳解	MTNS 最佳解誤差值	MTNS 平均最佳解誤差值	MGA	HTSACO
MT06	6x6	55	0.00%	0.00%	0.00%	0.00%
MT10	10x10	930	5.97%	11.58%	0.00%	0.00%
MT20	20x5	1165	1.10%	4.19%	0.00%	--
LA01	10x5	666	0.00%	0.61%	0.00%	0.00%
LA06	15x5	926	0.00%	0.00%	0.00%	0.00%
LA11	20x5	1222	0.00%	0.00%	0.00%	0.00%
LA16	10x10	945	0.21%	4.10%	0.00%	0.00%
LA21	15x10	1046	0.66%	4.61%	1.13%	0.00%
LA26	20x10	1218	0.00%	1.58%	0.00%	0.00%
LA31	30x10	1784	0.00%	0.00%	0.00%	0.00%
LA36	15x15	1268	2.24%	5.15%	1.78%	0.00%

資料來源:本研究整理

下表 10 中的執行時間比較的時間單位是秒(sec)表示各方法在執行上所耗費的時間，本研究在求解時間上有著很大的優勢，原因在於我們參數設定上總迭代數最大為 1000 次，與固定的鄰居拜訪數目 10 個，還有固定的禁忌列表長度 7，這些因素加上 MTNS 方法使得求解所耗費的時間有倍數的縮短，不管是求得最佳解的例題或是沒有求得的例題，在這樣的時間表現下本研究在某些例題上還是有所貢獻。

表 10 比較已知最佳解之誤差值

問題名稱	問題規模	MTNS 平均時間(sec)	HTSACO 平均時間(sec)	MGA 平均時間(sec)
MT06	6x6	0.15	2	13
MT10	10x10	85.1	489	292
MT20	20x5	85.1	--	204
LA01	10x5	21.9	54	37
LA06	15x5	1	98	99
LA11	20x5	1	113	197
LA16	10x10	49.4	158	232

LA21	15x10	124.3	1034	602
LA26	20x10	218	1956	1388
LA31	30x10	19.9	3597	3745
LA36	15x15	217.8	2983	1826

資料來源:本研究整理

6. 結論

在一個有效和有效率的禁忌搜尋下尋找出 JSP 問題的最佳解決方案，根據一個有效的鄰居結構與產生新的鄰居解的策略，而本研究所提出的 MTNS 方法是著重在用一個可行的鄰居結構經過 MTNS 產生新的鄰居解並且是可行的，因為要達到搜尋多樣化，所以在求解新的鄰居解時往往會在搜尋空間上大跳躍增加搜尋範圍，加入關鍵路徑在搜尋新解的時候有著兩個相鄰的關鍵路徑這兩個關鍵的地方做交換對於解的改善與跳動有著非常大的幫助，因此在本研究的數據分析顯示了在某些例題無法有效得到準確的最佳解而只能求得近似最佳解，可能是 MTNS 方法的搜尋模式不適用於這些例題，所以在這些例題上求解的精確度無法提高，不過在搜尋執行時間上的優勢比較一些混何其他方法的演算法較為快速，原因在於 MTNS 還是以禁忌搜尋法為主體，指針對鄰居解的搜尋做改變，相較於其他方法則混合了兩種演算法在運算時間上會耗費較多時間，本研究在執行時間上的優勢，雖然無法完全求得 0 誤差的結果，但是些許的差距皆可以再容許範圍裡，因此 MTNS 在時間上的優勢對此有益無害，未來必須再強化關鍵路徑上的方法改良與強化區域收斂上做改進，才可以融合多樣化搜尋達成快速正確無誤的解決方案。

參考文獻

- [1]王治平，實際零工式生產排程問題的派工法則，國立政治大學資訊管理研究所碩士論文，2003 年。
- [2]陳宜欣、陳稼興、許芳誠，「遺傳演算法

- 於 Job Shop 排程問題上的研究」, 技術學刊, 第十六卷, 第一期, 2001 年。
- [3] 黃俊傑, 應用高效率禁區搜尋演算法於有嚴重瓶頸處理機器的派工問題, 國立暨南國際大學資訊管理研究所碩士論文, 2006 年。
- [4] Adenso, D., “An SA/TA mixture algorithm for the scheduling tardiness problem”, *European Journal of Operational Research*, Vol. 88, 6-14, 1996.
- [5] Conway, R. W., “Priority dispatching and work-in-process inventory in a job shop”, *Journal of Industrial Engineering*, 16, 228-237, 1965.
- [6] Garey MR, Johnson DS, Sethi R, “The complexity of flow-shop and job-shop scheduling”, *Math Oper Res*, Vol.1, 117-129, 1976.
- [7] Herrmann, J. W., “Improving Manufacturing System Performance through Rescheduling”, <http://www.isr.umd.edu/~jwh2/papers/rs-20010621.pdf> Retrieved 2007-11-10.
- [8] Jain, A. S., and S. Meeran. ; A State of the Art Review of Job-Shop Scheduling Techniques ; http://users.otenet.gr/~eis/106_Jain.zip Retrieved 2007-11-10
- [9] Joseph Y-T. Leung, “Handbook of Scheduling: Algorithms, Models, and Performance Analysis”, Boca Raton : Chapman & Hall/CRC, 2004.
- [10] Kolahan, F. M. and Liang, M. Z., “Solving the combined part sequencing and tool replacement center: a Tabu search approach”, *Computer Industrial Engineering*, 65 28(4), 731-743, 1995.
- [11] L. Wang and D.Z. Zheng, “A Modified Genetic Algorithm for Job Shop Scheduling”, *Int J Adv Manuf Technol*, Vol.20, 72-76, 2002.
- [12] Leyuan, S., and Yunpeng, P., “An efficient search method for job-shop scheduling problems”, *Automation Science and Engineering, IEEE Transactions on* see also *Robotics and Automation*, Volume 2, No. 1, 73-77, 2005.
- [13] Liaw, C.-F., “A hybrid genetic algorithm for the open shop scheduling problem”, *European Journal of Operational Research*, Vol. 124, 28-42, 2000.
- [14] Masato Watanabe, Kenichi ida, Mitsuo Gen, “A genetic algorithm with modified crossover operator and search area adaptation for the job-shop scheduling problem”, *Computers & Industrial Engineering* (48), 743-752, 2005.
- [15] Michael Pinedo, Xiuli Chao, *Operations Scheduling with Applications in Manufacturing and Services*, Irwin/McGraw-Hill, Boston, 1999.
- [16] Nowicki E, Smutnicki C, “An advanced tabu search algorithm for the job shop problem”, *J Sched* 8:145-159, 2005.
- [17] Parunak, H. V. D., “Characterizing the Manufacturing Scheduling Problem”, *Journal of Manufacturing Systems*, Vol.10, 241-259, 1991.
- [18] Saidi-Mehrabad, Mohammad and Fattahi, Parviz, “Flexible job shop scheduling with tabu search algorithms”, *International Journal of Advanced Manufacturing Technology*, Vol. 32, No. 5-6, 563-570, 2007.
- [19] Scrich, Cintia Rigao, Armentano, Vinicius Amaral, and Laguna Manuel, “Tardiness minimization in a flexible job shop: A tabu search approach”, *Journal of Intelligent Manufacturing*, Vol.15, No. 1, 103-115, February, 2004.
- [20] V. P. Eswaramurthy & A. Tamilarasi, “Hybridizing tabu search with ant colony optimization for solving job shop scheduling problems”, *Int J Adv Manuf Technol* (40), 1004-1015, 2009.
- [21] Wen, U. P. and Ching-I, Y., “Tabu search methods for the flow shop sequencing problem”, *Journal of Chinese Institute of Engineers*, 20(4), 465-470, 1997.
- [22] Wesley, B. J., and J. B. Chambers, “Solving the job shop scheduling problem with tabu search”, *IEEE Transactions*, Vol.27, No. 2, 257-263, 1995.