

國立交通大學
工業工程與管理學系

碩士論文

以蟻群最佳化演算法求解流線型製造單元排程



**An Ant Colony Optimization Approach to
Scheduling Flowshop Manufacturing Cell**

研究生：李奕勳

指導教授：巫木誠 博士

中華民國一百年六月

以蟻群最佳化演算法求解流線型製造單元排程

**An Ant Colony Optimization Approach to Scheduling
Flowshop Manufacturing Cell**

研究生：李奕勳

Student：Yi-Hsun Li

指導教授：巫木誠博士

Advisor：Dr. Muh-Cherng Wu



Submitted to Department of Industrial Engineering and Management

College of Management

National Chiao Tung University

In Partial Fulfillment of the Requirements

for the Degree of Master of Science

in

Industrial Engineering

June 2011

Hsin-chu, Taiwan, Republic of China

中華民國一百年六月

以蟻群最佳化演算法求解流線型製造單元排程

研究生：李奕勳

指導教授：巫木誠 博士

國立交通大學工業工程與管理研究所

中文摘要

蟻群最佳化演算法是一種巨集啟發式演算法，被廣範應用於求解複雜的空間搜尋問題。過去的研究，大部份注重於如何應用或加強演算法的搜尋機制，因此，在本研究中提出了新的研究方向：採用新的解表達法，是否能增進蟻群最佳化演算法的表現？

研究中以固定序列式製造單元的排程為應用問題，比較兩種不同演算法的求解績效。兩種表達法的演算流程相同，但採用的解表達法不同。一種表達法是過去研究常用到的解表達法—舊解表達法，一種是 Wu *et al.* (2011) 用到的解表達法—新解表達法。實驗結果顯示，新解表達法在大部份的實驗情境下贏過過去常用的舊表達法，這樣的發現指出針對解表達法的創新及改良，是應用巨集啟發式演算法的研究新方向。

關鍵詞：蟻群最佳化演算法、解表達法、排程

An Ant Colony Optimization Approach to Scheduling

Flowshop Manufacturing Cell

Student : Yi-Hsun Li

Advisor : Dr. Muh-Cherng Wu

Department of Industrial Engineering and Management
National Chaio Tung University

Abstract

Ant Colony Optimization (ACO), a type of meta-heuristic algorithms, has been widely used in solving complex space-search problems. Most prior research focused on how to apply or enhance ACO to various problems. Aside from the traditional track, this research examines a new research issue—Can the adoption of a new solution representation scheme improve the performance of ACO? A Flowshop Manufacturing Cell Scheduling Problem with permutation is used as the problem context, and two ACOs are compared. The two algorithms, essentially the same in algorithmic flow, are distinct in using two different solution representation schemes (respectively called S_{old} and S_{new}). Noticeably, S_{old} was developed by prior studies and S_{new} is by Wu *et al.* (2011); the two algorithms are named $ACO-S_{old}$ and $ACO-S_{new}$ accordingly. Extensive numerical experiments reveal that the two algorithms performs equally well in small setup time (LSU) scenarios. Yet, $ACO-S_{new}$ outperforms $ACO-S_{old}$ at small and medium setup time (SSU/MSU) scenarios. This finding highlights an important new research track—exploring new solution representation schemes while applying meta-heuristic algorithms to various space-search problems.

Keywords: Ant Colony Optimization, solution representation, scheduling

誌謝

本論文得以順利完成，首要感謝的是巫木誠教授的悉心指導與教誨。除了在學術上獲益良多外，巫老師時常分享自己待人處事的經驗，強調誠信的重要，還有用有聰明、有效率的方法去解決問題。在老師的敦敦教誨下，有紀律的生活以及思考模式，使我在學術研究上獲益良多，得以終身受用。同時也感謝許錫美教授和彭德保教授在論文口試時，所給予的寶貴意見與指導，讓本論文更加完備。

在研究所的兩年中，感謝實驗室的同學曾偉杰、陳威宇、林耿漢、潘冠銘，陪我度過修課與撰寫論文的的日子，時常相互照應、鼓勵與討論。還有一起陪我渡過無數個夜晚的弟兄、劉子郁、張志揚、陳長鈺和許宏斌，使我在交大的生活變的愉快而充實。在此並特別感謝陳振富學長的幫助，指導我論文的撰寫，並提供意見解決在研究中遇到的諸多困難，使我得以順利完成研究。

最後，特別感謝我最親愛的家人，在論文撰寫期間給我無數的關愛與支持。特別是我的父母親李明通先生與謝麗淑女士，感謝您們多年來的辛勞與關懷，您們的體諒讓我得以專心在論文的研究上。在此，謹以此論文獻給我最敬愛的家人、師長與朋友。

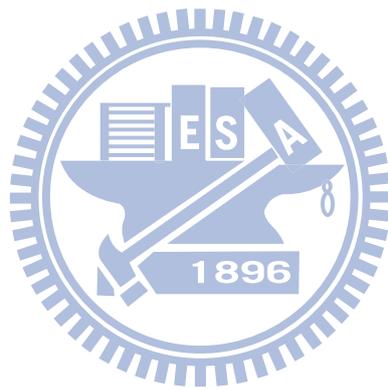
李奕勳 于 新竹交大

2011'7'1

目錄

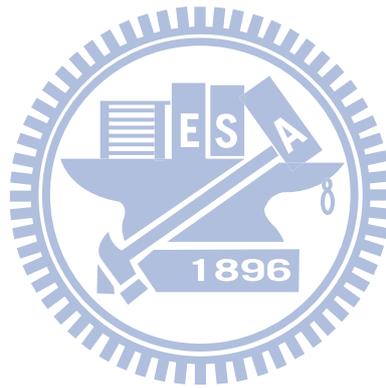
中文摘要.....	ii
Abstract.....	iii
誌謝.....	iv
目錄.....	v
表目錄.....	vii
圖目錄.....	viii
第一章 緒論.....	1
1.1 研究背景.....	1
1.2 研究議題.....	2
1.3 研究動機.....	2
1.4 研究方法.....	3
1.5 論文架構.....	4
第二章 文獻回顧.....	5
2.1 單元製造排程問題的求解方法.....	5
2.2 解表達法.....	7
2.3 蟻群最佳化演算法(Ant Colony Optimization, ACO).....	9
第三章 研究方法.....	13
3.1 研究問題.....	13
3.2 研究假設.....	13
3.3 解表達法.....	14
3.3.1 新解表達法.....	14
3.3.2 新解表達法.....	15
3.4 蟻群最佳化演算法.....	16
3.5 蟻群最佳化演算法與不同解表達法的應用.....	20
3.5.1 舊解表達法與蟻群最佳化演算法.....	20
3.5.2 新解表達法與蟻群最佳化演算法.....	22
第四章 實驗情境與結果.....	23
4.1 實驗情境.....	23
4.2 ACO 參數設定.....	24
4.2.1 費洛蒙蒸發率(ρ).....	24
4.2.2 期間常數(t_{max}).....	25
4.3 測試環境.....	26
4.4 實驗結果.....	26
4.5 比較結果分析.....	27
4.5.1 數值統計分析結果.....	27
4.5.2 統計檢定結果.....	29

第五章 結論與未來研究.....32
參考文獻.....34
附錄一、各個情境下 $d_{s,i}(\%)$ 值計算表格 A



表目錄

表 2.1、MCSP 相關文獻總表.....	7
表 2.2、舊解表達法相關文獻.....	8
表 2.3、新解表達法相關文獻.....	9
表 4.1、三種費洛蒙蒸發率實驗結果.....	25
表 4.2、不同的收斂代數下實驗結果.....	25
表 4.3、不同的收斂代數下求解時間.....	26
表 4.4、實驗數據平均結果.....	31
表 4.5、檢定結果.....	31



圖目錄

圖 2.1、演算法發展架構.....	7
圖 2.2、10 工件及 3 工件族以舊表達法表示例子.....	8
圖 2.3、10 工件及 3 工件族以新表達法表示例子.....	9
圖 3.1、舊解表達法示例.....	15
圖 3.3、工件族解讀.....	16
圖 3.4、解讀出完整工件加工序.....	16
圖 3.5、ACO 基本流程圖.....	17
圖 3.6、貪婪演算法求解示例.....	18
圖 3.7、輪盤法示例.....	19
圖 3.8、ACO 產生舊解表達法之解與目標值流程圖，以三工件族為例.....	21
圖 3.9、ACO 產生新解表達法之解與目標值流程圖.....	22
圖 4.1、各種情境下新解表達法之改善率.....	28
圖 4.2、新解表達法比舊解表達好的次數統計.....	28
圖 4.3、兩種解表達法平手的次數統計.....	29
圖 4.4、新解表達法不輸給舊解表達法之次數統計.....	29



第一章 緒論

1.1 研究背景

近年來，隨著消費者的消費行為與相關商業模式的改變，使得產品需求有著朝向少量但多樣的客製化趨勢，因此產品供應端的生產方便由過去的大量生產(如備貨型生產)轉為少量生產(如訂單式生產)。其中，因應這樣生產型態而生的生產方式之一－單元式製造系統(Cellular Manufacturing System, CMS)－成了現行具客製化特性的工廠生產方式之一。

單元製造系統是一種群組技術(Group Technology)的應用，是一產品為導向的生產方式，具有零工式生產(Job Shop)的彈性及流程式生產(Flowshop)的效率，能減少在製品存貨、減少設置整備時間及提高機器使用率等。其中，製造單元(Manufacturing Cell, MC)是由一群機器組成，負責製造特定數量的產品族(Part Family)。一產品族是由一群具有相同或相似加工需求的工件所組成，這些需求包含整備作業或加工治具等。在生產排程方面，不僅需要考慮工件族與工件族間(Among Part Family)的生產順序，也必須同時考慮工件族裡工件(Within Part Family)的加工順序。具有上述特性的的排程問題被統稱為製造單元排程問題(Manufacturing Cell Scheduling Problem, MCSP)。

MCSP 在過去相關的研究裡，已證明為 NP-Hard 問題(Schaller, 2000)，當問題規模變大時，即無法在合理的時間內利用傳統數學規劃法(如整數規劃、動態規劃、分支界限法等)求得最佳解。所以在近年來的研究裡，學者大多採用啟發式(heuristic)或巨集啟發式(meta-heuristic)演算法來求解。這些啟發式及巨集啟式演算法的特點在於能在合理的時間內求得可行解，雖然不一定是最佳解，但往往能得到品質不錯的近似最佳解。這類方法常見的有：基因演算法(Genetic Algorithm, GA)、模擬退火法(Simulated Annealing, SA)、粒子群演算法(Particle Swarm Optimization, PSO)……等。

為了更新符合實際情況，近年來多數排程研究會進一步考慮機器整備時間順序相依(Sequence Dependent Family Setup Times, SDFST)的狀況，也就是機器的整備時間會隨著工件族順序的不同而有所差異。

1.2 研究議題

在本篇研究中，討論的是流程式生產(Flowshop)排程，也就是在製造單元(Manufacturing Cell)中，工件族需要的機台組合不但一樣，且加工順序也一樣，這樣的製造單元又被稱為純流程式生產製造單元(Pure Flowline Manufacturing Cell)。此外，同時考慮順序相依家族整備時間及固定序列(Permutation)的特性，也是本研究問題的重點。其中，固定序列特性指的是各個工件族的加工順序在各個機台上皆相同，決定了第一台機台的加工順序後，第二台之後的機台上的加工順序皆比照第一台的順序決定排列。綜合上述各項特性的問題，稱為固定序特性具順序相依家族整備時間之流程式製造單元排程問題(Flowshop MCSP with SDFST under Permutation Schedule Problem) – 本篇研究主要的排程問題。

1.3 研究動機

近年來，啟發式及巨集啟發式演算法相當廣範地被運用於求解排程問題，其中，演算法的解搜尋機制影向著求解的品質。面對層出不窮的排程問題，單一的演算法搜尋機制不一定符合需求，因此，研究及改良演算法中的解搜尋機制(或稱進化機制)是過去學者的注重的核心之一，例如基因演算法中染色體的突變機制或交配機制，以及基因的篩選、淘汰準則。這些因應而生的改良方法的確能為各個問題求得品質不錯的解，也為各種演算法的應用奠定了典範，但在改良的方向上，卻少有研究注重於解表達法上。在 Wu *et al.* (2011)的論文中，認為演算法的改良除了解演化的機制外，解的表達法也會是影響演算法解品質的重要面項之一，相信相同的解搜尋機制配上較好的解表達法會產生品質較好的解。這樣的概

念在 Wu *et al.* (2011)的論文裡應用於基因演算法上，並採新解表達法求解固定序列特性具順序相依家族整備時間之流程式製造單元排程問題(Flowline MCSP with SDFST under Permutation Scheduling Problem)。

雖然過去的學者有提出發展新解達法的概念及新的解表達法，但是卻沒有一種表達法被證明可以在多數啟發式演算法中相較於其它解表達法具有一致性好的表現。因此，在本篇研究裡採用 Wu *et al.* (2011)新解表達法結合蟻群最佳化演算法(Ant Colony Optimization, ACO)來求解固定序列特性具順序相依家族整備時間之流程式製造單元排程問題並與 Lin *et al.* (2009b) 的舊解表達法做比較，試著證明新解表達法除了在基因演算法中能表現的比舊解表達法好之外，是否在蟻群最佳化演算法中也能有一致性的表現。

1.4 研究方法

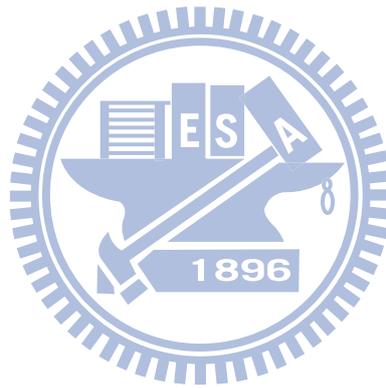
本篇論文研究的排程問題是固定序列特性具順序相依家族整備時間之流程式製造單元排程問題，重點將放在不同解表達法配上蟻群最佳化演算法求解的解品質比較。

研究用實驗參數與 Wu *et al.* (2011)一樣取照 Schaller *et al.* (2000)的實驗參數，其中，該篇論文分成長、中、短三類機台整備時間，每種整備時間又向下分成十種機台數與工件族數的組合，全部共 30 種情境。比較的解表達法為 Lin *et al.* (2009b) 中提到的舊解表達法及 Wu *et al.* (2011) 提到的新解表達法。兩種表法皆配上蟻群最佳化演算法(ACO)進行 30 種情境的實驗，最後取實驗結果目標值的平均做解品質的比較。

本篇研究中使用的績效衡量指標為最大完工時間(Makespan)，也就是從一開始到全部工件加工完成所需花費的加工時間(最後一個工件完成的時間)。

1.5 論文架構

本論文共有五章內容。第一章如上所述。第二章是相關的文獻整理。第三章針對問題與研究方法做詳細的定義及流程說明，並說明巨集演算法配上新及舊解表達法應用上的差異。第四章為研究方法的實驗結果，包含數據整理結果與統計分析。第五章，最後，提出未來可能的研究方向。



第二章 文獻回顧

本章將針對固定序列特性具順序相依整備時間之流程式製造單元排程問題的各項特性做文獻探討，主要分成三個小節，第一小節探討製造單元排程問題的求解方法，第二小節探討不同解表達法的應用，第三小節探討蟻群最佳化演算法及總結。

2.1 單元製造排程問題的求解方法

現今的生產環境下，如何讓生產有效率，也就是如何確保顧客的訂單能準時達交，是非常重要的議題，因此有了群組技術的出現。應用概念上主要分成依加工特性分群的工件族(Part Family)及因應生產流程需求而對機器分群的機台群(Cell)。因其能讓小型批量生產達到如大量生產的經濟效益兼具生產彈性，所以廣為學者所研討。

單元製造是現今盛行的一種群組技術的應用之一，近年來有非常多相關應用的研究。這樣的生產問題在過去研究中(Schaller *et al.*, 2000)已說明為 NP-hard 的問題，求解問題所花時間會隨著問題規模增大而呈指數上升，無法在有限時間內利用最佳化方法(Optimization Method)求得最佳解。因此，近年來這類問題學者大多利用啟發式或巨集啟發式演算法來求得近似最佳解。

在 Schaller *et al.* (2000)論文中，研究的是一單純流程式製造單元(pure flow-shop manufacturing cells)的排程問題，以印刷電路板(PCB)製造業中應用表面粘著技術 SMT(Surface Mounting Technology)的生產為背景。研究中，作者提出一混合式兩階段排程方法 CMD 來求解，並與其它十一種啟發式演算法比較，最後以實驗結果證明 CMD 的表現比較好。其中 CMD 是由 Campbell-Dudek-Smith's procedure(C)、Modified NEH procedure(M)及 Descent heuristic(D)三種啟發式演算法組成，C 負責工件族內工件加工序(Within Family Sequence)的排程，M 負責工件族間加工序(Among Family Sequence)，D 負責增進 CM 的結果品質。

在 França *et al.* (2005) 中，作者利用兩種進化演算法(Evolutionary Algorithm) – 基因演算法(Genetic Algorithm)及改良瀾集演算法(Memetic Algorithm with Local Search) – 來求解與 Schaller *et al.* (2000) 一樣的排程問題。其中，作者設計了階層化母代的母代保留架構(Population Structure)來限制可能被選取來重組(Recombination)型成子代的組合數量，增加了演化後的解品質。而改良後的瀾集演算法則是在演化過程中加入了區域搜尋(Local Search)機制，其不但增加保留的母代品質也增進了重組選取(Selection for recombination)過程的品質，讓瀾集演算法的表現勝過了基因演算法及 CMD (Schaller *et al.*, 2000)演算法。

在 Hendizadeh *et al.* (2008) 中，作者提出五種禁忌搜尋法(Tabu Search, TS)導向的演算法來求解與 Schaller *et al.* (2000)及 França *et al.* (2005)相同的排程問題，五種演算法的差別在於演算法中深度化與多樣化機制(Intensification and diversification schemes)的不同。作者強調藉由深度化與多樣化(Intensify and diversify)搜尋(演化)方向，才能讓演算法找到未發現的解空間，進而找到比較好的解。與過去的 CMD (Schaller *et al.*, 2000)及 MA (França *et al.*, 2005)比較，提出的五種演算法皆能求得品質較好的解，其中，又以 TS/SA 演算法表現最好，其特點在於加入模擬退火法(Simulated Annealing, SA)的概念後，可讓深度化及多樣化搜尋的比重達到一個較優的平衡點，同時具有較足夠的廣域及區域搜尋。

Lin *et al.* (2009a)提出一改良模擬退火法來求解與 Schaller *et al.* (2000)、França *et al.* (2005)及 Hendizadeh *et al.* (2008)相同的排程問題。其特點在於作者以科西方程式(Cauchy function)取代傳統模擬退火法中的波滋曼方程式(Boltzmann function)。其表現與過去文獻提出的方法比較，如 CMD (Schaller *et al.*, 2000)、MA (França *et al.*, 2005)、TS/SA (Hendizadeh *et al.*, 2008)及傳統模擬退火法(SA)等，皆具有較好解品質和求解速度。

在 Lin *et al.* (2009b)中，提出以三種著名的巨集啟發式演算法 – 基因演算法(GA)、模擬退火法(SA)及禁忌搜尋法(TS) – 來求解非固定序列特性具順序相依家

族整備時間製造單元排程問題(MCSP with SDFST in non-permutation)，提出考慮非固定序列(Non-permutation)的情境可以求得比考慮固定序列(permutation)情境更好的解。固定序列指的是在不同機台上工件族的加工順序需相同，而非固定序列則可以不同。過去相關研究為簡化問題，通常僅考慮固定序列的情況，所以在此篇論文中，便以實驗結果說明非固定序列式排程能達到最佳的效益。

表 2.1、MCSP 相關文獻總表

使用演算法	評估指標	作者(年份)
Heuristic Algorithms	Makespan	Schaller <i>et al.</i> (2000)
Genetic Algorithm	Makespan	França <i>et al.</i> (2005)
Tabu Search	Makespan	Hendizadeh <i>et al.</i> (2008)
Simulated Annealing	Makespan	Lin <i>et al.</i> (2009b)
Genetic Algorithm	Makespan	Lin <i>et al.</i> (2009a)

2.2 解表達法

MCSP 的問題，過去文獻已證明為 NP-hard 的問題，大多此類問題採用巨集啟發式演算法(Meta-heuristic)來求解。為因應解不完的排程問題，相對應的新演算法或改良後演算法也不斷出現。在演算法的發展中，大致可分為兩大類，一是進化法上的發展，一是解表達法上的發展(圖 2.1)。

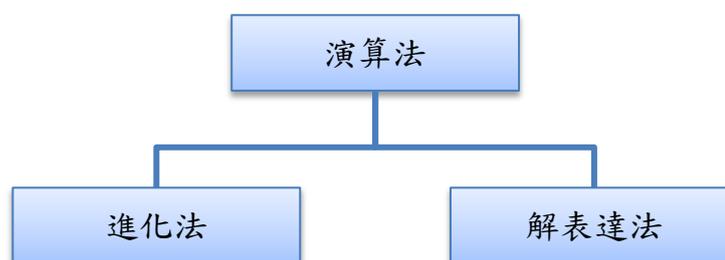


圖 2.1、演算法發展架構

進化法上的改良，可以因應不同的問題特性讓過程解有不同的進化、演進或保留方式，而解表達法上的改良往往影響到過程解(或最終解)儲存資訊的資料結

構。一個好的解表達法可以簡化解演進的複雜度進而加過求解時間；在同樣的進化法下讓進化機制有較高的機會選到較好的母代(Parent solution)以確保進化過程的品質。在 Wu *et al.* (2011)中即使用新解表達法於基因演算法中，其實驗數據顯示新解表法搭配基因演算法所求出的解品質及運算時間皆優於舊解表達法與基因演算法的搭配。

過去常見應用於 MCSP 中的兩種解表達法的文獻整理如下：

(1) 舊解表達法，主要分成工件族加工序及工件族內工件加工序兩部份：

表 2.2、舊解表達法相關文獻

評估準則	演算法	作者 (年份)
Makespan	Genetic Algorithm	Franca <i>et al.</i> (2005)
Makespan	5 TS-based Algorithm	Hendizadeh <i>et al.</i> (2008)
Makespan	Genetic Algorithm Simulated Annealing Tabu Search	Lin <i>et al.</i> (2009b)
Makespan	Simulated Annealing	Lin <i>et al.</i> (2009a)

家族序			族(1)加工序			族(2)加工序			族(3)加工序			
F1	F3	F2	J1	J3	J2	J5	J4	J6	J7	J8	J9	J10

圖 2.2、10 工件及 3 工件族以舊表達法表示例子

(2) 新解表達法，其為混合資訊之架構：

表 2.3、新解表達法相關文獻

評估準則	演算法	作者 (年份)
Makespan	Genetic Algorithm	Lu (2009)
Makespan Total tardiness	Genetic Algorithm	Tai (2010)
Makespan	Genetic Algorithm	Wu (2011)



圖 2.3、10 工件及 3 工件族以新表達法表示例子

2.3 蟻群最佳化演算法(Ant Colony Optimization, ACO)

蟻群系統演算法(Ant System, AS)最早是由 Dorigo (2000a) 基於自然界螞蟻行為所發展出來的一種人工智慧啟發式演算法。Dorigo (2000b) 修正 AS 太快收斂導致容易落入區域最佳解的缺點，提出改良的螞蟻演算法(Ant Colony System, ACS)。此後，Dorigo 正式將這一系列演算方法統稱為蟻群最佳化演算法。本研究中採用 AS 的流程。

ACO 的基本概念是來自自然界螞蟻外出覓食的行為。螞蟻覓食過程會在行經巢穴與食物間的路徑上留下費洛蒙(Pheromone)，為後續出發的螞蟻留下是否遵循著此一路徑繼續前進的標記。覓食初期螞蟻會四處隨機移動，並留下費洛蒙。當螞蟻遇到費洛蒙路徑時，會判斷並選擇一路徑前進，經過的路徑會留下額外的費洛蒙，使費洛蒙濃度增高，而沒經過的路徑，會隨時間的流逝使得費洛蒙不斷蒸發、濃度不斷降低。基本上，螞蟻路徑的選擇主要是取決於費洛蒙的濃度，濃度愈高愈能吸引螞蟻前往。因此，費洛蒙是螞蟻間的溝通工具。由於行經的路徑較短的螞蟻會較快覓食回巢，使路徑上單位時間內所留下的費洛蒙較多。而覓食

路徑較長者，覓食回巢的時間也較長，所以單位時間內所留下的費洛蒙較少，使路徑上的費洛蒙濃度漸漸降低，愈來愈不能吸引螞蟻前往。經過一段時間後，此起彼落的加強作用下，很快地所有螞蟻便會使用較短的路徑去覓食。

ACO 是由幾個重要元素所組成，相關內容分述如下：

(1) 人工螞蟻(Artificial ants)：

人工螞蟻是 ACO 進行隨機找解的基本單位，每隻人工螞蟻皆是根據上述自然螞蟻(Natural ants)的特性做適當修改而來，以符合演算法的特性。每隻人工螞蟻都具有下列四項功能：

1. 留下費洛蒙功能。
2. 感知路徑上費洛蒙濃度功能。
3. 判斷路徑上其它屬性(如距離長度、加工時間長短)功能。
4. 選擇下一拜訪節點並移動至該節點的功能。

(2) 費洛蒙(Pheromone)

自然界中螞蟻的視力不佳，所以螞蟻的交流覓食資訊的媒介都透過一種化學物質－費洛蒙－來達成。每隻螞蟻都具有遺留及感知費洛蒙的能力，所以費洛蒙濃度愈高的路徑愈能吸引螞蟻遵循前往，進而累積更多的費洛蒙，凸顯與其它路徑之間的優劣差異。

(3) 期望值(Desirability)或能見度(visibility)

期望值代表該節點屬性值(如路徑長度、加工時間)的優劣。以求解最短路徑長的 TSP 為例，各節點的期望值就是距離長度的倒數，代表路徑愈短，期望值會愈高。期望值愈高的節點，獲得當下一個拜訪點的機率就愈高。通常此特性為靜態(static)，屬於各個節點的期望值不會隨著找解過程的推移而有所改變。

(4) 轉換機率(Transition probability)

ACO 是一種機率性的找解方法，其螞蟻在進行路徑選擇或是下一個節點的選擇時，是利用一轉換機率行下一節點的選擇。考慮的因素包含費洛蒙強度及期望值。

(5) 揮發機制(Evaporation mechanism)

揮發機制是相對於螞蟻遺留費洛蒙的機制，主要的功能是讓所有路徑上的費洛蒙濃度以一定的比率及速度進行蒸發，以避免部份路段的費洛蒙不斷累積而致使所有螞蟻均依循相同路徑，而無法發揮全域搜尋的能力。這種因為部份路徑費洛蒙濃度過高而致使螞蟻無法考慮其它路徑的情形通常被稱為停滯(Stagnation)。揮發機制的存在不僅防止了停滯現象的發生，也讓 ACO 更符合自然螞蟻覓食的行為及精神。

由於 ACO 具有漸進式的正回饋機制及負回饋機制，讓它擁有不錯全域及區域搜尋能力，因此廣為學者於排程上所應用。過去 ACO 多應用於求解流程式(Flowshop)生產排程問題。

在 Merkle and Middendorf (2005)裡，作者利用 ACO 搭配論文提出的隨機螞蟻(Random Ants)概念來求解具固定序列特性之單機台流程式製造排程問題。因為在傳統 ACO 排程初期選擇的對象(工件)對後期對象的選擇有非常大的左右性，所以在論文中設置了隨機螞蟻機制，其在尋找下一個造訪點機制上有別於人工螞蟻(Artificial ants)，主要用來再加強 ACO 的全域搜尋的能力，減輕初期求解對後期的左右性，減慢求解的收斂速度。

在 Gajpal and Rajendran (2006)中，作者提出新蟻群最佳化演算法(New Ant Colony Optimization, NACO)來求解具固定序列特性之流程式製造排程問題，其特點在於能在求解最小化完成時間變異的指標中能比其它演算法表現的好。

在 Chang *et al.* (2008)中，因為 ACO 具有的揮發概念有避免陷入停滯(Stagna-

tion)的好處，因此作者提出了 Generating artificial chromosomes for Genetic Algorithm (ACGA)來求解具固定序列特性之單機台流程式生產排程問題。其方法特點在於在基因演算法建構基因序(Chromosomes)的機率模型中加入 ACO 的揮發機制，此機制讓建置新基因序時更能發現其它可能的解(全域搜尋)，進而達到更好的解品質。

在 Yagmahan and Yenisey (2010)中，作者提出利用多目標蟻群最佳化演算法 (Multi-Objective Ant Colony System)來求解多目標的單機台的排程問題。方法特點在於結合兩種目標值來建構費洛蒙路徑及加入類似禁忌搜尋法(Tabu Search)的區域搜尋概念。雖然研究中僅示例以最大完工時間(Makespan)及流程時間(flow-time)兩指標的求解結果，但我們不難從方法模型看到求解多個目標值的可能性。

由過去文獻顯示，ACO 是具有非常大彈性的巨集啟發式演算法，其大部份應用皆是用於求解流程式生產的問題，加上過去並沒有相關的 ACO 方法應用於固定序列特性具順序相依家族整備時間之流程式製造單元排程問題，所以本論文選用 ACO 為研究工具。



第三章 研究方法

本章裡，第一節先簡介研究的問題，第二節介紹研究用到的假設，第三節介紹相關的不同解表達法，第四節說明蟻群最佳化演算法及其如何應用在流程式生產排程問題，最後介紹蟻群最佳化演算法在不同解表達法上應用的差異。

3.1 研究問題

本論文研究問題如前面第一章所提到的，是固定序列特性具順序相依家族整備時間之流程式製造單元排程問題，是 NP-Hard 的問題，隨著工件數、工件族數的增加而大量增加求解複雜度(時間)，所以本篇採用蟻群最佳化巨集啟發式演算法(ACO)來求解。其中，研究重點放在新解表達法(Wu *et al.* 2011)及舊解表達法(Lin *et al.*, 2010b)與 ACO 應用上所產生的解品質差異，測試新解表達法與 ACO 的搭配是否能比舊解表達法的表現來的好。兩者的解品質以完成時間導(Completion-time based)的最大完工時間(makespan)為指標。

3.2 研究假設

固定序列特性具順序相依家族整備時間之流程式製造單元排程問題在過去已有一些學者研究過，如 Franca *et al.* (2005)、Schaller *et al.* (2009)、Lin *et al.* (2009a)、Wu *et al.* (2011)等，所以相關問題假設便採用過去文獻使用的假設，詳細內容如下：

- (1) 本研究僅限於分析靜態式(Static)流線型製造單元流程問題，表示工件的數量及其處理時間與工件族的數量及其家族整備時間是已知的，而且為非負整數。
- (2) 所有工件在時間為 0 時可進行處理(Ready Time=0)。
- (3) 工件加工完成，如果下一個加工機台沒有工件正在加工，可直接移至下一個機台加工。

- (4) 同一個工件族內的工件不能被分割處理，必須連續加工完同一個工件族內所有的工件每個工件一次只能經由一部機台加工，每部機台一次只能加工一個工件，不允許多個工件同時在同一機台上加工。
- (5) 在同工件族內，各工件間的整備時間忽略不計。
- (6) 機台前緩衝區內的在製品(Work-in-process)數量無上限。
- (7) 無當機情況發生。

3.3 解表達法

在本節中，將分別介紹不同的解表達法，分別是 Lin *et al.* (2009b)中提到的舊解表達法及 Wu *et al.* (2011)新解表達法，兩者皆是因應製造單元排程問題的設計，所以皆包含了工件族排程的概念。兩種解表達法分述如下。

3.3.1 新解表達法

在新解表達法中，主要分成兩個部份，第一部份為工件族加工序(Among Family Sequence)部份，記錄了工件族間的加工順序，第二部份則為各個工件族內工件加工順序(Within Family Sequence)，記錄同工件族中工件的加工順序。綜合上述兩種資訊，便可以得到全部工件的完整加工序。以圖 3.1 為例，我們由上部圖的家族序部份得知總共有三個工件族且其加工順序為 $F1-F3-F2$ 。從族(1)加工序部份可得知工件族(1)內有三個工件，分別為 $J1$ 、 $J2$ 和 $J3$ ，加工順序為 $J1-J3-J2$ 。剩下與此類推。按照家族序及族內加工序排列，既可得到如圖 3.1 下部圖所示之完整加工序： $J1-J3-J2-J7-J8-J9-J10-J5-J4-J6$ 。

家族序			族(1)加工序			族(2)加工序			族(3)加工序			
F1	F3	F2	J1	J3	J2	J5	J4	J6	J7	J8	J9	J10

J1	J3	J2	J7	J8	J9	J10	J5	J4	J6
----	----	----	----	----	----	-----	----	----	----

圖 3.1、舊解表達法示例

3.3.2 新解表達法

不同於舊解表達法中具明顯切割成的『工件族序』及『工件族內加工序』兩部份，在新解表達法中，採『混合式』的設計，也就是將工件族序及工件族內加工序兩部份的資訊合在一起，並減去多餘的部份(如工件族序部份)。在需要得到完整加工序時，再經由特別的解碼過程(Decoding Process)，取得家族序及族內加工序的資訊。

以圖 3.2 為例，圖中是一由新解表達法表示的完整解。Job 列部份共有十個格子，代表它表示著十個工件的加工排序。由 3.2 假設中，我們知道是一靜態式(Static)排程問題，所以加工件(Job)所屬的工件族(Family)是已知的，因此可以得到圖中 Family 列部份的資訊。從 Family 列可以得到共有三個工件族 F1、F2 及 F3。有了上述兩種資訊便可以開始解碼。解碼的第一步先決定工件族的加工順序，解讀方法為從 Family 列由左向右解讀，先讀到的工件族先排，已讀過的忽略，依序可得到 F1-F2-F3 加工順序(圖 3.3)。第二步決定各個工件族內工件的加工順序，解讀方式一樣為由左向右讀，同工件族工件先讀到的先排，以 F1 的三個 Job 為例，其結果就是 J1-J2-J5，其它工件族工件與此累推，結果如圖 3.4 所示。有了族內加工序後，便可依族間加工序排出完整的工件加工序 J1-J2-J5-J3-J6-J8-J4-J7-J9-J10。

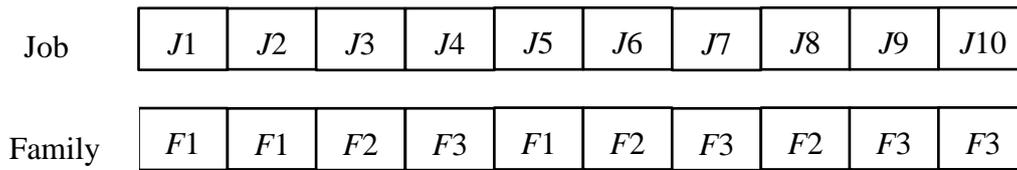


圖 3.2、10 個 Job 以新解表達法表示

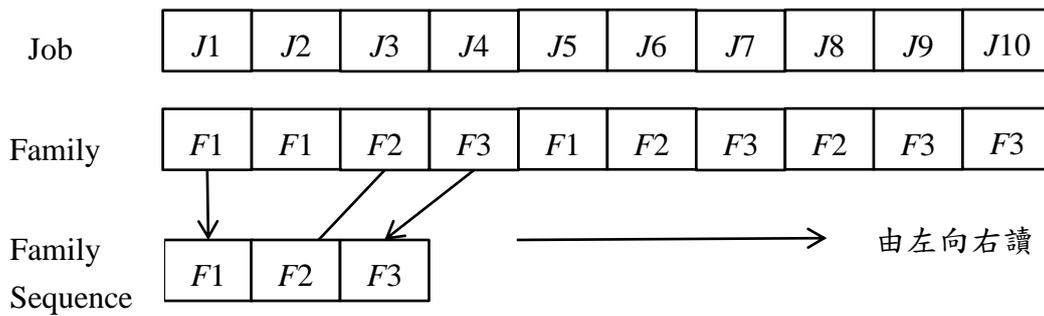


圖 3.3、工件族解讀

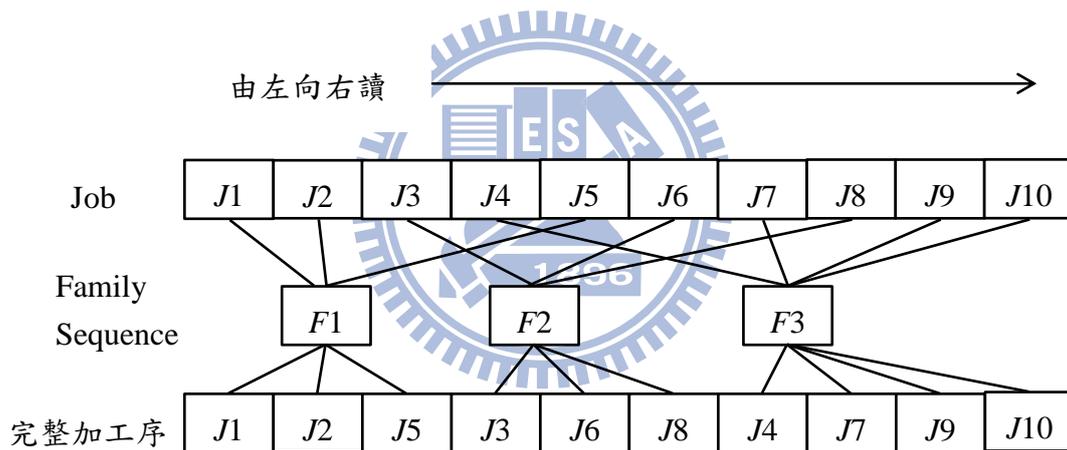


圖 3.4、解讀出完整工件加工序

3.4 蟻群最佳化演算法

本節將說明演算法基本流程及如何套用於排程問題中。ACO 的基本流程大致上分為四個步驟，流程圖如圖 3.5 所示。

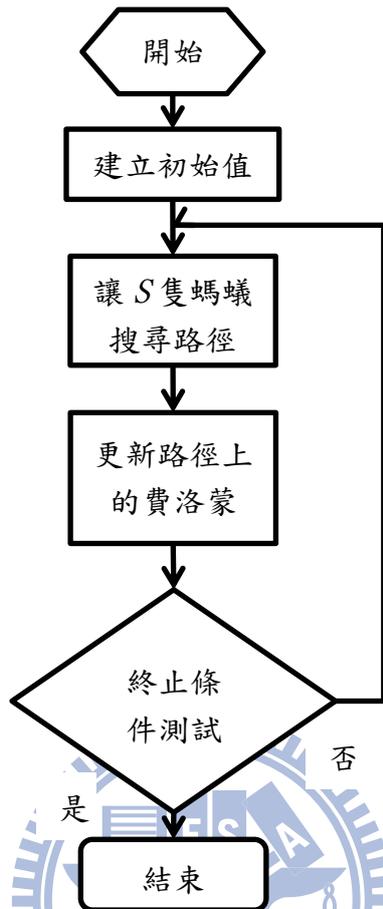


圖 3.5、ACO 基本流程圖

步驟一、起始狀態及初始值設立：

確立問題的數學模式後，便可正式導入 ACO 演算求解流程。開始演算前需設立的參數包括各條路徑上的初始費洛蒙濃度(Initial pheromone, τ_0)、費洛蒙蒸發率或揮發率(ρ)、每演化合放出覓食的人工螞蟻個數(S)、覓食路徑選擇機率常數 α 和 β 及執行期間常數(或稱可執行總次數, t_{max})。費洛蒙揮發率(ρ)通常設為 0.8~0.9 間。可令人工螞蟻個數(S)等於節點總數量或路徑總數。覓食路徑選擇機率常數 α 和 β 設為正整數，通常 β 比較大。起始費洛蒙濃度的設定可設為： $\tau_0 = (SL)^{-1}$ ， L 為貪婪演算法(Greedy Algorithm)所求解出的目標值(makespan)。

貪婪演算法是一種啟發式演算法(heuristic algorithm)，常用來求解最佳化問題(Optimization Problem)。它是一種很直接的求解概念，每次決策都是朝向目前

『最好』的方向前進。其決策不考慮目前的選擇對將來是否有不良的影響，期待以目前的選擇導出全域的最佳解，因此我們可以知道貪婪演算法不一定能求出最解。

圖 3.6 是一個利用貪婪演算法求解四工件加工順序的示例，假設以最短時間完成所有加工時間為目標。Stage1 裡以 Job3 的加工時間最短，所以 Stage1 會選擇 Job3。在 Stage2 裡以 Job2 的加工時間最短，所以會選擇 Job2。剩下與此類推，我們可以得到 Job3-Job2-Job1-Job4 的加工順序。

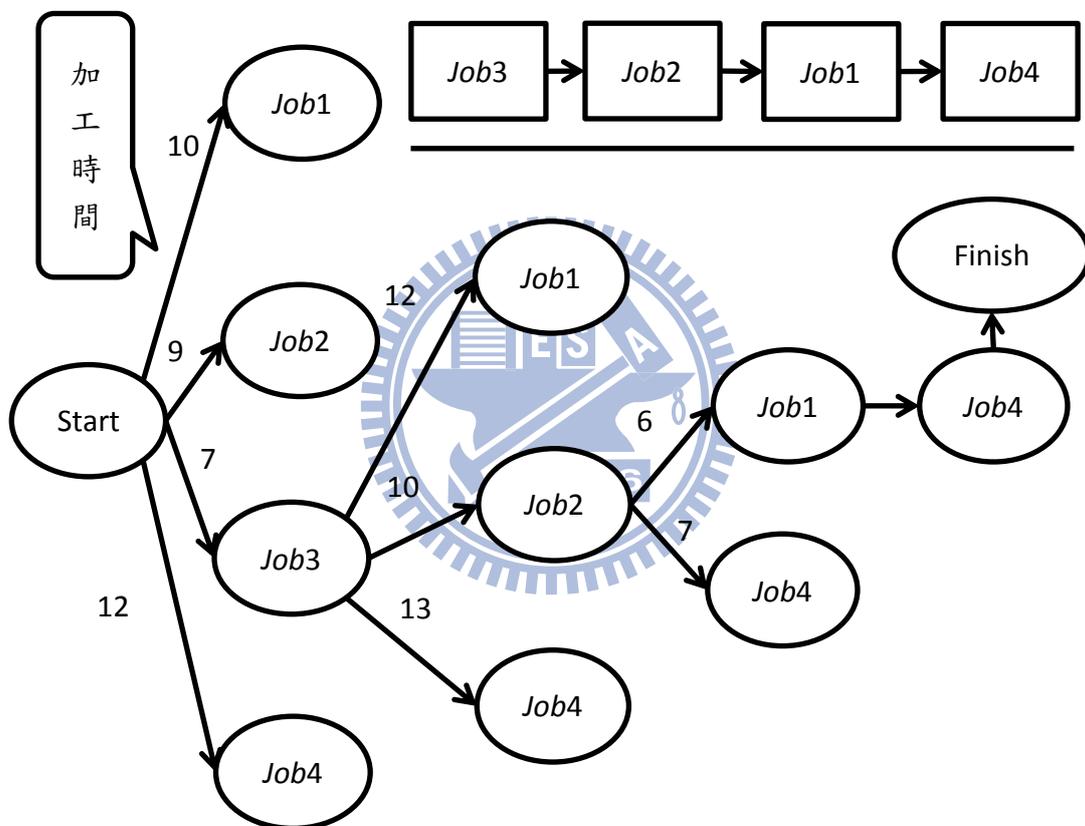


圖 3.6、貪婪演算法求解示例

步驟二、讓螞蟻搜尋路徑，建立費洛蒙路徑

演算的一開始，將 S 隻螞蟻隨機放置於節點(或工件)上，每隻螞蟻即依據當時的節點經過去的情況，計算轉換機率，以便人工螞蟻選擇下一個覓食的節點，與此類推，逐步完成一趟完整的覓食路徑。轉換機率計算公式如下：

$$p_{ij}^s = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum_{u \in J_s(i)} [\tau_{iu}(t)]^\alpha [\eta_{iu}]^\beta} & \text{if } j \in J_s(i) \\ 0 & \text{otherwise} \end{cases} \quad (\text{公式 3.1})$$

其中， $\tau_{ij}(t)$ ：代表節點*i*到節點*j*路徑上在時間*t*的費洛蒙濃度。

η_{ij} ：代表期望值，在本論文中設為加工時間加上整備時間的倒數。

$J_s(i)$ ：位於節點*i*的螞蟻*s*尚未經過的臨近節點集合。

α 、 β ：覓食路徑選擇機率常數，皆為大於整數。

有了各個候選節點的轉換機率後，再利用輪盤法選出下一覓食節點。輪盤法基本步驟是依各個節點的機率值大小，劃分一圓型輪盤所佔面積的大小，並為每塊面積設立刻度範圍值，最後隨機產生一界於0到1的隨機值，由隨機值落於的區塊決定哪個節點為下一個節點。

圖 3.7 是五個 Job 加工的例子，假設一開始已完成 Job4，那麼選出下一個 Job 的步驟即是依照(公式 3.1)計算每個 Job 的轉換機率值，然後依機率值劃分輪盤刻度值範圍。此例中 Job1 為 0~0.4，Job2 為 0.4~0.5，Job3 為 0.5~0.6，Job5 為 0.8~1.0。最後產生一隨機值 0.75，落於 Job3 的範圍，可知 Job3 為下一個加工對象。

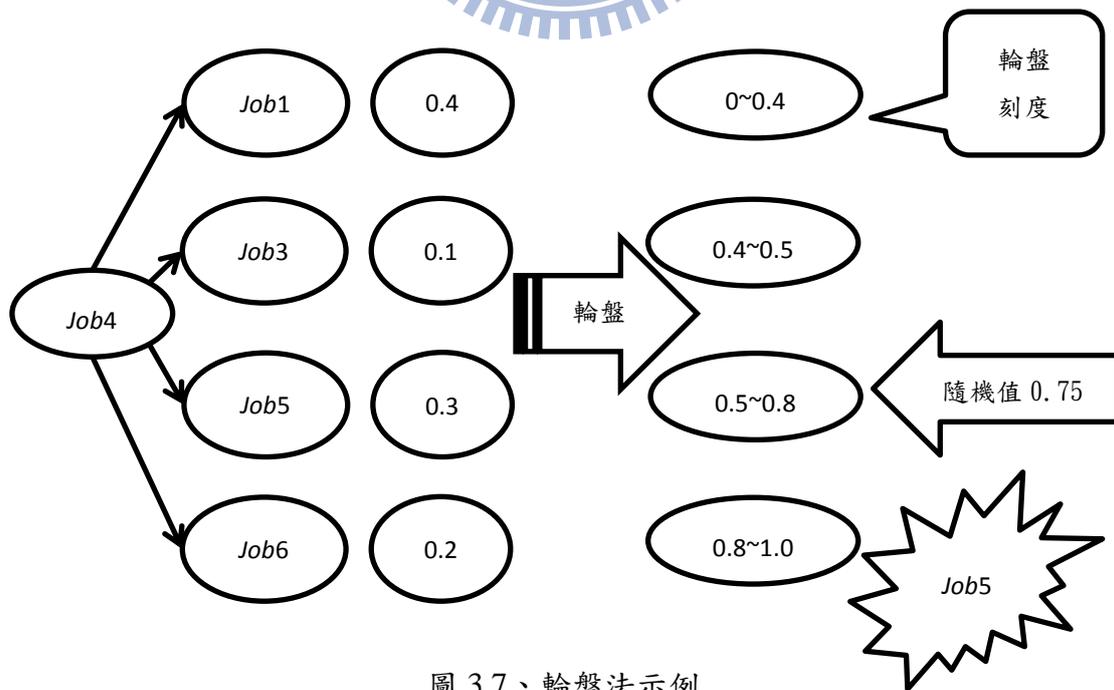


圖 3.7、輪盤法示例

步驟三、更新路徑上費洛蒙濃度

當 S 隻螞蟻皆完成他們的旅程後，即可執行路徑上費洛蒙濃度的更新動作。除了依照螞蟻覓食結果(解品質)分別留下不同量費洛蒙外，原本路徑上之費洛蒙也會隨時間流逝而逐漸揮發。更新的計算公式如下所示：

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \sum_{s=1}^S \Delta\tau_{ij}^s \quad (\text{公式 3.2})$$

其中， $\Delta\tau_{ij}^s$ ：第 s 隻螞蟻在路徑 ij 留下費洛蒙的量。

$$\Delta\tau_{ij}^s = \begin{cases} \frac{Q}{L_s} & \text{螞蟻 } s \text{ 曾使用過路徑 } ij \text{ 時} \\ 0 & \text{otherwise} \end{cases}$$

L_s ：第 s 隻螞蟻所建構出的路徑的目標值(如 makespan)。

Q ：為一大於零之常數(e.g. : 100)

步驟四、是否達到終止條件：

ACO 的停止條件通常設定為執行時間到達一時間上限或執行次數達到一次數上限值(t_{max})時停止，即 $t \geq t_{max}$ 。此時或此回合(step)所找到的最佳路徑(或稱加工順序)即為最佳解，目標值即為最佳解的績效值(如 makespan)。

3.5 蟻群最佳化演算法與不同解表達法的應用

因為解表達法的資料結構不同，所以應用於蟻群最佳化演算法中也有些許上的差異，本節將分成兩部份說明。

3.5.1 舊解表達法與蟻群最佳化演算法

舊表達法中，主要的特點在於明顯分成兩個部份，一部份是工件族的加工序(Family Sequences)，一部份是各個工件族內工件的加工序，兩份部的資訊處理是

獨立運作，所以運用於 ACO 中，也因此分成兩大類來進行。

以圖 3.8 來說明，首先看到解展開主要可以解分成如上述所說左右兩個區塊類別，其中右邊又可依照有幾個工件族分成獨立的幾個區塊。在圖中為三個工件族的例子，每個工件族區塊的資訊處理分別也是獨立運作。所以分離式表達法共可分為 $1+F$ 個資訊獨立的區堆， F 代表工件族個數。因為是獨立的，所以我們無法使用單獨一群螞蟻決定所有區塊的順序。因此我們得指派 $1+F$ 群螞蟻來分別執行 3.4 節所提之 ACO 步驟，決定各個區塊的順序後，最後才組合在一起，形成一組完整的解。有了工件族序及工件族內工件加工序資訊便可計算解的目標值。

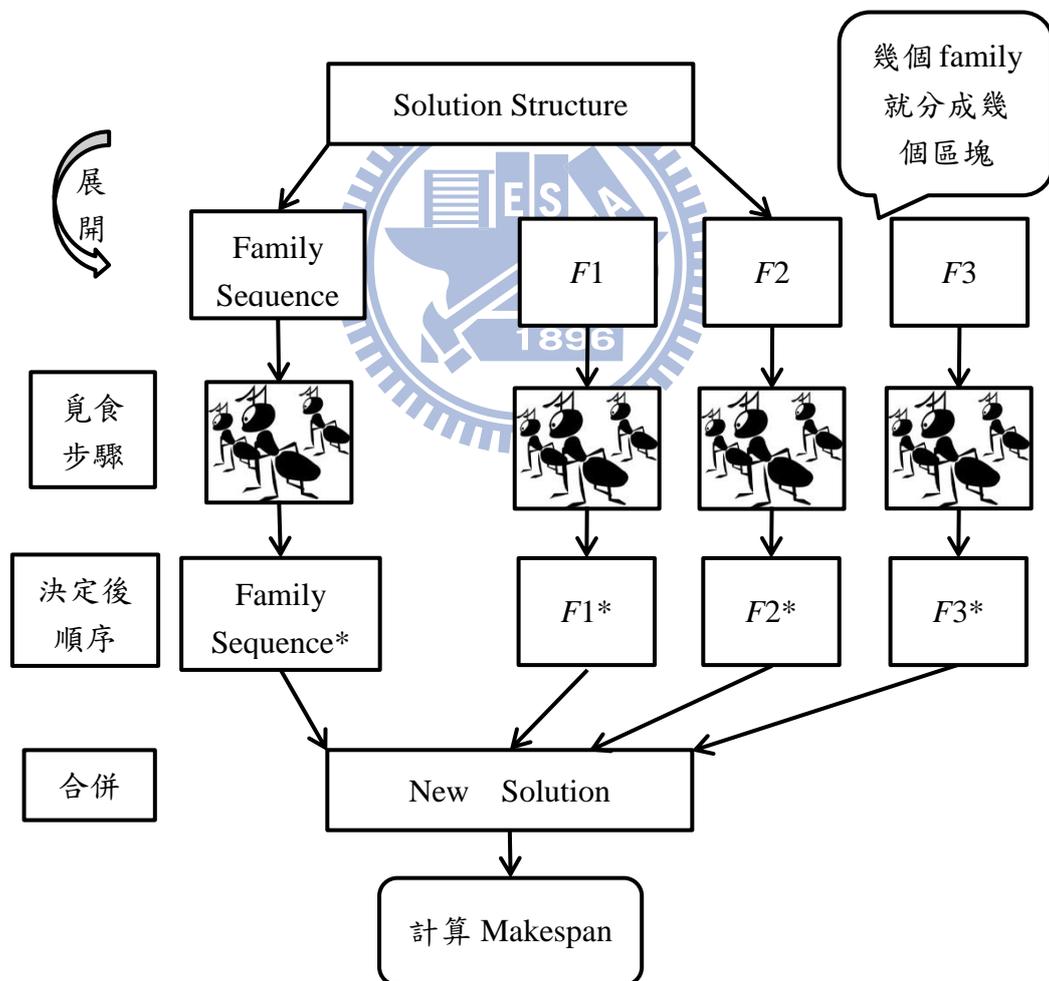


圖 3.8、ACO 產生舊解表達法之解與目標值流程圖，以三工件族為例

3.5.2 新解表達法與蟻群最佳化演算法

新解表達法不同於舊表達法的特點在於其資料結構存著混合的資訊，也就是工件族序與工件族內工件加工序無法由解的表面清楚看到。因為是混合的結構，所以解本身僅單一區塊。因此，應用於 ACO 上時，僅需一群螞蟻來建構屬於新表達法的解。建構完的解仍是單一的混合資料結構，在此得經過一串解碼過程方能得到用來計算目標值的資訊。利用 ACO 建立新表達法解的流程圖如圖 3.9 所示。

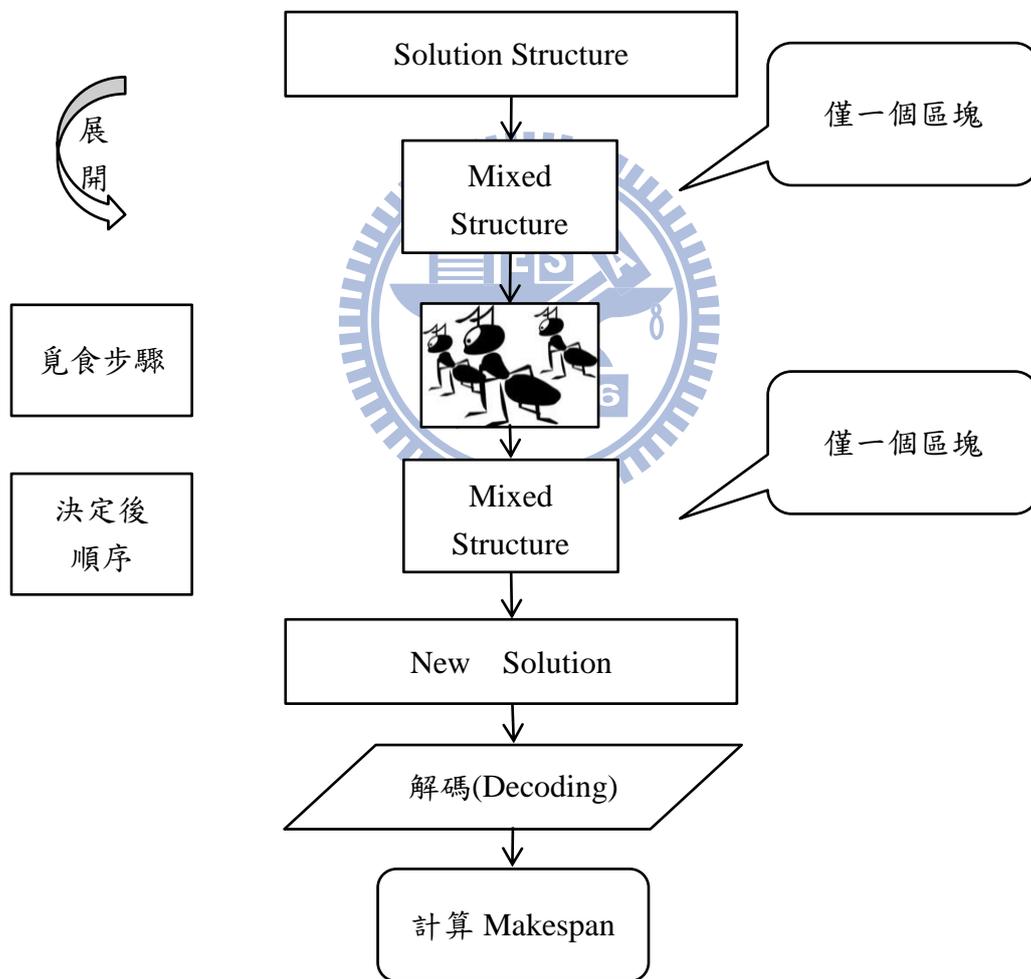


圖 3.9、ACO 產生新解表達法之解與目標值流程圖

第四章 實驗情境與結果

過去新解表達法已在與基因演算法(GA)搭配上表現的比舊解表達法好 (Wu *et al.* 2011), 為了驗證是否新解表達法也能在 ACO 上具一致性的表現, 所以本研究使用與 Schaller *et al.* (2000)相同的測試情境, 然後探討兩種表達的績效差異。以下將分別說明實驗情境及探討實驗結果。

4.1 實驗情境

本研究的實驗參數參照 Schaller *et al.* (2000), 其假設如下:

- (1) 工件在每部機台的處理時間服從離散型均勻分配 $U[1,10]$ 。
- (2) 有三種不同類別整備時間：
 - Small setups (SSU) : $U[1,10]$
 - Medium setups (MSU) : $U[1,50]$
 - Large setups (LSU) : $U[1,100]$
- (3) 每種整備時間下又分為 10 種情境組合：
 - $\{3, 3\}$ 、 $\{3, 4\}$ 、 $\{4, 4\}$ 、 $\{5, 5\}$ 、 $\{5, 6\}$ 、 $\{6, 5\}$ 、 $\{6, 6\}$ 、 $\{8, 8\}$ 、 $\{10, 8\}$ 、 $\{10, 10\}$
- (4) 工件族內的工件數為服從離散型均勻分配 $U[1,10]$ 。
- (5) 實驗基準
 - 每種情境組合皆跑 15 個相同的 seed
 - 每種情境組合皆跑 30 台不同的電腦, 產生 30 筆數據
 - 實驗結果為 30 筆數據的平均
 - 實驗總次數 : $3(\text{整備時間}) \times 10(\text{組合}) \times 15(\text{seeds}) \times 30(\text{電腦}) \times 2(\text{表達法}) = 27000(\text{次})$

4.2 ACO 參數設定

相關演算法參數設定如下：

- (1) 費洛蒙蒸發率 $\rho = 0.8$ 。
- (2) 人工螞蟻個數 $S = \binom{n}{2}$ ， n 為節點總個數。
- (3) 路徑選擇機率常數 $(\alpha, \beta) = (1, 2)$ 。
- (4) 期間常數(或稱可執行總次數) t_{max} : 連續 20000 代最佳解不變。

參數設定方面，費洛蒙蒸率及期間常數對於求解的影響較大，所以參數值需要特別考量。兩者考量分述如下。

4.2.1 費洛蒙蒸發率(ρ)

為了找尋一個較好的費洛蒙蒸發率，在其它參數為固定值的情況下，本研究實驗了 3 種比率：0.3、0.5 及 0.8。此三種蒸發率可以代表小、中及大三個方向的設定，其實驗結果如表 4.1。表中 C_n 及 C_t 分別代表新解表達法及舊解表達法求出的平均最大完工時間(Makespan)， $Ave.(%)$ 代表平均改善率， $T_n(sec)$ 及 $T_t(sec)$ 代表新解表達法及舊解表達法之求解時間， $Gap(sec)$ 代表求解時間的差值 ($T_n(sec) - T_t(sec)$)。從表中可以見到，在三種不同的蒸發率下，對解品質影及求解時間響差異並不大，代表蒸發率的設定不會重大影響兩解表達法與演算法配合間的差異。但由於 0.8 設定能讓兩種解表達法求出較佳的解，所以在本研究中設立費洛蒙為 0.8。

表 4.1、三種費洛蒙蒸發率實驗結果

Average of all Scenarios (Evaporation rate = 0.3)					
C_n	C_t	Ave. (%)	$T_n(see)$	$T_t(sec)$	Gap(sec)
404.62	404.79	0.11	62.30	20.54	-41.76
Average of all Scenarios (Evaporation rate = 0.5)					
C_n	C_t	Ave. (%)	$T_n(see)$	$T_t(sec)$	Gap(sec)
404.55	404.77	0.12	62.45	20.56	-41.89
Average of all Scenarios (Evaporation rate = 0.8)					
C_n	C_t	Ave. (%)	$T_n(see)$	$T_t(sec)$	Gap(sec)
404.48	404.68	0.11	62.66	20.23	-42.43

4.2.2 期間常數(t_{max})

本研究中，設定連續多代最佳解不變為期間常數，次數的設定會影響到演算法全域搜尋的能力，太小容易陷入區域最佳解，太大使求解時間過度拉長而降低解品質精進邊際效益(Margin value)，因此，期間常數的設定需同時考量求解時間及解品質。

從表 4.2 及表 4.3 中可以見到，隨著收斂代數的增加，兩種表達法的求解品質也跟著變好，求解時間也跟著拉長。然而，隨著收斂代數的倍數成長，求解時間跟著倍數拉長，解品質的進幅度卻差不多。就解品質精進與時間的相對邊際效益而言，收斂代數以 10000 較適合。因此，本研究中期間常數設定為 10000。

表 4.2、不同的收斂代數下實驗結果

收斂代數對解品質的影響		
收斂代數	舊表達法	新表達法
300	948.60	969.07
1000	941.53	957.80
2000	939.80	955.80
5000	936.13	949.13
10000	931.73	940.07
50000	920.60	930.40

表 4.3、不同的收斂代數下求解時間

收斂代數對求解時間的影響		
收斂代數	舊表達法	新表達法
300	2.78	8.39
1000	8.23	27.46
2000	15.19	52.11
5000	40.47	131.11
10000	75.87	275.51
50000	375.51	1154.45

4.3 測試環境

求解測試環境與設備資訊如下：

- (1) 使用語言：C++
- (2) 編譯器(Compiler)：MSVC9.0
- (3) 編寫環境(IDE)：Visual Studio 2008
- (4) 使用作業系統(OS)：Windows 7 Enterprise 32-bits
- (5) 記憶體(Memory)：4G
- (6) 中央處理器(CPU)：AMD Athlon(tm) II X4 640 Processor 3.00 GHz

4.4 實驗結果

本研究中定義 obj_{old} 和 obj_{new} ，分別為舊解表達法及新解表達法之各別目標函式值。實驗結果如表 4.1 所示，以最大完工時間(makespan)為指標。實驗結果表格的說明如下：

- (1) Scenario 指出家族相依整備時間的大小，共分成三類：SSU、MSU、LSU，

其後所接的數值代表工件族數及加工機台數。

- (2) Makespan 欄為解品質差異的相關統計，詳細子欄說明如下：

- N ：代表每種情境組合下總共跑出的資料筆數(總次數)。
- N_e ：兩解種表達法求解出目標值平手的次數。
- N_w ：舊解表達法求解出目標值優於整合式表達法的次數。
- C_n ：每種情境下新解表達法所求出的平均目標值。
- C_t ：每種情境下舊解表達法所求出的平均目標值。
- $Avg. (\%)$ ：每種情境下，30 筆數值的平均改善率，以%表示。
- 每筆資料的改善率(r)公式，其中 obj_{old} 及 obj_{new} 分別為舊解表達法及新解表達法應用在 ACO 上求出的目標值：

$$r = \frac{(obj_{old} - obj_{new})}{obj_{old}}$$

(3) Computation Time 欄為求解各項情境所花時間的統計資料，分述如下：

- $T_n(sec)$ ：新解表達法應用於 ACO 產生 30 筆資料的平均求解時間，以秒表示。
- $T_t(sec)$ ：舊解表達法應用於 ACO 產生 30 筆資料的平均求解時間，以秒表示。
- Gap ：為求解時間上的差值，公式如下：

$$Gap = T_t(s) - T_n(s)$$

4.5 比較結果分析

關於結果分析，下面將分成兩個部份來說明，第一部份以資料的數值分析來說明，第二部份以統計檢定結果來說明。

4.5.1 數值統計分析結果

從表 4.1 的最下方平均值來看，我可以見到新解表達法配上 ACO 的求解速度比舊解表達法來的慢，數據平均顯示慢了約 42.3 秒，但求解品質比較好，平

均 30 次中，有高達 20 次不輸給舊解表達法。以人類的反應速度來說，多約 43 秒的時間卻可以求得比較好的解，是可以接受的。

在圖 4.1 中顯示在大部份情境下，新解達法的應用相較於舊解表達法的表現，皆能有所改善(改善率大於 0)；在整備時間不大的情境(SSU 及 MSU)下，改善率較高。在圖 4.2 中顯示在中等情境下，新解表達法比較能贏舊解表達法。在圖 4.3 中顯示在小情境下(機台數及工件族數不多)，兩種表達法容易平手。在圖 4.4 中顯示隨著工件族數及機台數不大的情境下，新解表達法表現較佳。

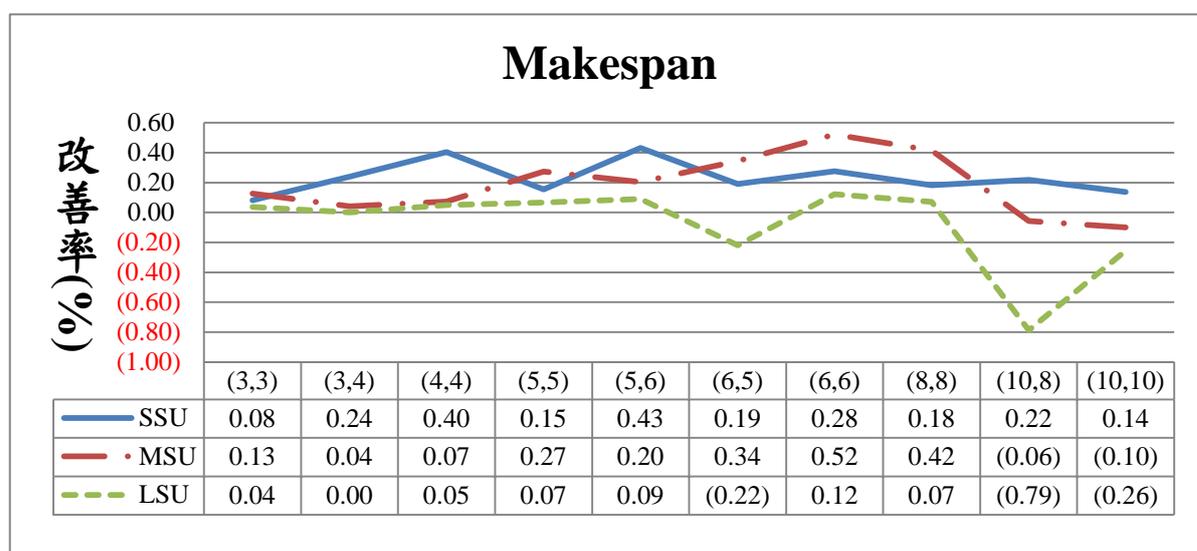


圖 4.1、各種情境下新解表達法之改善率

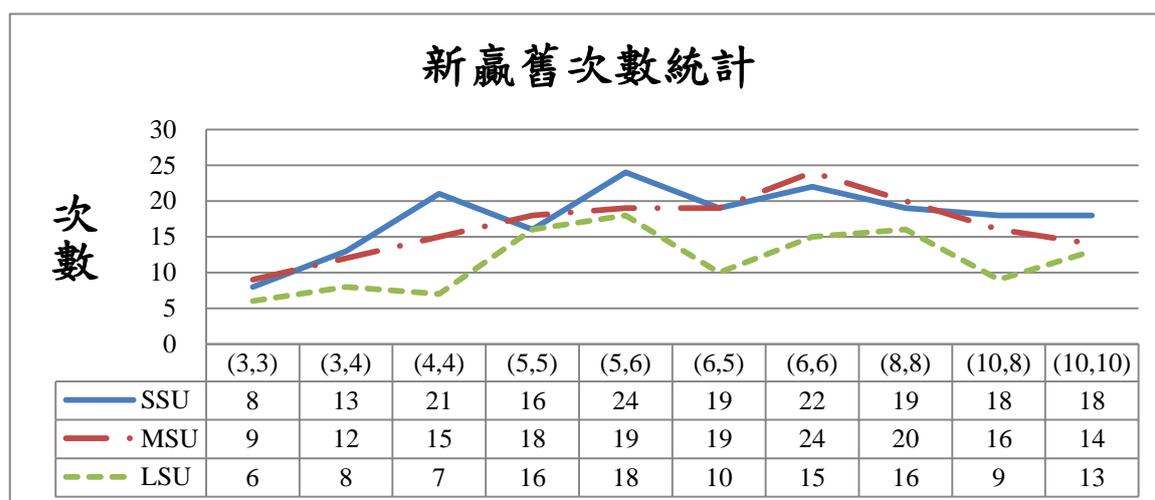


圖 4.2、新解表達法比舊解表達好的次數統計

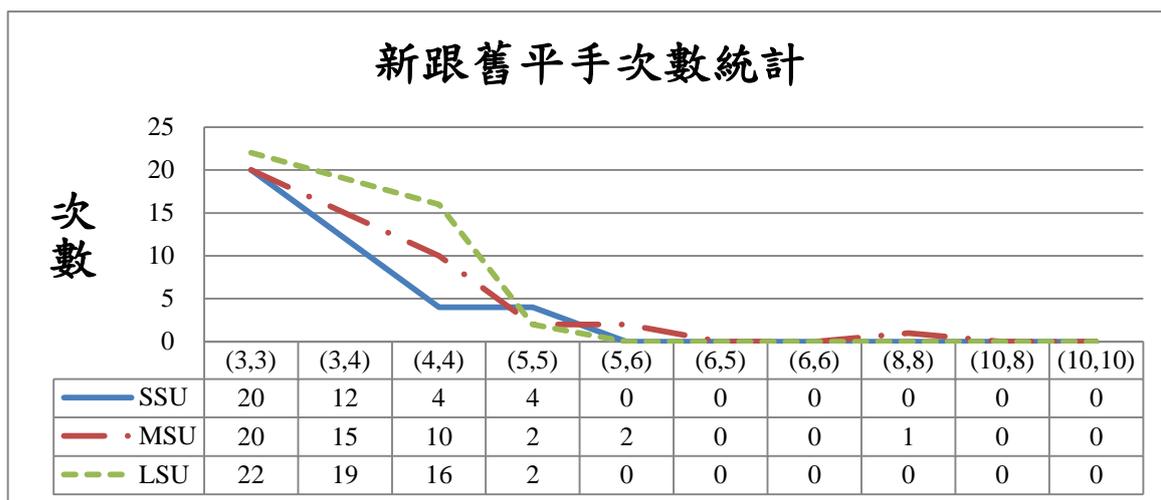


圖 4.3、兩種解表達法平手的次數統計

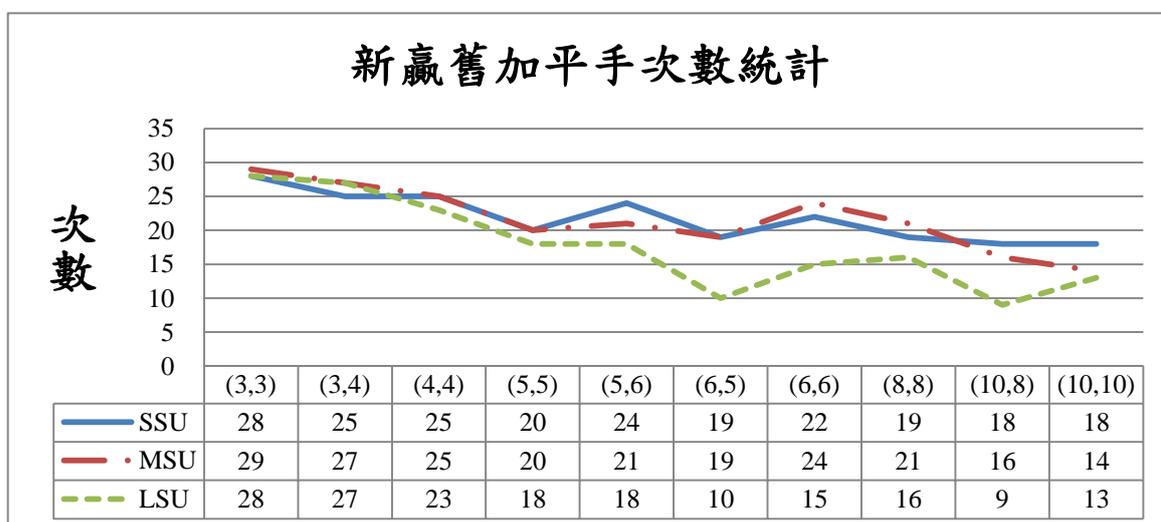


圖 4.4、新解表達法不輸給舊解表達法之次數統計

4.5.2 統計檢定結果

為了確認兩種表達法在 ACO 的應用表現上，是否有顯著的差異，我們利用 Paired Sample t-test 檢定方法來驗證。檢定分成個別檢定在 SSU、MSU、LSU 三種整備時間情境下及整體的表現。檢定的方法為檢定新解表達法相對於舊解表達法好的表現是否顯著，以相對改善率為樣本資料。為了能做整體解表達法表現的檢驗，檢定前需先將資料進一步正規化。假設 $S: \{SSU33, SSU34, \dots, MSU33, MSU34, \dots, LSU33, LSU34, \dots, LSU1010\}$ 代表所有 30 個情境的集合，而

$u_{new,s,i}$ 及 $u_{old,s,i}$ 分別代表在情境 s 下第 i 個實驗新、舊解表達法的解平均，其中 $s \in S, i = 1, 2, \dots, 30$ (每個情境 30 個實驗)。在情境 s 下的第 i 個正規化樣本資料 $d_{s,i}(\%) = (u_{old,s,i} - u_{new,s,i}) * 100 / u_{old,s,i}$ ，詳細資料如附錄 A 所示。

不同整備時間情境及整體的樣本平均：

$$\bar{d}_k(\%) = \frac{\sum_{s \in k} \sum_{i=1}^{30} d_{s,i}}{300}, k \in \{SSU, MSU, LSU\}、\bar{d}_{Total}(\%) = \frac{\sum_{s \in S} \sum_{i=1}^{30} d_{s,i}}{900}。$$

不同整備時間情境及整體的樣本標準差：

$$SD_k = \sqrt{\frac{\sum_{s \in k} (d_{s,i} - \bar{d}_k)^2}{299}}, k \in \{SSU, MSU, LSU\}、$$

$$SD_{Total} = \sqrt{\frac{\sum_{s \in S} (d_{s,i} - \bar{d}_{Total})^2}{899}}。$$

不同整備時間情境及整體檢定用 t 值：

$$t_k = \frac{\bar{d}_k}{SD_k / \sqrt{300}}, k \in \{SSU, MSU, LSU\}、t_{Total} = \frac{\bar{d}_{Total}}{SD_{Total} / \sqrt{900}}。$$

根據公式，可得表 4.5。表中分別列示 LSU、MSU、SSU 及整體檢定結果。結果顯示在 95% 信心水準 ($\alpha = 0.05$) 下，在大整備時間情境下，兩種解表達法的表現沒有顯著差異；在中、小整備時間情境下，新解表達法優於舊解表達法的表現且具有顯著的差異；以整體表現來說，新解表達法的表現贏過舊解表達法，且具有顯著的差異。

表 4.4、實驗數據平均結果

ID	Scenario	Makespan							Computation Time(sec)		
		N	N_w+Ne	N_e	N_w	Cn	Ct	$Ave. (%)$	Tn	Tt	Gap
1	SSU33	30	28	20	8	134.53	134.64	0.08	2.97	1.63	-1.34
2	SSU34	30	25	12	13	152.00	152.33	0.24	4.13	2.46	-1.67
3	SSU44	30	25	4	21	187.45	188.19	0.40	8.07	4.07	-4.00
4	SSU55	30	20	4	16	246.35	246.73	0.15	16.41	6.89	-9.51
5	SSU56	30	24	0	24	258.99	260.11	0.43	17.83	7.53	-10.29
6	SSU65	30	19	0	19	291.98	292.53	0.19	26.62	10.00	-16.62
7	SSU66	30	22	0	22	302.68	303.56	0.28	26.81	10.64	-16.17
8	SSU88	30	19	0	19	419.05	419.90	0.18	53.61	16.97	-36.64
9	SSU108	30	18	0	18	506.32	507.42	0.22	93.10	25.53	-67.57
10	SSU1010	30	18	0	18	550.38	551.06	0.14	94.99	27.65	-67.34
11	MSU33	30	29	20	9	160.12	160.32	0.13	3.06	1.72	-1.33
12	MSU34	30	27	15	12	185.11	185.20	0.04	4.32	2.45	-1.88
13	MSU44	30	25	10	15	238.42	238.64	0.07	11.31	5.28	-6.03
14	MSU55	30	20	2	18	309.14	309.93	0.27	25.62	10.49	-15.13
15	MSU56	30	21	2	19	321.53	322.24	0.20	24.81	10.65	-14.16
16	MSU65	30	19	0	19	367.65	368.90	0.34	42.85	16.05	-26.80
17	MSU66	30	24	0	24	386.41	388.44	0.52	44.33	16.91	-27.42
18	MSU88	30	21	1	20	531.28	533.63	0.42	90.55	28.86	-61.68
19	MSU108	30	16	0	16	657.39	657.27	-0.06	166.36	45.50	-120.86
20	MSU1010	30	14	0	14	689.10	688.47	-0.10	152.08	46.41	-105.68
21	LSU33	30	28	22	6	228.13	228.22	0.04	4.26	2.33	-1.93
22	LSU34	30	27	19	8	239.20	239.22	0.00	5.07	2.88	-2.19
23	LSU44	30	23	16	7	323.88	324.03	0.05	14.20	6.25	-7.95
24	LSU55	30	18	2	16	418.90	419.17	0.07	38.93	15.75	-23.18
25	LSU56	30	18	0	18	441.01	441.41	0.09	42.59	17.56	-25.03
26	LSU65	30	10	0	10	496.92	495.90	-0.22	79.48	26.77	-52.71
27	LSU66	30	15	0	15	519.83	520.50	0.12	71.55	25.77	-45.79
28	LSU88	30	16	0	16	719.10	719.71	0.07	153.06	52.26	-100.80
29	LSU108	30	9	0	9	895.01	888.54	-0.79	292.62	81.16	-211.46
30	LSU1010	30	13	0	13	956.47	954.18	-0.26	268.18	78.56	-189.62
Average		30	20.37	4.97	15.40	404.48	404.68	0.11	62.66	20.23	-42.43

表 4.5、檢定結果

Scenario	平均改善率(%)	t_0	$t_{\alpha=0.05}$	檢定結果
LSU	-0.08	1.59	1.96	沒有顯著
MSU	0.18	4.51	1.96	顯著贏
SSU	0.23	6.39	1.96	顯著贏
Total	0.11	4.39	1.96	顯著贏

第五章 結論與未來研究

本論文探討流線型製造單元的排程問題，以製造業中插件機加工過程為背景，其特點在於整備時間較單一工件加工時間長，因此需以工件分族(Family-based)的方式加工，所以排程上同時需考慮工件族間(Among Job Family)與工件族內(Within Job Family)工件的加工順序。

在 Tai *et al.* (2010)及 Wu *et al.* (2011)中，提出以新解達法搭配基因演算法來求解流線型製造單元排程問題，其結果顯示在相同的演算流程下，新解表達法相較於過去常用的舊解達法表現的好，但卻無法證明新解表達法在其它的巨集啟發式演算法也具有一致性的好。所以本研究以蟻群最佳化演算法搭配新、舊解表達法求解製造單元排程問題，看看新解達法是否在 ACO 上也具有一致性的好。

本研究中，有針對 ACO 的費洛蒙蒸發率及停止條件做前測實驗，此部份提供未來的相同排程問題研究或求解此類排程問題時一個參數選擇的方針。

最後的實驗結果顯示，新解表達法在平均上的表現贏過舊解表達法，在小整備時間情境下表現較優，在 95%信水準下統計顯示新解表達法在任何整備時間情境下，皆不輸給舊解表達法，整體表現顯著贏過舊表達法，表示新解表達法較適合用來求解流線製造單元排程，也驗證了改善解表達法確實可以增進求解的品質。

本研究中，費洛蒙路徑更新上採用單軌設計，也就是兩點間來回的費洛蒙量是相同的，此部份目前無法顯兩點間的順序關係，僅能顯示兩點相鄰關係的程度，所以未來可以採雙軌的方式來進行費洛蒙的更新，也就是來、回分成不同的路徑。此外，本研究中，起始點的選擇是採隨機的方式來選擇，此部份可以進一步加入虛擬起始點來加強起始點選擇的合理性，利用虛擬起始點跟可能的起始點間虛擬費洛蒙路徑的建立，加強路徑起始點選擇的收斂性。

本研究中，採用完工時間導向(Completion-time based)的最大完工時間(makespan)當指標值，未來可能嘗試採用交期導向(Due-day based)的相關指標(如

tardiness)來看看新表達法能否在與 ACO 搭配上一樣具有一致性的表現。

本論文主要注重於具固定序列特性的排程問題。過去較少有研究注重於非固定序列(Non-permutation)的排程問題，因此考慮加入非固定序列特性的排程問題是未來可以研究的方向之一。



參考文獻

- Chang, P. C., Chen, S. H., Fan, C. Y., & Mani, V. (2008). Generating artificial chromosomes with probability control in genetic algorithm for machine scheduling problems. *Annals of Operations Research*, 1-15.
- Dorigo, M., & Gambardella, L. M. (2002a). Ant colony system: A cooperative learning approach to the traveling salesman problem. *Evolutionary Computation, IEEE Transactions on*, 1(1), 53-66.
- Dorigo, M., Maniezzo, V., & Colorni, A. (2002b). Ant system: optimization by a colony of cooperating agents. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 26(1), 29-41.
- França, P. M., Gupta, J. N. D., Mendes, A. S., Moscato, P., & Veltink, K. J. (2005). Evolutionary algorithms for scheduling a flowshop manufacturing cell with sequence dependent family setups. *Computers & Industrial Engineering*, 48(3), 491-506.
- Gajpal, Y., & Rajendran, C. (2006). An ant-colony optimization algorithm for minimizing the completion-time variance of jobs in flowshops. *International Journal of Production Economics*, 101(2), 259-272.
- Hamed Hendizadeh, S., Faramarzi, H., Mansouri, S. A., & Gupta, J. N. D. (2008). Meta-heuristics for scheduling a flowline manufacturing cell with sequence dependent family setup times. *International Journal of Production Economics*, 111(2), 593-605.
- Lin, S. W., Gupta, J. N. D., Ying, K. C., & Lee, Z. J. (2009a). Using simulated annealing to schedule a flowshop manufacturing cell with sequence-dependent family setup times. *International Journal of Production Research*, 47(12), 3205-3217.
- Lin, S. W., Ying, K. C., & Lee, Z. J. (2009b). Metaheuristics for scheduling a non-permutation flowline manufacturing cell with sequence dependent family setup times. *Computers & Operations Research*, 36(4), 1110-1121.
- Merkle, D., & Middendorf, M. (2005). On solving permutation scheduling problems with ant colony optimization. *International Journal of Systems Science*, 36(5), 255-266.
- Schaller, J. E., Gupta, J. N. D., & Vakharia, A. J. (2000). Scheduling a flowline manufacturing cell with sequence dependent family setup times. *European Journal of Operational Research*, 125(2), 324-339.
- Wu, M. C., Tai, P. H., & Chiou, C. W. (2011). A Comparison of Two Chromosome Representation Schemes Used in Solving a Family-Based Scheduling Problem. *to be presented in International Conference of Flexible Automation &*

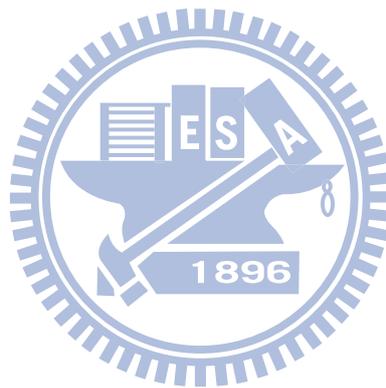
Intelligent Manufacturing (FAIM), June, 2011, Taiwan.

Yagmahan, B., & Yenisey, M. M. (2010). A multi-objective ant colony system algorithm for flow shop scheduling problem. *Expert Systems with Applications*, 37(2), 1361-1368.

呂佳玟，「應用基因演算法與家族式派工於傳輸整合步進機在小批量情境下之排程問題」，交通大學，碩士論文，2009。

馮正民、邱裕鈞，研究分析方法，1版，新竹市：建都文化，2004。

戴邦豪，「應用混合式染色體表達法於具順序相依家族整備時間之流線型製造單元排程」，交通大學，碩士論文，2010。



附錄一、各個情境下 $d_{s,i}(\%)$ 值計算表格

$d_{s,i}(\%)$	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17	C18	C19	C20	C21	C22	C23	C24	C25	C26	C27	C28	C29	C30
LSU33	0.15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.29	0.00	0.10	0.00	0.00	0.00	0.00	0.03	-0.26	0.00	0.00	0.00	0.00	0.75	-0.06	0.00	0.00	0.14	0.00	0.00	
LSU34	0.00	0.00	0.00	0.00	0.03	0.06	0.00	0.00	0.00	0.00	0.00	-0.09	0.00	0.00	0.00	0.00	0.17	0.17	0.02	0.09	0.35	0.00	-0.38	0.00	-0.44	0.08	0.00	0.00	0.00	
LSU44	0.00	0.39	-0.02	0.00	0.00	0.00	0.37	-0.04	0.53	-0.34	0.00	0.00	0.00	-0.25	0.06	0.00	0.24	0.00	0.02	0.00	0.00	0.71	-0.06	0.00	0.00	0.00	-0.04	-0.09	0.00	0.00
LSU55	-0.36	-0.13	0.10	-0.94	-0.17	-0.45	0.51	0.25	0.26	0.23	0.11	0.00	0.44	0.09	0.00	0.56	0.43	0.44	-0.48	1.01	-0.13	1.06	-0.32	-0.05	-0.30	0.09	-0.22	0.30	-0.45	0.13
LSU56	0.03	0.26	-0.18	0.76	0.47	-0.02	-0.54	-0.25	0.06	0.75	-0.50	-0.12	0.09	-0.05	-0.24	0.01	1.03	0.76	-0.48	-1.15	0.09	-0.91	-0.20	0.72	0.49	0.33	0.46	0.85	0.13	0.06
LSU65	0.64	-0.20	0.48	-0.45	-0.03	0.28	-0.24	-0.31	-0.21	-0.15	0.45	-1.01	1.10	-0.63	-1.15	0.04	-1.27	-1.43	0.50	0.67	-0.04	0.13	-0.33	0.24	-0.50	-0.76	-0.33	-0.74	-0.75	-0.59
LSU66	0.01	-0.05	-0.05	-0.39	-0.29	0.64	-0.18	-0.04	0.68	-0.83	-0.32	0.15	0.25	-0.20	-0.53	-0.66	-0.41	0.96	-0.57	1.18	0.21	-1.58	0.42	0.19	0.18	1.04	-0.28	0.71	-0.14	2.55
LSU88	-0.90	0.44	1.11	-1.13	-2.06	-1.00	-0.18	-0.16	0.13	-0.16	0.36	0.32	-0.41	3.82	0.22	-0.39	1.54	0.28	0.85	-1.07	-1.48	0.21	-1.16	0.31	0.51	1.00	0.54	-0.67	1.63	-0.32
LSU810	-0.19	0.65	-0.30	-3.18	-3.70	0.70	-1.93	-1.53	1.31	-2.61	-2.60	-0.97	0.09	-0.14	-0.62	2.32	-0.88	-3.72	2.58	1.98	-1.05	0.96	-0.35	-0.64	-1.28	-5.47	-1.56	-2.45	1.37	-0.39
LSU1010	-0.93	1.33	-0.84	-0.46	-0.52	0.05	0.14	0.60	0.31	1.37	-2.29	-0.45	-1.63	-1.49	-0.51	-1.48	0.14	1.04	-2.16	0.13	-0.70	-0.95	-0.17	0.27	-0.42	1.98	-1.67	-0.44	1.06	1.04
MSU33	0.00	0.00	0.00	0.29	0.00	0.00	0.00	0.20	0.07	0.00	0.32	0.00	0.00	0.63	0.00	0.00	-0.31	0.00	0.00	0.00	0.00	0.56	0.00	0.75	1.14	0.00	0.15	0.00	0.00	
MSU34	0.00	0.00	-0.38	0.00	0.03	-0.47	0.00	0.00	0.00	0.04	0.00	0.54	0.00	0.00	0.15	0.29	0.00	0.03	0.00	0.16	0.04	0.00	-0.42	0.31	0.29	0.11	0.00	0.55	0.00	0.00
MSU44	0.00	0.14	-0.09	0.00	0.00	0.00	-0.50	0.00	0.00	0.00	-0.84	0.19	0.32	0.23	0.59	0.06	0.53	0.06	-1.24	0.00	-0.06	0.91	0.04	0.00	0.00	0.39	0.60	0.05	0.28	0.52
MSU55	-0.84	1.20	-0.67	0.00	0.84	0.14	0.18	0.51	-0.21	0.37	-0.02	2.38	0.00	0.48	1.71	-0.92	0.34	-0.56	1.82	0.31	-0.23	-0.38	0.54	-0.39	0.06	0.02	0.42	0.88	-0.14	0.41
MSU56	-0.02	-0.05	0.54	0.19	-0.61	-0.09	0.14	-0.73	0.53	0.58	0.02	0.50	1.60	1.20	0.57	0.00	0.25	0.00	-0.90	0.12	0.18	-0.02	-0.42	0.30	-0.11	1.43	0.23	0.41	0.27	0.02
MSU65	-0.36	-0.17	-0.54	0.30	0.48	0.02	0.13	-0.31	2.53	0.40	-0.11	0.29	0.55	3.16	0.28	0.97	-0.17	-0.15	0.14	-0.33	0.59	-1.07	0.38	0.43	1.41	0.72	-0.05	-0.31	-0.21	1.06
MSU66	1.23	0.58	0.59	-0.45	0.94	-0.54	0.68	2.29	0.86	0.54	-0.48	-0.35	0.20	1.84	0.47	-0.83	0.19	0.43	0.99	0.48	0.32	0.27	0.25	-0.37	0.16	0.51	-0.37	0.16	1.99	1.41
MSU88	-1.54	0.79	0.00	-1.33	0.66	0.55	0.37	1.53	0.03	-0.02	-0.74	0.08	1.08	1.26	0.97	0.80	0.39	2.28	3.21	-0.34	-0.28	0.10	0.21	0.42	-0.50	0.07	-0.22	-0.12	1.17	1.57
MSU810	-0.48	0.54	-0.50	-0.56	-1.78	0.21	0.01	-0.48	-0.53	-0.72	-0.46	0.49	2.47	1.13	0.06	-1.33	0.63	-0.85	-0.65	-1.22	-0.83	-1.79	0.12	0.50	0.11	1.31	1.22	0.75	0.83	0.09
MSU1010	-0.59	-0.09	0.18	-0.05	-0.21	0.48	0.35	-0.39	-1.38	-1.16	-0.11	0.93	-0.70	0.85	-1.65	0.12	-0.13	-1.15	0.03	-0.33	0.61	-0.63	1.05	0.03	0.07	-0.67	-0.44	0.74	0.93	0.34
SSU33	0.00	0.00	0.05	0.00	-0.09	0.00	0.00	0.10	0.00	0.00	0.71	0.72	0.00	-0.05	0.20	0.00	0.00	0.00	0.13	0.00	0.00	0.00	0.00	0.37	0.00	0.00	0.00	0.00	0.33	0.00
SSU34	0.00	0.00	0.13	0.61	0.00	0.00	1.56	0.00	0.16	1.14	0.00	0.07	1.41	0.00	0.05	0.19	0.26	0.00	0.00	0.40	-0.20	1.19	0.00	1.07	-0.04	0.00	-0.11	-0.22	0.00	-0.46
SSU44	0.32	1.32	0.84	0.56	1.35	0.05	0.00	0.95	0.95	-0.12	0.00	0.16	0.42	0.77	0.25	0.63	0.21	0.39	-0.44	1.07	0.31	0.12	-0.19	-0.18	0.00	0.00	1.09	1.18	0.20	-0.07
SSU55	0.94	-0.02	0.73	-0.57	0.12	0.05	-0.03	0.25	0.07	-0.68	0.00	-0.08	0.43	0.89	-0.30	0.30	0.19	2.51	0.09	-0.79	0.00	1.50	-0.97	0.05	0.05	-0.23	0.00	0.28	0.00	-0.16
SSU56	0.72	0.61	0.46	1.40	1.04	0.16	-0.08	1.88	0.58	0.64	0.71	0.05	0.02	-0.23	0.05	0.12	1.56	0.40	-0.07	0.43	0.88	0.49	-0.07	0.40	0.12	0.81	0.37	0.50	-0.65	-0.33
SSU65	-0.60	-0.03	-0.73	1.51	-0.06	0.03	-0.62	-0.29	0.77	0.69	0.74	0.07	0.03	0.41	-0.87	1.06	0.61	0.20	0.06	1.07	-2.16	0.40	-0.50	-0.02	-0.14	1.34	1.12	0.55	0.51	0.61
SSU66	0.42	0.27	-0.07	-0.80	0.21	0.73	0.15	0.70	0.54	0.07	0.77	1.30	1.05	-0.06	-0.19	-0.12	0.42	-0.92	0.12	-0.50	0.18	-0.08	0.12	0.45	0.45	0.71	1.46	0.47	0.26	0.18
SSU88	0.40	0.88	0.38	-0.85	0.73	0.20	0.96	0.33	0.56	0.25	1.11	-2.16	-0.68	-0.52	-0.70	1.40	-1.05	0.03	1.28	-0.04	0.32	1.33	-0.37	-0.84	0.84	-0.85	-0.46	1.29	0.55	1.15
SSU810	-0.56	0.67	0.64	-0.89	-0.13	-0.52	-0.46	-1.06	0.26	0.50	0.44	0.29	0.40	-0.59	-0.34	-0.60	1.40	1.57	0.37	0.01	-0.43	1.89	1.26	0.51	0.25	0.93	0.55	0.96	-0.38	-0.38
SSU1010	1.93	-0.54	0.36	-0.11	0.40	0.71	-0.61	-0.37	-0.33	0.10	0.07	1.14	0.34	0.41	0.30	-0.67	-0.81	1.58	0.34	0.04	0.91	-0.05	-0.25	0.53	0.30	0.23	-0.87	-0.98	-0.33	0.39