

國立交通大學
工業工程與管理學系

碩士論文

以瀾集演算法求解流線型製造單元排程



A Memetic Algorithm Approach to
Scheduling Flowshop Manufacturing Cell

研究生：潘冠銘

指導教授：巫木誠 博士

中華民國一百年六月

以瀾集演算法求解流線型製造單元排程

A Memetic Algorithm Approach to
Scheduling Flowshop Manufacturing Cell

研 究 生：潘冠銘

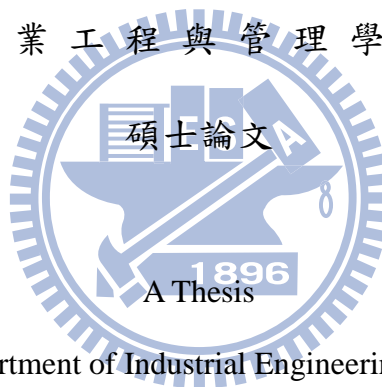
Student：Kuan-Ming Pan

指 導 教 授：巫木誠 博士

Advisor：Dr. Muh-Cherng Wu

國 立 交 通 大 學

工 業 工 程 與 管 理 學 系



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

以瀾集演算法求解流線型製造單元排程

研究生：潘冠銘

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

瀾集演算法是一種廣泛被使用拿來求解複雜空間搜尋問題的巨集啟發式演算法。對於各種問題，過去的研究往往著重在對於演算法進化部分的改善，本研究不同於以往的思維去嘗試一種新的研究議題，採用新的染色體表達法來改變瀾集演算法的績效。

本研究對於求解具有固定序列的流線型製造單元排程問題利用瀾集演算法搭配兩種不同的染色體表達法去進行比較；這兩種瀾集演算法具有相同的演算流程，但具有不同的染色體表達方式，一個是過去研究所使用的，本研究稱之為舊染色體表達法(S_{old})；另一個則是使用 Wu *et al.* (2011)所發明的，本研究將其稱為新染色體表達法(S_{new})。經由大量的數據分析後發現本研究所使用的瀾集演算法搭配新染色體表達法($Memetic-S_{new}$)大部分都優於瀾集演算法搭配舊染色體表達法($Memetic-S_{old}$)。此結果顯示一個新的重要研究方向，利用巨集啟發式演算法於不同空間求解問題時，新的染色體表達法可以改善空間求解問題的績效。

關鍵詞：瀾集演算法、染色體表達法、排程

A Memetic Algorithm Approach to Scheduling Flowshop Manufacturing Cell

Student : Kuan-Ming Pan

Advisor : Dr. Muh-Cherng Wu

Department of Industrial Engineering and Management
National Chiao Tung University

Abstract

The memetic algorithm, 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 the search mechanism 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 the memetic algorithm? A flowshop manufacturing cell scheduling problem with permutation is used as the problem context, and two memetic algorithms 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 *Memetic- S_{old}* and *Memetic- S_{new}* accordingly. Extensive numerical experiments reveal that *Memetic- S_{new}* mostly outperforms *Memetic- S_{old}* . This finding highlights an important new research track—exploring new solution representation schemes while applying meta-heuristic algorithms to various space-search problems.

Keywords: memetic algorithm, solution representation, scheduling

誌謝

本論文得以順利完成，首要感謝的是我的指導教授 巫木誠博士的敦敦教誨。除了在學術上不斷提供意見以及指導外，巫老師時常分享自己待人處事的經驗，強調誠信的重要，以及用有效率的方法去解決問題。在老師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，要感謝同實驗室的同學李奕勳、曾偉杰、林耿漢、陳威宇以及學弟妹林佳慧、徐千雯、彭勇漢、黃建翰、賴佳卉、譚浩，陪我度過修課與撰寫論文的的日子，時常相互鼓勵與討論。還有一起陪我渡過無數個夜晚的弟兄連德凱、侯博鈞、朱冠衡、洪正哲和盧柏翔學長，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，對於我的研究中許多困難的解決，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。特別是我的外公外婆楊宗槐先生與楊林蘭女士和我的母親楊麗蓉女士以及我的大舅魏賢文先生，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。還有感謝我的女朋友梁雅筑小姐，謝謝你對我的支持以及陪伴。在此，謹以此論文獻給我最敬愛的家人、師長與朋友。

潘冠銘 于 新竹交大

2011'7'1

目錄

中文摘要.....	ii
Abstract.....	iii
誌謝.....	iv
表目錄.....	vii
圖目錄.....	viii
第一章 緒論.....	1
1.1 研究背景.....	1
1.2 研究目的.....	2
1.3 研究範疇.....	2
1.4 研究流程與架構.....	3
第二章 文獻探討.....	5
2.1 排程問題之概論.....	5
2.1.1 排程問題之分類.....	5
2.1.2 排程問題之求解方法.....	6
2.1.3 排程績效衡量指標.....	8
2.2 流線型生產製造單元排程問題.....	9
2.3 固定序列(Permutation)與非固定序列(Non-Permutation).....	10
2.4 探討染色體表達法的差異.....	11
2.5 瀾集演算法.....	11
第三章 研究方法.....	16
3.1 研究問題描述.....	16
3.2 染色體的設計與解讀.....	17
3.2.1 染色體的設計.....	17
3.2.2 染色體的解讀.....	18
3.3 瀾集演算法求解方法.....	19
第四章 實驗情境與結果.....	26
4.1 實驗情境.....	26
4.2 終止條件與初始解的設定.....	27
4.2.1 終止條件的設定.....	27
4.2.2 初始解的產生方式.....	29
4.3 實驗結果與分析.....	30
4.4 統計檢定.....	33

第五章 結論與未來研究方向.....	35
5.1 研究結論.....	35
5.2 未來研究方向.....	36
參考文獻.....	37



表目錄

表 2.1 派工法則(資料來源：賴士葆(2004))	6
表 2.2 排程績效衡量指標.....	8
表 2.3 流線型生產製造單元排程問題的相關研究.....	9
表 2.4 固定序列與非固定序列的相關研究.....	10
表 2.5 瀾集演算法的相關文獻.....	12
表 4.1 結束條件兩千代與五千代間的統計檢定.....	29
表 4.2 結束條件兩千代與五千代間的計算時間.....	29
表 4.3 同 Seed 與同 Initial Solution 的統計檢定.....	30
表 4.4 本研究之實驗結果.....	31
表 4.5 本研究之統計檢定結果.....	34



圖目錄

圖 1.1 求解問題的精進方式.....	2
圖 1.2 流線型生產的製造單元加工方式.....	2
圖 1.3 順序相依的整備時間.....	3
圖 1.4 研究流程與架構.....	4
圖 2.1 固定序列與非固定序列.....	10
圖 2.2 舊染色體表達法.....	11
圖 2.3 新染色體表達法.....	11
圖 2.4 二元編碼方式.....	12
圖 2.5 實數編碼方式.....	13
圖 2.6 區域搜尋找解方式.....	13
圖 3.1 本研究問題範例.....	16
圖 3.2 舊染色體表達法之設計.....	18
圖 3.3 新染色體表達法之設計.....	18
圖 3.4 舊染色體表達法之解讀.....	19
圖 3.5 新染色體表達法之解讀.....	19
圖 3.6 瀾集演算法之流程架構.....	20
圖 3.7 模擬退火法之流程架構.....	22
圖 3.8 舊與新染色體表達法之 PBX 多點交配方式.....	23
圖 3.9 菁英政策.....	24
圖 3.10 舊與新染色體表達法之 Swap 突變方式.....	25
圖 4.1 本研究之實驗情境.....	27
圖 4.2 同初始解的產生方式.....	30
圖 4.3 本研究之績效改善率.....	32
圖 4.4 本研究之運算時間差距.....	32

第一章 緒論

1.1 研究背景

隨著科技技術不斷的演進，許多產業已經慢慢面臨到成本低、品質好、交期短的競爭環境下，而排程的意義就是在於將設備、人力或空間等可用資源做適當的分配，並透過作業效率、存貨狀況以及服務水準的調整，來完成某些特定作業的決策過程。因此所謂排程當然也包括決定每個工件的加工順序。

所謂的單元製造系統(Cellular Manufacturing System)即是各種工件(jobs)群組成不同的工件族(family)，將所需的機台組合在一起變成單元(cell)的形式，而這些不同的工件族在此單元(cell)上進行加工。單元製造系統對於許多產業而言，是一個相當實用的群組技術(Group Technology)，因此相關的製造單元排程問題(Manufacturing Cell Scheduling Problem; MCSP)變成近年來相當熱門的研究議題。此類排程問題需要考慮兩個部分，一個為決定工件族之間(Among Family)的加工順序，一個為決定工件族內工件(Within Family)的加工順序。

求解排程問題的方法，一般來說可分為最佳解求法及近似解求法，但是對於製造單元排程問題這種具有 NP-hard 特性而言，要在合理的時間內求得最佳解是相當困難的，因此利用一些啟發式演算法來求得近似解，相對的省時且符合經濟價值，例如基因演算法(Genetic Algorithms; GA)、模擬退火法(Simulated Annealing; SA)、塔布搜尋法(Tabu Search; TS)。不管是針對單一目標或者是多目標的追求下，都希望達到最佳化的效果。而所謂排程的好壞，一般來說可從交期相關(Due-date-based)，如最大延遲時間(Maximum Tardiness)、總延遲時間(Total Tardiness)、總加權延遲時間(Total Weighted Tardiness)或是完工時間相關(Completion-time-based)，如最大完工時間(Makespan)、總完工時間(Total Completion Time)、總加權完工時間(Total Weighted Completion Time)來做績效的評量。

1.2 研究目的

對於各種啟發式演算法的精進，可分為進化方法部分和染色體表達方法部分，如圖 1.1 所示，而近年來對於各種演算法求解各類型問題大部分著重在演算法的進化方法部分，像是改善突變的機率、交配的方式、停止條件的設定來試圖改善績效，且都有不錯的效果。但本研究想藉由彌集演算法(Memetic Algorithm; MA)的求解方法搭配 Wu *et al.* (2011)染色體表達法的改善概念，期望延續 Wu *et al.* (2011)的研究成果對於製造單元排程問題也能夠提升其求解品質。

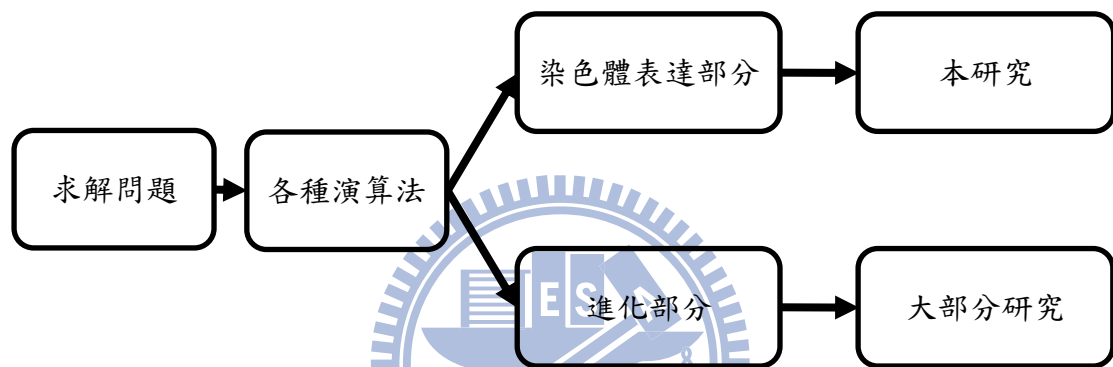


圖 1.1 求解問題的精進方式

1.3 研究範疇

本研究是屬於流線型生產的製造單元(Flowshop MC)，加工方式如圖 1.2 所示。就是在單元製造中，每個工件的加工順序皆是相同的，也就表示所需使用的機台順序也是相同的。

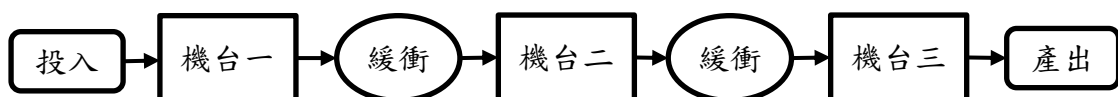


圖 1.2 流線型生產的製造單元加工方式

各個工件依造屬性紛紛被歸類至某個工件族中，同個工件族內的工件在同一部機台進行加工時的整備時間(Setup Time)相當的小，所以忽略不計，這也就是

為什麼要把工件群組成工件族的原因。但是當同一部機台進行下一個工件族的加工則一樣必須要有整備時間，而本研究所使用的整備時間是屬於順序相依的整備時間(Sequence Dependent Family Setup Time)，工件族在各機台進行加工前所需要的整備時間皆會因為前一個加工的工件族不同而有所不同，如圖 1.3 所示，當要進行工件族三(F_3)的加工時會因為前一個工件族的不同產生兩個不同的整備時間 S_{13} 、 S_{23} 。此外本研究具有固定序列(Permutation)的特性，當我們在一開始時決定了工件族間的加工順序後，工件族間在各個機台的加工順序皆相同。



圖 1.3 順序相依的整備時間

因此本研究即是探討利用濶集演算法搭配 Wu *et al.* (2011) 染色體表達法的概念來求解具有固定序列、順序相依整備時間的流線型生產製造單元排程問題，以最大完工時間當作績效衡量的指標，期望能夠延續 Wu *et al.* (2011) 的研究成果，在使用濶集演算法搭配 Wu *et al.* (2011) 染色體表達法上的績效能夠優於傳統的舊染色體表達法。

1.4 研究流程與架構

本論文內容共含五章。第二章開始對於本研究的問題及使用的方法做一些相關文獻的探討。第三章先對於問題進行描述，接著介紹染色體的相關設計與解讀，最後再對於使用的方法以及參數的設計做說明。第四章先說明實驗的情境，接著探討終止條件以及初始解的設定，最後對於結果進行分析。第五章則描述結論與未來研究方向。本研究流程如圖 1.4 所示。

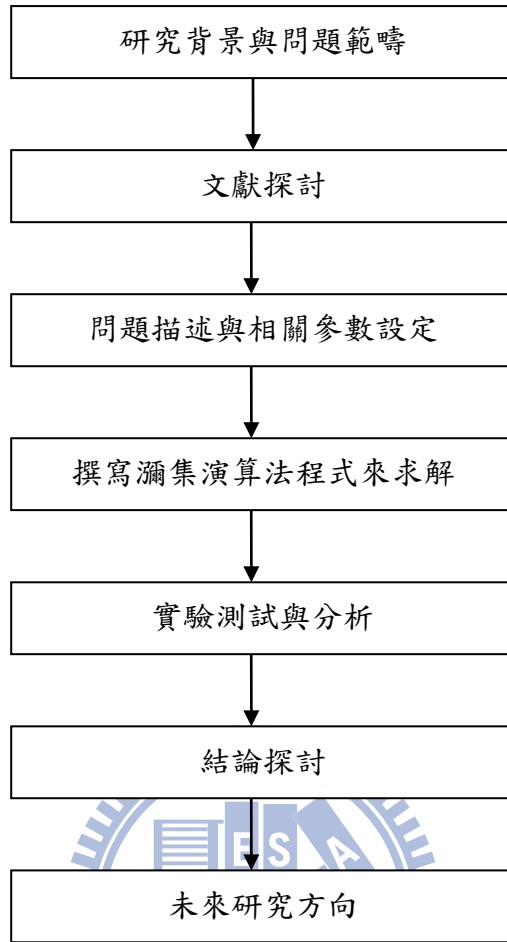


圖 1.4 研究流程與架構

第二章 文獻探討

本章文獻探討共分五個小節；第一節針對排程問題之概論進行探討；第二節探討流線型生產製造單元排程問題；第三節對於固定式與非固定序列(non-permutation)進行分類；第四節探討染色體表達法的差異；第五節討論濶集演算法。

2.1 排程問題之概論

如今市場競爭激烈，準時交貨以及滿足顧客需求並且提高服務水準變的相當重要，因此排程問題變成現在相當熱門的議題。所謂排程就是在探討如何在有限的資源下進行分配，決定各機台所有工作的加工開始時間和先後順序，以使得整體能夠達到最大的利益。

2.1.1 排程問題之分類

排程問題分類可以分成三類如下（資料來源：黃維曄(1994)、周書賢(2001)）：

(1) 依工件到達系統的時間點分類：

- (a) 靜態(Static)：假設訂單數目為已知且固定，所有工件同時到達生產系統中，即不考慮到達時間，在那段時間裡所有工件都是可以進行加工處理。
- (b) 動態(Dynamic)：假設訂單數目會隨著時間而有所不同，不斷會有新的工件加入，而且考慮到達時間，即所謂各工件到達生產系統時間不同。

(2) 依工件的作業特性來分類：

- (a) 確定性(Deterministic)：工件在各機台上的加工時間為已知且固定。
- (b) 隨機性(Stochastic)：工件在各機台上的加工時間不固定，且可能服從某項機率分配。

(3) 依照機台數目分類：

- (a) 單一機台(Single Machine)：只有一部機器，每一工件都由此機器加工。

(b) 多機台(Multiple Machine)：有二部或二部以上的機器去處理每一工件。

- 平行機(Parallel Machine)：同一個加工站，以多部相同加工性質的機器進行加工，而所有工件可以這些機器的任一機器上加工。
- 流線型工廠(FlowShop)：每一個工件均必須以相同的作業順序經過 M 部機器加工。
- 零工型工廠(JobShop)：每個工件可以有不同的生產途徑。

2.1.2 排程問題之求解方法

Jain and Meeran (1999) 將排程問題的求解大致分為五種模式：

(1) 派工法則(Dispatching Rules)：

此為最常使用的方法，用來安排待處理的工件來決定生產順序，如下所示：

表 2.1 派工法則(資料來源：賴士葆(2004))

派工法則	簡稱	意義	形式
1 先到先服務	FCFS	最早到的工作最先處理	靜態
2 最短處理時間	SPT	處理時間最短的先處理	靜態
3 最常處理時間	LPT	處理時間最長的先處理	靜態
4 最早到期日	EDD	工作到期日最早的先處理	靜態
5 最小剩餘工作時間	LWR	每一個工作剩下之未完成工作時間總和最小的優先處理	靜態
6 最小作業數	FOR	所需處理作業項目的總和最小的先做	靜態
7 浮時(寬放時間)	ST	距到期日所剩時間減所需處理時間最小的工作先處理	動態
8 浮時/所剩未完成作業數目	S/RW	浮時除以所剩之作業數目的比率最小的先處理	動態
9 關鍵比率(critical ratio)	CR	距到期日所剩時間除以完成工作所需時間，最小的優先處理	動態
10 客戶關係	PCO	關係較佳的客戶先處理	靜態
11 隨機選擇	RS	隨機選擇，選中的優先處理	動態

這類的方法是用於簡單的生產排程，如單機或者單階多機的問題。

(2) 最佳化方法(Optimization Method)：

早期對於排程問題大部分都偏重於最佳化方法，如線性規劃法(Linear Programming)、分枝界限法(Branch and Bound)、整數規劃法(Integer Programming)、動態規劃法(Dynamic Programming)。

這類方法僅能解決簡單的排程問題，原因是模式建立不易，且解題過程複雜，計算困難，所以會使整個模式失去真實性，與實際狀況無法吻合，故使用上較受限制。

(3) 啟發式演算法(Heuristic Algorithm)：

啟發式演算法是透過反覆進行的演算過程，近幾年已被廣泛的使用在排程問題，常見的有模擬退火法(Simulation Annealing)、蟻群演算法(Ant Colony Optimization)與基因演算法(Genetic Algorithm)等。

(4) 模擬法(Simulation)：

模擬法是以電腦系統中因果對應行動與事件之間的關係，用於評估系統功能及結果的工具。模擬法雖然能建構接近真實的系統，但卻不易直接藉由模擬本身尋找決策參數的最佳值。

(5) 人工智慧(Artificial intelligence)：

近十年來，人工智慧領域的發展可說是科技發展史上的重大突破。人工智慧包含模糊邏輯、分散式人工智慧、模糊專家系統及類神經網路等。人工智慧應用的範圍很廣，從模擬控制、圖樣辨識乃至最佳化及預測等。

2.1.3 排程績效衡量指標

表 2.2 排程績效衡量指標

類別	績效評估指標
時間相關 (Time-based Measures)	最大完工時間 (Makespan)
	總加權完工時間 (Total weighted completion time)
	平均流程時間 (Mean flow time)
	平均等待時間 (Mean waiting time)
	流程時間的變異 (Variance of Flow time)
	單位時間產量 (Throughput)
	生產週期 (Cycle Time)
交期相關 (Due-date related Measures)	最大延遲時間 (Maximum tardiness)
	總延遲時間 (Total tardiness)
	加權延遲時間 (Weighted tardiness time)
	延遲工件數 (Number of tardy jobs)
	總早交時間 (Total earliness)
	符合交期百分比 (%)
成本相關 (Cost-based Measures)	設置成本 (Setup cost)
	在製品存貨成本 (WIP inventory cost)
	延遲完工成本 (Tardiness cost)
	提早完工成本 (Earliness cost)
在製品及機器相關 (WIP and Machine Measures)	平均在製品數量 (Average WIP)
	平均機台使用率 (Average machine utilization)
	平均等待工件數 (Average number in queue)
	平均機台閒置時間 (Average idle time)

2.2 流線型生產製造單元排程問題

隨著經濟的快速成長，所謂的流行快速形成且轉變，使得許多企業必須要不斷的推陳出新，造就了許多中、小批量的生產模式，並且隨著客製化的生產型態日漸茁壯，以至於產品多樣少量化的形成，故在工廠的生產加工上常常會使用所謂的流線型生產製造單元，因為它可以減少整備時間、減少在製品數量、降低物料的搬運成本、提高機器的使用率並且簡化生產計劃和管控。

流線型生產製造單元的概念是根據其製造上的屬性，將所有工件分類成若干個不同的工件族並將機台分為若干個機器群以形成單元的模式進行群組技術的應用，而這些工件族皆以相同的加工步驟經過這些機器群，所謂製造上的屬性能夠導致工件族內的工件與工件之間是不需要整備時間。此類的問題使用了許多不同的方法來進行研究，舉例如表 2.3 所示。

表 2.3 流線型生產製造單元排程問題的相關研究

作者(年分)	研究方法	研究內容
Deutsch <i>et al.</i> (1998)	數學規劃模式	利用改良之 P-中位數模式(p-median model)來增加工件的相似性，並改善了單元形成問題中非線性目標函數的缺點。
Spiliopoulos and Sofianopoulou (1998)	啟發式演算法	使用樹狀搜尋啟發式演算法來處理單元製造系統問題，利用盡量縮小樹狀大小的步驟，此方法有效地降低運算成本並得到不錯的解。
Abdelmola <i>et al.</i> (1998)	數學規劃模式	使用一個雙階模式之二元整數規劃法來處理單元製造生產率問題。
Lin <i>et al.</i> (2009)	啟發式演算法	利用基因、塔布、模擬退火等三個演算法來比較固定序列與非固定序列的流線型單元製造排程問題。

2.3 固定序列(Permutation)與非固定序列(Non-Permutation)

所謂固定序列與非固定序列間的不同之處在於不同機台時，工件族間的加工順序是否可以改變，如圖 2.1 所示。

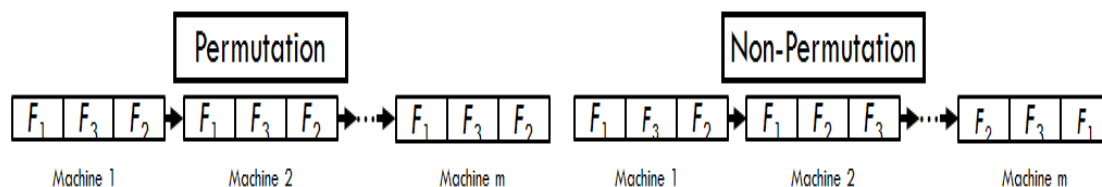


圖 2.1 固定序列與非固定序列

若是具有固定序列的特性即表示在一開始時就決定各工件族間的加工順序，此後就以這樣的順序在各機台上進行加工；相反地，若是具有非固定序列的特性，則在每個加工機台皆各自決定工件族間的加工順序。但無論如何工件族內的工件順序在開始時就決定了且皆不改變。本研究採用的是具有固定序列的特性。固定序列與非固定序列的相關研究如表 2.4 所示。

表 2.4 固定序列與非固定序列的相關研究

特性	使用方法	評估指標	作者(年分)	產品特性
Non-permutation	mixed-integer linear programming; Tabu search	total tardiness	Liao & Huang (2010)	Non family-based
	Simulated annealing; Genetic algorithm; Tabu search	Makespan Total completion time Total weighted completion time Maximum tardiness Total tardiness Total weighted tardiness	Lin et al. (2009)	family-based
permutation	genetic algorithm; variable neighborhood search	makespan	Zobolas et al. (2009)	Non family-based
	hybrid tabu search	makespan	Chen et al. (2008)	Non family-based
	hybrid genetic algorithm	makespan	Zhang et al. (2009)	Non family-based
	Branch and bound; Genetic algorithm	makespan	Bouabda et al. (2011)	family-based
	similar particle swarm optimization algorithm	makespan	Lian et al. (2006)	Non family-based

2.4 探討染色體表達法的差異

此類問題的傳統染色體表達通常像是 Lin *et al.* (2009)和 Bouabda *et al.* (2011)的文章中所呈現的，把染色體分成數段，第一段是表示各機台的工件族間加工先後順序，接下來的每一段分別代表各工件族內的工件加工順序，此處我們將其定義為舊染色體表達法(S_{old})，如圖 2.2 所示。

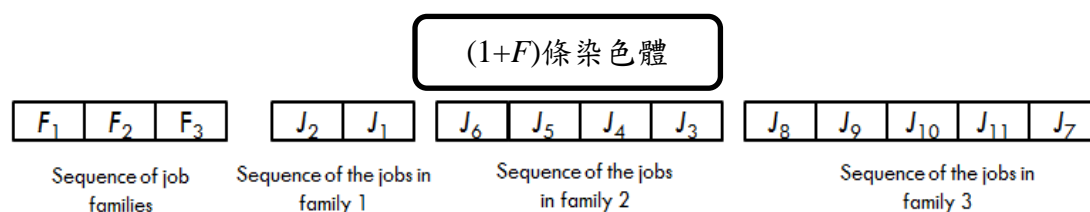


圖 2.2 舊染色體表達法

而本研究是沿用 Wu *et al.* (2011)的染色體表達法概念，認為染色體由一段來呈現即可，此處我們將其定義為新染色體表達法(S_{new})，如圖 2.3 所示。



圖 2.3 新染色體表達法

2.5 瀾集演算法

藉由 Dawkins 在 *The Selfish Gene* 這本書中對於瀾(Meme)的解釋，認為瀾是人與人交換想法的文化傳遞單元，瀾的概念是無所不在的，而每個人在進行傳遞時多少都會將資訊產生變化，但會留下所謂的精隨。

瀾集演算法(Memetic Algorithm; MA)是 Moscato (1989)根據文化的演化概念所發展出多點搜尋的方式，屬於一種整合式的演算法，利用全域搜尋(例如：GA)的方式來結合區域搜尋(例如：SA、Tabu、PSO)。本研究參考 Moscato (1989)所提出此演算法的組合概念利用 GA(全域)+SA(區域)來求解問題，即在基因演算法中的每一代皆使用 SA 來導入局部最佳化的概念。一些學者使用瀾集演算法的相

關文獻如表 2.5。

表 2.5 瀾集演算法的相關文獻

作者(年分)	評估指標	問題描述
França <i>et al.</i> (2001)	Total tardiness	利用瀾集演算法來求解單機台的排程問題
Chiang <i>et al.</i> (2010)	Total weighted tardiness	探討平行批次機台的排程問題，具有不相容(incompatible)的特性且工件到達屬於動態的
Neammanee & Reodecha (2009)	Total weighted tardiness	求解印刷電路板之單機台排程問題
Chiang <i>et al.</i> (2011)	makespan	求解多目標的固定序列流線型生產的排程問題

以下是一些瀾集演算法的重要名詞介紹：

基因編碼

編碼是在對於求解問題時，最先需要做的事情。針對問題的不同，將所有的問題變數轉換成適當的編碼使得問題變數以染色體的方式來呈現，常見的編碼方式有二元編碼(Binary Coding)和實數編碼(Real Coding)兩種。

二元編碼就是將染色體的各點用 0、1 表示，例如 0 表示此時間點作業暫停，1 表示此時間點進行作業，如圖 2.4 即表示為時間點 1 和時間點 3 有在進行作業，其他時間點作業暫停。

1	0	1	0	0	0
---	---	---	---	---	---

圖 2.4 二元編碼方式

實數編碼則是直接用數字來代表，例如將各個數字代表各個工件，利用擺放的先後位置來當作加工作業的先後順序，如圖 2.5 即表示工件加工順序依序為工件

1、工件 3、工件 5、工件 2、工件 6、工件 4。

1	3	5	2	6	4
---	---	---	---	---	---

圖 2.5 實數編碼方式

初始母代群體

由於瀾集演算法是屬於多點搜尋的方法，所以在一開始就必須產生多組的初始母代群體，產生初始母代群體的方法像是隨機產生、利用派工法則或是要與某篇論文比較而使用與某篇論文相同的初始母代群體。

初始母代群體的數量與求解效能相當有關，若數量太大雖然能夠得到較佳的解但是所需計算的時間較長，若數量太小會造成提早收斂且達不到預期的目標，此外若問題有訂定運算次數的話，所訂定的運算次數很大則可以搭配較大的初始母代群體，若所訂定的運算次數有限則初始母代群體就不能太大，因此在運算次數、初始母代群體的數量和求解效能上面的考量取捨是相當重要的。

區域搜尋

區域搜尋的意義就是在找尋目前解中的局部最佳解，如圖 2.6 所示將目前解移到最接近的一個局部最佳解位置的過程。不管在剛開始產生初始母代群體或是經過交配、突變所產生出的新解，都需要經過區域搜尋這個步驟，此步驟可以使得在整個瀾集演算法的過程中加速到達最佳解的位置。常見的區域搜尋方法像是交換(Swap)、插入(Insert)、甚至使用模擬退火法等等。

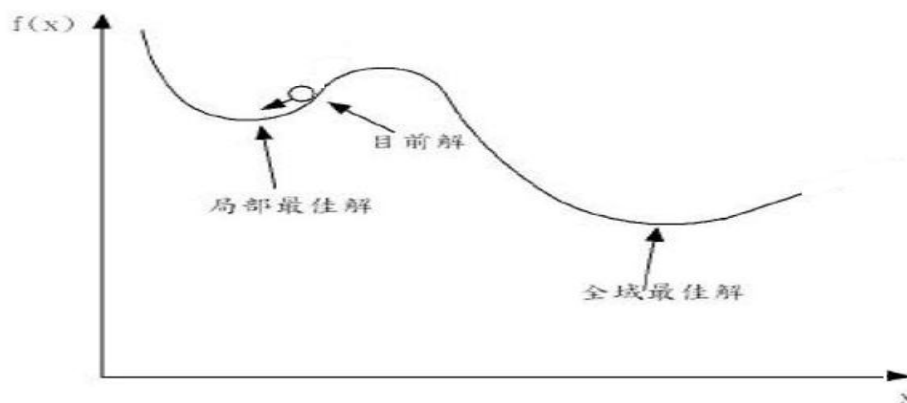


圖 2.6 區域搜尋找解方式

交配

從母代群體中隨機挑選出兩條染色體，將這兩對染色體進行配對交換基因的動作，產生出一個子代，此為模擬自然界生物交配的現象，希望透過這種方式所產生出的子代能夠保有兩個母代較好的基因片段。一般常見的交配方式有單點交配(One-point Crossover)、雙點交配(Two-point Crossover)、線性順序交配(Linear Order Crossover Operator)、部分相應交配(Partially Matched Crossover)。

突變

突變的意義是為了避免在求解一段時間後，造成收斂的現象而導致遺漏掉一些可能的解空間而落入局部最佳解。因此從母代群體中挑選出一條染色體來進行突變的動作，希望能夠跳脫出來局部最佳解繼續尋優。彌集演算法的突變與基因演算法最大的不同在於發生的時間點。彌集演算法的突變是當演化了一定的世代後造成收斂的現象才需要將最好的染色體除外的所有母代群體進行突變，而所謂的收斂在後面會進行介紹。常見的突變方法有交換(Swap)、插入(Insert)和反轉(Inverse)。

適應度函數值

適應度函數值即表示此染色體適合存在的程度。每個染色體根據研究的目標後利用目標值函數算出目標值，再利用適合度函數轉換成適合度函數值，以便於在決定進行染色體淘汰篩選時的重要指標。

菁英政策

菁英政策顧名思義就是留下好的淘汰不好的，只要每次產生出新的染色體時，為了要保留一定數量大小的母代群體，所以需要進行菁英化，經過不斷的世代演化和不斷的菁英政策才能夠找到最好的解。常見的菁英政策有輪盤法(Roulette Wheel Selection)和競賽法(Tournament Selection)。

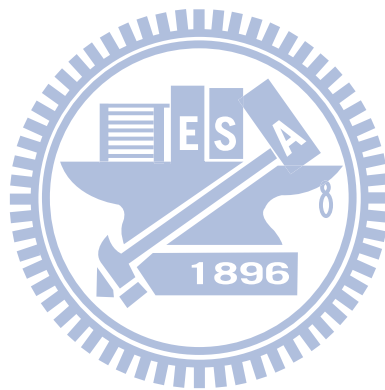
收斂

收斂的定義有很多種像是連續幾次運算皆未改變最佳解或是當運算次數到達一定的次數就直接認定或是母代群體中有多少對染色體皆相同時。

終止條件

即所謂結束尋優的條件。一般常見的像是運算次數到達設定條件、經過多少的運算時間、設定連續幾代未改變最佳解。

至於瀾集演算法的演算步驟流程以相關參數設定會在後面的章節進行介紹。



第三章 研究方法

本章研究方法共分三個小節；第一節對於研究的問題進行描述；第二節探討本研究染色體的設計與解讀；第三節對於瀾集演算法是如何求解做說明。

3.1 研究問題描述

本研究是利用瀾集演算法搭配新染色體表達法來針對具有固定序列、順序相依整備時間特性的流線型生產製造單元排程問題，目的在於使得我們的績效衡量指標能夠越小越好。本研究主要考慮的有兩個部分，一個為決定工件族之間的加工順序，一個為決定工件族內工件的加工順序，如圖 3.1 所示，假設一個問題具有九個工件，而這九個工件是屬於三個不同的工件族且分別皆需要三部機台進行加工，因此求解此類問題就要先經由步驟一將工件以工件族為類別進行分群，接下來像步驟一和步驟二利用瀾集演算法決定各機台工件族間加工順序的排序以及各工件族內工件加工順序的排序，最後經由步驟三得到實際工件在各機台的加工順序，如此便完成整個問題的求解。而問題的複雜度會隨著工件數和工件族數的增加而增加。

□ 例子說明：

- 3部機台
- 9個工件
- 3個family

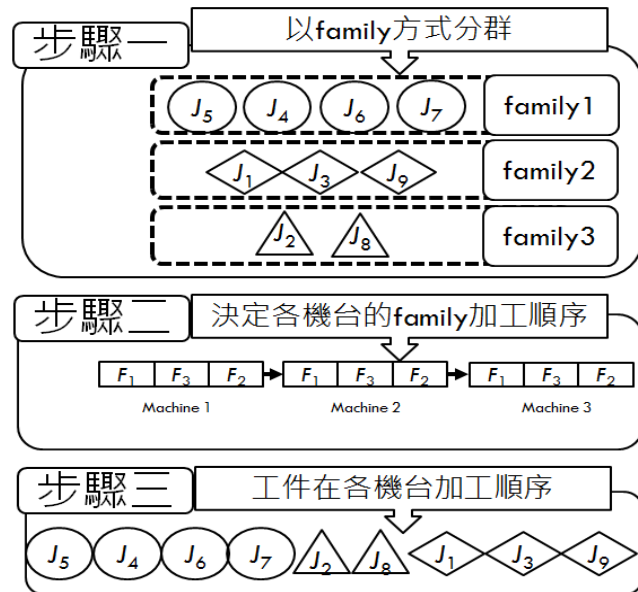
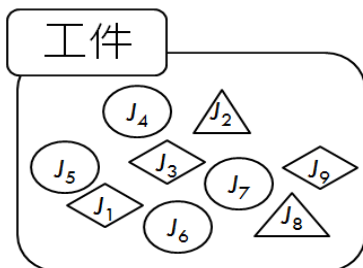


圖 3.1 本研究問題範例

本研究屬於靜態的排程問題有以下假設條件：

- (1) 各加工時間、工件族數、工件族中的工件數，各整備時間，皆為已知的非負整數
- (2) 開始時間設為 0
- (3) 同一個工件族的所有工件要做完才會換另一個工件族做
- (4) 同一個工件族內各工件的整備時間不計
- (5) 每部機台第一個加工的工件族不需要整備時間
- (6) 機台一次只加工一個工件，且不可以先做下一部機台
- (7) 沒有當機問題
- (8) 機台與機台間的運輸時間為 0
- (9) 機台前緩衝區的在製品數量無上限
- (10) 加工完之工件即可直接進入下一部機台加工或等候，不需要等到整個工件族完成加工



3.2 染色體的設計與解讀

此節主要介紹求解此類問題時，本研究的染色體是如何設計以及面對一條染色體如何能夠解讀成實際進行加工時可以使用的情形。

3.2.1 染色體的設計

因為本研究的問題相較一般排程問題需要額外考慮到工件族的部分，所以在染色體的表達上不能在光光只是表達所謂的工件加工順序，需要的是能夠表達工件族間在機台的加工順序和工件族內各工件的加工順序等兩個部分。

在傳統求解此類時，都使用像是圖 3.2 的染色體表達法，也就是將染色體分成數段，第一段先表示出工件族間在各機台的加工順序，接下來幾段才是表現各工件族內工件的加工順序，所以舊染色體表達法的長度即為〔工件族數+1〕個段

落，也就是會有〔機台數+工件數〕個基因所組成。

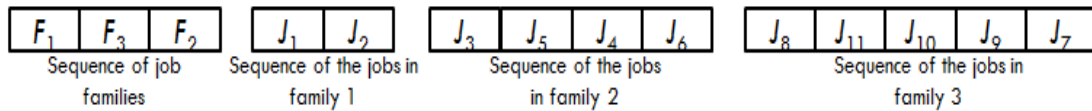


圖 3.2 舊染色體表達法之設計

而本研究求解此類問題所使用的染色體是使用新染色體表達法，把染色體的表達用一整條的表示方式即可，如圖 3.3 所示，所以本研究所使用的染色體的長度只有〔1〕個段落，而由〔工件數〕個基因所組成。

了解舊染色體表達法和新染色體表達方式後，重要的是面臨這樣一條染色體該如何解讀成實際能用的加工形式，此部分會在下一小節進行介紹。

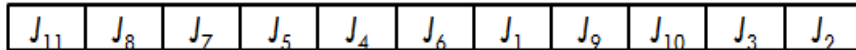


圖 3.3 新染色體表達法之設計

3.2.2 染色體的解讀

在染色體的解讀上可分為兩個部分，首先是解讀工件族間在機台的加工順序，接著再根據工件族間在機台的加工順序來解讀出各工件的加工順序。此處會介紹舊染色體表達法和新染色體表達法兩種不同的解讀方式。

假設欲求解的問題具有十一個工件，其分別屬於三個不同的工件族，工件一和工件二屬於工件族一，工件三到工件六屬於工件族二，工件七到工件十一屬於工件族三，這些工件需要經過三部機台進行加工，首先先介紹舊染色體表達法的所有解讀過程如圖 3.4 所示，經過瀾集演算法的不斷求解下所得到的舊染色體表達法先要進行工件族間在機台的加工順序解讀，根據染色體的第一段解讀出工件族間在機台的加工順序為工件族一、工件族三、工件族二，再根據此工件族間在機台的加工順序來排出各工件實際的加工順序為工件一、工件二、工件八、工件十一、工件十、工件九、工件七、工件三、工件五、工件四、工件六，如此便完成舊染色體表達法的解讀工作。

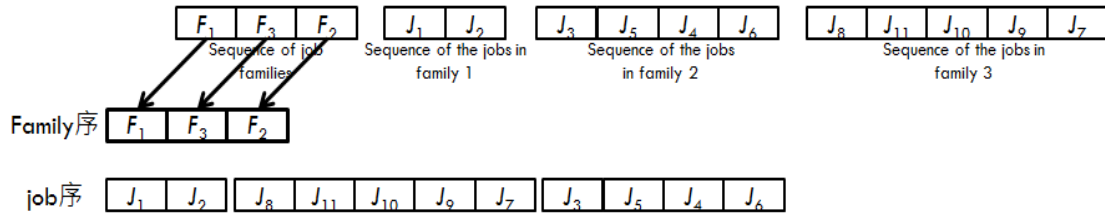


圖 3.4 舊染色體表達法之解讀

至於本研究所使用的新染色體表達法解讀方式如圖 3.5 所示，經過瀾集演算法的不斷求解下所得到的染色體先要進行工件族間在機台的加工順序解讀，根據此染色體所排的第一個工件為工件十一而得知第一個加工的工件族為工件族三，接著依序往後找出第一個非工件族三的工件為工件五，所以得知第二個進行加工的工件族為工件族二，以此方法繼續下去直到所有工件族的排序皆完成，其順序為工件族三、工件族二、工件族一。然後再根據此工件族間在機台的加工順序來排出各工件實際的加工順序為工件十一、工件八、工件七、工件九、工件十、工件五、工件四、工件六、工件三、工件一、工件二，如此便完成新染色體表達法的解讀工作。

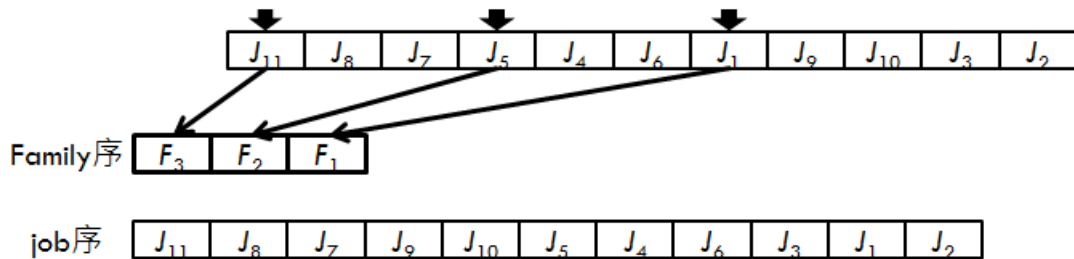


圖 3.5 新染色體表達法之解讀

3.3 瀾集演算法求解方法

本研究所使用求解此排程問題的方法為瀾集演算法，圖 3.6 為瀾集演算法的詳細求解流程，本研究對於瀾集演算法的使用方式是參考 Moscato (1989)所提出對於此演算法的發明想法與實際應用方式，利用 GA(全域)+SA(區域)去求解問題，接下來將會對於每個步驟進行說明。

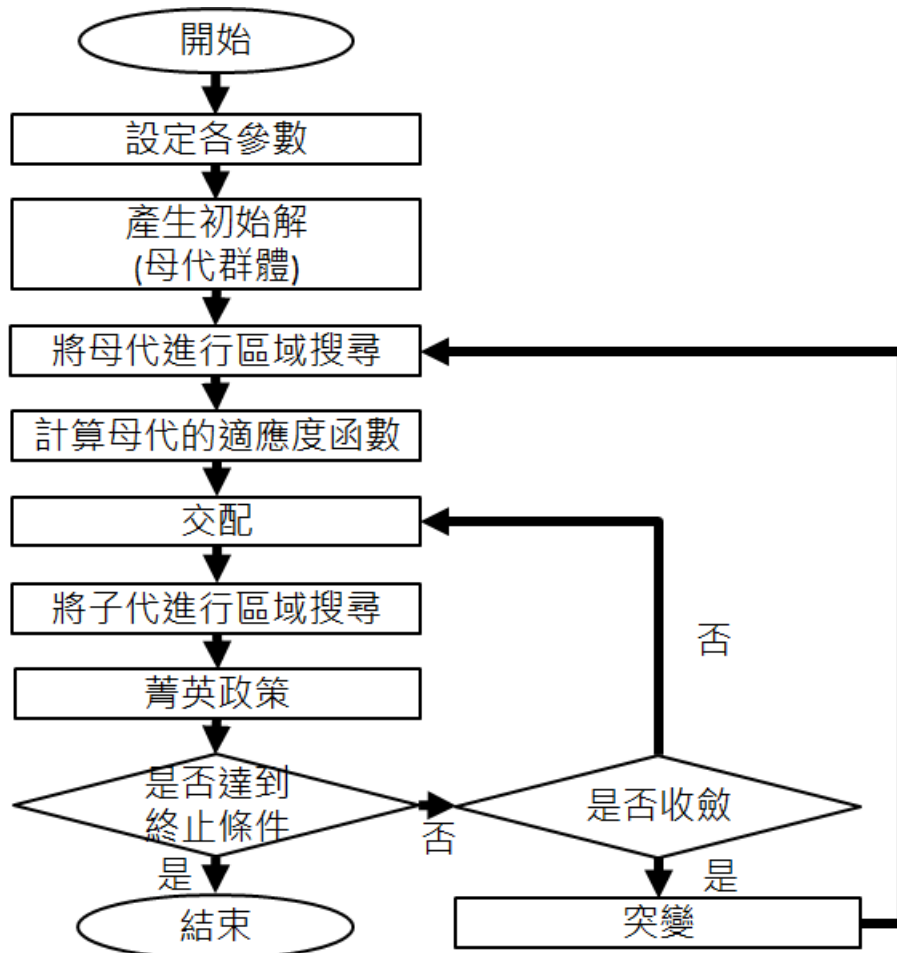


圖 3.6 瀾集演算法之流程架構

Step1. 設定各參數

利用瀾集演算法求解本研究的問題時，對於瀾集演算法的使用必須在一開始時將一些基本的參數進行設定，需要設定的參數如下：

- 初始解產生方式
- 初始母代群體之數目
- 決定區域搜尋方式
- 適應度函數值的計算方式
- 交配方式
- 菁英政策方式
- 突變方式

- 收斂條件
- 終止條件

Step2. 產生初始解

產生初始母代群體的方式是隨機產生的，但因為新舊染色體表達法不同，所以在此環節上會有問題產生，實際初始解的解決之道和產生方式會在 4.2.2 節做詳細說明。而母代群體的大小為一千條，這樣的母代群體大小配合本研究的龐大運算次數並不會造成問題。

Step3. 將母代進行區域搜尋

對於新產生的母代為了能夠加速整體演算法的精進，所以需要進行區域搜尋的動作，本研究所使用的區域搜尋方法是模擬退火法，相關流程如圖 3.7 所示(參考來源 Metropolis (1953))，而相關的參數設定如下：

- 起始溫度：1000 度
- 降溫係數：0.6
- 柯西函式： $\text{溫度}^2 / (\text{溫度}^2 + \text{目標值差距}^2)$
- 終止條件：溫度降到 10 度以下
- 初始解：需要進行區域搜尋的染色體
- 鄰近解：利用交換(Swap)得到

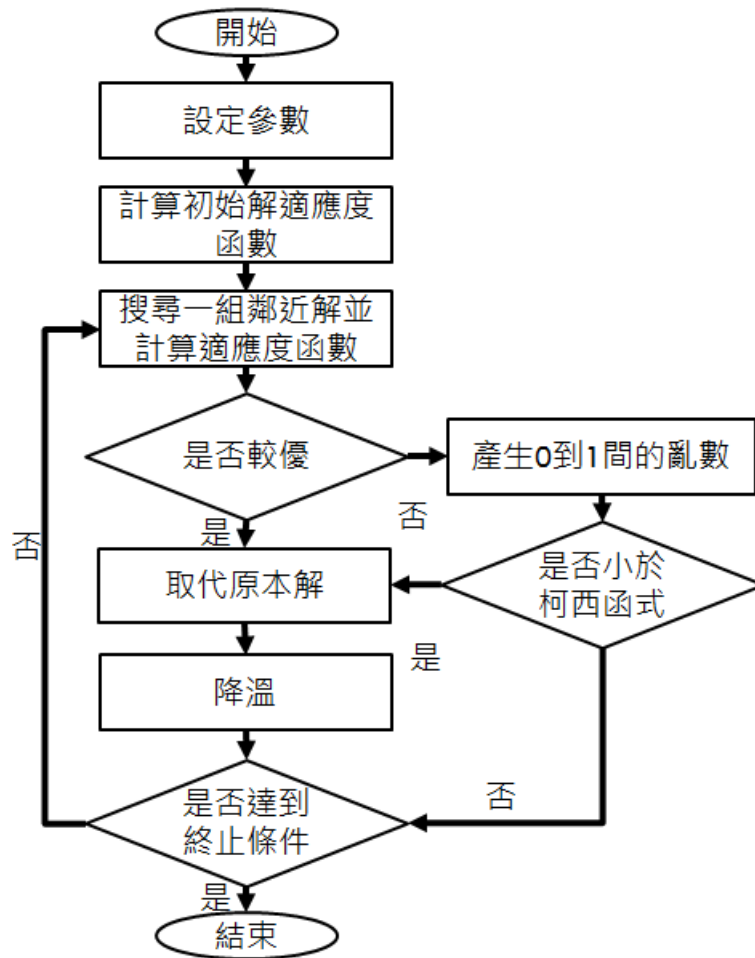


圖 3.7 模擬退火法之流程架構

Step4. 計算母代的適應度函數

將已經進行區域搜尋的母代分別計算其適應度函數值，以便於接下來進行各項工作時，判斷染色體的存在價值。本研究染色體的適應度函數值的計算方式是先將染色體的目標值計算出來，再經由適應度函數的轉換而得知，如下所示：

P ：表示初始族群染色體的數量

v_j ：表示第 j 個染色體的目標函數值 ($j \in P$)

f_j ：表示第 j 個染色體的適應函數值 ($j \in P$)

$$\text{適應度函數值： } f_j = \frac{1}{(1+v_j)}$$

本研究適應度函數值是越大越好。

Step5. 交配

做好 Step1 到 Step4 的工作後，算是完成演算法的準備工作，而此步驟即開始進行演化世代的動作，透過交配產生出新的子代，並透過不斷的世代演化來達成尋優的動作。本研究的交配方式是屬於 PBX 多點交配的方式如圖 3.8 所示，不管是舊染色體表達法或是新染色體表達法所使用的交配方式皆相同，先隨機從母代群體中挑選兩條染色體，再找出其中一條適應度函數值較好的當作母代一，再以同樣方式找出母代二來進行交配。利用一個 mask 將各點產生一個 0 或 1 的數字，當基因的位置是 1 時，則將母代一在此位置的染色體基因複製給子代，依序往後執行，等完畢後再將子代剩餘位置用母代二來補上。

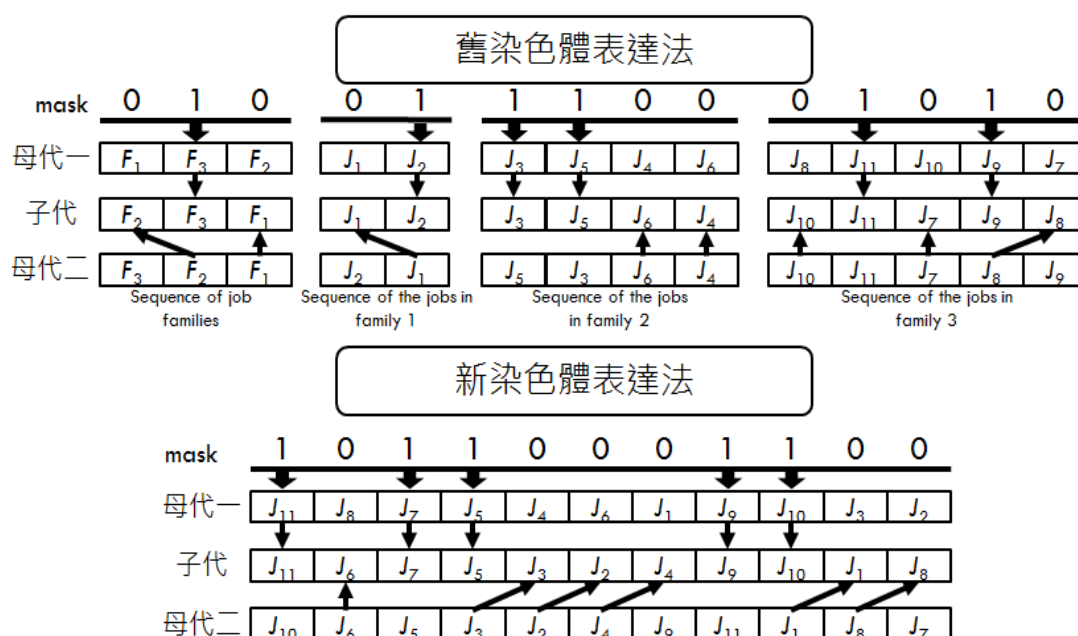


圖 3.8 舊與新染色體表達法之 PBX 多點交配方式

Step6. 將子代進行區域搜尋

將新產生的子代像 Step3 一樣的進行區域搜尋動作，因為瀾集演算法就是在每一代皆導入局部最佳化的概念，所以每個產生出的新解都要進行這樣的一個步驟。

Step7. 菁英政策

為了保持母代群體的大小，所以對於新產生出的子代要考慮它是否適合母代群體，因此本研究的菁英政策是從母代一和母代二以及子代中三個挑出兩個放回母代群體中，執行的方式如圖 3.9 所示，先分別算出此三條染色體的目標值，再經過轉換成適應度函數值，選擇較大的兩個進行菁英化保留下來，最後在把這兩個放回母代群體中，便完成了所謂的菁英政策。

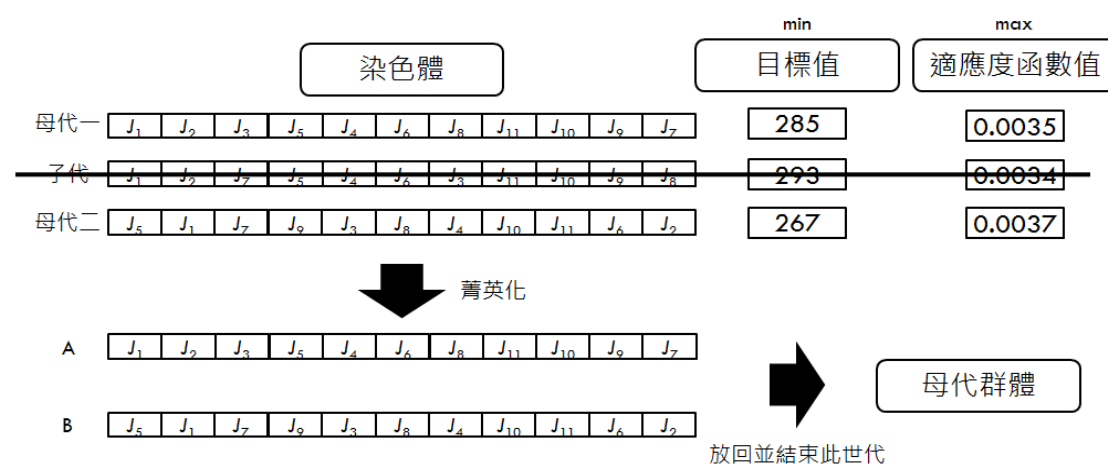


圖 3.9 菁英政策

Step8. 是否達到終止條件

到 Step7 時算是完成一個世代的演化，在進行下個世代前要先判定是否到達終止的條件，若已到達終止的條件則結束此尋優的過程；若還未到達終止的條件，則繼續進行 Step9。而本研究的終止條件會在 4.2.1 節做一個詳細的說明與設定。

Step9. 是否收斂

由 Step8 已經得知是否執行下一個世代的演化，但是在執行下一個世代前要先判定此母代群體是否已經造成收斂的現象，如此可避免此演算法的求解過程已經落入某個局部最佳解卻還一直往那個方向進行尋優的動作。本研究收斂的判定是當連續十個世代皆未改變最佳解。若判定為收斂，則進行 Step10；若還未收斂則跳回 Step5。

Step10. 突變

本研究所使用的突變時機與傳統的基因演算法不同，是當發現收斂時才需要使用，用以跳脫局部最佳解的框架。而本研究所使用的突變方式是將染色體進行連續十次的交換(Swap)動作，執行方式如圖 3.10 所示，先隨機從母代群體中挑選兩條染色體，再找出其中一條適應度函數值較好的當作母代，本研究所使用的新染色體表達法只需要以隨機找出兩點的方式來進行交換基因即完成一次的交換(Swap)動作，但是舊染色體表達法需要在每一段皆隨機找出兩點來進行交換基因才算完成一次的交換(Swap)動作，此為兩者不同之處，這也可能是造成本研究的運算時間較快的原因。做完此步驟則跳回 Step3。

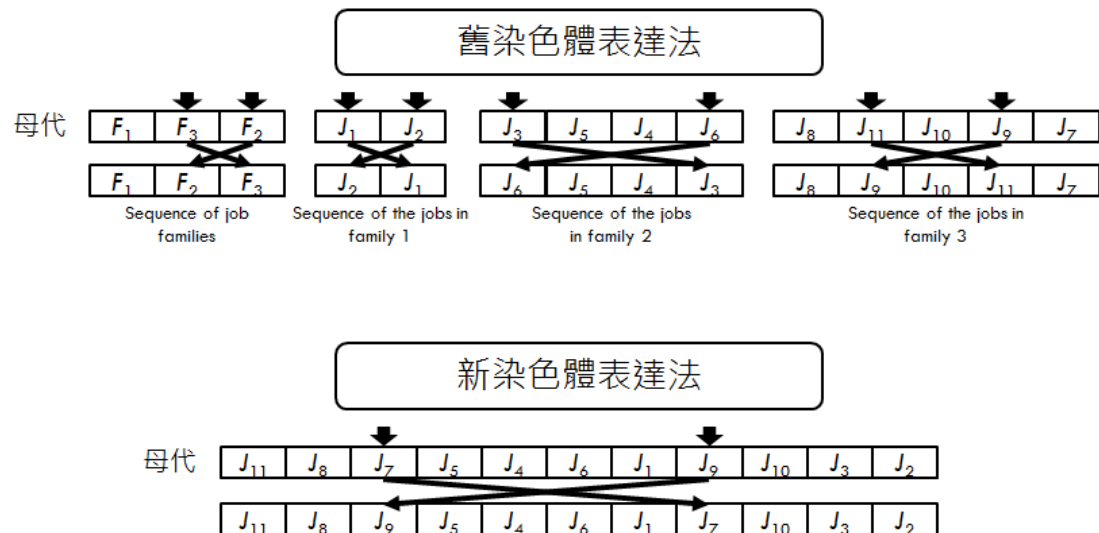


圖 3.10 舊與新染色體表達法之 Swap 突變方式

第四章 實驗情境與結果

為了要驗證本研究所提出的瀾集演算法搭配 Wu *et al.* (2011)染色體表達法對於具有固定序列、順序相依整備時間特性的流線型生產製造單元排程問題之效益，所以利用瀾集演算法搭配舊染色體表達法來當作比較對象，探討在績效指標最大完工時間與運算時間的差異，以對於本研究下最後的定論。本研究使用 Microsoft Visual Studio 2008 來進行程式的撰寫，跑實驗數據所使用的電腦配備如下所示：

- 作業系統(OS)：Windows 7 Enterprise 32-bits
- 中央處理器(CPU)：AMD Athlon(tm) II X4 640 Processor 3.00 GHZ
- 記憶體(Memory)：4.00 GB (2.75GB 可用)
- 系統類型：32 位元作業系統
- 使用語言：C++
- 編譯器(Compiler)：MSVC9.0



4.1 實驗情境

本研究為了要與舊染色體表達法進行比較，所以相關的研究參數皆依照 Schaller *et al.* (2000)所設計的去設定，相關參數設定如下所示：

- 使用三個不同大小的整備時間：
 - Small setups (SSU)：U[1,20]
 - Medium setups (MSU)：U[1,50]
 - Large setups (LSU)：U[1,100]
 - 各個工件族在各個機台的整備時間皆是不相同的
- 每種整備時間下皆有十個情境(工件族數，機台數)：(3, 3)，(3, 4)，(4, 4)，(5, 5)，(5, 6)，(6, 5)，(6, 6)，(8, 8)，(10, 8)，(10, 10)
- 工件族內的工件個數為均勻分配 U[2,10]

- 各個工件的加工時間為均勻分配 $U[1,10]$
- 每個整備時間的每個情境下分別放入 30 個問題(instance)去各跑十五筆數據 (seed)

因此本研究在這樣實驗情境下，總共會產生 $2*3*10*30*15=27000$ 筆數據如圖 4.1 所示。

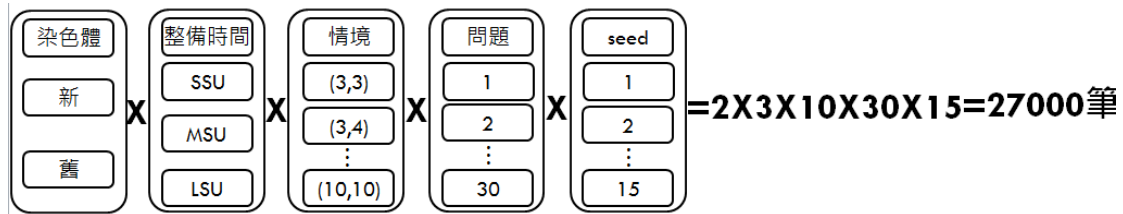


圖 4.1 本研究之實驗情境

4.2 終止條件與初始解的設定

此節主要介紹本研究在利用瀾集演算法求解此類問題時，對於終止條件的設定和初始解的產生方式上去做探討與決定。

4.2.1 終止條件的設定

本研究使用連續未改變最佳解多少代當作終止的條件，至於要多少代才算適合，本研究利用各種代數來做統計檢定，觀察兩兩代數下的〔新染色體表達法與新染色體表達法〕和〔舊染色體表達法與舊染色體表達法〕這兩種的目標值是否有顯著的差異。

本研究使用 Hendizadeh *et al.* (2008) 在研究類似問題時用 Paired t-test 來進行統計檢定，以確認本研究所設定的參數。以一個問題當單位，所以先將問題中的 15 個 seed 做平均，每個整備時間下的每個情境都挑選一個問題當作代表來進行檢定，相關符號設定如下：

- $A - S_{old,i}$, $i=LSU33, LSU34, \dots, SSU1010$

(較大的代數 A 下舊染色體表達法各種整備時間與情境下的各個問題目標值)

- $B - S_{old,i}, i=LSU33, LSU34, \dots, SSU1010$
(較小的代數 B 下舊染色體表達法各種整備時間與情境下的各個問題目標值)
- $A - S_{new,i}, i=LSU33, LSU34, \dots, SSU1010$
(較大的代數 A 下新染色體表達法各種整備時間與情境下的各個問題目標值)
- $B - S_{new,i}, i=LSU33, LSU34, \dots, SSU1010$
(較小的代數 B 下新染色體表達法各種整備時間與情境下的各個問題目標值)
- n : 資料數
- $d_i(\%)$: 第 i 個 **Scenario** 的改善率
- $\bar{d}(\%)$: 平均改善率
- S_d : 改善率標準差
- t_0 : t 值
- t : 檢定值



以舊染色體表達法與舊染色體表達法間的檢定為例的相關計算如下：

- $n=30$
- $d_i(\%) = \frac{(B - S_{old,i}) - (A - S_{old,i})}{(B - S_{old,i})} \times 100, i=LSU33, LSU34, \dots, LSU1010$
- $\bar{d}(\%) = \frac{\sum_{i=LSU33}^{LSU1010} d_i}{n}$
- $S_d = \left[\frac{\sum_{i=LSU33}^{LSU1010} (d_i - \bar{d})^2}{n-1} \right]^{\frac{1}{2}}$
- $t_0 = \frac{\bar{d}}{S_d / \sqrt{n}}$
- $t=1.96$ (查 t 表，信賴水準 95%，自由度= $n-1$)

當我們所計算出的 t_0 大於查表的 t 時，則表示結束代數大的所產生的目標值顯著贏結束代數小的，當我們所計算出的 t_0 沒有大於查表的 t 時，則表示兩種代數之間是沒有顯著差異的。表 4.1 為結束條件兩千代與五千代間的統計檢定實驗結果，有此可以看出五千代是顯著贏兩千代的，但考慮到平均改善率分別只有

0.11%和 0.05%，且由表 4.2 可以看出結束條件 5000 代所需計算的時間是 2000 代的兩倍多，所以本研究還是決定使用連續未改變最佳解兩千代當作結束的條件。

表 4.1 結束條件兩千代與五千代間的統計檢定

收斂代數 2000 代 VS 5000 代				
Scenario	平均改善率(%)	t_0	$t_{\alpha, n-1}$	統計結果
舊染色體表達法	0.1143	4.22	2.05	顯著贏
新染色體表達法	0.0564	3.50	2.05	顯著贏

表 4.2 結束條件兩千代與五千代間的計算時間

計算時間(sec)		
	2000 代	5000 代
舊染色體表達法	304.35	767.17
新染色體表達法	254.54	596.98

4.2.2 初始解的產生方式

本研究原先使用相同的 15 個隨機亂數起始點去產生新舊染色體表達法初始解，也就是所謂同 seed 的概念，但發現因為新舊染色體表達法的不同，導致上述方式所產生的新舊染色體表達法的初始解其實是不相同的。所以我們利用了如圖 4.2 的方式，先產生新染色體表達法的初始解，再經由轉換，產生出舊染色體表達法的初始解，以達成初始解的相同，如此便達到所謂同 initial solution 的概念。

本研究也利用同 seed 與同 initial solution 去做與 4.2.1 相同的統計檢定，比較結果如表 4.3 所示。由此可以看出在求解此類問題時，初始解使用同 seed 或同 initial solution 皆是可以的。但因為初始解的不同，對於我們在最後下結論要證明我們所使用的瀾集演算法搭配新染色體表達法贏舊染色體表達法是不合理的，因為求解問題的本身不是在同一個起跑點上的。所以本研究還是選擇同 initial solution 去做。

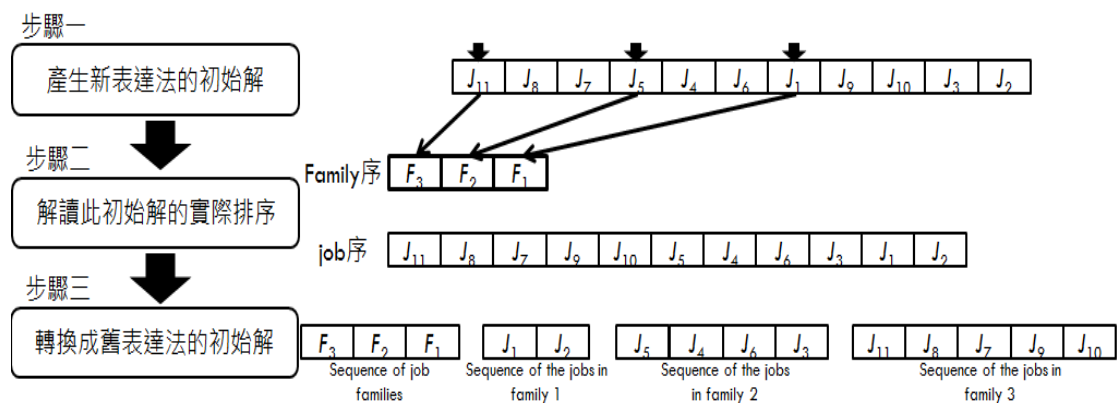


圖 4.2 同初始解的產生方式

表 4.3 同 Seed 與同 Initial Solution 的統計檢定

同 Seed VS 同 Initial solution				
Scenario	平均改善率(%)	t_0	$t_{\alpha, n-1}$	統計結果
舊染色體表達法	0.0003	0.02	2.05	沒有顯著
新染色體表達法	0.0000	0.00	2.05	沒有顯著

4.3 實驗結果與分析

本研究的實驗結果如表 4.4 所示，至於各個欄位所代表的意義如下所示：

- Scenario：表示各個情境的意思，包含整備時間大小、工件族數、機台數
- N ：問題的數目
- $Nw+Ne$ ：贏加平手的數目(三十個問題中新染色體表達法比舊染色體表達法)
- Ne ：平手的數目
- Nw ：贏的數目
- Cn ：瀾集演算法搭配新染色體表達法的三十個問題平均目標值
- Ct ：瀾集演算法搭配舊染色體表達法的三十個問題平均目標值
- $Ave. (%)$ ：此情境三十個問題的績效指標平均改善率，改善率=(舊染色體表達法的目標值-新染色體表達法的目標值)/舊染色體表達法的目標值*100%
- Tn ：新染色體表達法在此情境下三十個問題的平均時間

- T_t : 舊染色體表達法在此情境下三十個問題的平均時間
- Gap : 此情境三十個問題的平均運算時間差距($T_t - T_n$)

表 4.4 本研究之實驗結果

ID	Scenario	makespan							Computation Time(sec)		
		N	N_w+N_e	N_e	N_w	C_n	C_t	$Ave. (%)$	T_n	T_t	Gap
1	SSU33	30	30	30	0	134.47	134.47	0.00	103.65	114.58	10.93
2	SSU34	30	28	24	4	151.09	151.16	0.04	119.72	133.29	13.57
3	SSU44	30	28	20	8	185.73	185.86	0.07	140.03	161.60	21.57
4	SSU55	30	29	7	22	242.75	243.52	0.30	192.94	228.08	35.14
5	SSU56	30	27	3	24	254.40	255.52	0.42	208.52	243.86	35.35
6	SSU65	30	27	3	24	287.08	288.14	0.36	222.74	268.22	45.49
7	SSU66	30	30	2	28	296.24	297.76	0.50	242.43	300.11	57.69
8	SSU88	30	30	0	30	407.32	410.49	0.76	367.70	458.52	90.82
9	SSU108	30	29	0	29	489.80	493.12	0.65	496.29	601.95	105.65
10	SSU1010	30	30	0	30	531.81	537.63	1.07	595.49	682.27	86.78
11	MSU33	30	30	30	0	160.00	160.00	0.00	101.72	113.46	11.74
12	MSU34	30	30	28	2	184.69	184.70	0.01	111.56	123.87	12.31
13	MSU44	30	29	28	1	237.62	237.66	0.01	130.71	149.08	18.37
14	MSU55	30	28	13	15	306.36	306.87	0.16	180.12	209.93	29.81
15	MSU56	30	30	11	19	317.87	318.64	0.23	195.21	238.89	43.68
16	MSU65	30	28	7	21	362.98	363.77	0.20	210.27	258.46	48.19
17	MSU66	30	29	5	24	380.96	382.23	0.32	237.70	283.39	45.69
18	MSU88	30	30	1	29	513.31	516.64	0.63	386.13	464.74	78.61
19	MSU108	30	25	0	25	630.48	632.97	0.39	513.40	635.41	122.01
20	MSU1010	30	27	0	27	663.28	666.91	0.53	567.76	722.64	154.88
21	LSU33	30	30	30	0	228.03	228.03	0.00	106.68	114.71	8.03
22	LSU34	30	30	30	0	239.00	239.00	0.00	108.75	120.63	11.88
23	LSU44	30	30	28	2	323.46	323.48	0.01	127.24	144.11	16.87
24	LSU55	30	30	19	11	416.20	416.50	0.07	169.73	198.44	28.71
25	LSU56	30	30	13	17	437.41	437.99	0.13	190.04	231.05	41.01
26	LSU65	30	29	16	13	492.11	492.51	0.08	195.85	239.45	43.59
27	LSU66	30	30	13	17	514.47	515.28	0.15	212.04	268.07	56.03
28	LSU88	30	30	0	30	702.39	706.30	0.55	361.59	468.91	107.33
29	LSU108	30	22	0	22	852.43	854.11	0.19	480.56	625.20	144.64
30	LSU1010	30	29	0	29	912.75	916.96	0.46	588.56	725.85	137.29
Average of all Scenarios		30	28.80	12.03	16.77	395.22	396.61	0.28	262.17	317.63	55.46

從表 4.4 來看績效衡量指標的平均改善率或運算時間的話，不管情境大小為何，瀾集演算法搭配新染色體表達法在大部分的問題下都比瀾集演算法搭配舊染色體表達法)來的好，但由所有情境平均的地方可以發現績效指標平均改善率為 0.28% 表示改善效果並不明顯。

但若將表 4.4 依照不同的整備時間大小去對於各個情境的績效衡量指標平均改善率做探討，如圖 4.3 可以看出在指標改善率上不管整備時間的大小如何，呈現情境越大，改善效果越顯著的現象。

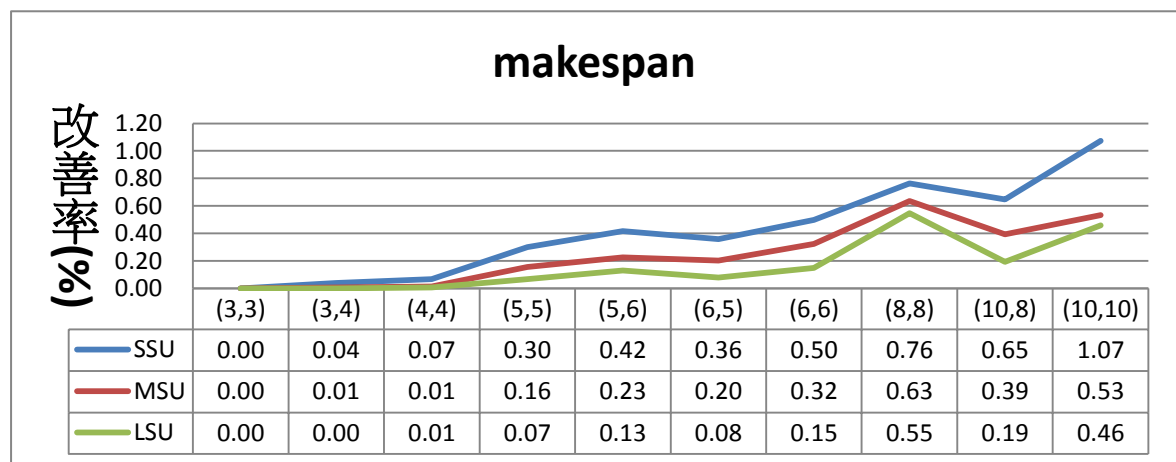


圖 4.3 本研究之績效改善率

圖 4.4 則可以看出在運算時間改善率上不管整備時間的大小如何，呈現情境越大，差距越大的現象。這兩個部分都有一個相同的特性就是情境越大對於本研究所使用的新染色體表達法搭配瀾集演算法是越來越有優勢的。

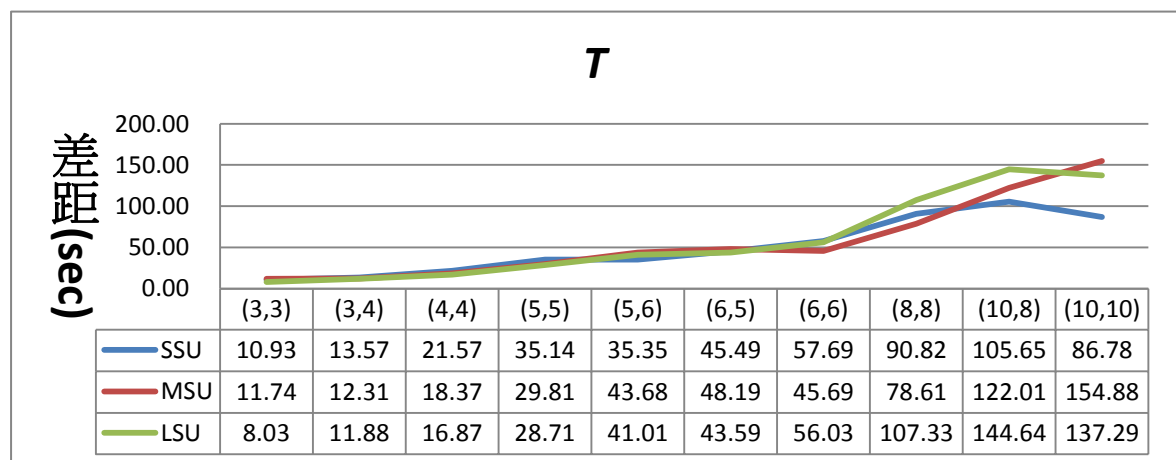


圖 4.4 本研究之運算時間差距

再從表 4.4 可以發現各種整備時間下的各種情境在 30 個問題的平均中贏加平手次數 28.8 個以及贏的次數 16.77 個，表示瀾集演算法搭配新染色體表達法這個方向的确優於瀾集演算法搭配舊染色體表達法，且隨著情境的加大這個效果也越來越顯著。本研究會在 4.4 小節進行統計檢定來驗證結果。

4.4 統計檢定

延續上一節所做的數據資料，使用與 4.2.1 相同的統計檢定，以一個問題當單位，所以先將每個問題中的 15 個 seed 做平均，接著把所有的數據資料依據三個不同大小的整備時間以及全部共分成四個大區塊(LSU、MSU、SSU、Total)，此四大區塊分別有 300(10 個情境*30 個問題)、300(10 個情境*30 個問題)、300(10 個情境*30 個問題)、900(300+300+300)筆資料，相關符號設定如下：

- $Memetic - S_{old,i,j}$, $i=LSU33, LSU34, \dots, SSU1010, j=1, 2, \dots, 30$
- (舊染色體表達法各種整備時間與情境下的各個問題目標值)
- $Memetic - S_{new,i,j}$, $i=LSU33, LSU34, \dots, SSU1010, j=1, 2, \dots, 30$
- (新染色體表達法各種整備時間與情境下的各個問題目標值)
- n ：資料數
- $d_{i,j}(\%)$ ：第 i 個 **Scenario** 下第 j 個問題的改善率
- $\bar{d}(\%)$ ：平均改善率
- S_d ：改善率標準差
- t_0 ：t 值
- t ：檢定值

以 LSU 為例的相關計算如下：

- $n=300$
- $d_{i,j}(\%) = \frac{(Memetic - S_{old,i,j}) - (Memetic - S_{new,i,j})}{(Memetic - S_{old,i,j})} \times 100, i=LSU33,$

LSU34, ..., LSU1010, $j=1, 2, \dots, 30$

- $\bar{d}(\%) = \frac{\sum_{i=LSU33}^{LSU1010} \sum_{j=1}^{30} d_{i,j}}{n}$
- $S_d = \left[\frac{\sum_{i=LSU33}^{LSU1010} \sum_{j=1}^{30} (d_{i,j} - \bar{d})^2}{n-1} \right]^{\frac{1}{2}}$
- $t_0 = \frac{\bar{d}}{S_d/\sqrt{n}}$
- $t=1.96$ (查 t 表，信賴水準 95%，自由度= $n-1$)

當我們所計算出的 t_0 大於查表的 t 時，則表示瀾集演算法搭配新舊染色體表達法是有顯著的差異，但還需觀察平均改善率 \bar{d} ，若此值為正值表示瀾集演算法搭配新染色體表達法是顯著贏瀾集演算法搭配舊染色體表達法，若為負值則結果相反，但當我們所計算出的 t_0 沒有大於查表的 t 時，則表示兩者之間是沒有顯著差異的。表 4.5 為統計檢定的實驗結果，由此可以看出不管是哪個區塊瀾集演算法搭配新染色體表達法都是顯著贏瀾集演算法搭配舊染色體表達法。這結果大大的證明了本研究所提出的想法。

表 4.5 本研究之統計檢定結果

整備時間	$\bar{d}(\%)$	t_0	$t_{\alpha, n-1}$	統計結果
LSU	0.16	10.00	1.96	顯著贏
MSU	0.25	11.98	1.96	顯著贏
SSU	0.42	16.44	1.96	顯著贏
Total	0.28	21.77	1.96	顯著贏

第五章 結論與未來研究方向

本章共分兩個小節；第一節針對本研究進行結論的探討；第二節說明未來的研究方向。

5.1 研究結論

本研究藉由瀾集演算法搭配 Wu *et al.* (2011) 染色體表達法對於具有固定序列、順序相依整備時間特性的流線型生產製造單元排程問題提升其求解品質，利用瀾集演算法搭配舊染色體表達法來當作比較對象，探討在績效指標最大完工時間與運算時間的差異。

本研究所使用的實驗包含三種整備時間(SSU、MSU、LSU)與十種情境(Family數，機台數):{(3, 3), (3, 4), (4, 4), (5, 5), (5, 6), (6, 6), (8, 8), (10, 8), (10, 10)}的組合。這些組合針對不同的工件數與不同的加工時間做30個實驗問題。為了減少因為不同初始解所造成結果的影響，每個實驗問題包含15個Seed。實驗的初始解在新舊表達法上皆使用同Initial Solution，而其終止條件為連續未改變最佳解兩千代。

本研究經過了大量的實驗分析後，認為本研究所使用的新染色體表達法搭配瀾集演算法在績效指標上的確優於舊染色體表達法搭配瀾集演算法，雖然本研究在績效指標和運算時間上的改善率並不多，但由贏加平手的次數以及統計檢定的結果可以看來本研究採用新的染色體表達法來求解問題對於改善績效上是有顯著的幫助，且當情境加大時，績效上面的改善率以及運算時間的節省上都大幅增加。

此結果顯示一個新的重要研究方向，利用巨集啟發式演算法於不同空間求解問題時，新的染色體表達法可以改善空間求解問題的績效。

5.2 未來研究方向

本研究未來希望嘗試其它的績效指標，像是交期相關(Due-date related Measures)的總延遲時間 (Total tardiness)，是否在結果上也能有相同的成效，以確認結論的正確性。接著考慮情境大小對於結果影響的因素，進而嘗試探討本研究所使用的染色體表達法搭配瀾集演算法優於舊染色體表達法搭配瀾集演算法的原因。若能了解原因，在未來求解其他相關的問題上，也能夠用相同的結論進行套用。

根據本研究的結論，可以考慮嘗試具有非固定序列特性的流線型生產製造單元排程問題，探討結果上面是否具有一致性，且使用具有非固定序列的特性相較於固定序列，具有較大的求解空間，雖然在求解時間上可能較久，但可以使得績效指標提升。本研究也可嘗試其它的巨集啟發式演算法，將其皆綜合起來，尋求一致性的結論。



參考文獻

- Abdelmola, A.I., Taboun, S.M., and Merchawi, S., 1998. Productivity Optimization of Cellular Manufacturing Systems. *Computers industrial engineering*, 35 (3-4), 403–406.
- Bouabda, R., Jarboui, B., Eddaly, M., and Rebai, A., 2011. A branch and bound enhanced genetic algorithm for scheduling a flowline manufacturing cell with sequence dependent family setup times. *Computers & Operations Research*, 38, 387–393.
- Chen, J.S., Pan, C.H., and Wu, C.K., 2008. Hybrid tabu search for re-entrant permutation flow-shop scheduling problem. *Expert Systems with Applications*, 34, 1924–1930.
- Chiang, T.C., Cheng, H.C., and Fu, L.C., 2010. A memetic algorithm for minimizing total weighted tardiness on parallel batch machines with incompatible job families and dynamic job arrival. *Computers & Operations Research*, 37, 2257–2269.
- Chiang, T.C., Cheng, H.C., and Fu, L.C., 2011. NNMA: An effective memetic algorithm for solving multiobjective permutation flow shop scheduling problems. *Expert Systems with Applications*, 38, 5986–5999.
- Deutsch, S.J., Freeman S.F., and Helander, M., 1998. Manufacturing cell formation using an improved P-Median model. *Computers industrial engineering*, 34 (1), 135–146.
- França, P.M., Mendes, A., and Moscato, P., 2001. A memetic algorithm for the total tardiness single machine scheduling problem. *European Journal of Operational Research*, 132, 224–242.
- Hendizadeh, S.H., Faramarzi, H., Mansouri, S.A., Gupta, J.N.D., and ElMekkawy,

- T.Y., 2008. Meta-Heuristics for scheduling a flowline manufacturing cell with sequence dependent family setup times. *International Journal of Production Economics*, 111, 593-605.
- Jain, A.S. and Meeran, S., 1999. Deterministic job-shop scheduling: past, present and future. *European Journal of Operational Research*, 113, 390–434.
- Liao, L.M. and Huang, C.J., 2010. Tabu search for non-permutation flowshop scheduling problem with minimizing total tardiness. *Applied Mathematics and Computation*, 217, 557–567.
- Lian, Z., Gu, X., and Jiao, B., 2006. A similar particle swarm optimization algorithm for permutation flowshop scheduling to minimize makespan. *Applied Mathematics and Computation*, 175, 773–785.
- Lin, S.W., Ying, K.C., and Lee, Z.J., 2009. Metaheuristics for scheduling a non-permutation flowline manufacturing cell with sequence dependent family setup times. *Computers & Operations Research*, 36, 1110–1121.
- Metropolis, N., Rosenbluth, A., Rosenbluth, M., and Teller, A., 1953. Equation of State Calculations by Fast Computing Machines. *J. Chemical Physics*, 21 (6), 1087–1092.
- Moscato, P., 1989. On evolution, search, optimization, genetic algorithms and martial arts: Toward memetic algorithms. Caltech Concurrent Computation Program, California Institute of Technology, Pasadena, Tech. Rep.790.
- Neammanee, P. and Reodecha, M., 2009. A memetic algorithm-based heuristic for a scheduling problem in printed circuit board assembly. *Computers & Industrial Engineering*, 56, 294–305.
- Schaller, J. E., Gupta, J. N. D., and Vakharia, A. J., 2000. Scheduling a flowline manufacturing cell with sequence dependent family setup times. *European Journal of Operations Research*, 125, 324–339.

- Spiliopoulos, K. and Sofianopoulou, S., 1998. An optimal tree search method for the manufacturing systems cell formation problem. *European Journal of Operational Research*, 105, 537–551.
- Wu, M.C., Tai, P.H., and 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.
- Zhang, Y., Li, X., and Wang, Q., 2009. Hybrid genetic algorithm for permutation flowshop scheduling problems with total flowtime minimization. *European Journal of Operational Research*, 196, 869–876.
- Zobolas, G.I., Tarantilis, C.D., and Ioannou, G., 2009. Minimizing makespan in permutation flow shop scheduling problems using a hybrid metaheuristic algorithm. *Computers & Operations Research*, 36, 1249–1267.
- 周書賢，「開放型工廠派工法則之模擬研究」，朝陽科技大學工業工程與管理系碩士班，2001。
- 黃維樺，「彈性製造系統中考慮運輸時間之排程探討」，中原大學工業工程系碩士論文，1994。
- 賴士葆，「生產作業管理」，華泰書局，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」，國立交通大學工業工程與管理學系碩士論文，2010。