

國立交通大學
工業工程與管理學系

碩士論文

以模擬退火法求解流線型製造單元排程



研究 生：曾偉杰
指 導 教 授：巫木誠 博 士

中華民國一百年六月

以模擬退火法求解流線型製造單元排程

A Simulated Annealing Approach to Scheduling Flowshop Manufacturing Cell

研究 生：曾偉杰

Student : Wei-Jie Tseng

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

以模擬退火法求解流線型製造單元排程

研究生：曾偉杰

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

模擬退火法是一種啟發式巨集演算法，目前已經被廣泛地運用在求解許多複雜的空間搜尋問題。在之前的研究，專注在如何應用或是提升模擬退火法本身演算法的精進。除了以前的研究議題之外，本研究進行了一個新的議題，一個新的染色體表達法機制應用在模擬退火法是否也能夠提升績效？

流線式製造單元排程問題是一種排成問題，而本論文以此問題為基準來探討兩種不同的染色體表達法應用在模擬退火法上的比較。這兩種方法在流程上是相同的，不同的是在染色體的表達方式上(稱為 S_{old} 和 S_{new})。 S_{old} 是以前的研究所發展出來的方法，而 S_{new} 是由巫木誠(2011)所發展出來的方法，因此這兩種演算法分別稱為 $SA-S_{old}$ 和 $SA-S_{new}$ 。

大量的實驗數據顯示出這兩種演算法在小和中的整備時間(SSU/MSU)情境下，有相同的解品質，然而在大的整備時間(LSU)下， $SA-S_{new}$ 相較於 $SA-S_{old}$ 有更佳的解品質。這一項研究凸顯了一個重要的研究議題，新的染色體表達法運用在啟發式巨集演算法去求解問題，會有不同的結果。

關鍵詞：模擬退火法、表達法、排程

A Simulated Annealing Approach to Scheduling Flowshop Manufacturing Cell

Student : Wei-Jie Tseng

Advisor : Dr. Muh-Cherng Wu

Department of Industrial Engineering and Management
National Chiao Tung University

Abstract

Simulated Annealing (SA), 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 SA 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 SA? A scheduling problem called Flowshop Manufacturing Cell is used as the problem context, and two SAs 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 $SA-S_{old}$ and $SA-S_{new}$ accordingly. Extensive numerical experiments reveal that the two algorithms performs equally well in small and medium setup time (SSU/MSU) scenarios. Yet, $SA-S_{new}$ outperforms $SA-S_{old}$ at large setup time (LSU) 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: Simulated Annealing, solution representation, scheduling

致謝

本論文能夠完成，最重要感謝的是我的指導教授巫木誠博士的細心指導。在整個研究的過程中，碰到瓶頸相當多次，巫老師的肯定與鼓勵，並為我找到對的方向，使我能順利完成這篇碩士論文。在巫老師的敦敦教誨下，使我在唸碩士這兩年來受益良多。而巫老師的教導總是從做人處事的方法、道理下手，這些對我來說，都是未來仍非常受用的。同時也感謝許錫美教授、彭德保教授在論文口試時，所給予的寶貴意見與指導，讓本論文更加完善。

在研究所的兩年中，要感謝同窗的李奕勳、潘冠銘、林耿漢以及陳威宇，陪我一起修課，一起撰寫論文，彼此互相扶持，在我有困難的時候，都不吝惜給與幫助，使我在交大碩士的學習過程中非常順利。也並特別感謝陳振富學長的幫助，在做論文能夠得心應手。我也非常感謝劉芯妤，在碩二的後半段，因為有她一直陪著我，分擔我的煩惱，讓我心情能夠有個抒發的管道，並且對於我的論文或多或少也給與了建議，使我得到相當多的幫助。

我也特別感謝我的家人，在研究所兩年內，我就發生了好多次困境、低潮期，家人總是以我的學業放第一優先考量，如果沒有他們的支持，相信我這兩年的學習過程會更困難許多。最後謹以此論文獻給我最敬愛的家人、師長與朋友。

曾偉杰 于 新竹交大

2011'7'1

目錄

中文摘要.....	.ii
Abstract.....	.iii
致謝.....	.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 研究議題.....	2
1.4 研究方法.....	3
1.5 論文組織.....	3
第二章 文獻探討.....	4
2.1 排程問題之定義與分類.....	4
2.1.1 排程的機器環境.....	4
2.1.2 排程的績效指標.....	5
2.2 製造單元排程問題求解方法.....	6
2.2.1 最佳解法(Optimization Method)	6
2.2.2 近似求解法(Approximation Method)	6
2.3 家族整備時間.....	7
2.3.1 順序相依家族整備時間.....	8
2.3.2 順序獨立家族整備時間.....	8
2.4 模擬退火法.....	8
2.4.1 簡介.....	8
2.4.2 重要名詞解釋.....	10
第三章 研究方法.....	11
3.1 研究問題描述.....	11
3.2 染色體設計與解讀.....	12

3.2.1 舊染色體設計與解讀.....	12
3.2.2 新染色體設計與解讀.....	14
3.3 模擬退火法求解方法.....	15
第四章 實驗情境與結果.....	22
4.1 測試情境.....	22
4.2 前測實驗.....	23
4.2.1 同步初始解設定.....	23
4.2.2 終止條件參數設定.....	24
4.3 實驗結果與分析.....	25
4.4 統計檢定.....	29
第五章 結論與未來研究方向.....	31
5.1 結論.....	31
5.2 未來研究方向.....	32
參考文獻.....	33



表目錄

表 2.1 績效指標分類.....	5
表 2.2 求解方式分類.....	7
表 2.3 家族整備時間分類.....	8
表 4.1 同步初始解比較.....	24
表 4.2 停止次數測試.....	25
表 4.3 結果統計表.....	27
表 4.4 全部結果的平均值.....	28
表 4.5 檢定結果.....	30

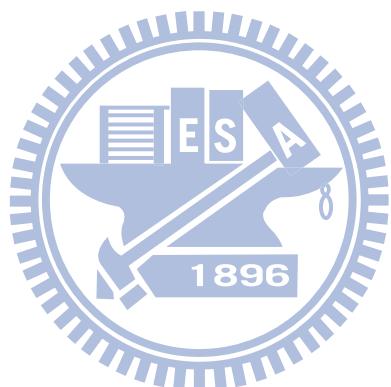


圖 目 錄

圖 1.1 流線式生產.....	2
圖 2.1 模擬退火法尋優過程示意圖.....	9
圖 3.1 染色體表達方式.....	11
圖 3.2 舊染色體的設計.....	13
圖 3.3 舊染色體的解讀.....	13
圖 3.4 新染色體的設計.....	14
圖 3.5 新染色體的解讀第一部分.....	15
圖 3.6 新染色體的解讀第二部分.....	15
圖 3.7 模擬退火法流程圖.....	16
圖 3.8 新表達法初始解轉換成舊表達法.....	18
圖 3.9 舊染色體解讀法交換步驟.....	19
圖 3.10 新染色體解讀法交換步驟.....	19
圖 4.1 改善率.....	29
圖 4.2 時間改善率.....	29



第一章 緒論

第一章分為五個小節，第一節為研究背景，第二節為研究目的，第三節為研究議題，第四節為研究方法，第五節為論文組織。

1.1 研究背景

排程(scheduling)在製造業中扮演一個非常重要的腳色，一旦排程決定，便開始生產，因此在生產開始之前，一個好的排程可以提升整體生產效率，在有限的資源與時間下，使產出最大化，提高機台利用率，而排程有不同的目標，也就是績效指標，如最小完工時間(Minimization Makespan)、總延遲時間(Total Tardiness)，每一個指標都有不同的意義。

單元製造系統(Cell Manufacturing System)是群組技術(Group Technology)的重要應用，它具有流線式生產(Flowshop)的效率和零工式生產(Jobshop)的彈性這兩項優點，單元式製造能夠有效的減少存貨，提高機台使用率，故製造單元排程問題(Manufacturing Cell Scheduling Problem)的議題已被廣泛討論研究。單元製造與分離式製造有所不同，工件(Parts/Jobs)必須以群組的方式存在，類似的工件被分類到同一個群族，形成一個工件族(Part Family)，同一個工件族內的工件在機台上不需要花費整備時間(Setuptime)，因此在生產排程時必須要考慮到兩個層面，工件族間(Among Family)的加工順序和工件族內工件(Within Family)的加工順序，所以單元製造系統較一般分離式排程問題更為複雜。

通常的排程為 NP-Hard (Garey & Johnson,1979)的問題，因為在實際情況下，工件數和機台數不會是一個很小的範圍，這樣一來使得解的排列方式變得很複雜，若是規模較小的時候，可以利用最佳解求解方式，如整數規劃、動態規劃，來求得最佳解，反之，一旦問題變得較大，以及上述所提到，單元製造系統比分離式排程問題更為複雜，此時最佳解求解方式變得不僅費時，還不能求得解，因此就

必須要利用啟發式演算法來進行求解，例如基因演算法(Genetic Algorithms ; GA)、塔布搜尋法(Tabu Search ; TS)、模擬退火法(Simulated Annealing ; SA)等，這些方法來求解，這些演算法可以在相對較短的時間內，求得近似最佳解，甚至最佳解。

1.2 研究目的

許多關於基因演算法的研究大多著重在染色體進化機制的改善，例如染色體的交配方式，染色體突變的方式，或是染色體的篩選，這些方法確實對演算法有所貢獻，而只有少數的論文著重在基因染色體表達法的改善，而 Wu *et al.*(2011)就是在染色體的表達法上著手，利用不同的(新)染色體表達方法，與傳統的(舊)染色體表達法進行比較，結果，確實在績效上有更好的表現，因此，我們想延續這樣的結論，在基因演算法下，新的表達法可以改善績效，若是在其他的演算法上，利用新的表達法是否也是能達到相同的結果，得到更佳的解。



1.3 研究議題

本篇探討的是流線式生產排程，加工的方式如圖 1.1 所示，機台依序排列成一條生產線，每一個工件的加工順序都相同且要經過每一部機台，工件從機台一投入，加工完成後，送至機台二加工，完成後，再轉送至機台三，最後加工完成，工件退出，即是產出，這樣的生產模式即是流線式生產。

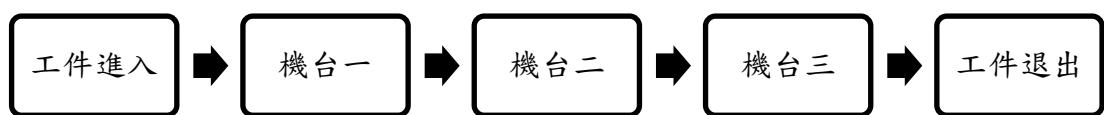


圖 1.1 流線式生產

本篇另外一個議題即是固定序列特性(Permutation Scheduling)之流線式生產排程，由於排程的加工順序，是先排列好工件族間的順序，再排列工件族內工件的順序，而固定序列特性，即是當一開始在機台一上決定好了工件族間和工件族內的加工順序後，在之後的機台上，都必須要使用相同的加工順序來進行工件的加工，不可以任意的更換工件族間或工件族內工件的順序，由於工件的加工順序直接影響到了機台的整備時間，為了能夠得到最佳的效益，工件族間和工件族內工件的排程問題顯得格外重要。

1.4 研究方法

本論文主要是應用模擬退火法針對具順序相依整備時間與固定序列特性之流線式生產排程問題來進行求解，本研究的重點是想驗證從染色體表達法的改變而達到的改善，是否應用到其他的啟發式演算法上，也會有相同的效果。為了能夠有相同的基準，本篇的研究數據是依據 Schaller *et al.* (2000)這篇論文，將整備時間分成長、中，短三種，而在每個整備時間下，機台與工件的數目一共有十種的組合，整合起來共有三十種的情境來進行驗證。

1.5 論文組織

在接下來的章節中，第二章探討與本篇相關的文獻，第三章介紹研究所使用到的演算法與染色體的設計解讀方法與求解的過程，第四章為實驗情境與結果，第五章為結論與未來研究的方向。

第二章 文獻探討

本章共分成四個小節來進行探討：第一節，針對排程的問題進行分類，第二節，探討目前解決排程問題的求解方法，第三節，對家族整備時間進行分類，第四節，對本篇會使用到的模擬退火演算法進行探討。

2.1 排程問題之定義與分類

排程是一種資源分配的決策活動，在一段時間內在限定的生產限制之下，來配置可用的資源，決定機台上工作的加工順序，以達到最佳績效。作業排程又可分為兩大類：負荷安排與工作的處理優先順序，而工作的處理優先順序是指工作的作業順序和時間配置，是序列的排序步驟。

因此，工作的作業先後順序。估計時間和資源產能，都是需要考慮的因素。

2.1.1 排程的機器環境

根據 pinedo (2002)，排程的機器環境主要可分類為下列幾種：

1. 單一機台(Single machine)：

單一機台是所有機器環境中最為單純的，所有的工件加工都在同一台機台上進行加工，所有的工件都會經過這部機台。

2. 完全相同平行機台(Identical machines in parallel)：

許多台功能相同的機台彼此之間不受影響，可以互相獨立的進行加工，任一工件可在任一機台上進行加工動作。

3. 流線型生產(Flowshop)：

所有的工件都會有固定順序的加工步驟，且每一個工件都需經過一部以上的機台加工才能完成。

4. 零工式生產(Jobshop)：

每一個工件要先決定好加工流程，每個工件會有自己加工途徑，彼此不同。

2.1.2 排程的績效指標

績效指標的設立，影響到排程的求解，在開始時，就必須要對目標有所認知，選取適合的衡量績效指標來進行評估。排程中常見的績效指標有下列幾個：

1. 最大流程時間
2. 平均流程時間
3. 最大延遲時間
4. 平均延遲時間
5. 延遲工作百分比

表 2.1 為績效指標的分類。

表 2.1 繢效指標分類

分類	績效衡量指標
完成時間相關	總完成時間 (Completion time) 加權完成時間 (Weighted completion time) 最大完成時間 (Makespan)
流程時間相關	最大流程時間 (Makespan) 平均流程時間 (Mean flow time)
交期相關	總延遲時間 (Total tardiness) 平均延遲時間 (Mean tardiness) 最大延遲時間 (Maximum tardiness)
機台及暫存區相關	平均機台利用率 (Average machine utilization) 平均在製品量 (Average WIP)
成本相關	在製品存貨成本 (WIP inventory cost) 設置成本 (Setup cost)
物料搬運相關	平均物料搬運系統使用率

2.2 製造單元排程問題求解方法

單元製造系統是群組技術的重要應用之一，而群組技術的觀念最早來自於 Mitrofanov (1966)，主要的理念是將機台和工件先進行分群的動作，被分組的工件群內的工件彼此之間的加工特性相似，使得工件族內的工件在進行互換時，不需要浪費太多的整備時間，不同的工件群才需要花費較多的整備時間，而被分組而成的機台群，形成若干的製造單元，機台群內彼此可以互相支援產能，以達到能夠以小批量生產型態卻能大量生產的效益，同時兼具彈性及效率。

而生產排程的問題，大多屬於 NP-hard 的問題，隨著問題的複雜度的增加，求的最佳解便的耗時且困難。通常，在小規模的情況之下，大多都能利用最佳解法(Optimization Methods)求得最佳解，但對於較大的規模，最佳解法變的不可行，取而代之的是利用近似求解法(Approximation Methods)來逼近最佳解。過去的文獻中，在解決排程問題的方法，大致上可以分為兩大類：最佳解法，和近似求解法。



2.2.1 最佳解法(Optimization Method)

以作業研究的方法，主要是根據問題的目標以及資源的限制條件下，建立模式來求解，但在實際的應用上，沒有得到很大的效果，主要是因為模式的建構不易，過多的假設和限制條件，求解過程太過複雜，都是造成這類方法無法被有效廣泛的利用，其他的包含分支界限法(Branch and Bound)、動態規劃法(Dynamic programming)，線性規劃法(Linear programming)等。

2.2.2 近似求解法(Approximation Method)

當排程問的機台、工件數目、限制條件變多，問題的複雜度也相對的增加，如此一來，要透過最佳解法來求得最佳解變得耗時且困難，未來能夠達到解的品質和運算資源上的平衡效益，因此必須利用近似求解法。近似求解法包含：啟發

式演算法(heuristic)和區域搜尋法(local search)兩種。近似解法能夠在較短的時間內，在容忍的誤差之下在可行區域找尋到最佳的可行解或最佳解。主要的優點是在於計算的效率極高，不容易受到問題的大小影響並且解的品質也會有一定的水準。而啟發式演算法常見的有基因演算法(GA)、塔布搜尋法(TS)，模擬退火法(SA)等。

綜合上述提到的兩點，整理成表 2.2

表 2.2 求解方式分類

主概念	使用方法	作者 (年份)
Optimization algorithms	Linear Programming	Javadi <i>et al.</i> (2008)
	Branch and Bound	Hiroshi <i>et al.</i> (1997)
	Dynamic Programming	Webster (2001)
Approximation algorithms	Genetic Algorithms	França <i>et al.</i> (2005)
	Tabu Search	Hendizadeh <i>et al.</i> (2008)
	Simulated Annealing	Lin <i>et al.</i> (2009)
	Simulated Annealing	Naderi <i>et al.</i> (2009)

2.3 家族整備時間

單元製造系統來自於群組技術的概念，也就是將加工特性相同的工件分類到同一個工件族內，因為不同的工件族間的特性不同，相對於工件族內相同加工特性的工件整備時間，工件族間的整備時間來的大很多，因此工件族內的整備時間可以不用考慮，只需要考慮工件族間的整備時間，而家族整備時間由 Potts & Kovalyov (2000)提出，可大致分為兩大類：順序相依家族整備時間(Sequence-Dependent Family Setup Time)與順序獨立家族整備時間(Sequence-Independent Family Setup Time)。

2.3.1 順序相依家族整備時間

家族之間的整備時間會受到前一個家族的影響，前一個被加工的家族，會影響到下一個加工的家族的整備時間，也就是說整備時間是受到前一個家族以及下一個加工的家族，兩個因素的影響。

2.3.2 順序獨立家族整備時間

家族之間的整備時間不會受到前一個家族的影響，不論前一個被加工的家族是誰，都不會影響到下一個加工的家族的整備時間，也就是說整備時間只受到下一個加工的家族自己本身的因素影響。

家族整備時間整理如表 2.3

表 2.3 家族整備時間分類

主概念	使用方法	作者 (年份)
順序相依家族整備時間	塔布演算法	Hendizadeh <i>et al.</i> (2008)
	各種演算法	Lin <i>et al.</i> (2009)
順序獨立家族整備時間	動態規劃	Cheng and Wang (1998)
	優先順序派工法則	Gupta <i>et al.</i> (2000)

2.4 模擬退火法

模擬退火法被廣泛運用於求解許多複雜的問題，下面會針對模擬退火法的特性來進行較紹。

2.4.1 簡介

模擬退火法(Simulated Annealing ; SA)最早的想法是 Metropolis *et al.* (1953) 提出，用以模擬一組原子由一特定高溫逐漸達到冷卻的行為。Kirkpatrick *et al.* (1983)提出並成功地應用在組合最佳化問題中，退火是一種物理過程，一種金屬

物體再加熱至一定的溫度後，它的所有分子在狀態空間中自由運動。隨著溫度的下降，這些分子逐漸停留在不同的狀態。在溫度最低時，分子重新以一定的結構排列，而分子的分布也就是以前面所述的以波茲曼(Boltzamnn)概率分布。模擬退火法中的溫度是隨著退火的時候有所改變，因此如何對溫度作有效的調整就變成整個模擬退火法最重要的一環。

模擬退火法建構在物理模型上，它具備了兩個條件，第一，當溫度夠高時，系統的組態能自由變化，可以在能量表面自由移動或稱為做無規則行走(random walk)，也就是能夠自由選擇可行解；第二，當溫度變小時，系統的組態在能量表面移動將受到限制，並逐漸的向低能量的區域集中，在每一次的疊代過程中，都是以目前解做為中心然後隨機產生新的鄰近解，當鄰近解的目標函數值比目前解的目標函數值較佳時，就以鄰近解取代目前解，如果產生的解比目前解差時，模擬退火法會利用機率函數和控制溫度參數來判斷是否接受新解，這也使得模擬退火法具有能力跳脫區域最佳解，透過降溫的動作來控制收斂的速度，隨著溫度的下降，接受較差解的機率也越來越小，當溫度降到低點時，僅接受較佳的解，進而達到收斂。

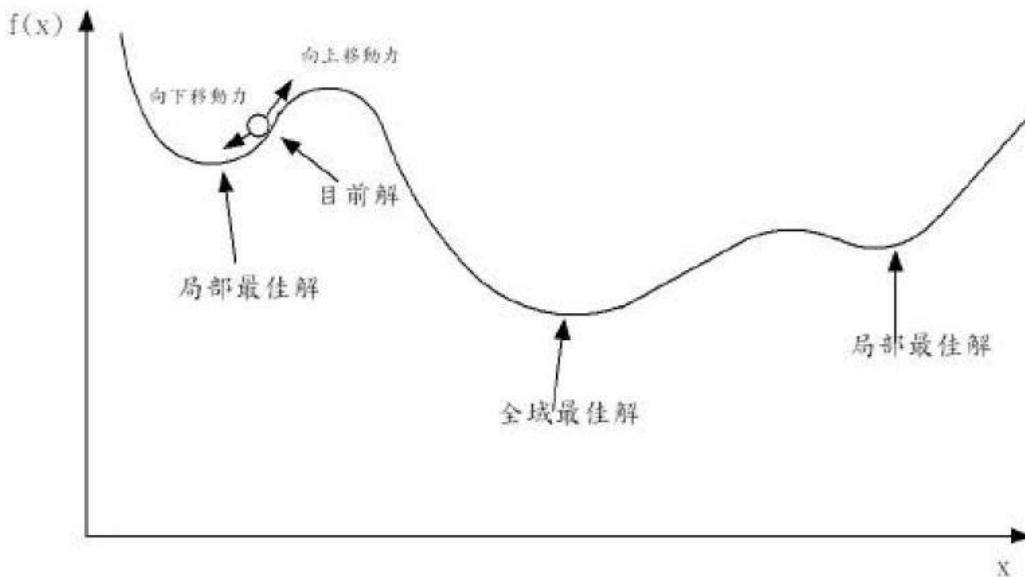


圖 2.1 模擬退火法尋優過程示意圖

2.4.2 重要名詞解釋

1. 狀態：

是指在尋優過程中找到的解，也就是在可行解區域所找到的解。

2. 能量：

在求解問題時，能量即表示問題的目標函數值。

3. 基礎狀態：

應用在求解問題時，是指在尋優過程中能夠找到的最佳解。

4. 波茲曼函數：

設定接受機率的方式為 $PR(A) = \min\{1, \exp(-\Delta f/Tk)\}$ ，利用亂數產生一個介於 0,1 的值 R ，若 $R > PR(A)$ 則接受新狀態 k 。其中 Δf = 狀態 k 之值 - 狀態 $k-1$ 之值； Tk 表第 k 個狀態的控制溫度。

5. 控制溫度：

Tk 用以決定接受機率之高低，適當的控制溫度可使 SA 能有效地跳脫局部解，又能在有效時間內收斂，所以在決定 Tk 是很重要的，適當的控制溫度能夠達到最好的效率，不會浪費時間在局部解裡面搜尋局部最佳解，能夠適時地跳出，更能在快速的時間內收斂達到全域最佳解。

6. 冷卻時程：

一般 SA 在設計初期會讓接受機率較高，以擴大搜尋範圍，但在後期會使接受機率降低以利收斂。

7. 鄰域搜尋法：

是指由起始之解為設定方式，以及由某一可行解尋找下一個鄰域可行解的尋優方式。

8. 停止條件：

用以判斷 SA 重複尋優動作是否結束的依據。設定總尋優次數達到一定數量，或狀態值持續未改善的尋優次數到一定數量。

第三章 研究方法

在本章節會介紹問題的描述，再介紹舊染色體與新染色體的設計與解讀方式的不同，最後說明模擬退火法的詳細求解步驟，並且應用在不同的染色體表達法上，進行驗證。

3.1 研究問題描述

在以往的排程問題大多屬 NP-hard，都是利用演算法的方式來進行求解的動作，而在過去的文獻當中，大多其實都是集中在演算法的進化機制中進行改變以達到更佳的效果，而在 Wu *et al.*(2011)的論文中，在應用基因演算法的情況下，利用不同的染色體表達法確實能夠達到改善，故這次的研究主要是針對在具順序相依家族整備時間之流線型製造單元排程的情況下，應用基因演算法在不同的染色體表達法之下，能夠有效的改善績效值為前提，套用到其他的演算法上，利用這兩種不同的染色體表達法是否一樣能夠達到相當程度的改善，主要的目的也是希望能夠排列出最佳的工件排序組合時，並且最佳化績效指標，降低最大完工時間。

這次研究的重點在於染色體的表達方式不同，因為加入家族的概念，使得染色體的解讀方式更加複雜，如圖 3.1 所示。

演算法	染色體設計方式	作者(年份)
Genetic algorithm	F2 F1 F3 J2 J2 J4 J3 J5 J9 J6 J7 J8	Schaller <i>et al.</i>
		Franca <i>et al.</i>
		Lin <i>et al.</i>
	J4 J9 J5 J1 J3 J7 J6 J2 J8	Wu <i>et al.</i>

圖 3.1 染色體表達方式

本篇的問題限制條件假設相同於 Schaller *et al.* (2000)這篇論文：

1. 本研究僅限於分析靜態式
2. Ready Time=0
3. 同一個工件族內的工件不能被分割處理
4. 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
5. 每個 family 至少要有兩個工件
6. 緩衝區內的在製品無上限
7. 工件完成後即可立即到下一站

更詳細的染色體設計與解讀方式將在下面章節做更詳細的介紹

3.2 染色體設計與解讀

這次主要是研究針對染色體表達方式的不同所造成的改善是否可以應用在其他的演算法上，兩種染色體表達的方式分別為傳統的染色體表達法，稱為舊表達法，和 Wu *et al.*(2011)染色體表達法，稱為新表達法，而下面會分別對舊染色體表達法和新染色體表達法這兩種，在染色體的設計與解讀上的不同，做詳盡的解說。

3.2.1 舊染色體設計與解讀

舊的染色體在設計上，使用一條染色體來表示所有工件的排列順序並且記錄各工件所屬的工件族，每一個工件或每一個工件族視為一個基因，舊染色體將染色體分成 $(1+F)$ 段， F 為工件族數目，如圖 3.2 所示，這例子有三個工件族，共有九個工件，第 1 段紀錄的是工件族在機台上加工的順序，第 $(1+1)$ 段紀錄第一個工件族擁有的兩個工件，且第一個工件族內工件的加工順序為 J_1 、 J_2 ，第 $(1+2)$ 段是紀錄第二個工件族有三個工件，第二個工件族內工件的加工順序為 J_4 、 J_3 、 J_5 ，第 $(1+3)$ 段紀錄第三個工件族有四個工件，第三個工件族內工件的加工順序為

J_9 、 J_6 、 J_7 、 J_8 。

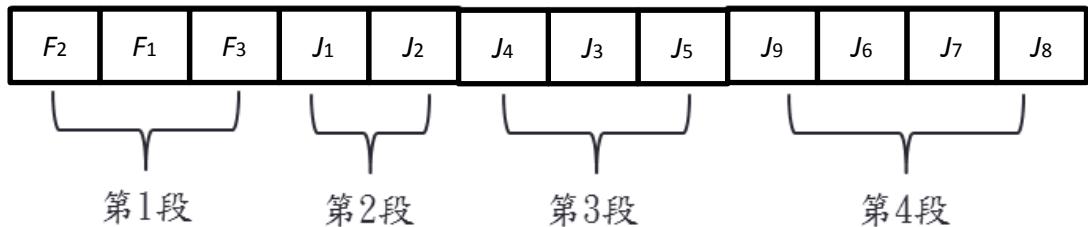


圖 3.2 舊染色體的設計

在染色體解讀主要是決定出兩個部分，第一個部分是工件族之間的加工順序，第二個部分是工件族內的加工順序，在解讀染色體時先從第一段開始解讀起，也就是 $(1+F)$ 段的第一段部分，決定出工件族之間的加工順序，如圖 3.3 所示，一開始就決定了各機台加工工件族的順序是以 F_2 、 F_1 、 F_3 的工件族順序進行加工，而當決定好了各工件族的加工順序後，便依照工件族內工件的順序將工件依序排列出，首先，先排列 F_2 內的工件， J_4 、 J_3 、 J_5 ，在排列 F_1 內的工件 J_1 、 J_2 ，最後排列 F_3 內的工件， J_9 、 J_6 、 J_7 、 J_8 ，如此一來可以解讀出整體的工件加工順序 J_4 、 J_3 、 J_5 、 J_1 、 J_2 、 J_9 、 J_6 、 J_7 、 J_8 ，將解讀出來後的染色體便可以讓進入演算法，計算出其適應值。

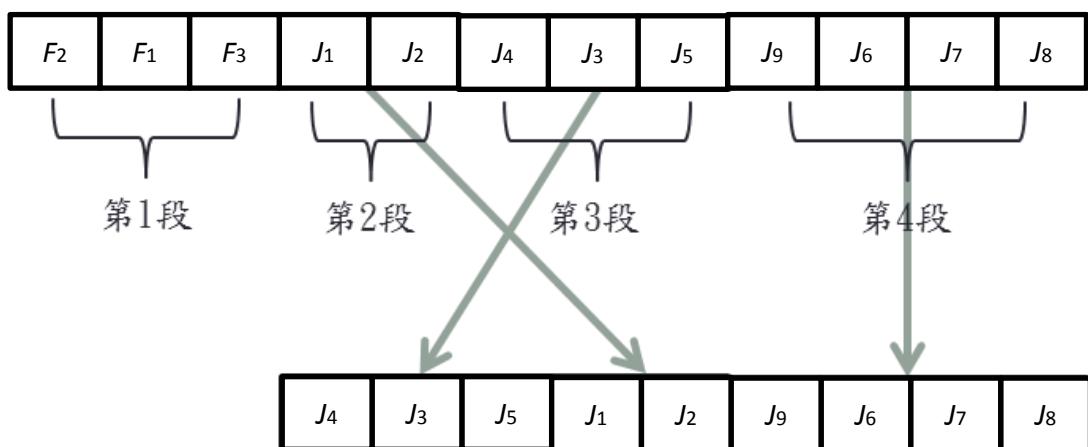


圖 3.3 舊染色體的解讀

3.2.2 新染色體設計與解讀

在新染色體的設計方法上，主要的概念是想將染色體只用一條來顯示出全部的工件與其附屬的工家族屬性，不將染色體分段顯示，每一個工件視為一個基因，每一個基因都有他所附屬的屬性，也就是所屬的工件族，如圖 3.4 所示，共有三個工件族，九個工件，染色體顯示出工件的加工順序， J_4 、 J_9 、 J_3 、 J_1 、 J_6 、 J_7 、 J_5 、 J_2 、 J_8 ，而下面就是每一個工件所對應到的工件族屬性，有就是說 J_1 、 J_2 是屬於第一個工件族， J_3 、 J_4 、 J_5 ，是屬於第二個工件族， J_6 、 J_7 、 J_8 、 J_9 ，是屬於第三個工件族，但這個工件的屬性不屬於染色體的一部分，只是每個工件所附屬的屬性。



在新染色體的解讀上，主要也是分成兩個部分，第一個部分先解讀出工件族間的加工順序，第二部分在決定各工件族內工件的加工順序，在解讀染色體時，由左向右讀取染色體，如圖 3.5 所示，當一開始讀取到 J_4 時，便會找出它屬性的工件族 F_2 ，並且記錄下來，接這往下一個基因 J_9 解讀，它的工件族屬性是 F_3 ，一樣記錄下來，再往下一個基因 J_3 解讀，它的工件族屬性是 F_2 ，由於之前已經記錄過 F_2 ，便跳過此基因往下一個基因 J_1 進行解讀， J_1 的工件族屬性是 F_1 ，如此一來，便可以完成第一部份，得到工件族間的加工順序 F_2 、 F_3 、 F_1 。

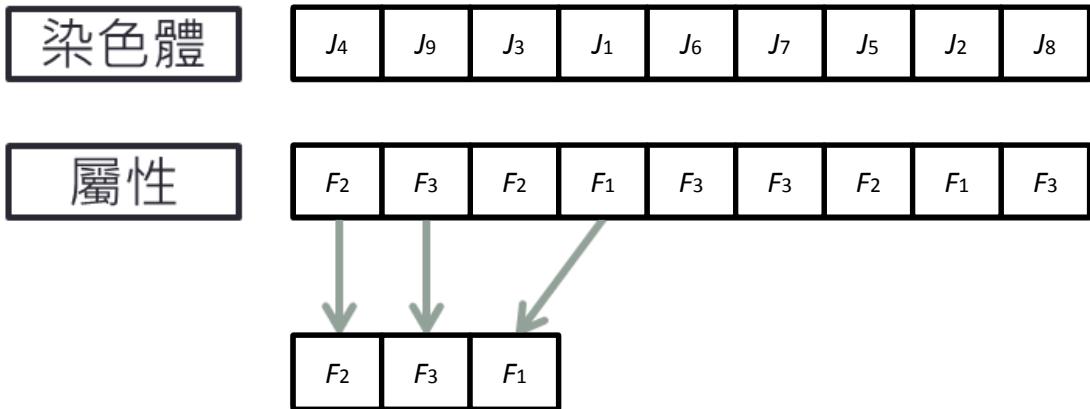


圖 3.5 新染色體的解讀第一部分

接下來進行工件族內工件的加工順序的解讀，如圖 3.6 所示，由於已經決定了工件族間的加工順序 F_2 、 F_3 、 F_1 ，所以從染色體中，由左至右找出所屬於 F_2 的工件，然後依照所讀到的順序排列下來，便可以得到 J_4 、 J_3 、 J_5 ，再進行下一個工件族 F_3 ，由左至右找出染色體內屬於 F_3 的工件，在依照讀取的順序排列下來，可以得到 J_9 、 J_6 、 J_7 、 J_8 ，以此類推到最後一個工件族 F_1 ，可以得到 J_1 、 J_2 ，如此一來可以得到一條新的染色體 J_4 、 J_3 、 J_5 、 J_9 、 J_6 、 J_7 、 J_8 、 J_1 、 J_2 的工件加工順序，經由演算法的計算可以得到其適應值。

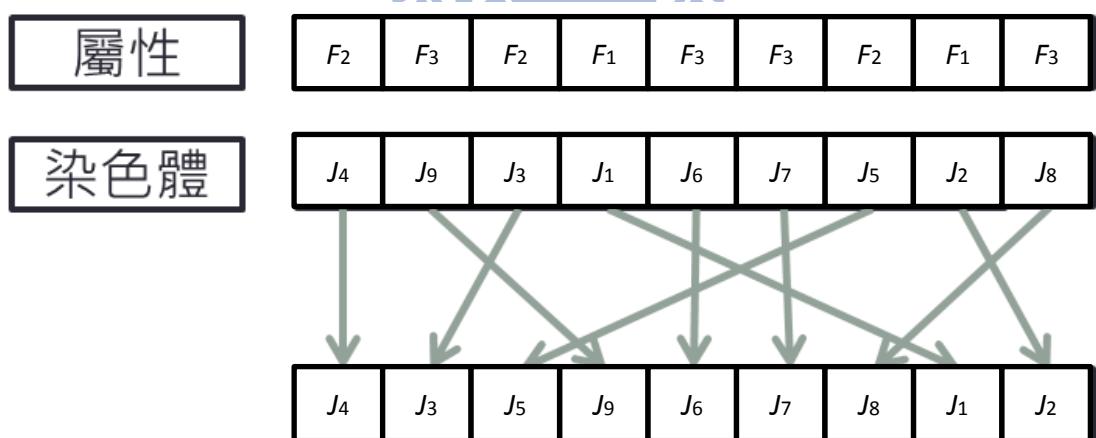


圖 3.6 新染色體的解讀第二部分

3.3 模擬退火法求解方法

本研究是要利用不同的演算法來驗證不同的染色體表達法是否能夠真的達

到有效的改善績效值，在這次研究中，是要應用模擬退火法套用在這兩種演算法上，來探討是否利用模擬退火法有能夠達到相同的目標，接下來就是來詳盡的介紹模擬退火法的求解方法，圖 3.7 為模擬退火法的求解流程圖。本研究所使用的模擬退火法的基本流程架構是使用 Kirkpatrick *et al.* (1983)所提出的。

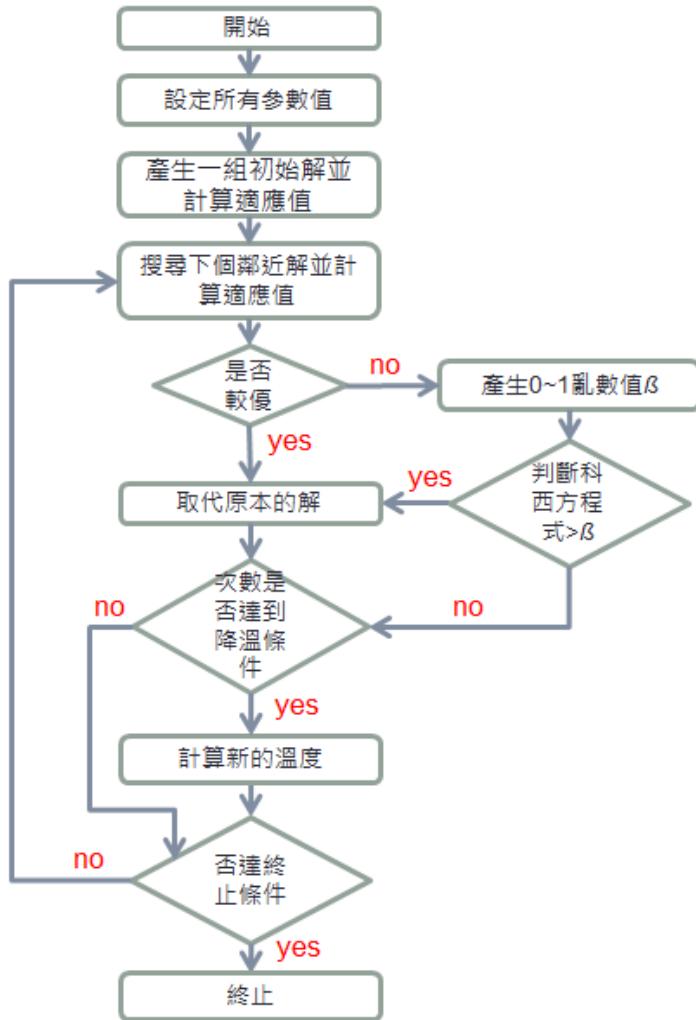


圖 3.7 模擬退火法流程圖

模擬退火法的操作步驟如下：

Step 1：設定所有參數值

在程式一開始的時候要先設定好模擬退火法的相關參數，這些參數會影響到求解所花的時間以及最終解的品質，而本篇主要是以 Lin *et al.* (2009)為基礎，所以相關的參數的設定，也主要是參考 Lin *et al.* (2009)這篇所得到的，例如初始溫度，降溫速度，降溫條件，判斷亂數值，終

止條件等，本篇的起始溫度給予 $T_0 = 1000$ ，而其他相關參數的詳細設定方式將在以下的步驟內提到。

Step 2：產生一組初始接並計算適應值

在最一开始的初始解是以隨機產生的方式，隨機產生一組初始解，之後便決定好工件的加工順序，在經過由 C++所撰寫好的模擬退火法模擬程式，便可以求出染色體的目標值，然後會運用一個適合度函數運算式將目標值轉換成適應值，因為適合度函數能判斷染色體好壞的衡量水準，如此一來，以利判斷在染色體進化過程中，染色體的優劣程度，此排程問題主要是要最小化目標績效值，而經由適合度函數的轉換後，變成最大化適應函數值，即是目標值越小。轉換公式如下：

v_j ：表示第 j 個染色體的目標函數值

f_j ：表示第 j 個染色體的適應函數值

$$f_j = \frac{1}{(1 + v_j)}$$

但由於新舊表達法的不同，若兩種表達法都是以亂數產生初始解，解讀後的加工順序會變得不一樣，而初始解的不同會影響到最後的解品質，我們為了刪除這樣的誤差，能盡量使兩種表達法能夠在同樣的水平基準下進行各自的演化求解過程，所以我們在新的表達法的初始解利用隨機產生，再利用新表達法的初始解轉換成舊表達法的形式，來當成舊表達法的初始解，這樣一來在新舊表達法的初始解就會變得一致，便可排除初始解會導致最後解品質的誤差。

轉換的方法如下圖 3.8 所示，在新表達法之下產生的亂數初始解後，由左向右讀，讀取工件所屬的工件族， J_4 為 F_2 ，便記錄下來，以此類推， J_9 為 F_3 ，但 J_3 為 F_2 ，而 F_2 已經被記錄過，便跳過往後讀取， J_1 為 F_1 ，一但全部的家族都被記錄後，結束第一階段家族序的讀取。接下來要記錄工件的加工序， F_1 內的工件加工順序依照新表達法的工件順序，

由左向右把 F_1 的工件依序記錄下來，為 J_1 、 J_2 ，再依此類推， F_2 內的工件加工順序為 J_4 、 J_3 、 J_5 ， F_3 內的工件加工順序為 J_9 、 J_6 、 J_7 、 J_8 ，如此便完成了新表達法的初始解轉換成舊表達法的初始解步驟。

新解讀法

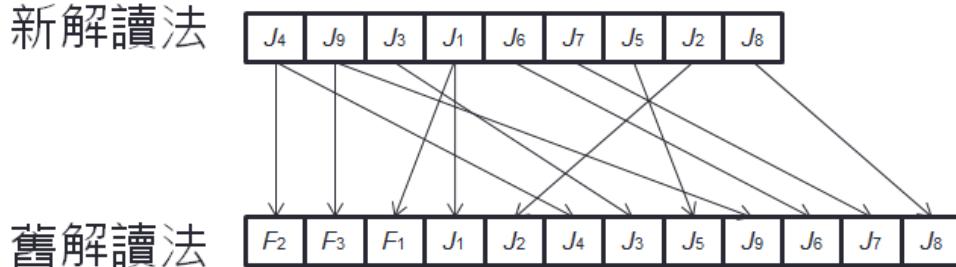


圖 3.8 新表達法初始解轉換成舊表達法

Step 3：搜尋下個鄰近解並計算適應值

在找尋下一個鄰近解時，主要是依照目前的染色體來生成下一個鄰近解染色體，而產生鄰近解的方式是以基因兩兩交換的方式來進行操作，由於在染色體的解讀方式分為舊染色體與新染色體兩種，下面會分別針對這兩種染色體兩兩交換的方式來進行介紹。

舊染色體表達法的搜尋鄰近解的方式如圖 3.9 所示，在舊的染色體設計中，會將染色體分成 $(1+F)$ 段，因此在找尋下一個鄰近解時，會在每一個區段內，各自隨機產生兩個點來進行兩兩交換，在第一段染色體中，選取 F_2 和 F_3 進行兩兩交換，第二段中，選取 J_1 和 J_2 進行兩兩交換，以此類推到最後一段，經過兩兩交換後便可以得到一組新的染色體，而經由解讀後的工件排序如圖 3.8 所示，最後再將解讀後的染色體經由程式運算，可以得到一個新的鄰近解的目標值，通過適應值函數的轉換便得到其適應值。

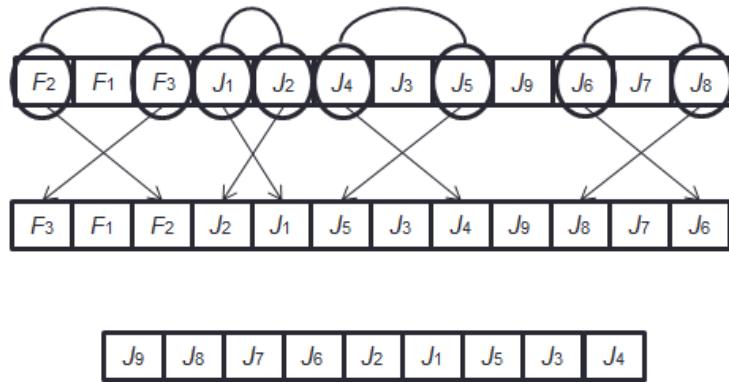


圖 3.9 舊染色體解讀法交換步驟

新染色體表達法的搜尋鄰近解的方式如圖 3.10 所示，因為在新的染色體表達法中，只用一條染色體來進行表達，並不會有分段的現象，所以在搜尋鄰近解的過程中，整條染色體只會隨機產生兩個點來進行兩兩交換，在圖 3.10 中可看出選取了 J_9 和 J_1 來進行交換的動作，交換後可以得到一條新的染色體，通過染色體的解讀可以得到新的工作排序，如圖 3.10 所示，最後再將解讀過後的染色體經由程式的運算，可以得到其目標值，利用適應值函數轉換將其目標值轉換成適應值。

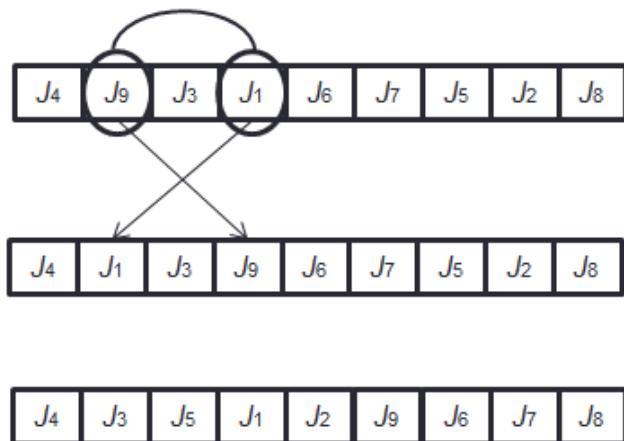


圖 3.10 新染色體解讀法交換步驟

Step 4：判斷是否較優

經由計算後的鄰近解適應值將與原本的適應值進行比較，若鄰近解的適應值沒有優於原本的適應值，繼續下一個步驟 Step 5，但若是新產

生解的適應值優於原本解的適應值，直接跳到 Step 7 來進行。

Step 5：產生 0~1 的亂數值 β

在此步驟會隨機產生一個 0~1 的亂數值 β ，然後與之後的科西方程式來做比較的動作。

Step 6：利用科西方程式判斷

先前介紹到的模擬退火方法，再進行較差的解取代動作時，是以波茲曼函數 $PR(A) = \min\{1, \exp(-\Delta f/Tk)\}$ 來設定接受機率的條件，不過在 Lin *et al.* (2009) 中提到的模擬退火法，是使用另外一種判斷式

$\frac{T^2}{(T^2 + (\Delta E)^2)}$ 來進行取代動作的判斷，為了讓比較的方式有相同的基礎，

本篇所應用到的模擬退火法的判斷式也改成與 Lin *et al.* (2009) 相同，

以科西方程式 $\frac{T^2}{(T^2 + (\Delta E)^2)}$ 來進行，若計算出來的科西方程式大於所生成的亂數 β 值，便執行 Step 7，相反的若是科西方程式小於產生的亂數值 β ，新產生的鄰近解將刪除掉，利用原本的解來繼續進行，接著執行 Step 8。

Step 7：取代原本的解

此步驟將新產生的解取代掉原本的解，並保留住新的解。

Step 8：次數是否達到降溫條件

模擬退火在執行到一定的條件時，便會執行降溫的動作，而在本篇

降溫的條件也是基於 Lin *et al.* (2009) 為基礎，將降溫條件設為

$liter = (F * m + n) * 1000$ ，其中 F 表示工件族數目， m 表示機台數目， n 表示總工件數，若執行的次數有到達降溫條件，執行 Step 9，反之則跳回 Step 10。

Step 9：計算新的溫度

本篇的降溫動作是以一定比率來進行降溫，降溫速度給予 $\alpha = 0.9$ ，使用 $\alpha \times T$ 來生成下一個溫度。

Step 10：是否達到終止條件

由於每次產生新一代的解，都會把最佳的解給記錄下來，若產生的新解的值有優於最佳解，便會取代掉原本的最佳解，反之則否，而此演算過程的終止條件就是當最佳解連續幾代都不進行改變，我們視為演化到了極限，也就是說如果再多跑下去，最佳值也不會找到更好的解，此終止條件的世代數目，我們經過測試後得到了當最佳值連續 480 萬次的世代交替都不改變，我們將會終止。



第四章 實驗情境與結果

本次的研究主要是探討兩種不同的染色體表達法套用在不同的演算法上是否有相同的效果，而實驗情境是依據Schaller *et al.* (2000)而來，因此在基於使用上的基準相同，故本篇採用與Schaller *et al.* (2000)相同的情境來進行實驗，探討這兩種不同的染色體表達所帶來成效的不同。本研究的操作環境如下：

使用語言：C++

編譯器(Compiler)：MSVC9.0

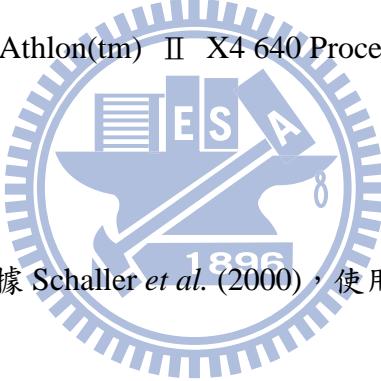
編寫環境(IDE)：Visual Studio 2008

使用作業系統(OS)：Windows 7 Enterprise 32-bits

記憶體(Memory)：4G

中央處理器(CPU)：AMD Athlon(tm) II X4 640 Processor 3.00 GHZ

4.1 測試情境



在這次的研究環境依據 Schaller *et al.* (2000)，使用相同的操作環境，假設如下：

1. 工件在每一個機台上的處理時間服從均勻分配 $U[1,10]$ 。
2. 工件族內的工件數目服從均勻分配 $U[1,10]$ 。
3. 具有三種不同的整備時間，大、中、小型：

Small setups (SSU) : $U[1,10]$

Medium setups (MSU) : $U[1,50]$

Large setups (LSU) : $U[1,100]$

4. 每種整備時間類別下有 10 種情境 $[F/m]$ ， F 表工件族數目， m 表機台數目： $[3/3]$ 、 $[3/4]$ 、 $[4/4]$ 、 $[5/5]$ 、 $[5/6]$ 、 $[6/5]$ 、 $[6/6]$ 、 $[8/8]$ 、 $[10/8]$ 、 $[10/10]$ 。
5. 每種情境產生 30 個問題，每個問題執行 15 個 SEED。

4.2 前測實驗

本章節將分成兩個部分，第一部分介紹為何演算的種止條件要設定為 480 萬個世代沒有進化，判定為終止，第二部分將分析同初始解與不同初始解的差異。

4.2.1 同步初始解設定

由於本篇是研究針對兩種不同的表達法上的研究，而在一開始產生初始解時，雖然是使用相同的亂數起始點，但表達法的不同卻會產生不同的起始解，我們認為這樣的方式在比較兩種表達法應用在同樣一套演算法流程上，會有失公平，因此我們利用新表達法所亂數產生的初始解進而轉換成舊表達法的初始解，以利舊表達法接下來的演算過程，如此一來，在新舊表達法上就會有相同的初始解。

而為了想要驗證這樣的轉換方式在舊表達法上是否確實會有影響，我們進行了實驗，舊表達法自己產生的初始解與利用新表達法的初始解轉換成舊表達法的初始解，這兩種方法的比較，並利用統計檢定來驗證結果。

我們以 Paired t-test 為統計檢定方式，假設 $S : \{SSU33, SSU34, \dots, MSU33, MSU34, \dots, LSU33, LSU34, \dots, LSU1010\}$ 代表 30 個情境的集合。 $u_{s,i,1}$ 代表在情境 s 下，以不同初始解為搜尋起點的新舊表達法目標值平均，而 $u_{s,i,2}$ 代表在情境 s 下，以相同初始解為搜尋起點的新舊表達法目標值平均，其中 $s \in S, i = 1, 2$ (1 代表新表達法，2 代表舊表達法)。計算公式如下所示。

$$d_{s,i}\% = (u_{s,i,1} - u_{s,i,2}) * 100 / u_{s,i,1}$$

新舊表達法在 30 個 instance 的樣本總平均：

$$\bar{d}_i\% = \frac{\sum_{s \in S} d_{s,i}}{30}$$

新舊表達法在 30 個 instance 的樣本標準差：

$$SD_i = \sqrt{\frac{\sum_{s \in S} (d_{s,i} - \bar{d}_i)^2}{29}}$$

計算 t_0

$$t_0 = \frac{\bar{d}_l}{SD_i / \sqrt{30}}$$

下表 4.1 顯示出，因為不論轉換與否，新表達法都不會受到影響，而舊表達法在這兩者上會有不同，不過經過統計檢定後，發現沒有顯著差異，因此我們認為若是表達法不同，亂數產生的初始解與利用新表達法的初始解轉換成舊表達法的初始解，兩者沒有差異，也就是說不論使用哪種方式，皆可以。而本篇還是使用新表達法所產生的初始解轉換成舊表達法的初始解的方式來進行實驗。

表 4.1 同步初始解比較

同初始解 vs 不同初始解				
Scenario	平均改善率(%)	$t_{(\alpha, n-1)}$	t 值	統計結果
舊表達法	-0.02	1.07	2.05	沒有顯著差異
新表達法	0.00	0.00	2.05	沒有顯著差異

4.2.2 終止條件參數設定

為了能夠得到連續最佳解多少世代當作收斂條件，我們進行了實驗，並將實驗的結果進行統計檢定。

我們挑選連續 480 萬代最佳解未改變與連續 550 萬代最佳解未改變，並且新表達法與新表達法，舊表達法與舊表達法，彼此兩兩比較。

檢定方式與 4.2.1 相同，以 Paired t-test 為檢定方式，假設 $S : \{SSU33, SSU34, \dots, MSU33, MSU34, \dots, LSU33, LSU34, \dots, LSU1010\}$ 代表 30 個情境的集合。
 $u_{s,i,5500000}$ 代表在情境 s 下，最佳解連續 5500000 代不變的新舊表達法目標值平均，而 $u_{s,i,4800000}$ 代表在情境 s 下，最佳解連續 4800000 代不變的新舊表達法目標值平均，其中 $s \in S, i = 1, 2$ (1 代表新表達法，2 代表舊表達法)。計算公式如下所示。

$$d_{s,i}\% = (u_{s,i,5500000} - u_{s,i,4800000}) * 100 / u_{s,i,1}$$

新舊表達法在 30 個 instance 的樣本總平均：

$$\bar{d}_i\% = \frac{\sum_{s \in S} d_{s,i}}{30}$$

新舊表達法在 30 個 instance 的樣本標準差：

$$SD_i = \sqrt{\frac{\sum_{s \in S} (d_{s,i} - \bar{d}_i)^2}{29}}$$

計算 t_0

$$t_0 = \frac{\bar{d}_i}{SD_i / \sqrt{30}}$$

由下表 4.2 可以看出，不論在新表達法或是舊表達法之下，最佳解連續 480 萬代未改變與最佳解連續 550 萬代未改變的兩種最終解並沒有顯著差異，也就是說，若收斂條件設定為最佳解連續 480 萬代未改變所求出的解為基礎，再將收斂條件拉大到最佳解連續 550 萬代未改變，解品質沒有進步，因此我們便考慮較小的收斂條件，最佳解連續 480 萬代未改變當作我們的最終終止條件。

表 4.2 停止次數測試

收斂代數(480 萬 vs 550 萬)				
Scenario	平均改善率(%)	$t_{(\alpha, n-1)}$	t 值	統計結果
舊表達法	0.01	1.99	2.05	沒有顯著差異
新表達法	0.01	1.44	2.05	沒有顯著差異

4.3 實驗結果與分析

本次研究是比較兩種同染色體表達法應用在模擬退火法的優劣比較，而為了能夠看出優劣，定義了一個改善率的計算式，計算方式如下。

改善率： $r = \frac{(obj_{SA-FT} - obj_{SA-F})}{obj_{SA-FT}} * 100\%$ ，其中 obj_{SA-FT} 表示利用舊染色體表

達法應用在模擬退火法上， obj_{SA-F} 表示利用新染色體表達法應用在模擬退火法上，如此便可評估兩種不同的表達法之求解品質的依據，結果如表 4.3 所示。

其中表格最上方的內容為下列所式：

Scenario：表示為整備時間的情境大小，有 SSU、MSU，LSU 三種不同大小的整備時間情境

N ：每種情境總共的問題數有 30 個

Nw ：在 30 個問題中，新的表達法贏舊的表達法的問題數

Ne ：在 30 個問題中，新的表達法與舊的表達法平手的問題數

Cn ：在 30 個問題中，新表達法的平均 makeapsn

Ct ：在 30 個問題中，舊表達法的平均 makeapsn

$Ave.$ ：平均改善率，若為正值，表示新的解品質較優

$Max.$ ：最大改善率，若為正值，表示新的解品質較優

Tn ：新的表達法的運算時間

Tt ：舊的表達法的運算時間

Gap ：新表達法與舊表達法求解時間的改善率，若為正值表示新的表達法擁有較快速的求解時間

表 4.3 結果統計表

ID	Scenario	makespan							Computation Time(sec)		
		N	N _{w+Ne}	N _e	N _w	C _n	C _t	Ave. (%)	T _n	T _t	Gap
1	SSU33	30	30	22	8	134.49	134.85	0.37	2.36	2.59	0.23
2	SSU34	30	24	12	12	151.45	151.76	0.29	3.40	3.46	0.06
3	SSU44	30	17	8	9	186.50	186.43	-0.03	5.43	5.68	0.25
4	SSU55	30	13	4	9	244.89	244.78	-0.03	10.18	10.97	0.79
5	SSU56	30	12	3	9	256.94	256.86	-0.02	12.27	13.20	0.92
6	SSU65	30	14	2	12	289.85	289.71	-0.04	14.79	15.83	1.04
7	SSU66	30	15	2	13	299.47	299.45	0.01	17.71	18.97	1.26
8	SSU88	30	8	0	8	413.55	413.03	-0.13	43.13	45.94	2.80
9	SSU108	30	4	0	4	499.67	497.49	-0.45	66.43	70.77	4.33
10	SSU1010	30	7	1	6	543.64	542.33	-0.25	87.87	92.50	4.63
11	MSU33	30	29	15	14	160.04	160.56	0.36	2.09	2.35	0.26
12	MSU34	30	29	19	10	184.80	185.18	0.24	3.13	3.21	0.08
13	MSU44	30	28	16	12	237.85	238.09	0.11	5.48	5.64	0.16
14	MSU55	30	21	6	15	307.71	307.95	0.09	10.13	10.91	0.78
15	MSU56	30	18	4	14	319.82	320.08	0.10	11.64	12.66	1.03
16	MSU65	30	16	2	14	364.98	365.16	0.05	14.48	15.57	1.09
17	MSU66	30	17	3	14	383.88	383.82	-0.01	17.71	18.98	1.27
18	MSU88	30	10	0	10	520.21	519.44	-0.14	42.77	45.64	2.87
19	MSU108	30	13	0	13	639.66	638.26	-0.22	68.49	72.61	4.12
20	MSU1010	30	8	0	8	674.35	672.64	-0.25	85.49	90.63	5.14
21	LSU33	30	30	19	11	228.04	228.49	0.21	2.54	2.75	0.21
22	LSU34	30	29	15	14	239.03	239.64	0.28	2.82	2.98	0.16
23	LSU44	30	27	10	17	323.52	324.41	0.31	5.18	5.37	0.19
24	LSU55	30	28	6	22	416.92	417.69	0.19	9.73	10.56	0.83
25	LSU56	30	22	2	20	438.81	439.16	0.08	12.07	12.98	0.92
26	LSU65	30	24	3	21	493.08	493.93	0.18	14.79	15.75	0.96
27	LSU66	30	24	3	21	516.01	516.67	0.13	16.83	18.17	1.34
28	LSU88	30	22	0	22	707.75	708.71	0.14	41.28	44.26	2.98
29	LSU108	30	21	0	21	856.95	857.85	0.10	65.81	70.52	4.71
30	LSU1010	30	22	0	22	920.54	922.01	0.16	88.00	92.88	4.88
Average of all Scenarios		30	19.40	5.90	13.50	398.48	398.55	0.06	26.13	27.81	1.68

由表 4.3 看出，在大部分的情境之下，新染色體表達法的解品質大多不會優於舊染色體表達法的解品質，只有在大的整備時間下，新染色體的表達法才會明顯優於舊染色體表達法，在大多中、小整備時間之下，兩種表達法的差距其實並不明顯，因此推論新染色體表達法應用在模擬退火法的情況下，只有在大的整備時間之下才會適用。

表 4.4 為各情境與全部情境的平均值，從表中可以看出，隨著整備時間的越來越大，新表達法營的數量會越來越多，而在平均改善率上，也會越來越多。在整體平均的情況下，新染色體表達法贏過或平手於舊染色體表達法的次數總和，在 30 次的問題中，有 19.4 次，在平均改善率也是以些微的差距，以 0.06% 的差距贏舊染色體表達法，另外可以注意到一點的是，在運算的時間上，不論情境的大小或者機台與工件族的數目大小，新染色體解的運算時間全部優於舊染色體表達法，且平均的時間改善率為 1.68%，新染色體表達法在運算時間全部進步。

表 4.4 全部結果的平均值

Scenarios	makespan 1896							Computation Time(sec)		
	N	Nw+Ne	N _e	N _w	Cn	Ct	Ave. (%)	Tn	Tt	Gap
SSU	30	14.4	5.4	9	302.05	301.67	-0.03	26.36	27.99	1.63
MSU	30	18.9	6.5	12.4	379.33	379.12	0.03	26.14	27.82	1.68
LSU	30	24.9	5.8	19.1	514.06	514.86	0.18	25.91	27.62	1.72
Total	30	19.4	5.90	13.50	398.48	398.55	0.06	26.13	27.81	1.68

圖 4.1 為三種情境的改善率狀況，可以看出各種情境下，只有在大的整備時間以及在中、小整備時間下，機台與工件族數目偏小的情況下，新染色體表達法才會贏過舊染色體表達法。

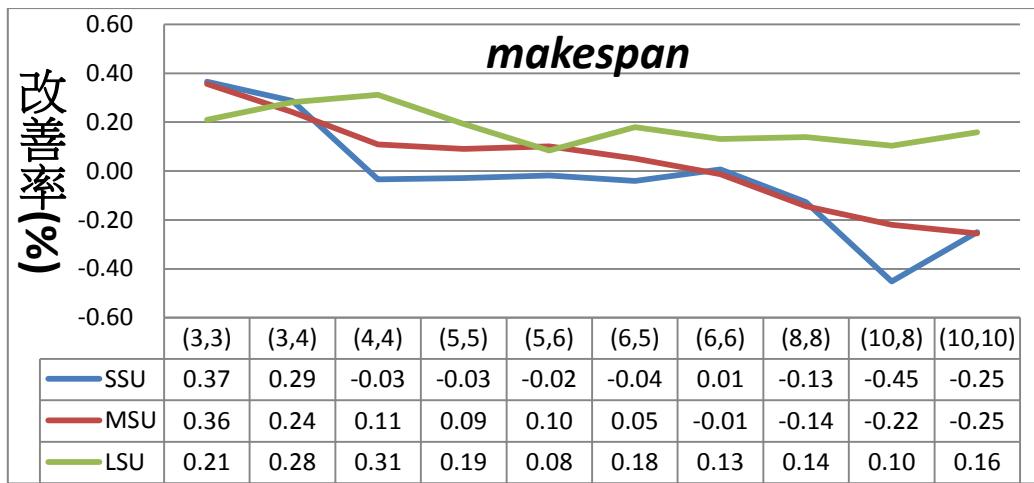


圖 4.1 改善率

圖 4.2 為三種情況的時間差距狀況，可以看出在大、中，小情境之下，新染色體表達法運散時間都以一定的比率優於舊染色體表達法，尤其在機台數與工件族數目較大的情況之下，更顯得明顯。

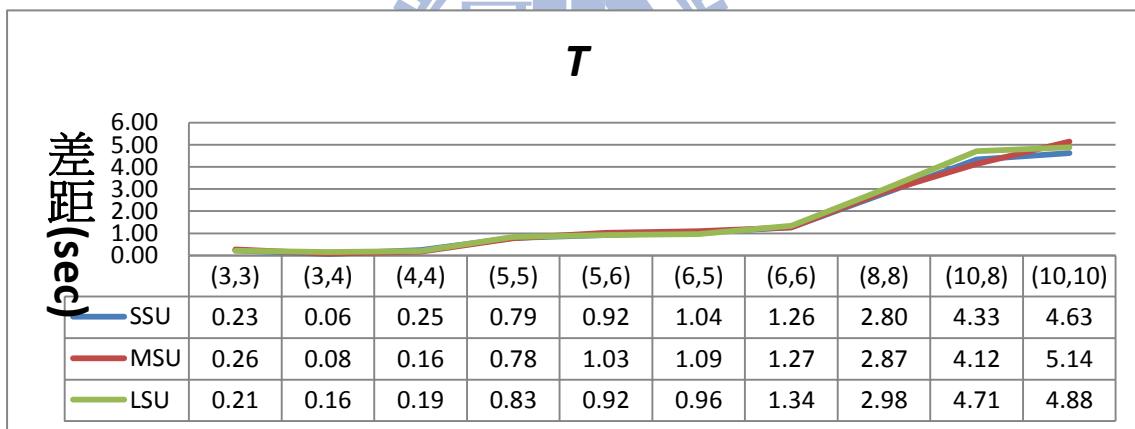


圖 4.2 時間改善率

4.4 統計檢定

為了更進一步的確認上述數據的真實性，我們利用 Paired Sample t-test 統計方法來驗證，我們將檢定分成四個部分，其中包含 SSU、MSU、LSU 三種整備時間情境下的部分以及整體的平均表現。

檢定的目的是想確信在統計檢定下，新表達法是否有顯著的優於舊表達法，

以相對的改善率為樣本，來進行我們的檢定，而為了能夠做整體平均的表現績效，在檢定之前我們需要先行將資料做正規化的動作。

假設 $S : \{\text{SSU33}, \text{SSU34}, \dots, \text{MSU33}, \text{MSU34}, \dots, \text{LSU33}, \text{LSU34}, \dots, \text{LSU1010}\}$

表示 30 個情境的集合， $u_{new,s,i}$ 及 $u_{old,s,i}$ 代表在情境 s 下第 i 個實驗新、舊表達法的解平均，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 $d_{s,i} = (u_{old,s,i} - u_{new,s,i}) * 100\% / u_{old,s,i}$ 。

不同整備時間情境及整體的樣本平均：

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

不同整備時間情境及整體的樣本標準差：

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

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

不同整備時間情境及整體檢定用 t 值：1896

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

經過檢定後，由表 4.5 可知，在 95% 信心水準($\alpha = 0.05$)下，在中與小的整備時間情境下，兩種表達法的表現沒有顯著差異；在大整備時間情境下，新表達法的解品質顯著優於舊表達法的解品質；而在整體平均表現上，新表達法的解品質顯著優於舊表達法的解品質。

表 4.5 檢定結果

Scenario	平均改善率	t_0	t 值	統計結果
LSU	0.18	8.69	1.96	顯著贏
MSU	0.03	1.27	1.96	沒有顯著
SSU	-0.03	0.93	1.96	沒有顯著
Total	0.06	3.97	1.96	顯著贏

第五章 結論與未來研究方向

第五章分為兩小節，第一節為本篇論文的結論，第二節為未來還可探討的研究方向。

5.1 結論

本論文主要是應用模擬退火法針對具順序相依整備時間與固定序列特性之流線式生產排程問題來進行求解，本研究的重點是想驗證從染色體表達法的改變而達到的改善。為了能夠有相同的基準，本篇的研究數據是依據 Schaller *et al.* (2000)這篇論文，將整備時間分成短、中，長三種(SSU、MSU、LSU)，而在每個整備時間下，機台與工件的數目一共有十種的組合(Family 數，機台數):{(3, 3), (3, 4), (4, 4), (5, 5), (5, 6), (6, 6), (8, 8), (10, 8), (10, 10)}, 整合起來共有三十種的情境來進行驗證，每個實驗情境又包含了 15 個 seed，並且取其平均，而實驗的初始解在新舊表達法上為相同的起始解，終止條件設為最佳解連續 480 萬代未改變。

經過實驗與統計檢定後的數據顯示，在大的整備時間之下，新的染色體表達法在 makespan 績效指標之下，會顯著的優於舊的表達法，平均改善率為 18%，而在小與中的整備時間之下，新的解讀法和舊的解讀法兩者之間在 makespan 績效指標之下沒有顯著的差異，也就是說隨著機台的整備時間越大，新表達法的求解品質會越顯著優於舊表達法的求解品質。而新舊表達法的求解時間，新的表達法不論在大、中，小的整備時間上，都會稍微較舊的表達法快上一點，尤其在機台數目與工件族數目較大時更為顯著。

因此在最小化最大完工時間為目標之下，在大的整備時間下，SA 應用新表達法會比舊表達法得到更佳的工件加工的排程組合。

5.2 未來研究方向

針對本篇最後的結論，在小、與中的整備時間之下，新的表達法與舊的表達法會有差不多的解品質，而在大的整備時間之下，新的表達法會優於舊的表達法，來進行研究為何會造成如此的結果，可以試著從演算法的進化特性來進行分析，另外去分析各個整備時間所對模擬退火法所造成的影響。

而為了希望能夠多方面的進行研究，可以考慮使用考慮 non-permutation 的情況，因為在開始排列工件族順序時，只是很單純的以第一台機台最為考量，後續的機台便依據第一台的加工順序往下加工，另外，在上段論點中提到，隨著情境的增大，新表達法有逐漸趨近舊表達法，若是加大情境，或許會有不同的結果，因此未來希望能夠將原本的 permutation 改為 non-permutation 的情況，再利用兩種不同的表達法來進行比較，探討結果會有什麼不同的情況，另外可以將原本的情境再擴大，使得目前的情境更為複雜，例如：增加機台，家族的數量，來進行後續的研究方向。



參考文獻

- Cheng, J., Kise, H., and Matsumoto, H., 1997. A branch-and-bound algorithm with fuzzy inference for a permutation flowshop scheduling problem. *European Journal of Operational Research*, 96, 578-590.
- Cheng, T.C.E. and Wang G., 1998. Batching and scheduling to minimize the makespan in the two-machine flowshop. *IIE Transactions*, 30, 447-453.
- Franca, P.M., Gupta, J.N.D., Mendes, A.S., Moscato, P., and Veltink, K.J., 2005. Evolutionary algorithms for scheduling a flowshop manufacturing cell with sequence dependent family setups. *Computers & Industrial Engineering* 48, 491-506.
- Hendizadeh, S.H., Hamidreza, F., 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. *Int. J. Production Economics* 111, 593-605.
- Javadia, B., Saidi-Mehrabadb, M., Hajic, A., Mahdavia, I., Jolaid, F., and Mahdavi-Amirie, N., 2008. No-wait flow shop scheduling using fuzzy multi-objective linear programming. *Journal of the Franklin Institute* , 345, 452–467
- Kise, H., Cheng, J., Matsumoto, H., 1997. A branch-and-bound algorithm with fuzzy inference for a permutation flowshop scheduling problem. *European Journal of Operational Research*, 96, 578-590.
- 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.
- Lin, S.W., Gupta, J.N.D., Ying, K.C., and Lee, Z.J., 2009. Using simulated annealing

- to schedule a flowshop manufacturing cell with sequence-dependent family setup times. *International Journal of Production Research*, **47**, 3205-3217.
- 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
- Mitrofanov, S.P., 1966. The scientific principles of group technology, National Landing Library translation, Yorkshire, Boston Spa, UK.
- Naderi, B., Zandieh, M., Khaleghi, A., Balagh, G., and Roshanaei, V., 2009. An improved simulated annealing for hybrid flowshops with sequence-dependent setup and transportation times to minimize total completion time and total tardiness. *Expert Systems with Applications*, **36**, 9625–9633
- Pinedo, M., 2002. Scheduling Theory, Algorithms, And Systems, Second Edition, *Prentice Hall*, 14-15.
- Scott, W. and Meral, A., 2001. Dynamic programming algorithms for scheduling parallel machines with family setup timesm. *Computers & Operations Research*, **28**, 127-137.
- 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 Operational Research*, **125**, 324-339.
- Webster, S., Azizoglu, M., 2001. Dynamic programming algorithms for scheduling parallel machines with family setup times machines with family setup times. *Computers & Operations Research*, **28**, 127-137.
- 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.

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，國立交通大學，碩士論文，2009。

湯璟聖，動態平衡機群排程的探討，中原大學，工業工程學系碩士論文，2003

馮正民、邱裕鈞，研究分析方法，1 版，新竹市：建都文化，民國 93 年。

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

排成，國立交通大學，工業工程與管理系碩士論文，2010

