

國立交通大學
工業工程與管理學系

碩士論文

以粒子群演算法求解流線型製造單元排程



**A Particle Swarm Optimization
Approach to Scheduling Flowshop
Manufacturing Cell**

研究生：陳威宇

指導教授：巫木誠 博士

中華民國一百年六月

以粒子群演算法求解流線型製造單元排程

A Particle Swarm Optimization Approach to Scheduling Flowshop Manufacturing Cell

研究生：陳威宇

Student：Wei-Yu Chen

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

以粒子群演算法求解流線型製造單元排程

研究生：陳威宇

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

粒子群演算法已被廣泛用於解決複雜的空間搜索問題。以前的研究大多專注於改進粒子群演算法的搜尋機制。本研究不同於傳統的研究方向，探討新的研究議題「採用新的解表達法是否可以提升粒子群演算法的績效」？本研究以流程式製造單元排程問題為背景，討論兩種粒子群演算法的績效差異。這兩種演算本質上是相同的演算流程，但卻是搭配截然不同的表達法(分別稱作 S_{old} 和 S_{new})。其中， S_{old} 是以前研究所用的表達法，而 S_{new} 是採用 Wu *et al.*, (2011)所提倡的新表達法。這兩種演算法分別稱作 $PSO-S_{old}$ 和 $PSO-S_{new}$ 。本研究採用大量的實驗數據證明，發現在小整備時間 (SSU)下兩種演算表現都不錯。但是，在中和大的整備時間 (MSU/LSU)情境下， $PSO-S_{new}$ 顯著優於 $PSO-S_{old}$ 。本研究實驗結果凸顯了新的研究方向--「針對不同的空間搜尋問題，可以探索新的表達法搭配啟發式演算法的績效」。

關鍵詞：粒子群演算法、解表達法、流程式製造單元

A Particle Swarm Optimization Approach to Scheduling Flowshop Manufacturing Cell

Student : Wei-Yu Chen

Advisor : Dr. Muh-Cherng Wu

Department of Industrial Engineering and Management

National Chiao Tung University

Abstract

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

誌謝

本論文可以順利完成，要感謝指導教授巫木誠博士對我的細心指導。還記得當初懵懂無知的我完全不知如何開始寫論文，幸好有老師的指導，使學生能有方向的去作研究。而在研究的過程中，時常碰到困難與挫折，老師總是耐心的指導，鼓勵我去學習並解決問題。尤其是遇到迷失研究方向的時候，老師總是能看透問題的本質，使我能夠了解問題之所在並解決問題。不僅如此，老師也十分重視誠信。教導學生無論做甚麼事都要誠實，一旦答應別人的事就一定要做到。只有誠信才能讓別人願意跟自己合作，也才能及時面對問題與解決問題。

在研究所的兩年裡，有李奕勳、曾偉杰、林耿漢、潘冠銘的陪伴，讓本來無聊的日子充滿歡笑。即使面對種種困難，002 實驗室的同學們也總是互相打氣，互相幫助。感謝振富學長即使很忙，也總是願意撥出時間來幫助我解決研究上的困難。同時也要感謝我的室友峰銘與盛中，當我需要幫忙時，他們總是無怨無悔地幫助我。還有一大群的朋友們與死黨，陪我一起度過這難忘的日子。希望在未來的日子，所有的朋友們無論在工作上、事業上、感情上都能一帆風順。

最後，要感謝的就是我的家人。無論遇到甚麼事，都默默地在背後支持我。讓我能夠專心的學習，順利完成學業。

陳威宇 於 新竹交大

目錄

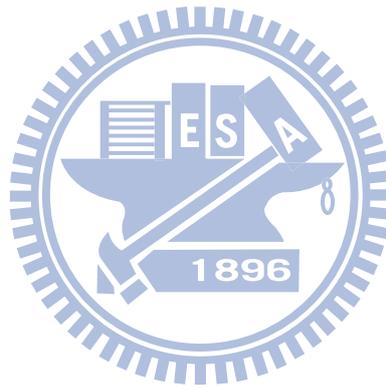
中文摘要.....	ii
Abstract.....	iii
誌謝.....	iv
表目錄.....	vii
圖目錄.....	viii
第一章 緒論	1
1.1 研究背景與動機.....	1
1.2 研究目的.....	2
1.3 研究議題與限制.....	2
1.4 論文架構介紹.....	3
第二章 文獻探討	4
2.1 流線式製造單元排程(Flowshop Manufacturing Cell Scheduling).....	4
2.2 順序相依家族整備時間(sequence dependent family setup times).....	5
2.3 固定序列(permutation).....	6
2.4 粒子群演算法(Particle Swarm Optimization).....	7
第三章 研究方法	10
3.1 研究問題.....	10
3.2 表達法設計與解讀.....	10
3.2.1 舊表達法(S_{old}).....	10
3.2.2 新表達法(S_{new}).....	11
3.3 粒子群演算法求解方法.....	11
3.3.1 新表達法結合粒子群演算法($PSO-S_{new}$)求解流程與步驟.....	11
3.3.2 舊表達法結合粒子群演算法($PSO-S_{old}$)求解流程與步驟.....	14
第四章 實驗情境與結果	15
4.1 測試情境.....	15
4.1.1 運算停止條件.....	16
4.1.2 相同的初始解.....	16
4.2 粒子群演算法參數設定.....	17
4.3 實驗結果與分析.....	18
4.3.1 Makespan 改善率比較結果.....	20
4.3.2 時間比較結果.....	20
4.3.3 Makespan 比較結果.....	21
4.3.4 實驗結果.....	23
4.4 統計檢定.....	24

第五章 結論與未來研究方向	26
5.1 研究結論.....	26
5.2 未來研究方向.....	27
參考文獻.....	28



表目錄

表 3.1 粒子的位置與速度.....	12
表 4.1 收斂代數統計結果.....	16
表 4.2 起始解統計結果.....	17
表 4.3 $PSO-S_{old}$ vs $PSO-S_{new}$ 30 個問題比較結果- Makespan	19
表 4.4 實驗結果.....	23
表 4.5 統計檢定結果.....	25



圖目錄

圖 2.1 流線式製造單元.....	4
圖 2.2 順序相依家族整備時間.....	5
圖 2.3 固定序列.....	6
圖 2.4 粒子群演算法流程圖.....	8
圖 3.1 舊表達法.....	10
圖 3.2 新表達法.....	11
圖 3.3 粒子位置對應到的加工順序.....	12
圖 3.4 表達法結合粒子群演算法流程圖.....	13
圖 3.5 舊表達法結合粒子群演算法空間產生示意圖.....	14
圖 4.1 演算法求解品質改善率.....	20
圖 4.2 演算法求解時間差距.....	21
圖 4.3 SSU 下 $PSO-S_{new}$ 優於 $PSO-S_{old}$ 的次數統計.....	22
圖 4.4 MSU 下 $PSO-S_{new}$ 優於 $PSO-S_{old}$ 的次數統計.....	22
圖 4.5 LSU 下 $PSO-S_{new}$ 優於 $PSO-S_{old}$ 的次數統計.....	23

第一章 緒論

1.1 研究背景與動機

針對產品的生產流程通常可以分成流線式製造(Flowshop Manufacturing)與零工式製造(Jobshop Manufacturing)。流線式製造主要是將產品依相同的生產順序製造，因此生產效率很高。零工式製造是將功能相同的機台放置在一起，組成各個部門(Department)，工件在各部門間進行不同的加工。由於功能相同的機台放在一起，所以零工式製造保有生產彈性的特色。為了保有流線式製造的效率與零工式製造的彈性，單元製造系統(Cellular Manufacturing System)逐漸被人們所重視。

單元製造系統是由許多製造單元(Manufacturing Cell)組成的群組技術。每一個製造單元是由許多製造相同特性工件的機台所組成。而單元製造系統主要的目的就是將相同特性工件所組成的工件族分配給各個製造單元加工，達到最大的經濟效益。其所面臨的問題就是決定工件族的加工順序(Among Family)與工件族裡工件的加工順序(Within Family)，由此可知單元製造系統相較於傳統的排程更加的複雜。隨著加工的工件越來越多，數學規劃求解已經很難在短時間內求的最佳解，取而代之的是啟發式演算法。像是模擬退火法(Simulated Annealing ; SA)、基因演算法(Genetic Algorithms ; GA)等啟發式演算法都被運用在求解複雜度很高的問題上。

在現實生產製造中，製造單元裡的整備時間通常都是順序相依整備時間(Sequence Dependent Setup Time)。也就是機台加工工件隨著工件族的不同，整備時間通常也跟著不同。為了更有效率的求解此類複雜的問題，將範圍限制在固定序列(Permutation)。固定序列最大的特色是當第一台機台決定了個工件族加工的順序後，後面的機台皆依所排定的工件族順序進行加工。

1.2 研究目的

本研究主要探討在考量順序相依整備時間與固定序列前提下，進行單元製造系統的排程規劃。根據 Lin *et al.*, (2009)提出舊的表達法為標竿，並以 Wu *et al.*, (2011)所提到新的表達法為比較對象。舊的表達法分成工件族加工順序與工件族內工件的加工順序兩個部分。先確定機台上工件族加工順序，再按照工件族加工順序依次填入工件族內工件的加工順序。新的表達法則是將舊的兩部分整合在一起，將各工件所屬的工件族資訊隱含在工件的加工順序裡。並運用粒子群演算法在有限的時間內，求得近似最佳解。目的希望運用粒子群演算法，針對順序相依整備時間與固定序列的流線式製造單元問題，將舊與新的表達法進行比較。目的如下：

- 新表達法是否比舊表達法更好。
- 探討造成此結果的原因。



1.3 研究議題與限制

本研究在探討如何將擁有相同加工屬性工件所組成的工件族，分配到各個製造單元加工的單元製造排程問題。單元製造排程問題範圍很廣，本研究只討論純粹流線式製造單元(Pure Flowline Manufacturing Cell)。純粹流線式製造單元定義工件族裡的工件其加工順序在每個機台上皆一樣。在實務上，當機台加工不同工件族時，最需要注意的就是機台的整備時間。如果加工的工件族不停的換，整備時間也會跟著增加，無形中造成完工時間變長。這類順序相依整備時間(SDFSTs)問題若沒有良好的控管，不僅僅讓昂貴的機台沒有辦法充分利用外，還會造成遲交罰款與信譽受損等嚴重後果。面對不同工件族的加工，機台的整備時間越少越好。為了提升加工速度，同工件族的工件會優先加工。在求解上，為了縮短求解時間，採用粒子群演算法(Particle Swarm Optimization)並只考慮固定序列排程(Permutation Scheduling)。目前以最小化最大完工時間為評估標準(Makespan)，尋

找最佳的工件族加工順序與工件族內工件的加工順序。本研究的限制主要如下:

- 工件的數量與其所屬的工件族已知。
- 同一個工件族內的工件不能被分割處理。
- 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。
- 每部機台工件族的加工順序皆相同。

1.4 論文架構介紹

本論文內容架構分成五章。第一章緒論，說明研究背景、動機、目的，以及研究議題與限制。第二章文獻探討，針對單元製造系統排程問題與粒子群演算法相關文獻加以整理，並與本研究進行參考和比較。第三章研究方法，運用粒子群演算法與表達法的結合，求解單元製造排程問題。第四章實驗情境與結果，分析舊與新表達法在粒子群演算法下的結果。第五章說明研究結論與未來研究的方向。



第二章 文獻探討

2.1 流線式製造單元排程(Flowshop Manufacturing Cell Scheduling)

單元製造系統(Cellular Manufacturing System)先將擁有相似製造特性的工件分成不同的工件族,接著再將某些工件族都需要經由相同機台加工的機台群組在一起形成製造單元(Manufacturing Cell)。製造單元主要又可以分成流線式與零工式兩類,而當進入某個製造單元的所有工件族都需要依機台順序加工,則稱為流線式製造單元。由圖 2.1 可以看出,工件族 F1 需要按順序經由每一台機台加工。

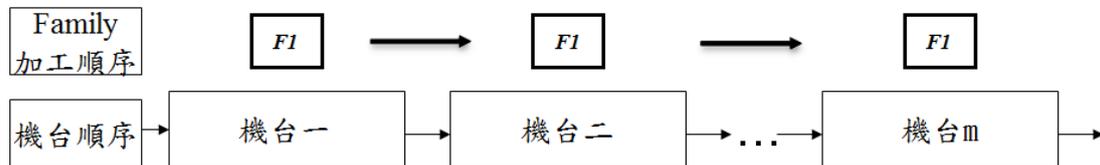


圖 2.1 流線式製造單元

當同一工件族加工時,由於工件特性相同,所以機台所需整備時間很少或不需要,直到下一個工件族來到,才需要較長的整備時間。所以只要能有效地減少工件族交換的時間,就可以提升機台的利用率。

Franc *et al.*, (2005) 考量順序相依機台整備時間並運用基因(GA)與滲集(Memetic)兩種演算法求解。發現在目標為最小化最大完工時間,可以有效地找到固定序列排程的最佳解。

Parthasarathy & Rajendran, (1998) 在只考慮固定序列排程情形,對於流線式製造單元進行研究,發現面對隨機插件的擾動下,模擬退火法(SA)在最小化遲交工件數與最小化權重遲交工件數都有良好的績效。

Lin *et al.*, (2009) 針對順序相依整備時間下流線式製造單元進的問題探討,發現在幾個不同的衡量標準下,以模擬退火法為基本的啟發式演算法都有很高的績效。此時的退火機制由柯西函數取代波茲曼函數。

2.2 順序相依家族整備時間(sequence dependent family setup times)

面對製造單元的問題時，有效控制整備時間的能力是提升生產效率的主要因素之一。家族整備時間是指機台加工不同工件族的工件時，需要換模具或零件所需要的時間。當機台因為工件族轉換不同時，其整備時間也會跟著不同。由圖 2.2 所示，機台加工工件族 1 後要加工工件族 3 需要整備時間 S_{13} ，同樣的機台加工工件族 2 後要加工工件族 3 需要整備時間 S_{23} ，當 S_{13} 與 S_{23} 不同時就是順序相依家族整備時間。換言之，當整備時間不會因為工件族換的不同而相異時，就是順序獨立家族整備時間。

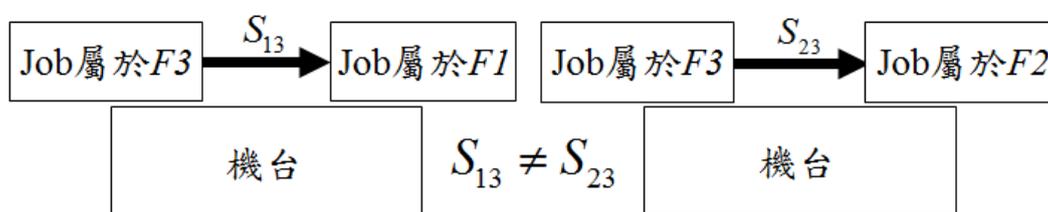


圖 2.2 順序相依家族整備時間

Naderi *et al.*, (2010)以總完工時間最小化為目標，針對順序相依家族整備時間的零工式問題研究。由歷史資料可以看出模擬退火法績效好壞取決於其操作過程與參數設定，尤其是鄰近搜尋架構的部分。作者藉由田口法找出不同作業過程與參數組合，並結合模擬退火法，提出了有效的鄰近搜尋架構。實驗結果證明以 Taillard 衡量準則，此種新型模擬退火法表現績效優異。

Hendizadeh *et al.*, (2008)發展出以塔布演算法為基礎的變化型演算法，考慮順序相依整備時間的流線型製造單元，並以最小化最大完工時間為目標。此變化型演算法還接受了模擬退火法裡即使結果變差也接受的概念，大大增加了本身的求解能力。因為結果變差可能只是暫時的，在比較差結果的附近或許有更好的解。計算結果顯示此變化型塔布演算法在求解時間與績效皆比目前最好的啟發式演算法更佳。

2.3 固定序列(permutation)

製造單元除了考量整備時間之外，決定工件族的加工順序也十分重要。當工件族在每個機台加工的順序都一樣時，就稱作固定序列。如圖 2.3 所示，當機台一加工工件族順序為 $F_2 \rightarrow F_1 \rightarrow F_3$ 之後，機台二與機台三皆以此工件族順序加工。換言之，當機台上加工工件族的順序會變化時，就是非固定序列(Non-permutation)。由上述可知，當機台以固定序列來排程時，求解的複雜度會降低，運算時間也會減少。

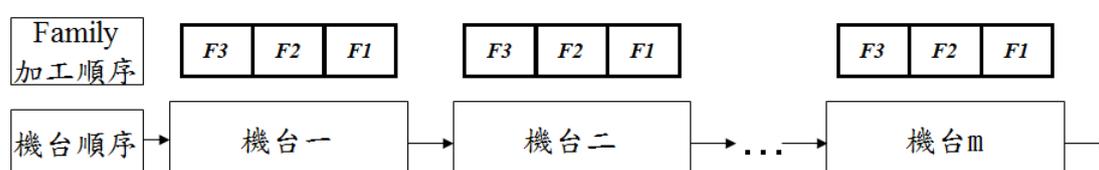


圖 2.3 固定序列

Bouabda *et al.*, (2010) 發展出基因演算法和分支界線法結合的方法，針對順序相依流線型製造單元問題，目標為最小化最大完工時間。利用分支界線法將目前的問題分解成幾個子問題，再用基因演算法提升目前解的品質。根據實驗結果，此方法在演算法演化的部分與區域收尋的過程都表現的不錯。

Ying *et al.*, (2010) 考慮順序相依流線型製造單元問題，運用模擬退火法探討固定序列與非固定序列所造成的影響。結果說明不管是在完工時間 (completion-time) 或到期日 (due-date) 的績效指標下，非固定序列都比固定序列都要來的好。此研究建議，如果計算時間許可，盡量採用非固定序列進行排程。

Schaller *et al.*, (2000) 考慮順序相依流線型製造單元問題，下界法能有效的找到固定序列排程，從而獲得最佳的最大完工時間與啟發式演算法。

2.4 粒子群演算法(Particle Swarm Optimization)

為了解決數學規劃無法求解或很難求解大規模情境的問題，人們開始運用起巨集啟發式演算法。粒子群演算法 (PSO) 就是屬於巨集啟發式演算法的一種，由 James Kennedy 和 Russell Eberhart 在 1995 年時所提出。這個演算法最初是藉由觀察真實世界中鳥類覓食的行為研究所得啟發的。觀察者發現大自然中有很多鳥在尋找食物，當有鳥兒找到食物後，附近的鳥會朝找到食物的地方靠近。在粒子群演算法中，空間中每個粒子就像是每隻鳥。每個粒子都代表一組可行解，而這些粒子會朝著有最佳解的粒子靠近。每個粒子的位置以 x_{id} 表示，速度以 v_{id} 表示，其中 i 代表第 i 個粒子， d 代表的是粒子所在空間的維度。由於各粒子都是可行解，所以經過運算後粒子都會有一個適應值 (Fitness value)。每個粒子都可以找到一個的最佳適應值 $pbest_i$ 及最佳位置 p_{id} 。 p_{id} 稱為粒子目前最佳值之位置。隨著世代交替，同時也會得到全體中最佳值 $gbest$ 及最佳位置 p_{gd} 。藉由粒子本身經驗以及群體經驗不斷修正，使所有粒子逐漸接近最佳解。

PSO 演算步驟如下所示(Eberhart and Shi,2001)：

- (1) 在 d 維空間中隨機產生 n 個粒子，隨機產生粒子的初始位置 x_{id} 及速度 v_{id} ，由這些粒子所組成粒子群體。此時為初始世代 $t=0$ 。
- (2) 計算每個粒子 i 於 d 維空間中之目標適應值 (Fitness)。
- (3) 將每個粒子之適應值與個體最佳值 $pbest_i$ 做比較，若目前現在粒子 i 的適應值優於 $pbest_i$ ，則 i 粒子的適應值取代 $pbest_i$ ，粒子之位置取代 p_{id} 。
- (4) 各粒子之適應值與 $pbest_i$ 比較後，再與目前群體最佳值 $gbest$ 做比較，若粒子之適應值優於 $gbest$ ，則粒子之適應值取代 $gbest$ ，粒子之位置取代 p_{gd} 。
- (5) 每個粒子依據更新位置與速度之公式，尋找出下個世代之新的位置及新的速度。
- (6) 若滿足終止條件即停止，否則回到步驟 2， $t = t+1$ ，通常終止條件為達到最佳解，或達到自行設定之世代數。

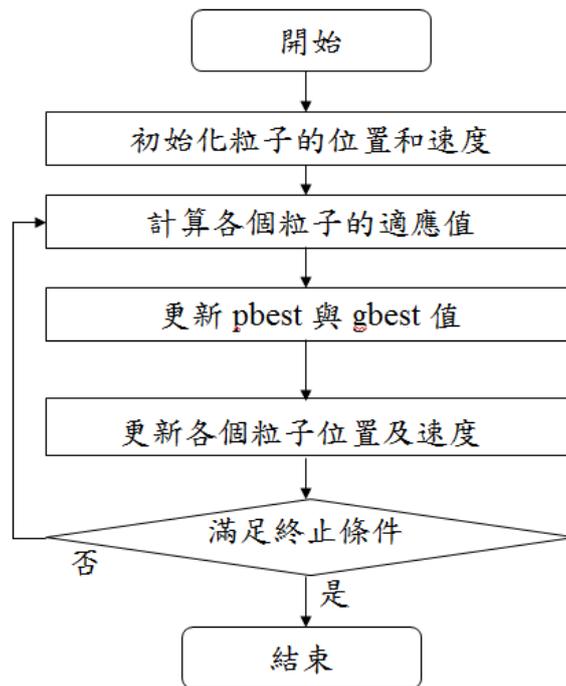


圖 2.4 粒子群演算法流程圖

Liao *et al.*, (2007)運用粒子群演算法離散機制的延伸，處理流線式排程問題。在此提出的演算法將粒子與速度重新定義，可以更有效率的將粒子轉換成新的排序。並將此新型粒子群演算法與傳統基因演算法比較，計算結果說明求解能力較佳。此外，又提出區域收尋與粒子群演算法的結合，稱作 PSO-LS，計算顯示在總流程時間績效下結果良好，但卻需要較長的運算時間。

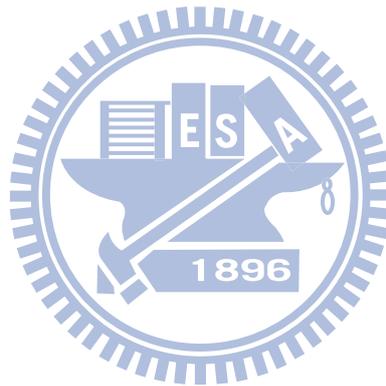
Tasgetiren *et al.*, (2007)探討固定序列流線式排程問題，運用粒子群演算法求解，目標是最小化最大完工時間與總流程時間。此篇文章是第一個運用粒子群演算法求解此類問題，提出利用最小位置值(smallest position value; SPV)將連續的粒子群演算法應用於排序問題。文中也提到藉由有效的區域搜尋法 VNS(variable neighborhood search)與粒子群演算法結合，提升求解品質與績效。

Zhang *et al.*, (2009)提出兩階段交替粒子群演算法(Alternate two phases particle swarm optimization algorithm; ATPPSO)去求解流線式排程問題，目標是最小化最大完工時間。它包含吸引與排斥兩個過程，並且擇一進行。為了抑制過早收斂的缺點，在排斥過程中每個粒子將飛往那些有更佳解的區域。本文也提出以

矩陣為基礎的最大完工時間計算方式，能提高演算法的運算速度。結果顯示 ATPPSO 在解的品質與收斂速度都優於其他的演算法，而且可以有效用來求解大情境的流線式排程問題。

Liu *et al.*, (2007)提出以粒子群演算法為基礎的 Memetic 演算法，求解固定序列流程式排程問題。針對此(NP)hard 問題，希望能達到最小化最大完工時間。PSOMA 運用 PSO 的演化機制，藉由個體的進步與群體的合作與競爭，能有效地探索解空間。

Kuo *et al.*, (2009)發展出新的混合式粒子演算法模型(HPSO)，結合隨機密鑰編碼與個別強化計畫，搭配粒子群演算法求解流線式排程問題。其目的是找出最佳的加工序，使 Makespan 最小。實驗結果表明，該 HPSO 方案遠遠優於過往以基因演算法為主的方案。



第三章 研究方法

本章探討如何運用Wu *et al.*, (2011)所提到的新表達法，結合本研究所建構的粒子群演算法，來求解製造單元排程問題。本章主要分成表達法設計與解讀和粒子群演算法求解方法兩個部分。

3.1 研究問題

本研究探討在粒子群演算法下，新表達法(S_{new})是否優於舊表達法(S_{old})。

3.2 表達法設計與解讀

本節分成舊(S_{old})與新(S_{new})表達法兩部分，並說明各自設計理念與解讀方法。

3.2.1 舊表達法(S_{old})

設計理念:由圖3.1可以看出舊表達法分成1+Family個部分。最前面的部分是設計用來表達機台上加工工件族的順序。後面的部分則是分別表達每個工件族內工件的加工順序。



圖 3.1 舊表達法

解讀方法:先解讀第一部分，得到工件族加工順序為 $F3 \rightarrow F2 \rightarrow F1$ 。接著解讀F3的工件加工順序是 $J5 \rightarrow J6$ ，F2的是 $J4 \rightarrow J3$ ，F1的是 $J2 \rightarrow J1$ 。所以經由舊表達法最後得到 $J5 \rightarrow J6 \rightarrow J2 \rightarrow J1 \rightarrow J4 \rightarrow J3$ 的加工順序。

3.2.2 新表達法(S_{new})

設計理念:由圖3.2可以看出新表達法只有一個部分，但是卻包含了工件所屬工件族的資訊。

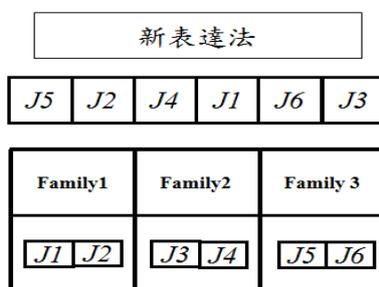


圖 3.2 新表達法

解讀方法:新表達法解讀分成兩個部分，先決定工件族的加工順序，再決定工件族內工件的加工順序。首先解讀到J5屬於F3，接著解讀到J2屬於F2，最後是J4屬於F1。再將工件依得到的工件族加工序解讀，由F3得到J5→J6，F2得到J2→J1，由F1得到J4→J3。最後再將其串起來得到J5→J6→J2→J1→J4→J3。

3.3 粒子群演算法求解方法

3.3.1 新表達法結合粒子群演算法($PSO-S_{new}$)求解流程與步驟

Step1:設定粒子的相關參數:包括最小位置 x_{min} 與最大位置 x_{max} ，最小速率 v_{min} ，最大速率 v_{max} ， r_1 與 r_2 皆為服從均勻分配0~1之間的亂數，加速係數 c_1 與 c_2 。

Step2:藉由公式(1)與(2)隨機產生 k 個具有 d 維空間的粒子，初始位置以 x_{id}^0 及速度 v_{id}^0 表示。粒子的數目 k 通常是所需加工工件數目的兩倍， d 維空間取決於工件數目。由這些粒子所組成的群體，稱為粒子群體。此時為初始世代 $t=0$ 。

$$x_{id}^0 = x_{min} + (x_{max} - x_{min}) * r_1 \quad (1)$$

$$v_{id}^0 = v_{min} + (v_{max} - v_{min}) * r_2 \quad (2)$$

由公式(1)與(2)可以求出各粒子的位置與速度，如表 3.1。 x_{id} 代表第 i 個粒子在第 d 個維度的位置， v_{id} 代表第 i 個粒子在第 d 個維度的速度。

表 3.1 粒子的位置與速度

Dimension	1	2	3	4	5	6
x_{id}^t	1.8	0.9	3.1	1.7	0.2	2.6
v_{id}^t	3.7	2.2	3.1	-0.4	-0.2	3.5

Step3:將各粒子的位置轉換成工件的加工順序。由表 3.1 可以得到最小 x_{id} 是 0.2，其所對應的維度是 5，就將其解讀成工件 5 優先加工。第二小的 x_{id} 是 0.9，其所對應的維度是 2，所以第二個加工的是工件 2。以此類推如圖 3.3，可以得到加工順序為 $J5 \rightarrow J2 \rightarrow J4 \rightarrow J1 \rightarrow J6 \rightarrow J3$ 。又因為新型表達法的設計，可以得到 $J1$ 與 $J2$ 屬於 Family1， $J3$ 與 $J4$ 屬於 Family2， $J5$ 與 $J6$ 屬於 Family3。並由新型表達法的解讀可以得到 $J5 \rightarrow J6 \rightarrow J2 \rightarrow J1 \rightarrow J4 \rightarrow J3$ 的最終加工順序。

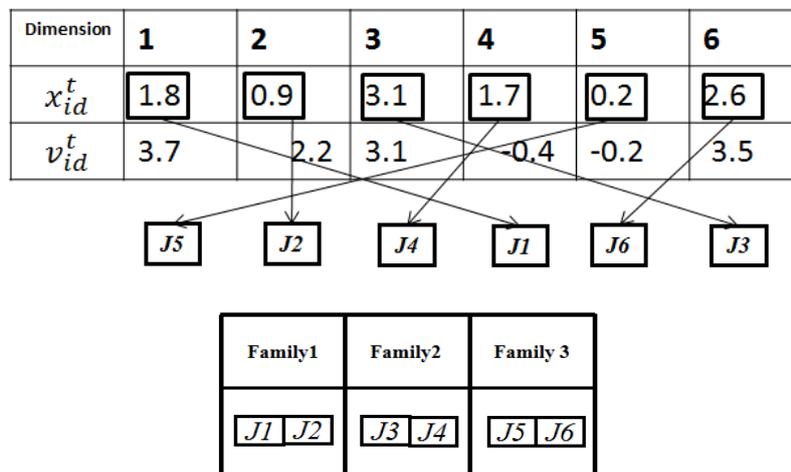


圖 3.3 粒子位置對應到的加工順序

Step4:藉由上一步驟可以得到最終加工順序，求出每個粒子之適應值。在世代 $t=0$ 時，粒子 i 的適應值儲存為 $pbest_i$ ，代表粒子 i 目前的最佳適應值。同時，也將粒子 i 所在的位置儲存為 p_{id} ，代表粒子 i 目前的最佳位置。當下一世代時，若目前的粒子適應值優於 $pbest_i$ ，則粒子之適應值取代 $pbest_i$ ，粒子目前的位置取代

p_{id} 。

Step5: 在世代 $t=0$ 時，從所有 k 個粒子選出最佳的 $pbest_i$ 儲存為 $gbest$ ，代表所有粒子目前的最佳適應值。同時，也將此粒子所在的位置儲存為 p_{gd} ，代表所有粒子目前的最佳位置。等下一世代時，一樣從所有 k 個粒子選出最佳的 $pbest_i$ ，再與目前群體最佳值 $gbest$ 做比較，若粒子的 $pbest_i$ 適應值優於 $gbest$ ，則粒子之適應值取代 $gbest$ ，粒子之位置取代 p_{gd} 。

Step6: 由公式(3)與(4)將每個粒子的位置與速度更新，尋找出下個世代新的位置及新的速度。

$$v_{id}^t = v_{id}^{t-1} + c_1 r_1 (p_{id}^{t-1} - x_{id}^{t-1}) + c_2 r_2 (p_{gd}^{t-1} - x_{id}^{t-1}) \quad (3)$$

$$x_{id}^t = x_{id}^{t-1} + v_{id}^t \quad (4)$$

Step7: 若演算世代數 t 達到要求即停止，否則回到步驟 2， $t = t + 1$ 。

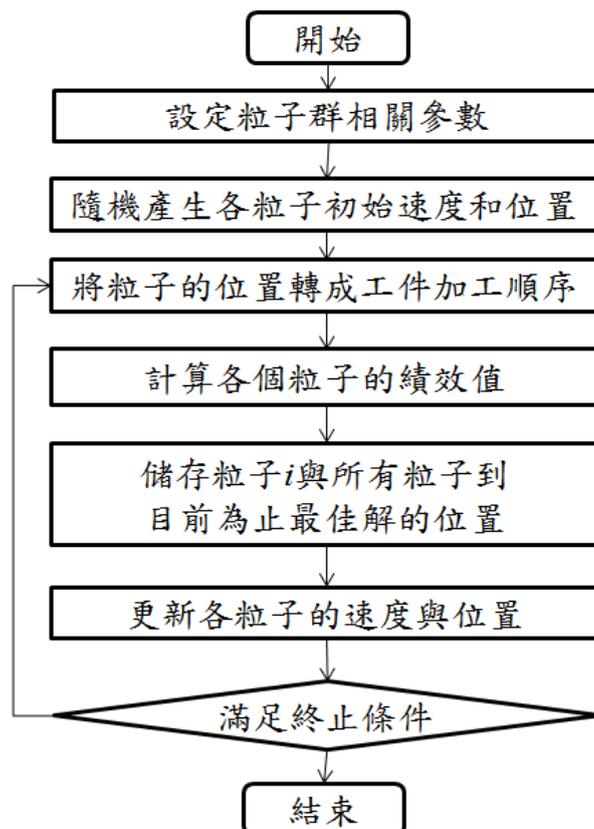


圖 3.4 表達法結合粒子群演算法流程圖

3.3.2 舊表達法結合粒子群演算法($PSO-S_{old}$)求解流程與步驟

舊表達法結合粒子群演算法($PSO-S_{old}$)其求解方法與 $PSO-S_{new}$ 幾乎一樣，除了在Step3需要創造1+Family個空間，從而得到機台上加工工件族的順序與每個工件族內工件的加工順序。

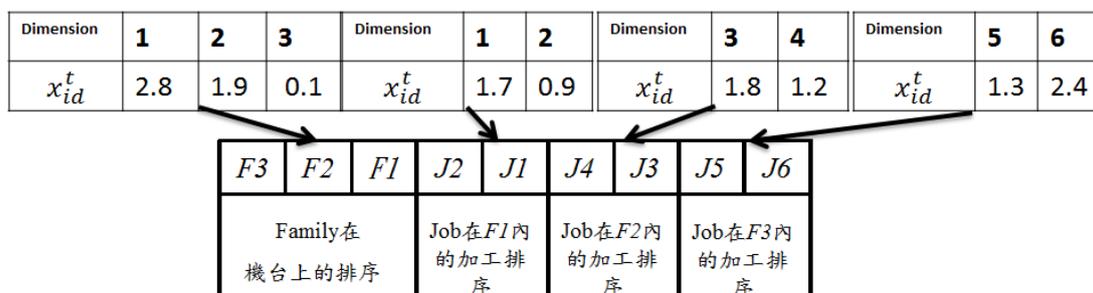


圖 3.5 舊表達法結合粒子群演算法空間產生示意圖

由圖3.5所示，此時粒子群演算法需要創造A、B、C、D四個空間。A空間可以求得機台上工件族的加工順序，B、C、D三個空間則是要分別求個工件族裡工件的加工順序。尤其是到了C空間，雖然也是兩個維度，但為了解讀需求必須要轉成3、4。D空間也是一樣，維度的解讀必須要從之前工件族所有的工件數開始。還需要注意的是由4個空間所組成的一個加工順序，若遇到更好的適應值時4個空間的最佳位置要一起更新。

第四章 實驗情境與結果

本研究用C++程式語言撰寫，其他作業環境與系統如下所示：

- 編譯器(Compiler)：MSVC9.0
- 編寫環境(IDE)：Visual Studio 2008
- 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

本研究測試情境設定參考Schaller *etal.*, (2000)如下：

所有實驗分成30個實驗情境，包含3種整備時間與10種由不同工件族數和機台數所構成的組合。3種整備時間為Small setups(SSU)：U[1, 20]、Medium setups(MSU)：U[1, 50]、Large setups(LSU)：U[1, 100]，分別為工件處理時間U[1, 10]的2、5、10倍。而每種整備時間類別下有10種組合(Family數，機台數)： $\{(3, 3), (3, 4), (4, 4), (5, 5), (5, 6), (6, 6), (8, 8), (10, 8), (10, 10)\}$ 。在每個工件族裡，工件的數目服從U[2, 10]。

在此30個實驗情境下，每個實驗情境又包含30個實驗問題。其原因是每個實驗情境都有無窮個可能，對於不同的工件數與不同的加工時間無法一一討論，於是隨機選取30個實驗問題進行探討。

每個實驗問題又包含15個Seed，每個Seed代表不同的起始解。對於Meta-Heuristic，不同的起始點會得到不同的結果。為了減少因為不同起始解所造成不同結果的影響，所以需要15次實驗取其平均。

在實驗之前，需先做兩個前測實驗。一個是終止代數的測試，其目的是希望能找到演算終止的世代數。另一個實驗是測試同初始解，其目的是想了解對於同Seed產生的初始解是否會影響實驗的結果。

4.1.1 運算停止條件

為了測試不同終止條件對Makespan的影響，以最佳解維持三百萬代不變與最佳解維持三百五十萬代不變兩種不同的終止條件進行實驗。由表4.1可以看出在舊表達法下，平均改善率為0.00%，統計結果為沒有顯著差異。說明當終止條件設為最佳解維持三百萬代時，舊表達法已經不會再進化了。在新表達法下，雖然統計結果顯示顯著贏，表示新表達法仍然在進化，但是平均改善率只有0.03%，非常的小。所以本實驗將終止條件設為最佳解維持三百萬代不變。

表 4.1 收斂代數統計結果

收斂代數				
表達法	平均改善率(%)	t_0	$t_{a,n-1}$	統計結果
舊表達法	0.00	1.37	2.05	沒有顯著
新表達法	0.03	3.12	2.05	顯著贏

4.1.2 相同的初始解

在過往的研究，大多專注於改進粒子群演算法的搜尋機制，所以同一seed即代表相同初始解。而本研究探討新的研究議題--「採用新的解表達法是否可以提升粒子群演算法的績效」。由於新的表達法，造成同一seed在不同的表達法下會有不同的初始解。為了正確比較 $PSO-S_{old}$ 與 $PSO-S_{new}$ 的績效，本研究將 $PSO-S_{old}$ 的初始解轉成 $PSO-S_{new}$ 的解讀型式，讓 $PSO-S_{new}$ 得到的初始解會與 $PSO-S_{old}$ 的初始解相同。由表4.2可以看出，無論是在新表達法或舊表達法下，同Seed與同初始解皆沒有顯著差異。理論上兩種表達法要進行比較，必須要有同樣的初始解下，才能比較其優劣。此結果說明，同Seed與同初始解其結果差異不大。

表 4.2 起始解統計結果

起始解				
表達法	平均改善率(%)	t_0	$t_{\alpha, n-1}$	統計結果
舊表達法	0.23	1.50	2.05	沒有顯著
新表達法	0.04	0.00	2.05	沒有顯著

4.2 粒子群演算法參數設定

1. 初始粒子個數:兩倍的工件數
2. $x_{min}=0, x_{max}=4$
3. $v_{min}=-4, v_{max}=4$
4. $r_1 \in U[0, 1], r_2 \in U[0, 1]$
5. 粒子加速度: $C1=C2=2$
6. 速度限制在-4~4之間
7. 初始位置在0~4之間
8. 運算停止條件:最佳解維持3百萬代不變



4.3 實驗結果與分析

實驗結果如表 4.2 所示，粒子群演算法在舊與新表達法比較的結果。表格內容說明如下：

1. $SSU(F+M)$ 、 $MSU(F+M)$ 、 $LSU(F+M)$ ：分別表示實驗的三種不同大小之整備時間情境。 F 為工件族數量； M 為機台數量。
2. N ：每種情境下的實驗問題數目。
3. N_e ：兩種演算法改善率平手的數目。
4. N_w ：本研究提出之 $PSO-S_{new}$ 改善率贏過 $PSO-S_{old}$ 的數目。
5. Ave ：每種情境下，30 個實驗問題的平均改善率。
6. C_n ：每種情境下， $PSO-S_{new}$ 在 30 個實驗問題的平均 makespan。
7. C_t ：每種情境下， $PSO-S_{old}$ 在 30 個實驗問題的平均 makespan。
7. T_t ：每種情境下， $PSO-S_{old}$ 在 30 個實驗問題的平均求解時間。
8. T_n ：每種情境下， $PSO-S_{new}$ 在 30 個實驗問題的平均求解時間。
9. Gap ：表示求解時間之差距 $Gap = T_t - T_n$

表 4.3 $PSO-S_{old}$ vs $PSO-S_{new}$ 30 個問題比較結果- Makespan

	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	26	4	134.47	134.50	0.03	43.32	24.85	-18.46
2	SSU34	30	25	15	10	151.33	151.43	0.09	53.57	28.27	-25.30
3	SSU44	30	25	2	23	186.29	186.63	0.20	74.53	36.03	-38.50
4	SSU55	30	16	4	12	244.53	244.52	0.03	115.14	49.27	-65.87
5	SSU56	30	16	2	14	256.63	256.43	-0.06	121.26	51.58	-69.69
6	SSU65	30	13	1	12	289.54	289.37	-0.04	145.57	60.21	-85.36
7	SSU66	30	14	1	13	299.25	298.78	-0.12	156.62	63.05	-93.57
8	SSU88	30	2	0	2	413.88	410.39	-0.83	247.20	93.96	-153.24
9	SSU108	30	0	0	0	500.23	492.86	-1.49	346.55	122.31	-224.24
10	SSU1010	30	0	0	0	544.14	535.45	-1.61	377.77	141.52	-236.25
11	MSU33	30	30	22	8	160.02	160.06	0.03	39.57	23.21	-16.35
12	MSU34	30	28	16	12	184.77	184.83	0.04	46.37	26.15	-20.23
13	MSU44	30	28	8	20	237.79	238.16	0.15	66.33	35.09	-31.23
14	MSU55	30	25	2	23	307.62	308.70	0.36	108.56	48.64	-59.92
15	MSU56	30	23	2	21	319.78	320.67	0.29	108.59	49.74	-58.86
16	MSU65	30	26	0	26	365.07	367.90	0.79	148.24	57.78	-90.46
17	MSU66	30	23	0	23	384.02	385.84	0.47	150.36	62.16	-88.20
18	MSU88	30	26	0	26	521.75	525.42	0.72	246.85	94.80	-152.05
19	MSU108	30	16	0	16	643.90	644.54	0.11	359.09	121.63	-237.46
20	MSU1010	30	9	0	9	677.97	675.14	-0.41	367.79	129.69	-238.10
21	LSU33	30	30	27	3	228.03	228.04	0.01	46.40	13.92	-32.48
22	LSU34	30	29	16	13	239.01	239.08	0.03	42.83	13.15	-29.67
23	LSU44	30	28	7	21	323.52	324.08	0.18	62.36	17.96	-44.40
24	LSU55	30	28	0	28	417.07	420.15	0.75	103.89	25.22	-78.67
25	LSU56	30	26	0	26	439.02	442.09	0.72	114.49	27.33	-87.16
26	LSU65	30	30	0	30	493.49	501.41	1.59	141.50	31.04	-110.47
27	LSU66	30	29	0	29	516.50	524.87	1.60	144.52	31.70	-112.82
28	LSU88	30	30	0	30	711.48	725.01	1.87	246.92	47.69	-199.24
29	LSU108	30	29	0	29	868.73	881.69	1.46	341.70	61.64	-280.06
30	LSU1010	30	26	0	26	931.87	941.53	1.03	370.01	71.08	-298.93
Average of all Scenarios		30	22.00	5.03	16.97	399.72	401.32	0.27	164.60	55.36	-109.24

4.3.1 Makespan 改善率比較結果

以改善率 $r = (obj_{PSO-S_{old}} - obj_{PSO-S_{new}}) / obj_{PSO-S_{new}} \times 100\%$ 來評估求解品質。實驗結果分析如下：

由圖4.1所示，在小的整備時間下，平均改善率隨著工件族與機台數增加而逐漸降低。在大的整備時間下，平均改善率卻隨著工件族與機台數增加逐漸提高。結果說明當整備時間越大，機台數與工件族數越多， $PSO-S_{new}$ 會比 $PSO-S_{old}$ 越好。

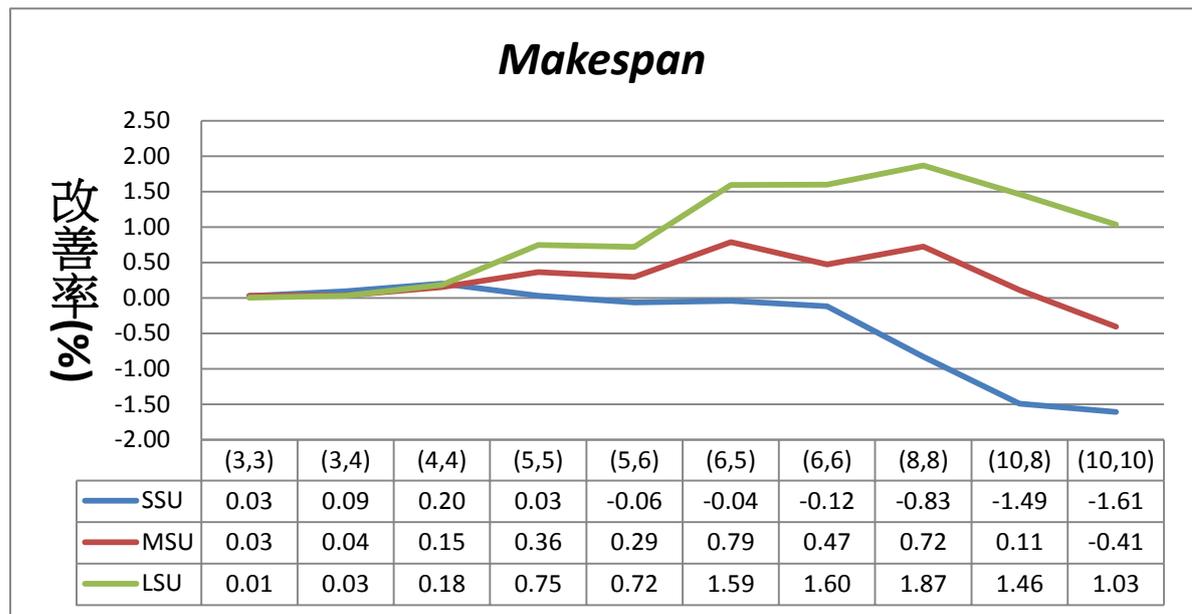


圖 4.1 演算法求解品質改善率

4.3.2 時間比較結果

由圖4.2可以看出，新表達法在任何情境下求解效率都比舊的表達法差。但是，運算時間差距最大的也不超過300秒，表示求解時間差距是很短的。所以只要新表達法的求解品質較好，仍然可以接受新表達法的概念。

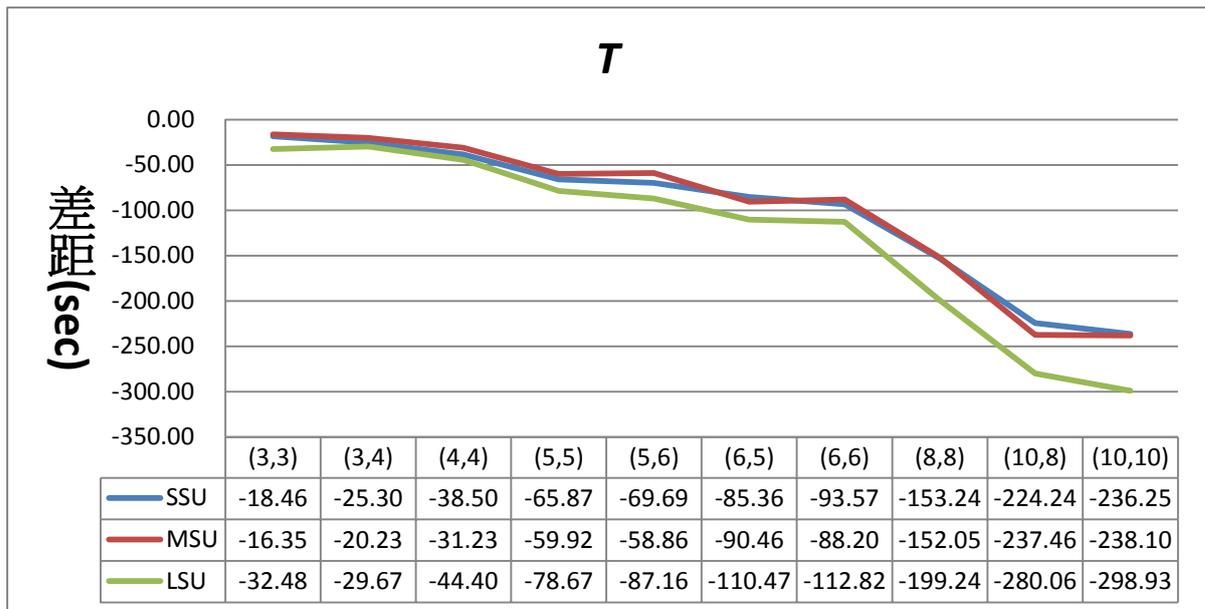


圖 4.2 演算法求解時間差距

4.3.3 Makespan 比較結果

由圖 4.3、圖 4.4 與圖 4.5 可以看出在總比較次數 30 次下，SSU33、MSU33 與 LSU33 新舊表達法的績效皆相似。說明新表達法在工件族與機台數小時解的品質只優於舊表達法一些，大多與舊表達法一樣。另一方面，在 SSU 情境下，新表達法贏舊表達法的次數隨著工件族數目與機台數目增加下降。反觀在大的整備時間(LSU)裡，新表達法優於舊表達法的次數隨著工件族與機台數增加越來越好。說明在大的整備時間下，當工件族與機台數越多時，新表達法獲得的解品質會越顯著優於舊表達法。

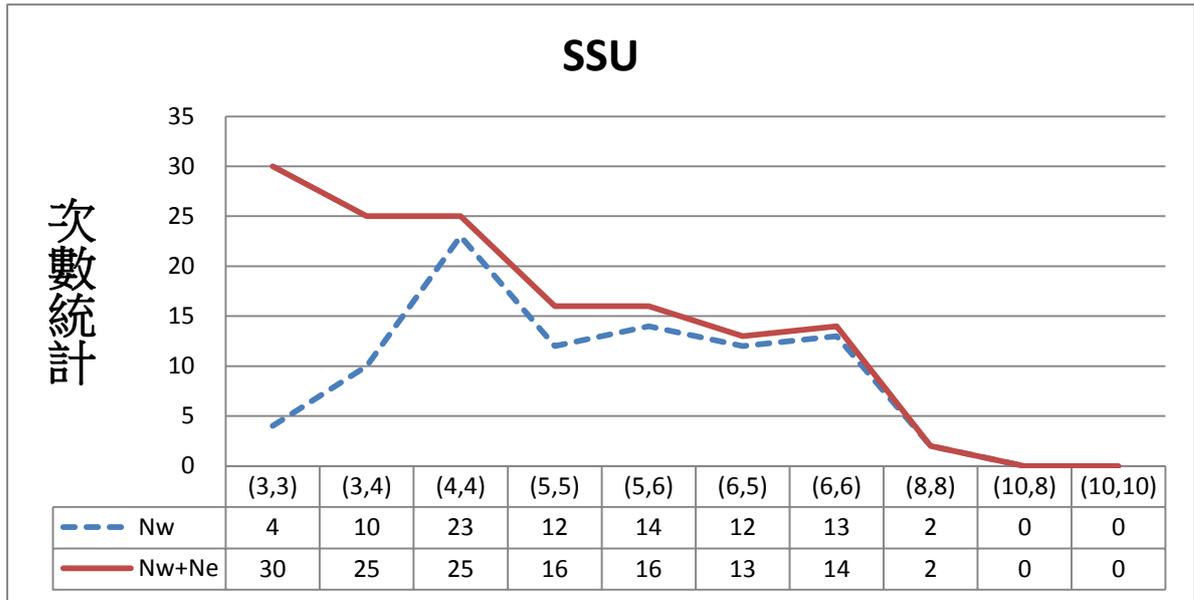


圖 4.3 SSU 下 $PSO-S_{new}$ 優於 $PSO-S_{old}$ 的次數統計

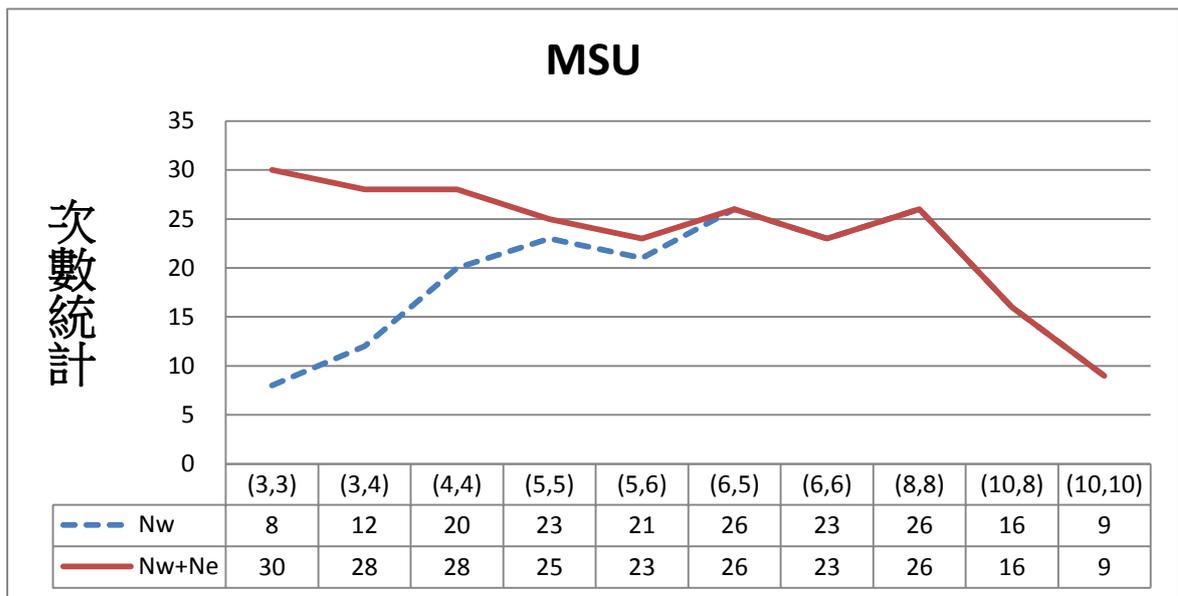


圖 4.4 MSU 下 $PSO-S_{new}$ 優於 $PSO-S_{old}$ 的次數統計

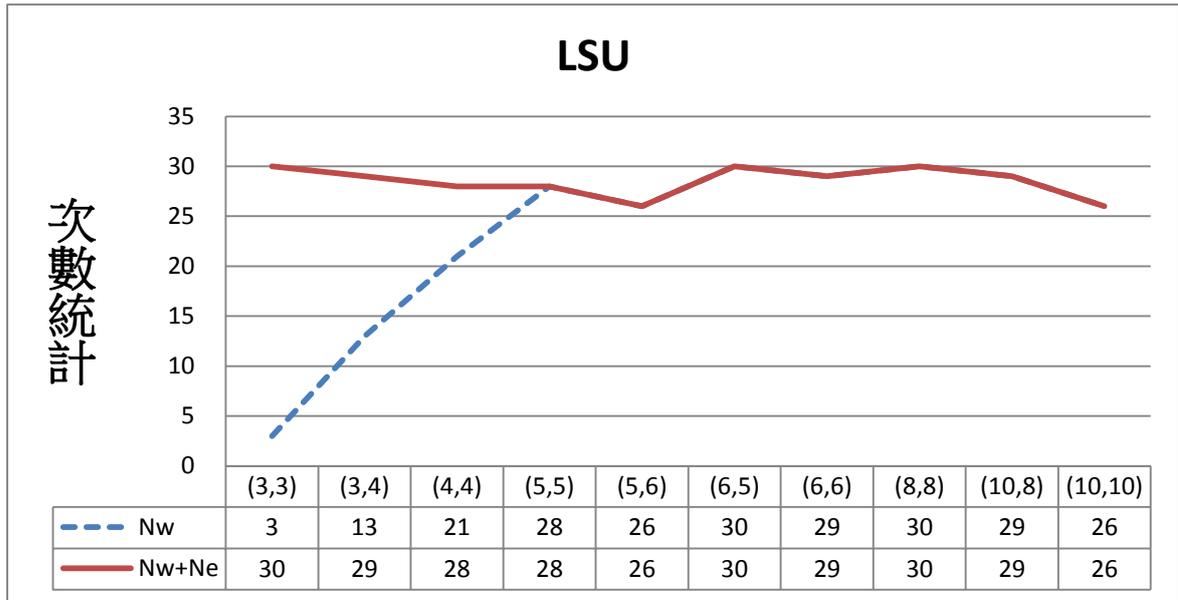


圖 4.5 LSU 下 $PSO-S_{new}$ 優於 $PSO-S_{old}$ 的次數統計

4.3.4 實驗結果

由表 4.4 可以知道在 SSU 情境下平均改善率為-0.38%，MSU 與 LSU 分別是 0.26% 與 0.92%，此結果說明隨著整備時間的增加 $PSO-S_{new}$ 會比 $PSO-S_{old}$ 越來越好。由贏的次數(N_w)觀察，SSU、MSU 與 LSU 分別是 9、18 與 24，也同樣說明隨著整備時間的增加 $PSO-S_{new}$ 會比 $PSO-S_{old}$ 越來越好。總平均改善率為 0.27% 可以說明本研究所提倡的新型表達法在粒子群演算法下，優於舊的表達法。新表達法結合粒子群演算法($PSO-S_{new}$)在 30 個情境下平均求解時間 165 秒，雖然求解時間輸給舊表達法($PSO-S_{old}$)55 秒，但運算時間仍然可以接受。

表 4.4 實驗結果

Scenario	makespan							Computation Time(sec)		
	N	N_w+Ne	N_e	N_w	C_n	C_t	Ave. (%)	T_n	T_t	Gap
Average of SSU Scenarios	30	14	5	9	302	300	-0.38	168	67	-101
Average of MSU Scenarios	30	23	5	18	380	381	0.26	164	65	-99
Average of LSU Scenarios	30	29	5	24	517	523	0.92	161	34	-127
Average of all Scenarios	30	22	5	17	400	401	0.27	165	55	-109

4.4 統計檢定

本研究利用 Paired Sample t-test 來檢定新舊表達法在 PSO 上，是否有顯著的差異。檢定分成 SSU、MSU、LSU 三種整備時間情境下與整體四個部分。令 $S : \{SSU33, SSU34, SSU44, SSU55, SSU56, SSU65, SSU66, SSU88, SSU1010, MSU33, MSU34, \dots, LSU33, LSU34, \dots, LSU1010\}$ 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 個實驗問題新、舊解表達法的解的平均(每個實驗問題有 15 個 seed，由 15 個 seed 所得到解的平均)，其中 $s \in S, i = 1, 2, \dots, 30$ (每個情境有 30 個實驗問題)。在情境 s 下的第 i 個實驗問題，將其樣本資料正規化得到 $d_{s,i} = (u_{old,s,i} - u_{new,s,i}) / u_{old,s,i} * 100\%$ 。

三種整備時間情境與整體的樣本平均：

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

三種整備時間情境與整體的樣本標準差：

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

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

三種整備時間情境與整體檢定用 t 值：

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

表 4.5 統計檢定結果

Scenario	平均改善率	t_0	t 值	統計結果
LSU	0.92	15.58	1.96	顯著贏
MSU	0.25	6.27	1.96	顯著贏
SSU	-0.39	8.50	1.96	顯著輸
Total	0.26	7.87	1.96	顯著贏

由表4.5可以得知，在 $\alpha=0.05$ 的顯著水準下，新表達法在SSU情境顯著輸給舊表達法。但是在MSU、LSU與整體情境下，新表達法皆顯著優於舊表達法。



第五章 結論與未來研究方向

5.1 研究結論

在過往的研究中，大多是研究如何改進粒子群演算法的搜尋機制。本研究探討採用新表達法($PSO-S_{new}$)是否優於舊表達法($PSO-S_{old}$)。

實驗方法是運用相同的粒子群演算法流程，將新表達法($PSO-S_{new}$)與舊表達法($PSO-S_{old}$)進行比較。實驗分成30個實驗情境，包含3種整備時間(SSU、MSU、LSU)與10種組合(Family數，機台數):{(3, 3), (3, 4), (4, 4), (5, 5), (5, 6), (6, 6), (8, 8), (10, 8), (10, 10)}。每個實驗情境又針對不同的工件數與不同的加工時間探討30個實驗問題。為了減少因為不同起始解所造成不同結果的影響，每個實驗問題又包含15個Seed並取其平均。實驗的初始解在新舊表達法上皆為相同的起始解，而其終止條件為最佳解維持三百萬代不變。

由實驗結果顯示，新表達法結合粒子群演算法($PSO-S_{new}$)在makespan績效指標下，對於大的整備時間，求解品質會隨著工件族數目與機台數目增加而提高。雖然 $PSO-S_{new}$ 求解速度較慢，但是求解時間皆不超過400秒，運算時間很短，重要的是求解品質顯著優於 $PSO-S_{old}$ 。由統計結果得知，在SSU情境下，新表達法($PSO-S_{new}$)顯著輸給舊表達法($PSO-S_{old}$)。在MSU與LSU情境下，新表達法($PSO-S_{new}$)顯著贏舊表達法($PSO-S_{old}$)。在總情境下，新表達法($PSO-S_{new}$)也顯著贏舊表達法($PSO-S_{old}$)。所以，在以最小化最大完工時間的目標下，新表達法在PSO演算法中比舊表達法得到更佳的生产排程組合。

本研究貢獻在於凸顯了新的研究方向--「針對不同的空間搜尋問題，可以探索新的表達法搭配啟發式演算法的績效」。

5.2 未來研究方向

從實驗結果可以得到，在 SSU 情境下新表達法無法獲得優於舊表達法，但是在 MSU、LSU 情境下新表達法卻顯著優於舊表達法。可以嘗試藉由分析粒子群演算法的特性，去分析在不同表達法下整備時間所造成的影響。本研究目前只針對 Makespan 比較，還沒探討不同績效指標下表達法差異會造成的影響，所以未來也可以多加入其他績效指標進行考量。由文獻得知，在過往研究中發現，非固定序列(non-permutation)會比固定序列(permutation)得到的結果更佳。本研究只專注於固定序列(permutation)，故未來可以探討在非固定序列(non-permutation)裡新表達法與舊表達法的求解績效。也可以深入探討非固定序列是否比固定序列好，以及其原因。

同時，可以比較此表達法在其他 Meta-heuristic 上的績效，並分析在不同演算法下所造成的影響與原因。除了針對流程式製造單元問題外，本研究所提出表達法的概念也可以嘗試應用到其他的排程問題。



參考文獻

英文文獻

- Boudbda, R., Jarboui, B., Eddaly, M., and Rebai, A., 2010. 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.
- Eberhart, R.C., and Shi, Y., 2001. Particle Swarm Optimization: Developments, Application and Resources. *Proceedings of the 2001 Congress on Evolutionary Computation*, 1, 81–86.
- Franca, P.M., Gupta, J.N.D., Mendes, A. S., 2005. Evolutionary algorithms for scheduling a flowshop manufacturing cell with sequence dependent family setups. *Computers and Industrial Engineering* 48 (3), 491–506.
- Hendizadeh, S.H., Faramarzi, H., and Mansouric, S.A., Gupta J.N.D., ElMekkawy T.Y., 2008. Meta-heuristics for scheduling a flowline manufacturing cell with sequence dependent family setup times. *International Journal of Production Economics* 111 (2), 593–605.
- Kuo, I.H., Horng, S.J., Kao, T.W., Lin, T.L., Lee, C.L., Terano, T., and Pan, Y., 2009. An efficient flow-shop scheduling algorithm based on a hybrid particle swarm optimization model. *Expert Systems with Applications* 36, 7027-7032.
- Liao, C.J., Tseng, C.T., and Luarn, P., 2007. A discrete version of particle swarm optimization for flowshop scheduling problems. *Computers & Operations Research* 34, 3099-3111.

- 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.
- Liu, B., Wang, L., and Jin, Y.H., 2007. An Effective PSO-Basd MEmetic Algorithm for Flow Shop Scheduling. *IEEE Transactions on Systems, Man, and Cybernetics* 37(1),18-27.
- Naderi, B., Fatemi Ghomi, S.M.T., and Aminnayeri, M., 2010. A high performing metaheuristic for job shop scheduling with sequence-dependent setup times. *Applied Soft computing* 10, 703–710.
- 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 (2), 324–339.
- Parthasarathy, S., Rajendran, C., 1998. Scheduling to minimize mean tardiness and weighted mean tardiness in flowshop and flowline-based manufacturing cell. *Computers ind.Engng* 34(2), 531-546.
- Tasgetiren, M.F., Liang, Y.C., and Sevkli, M. Gencyilmaz, G., 2007. A particle swarm optimization algorithm for makespan and total flowtime minimization in the permutation flowshop sequencing problem. *European Journal of Operational Research*, 177 ,1930–1947.
- Ying, K.C., Gupta, J.N.D., Lin, S.W., and Lee, Z.J., 2009. Permutation and nonpermutation schedules for the flowline manufacturing cell with sequence dependent family setups. *International Journal of Production Research* 48 (8), 2169- 2184.

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, C.S., Sun, J.G., 2009. An alternate two phases particle swarm optimization algorithm for flowshop scheduling problem. *Expert systems with Application* 36, 5162- 5167.

