

# 模擬退火法於考量整備與拆卸時間之具非等效平行機台多階段流程型排程問題研究

## An Approach of Simulated Annealing in Multistage Flow Shop Scheduling with Unrelated Parallel Machines

葉玉玲<sup>1</sup> Yukling Yip      駱景堯<sup>2</sup> Chinyao Low      洪正鴻<sup>2</sup> Cheng Horn Horng  
大葉大學電算中心                      大葉大學工業工程與科技管理學系

<sup>1</sup>Computer Center, Da-Yeh University and <sup>2</sup>Department of Industrial Engineering and Technology Management, Da-Yeh University

(Received October 13, 2003; Final Version November 28, 2004)

**摘要：**本研究針對具非等效平行機台之多階段流程型的工廠排程問題，加入獨立且可分割整備時間及相依拆卸時間作為考量的因素；在總流程時間最小化之目標下，將工件安排至各階段的平行機台上加工，並決定其工件之加工順序。研究首先以0-1整數規劃建構出探討問題之最佳解模式，但有鑒於數學模式求解耗時且僅適用於小規模問題之求解，故針對較大規模問題，發展出一以模擬退火法為基礎之啟發式演算法，用以求解於中、大規模問題，以期可以在容許時間內搜尋到最佳解或一近似最佳解。研究中除對於影響演算法之參數進行分析與設定，以求演算法之穩定性；並對起始解對於演算法之最終求解結果之影響進行分析與比較，期能以此結果提供後續研究之參考。

**關鍵詞：**非等效平行機台、多階段流程型、整備時間、相依拆卸時間、模擬退火法

**Abstract :** In this research, an unrelated parallel machine flow shop scheduling problem of minimizing mean flow time is considered. In the past few years, in order to simplify the scheduling problem, most of research assumes setup and removal times are even negligible or part of the processing time. In this research, we look the setup, processing, and removal times as separable, then take the

sequence-independent setup and dependent removal times into account. To solve the addressed problem two different solving models are developed. First, a 0-1 integer programming model is constructed; however, the mathematical model is too time consuming to solve the medium or large size problem, thus, a simulated annealing based heuristic is proposed to get an near optimal schedule in a reasonable computation time. During the research, the parameters used in the heuristics that affect the solution quality and efficiency are analyzed and designed; then for the constructed heuristic, a good parameter setting is suggested. The experimental results are reported, and provided for the references for the further research.

**Keywords** : unrelated parallel machine, flow shop, setup, removal, simulated annealing.

## 1. 緒論

生產排程 (production scheduling) 是工廠在成本控制上的重要角色，是一門有效配置可利用資源，以滿足特定績效衡量準則的一種技術，排程問題在學術領域上的應用已相當廣泛和成熟。隨著科技的進步，產業的變遷，現今工廠的生產型態已經日趨複雜，近幾年來電子和半導體產業 (semiconductor industry) 蓬勃發展，不但在台灣的產業界中扮演著重要角色更成為台灣的經濟命脈。在電子和半導體廠中，各製程的生產設備大多十分昂貴，其設備的單位產能所創造出的利潤也相當可觀，在電子和半導體產業中生產型態有多數是屬於多階段平行機台流程式的型態 (flow shop with multiple processors; FSMP)，因此如何有效的將設備作分配，以提高生產量、降低成本，成為電子和半導體產業界中重要的課題。現將本研究考慮之問題描述如下：考量一由 $N$ 個不同工件，與 $M$ 個加工中心 ( $k = 1, 2, 3, \dots, M$ ) 所組成之生產系統。系統中每一加工中心 $k$ 具有  $m_k$  台非等效機台。每一工件 $i$  ( $i = 1, 2, 3, \dots, N$ ) 具有 $M$ 項作業 ( $j = 1, 2, 3, \dots, M$ )，系統中每一工件 $i$ 需經過每一個加工中心一次且只有一次，另每一工件 $i$ 經過每一加工中心之順序為已知且皆相同。每一工件 $i$ 於加工中心 $j$ 中之任一機台 $k$ 上有一已知之作業時間  $P_{ijk}$ ，且機台 $k$ 在處理任一工件 $i$ 時亦具有一已知之獨立整備時間  $S_{ijk}$  與作業相依移除時間  $R_{ijk}$ 。任一工件 $i$ 不得在不同機台上同時處理，且不得提早釋出 (不得分割)。至於在整備時間之考量上，本研究採用所謂之內部整備 (internal setup) - 機台靜止之狀態下方能進行整備作業；但機台在閒置時即可進行整備工作，以縮短工件之流程時間 (楊文華，民85)。本研究在上述排程問題規模之描述下，尋求一能滿足最小化總流程時間之最佳 (近似最佳) 解。

一般而言，平行機台之種類依其加工效率的差異，大致可分為三類：(1) 完全相同的機台 (identical machine) - 同一工件在不同的平行機台作加工時，其加工時間都完全相同，(2) 等效機

台 (uniform machine) - 同一工件在不同的平行機台作加工時，其加工時間會呈現比例關係，(3) 非等效平行機台 (unrelated parallel machine) - 同一工件在不同的平行機台上作加工，其加工時間不同且可能不呈比例關係。回顧過去相關文獻，有關多階段平行機台排程的相關研究上面，在完全相同的平行機台 (identical machine) 上有 Azizoglu and Kirca (1998) 利用幹支界限法 (branch and bound) 求解以總延遲時間最小化為目標之平行機排程問題，Alidaee and Rosa (1997) 也針對解總延遲時間 (tardiness) 最小化之平行機排程問題提出一啓發式求解方法，Santos et. al. (1995) 考量多階段相同平行機台環境中，提出一全域下限法，用以求解最小化最大完工時間，Brah and Loo (1999)，考量多階段下相同平行機的環境，以最大完工時間為目標下，分別對五種方法進行比較。等效平行機台上的研究則有 Koulamas and Kyparisis (2000) 延伸 EDD (earliest due date) 派工法則，對目標值為最小化延遲時間進行求解。Dessouky and Dessouky (1998) 利用幹支界限法對於多階段流程式考慮  $n$  個完全相同的工件在等效平行機台的環境裡，目標值為最小化最大完工時間 (makespan) 來進行求解。至於在非等效平行機台的研究中，Suresh and Chaudhuri (1996) 對於單階段下考慮  $m$  台非等效平行機台，以最小化最大總完工時間和最小化最大總延遲時間為目標考量下對此排程問題提出一利用塔布搜尋 (tabu search) 為基礎的演算法則進行求解。在國內，近幾年來也有不少學者針對平行機台進行研究，張百棧 (民 91) 針對非等效平行機台，使用基因演算法，以最小化廢料成本和最大完工時間總和最小化為目標，發展一演算法和傳統派工方式進行比較。張毓仁 (民 89) 研究多階段不等效平行機排程問題針對所有工作在某些製程可選擇自製或外包機台加工的情況，在有交期時間的限制下，以最小化總成本為目標，提出啓發式演算法並建立數學規劃模式。

將整備時間納入具平行機台考量之流程式排程環境中則有：Allahverdi (2000) 針對可分割獨立整備時間考量之雙機流程式問題，在以平均總流程時間 (mean flow times) 最小化為目標下，提出一演算法則用以求解最佳排序。Allahverdi and Aldowaisan (2000) 再將其上述問題擴展為三機流程之型態並增加其問題限制，在平均流程時間最小化為目標下求解最佳排序。Franca et al. (1996) 則在考量相依整備時間之多階段且各階段具  $m$  台相同平行機台之排程環境中，利用塔布搜尋 (tabu search) 求解總完工時間最小化之排序。Kim et al. (2002) 則在考慮相依整備時間之非等效平行機台排程問題之上求解上以模擬退火法進行求解最佳化排序。Balakrishnan and Kanet (1999) 在考量相依工件整備時間之  $n$  工件， $m$  台等效平行機台之問題中，針對交期延遲成本最小化之為目標，建構出數學模式，並利用 Lingo 6.0 進行小問題的求解。國內近年來也有許多學者對於考慮整備時間之平行機台之排程環境進行相關之研究，如：田國興 (民 88) 針對具平行機台之多階段流程型工廠排程問題，提出啓發式方法迅速有效地求解，以期在不同工件數、工作站數、和機台負荷率下，得到較佳的總排程時間。

至於在過去的排程研究中，同時將整備與拆卸時間納入考量的研究，則大都以流程式排程問題考量為主，如：Yoshida and Hitomi (1979) 擴展傳統雙機排程問題模式，將整備與處理時間分

離，並且修正 Johnson's 演算法求得最佳解。Sule (1982) 延續上述問題將其擴展，將每項工件的處理加工時間分為獨立整備、拆卸相依時間及工件加工時間。Proust et. al. (1991) 探討總完工時間為最小化且整備時間及拆卸時間均為獨立之  $m$  機流程式問題，對小問題提出一分支界限法，另外對大型問題提出四個啟發式演算法。而駱景堯等 (民 92) 則考量獨立整備及相依拆卸時間之開放型工廠排程環境，針對延遲最小化之目標，提出一數學模式與啟發式求解法。

透過上述相關文獻之回顧，截至目前對於具平行機台考量之流程式排程問題的探討中，雖有許多學者提出研究看法，但是對於在多階段下考量非等效平行機台的研究還較少提出，本研究擴展其問題，考量獨立可分割的整備時間並且同時考量排序相依拆卸時間，使之接近實際環境；研究首先建構該問題之數學模式用於小規模問題的求解，隨後並提出一以模擬退火法為基之啟發式法進行中、大規模問題的求解。

## 2. 數學規劃求解模式建構

針對本研究將獨立整備時間與相依移除時間納入多階流程型排程問題考量之主題，本節透過 0-1 整數規劃建構數學規劃模式以求取問題之最佳解，由於模式僅能對小問題規模之排程問題求得最佳解，且求解甚為耗時，故此模式僅供驗證第三節針對中、大規模問題所建構啟發式求解演算結構之正確性與適用性。

### 2.1 符號說明

針對本研究所用之符號定義說明如下：

$i$ ：工件編號  $i=1,2,\dots,n$ 。

$j$ ：階段編號  $j=1,2,\dots,l$ 。

$k$ ：機台編號  $k=1,2,\dots,m_j$ 。

$S_{ijk}$ ：工件  $i$  在階段  $j$  中的機台  $k$  上所需之整備時間。

$P_{ijk}$ ：工件  $i$  在階段  $j$  中的機台  $k$  上所需之處理時間。

$H$ ：為一極大懲罰值。

$TS_{ij}$ ：工件  $i$  在階段  $j$  的整備開始時間。

$TS_{ijk}$ ：工件  $i$  在階段  $j$  的機台  $k$  上之整備開始時間。

$TP_{ij}$ ：工件  $i$  在階段  $j$  的處理開始時間。

$TP_{ijk}$ ：工件  $i$  在階段  $j$  的機台  $k$  上之處理開始時間。

$TR_{ij}$ ：工件  $i$  在階段  $j$  的拆卸開始時間。

$TR_{ijk}$ ：工件  $i$  在階段  $j$  的機台  $k$  上拆卸開始時間。

$F_i$ ：工件  $i$  的完工時間。

$X_{ijk}$  : 1, 工件*i*在階段*j*上的機台*k*作加工。  
0, 其它。

$U_{ii'jk}$  : 1, 同在在階段*j*之機台*k*上加工之工件, 工件*i*為工件*i'*之前項作業。  
0, 其它。

$Z_{ii'jk}$  : 1, 同在在階段*j*之機台*k*上加工之工件, 工件*i*為工件*i'*的前一項作業 (immediately preceding operation)。  
0, 其它。

$Y_{ii'jk}$  : 1, 工件*i*與工件*i'*都是在階段*j*中的機台*k*加工。  
0, 其它。

$E_{jk}$  : 所有在階段*j*機台*k*上加工的工件集合。

$O_{ij}$  : 工件*i*在階段*j*上。

$m_j$  : 階段*j*中平行機台的數量。

$A_{jk}$  : 所有可能在階段*j*中機台*k*上加工工件的集合。

$A_{ijk}$  : 階段*j*機台*k*上加工的最後一個工件。

$C_{jk}$  : 代表階段*j*中機台*k*上加工之所有工件之拆卸時間矩陣。

$R_{ii'jk}$  : 在階段*j*的機台*k*上, 工件*i*為工件*i'*的後一項作業時, 機台所需之拆卸時間。

拆卸時間矩陣 :

$C_{jk}$  :

From\To	J	J	J	J	J
	ob <sub>1</sub>	ob <sub>2</sub>	ob <sub>i</sub>	ob <sub>i'</sub>	ob <sub>fn</sub>
Job <sub>1</sub>	∞	R	F	R	R
Job <sub>2</sub>		1 <sub>2jk</sub>	1 <sub>ijk</sub>	1 <sub>i'jk</sub>	1 <sub>fnjk</sub>
.		∞			
.					
.					
Job <sub>i</sub>			∞	R	R
Job <sub>i'</sub>				∞	if <sub>fnjk</sub>
.			F		R
.			i <sub>ijk</sub>		i <sub>fnjk</sub>
.					
.					
Job <sub>i</sub>					R
					if <sub>fnjk</sub>

## 2.2 模式建構

針對本研究以總流程時間最小化之排程目標，其數學模式描述如下：

$$\text{Minimize } \sum_{i=1}^n F_i$$

subject to:

$$TR_{ij} + \sum_{k=1}^{m_j} \sum_{i \in E_{jk}} (R_{ii'jk} \times Z_{ii'jk}) = F_i \quad \forall i \quad (1)$$

$$TS_{ij} + \sum_{k=1}^{m_j} (X_{ijk} \times S_{ijk}) = TP_{ij} \quad \forall i, j, k = 1, 2, 3, \dots, m_j \quad (2)$$

$$TP_{ij} + \sum_{k=1}^{m_j} (X_{ijk} + P_{ijk}) = TR_{ij} \quad \forall i, j, k = 1, 2, 3, \dots, m_j \quad (3)$$

$$\sum_{k=1}^{m_j} X_{ijk} = 1 \quad \forall i, j, k = 1, 2, 3, \dots, m_j \quad (4)$$

$$TR_{ij} \leq TP_{i(j+1)} \quad \forall i, j = 1, 2, 3, \dots, l-1 \quad (5)$$

$$TS_{ijk} = TS_{ij} \times X_{ijk} \quad \forall i, j, k = 1, 2, 3, \dots, m_j \quad (6)$$

$$TP_{ijk} = TP_{ij} \times X_{ijk} \quad \forall i, j, k = 1, 2, 3, \dots, m_j \quad (7)$$

$$TR_{ijk} = TR_{ij} \times X_{ijk} \quad \forall i, j, k = 1, 2, 3, \dots, m_j \quad (8)$$

$$U_{ii'jk} + U_{i'ijk} = X_{ijk} \times X_{i'ijk} \quad \forall i, i' \in E_{jk}; \quad \forall j, k = 1, 2, 3, \dots, m_j \quad (9)$$

$$X_{ijk} = \sum_{i' \in A_{jk} \cup A_{jk}} Z_{ii'jk} \quad \forall i, j, k = 1, 2, 3, \dots, m_j \quad (10)$$

$$\sum_{i' \in A_{jk}} Z_{ii'jk} \leq 1 \quad \forall i, j, k = 1, 2, 3, \dots, m_j \quad (11)$$

$$\sum_{i \in E_{jk}} Z_{ii'jk} \leq 1 \quad \forall i, j, k = 1, 2, 3, \dots, m_j \quad (12)$$

$$U_{ii'jk} - Z_{ii'jk} \geq 0 \quad \forall i, i', \forall j, k = 1, 2, 3, \dots, m_j \quad (13)$$

$$U_{ii'jk} + Z_{ii'jk} \leq 1 \quad \forall i, i', \forall j, k = 1, 2, 3, \dots, m_j \quad (14)$$

$$TR_{ij} + R_{i'ijk} \times Z_{i'ijk} - H \times (1 - Z_{i'ijk}) \leq TS_{ij} \quad \forall i, i', \forall j, k = 1, 2, 3, \dots, m_j \quad (15)$$

(1)式定義每一工件的完工時間;每一工件在最後一階段的開始拆卸時間加上相依拆卸時間，等於此工件的完工時間。(2)式表示任一工件，進行處理作業時需已經完成整備作業。(3)式表示任一作業，當處理作業結束後立即進行拆卸作業。(4)式描述任一工作需要任一階段中的任一機台作加工乙次。(5)式因為本研究所考量的整備作業為可分割處理作業，所以當工件*i*在上一階段開始處理作業時，下一階段允許開始作整備。意即當上一階段的工件*i*處理作業完成便可立即在下一階段進行處理作業。(6)式用來決定出任一工件在任一階段中的任一機台*k*上的開始整備時間。(7)式用來決定出任一工件在任一階段中的任一機台*k*上的開始處理時間。(8)式用來決定出任一工件在任一階段中的任一機台*k*上的拆卸開始時間。(9)式決定任兩個可能在同一階段中同一機台上作加工，工件的先後關係。(10)式決定任一工件的後一項作業，當  $X_{ijk} = 1$ ；代表工件*i*在階段*j*中的機台*k*上作加工則工件*i*在階段*j*中機台*k*上必有一後項作業。(11)式說明任一*j*階段中在機台*k*上，加工之末項工件，最多只有一個或是沒有任何工件在此機台上加工。(12)式說明在任一階段*j*機台*k*加工的工件，工件之前一項加工工件最多只有一個，因此限制式是小於或等於1表示。(13)與(14)兩式用來定義同機台上加工之兩作業間之關係，以  $U_{i'jk}$  與  $Z_{i'jk}$  描述兩者之間之相互關係。 $U_{i'jk}$  與  $Z_{i'jk}$  之關係如表1所示。表1說明當工件*i*為工件*i'*之前項作業時， $U_{i'jk} = 1$ ，此時  $Z_{i'jk}$  可為0或1；但若工件*i*不為工件*i'*之前項作業時， $U_{i'jk} = 0$ ，此時  $Z_{i'jk}$  必定為0。因而可用  $U_{i'jk} - Z_{i'jk} \geq 0$  來描述之。

然而，(14)式並無法完全限制所有  $U_{i'jk}$  之值，僅能限制單向  $U_{i'jk}$  之值，因而  $U_{i'jk}$  需考量雙向之關係，將  $U_{i'jk}$  與  $Z_{i'jk}$  納入考量；其關係如表2所示。表2說明當工件*i*不為工件*i'*之前項作業時， $U_{i'jk} = 0$ ，此時  $Z_{i'jk}$  可為0或1；但若工件*i*為工件*i'*之前項作業時， $U_{i'jk} = 1$ ，此時  $Z_{i'jk}$  必定為0。因而可用  $U_{i'jk} + Z_{i'jk} \leq 1$  描述之。

表1  $U_{i'jk}$  與  $Z_{i'jk}$  之邏輯關係表

	Case1	Case2	Case3
$U_{i'jk}$	1	1	0
$Z_{i'jk}$	1	0	0

表2  $U_{i'jk}$  與  $Z_{i'jk}$  之邏輯關係表

	Case1	Case2	Case3
$U_{i'jk}$	0	0	1
$Z_{i'jk}$	0	1	0

(15)式決定任兩工件在同一加工中心之同一機台上作業之順序關係。若工件*i*為工件*i'*之前項作業時， $Z_{ii'jk} = 1$ ，工件*i'*之開始整備時間必須大於或等於其前項工件*i*於該加工中心之完工時間。反之， $Z_{ii'jk} = 0$ 則為一無效限制式。

### 2.3 線性化

由於上述所建構之數學模式中，限制式(6)、(7)、(8)、與(9)均為非線性限制式，可利用額外之變數設定將之改寫為線性限制式，以利數學模式之求解速度。

#### 限制式(6)、(7)、(8)之線性化

限制式(6)  $TS_{ijk} = TS_{ij} \times X_{ijk}$  可改寫為：

$$TS_{ij} = \sum_{k=1}^{m_j} TS_{ijk} \quad \forall i, j, k \in m_j \quad (6.1)$$

$$TS_{ijk} \geq 0 \quad \forall i, j, k \in m_j \quad (6.2)$$

$$TS_{ijk} \geq TS_{ij} - H \times (1 - X_{ijk}) \quad \forall i, j, k \in m_j \quad (6.3)$$

說明：

由於  $\sum_{k=1}^{m_j} X_{ijk} = 1$  且  $TS_{ijk} = TS_{ij} \times X_{ijk}$

$$\sum_{k=1}^{m_j} TS_{ijk} = TS_{ij} \times \sum_{k=1}^{m_j} X_{ijk} = TS_{ij} \quad (6.1) \text{ 且 } TS_{ijk} \geq 0 \quad (6.2)$$

若考慮變數*A*與*B*，*A*、*B*皆大於等於0且為正整數；若  $A \geq B$  且  $A \leq B$ ，則  $A = B$ 。應用上述之觀念，利用變數  $X_{ijk}$  決定  $TS_{ijk}$  的值，可將限制式描述為  $TS_{ijk} \geq TS_{ij} - H \times (1 - X_{ijk})$  (6.3)。因為  $X_{ijk}$  為0-1變數，若工件*i*在加工中心*j*之機台*k*上加工，則  $X_{ijk}$  為1；故(6.3)式  $TS_{ijk} \geq TS_{ij}$ ，但因  $TS_{ij} \leq TS_{ijk}$ ，所以  $TS_{ijk} = TS_{ij}$ 。反之，若工件*i*不在加工中心*j*之機台*k*上加工，則  $TS_{ijk} > -\infty$ ，因而  $TS_{ijk} = 0$ 。

同理，限制式(7)  $TP_{ijk} = TP_{ij} \times X_{ijk}$  可改寫為：

$$TP_{ij} = \sum_{k=1}^{m_j} TP_{ijk} \quad \forall i, j, k \in m_j \quad (7.1)$$

$$TP_{ijk} \geq 0 \quad \forall i, j, k \in m_j \quad (7.2)$$

$$TP_{ijk} \geq TP_{ij} - H \times (1 - X_{ijk}) \quad \forall i, j, k \in m_j \quad (7.3)$$

限制式(8)  $TR_{ijk} = TR_{ij} \times X_{ijk}$  可改寫為：



$$TR_j = \sum_{k=1}^{m_j} TR_{jk} \quad \forall i, j, k \in m_j \tag{8.1}$$

$$TR_{ijk} \geq 0 \quad \forall i, j, k \in m_j \tag{8.2}$$

$$TR_{ijk} \geq TR_j - H \times (1 - X_{ijk}) \quad \forall i, j, k \in m_j \tag{8.3}$$

限制式(9)之線性化

限制式(9)  $U_{i'jk} + U_{ijk} = X_{ijk} \times X_{i'jk}$  可藉由(9.1)至(9.3)式描述之：

$$Y_{i'jk} \leq 0.5 \times (X_{ijk} + X_{i'jk}) \tag{9.1}$$

$$0.5 \times (X_{ijk} + X_{i'jk}) \leq Y_{i'jk} + 0.5 \quad \forall i, i' \in E_{jk}, \forall j, k \in m_j$$

$$U_{i'jk} + U_{ijk} = Y_{i'jk} \quad \forall i, i' \in E_{jk}, \forall j, k \in m_j \tag{9.2}$$

$$Y_{i'jk} = Y_{ijk} \quad \forall i, i' \in E_{jk}, \forall j, k \in m_j \tag{9.3}$$

說明：

首先設定  $Y_{i'jk} \in \{0,1\}$ ，為建立  $X_{ijk}$ 、 $X_{i'jk}$  與  $Y_{i'jk}$  之關係，可利用  $Y_{i'jk}$  範圍的限制來取代  $X_{ijk} \times X_{i'jk}$ 。表3為  $X_{ijk}$ 、 $X_{i'jk}$  之四種組合情形，經由(9.1)式可證明  $Y_{i'jk}$  與  $X_{ijk} \times X_{i'jk}$  可獲得相同之結果。

Case 1:  $0.5 \leq Y_{i'jk} \leq 1$  ,  $Y_{i'jk} = 1$

Case 2:  $0.5 \leq Y_{i'jk} \leq 0.5$  ,  $Y_{i'jk} = 0$

Case 3:  $0 \leq Y_{i'jk} \leq 1$  ,  $Y_{i'jk} = 0$

Case 4:  $-0.5 \leq Y_{i'jk} \leq 0$  ,  $Y_{i'jk} = 0$

**表3 變數邏輯關係表**

	Case1	Case2	Case3	Case4
$X_{ijk}$	1	1	0	0
$X_{i'jk}$	1	0	1	0
$X_{ijk} \times X_{i'jk}$	1	0	0	0
$Y_{i'jk}$	1	0	0	0

其次，以(9.2)式取代 $U_{i'jk} + U_{ijk} = X_{ijk} \times X_{i'jk}$ ，因為 $Y_{i'jk} \in \{0,1\}$ ，故可以避免 $U_{i'jk} + U_{ijk} = 2$ 的情形發生。(9.3)式則說明該變數僅決定兩作業是否於同一機台上進行加工，並不考慮相互間加工順序之問題。

### 3. 啟發式求解模式建構

由於本研究所探討之主題，過去類似研究已將之歸類為NP-hard之問題 (Brah and Loo, 1999; Franca *et al.*, 1996; Kim *et al.*, 2002)，第2節所建構之最佳解數學模式僅適用於小規模問題的求解，且求解耗時，故本研究發展出一以模擬退火演算法為基礎的啟發式演算法，用於中、大規模的問題求解，以期可以在容許時間下求得最佳解或一近似最佳解，說明如下。

#### 3.1 模擬退火法(simulated annealing)

模擬退火演算法含有重複搜尋、計算目標成果之特性，並透過隨機過程在特定條件下接受目標函數較差的解，而跳脫陷入區域最佳的情況，使搜尋目標函數達到總體最小化。因此本研究利用模擬退火演算法求解所描述之開放型工廠排程問題，配合演算法邏輯，當溫度逐漸降低，排程結果將會逼近一近似最佳解。另者，模擬退火法雖有不錯之演算成效，但所需之執行時間略長 (Alidaee and Rosa, 1997; Azizoglu and Kirca, 1998)，針對此一缺點本研究亦另外建立停止準則，以兼顧求解之品質及時效。以下各節將針對初始解、移步、停止準則等演算法之結構進行設計與說明，隨後並對起始溫度、馬可夫鏈長度、降溫率等參數做深入探討，期望透過有效之設計，增加演算法的求解品質與執行效率。本研究所建構之模擬退火演算程序說明如下：

步驟 1. 起始狀態之設定

步驟 1.1 產生初始解 $s_0$ (見 3.2 節)，並令特定溫度 $T$ 下之最佳解 $s_T^* := s_0$ ，全域最佳解 $s_G^* := s_0$

步驟 1.2 演算法參數之設定 (起始溫度、馬可夫鏈長度、與降溫率)

步驟 2. 演算法程序是否終止之判斷 (見 3.4 節)

步驟 3. 特定溫度 $T$ 下之求解搜尋 (搜尋所預設之馬可夫鏈長度或滿足 3.4 節所設定之終止條件)

步驟 3.1 進行移步 (見 3.3 節) 以產生新解 $s_T^1$

步驟 3.2(a) 若兩目標值之差

$E = f(s_T^1) - f(s_T^0) < 0$ ，則進行特定溫度 $T$ 下求解狀態之更新，令該溫度 $T$ 下之最佳解值 $s_T^* := s_T^1$ ；並更新溫度 $T$ 下求解改善次數。如果 $f(s_T^*) - f(s_G^*) < 0$ ，則令全域最佳解 $s_G^* := s_T^*$ ，並更新全域最佳解改善次數。此處 $s_T^0$ 與 $s_T^1$ 分別為同一溫度下之連續兩移步解。

步驟 3.2(b) 若兩目標值之差

$E = f(s_T^1) - f(s_T^0) \geq 0$ ，則計算機率值 $U = \text{EXP}(-E/T)$ ，並隨機產生另一機率值 $P$ ；若 $P > U$ ，則令 $s_T^0 := s_T^1$ ，並重置 (reset) 溫度 $T$ 下求解改善次數。

步驟4. 進行降溫；重複執行步驟2至步驟4。

本研究選用模擬退火法作為所提出之演算法的基礎，演算法架構如圖1所示。

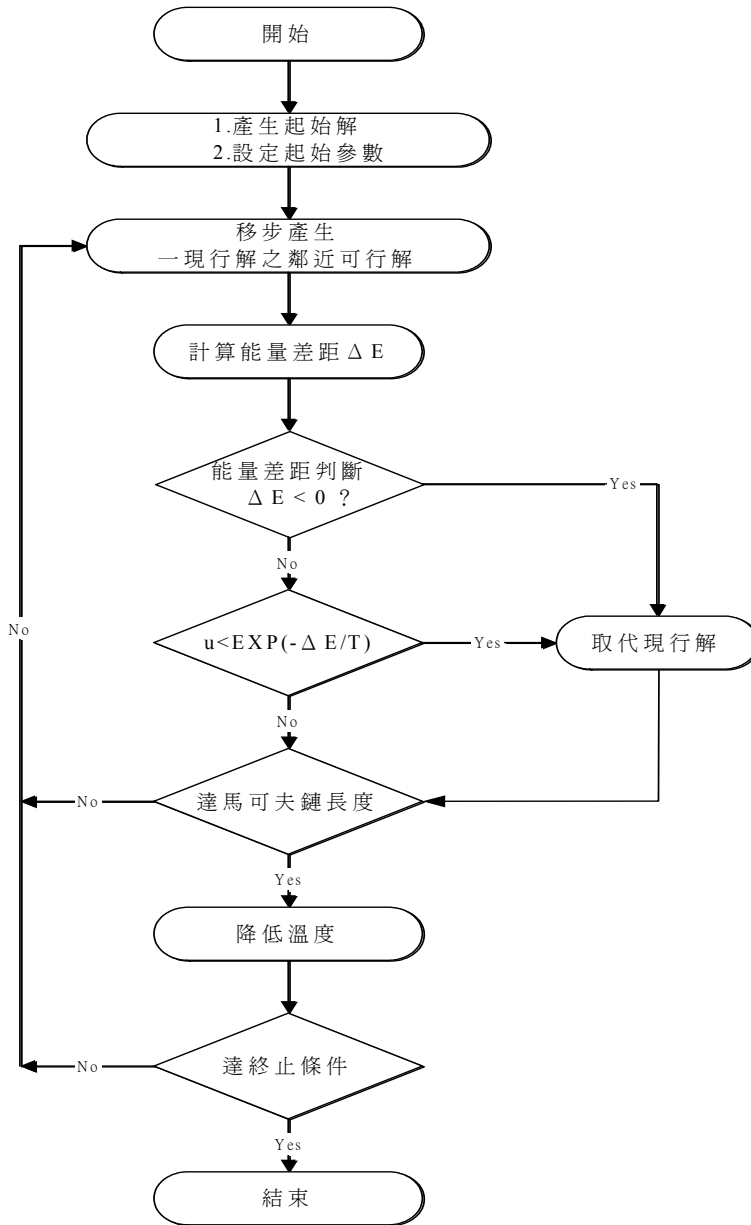


圖 1 啟發式演算法架構圖

### 3.2 起始解建構

過去已有多位學者(張百棧, 民91; Alidaee and Rosa, 1997; Brah and Loo, 1999; Koulamas and Kyriaris, 2000) 在研究結果中指出, 好的起始解, 對於整個後續收尋有很大的影響, 本研究考慮多階段非等效平行機流程型的排程問題, 同時考慮可分割的整備時間及相依拆卸時間, 目前的文獻中並無學者針對此問題提出一好的起始解, 因此本研究對於在流程型問題中, 以總流程時間最小化為目標考量之問題, 常被作為起始解選用的NEH法則及SPT、LPT法則, 修改NEH法則延伸出其五種法則 (H1、H2、H3、H4、H5), 及修改SPT (H6)、與LPT (H7) 共計七種法則, 以下僅對於演算法H4的演算步驟進行說明, 其餘六種法則之演算步驟, 請參見附錄。

H4 演算步驟:

- 步驟1. 計算出每個工件在每一階段的所處理時間, 獲得每一工件的總處理時間, 依照各工件的總處理時間大小, 由小到大依序對工件排列, 獲得一組合排序, 定義此排列組合為初始加工順序。
- 步驟2. 將所得的初始加工順序中, 挑選最前面兩個工件做互換, 可得到新的組合排序。
- 步驟3. 分別將所產生的組合排序, 進入各階段中進行加工處理。
  - 步驟3.1 工件進入各階段加工的順序為依照組合排序中工件之先後順序。
  - 步驟3.2 工件 $i$ 在階段 $j$ 之機台選取規則; 選用能較先完成工件 $i$ 在階段 $j$ 加工作業之機台。(工件在機台指派上若遇有相同的完工時間時, 以機台編號順位優先選取)。
  - 步驟3.3 計算所有可能組合排序之總流程時間, 並以總流程時間最小者, 作為下一迭次初始的加工順序。(組合排序遇有相同的最佳總流時間時, 以先產生出來的組合, 優先選取)
- 步驟4. 選取下一個尚未作過互換的工件, 並向前插入所有可能位置, 產生新的組合排序後重複步驟3直到所有工件都已作過互換。

### 3.3 移步

本研究考慮三種移步方法, 說明如下:

移步一:

僅針對第一階段, 隨機選取兩工件作互換, 往後之階段工件進入的順序則, 選取能較先進入下一階段的工件, 優先進行指派 (FIFO)。

移步二:

隨機產生一0、1變數 $z$ 值。

若 $z=0$ 則:

採用移步一。

若 $z=1$ 則:

隨機選取非第一階段以後之任一 $j$ 階段，作為改善階段，在此 $j$ 階段上，選取任兩機台考量：

- (1) 若此任兩機台非同機台，則任意選取此兩機台上加工的工件作交換，如圖2所示，往後的 $j+1$ 階段選擇能較先進入階段加工的工件，優先進行機台選派。
- (2) 若選取的兩機台是同一機台，則挑選此機台上有相鄰關係之兩工件作互換，如圖3所示，往後的 $j+1$ 階段選擇能較先進入階段加工的工件，優先進行機台選派。

移步三：

隨機選取任一階 $j$ 段作為改善階段，在此 $j$ 階段上，隨機選取任意兩工件作互換，往後的 $j+1$ 階段，則選取能較先進入下一階段的工件，先進行機台指派。

### 3.4 演算法停止準則

模擬退火演算法所使用的停止準則是用來終止退火改善過程，求得問題最終解。一般使用的停止準則為進行以下三種測試：

- (1) 所考慮的迭代次數是否達到，若達到則終止演算法。
- (2) 最終溫度是否已經降低到預設值以下，已達成則終止。

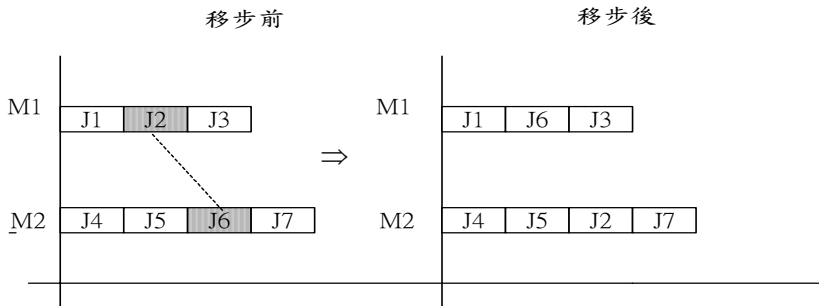


圖2 機台不同時移步示意圖

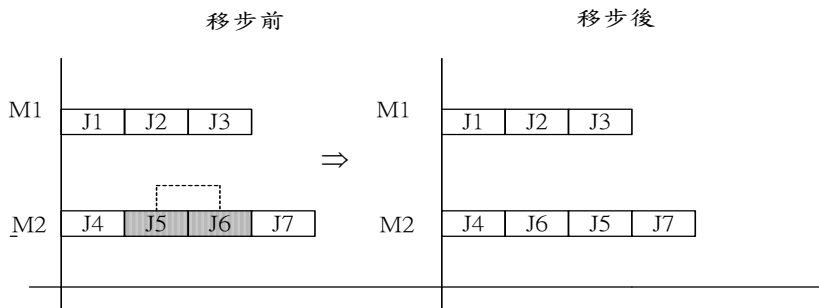


圖3 機台相同時移步示意圖

(3) 在某一溫度下，接受解目標值改變的次數是否達到預設的次數以上，未達成則終止。

本研究採用之停止準則說明如下：若溫度降低至預設終止溫度以下或全域較佳解(global best solution)改善之次數超過預設之數值(本研究設定為5次)，則終止流程。另外，為加速所建構模擬退火演算程序之搜尋，當特定溫度下之搜尋過程中(設定之馬可夫鏈長度)求解連續改善超過預設之數值(經過先期實驗之結果，本研究將此一預設之數值依其問題規模之大小分別設定為10與5)則終止此特定溫度下之搜尋過程而進行下一階段之降溫程序。

## 4. 實驗分析與討論

本研究針對考量整備時間與拆卸時間之總流程時間最小化之具非等效平行機台之多階流程型工廠排程問題，首先建構數學模式，並且發展一初始解演算程序，配合模擬退火演算程序進行求解之改善。實驗之分析分為數學規劃模式之求解與啟發式求解演算法之求解分析等兩部分。數學模式之求解軟體為 Extended LINGO 6.0，演算法程式編譯軟體 Borland C++ Builder 5.0，測試電腦CPU為 Pentium III 667MHZ，RAM為 256MB，作業系統為 Microsoft Windows 2000 Professional，針對各種排程環境以隨機產生範例對所建構之演算法進行測試與比較分析。

### 4.1 實驗數據之產生與參數設定

本研究所探討之環境數據設定如下：

- (1) 工件數目( $n$ )：系統中考慮5、15、30、50項工件。
- (2) 階段數目( $m$ )：系統中考慮5、15、20階段。
- (3) 平行機台( $k$ )：系統中考慮2、3部平行機台。
- (4) 問題規模( $n \times m$ )：考慮 $5 \times 5$ 、 $5 \times 15$ 、 $5 \times 20$ 、 $15 \times 5$ 、 $15 \times 15$ 、 $15 \times 20$ 、 $30 \times 5$ 、 $30 \times 15$ 、 $30 \times 20$ 、 $50 \times 5$ 、 $50 \times 15$ 、 $50 \times 20$ ，等12種環境問題規模。

本研究所探討之相關參數設定如下：

- (1) 本研究的工件加工時間包含整備( $S$ )、處理( $P$ )、拆卸( $R$ )三種作業時間，使用七組時間比例作為工件加工時間的考量。表4描述出三種加工作業與七組比例時間的關係。在產生各工件的作業時間時，首先先隨機產生一 $F$ 值， $F$ 值服從 $U(10,100)$ 。  
 整備時間( $S$ ):將此 $F$ 值乘上該工件整備時間的比例。  
 處理時間( $P$ ):將此 $F$ 值乘上該工件處理時間的比例。

表 4 七組時間比例表

組別	一	二	三	四	五	六	七
$S:P:R$	10:1:10	5:1:5	3:1:3	1:1:1	1:3:1	1:5:1	1:10:1

拆卸時間( $R$ ):因為本研究所考慮的工件拆卸時間會隨著後項加工工件的不同而有不相同的值產生,所以利用下述:上( $RU$ )、下限值( $RL$ )公式,產生出該工件在產生拆卸時間上的區間分布 $U(RL,RU)$ ,產生出該工件與其它工件彼此間拆卸時間的值。

上、下限值( $RL$ )公式:

$$RL = (F \times \text{該工件拆卸時間的比例}) - 4。$$

$$RU = (F \times \text{該工件拆卸時間的比例}) + 4。$$

- (2) 模擬退火演算法之起始溫度:分別有20、60、100、200四種起始溫度,進行實驗。
- (3) 模擬退火演算法之降溫率:分別有0.95、0.9、0.8三種降溫水準進行實驗。
- (4) 模擬退火演算法之馬可夫鏈長度:分別有10、20、30三種馬可夫鏈長度進行實驗。
- (5) 實驗題數:在十二種環境下分別考量七種比例題組,每種環境,產生十題,共計 $12 \times 7 \times 10 = 840$ 題。

#### 4.2 演算法之參數分析與設定

由於演算法之參數設定對於求解之品質與效率有較大之影響,故本研究亦針對4.1節中所提之12種不同之問題規模分別於各種參數設定下(4.1節)進行執行結果分析,期能在兼顧求解品質與求解時間下尋求較佳之參數設定。本節首先在不分問題規模下對影響模擬退火演算法求解品質與運算時間之三項參數;起始溫度、降溫率、馬可夫鏈長度作ANOVA分析,結果如表5所示。表5之分析結果顯示:在顯著水準 $\alpha = 0.05$ 下求解品質對於問題規模、起始溫度、降溫率、與馬可夫鏈長度等因子,都有顯著的影響。進而針對不同的問題環境下與對本演算法在求解品質與運算時間上有影響的三項參數;起始溫度、降溫率、馬可夫鏈長度進行分析。

由於影響模擬退火演算法之三項參數(起始溫度、馬可夫鏈長度、與降溫率)及其參數間之交互作用在誤差水準( $\alpha = 0.05$ )下對於求解之時間皆有顯著之影響,故僅彙整各種環境下影響

表5 影響演算法求解品質之參數ANOVA表(不分問題規模)

Effect	SS	DF	MS	F	P
A	1.071	11	0.0974	433.3	0.000*
B	0.043	3	0.0143	63.6	0.000*
C	0.121	2	0.0605	270.3	0.000*
D	0.139	2	0.0695	308.9	0.000*
A×B	0.022	33	0.0007	2.9	0.000*
A×C	0.026	22	0.0012	5.3	0.000*
A×D	0.029	22	0.0013	5.8	0.000*
誤差	6.695	29808	0.0002		

註:A-問題規模 B-溫度 C-降溫率 D-馬可夫鏈長度

求解品質之參數如表6。表6中由於各種問題規模下之因子交互作用皆無顯著影響故不列示。由表6之結果可知：在各種不同問題規模下之求解品質有著不同之因子效用，故本研究將各影響因子於不同問題規模之執行結果更進一步進行Duncan's分析，以尋求各影響因子在不同問題規模之最穩健設定。若在執行效率（時間）之考量下，由於各參數對不同之問題規模皆有顯著之影響，且其執行時間隨者各參數水準之提高而變長；因此在執行時間與求解品質兼顧之雙重考量下，當一參數對於求解品質之影響不顯著時，則該因子即設定於最低之水準以縮短求解之執行時間。各種不同問題規模下之參數設定方式彙整如表7。

### 4.3 起始解比較分析

本節將對於所提出的七種起始解方法在所考量的 840 題實驗題數中，進行實驗分析，共計分析次數  $840 \times 7 = 5880$  次。實驗結果使用平均相對誤差 (MRPD)，做為評估的指標，公式說明如下：

$$\text{平均相對誤差值 (MRPD) 公式: } \text{MRPD} = \frac{f_n - f^*}{f^*}$$

此處  $f_n$  = 第n種方法之求解值。  $f^*$  = 七種方法中最佳的解值。

七種起始解法在各種問題規模下（12種）之比較結果彙整如表8。由表8可明顯看出：H4在求解的品質上，所出現的相對誤差遠較其它起始解法為小，而且在出現較佳解的次數上，明顯多於其它六種起始解法。

表6 各種問題規模之因子影響彙整表

	5×5	5×15	5×20	15×5	15×15	15×20	30×5	30×15	30×20	50×5	50×15	50×20
B	✓			✓	✓	✓	✓	✓	✓	✓	✓	✓
C	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
D		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

註： B-溫度 C-降溫率 D-馬可夫鏈長度

表7 各種問題規模參數設定彙整表

問題規模	起始溫度	降溫率	馬可夫鏈	問題規模	起始溫度	降溫率	馬可夫鏈
5×5	20	0.90	10	30×5	100	0.95	30
5×15	20	0.95	20	30×15	100	0.95	30
5×20	20	0.95	20	30×20	200	0.95	30
15×5	100	0.95	30	50×5	200	0.95	30
15×15	100	0.95	30	50×15	200	0.95	30
15×20	100	0.95	30	50×20	200	0.95	30



表 8 七種起始解法之求解結果比較表

環境	H1	H2	H3	H4	H5	H6	H7
5×5	1.5645(31)	2.5891(13)	4.7071(1)	1.5107(24)	2.3146(13)	5.4512(0)	10.5053(0)
5×15	2.2846(26)	2.2540(3)	2.8904(3)	0.7985(29)	1.3840(17)	3.0705(3)	9.7202(0)
5×20	1.4553(24)	1.4587(22)	2.6024(5)	0.9839(23)	1.8545(12)	3.3220(4)	7.5248(2)
15×5	4.1668(8)	7.0663(4)	9.5988(0)	1.0006(36)	1.1045(28)	7.0117(0)	23.3874(0)
15×15	4.1190(4)	3.6991(2)	4.8373(0)	0.5832(36)	0.7710(28)	4.5749(0)	13.8797(0)
15×20	3.6505(7)	3.2230(5)	4.5475(2)	0.6162(33)	0.9255(23)	5.0708(0)	12.3855(0)
30×5	4.0533(5)	9.8984(0)	11.7115(0)	0.7285(36)	0.8164(29)	8.8639(0)	23.4761(0)
30×15	4.5351(1)	5.2375(0)	6.7286(0)	0.4998(40)	0.6077(29)	5.8862(0)	14.4210(0)
30×20	4.5854(2)	4.1609(0)	5.9476(0)	0.3829(41)	0.7313(26)	4.6225(0)	12.7166(0)
50×5	4.9279(0)	10.4524(0)	12.7280(0)	0.8056(31)	0.7501(39)	8.1862(0)	23.6928(0)
50×15	5.4894(0)	6.2203(0)	7.7036(0)	0.5938(34)	0.6917(36)	6.1607(0)	14.2982(0)
50×20	4.7148(1)	5.2356(0)	6.5703(0)	0.6498(37)	0.5894(32)	5.1647(0)	12.5958(0)
sum	45.54(109)	61.49(49)	80.57(11)	9.15(394)	12.54(312)	67.38(7)	178.60(2)

(n):表示有較佳解出現的次數

為求所選用的起始解法之正確性，進而以H4於小規模問題上和最佳解數學模式進行比較分析。礙於Lingo6.0對於此所研究問題最多只能求解到4工件\*3階段，所以我們在此只對Lingo6.0所能計算的最大題形範圍內考慮兩組環境，階段中的機台數目分別為 $m=2$ 、 $m=3$ 以及選用三組比例題組，在此三種考量環境下各產生五題，共計 $2*3*2*5=60$ 題，進行數學模式最佳解與H4之解值及時間比較分析。各問題規模下之解值與執行時間彙整如表9所示。表9之比較結果可看出在運算之時間上數學模式之求解相當費時，而演算法H4在求解小規模問題時相較於數學模式之求解有較佳之求解效率。在最佳解出現次數上，H4出現最佳解次數共計有22次，雖不能每題都是最佳解，但在平均解值上，H4也都能很接近最佳解的平均解值。由於H4在和其它方法比較後在求解品質上有較佳的表現，所以在建構所提出的以模擬退火法為基之啟發式演算法中將採演算法H4作為起始解之產生結構。

表 9 數學模式解與起始解方法四解值與運算時間比較表

問題規模	數學模式		H4	
	平均最佳解值	平均時間(秒)	平均解值	平均時間(秒)
3×3	862.7	3.84	867.8(12)	0.000
4×3	1331.0	265.2	1352.1(10)	0.000

(n):表示和最佳解相同的次數

#### 4.4 移步之選用

為求所建構的啟發式演算法在求解品質的穩定，針對3.3節中所提出的三種移步方式，在模擬退火演算法下使用相同的參數設定，進行比較分析。三種移步方式之求解品質與求解時間之ANOVA分析結果分別如表10、表11所示。表10之結果顯示，在顯著水準 $\alpha=0.05$ 下，三種移步方式在求解品質上有顯著影響。表11的結果則說明三種移步方式在求解時間上並無顯著影響。

由於表10中得知移步的方式對於求解的品質上有顯著影響，進而對三種移步方式之求解品質作進一步探討，經過Duncan's分析後發現移步二在求解品質上表現較優於其它兩種移步方式，所以本研究後續進行所採用的移步方式，選用移步二。

#### 4.5 啟發式演算法結果之比較

本節將對所提出以模擬退火法為基的啟發式演算法，在小規模問題上，用數學模式最佳解與啟發式演算法進行比較；較大規模的問題由於Lingo 6.0無法求算出最佳解，所以將所提出的其餘六種起始解方法套入模擬退火法改善程序中進行解值收尋改善與所建構之啟發式演算法進行比較分析。

##### 4.5.1 數學模式結果與啟發式演算法結果之比較

本節以Lingo6.0計算出小問題的最佳解與所提出的啟發式演算法，進行比較分析，用以驗證所提出的演算法在小規模問題的求解效率。驗證過程中演算法在參數的組合上，採用4.2節中，十二種問題環境下，所提出的最小規模問題5×5之建議參數組合，設定起始溫度20，降溫率0.9，馬可夫鏈長度10，和數學模式最佳解進行比較，比較分析結果彙整如表12所示。由表12之結果可得所提出的啟發式演算法（H4SA）在小規模問題求解上，均能達到最佳解，在時間上也都比數學模式在求解運算時間短。

表 10 三種移步之求解品質 ANOVA 分析表

Effect	SS	DF	MS	F	P
移步	0.008	2	0.0040	20.30	0.000*
誤差	0.210	1005	0.0002		

表 11 三種移步之求解時間 ANOVA 分析表

Effect	SS	DF	MS	F	P
移步	2.230	2	1.115	0.034	0.967
誤差	33246.910	1005	33.082		

表 12 數學模式與啟發式演算法之解值比較表

問題規模	數學模式		H4SA	
	平均最佳解值	平均時間(秒)	平均解值	平均時間(秒)
3×3	862.7	3.84	862.7(30)	0.017
4×3	1331.0	265.20	1331.0(30)	0.019

(n):表示與最佳解相同次數

#### 4.5.2 啟發式演算的比較分析

本節將所提出的其它六種起始解方法，架入模擬退火法之迭代改善架構中，和所提出的啟發式演算法（H4SA）進行比較。在演算法參數的設定上使用4.2節所建議的參數，實驗結果以平均相對誤差值作為七種方法評估的指標，分析結果如表13所示。表13之結果顯示：H4SA在所考量的十二種問題環境下，平均相對誤差值在七種演算法中平均而言為最佳，出現較佳解次數上，也是在七種演算法中最為多次的。將表8七種起始解演算法之求解品質與表13做比較分析後，可清楚看出所有方法在使用模擬退火法之迭代改善架構進行收尋改善後，都能有明顯的改善，可以驗證出模擬退火法對於此類問題求解上有相當不錯的表現。而H4SA在出現最佳解次數上雖少於起始解的比較結果，但是平均相對誤差上有相當程度的改善，由此可看出H4SA在此類問題的求解上，求解品質的穩定。

表 13 七種起始解法+SA 迭代改善程序之求解比較表

	H1SA	H2SA	H3SA	H4SA	H5SA	H6SA	H7SA
5×5	0.509(33)	1.027(24)	1.160(18)	0.816(26)	1.179(20)	0.971(22)	1.404(11)
5×15	0.436(33)	0.669(22)	0.815(13)	0.391(35)	0.829(15)	0.695(20)	0.743(11)
5×20	0.598(31)	0.671(30)	1.126(18)	0.525(28)	0.907(13)	0.812(22)	0.945(18)
15×5	1.513(9)	1.798(9)	1.564(9)	1.068(13)	1.097(15)	1.107(15)	1.462(4)
15×15	1.340(14)	1.454(9)	1.757(3)	1.001(13)	1.009(14)	0.909(9)	1.571(8)
15×20	0.975(13)	1.508(5)	1.335(10)	0.819(16)	0.818(17)	0.888(5)	1.359(4)
30×5	1.653(18)	5.249(2)	4.378(3)	1.097(17)	0.728(23)	1.645(7)	3.802(0)
30×15	1.220(14)	3.556(2)	3.951(1)	0.764(25)	0.787(19)	1.306(8)	3.070(1)
30×20	1.145(12)	2.486(8)	2.960(1)	0.595(27)	0.798(10)	0.897(12)	2.311(0)
50×5	1.700(10)	7.899(0)	8.006(0)	0.733(25)	0.632(28)	1.447(7)	5.755(0)
50×15	1.082(12)	5.312(0)	6.583(0)	0.448(36)	0.884(20)	1.417(2)	4.636(0)
50×20	1.178(10)	4.325(0)	5.271(0)	0.471(29)	0.726(26)	1.184(5)	4.105(0)
sum	13.35(209)	35.96(111)	38.91(76)	8.73(290)	10.39(220)	13.28(134)	31.16(57)

(n)代表出現較佳解的次數

## 5. 結論

本研究針對多階段非等效平行機台考量獨立整備時間及相依拆卸時間之流程式工廠排程問題，以總流程時間最小化為目標下，建構出數學模式以求取最佳解，並發展出一以模擬退火法為基之啟發式演算法用於中、大規模問題的求解。總結分析如下：

- (1) 由於多階段平行機台問題，屬於高複雜度的問題，數學模式最佳解，最大只能求取到 4 工件 \* 3 階段的問題規模，且最佳解再執行運算上相當耗時。
- (2) 在七種起始解方法的比較分析後，發現對於所研究之問題在起始加工順序的安排上以總處理時間由小到大依序之觀念作為工件加工順序安排之基準，一般而言，此類法則在求解品質上有較佳的表現。
- (3) 本研究所採用模擬退火法為基礎建構出啟發式演算法，配合所建構之起始解演算法則與移步法則對於探討之問題可獲得相當不錯之求解品質與求解效率。
- (4) 一個求解表現較好之起始解演算法對於最終解之求解品質有顯著之影響。

本研究針對探討問題所提出之移步法則，對於起始解演算法H4之改善效果不若其他種起始解法，可解釋為一個好的移步方式，對於最終解之影響甚大，一求解品質較差之起始解法則，若能配之以一較佳之移步，雖無法使最終解為最佳，但仍可獲得相當之求解品質。

## 誌謝

本論文承蒙國科會部分經費補助（計畫編號：91-2213-E-212-019）得以完成，謹此誌謝。

## 附錄：七種起始解演算法之簡介

以下將簡介所提出的起始解演算方法：

### • H1:

步驟0. 設定起始階段 $j$ ， $j=1$ 。

步驟1. 計算每個工件在每一階段之所有處理時間，將工件的總處理時間由小到大依序對工件排序，定義此順序為初始加工順序。

步驟2. 將所得的初始加工順序中，挑選最前面兩工件作互換，可得到新的組合排列。

步驟3. 分別將所產生的組合排列，依序進入階段 $j$ 中進行加工處理。

步驟3.1 工件依組合排列中的順序進入 $j$ 階段加工。

步驟3.2 工件 $i$ 在階段 $j$ 之機台選擇，為選用能較先完成工件 $i$ 在階段 $j$ 加工之機台。(工件在機台指派上若遇有相同的完工時間時，以機台編號順位優先選取)

步驟3.3 計算所有可能組合排列在 $j$ 階段中的完工時間，選擇較小者，為新的初始加工順序。

步驟4. 由步驟3.3中所獲得的初始加工順序，選擇下一個未作過互換的工件，並向前插入所有可能位置產生新的組合排列。

步驟5. 重複步驟3，直到所有工件都有互換過。

步驟6. 計算 $j$ 階段各工件的處理完工時間，由小到大對工件作排序(亦即找能較先進入下階段作處理作業的工件)，作為 $j+1$ 階段的初始加工順序。

步驟7.  $j=j+1$ ，重複步驟2。

• H2:

步驟1. 計算出每一工件在每一階段的所有處理時間，獲得每一工件的總處理時間，依照各工件的總處理時間的大小，由大到小依序對工件排序，獲得一組合排列，定義此組合排列為初始加工順序。

步驟2. 將所得的初始加工順序中，挑選最前面兩個工件做互換，可得到新的組合排列。

步驟3. 分別將所產生的組合排列，進入各階段中進行加工處理。

步驟3.1 工件進入各階段加工的順序為依照組合排列中工件的先後順序。

步驟3.2 工件 $i$ 在階段 $j$ 之機台選取；為選用能較先完成工件 $i$ 在階段 $j$ 加工作業之機台。(工件在機台指派上若遇有相同的完工時間時，以機台編號順位優先選取)

步驟3.3 計算所有可能組合排列之總流程時間，並以總流程時間最小者，作為下一迭次初始的加工順序。(組合排列中，遇有相同的最佳總流時間時，以先產生出來的組合，優先選取)

步驟4. 選取下一個尚未作過互換的工件，並向前插入所有可能的位置，產生新的組合排列後重複步驟3直到所有工件都已作過互換。

• H3:

步驟0. 設定起始階段 $j$ ， $j=1$ 。

步驟1. 計算出每一工件在每一階段的所有處理時間，獲得每一工件的總處理時間，依照各工件的總處理時間的大小，由大到小依序對工件排列，獲得一排列組合，定義此組合排列為初始加工順序，

步驟2. 將所得的初始加工順序中，挑選最前面兩個工件做互換，可得到新的組合排列。

步驟3. 分別將組合排列，依序排入階段 $j$ 。

步驟3.1 工件進入階段的順序，為依照組合排列中，工件的先後順序。

步驟3.2 工件在階段上排入機台的選擇，為選用能較先完成工件 $i$ 在階段 $j$ 加工作業之機台。

步驟4. 往後的 $j+1$ 階段，工件進入的順序為；對工件在 $j$ 階段上開始處理完工時間之大小，由小到大對工件排序，得一新組合排列，重複步驟3。

步驟5. 選取所有組合排列中，總流程時間較小者，作為下一迭次的初始組合排列。

步驟6. 由初始的組合排列中，挑選下一個未排的工件，往前插入所有可能的位置，產生出新的組合排列。

步驟7.  $j=1$ ，重複步驟3，直到所有工件都已做過交換。

• H5:

步驟0. 設定起始階段 $j, j=1$ 。

步驟1. 計算出每一工件在每一階段的所有處理時間，獲得每一工件的總處理時間，依照各工件的總處理時間的大小，由小到大依序對工件排列，獲得一排列組合，定義此組合排列為初始加工順序，

步驟2. 將所得的初始加工順序中，挑選最前面兩個工件做互換，可得到新的組合排列。

步驟3. 分別將組合排列，依序排入階段 $j$ 。

步驟3.1 工件進入階段的順序，為依照組合排列中，工件的先後順序。

步驟3.2 工件在階段上排入機台的選擇，為選用能較先完成工件 $i$ 在階段加工作業之機台。

步驟4. 往後的 $j+1$ 階段，工件進入的順序為；對工件在 $j$ 階段上開始處理完工時間之大小，由小到大對工件排序，得一新組合排列，重複步驟3。

步驟5. 選取所有組合排列中，總流程時間較小者，作為下一迭次的初始組合排列。

步驟6. 由初始的組合排列中，挑選下一個未排的工件，往前插入所有可能的位置，產生出新的組合排列。

步驟7.  $j=1$ ，重複步驟3。

• H6:

步驟0. 設定起始階段 $j, j=1$ 。

步驟1. 計算出每一工件在所有階段的處理時間，獲得每一工件的總處理時間，依照各工件的總處理時間大小，由小到大依序對工件排列，獲得一組排列。

步驟2. 將此一組排列中依工件先後的順序，進入階段 $j$ 作機台選派，工件 $i$ 在階段 $j$ 加工之機台選取法則；為選用為能較先完成工件 $i$ 在階段 $j$ 加工作業之機台。

步驟3. 計算 $j$ 階段中各工件的處理時間由小到大對工件作排序，獲得一新排列，

步驟4.  $j=j+1$ ，重複步驟2。

• H7:

步驟0. 設定起始階段 $j, j=1$ 。

步驟1. 計算出每一工件在所有階段的處理時間，獲得每一工件的總處理時間，依照各工件的總處理時間大小，由大到小依序對工件排列，獲得一組排列。

步驟2. 將此一組排列中依工件先後的順序，進入階段 $j$ 作機台選派，工件 $i$ 在階段 $j$ 加工之機台選取法則；為選用為能較先完成工件 $i$ 在階段 $j$ 加工作業之機台。

步驟3. 計算 $j$ 階段中各工件的處理時間由小到大對工件作排序，獲得一新排列。

步驟4.  $j=j+1$ ，重複步驟2。

## 參考文獻

- 田國興,「有設置時間之流程型工廠多階段平行機總排程時間最小化問題」,中原大學工業工程研究所碩士論文,民國 88 年。
- 張百棧,「基因演算法於非等效平行機台排程應用」,工業工程學刊,第十九卷第二期,民國 91 年,79-95 頁。
- 張毓仁,「多階段不等效平行機之排程研究」,中原大學工業工程研究所碩士論文,民國 89 年。
- 楊文華,「考慮設置時間下之相關排程問題研究」國立台灣工業技術學院管理技術研究所博士論文,民國 85 年。
- 駱景堯、葉進儀、駱芳梧,「整備與拆卸時間考量之開放型排程問題之求解模式建立」,工業工程學刊,第二十卷第六期,民國 92 年,565-574 頁。
- Adamopoulos, G. I. and Pappis, C. P., "Scheduling Under a Common Due-Date on Parallel Unrelated Machines," *European Journal of Operational Research*, Vol. 105, 1998, pp.494-501.
- Alidaee, B. and Rosa, D., "Scheduling Parallel Machines to Minimize Total Weighted and Unweighted Tardiness," *Computers and Operations Research*, Vol. 24, No. 8, 1997, pp.775-788.
- Allahverdi, A., "Minimizing Mean Flowtime in a Two-Machine Flowshop with Sequence-Independent Setup Times," *Computers and Operations Research*, Vol. 27, 2000, pp.111-127.
- Allahverdi, A. and Aldowaisan, T., "No-Wait and Separate Setup Three-Machine Flowshop with Total Completion Time Criterion," *International Transactions in Operational Research*, Vol. 7, 2000, pp.245-264.
- Azizoglu, M. and Kirca, O., "Tardiness Minimization on Parallel Machines," *International Journal of Production Economics*, Vol. 55, 1998, pp.163-168.
- Balakrishnan, N. and Kanet, J. J., "Early/Tardy Scheduling with Sequence Dependent Setups on Uniform Parallel Machines," *Computers and Operations Research*, Vol. 26, 1999, pp.127-141.
- Brah, S. A. and Loo, L. L., "Heuristics for Scheduling in a Flow Shop with Multiple Processors," *European Journal of Operational Research*, Vol. 113, 1999, pp.113-122.
- Dessouky, M. M. and Dessouky, M. I., "Flow Shop Scheduling with Identical Jobs and Uniform Parallel Machines," *European Journal of Operational Research*, Vol. 109, 1998, pp.620-631.
- Franca, P. M., Gendreau, M. and Laporte, G., "A Tabu Search Heuristic for the Multiprocessor Scheduling Problem with Sequence Dependent Setup Times," *International Journal of Production Economics*, Vol. 43, 1996, pp.79-89.
- Kim, D-W., Kim, K-H., and Jang, W-S., "Unrelated Parallel Machine Scheduling with Setup Times Using Simulated Annealing," *Robotics and Computer Integrated Manufacturing*, Vol. 18, 2002,

pp.223-231.

Koulamas, C. and Kyparisis, G. J., "Scheduling on Uniform Parallel Machines to Minimize Maximum Lateness," *Operations Research Letters*, Vol. 26, 2000, pp.175-179.

Proust, C., Gupta, J. N. D., and Deschamps, V., "Flowshop Scheduling with Set-up, Processing and Removal Times Separated," *International Journal of Production Research*, Vol. 29, 1991, pp.479-493.

Santos, D. L., Hunsucker, J. L., and Deal, D. E., "Global Lower Bounds for Flow Shop with Multiple Processors," *European Journal of Operational Research*, Vol. 80, 1995, pp.112-120.

Santos, D. L., Hunsucker, J. L., and Deal, D. E., "An Evaluation of Sequencing Heuristics in Flow Shops with Multiple Processors," *Computers Industrial Engineering*, Vol. 30, No.4, 1996, pp.681-692.

Sule, D.R., "Sequencing n Jobs on Two Machines Separated," *Naval Research Logistics Quarterly*, Vol. 29, 1982, pp.517-519.

Suresh, V., and Chaudhuri, D., "Bicriteria Scheduling Problem for Unrelated Parallel Machines," *Computers and Operations Research*, Vol.30, No. 1, 1996, pp.77-82.

Yoshida, T. and Hitomi, K., "Optimal Two-Stage Production Scheduling with Setup Times Separated," *AIIE Transactions*, Vol. 11, 1979, pp.261-263.