

最適生產週程時間與工作站佈置之研究

The Study of Optimal Cycle Time and Workstation Layout

藍俊雄¹ 楊禮鴻²

摘要

本研究主要探討生產線平衡問題，並提出一個單一模式下生產線平衡之型二問題模式(Type II of single-model assembly line balancing problem, T2P)。本模式乃藉由數值範例模擬在不同的給定工作站數目下，利用 LINGO 套裝軟體來求取最佳的生產週程時間(Cycle Time)，並規劃出各工作站佈置的最佳解，因此本研究乃屬於動態的型二問題(Type II Problem)。另外，本研究亦提供了一套具備了高度的可重複性且完善的研究流程，並將分析結果載入至視窗軟體介面，發展出專屬的生產線佈置的決策支援平台。綜言之，本研究乃運用 LINGO 求解生產線平衡的型二問題，並嘗試開發一套決策支援系統(Decision Support Systems, DSS)檢核表，以為生產線設計佈置者提供一個嶄新且迅速的佈置理念。

關鍵字：生產線平衡、型二問題、週程時間、工作站佈置

Abstract

This study discusses an assembly line balancing problem and proposes one type II of single-model assembly line balancing problem (T2P). The discussed model tries to find out the optimal number of workstations for a given numerical example, and then uses LINGO 9.0 software to determine the optimal cycle time and the layout in every workstation; therefore, this study is categorized as the dynamic type II Problem. In addition, this study has also provided a set of high repeatability and perfect research procedure, and the analysis result can be input into the developed computer interface to be the exclusive assembly line balancing of decision support platform. The LINGO 9.0 is applied to solve such type II problem for assembly line balancing, and a decision support system (DSS) examined table is established, so that provide the assembly line balancing engineer a tool for quick workstation layout.

Keywords: Assembly Line Balancing, Type II Problem, Cycle Time, Workstation Layout

¹南華大學管理科學研究所教授

²南華大學管理科學研究所碩士

1. 前言

由於科技的進步致使產業結構不斷地改變，從人工製造演進到自動化設備的生產，而生產方式的改變說明了企業經營乃在於提高製造效率、降低人工成本、並獲取最大的利潤為目標(劉慧芬, 2002)。事實上，生產週程時間(Cycle Time)決定了顧客最關注的訂單交期日(Order Due Date)，同時也扮演著產能規劃與工單排程的基本參考依據(黃宏文, 2002)。有鑑於此，有關生產線平衡問題(Assembly Line Balancing Problem, ALBP)也就成為眾多研究的重點了。

生產線設計的研究發展至今已超過四十年了，最早的研究是在1954年Bryton定義關於生產線平衡問題，開啟了這個學門的研究，隔年1955年Salveson首先建構第一個生產線平衡問題的數學模型並進而提出一個數量化的求解步驟(Erel and Gokcen, 1999)。在這段期間中陸陸續續有許多相關的研究被提出，其中Kim等學者在1996年將生產線平衡問題分成五大類型的問題，其中型一問題(Type I Problem)與型二問題(Type II Problem)是兩種最基本的生產線平衡之最佳化問題。

上述的兩大問題廣泛地被應用在生產系統的佈置設計上。其中大多數研究集中於要求某一產品(或者一個規定的週期)在滿足規定的市場需求條件下，如何使工作站的數量減到最小的設計方法；這類的問題被稱之為型一問題(Ugurdag *et al.*, 1997)，或者在固定的工作站數量且不擴增新機器的條件下，如何使生產數量達到最佳化的情況，也可藉由生產線平衡來達成(Chen *et al.*, 2002)；上述這些問題都在生產線平衡上經常發生，此類問題稱之為生產線平衡的型二問題(Type II Problem)，即給定的工作站的數量使週程時間最小化，而此型二問題亦是本研究的研究焦點。

在二十世紀的60年代和70年代對於生產線平衡的研究是十分受歡迎的。自從Baybars在1986驗證出生產線平衡問題本身即是非多項式(NP-hard)型的問題之後，在這段時間中，針對ALBP之求解方式也相繼的被提出(Ghosh and Gagnon, 1989)。大多數學者早期使用的方法是以一連串的生產線平衡的型一問題(Type I Problem)設計來解決型二問題(Hackman *et al.*, 1989)。這類方法最初是在1961年由Helgeson和Bimie所提出，之後陸續有相關方法被提出；一直發展至今ALBP最佳化的解決辦法已經可以透過啟發式演算法以及套裝軟體來獲取。

本研究嘗試運用數學模式來處理生產線平衡的問題，因此本研究依據數值範例建構出一個生產線平衡之型二問題的數學模式，並以LINGO-Extended Version套裝軟體求取最佳解；本模式乃藉由數值模擬在不同的給定工作站數目下求取最佳的生產週程時間，並規劃出各工作站佈置的最佳解，因此本研究乃屬於動態的型二問題。本研究並依其分析結果，開發一套決策支援系統(Decision Support Systems, DSS)，以為決策經理人提供在工作站設計時之參考依據。

2. 符號說明

$$x_{ik} : \begin{cases} 1 & \text{若第 } i \text{ 個工作單元指派到第 } k \text{ 個工作站；} \\ 0 & \text{其他。} \end{cases}$$

$$x_{jk} : \begin{cases} 1 & \text{若第 } j \text{ 個工作單元指派到第 } k \text{ 個工作站；} \\ 0 & \text{其他。} \end{cases}$$

k : 工作站的數目

T_i : 第 i 個工作單元的作業時間

CT : 生產週程時間

3. 構建數學模式

基於上述的符號說明，本研究提出一個數學模式稱之為單一模式下生產線平衡之型二問題模式(Type II of single-model assembly line balancing problem, T2P)，並將此模型說明如下：

$$\text{T2P} \left\{ \begin{array}{l} \min_{x_{ik}, k} CT \quad (1) \\ \text{s.t.} \quad \sum_i x_{ik} = 1 \quad \forall k \quad (2) \\ kx_{jk} - kx_{ik} \geq 0 \quad \forall i < j \quad (3) \\ \sum_i (T_i \times x_{ik}) \leq CT \quad \forall k \quad (4) \\ x_{ik} = 0 \text{ or } 1 \quad (5) \end{array} \right.$$

T2P 模式之目標函數式(1)是針對某一給定的生產過程，追求一組最佳的生產線佈置以使生產週程時間最小化。限制式(2)是指每一個工作單元必須且只能分配給一個工作站，不能重複作指派；限制式(3)表示後行的生產單元 j 配置在的工作站必須在先行生產單元 i 所在的工作站之後或在相同的工作站中；限制式(4)表示當所有生產單元均佈置入工作站後，所有佈置有生產單元的工作站中，擁有最大所需製程時間的工作站之製程時間，即為該生產佈置之生產週程時間；限制式(5)代表每一個生產單元的佈置變數為 0 或 1 之二元變數。

k 與 x_{ik} 乃為本模式中之兩個決策變數。下節中將介紹本研究所使用之數值範例，並將數值範例透過 LINGO 語法使之程式化，使可藉由 LINGO 套裝軟體進行求解。經由執行求解之後，最佳之工作站配置數目 k^* 與最佳之生產週程時間 CT^* 均可同時被求出。

4. 數值範例與 LINGO 語法編輯

本研究嘗試以一個數值範例進行說明如何求解生產線平衡之型二問題。探討的數值範例其各工作單元的先後順序可繪製成網路先行圖(如圖 1 所示)。圖 1 中共有 26 個工作單元(由 A 到 Z)，欲指派到 k 個工作站中，其中 k 為可變的變數。圖中所有的工作單元所需的作業時間詳列於表 1：

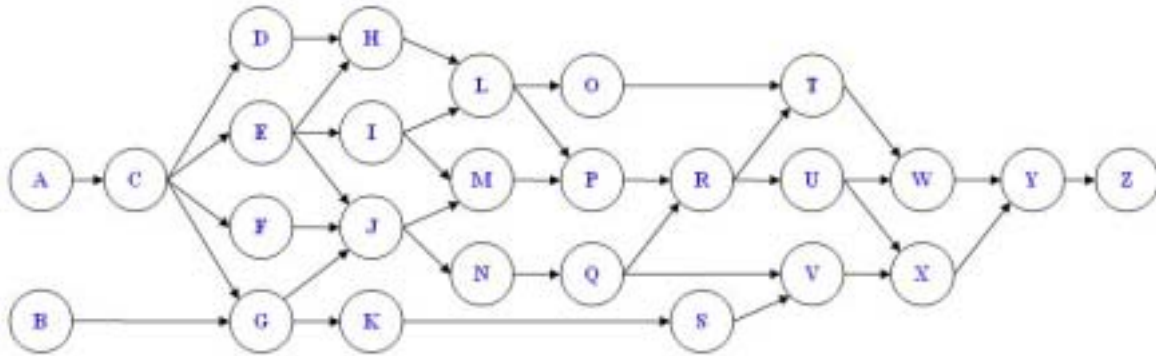


圖 1 網路先行圖

表 1 所有工作單元的作業時間

工作單元	A	B	C	D	E	F	G	H	I	J	K	L	M
時間(min)	77	73	66	25	36	71	47	15	82	98	90	62	20
工作單元	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
時間(min)	55	24	18	88	72	45	62	38	36	45	67	53	18

在完成集合宣告與資料輸入之後，本研究將利用LINGO語法加以編輯模型。首先下達本模式之目標為追求最小化的生產週程時間，接著下達一連串的限制條件，如讓每一個工作單元僅允許配入一個工作站內、每一個工作單元只能與其先前的單元佈置在同一個工作站內或佈置於之後的工作站中；此外本研究並限制生產週程時間必須要大於或等於每一工作站中被佈置在該工作站內的所有工作單元作業時間之總和。本模式包含整數變數、0與1變數，所以求解方法乃採用枝節法(branch-and-bound, B-and-B)來求取最佳解，完整的語法編輯詳列於附錄A中。

5. 模擬分析

本節主要是探討前述的數值範例在給定的不同工作站數目 K (由1變動到23)的要求下，最佳解(最小週程時間)的變動情形。所有數值模擬的求解過程均是在同一台個人電腦上完成，個人電腦的硬體配備為：CPU為Intel Pentium 4 3E GHz，記憶體為1GB。軟體方面：作業系統為Microsoft Windows XP Professional，使用的求解軟體及應用程式為LINGO 9.0、Microsoft Visual C++及Microsoft Office中的Excel。經由LINGO求解後的執行結果摘錄於表2。當 K 增加時，生產所需的週程時間會呈現遞減的情況，而生產率則呈現遞增的情況。且當可佈置的工作站數超過17(含 $K=17$ 的情況)時，則最佳的生產週程時間會呈現穩定不變的情況(此時生產週程時間為98分鐘)，而當可佈置的工作站數大於23

的情況時，此時求解的過程開始出現冗餘工作站的現象，此現象代表工作站數目已超過可佈置的工作單元數目，故本研究所能討論的工作站數目(K)範圍介於1至23之間。

表2 執行結果分析

工作站數目	生產週程時間(mins)	生產率	最小空間長度	最小空間寬度	最小空間
1	1383	0.0007	11a	4b	44ab
2	693	0.0014	9a	3b	27ab
3	463	0.0022	6a	3b	18ab
4	350	0.0029	6a	3b	18ab
5	281	0.0036	4a	3b	18ab
6	239	0.0042	4a	2b	12ab
7	204	0.0049	3a	2b	6ab
8	178	0.0056	4a	2b	8ab
9	164	0.0061	3a	2b	6ab
10	143	0.0070	3a	2b	6ab
11	137	0.0073	3a	2b	6ab
12	133	0.0075	3a	2b	6ab
13	118	0.0085	3a	2b	6ab
14	110	0.0091	2a	2b	4ab
15	108	0.0093	2a	2b	4ab
16	100	0.0100	2a	2b	4ab
17	98	0.0102	2a	2b	4ab
18	98	0.0102	2a	2b	4ab
19	98	0.0102	2a	2b	4ab
20	98	0.0102	2a	2b	4ab
21	98	0.0102	2a	2b	4ab
22	98	0.0102	2a	2b	4ab
23	98	0.0102	2a	2b	4ab

數值範例透過不同工作站數目時的模擬，其最佳解的佈置情形列於圖2；本範例並假設每個工作單元所佔空間的長與寬均相似(設一個單元的長為1a單位，一個單元的寬為1b單位)，因此藉由計算不同工作站數目時的最佳佈置所需的長寬的單位數，可計算出整體生產線所需的最佳空間大小，並在有限資源利用的前提之下，透過決策支援系統(Decision Support Systems, DSS)提供最佳的工作站佈置數目，以決定如何設計以符合產業自身的產品線。

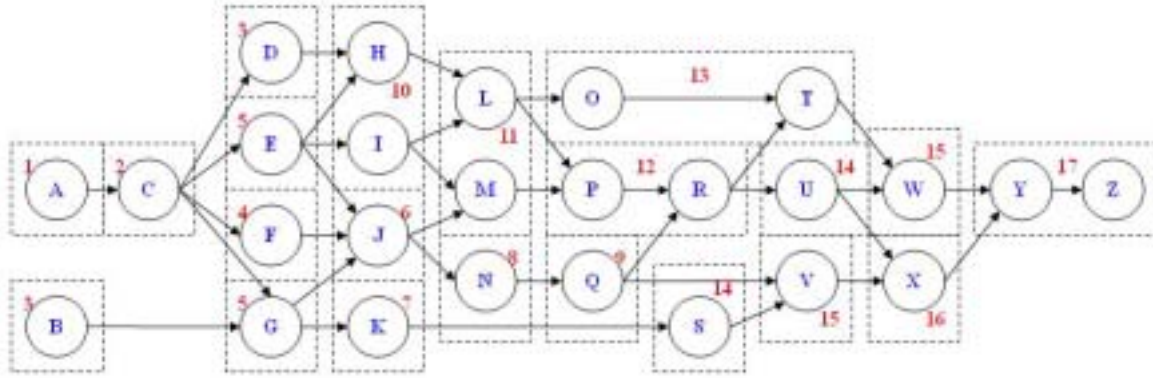


圖2 最佳工作站佈置圖

本研究將上述模擬分析之結果，藉由使用Microsoft Visual Basic 6.0設計程式，發展一套生產線平衡決策支援系統，如圖3所示，本系統亦可直接提供此數值範例的最小生產週程時間之佈置為工作站數目為17時，此時最小生產週程時間為98分鐘，生產率為0.0102，而廠房所需的空間大小為**4ab**，並詳列工作站佈置情況與佈置圖，以為生產線最佳設計之參考依據。

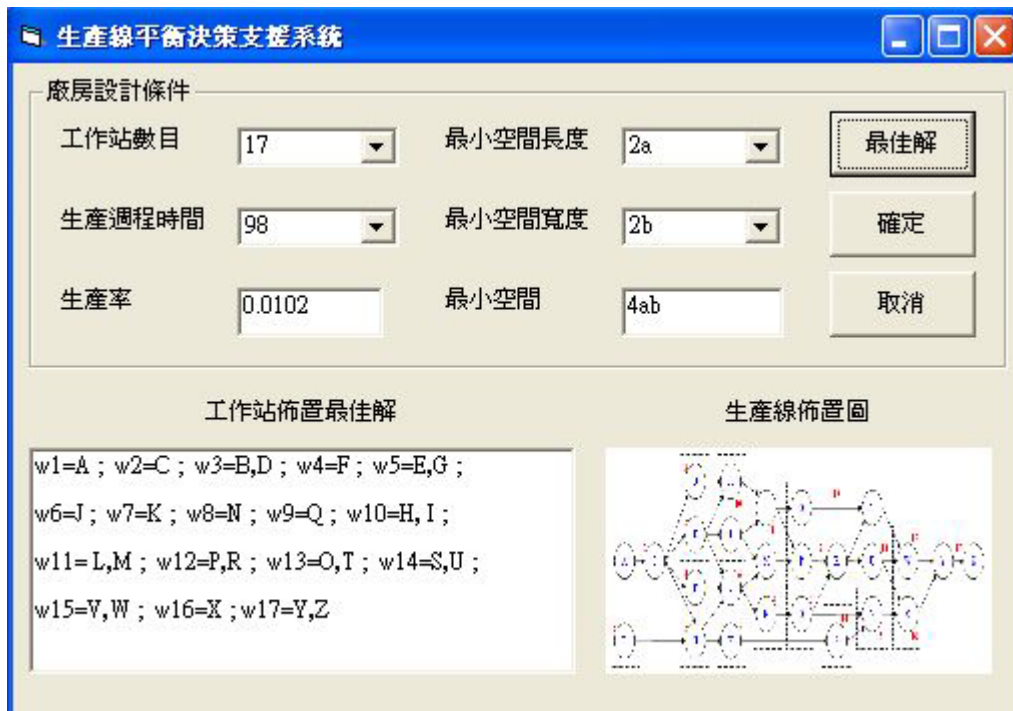


圖3 生產線平衡決策支援系統執行圖

此外，在已知廠房可用空間的大小與尺寸(長度與寬度)，並在訂單的產品的訂購數量與交貨時間已知的情形之下(此時最低生產率可計算得出)，最後可透過空間的大小尺寸與所需生產率的配合，將可藉由資料庫搜尋法則快速地得出最佳的工作單元與工作站

佈置(Layout)的情況。舉例來說：當一廠房的可用空間為 **8ab** 單位(其中長度與寬度分別為 **4a** 單位與 **2b** 單位)，且有一訂單數量為 360000，在一天工作 12 小時的情況之下，要求在 300 個工作天內完成，所以此生產線的最低生產率為 0.003，而後將必要的資訊輸入本系統的指定位置，透過此系統的搜尋技術得知：此時可顯示出工作站數目為 6 和 8 之時的佈置情況皆符合最低生產率與空間尺寸的限制，所得之最佳佈置所產生的生產率均符合大於 0.003 的要求，因此本系統提供一個具彈性且更接近實務的佈置理念。

6. 結論與建議

本研究所建構的單一模式下生產線平衡之型二問題模式，在過去的研究中已有多數的學者利用啟發式演算法來求解，但其設計流程過於冗長且繁瑣，且通常僅適用於單一個案且不具彈性，因此近年來研究者探討生產線的平衡問題的意願較低。而本研究藉由 LINGO 套裝軟體，藉由語法限制的下達，並透過決策支援平台，可使此類問題擁有絕佳的求解能力與更具彈性的求解過程。

本研究另一主要貢獻為當廠房之可用空間的大小與尺寸、訂單的訂購數量與交貨時間已知的情形之下，可藉由決策支援系統以快速得出最佳的生產週程時間與工作站佈置情形。不同工作站數目的佈置方法均有相對應的生產率與空間大小，此可供廠房空間規劃之參考依據。再者，本研究雖僅以單一的數值範例來進行介紹，但絕對提供了一套完善的研究流程。研究者僅需根據本研究所提供的流程架構，即使面對不同的實務案例，依據執行結果，並輸入至生產線平衡決策支援平台，還是可以得出所需的結果。換句話說，本研究的本質乃具備了高度的可重複性與彈性。

後續研究亦可陸續討論各種不同類型的生產線平衡問題，並利用 LINGO 套裝軟體來求取最佳解，亦可嘗試開發更具知識化與彈性決策支援平台，以提供生產線設計佈置者更快速的決策參考依據。

參考文獻

1. 黃宏文(2002)，「晶圓製造廠區段基礎式週程時間估算法」，交通大學工業工程與管理學系博士論文
2. 劉慧芬(2002)，「雙機分工模式下多樣 PCB 類型製造時間最佳化問題之研究」，元智大學工業工程與管理學系碩士論文。
3. Baybar, I. (1986), "A Survey of Exact Algorithms for the Simple Assembly Line Balancing Problem", *Management Science*, 32, pp.909-932.
4. Bryton B. (1954), "Balancing of a Continuous Production Line", *Management Science thesis, North-Western University*.
5. Chen, R. S., K. Y. Lu & S. C. Yu (2002), "A hybrid genetic algorithm approach on multi-objective of assembly planning problem", *Engineering Applications of Artificial Intelligence*, 15, pp.447-457.

6. Erel E. & H. Gokcen (1999), “Shortest-route formulation of mixed-model assembly line balancing problem”, *European Journal of Operational Research*, 116, pp.194-204.
7. Ghosh, S. & R. J. Gagnon (1989), “A Comprehensive Literature Review and Analysis of the Design”, *Balancing and Scheduling of Assembly Systems, International Journal of Production Research*, 27, pp.637-67.
8. Hackman, S. T., M. J. Magazine & T. S. Wee (1989), “Fast, effective algorithms for assembly line balancing problems”, *Journal of Operational Research*, 37(6), pp.916–924.
9. Heizer, J. & B. Render (2001), *Operations Management*, 6th ed., New Jersey: Prentice Hill, Inc.
10. Helgeson, W. R. & D. D. Bimie (1961), “Assembly Line Balancing using the Ranked Positional Weight Technique”, *Journal of Industrial Engineering*, 12, pp.394-398.
11. Kim, Y. K., Y. J. Kim, & Y. Kim (1996), “Genetic algorithms for assembly line balancing with various objectives”, *Computers and Industrial Engineering*, 30(3), pp.397-409.
12. *LINGO user’s guide* (2003), Chicago, IL: LINDO System Inc.
13. *Optimization Modeling with LINGO* (2003), 5th ed., Chicago, IL: LINDO System Inc.
14. Salveson ME. (1955), “The Assembly Line Balancing Problem”, *J. Ind. Eng.*, 6, pp.18-25.
15. Ugurdag, H. F., R. Rachamadugu & C. A. Papachristou (1997), “Designing paced assembly lines with fixed number of stations”, *European Journal of Operational Research*, 102, pp.488-501.

附錄A

A.1 集合宣告

在這個模式中，其主集合(Primitive Sets)有兩個，分別為TASK和STATION。從這些主集中又可以創造出兩個延伸集合(Derived Sets)。而有關工作站數 k 變數是藉由使用Microsoft Visual C++設計成迴圈使其能夠從1依序模擬至24。在TASK這個延伸集合中的每一元素均由生產線中各工作單元所構成。

```
TASK/ A B C D E F G H I J K L M N O P Q R S T U V W X Y Z /: T;
```

```
STATION/ 1..k /;
```

PERD是第一個延伸集合，其中的各元素均由 $(TASK^1, TASK^2)$ 所構成。這個集合主要是設定任意個生產單元($TASK^2$)與其先行單元($TASK^1$)之關係。例如：在這集合中第一個成員(A, C)即表示工作單元(A)為工作單元(C)的先行工作單元。

```
PRED(TASK, TASK)/ A,C B,G C,D C,E C,F C,G D,H E,H E,I E,J F,J
G,J G,K H,L I,L I,M J,M J,N K,S L,O L,P M,P N,Q O,T P,R
Q,R Q,V R,T R,U S,V T,W U,W U,X V,X W,Y X,Y Y,Z /;
```

TXS是第二個延伸集合，由TASK與STATION集合所構成的複合式延伸集合，X代表TXS此延伸集合中的元素代號，假使 $X(A, 1)=1$ 表示工作單元(A)佈置在第1個工作站之中。

```
TXS( TASK, STATION): X;
```

A.2 資料設定

集合宣告完後的資料設定將每一生產單元所需的作業時間並順序依序載入。

```
T= 77 73 66 25 36 71 47 15 82 98 90 62 20 55 24 18 88 72
45 62 38 36 45 67 53 18;
```

4.3 目標函數設定

本研究的目標函數為針對某一給定的生產過程，追求一組最佳的生產線佈置以使生產週程時間最小化，其設計如下：

```
MIN = CYCTIME;
```

4.4 限制式設定

本模型主要有五大項限制條件：每一個工作單元僅能佈置在一個工作站；每一工作單元必須佈置在先行工作單元的相同工作站或是後面的工作站；生產週程時間必須要大於或等於每一工作站中被佈置在該工作站內的所有生產單元作業時間之總和。

```
@FOR( TASK( I): @SUM( STATION( K): X( I, K)) = 1);
```

```
@FOR( PRED( I, J):
```

```
@SUM( STATION( K):
```

```
K * X( J, K) - K * X( I, K)) >= 0);
```

@FOR(STATION(K):

@SUM(TXS(I, K): T(I) * X(I, K)) <= CYCTIME);

下列設定使TXS集合為0、1變數：

@FOR(TXS: @BIN(X));