

# 行政院國家科學委員會補助專題研究計畫成果報告

.....  
300mm 多腔製程設備之自動建模及最佳排程  
.....

計畫類別：個別型計畫

計畫編號：NSC 90-2212-E-009-066

執行期間：90年8月至91年7月

計畫主持人：李安謙 交通大學機械系 教授

共同主持人：

計畫參與人員：洪榮煌 交通大學機械系 博士班學生  
張德森 交通大學機械系 碩士班學生

本成果報告包括以下應繳交之附件：

赴國外出差或研習心得報告一份

赴大陸地區出差或研習心得報告一份

出席國際學術會議心得報告及發表之論文各一份

國際合作研究計畫國外研究報告書一份

執行單位：

中華民國 91 年 10 月 16 日

# 300mm 多腔製程設備之自動建模及最佳排程

## Automatic modeling and optimal scheduling of cluster tool for 300 mm wafers

計畫編號：NSC 90-2212-E-009-066

執行期限：90年8月 - 91年7月

主持人：李安謙 E-mail: [aclee@cc.nctu.edu.tw](mailto:aclee@cc.nctu.edu.tw)

執行單位：交通大學機械系

### 一、中文摘要(關鍵詞：集束型製程設備、Petri Nets、即時啟發式搜尋法)

考慮集束型製程設備 (Cluster Tool) 的排程問題，可利用 Petri Nets 建構模型，並藉由搜尋分析模型所產生的延展樹 (reachability tree) 來找出最佳排程。雖然搜尋得到的排程即為最佳，但往往需時數天才能找出結果。若以工業上實用的角度而言，上述搜尋排程法是不切實際的。

本計畫使用即時啟發式搜尋法 (Real-Time Heuristic Search) 應用到上述之排程搜尋的問題上，以即時搜尋的方式獲得最佳或較佳的排程結果，並透過與離線式搜尋法所得之排程結果比較，來說明即時搜尋所得之排程時間，其搜尋結果與離線式搜尋法所得之排程時間幾乎相同。此外，亦比較即時搜尋法與離線搜尋法之兩者平均每步搜尋時間，利用即時搜尋法之平均每步搜尋時間並不會隨排程總處理程序的增加而增加。

### 英文摘要(Keyword: Cluster Tool, Petri Nets, Real-Time Heuristic Search)

Considering the optimal scheduling question of the Cluster Tool, which can use Petri Nets model to generate and search the reachability tree. But this method take much time in searching procedure. In the practical use of industry, this solution is not proper.

In this paper, we develop a method that is applying real-time heuristic searching

algorithm to obtain optimal or near-optimal scheduling. To compare the spending time of our searching result and off-line searching result, they are almost equal. We also compare the searching time of real-time and off-line algorithm. The real-time searching time will not increase with the growth of scheduling procedure.

### 二、計劃緣由與目的

面對半導體製造業對於高潔淨度及高生產量的需求，半導體設備製造廠商推出具有多製程腔體的集束型製程設備 (Cluster Tool)。該設備為一高度自動化的製造系統，其內包含許多製程模組，晶圓透過輸送模組的運送在各個製程模組之間移動，完成其所需的製程。由於處理過程當中完全採用機械來運送晶圓，因此減少了晶圓的人為污染，相對地也提升了生產良率與產能。

為了提升集束型製程設備的產能，找尋一個有效率的排程成為眾多論文研究 [1][2][3][4][5] 的主要焦點。由不同的輸送程序、晶圓個數、製程腔體處理時間及傳輸晶圓時間所產生的排程問題，透過 Petri Nets 對於集束型製程設備的模型建立，以延展樹 (reachability tree) 分析的方式將所有的排程列舉出來，逐一比較各個排程所消耗的時間，來找出最佳的排程。由於此法的所有過程皆可轉換為電腦的演算法，因此若能夠將此法與集束型製程設備軟體控制器的設計相結合，實際應用到設備的控制上，將大幅且直接改善集束型製程設備的產能。然而以目前搜尋法的搜尋速度來搜尋一最佳排程往往耗時數小時、數

天、甚至數個禮拜，因此以實用的觀點而言，將搜尋最佳排程法與集束型製程設備軟體控制器相結合是不切實際的。

集束型製程設備軟體控制器為一即時系統 (Real-time system)，內部有許多同時執行的程序，而這些程序必須在系統要求的時間限制下更迭地執行，藉以達到即時監控與回應可能發生的狀況等。因此若要將搜尋最佳排程的方式應用到集束型製程設備軟體控制器，首先必須使其滿足即時的特性。本計畫的研究目的即在於利用即時的搜尋法則，在滿足即時系統的時間限制下，搜尋出目前所可能進行的最佳排程，使搜尋最佳排程的方法能夠結合到集束型製程設備軟體控制器的設計當中。

由於利用 Petri Nets 模型所產生的延展樹並不能完整地表達系統的資訊，因此在進行排程即時搜尋之前，必須對延展樹作一修正，才能獲得正確之排程結果。在本計畫中，將說明延展樹的修改過程，及如何將即時搜尋法應用到延展樹當中以獲得正確的排程，並比較即時搜尋與離線搜尋之排程結果，說明即時搜尋之可行性。

對於以 Petri Nets 分析集束型半導體製程設備排程問題而言，由於延展樹的大小會以指數的模式展開，因此必須藉由效能良好的搜尋演算法進行搜尋，才能快速地（或在合理的時間內）獲得結果。啟發式搜尋演算法 (Heuristic Search) [6]，其為一種透過縮小搜尋空間來增加搜尋速度的演算法，相較於不縮減空間直接搜尋而言[4]，此種搜尋法已可將搜尋的時間控制在合理的範圍，如 Shih 與 Sekiguchi [7]所使用的橫樑搜尋法 (Beam Search)、Lloyd 等人 [8]所使用的分支與邊界搜尋法 (Branch & Bound Search) 及 Lee[2]與 Yim[9]等人所使用的 A\*演算法，此外還有將上述搜尋演算法改良的論述，如 Sun 等人[10]使用限制回溯 (backtracking) 能力的 A\*演算法，使搜尋狀態空間變得更小。Moro, A.R. 等人[11]利用系統性的方法來避免不可能為最佳的路徑產生，減小搜尋空間，以及利用定義有限的搜尋視窗 (search window) 增加搜尋到最佳路徑的機率等。

啟發式搜尋演算法雖然已經大幅減少搜尋所需的時間，不過當搜尋狀態空間會隨時間改變或狀態空間過於巨大時，此種搜尋方式將仍然無法應付。即時啟發式搜尋法 (Real-Time Heuristic Search) [12]提供了一個解決上述問題的搜尋方法，此方法將雙人遊戲 (Two-player Game) 的假設結合到單一代理人啟發式搜尋問題 (Single-agent Heuristic Search Problems) 上，使得不需等到搜尋完全結束，即可先取得部分搜尋結果。

此外，對於利用搜尋延展樹獲得排程問題之另一重點為系統的自動建模。在[4]當中，以由底而上的方式建立 200mm 集束型半導體製程設備的 Petri Nets 模型，將其硬體元件之單元 Petri Nets 數學化後，依據設備目前之硬體配置排列組合獲得全系統之轉換矩陣，其後並利用轉換矩陣找出其延展樹 (reachability tree)，在此延展樹當中，每一個路徑即為一動作程序，當獲得集束型半導體製程設備所有的動作程序後，將每一種可達最終狀態之路徑所花費的時間進行比較，最後透過搜尋的方式找出花費時間最短的路徑，得到該硬體配置下的最佳排程。[2][5]亦提供一套建模方式，與[4][13]不同的是其採用 P-Timed Petri Nets，將動作當作一個模態 (place)，解決了採用 T-Timed-Petri Nets 在並行事件的情況下因多個態變同時被初始化而所產生的死結問題。分析集束型製程設備的動作並尋找排程，除了以上所述的 Petri Nets 方式以外，另外亦可以用時序圖 (time diagram) 方式，如[1]之方法。

### 三、研究方法及成果

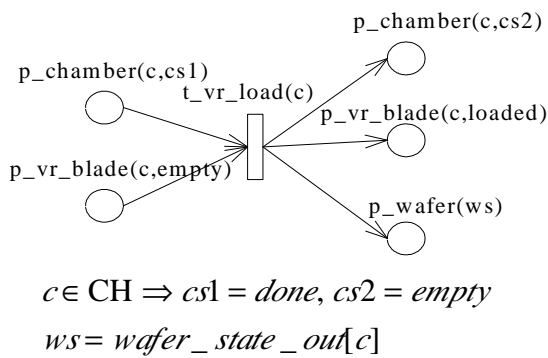
#### 3.1 集束型製程設備之單元模型與自動建模

為了要將排程搜尋與控制器結合，因此必須將集束型製程設備的 Petri Nets 建模過程自動化，如此不但可依據控制器設定之硬體配置及晶圓處理程序動態來產生該設定之 Petri Nets 模型，亦可以減少因人為建模所產生的錯誤及耗時的缺點。在[4]當

中，以 T-Timed Petri Net 發展了一套 200mm 集束型半導體製程設備的自動建模方法，由於其相較於[5] 以 P-Timed Petri Net 建模方式簡單且易於轉成程式碼進行搜尋，因此不採用[5]之模型，而直接修改[4]之方法來找出 300mm 集束型製程設備自動建模方式。

以硬體的角度而言，基本上 300mm 比 200mm 集束型製程設備多了前端處理模組，且承載室的部分亦有所不同，此部分的單元 Petri Nets 模型必須建構或修改，才能由[4]獲得 300mm 集束型製程設備之 Petri Nets 模型，以下為建構及修改之結果：

300mm 集束型設備的動作基本上可分為八種，分別為製程腔體處理晶圓、承載室抽真空或破真空、真空機械臂拿取晶圓、真空機械臂置放晶圓、旋轉真空機械臂、大氣機械臂拿取晶圓、大氣機械臂放晶圓及大氣機械臂旋轉。以真空機械臂在製程腔體處 c 抓取晶圓為例，所構成的 Petri Net 如下所示：



在建模程序當中，我們將靜態承載室的狀態分成兩種，第一種狀態 (LLS1) 用來表示承載室內晶圓及承載室的情形，共可分成六種子狀態，此六種子狀態來自於晶圓的承載、處理情形 (processed、unprocessed 及 slot) 三種，和承載室內氣壓狀態 (vacuum 及 atmosphere) 兩種的排列組合。第二種狀態 (LLS2) 則用來表示承載室槽位 (slot) 是否在放滿的狀態下進行回復壓力 (至大氣壓) 或抽氣。舉例來說：假設靜態承載室內部有 4 個槽位，晶圓盒內有 15 片晶圓需要處理，若靜態承載室每次以四片為單位進行抽氣或回復壓力 (至

大氣壓)，則最後一次將僅能填滿三個槽位，因此以第二種狀態來表示此種情況。此外在上述的模型當中，除了設備的動作外，另外加上一個靜態承載室由滿到未滿即進行處理的事件態變，用來表示靜態承載室進行抽氣及回復壓力 (至大氣壓) 之限制條件的改變。

晶圓狀態 WS 配合 p\_wafer (ws) 態變的使用則是用來限制晶圓的處理程序，在此假設製程過程當中，不會有重複使用同一製程腔體的情形發生。對於集束型製程設備而言，晶圓的處理程序通常有三種，分別為並行式、依序式及混合式，透過限制拿或放晶圓時的晶圓狀態，使得所建立的模型其晶圓的處理程序能夠依照我們所要的處理程序來進行，如下例所示：

假設使用一個晶圓盒，兩個靜態承載室、三個製程腔體，其中製程腔體 A 及製程腔體 B 具有相同製程 (第一道製程)，另一個製程腔體 C 則為另一製程 (第二道製程)。晶圓的處理程序為先進行第一道製程之後，再進行第二道製程，其晶圓狀態如下表所示：

	wafer_state_in	wafer_state_out
晶圓盒	4	0
靜態輸入承載室	0	1
靜態輸出承載室	3	4
製程腔體 A	1	2
製程腔體 B	1	2
製程腔體 C	2	3

上表中所表示的 wafer\_state\_in 指的是進入該硬體元件的晶圓狀態，wafer\_state\_out 指的是離開該硬體元件的硬體狀態。透過上述的方法，即可針對我們所要的晶圓處理程序進行建模的工作

以下將實際地利用單元 Petri Nets 來建立一個 300mm 集束型製程設備之全系統 Petri Nets 模型，並找出其轉換矩陣。假設使用一個晶圓盒，兩個靜態承載室及一個製程腔體，總共有 23 片晶圓需要處理，則單元 Petri Nets 模型組合後將得到圖 1 所示之全系統 Petri Nets 模型。

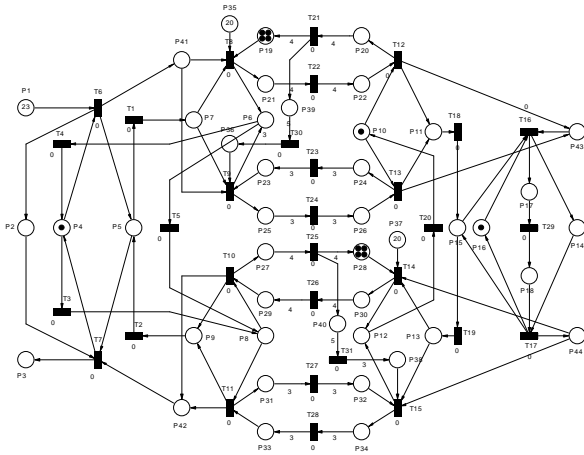


圖 1 300mm 集束型製程設備（單一製程腔體）之 Petri Nets 模型

### 3.2 即時啟發式搜尋法概念簡介

在人工智慧[6][15]的領域，許多問題的解答必須仰賴有系統的試誤法（trial and error）來解決。啟發式搜尋（Heuristic Search）即是一種透過對狀態空間有效地進行搜尋來獲得解答的基本方法。它透過啟發函數估算由系統所展開的狀態空間中，目前狀態相對於目標狀態的成本（cost），利用估算的結果來判斷搜尋的路徑。啟發式搜尋法通常運用在兩種問題上，分別為(1)單一代理人問題（Single-agent problem）及(2)雙打遊戲（Two-player game）。所謂的單一代理人問題指的是由單一主體主導系統的狀態變化，系統希望快速地由初始狀態演化到目標狀態所產生的問題，例如數字拼盤遊戲；而雙打遊戲指的是系統的狀態變化由不同主體交互主導，系統希望快速地找到由初始狀態演化到目標狀態（使某一方勝利）的路徑，例如象棋、西洋棋等遊戲。對於集束型半導體製程設備的排程搜尋問題而言，即屬於單一代理人問題。

上述之兩種問題在啟發式搜尋法的應用上各有其優缺點，以單一代理人問題而言，由於其狀態改變僅由單一主體主導，透過啟發式搜尋法（如 A\*）的搜尋，若啟發函數的估計並沒有超出真實解的值，則 A\* 將可保證找到的解為最佳解。不過在最佳解找到以前，其搜尋結果將無任何意義。也因為如此，當搜尋空間變的極大時

，解決單一代理人問題所需耗費的時間將變得非常冗長且不合理；而對於雙打遊戲而言，由於其系統狀態之演變由不同的主體交互主導，因此其對於啟發式搜尋演算法的應用與單一代理人問題亦有所不同，其強調在搜尋結束之前即可應用其搜尋結果，因此必須透過限定搜尋範圍，在範圍之內尋找較佳的解，此與單一代理人問題透過整個狀態空間的搜尋來獲得最佳解有所不同。

即時啟發式搜尋法的目的即是希望將雙打遊戲的特性應用在單一代理人問題的啟發式搜尋上。即限定搜尋範圍、在限定的時間之內獲得部分搜尋結果之單一代理人啟發式搜尋法，使得當單一代理人問題不必等待搜尋完畢即可即時獲得搜尋結果。如此即可解決當搜尋空間變得極大時獲得解的時間變得不合理之情形。

### 3.3 RTA\* 搜尋演算法

如[12]所提出的 RTA\*（Real-Time A\*）搜尋演算法即為一種即時的啟發式搜尋法，它的特色為解決了因即時決策錯誤所造成的狀態演變路徑偏離問題，透過 RTA\* 所提供的機制將可保證即時搜尋最後能到達目標狀態，其主要概念為當由目前狀態藉由回到之前狀態到達最終狀態所需之估計成本小於由目前狀態繼續向前到達最終狀態之估計成本時，即返回之前之狀態。以下為其演算步驟：

對於目前之狀態  $x$

步驟 1) 將每個與  $x$  相鄰的狀態  $x'$  之  $h(x')$  計算出來

步驟 2) 將目前狀態  $x$  之  $h(x)$  更改為  $\min_{x'} \{h(x') + g(x')\}$

步驟 3) 變換目前之狀態到具有最小之  $h(x')$  之相鄰狀態

其中  $h(s)$  為狀態  $s$  到目標狀態所花費之成本估計值， $g(s)$  為目前狀態到狀態  $s$  所花費之成本。

我們可以透過限定  $h$  的估計範圍來控制上述步驟執行一次的時間，如此達成即時獲得搜尋結果的功能。此外，當搜尋的路徑發生偏離時，由於估計的範圍固定，

RTA\*演算法不斷透過新接觸到的狀態空間的資訊修正搜尋的路徑，最終仍然能回到到達目標狀態的路徑上。以下用一個簡單的例子來說明 RTA\*演算法的作用：

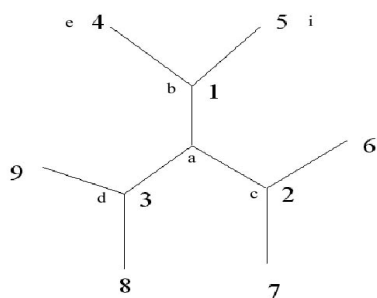


圖 2 RTA\* 演算法範例

如上圖所示，假設此為一狀態空間，初始狀態為節點 a，所有的邊之成本為 1，而圖上每一節點旁邊之數值為啟發式估計所得之  $h$  值。首先由初始狀態節點 a 的鄰近節點 b、c 及 d 開始產生並計算該節點之(到目標節點之)總成本  $f$  值，分別如下：  
 $f(b) = g(b) + h(b) = 1 + 1 = 2$ ， $f(c) = g(c) + h(c) = 1 + 2 = 3$ ， $f(d) = g(d) + h(d) = 1 + 3 = 4$ 。

由以上可知  $f(b)$  最小，因此將目前狀態由節點 a 變換到節點 b，並讓節點 a 之  $h(a)$  值等於  $f(c)$  (次小)。其次再將節點 b 的鄰近節點 e、i 產生並計算其總成本值即  $f(e) = g(e) + h(e) = 1 + 4 = 5$ ， $f(i) = g(i) + h(i) = 1 + 5 = 6$ ，利用之前所得之  $h(a)$  值計算  $f(a) = g(a) + h(a) = 1 + 3 = 4$ ，此時目前狀態再度回到節點 a，而節點 b 之  $h(b)$  值成為  $f(e) = 5$ ，依此步驟繼續下去不斷地在同一區域進行搜尋，最終將可到達其最終狀態。由此可看出當使用 RTA\*演算法時，每當移動一步，便從狀態空間取得更多的資訊，使修正程序不會無止境的繼續下去，當獲得足夠多的資訊時，便可到達目標狀態。

在 RTA\*當中計算節點 n 的成本方式與 A\*相同，皆為  $f(n) = g(n) + h(n)$ ，不同的是 RTA\*所用的  $g(n)$  並不是相對於初始狀態來估算，而是相對於目前狀態，因此其所展開的狀態空間並不需如同 A\*隨深度呈指數擴大。

綜合上述，當將此法應用到半導體集束型製程設備控制器的排程搜尋當中，由

於我們將機台之動作當作一個狀態來處理，因此可以利用進行該動作之所需時間來當作由目前狀態到該狀態之所需成本，把進行該動作到該批晶圓全部完成(即排程結束)所需之時間當作該狀態到最終狀態所需之成本來進行估計，並設定適當的搜尋範圍，此搜尋範圍的大小需在移動每步所需之計算量與完成此排程所需之總移動步數之間進行取捨。當搜尋範圍愈大時，則移動每步的計算量將會相對增加，但卻使完成此排程的總步數減少；反之，當搜尋範圍減小時，移動每步的計算量將會減少，而完成此排程的總步數會增加。

### 3.4 200 mm 集束型製程設備排程之搜尋效能分析與比較

在此節當中，我們將利用[4][16][19]所提出的排程之穩態晶圓處理時間與本計畫利用即時搜尋伸展樹所得排程之穩態晶圓處理時間相比較，來驗證使用本計畫方法所得結果之正確性與效果。其中[4]之排程是以暴力搜尋伸展樹的方式獲得，[19]是以時序圖的方式推得，[16]則依據[19]的方法將其推廣，獲得在不同硬體型態及配置下的排程。以下先針對這幾篇論文所推導而得的晶圓生產週期公式(Fundamental Period)進行介紹，之後將以其來進行比較。

公式內所用之符號說明

L：卡匣承載室數

N：製程腔體數

$P_i$ ：製程腔體  $i$  處理時間(例： $P_1$ ：製程腔體 1 處理時間)

$P_{\max}$ ：所有製程腔體中之最長處理時間  
 ( $P_{\max} = \max_{i=1 \sim N}(P_i)$ )。

$P_{id}$ ：並行處理時，全部製程腔體均相同，處理時間皆為  $P_{id}$ 。

$R_{load}$ ：機械臂拖盤上有晶圓時，在兩製程腔體間移動的時間

$R_{unload}$ ：機械臂拖盤上無晶圓時，在兩製程腔體間移動時間

$T_{unload}$ ：機械臂放晶圓至製程腔體之時間

$T_{load}$ ：機械臂從製程腔體拿晶圓之時間



T：在[16][19]中時序圖推導所定義。定義為當  $R_{load} = R_{unload}$  且  $T_{unload} = T_{load}$

$$T = T_{load} + R_{load} = T_{load} + R_{unload} = T_{unload} + R_{load} = T_{unload} + R_{unload}$$

FP：晶圓生產週期（Fundamental Period，或稱基礎週期）

M：晶圓生產週期中，處理之晶圓數。例如兩製程腔體，依序處理時  $M=1$ ，並行處理時  $M=2$ 。

TP：晶圓生產週期內，平均一晶圓生產時間。

$$TP = FP/M \quad (1)$$

由時序圖[16]推導出之穩態公式：

$$FP = 2(N+1)T \quad \text{when } P_{max} \leq T(2N-1.5) \quad (2)$$

由時序圖[19]推導出之穩態公式：

$$FP = 2(N+1)T \quad \text{when } P_{max} < T(2N-1) \quad (3)$$

單托盤機械臂，並行處理：

由時序圖[16]推導出之穩態公式：

$$FP = 4NT \quad \text{when } P_{id} \leq (4N-3.5)T \quad (4)$$

由時序圖[19]推導出之穩態公式：

$$FP = P_{id} + 3.5T \quad \text{when } P_{id} > (4N-3.5)T \quad (5)$$

以下將以上述的穩態公式及[4]之論程式實際應用一些例子來驗證及比較本計畫所做出的排程結果。

#### ■ 比較案例一：

硬體設定： $L = 1$ ， $N = 4$ ， $P_1 = P_{max}$ ， $P_2 = P_3 = P_4 = 10 \text{ sec}$ ， $R_{load} = R_{unload} = 5 \text{ sec}$ ， $T_{load} = T_{unload} = 5 \text{ sec}$ ，單拖盤機械臂，依序處理：

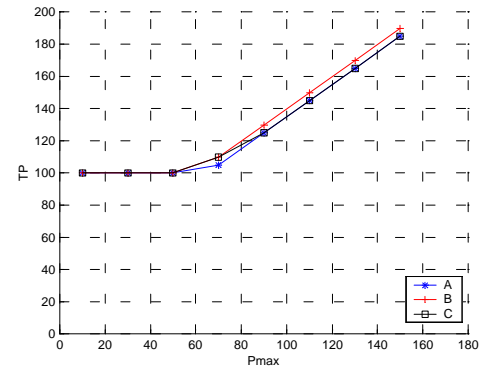


圖 3 單拖盤機械臂、四製程腔體、依序處理排程效能比較圖

如圖所示，其中線段 A 為時序圖[16]及[4]論程式所得排程之結果，線段 B 為時序圖[19]所推得之排程結果，而線段 C 是本計畫利用即時搜尋所得排程之結果。從上圖可以看出，利用本計畫所得的排程除了在轉折點附近以外，其餘皆與[4]及[16]所得相同。這是由於本計畫所使用的搜尋排程方式並非如同[4]所用之方式為暴力對所有可能的排程進行搜尋比較，因此在某些情形下所得的排程結果並非為最佳。此外線段 A 之排程為預期式排程、線段 B 之排程為簡單式排程，而本計畫之排程並不一定為預期式排程或簡單式排程。

#### ■ 比較案例二：

硬體設定： $L = 1$ ， $N = 2$ ， $P_{id} = P_{max}$ ， $R_{load} = R_{unload} = 5 \text{ sec}$ ， $T_{load} = T_{unload} = 5 \text{ sec}$ ，單拖盤機械臂，並行處理：

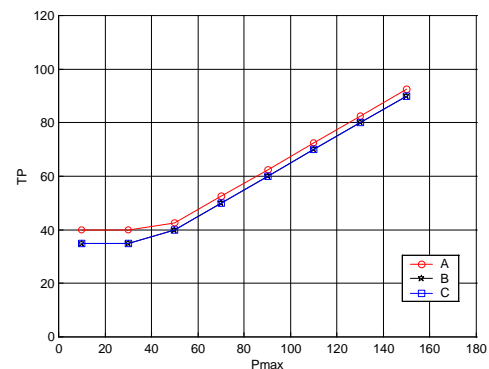


圖 4 單拖盤機械臂、兩製程腔體、並行處理排程效能比較圖

如上圖所示，其中線段 A 為利用[16]及[19]之穩態公式求得，線段 B 為利用[4]之論程式所得之結果，線段 C 為本計畫

所得排程之結果，由於本計畫程式所獲得之排程為預期式排程，因此效能與同為預期式排程之線段 B 效能相同。而線段 A 之排程因為簡單式，所以其排程之效能較差。

### 3.5 搜尋效能之討論

本計畫以 P4-1.4G、128MBRAM 之電腦硬體規格來實際測試以 Java 撰寫而成之搜尋程式的效能。測試的方法為以同樣的集束型製程設備設備機台硬體設定，處理不同片數的晶圓，觀察搜尋每一步動作所需之平均時間。以下假設小中取小之搜尋深度為 6 時，其所得之結果如下：

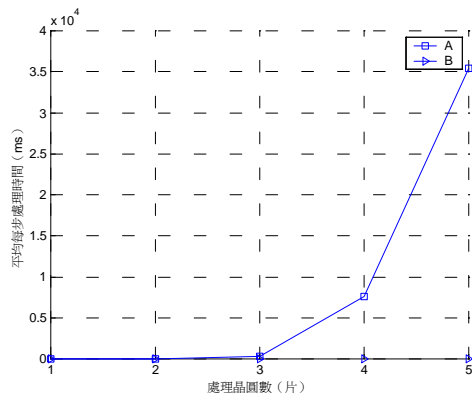


圖 5 處理晶圓總數與每步搜尋時間關係圖

圖 5 所表示的為處理晶圓總數與平均每步搜尋時間的關係。其中線段 A 為[4] 論程式所測得之結果，線段 B 為本計畫程式測得之結果。當處理晶圓總數愈多時，其處理的總步數愈多。由上圖可看出，利用暴力搜尋法，其平均每步搜尋時間會隨著晶圓處理總數增加而增加，而使用即時搜尋（即本計畫）並不會有此現象，其平均每步搜尋時間與晶圓處理總數無關。

### 四、結論與討論

在本計畫當中，首先以[4]之方法為基礎，建立 300mm 集束型製程設備的單元 Petri Nets 模型，並以即時啟發式搜尋法 RTA\* 來對 Petri Nets 模型所產生的延展樹進行搜尋，其所獲得之排程能以即時的方式求出，不需等待全部搜尋完成即可獲得結果，利用此法配合半導體集束型製程設備 Petri

Nets 自動建模，便可將此種產生排程的方法結合到控制器的設計當中。

未來工作

1. 在本計畫當中僅就排程器的部分作了研討，還未實際將該排程器與真實之半導體集束型製程控制器相結合，因此未來可將該排程器配合 SEMI 標準所規範的集束型製程設備控制架構，實際設計出一個高效能的控制器。
2. 本計畫所使用的 Petri Nets 模型與真實機台情形仍有諸多差異，例如機台不正常中斷、控制訊息延遲及製程延遲處理 7 等真實世界會發生的情形在本計畫所用的 Petri Nets 模型並沒有作相對的處理，因此若未來要與控制器相結合，這部分的模擬將是非常重要的。
3. 在 RTA\* 部分，對於所使用的啟發式函數而言，若該函數估計的愈準確其搜尋的效能愈高。在本計畫當中，本計畫僅討論了一種取法，未來可對不同的啟發式函數取法進行研究、比較，取得使搜尋效率更高的啟發式函數。
4. RTA\* 僅是即時搜尋法中的一種，還有其他如 LRTA\* 等搜尋法，未來可將各種不同的搜尋應用到排程搜尋的問題上，研究各種不同的方法對於排程搜尋問題的適用性及比較其效能。

### 五、參考文獻

- [1] Shadi Rostami, Babak Hamidzadeh, Dan Camporese, "An Optimal Scheduling Technique for Dual-arm Robots in Cluster tools with Residency Constrains", *Proceedings of the 39<sup>th</sup> IEEE Conference on Decision and Control*, Sydney, Australia December, 2000.
- [2] D. Y. Lee, Frank DiCesare, "Scheduling Flexible Manufacturing Systems Using Petri Nets and Heuristic Search", *IEEE Transactions on Robotics and Automation*, Vol. 10, No. 2, pp. 123-132, April. 1994.
- [3] M. C. Zhou, K. Venkatesh, *Modeling, Simulation and Control of Flexible*



- Manufacturing System*, World Scientific, 1999.
- [4] 王喬,多腔式製程設備之建模及自動化排程,國立交通大學機械工程研究所碩士論文,民國 89 年。
- [5] 鄭隆傑,300mm 型多腔式製程設備之建模及自動化排程,國立交通大學機械工程研究所碩士論文,民國 90 年。
- [6] J. Pearl, *Heuristic: Intelligent Search Strategies for Computer Problem Solving*, MA, Addison-Wisely, 1984.
- [7] Shih H. M. and Sekiguchi T, "A Timed Petri Nets and beam search based on-line FMS scheduling system with routing flexibility", *Processdings of the 1991 IEEE International Conference on Robotics and Automation*. Sacramento California, pp. 2548-2553, 1991
- [8] Lloyd, S., Yu, H. and Konstants, N. "FMS scheduling Using Petri Nets Modeling and Brach & Bound Search. Proc", *IEEE International Symposium on Assembly and Task Planning, Pittsburgh*, pp. 141-146, 1995
- [9] Yim, S.J. and Lee, D.Y., "Multiple Objective Scheduling for Flexible Manufacturing Systems Using Petri Nets and Heuristic Search", *IEEE Int. Conf. On Systems , Man, And Cybernetics. Information Intelligence and Systems*, pp. 2984-2989, 1996
- [10] Sun, T., Cheng, C. and Fu, L: "A Petri Nets Based Approach to Modeling and Scheduling for an FMS and a Case Study" *IEEE Transactions on Industrial Electronics*, Vol 41, pp.593-601, 1994
- [11] Moro, A.R.; Yu, H.; Kelleher G, "Advanced scheduling methodologies for flexible manufacturing systems using Petri Nets and heuristic search", *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on , Vol3 , 2000*
- [12] Richard E. Korf, "Real-Time Heuristic Search", *Artificial Intelligence*, Vol. 42, pp.189-211, 1990.
- [13] N. Nilsson, *Principle of Artificial Intelligence*. Palo Alto, CA: Tioga, 1980.
- [14] Murata T., "Petri Nets: Properties, Analysis and Application", *Proceedings of The IEEE*, Vol.77, No.4, April 1989.
- [15] W. M. Zuberek, " Timed Petri Nets Models of Cluster Tools", *Systems, Man, and Cybernetics, 2000 IEEE International Conference on , Vol. 4*, pp. 3063 -3068, 2000 .
- [16] 李金德,集束型半導體生產設備模式的建立及性能分析,私立中原大學機械工程研究所碩士論文,民國 88 年。
- [17] Books of SEMI Standards, *SEMI*, 1999.
- [18] 胡竹生、尹燕陶,即時多工核心程式設計,全華科技圖書,民國 86 年
- [19] T.L. Perkinson, P.K. McLarty, R.S. Gyurcsik, "Single-Wafer Cluster Tool Performance: An Analysis of Throughput" , *IEEE Trans. Semiconduct. Manufact.*, Vol7, pp.369-373, Aug.1994.