

創新排列與基因演算法於電腦輔助自動排版系統之研究

賴元隆¹ 洪瑞斌²

¹大葉大學機械與自動化工程學系

51591 彰化縣大村鄉山腳路 112 號

²勤益科技大學自動化工程學系

411 台中縣太平市中山路一段 215 巷 35 號

摘要

一個排版問題主要由兩個過程來完成，一是零件排列次序 (permutation) 之決定，另一是排放零件至基板的排置 (placement) 方法；這兩個過程就主宰了一次排版結果的優劣。如果次序對了，排置方法也滿足需求，這個時候我們就可以找到正解或最佳解；但是往往無法得知正確的排列次序，因此需要利用搜尋法則來尋找最佳解。但是如果排置的方法不佳，即使排列的次序對了，排版出來的結果也不盡完善。本研究針對具間隙零件的排版需求，開發創新的排置演算法 (placement algorithm)，結合基因演算法 (genetic algorithm) 之特點而來進行排版系統之最佳決策，找出最好的排版效果。二維排版問題最佳化的目的乃將特定數量的相同或不同工件 (piece) 排列於基材 (sheet) 上，期使基材使用率最高或浪費最少為目標，如此基板的使用率最高以減少物料成本。排版問題依據不同的限制與目的存在於不同的產業中，如半導體印刷電路板、複合材料、高分子板材、玻璃切割、實木成型、家具裝潢、鞋業生產、服裝裁剪、皮革下料、建築包工、造船鋼板、車輛和發電設備生產中都存在大量的下料或精密排版切割問題。

關鍵詞：自動排版，排置演算，基因演算

New Placement and Genetic Algorithms for a Computer-Aided Nesting System

YUAN-LUNG LAI¹ and JUI-PIN HUNG²

¹*Department of Mechanical and Automation Engineering, Da-Yeh University
No. 112, Shanjiao Rd., Dacun, Changhua, Taiwan 51591, R.O.C.*

²*Department of Automation Engineering, National Chin-Yi University of Technology
35, Lane.215, Sec. 1, Chung-Shan Rd., Taiping, Taiwan 411, R.O.C*

ABSTRACT

A nesting problem can be solved by two processes, one being determining the permutation for sequences of pieces; the other, applying a placement algorithm for positioning all pieces on the sheet. These two important processes dominate the solutions for the nesting problem. If the sequence of permutation is available and the placement rule also meets the packing demand, then the exact or optimum solutions can be found. However, whether the sequence of permutation is adequate

remains unknown. Even if the sequence is appropriate for the permutation, without a suitable placement algorithm the final results are also unacceptable. In this research, we developed a new placement algorithm for pieces with gaps in automatic nesting operations, by which a genetic algorithm approach to optimal determination of hollow nesting systems was adopted. The objective of the two-dimensional optimal nesting problem is to determine an effective usage of a stock sheet under different configurations of the pieces. Such determination ensures that the material utilization rate is the highest, thereby reducing the cost of materials. Nesting problems are encountered in many different industries, incorporating different constraints and objectives. E.g., the PCB, wood, polymer-plate, glass, paper, metal, textile and leather industries are substantially concerned with the cutting of arbitrary-shaped figures.

Key Words: automatic nesting, placement algorithm, genetic algorithm

一、前言

排版問題是許多行業所遭遇到的問題，如紡織業、出版業、製鞋業、鈹金業、傢俱業、建築業與 VLSI (very large scale integrated circuit) 的平面規劃等。一個排版問題主要由兩個過程來完成，一是零件排列次序 (permutation) 之決定，另一是排放零件至基板的排置 (placement) 方法；這兩個過程就主宰了一次排版結果的優劣。本文針對刀具加工排版需求而提出新演算法，對具間隙零件的排版需求，開發創新的排置演算法 (placement algorithm)，結合基因演算法 (genetic algorithm) 之特點而來進行排版系統之最佳決策，找出最好的排版效果。最佳化排版的目的乃在將相同或不同型零件物體排列於原料基板上，並使其利用率最高，廢料率最低。而上述各行業排版作業之零件與基材不論屬規則型或不規則型，都需要一套快速而有效之自動排版系統來取代人工作業以降低成本，縮短產品設計與加工之週期時間。基本上，最佳化排版是要搜尋一組每個零件在原料基板上的位置與旋轉角度，讓基板使用率最高，因此，許多搜尋的方法已被提出來解決此一問題，如模擬退火法 (simulating annealing, SA)、塔布搜尋法 (tabu search, TS) 與基因演算法 (genetic algorithm, GA)。前兩種搜尋的方法基本上屬於局部搜尋，藉由定義解答空間 (solution space) 及鄰近區域 (neighborhood) 之結構，允許對問題解答無改善性 (non-improving) 的變化，定義成本函數 (cost function) 為搜尋的標準，當有一組解答之成本函數低於目前之成本函數則該組解答即取代為目前之最佳解。使用 SA 或 TS 方法作為搜尋的技術時，其解答空間，鄰近區域結構及其成本函數必須適當定義，才能使其成功的機率加大。由於搜尋的方法較耗時，而且並不一定能保證當最佳解存在時能找到其解

或逼近其解，許多學者即提出啟發式經驗法則 (heuristics) 或者藉由這些經驗法則與之前搜尋方法結合，期能減少不必要的搜尋，增加排版的效率。

技術持續不斷地進步，產業要面對競爭也需順勢隨潮流不斷的升級，從以往的勞力密集產業，逐漸轉型為自動化、智慧型的生產方式，因此產業界莫不一直在思考如何在這競爭的環境中精簡人力、提高效率及降低生產成本，以獲致更多的利潤和提高產業的核心競爭力。對某些產業來說，切割排版問題是製造生產的關鍵問題，此課題佔成本支出重要的一部份，如半導體印刷電路板切割、複合材料、高分子板材、玻璃切割、家具裝潢、鞋業生產、服裝裁剪、皮革下料、建築包工、造船鋼板、車輛和發電設備生產中都存在大量的下料或精密排版問題。所謂的切割排版問題，是在一已知大小的原物料版材上，將要排列之特定數量的幾何圖形擺放上去，調整擺放位置，使得切割後，原物料浪費最少，即板材的使用率最高為目標。而目前仍有些業界仍是利用人力來完成切割排版等相關作業，以鈹金業來說，他們傳統的排版過程仍是以有經驗的相關人員，依照經驗及目視法則來進行，將要切割的形狀試著排出一較令人滿意的排版結果，由於此法麻煩、缺乏科學分析，不但耗費時間且缺乏效率，可能造成原物料使用率低的情況，但是最大的缺點是極可能造成廢品及浪費的情形發生，所以發展出一套有效的排版系統，利用電腦代替人工來進行自動排版，實是一個非常迫切的需求，尤其對於單位價格較高之原料，如特殊金屬、塑材及皮革等，更顯得重要，除可縮短產品設計與加工之週期時間，提昇產能降低成本，增加專業獲利能力，提高產業競爭力。

二、文獻回顧

排版 (nesting) 是常應用在裁切 (cutting) 與堆疊 (packing) 上的一個問題，許久以來一直有文獻在探討這個很經典 (classical) 的問題，這樣的一個 NP-Hard 的問題 (non-deterministic polynomial) [28, 29, 35]，此類問題至今仍未找到一個多項式複雜度的決定性演算法，且一般相信沒有多項式複雜度的決定性演算法存在。排版問題在許多的產業都有重要性，我們可由最近的文獻資料可看出，這個問題一直都是許多研究者想要有所突破的技術 [5, 6, 8, 19, 20, 22, 31, 40, 42, 43]。我們作一個簡單說明，如果以七個零件的排版問題為例，全部嘗試的數目為 5,040 次，我們有萬分之二不到的機率會猜到可能的正解 (假如存在的話)；但如果以十七個零件的排版問題為例，全部嘗試的數目高達 3,556,87,428,096,000 次，這是個大數目，而且簡直是個天文數字，很難想像如果要嘗試這種列舉法要消耗多少運算資源。相同的狀況，如果我們利用基因演算法來搜尋最佳排列組合，例如使用族群數 50，演算 20 個世代，最多會嘗試 1,000 次的評估，我們可以在這 1,000 次的嘗試中找出相對的最佳解。就因為排版問題的求解是個指數型困難的問題，以致有許多研究嘗試以新的演算方式來改善尋找最佳解的搜尋過程 [7, 9, 10, 13, 16, 17, 24, 30, 33, 39]。

排版問題是個複雜的問題，大致可以由三個面向來探討，一是零件 (piece) 與基板 (sheet) 的幾何圖形表達方式，其次是零件排進基板的方法，另一個則是搜尋最佳安排的結果。以圖形表達而言，可分為幾個方面，一為從矩形基板中切割出矩形外形的工件，一種為從矩形基板中切割出不規則外形的工件，另一種為從不規則基板中切割出不規則外形的工件；由圖 1 中可看出基板的可選擇類型。較早研究方法 [16, 17] 試圖使用數學方程式解決排版問題，例如使用線性規劃或是動態規劃，他們限制在處理的矩形或較簡單的多邊形工件。當遇到較複雜的幾何圖形，上述的方法便無法使用。在許多產業製造上，不規則幾何的工件排版越來越重要，也有一些研究致力於發展用簡易的方式，表達不規則幾何的工件與基板，例如 Freeman 與 Shapira [15] 在研究中提出利用最小矩形來表達工件。Dorit 與 Ben-Bassa [11] 將工件的輪廓使用六邊形的簡化方式來表達，Ma, Cannon 與 Kumara [32] 使用神經網路產生不同尺寸的 2D 圓形或 3D 圓球來表達不規則幾何圖形，使用在機械手臂移動碰撞問題上。Tay 等 [41] 指出在處理零件排置時，零件間邊界至少

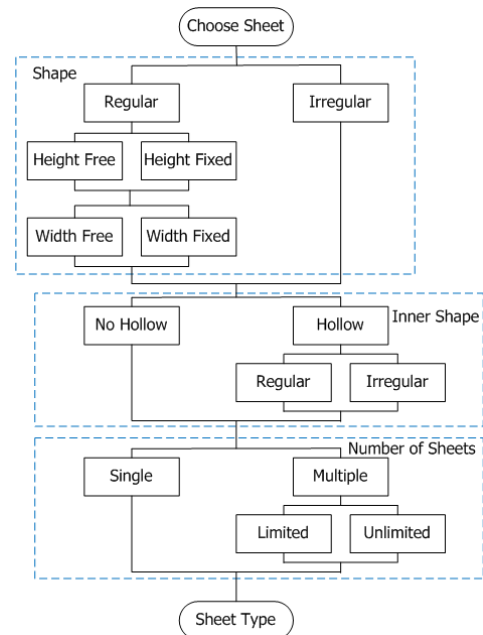


圖 1. 基板的選擇類別

有一點會接觸，因此利用零件沿著已安排件邊界移動，並旋轉零件試著找尋安排位置及角度。此方法對複雜幾何的零件與基板，可成功減少安排可行位置的時間。不過，本方法問題在於安排時，以零件座標為主，有時很難符合基板座標。Ramesh 與 Ramesh [37, 38] 利用分離式資料結構來表達零件及基板的幾何圖形。零件及基板使用固定單位相同的方格子表示，使用分離式資料結構來表達工件及基板，就某種意義來說，可以快速將複雜幾何圖形化為簡單資料結構。然而，限制固定單位相同的方格子，會造成安排時，失去較佳的工件位置及角度。例如，基板只要佔到某方格子的一小部份，該某方格子就不可安排其它工件，將造成浪費。此外，若處理大型工件，方格子分割將會變成相當大的數量，可能也會導致花費相當長的時間才可處理完成排版問題。

除了不規則基板與工件的幾何圖形問題外，一些啟發式的排版方法被發展出來，有些研究是利用人的經驗方式，調整改變最後結果。Prasad 等 [36] 提出一種滑動工件的方法，其方法主要解決在安排工件時，若產生工件與工件或工件與基板間的空隙，快速調整工件位置，以消除不必要的空隙。Gomes 與 Oliveira [18] 提出工件兩兩交換的方法，此方法可以在排版時，填補不必要的中間空洞且可以改善排版效果及效率。一般而言，這些方法有時只適用於特殊的案例。近年來，一些不同的搜尋演算法則被提出，應用於工件安排

基板位置的搜尋，例如基因演算法 [14, 21, 23, 26]、模擬退火法 [12, 22, 25]、塔布搜尋法 [1, 4, 27]、進化演算法 (naive evolution) [22] 與群體啟發法 (population heuristic) [2, 3, 34]，這些方法可決定安排工件或基板的最理想順序及最佳位置，更提高排版工作的效率。例如，Dagli 和 Tatoglu [10] 提出一套方法將排版問題分為兩個階段，在第一階段使用數學規劃來決定起始的排版樣式，然後在第二階段針對第一階段的起始樣式來進行經驗法則之排版，這些方法的共同的限制乃其無法對含曲線之不規則型材適當逼近，及其對好的排版配置將不會是正交 (orthogonal) 的假設。Jakobs [23] 使用基因演算法，以基板左下方為安排目標位置，求出最佳安排工件順序及最佳工件位置與旋轉角度。Fischer 與 Dagli [14] 也使用基因演算法，以工件排序進而減少安排工件順序數目。Bennell 與 Dowland [4] 提出利用塔布搜尋法，避開不必要搜尋次數。雖然一個複雜的排版問題，可能會有無限多解，但是上述的研究已經可以發現較好的排版結果。不論如何，搜尋演算法則通常須要高計算能力的設備，來決定安排可行的位置，換言之，所有的工件，盡可能安排於基板上且工件間無重疊。

實際上，大部分零件加工業者所能使用到的排版系統都需要面對零件空隙間的問題，但是在國內外的文獻上卻是少有研究在空隙問題上著墨，而把重點放在搜尋最佳位置的排列次序。因為在搜尋位置的排列次序上已經夠複雜了，若是再加上圖形物件還要處理空隙條件的複雜問題，困難度再增加，運算處理的時間相對地也會再往上提高。一般而言，排版的方法可以區分成兩種：自動排版法與手動排版法，在手動排版方面，國內已有幾家軟體廠商發展出電腦輔助工件排版及數值控制加工系統，提供工件排版與 NC 切割的環境。就我們的瞭解，國內、外少有專門的文章專門介紹排版時考慮刀徑所造成的影響，因為切削刀具的圓徑大小將會影響到排版的位置，零件與零件之間要有空間讓切削路徑通過，否則會造成過切的現象，造成廢料的發生。

三、排版方法與流程

(一) 定義分類

首先，我們對排版問題作一定義說明；我們對排版作一定義：「將許多較小元件，以適合的方法及適當的位置排入較大的元件內，以符合最大的操作效益。」文獻中對操作的元件名稱不一；在此，較小的元件我們稱之為零件 (pieces)，

最大的元件我們稱之為基板 (sheets)，小零件可以被排入至大基板內；由圖 1 的基板選擇類別與圖 2 中零件的選擇類別可組合各種不同的排版問題。由大部分的研究文獻中，我們可以將二維矩形排版問題從三大類，六個條件不同的角度來檢視：

1. 條狀堆積 (strip packing problem)：一個基板，多種不同的零件。基板尺寸沒有固定，基板的寬度是一個固定值，但是在高度沒有限制，我們的目標函數是要找出最小的高度值，在這個寬度與高度的範圍內，依照設定的條件要將所有的零件擺放進去，大部分的研究都屬於這類的問題。
2. 面積最小化 (area minimization problem)：一個基板，多種不同的零件。基板尺寸沒有固定，基板的寬度是一個可變值，高度也是一個可變值，我們的目標函數是要找出最小的面積值，在這個寬度與高度的範圍內，依照設定的條件要將所有的零件擺放進去。
3. 二維裝填問題 (2D bin packing problem)：基板數目不設限，多種不同的零件，通常一個基板不敷使用。基板尺寸固定，基板的寬度是一個固定值，基板的高度也是一個固定值，我們的目標函數是要找出最小的基板數目，在這個寬度與高度的範圍內，依照設定的條件要將所有的零件擺放進去。

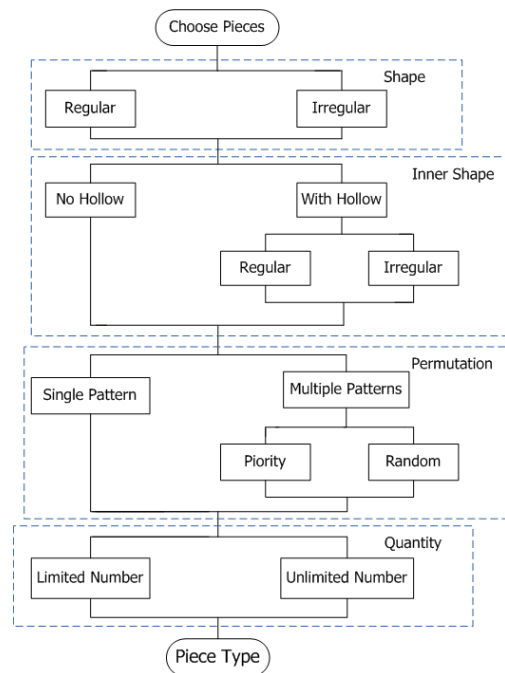


圖 2. 零件的選擇類別

4. 二維裝袋問題 (2D knapsack problem)：一個基板，多種不同的零件。基板尺寸固定，基板的寬度是一個固定值，基板的高度也是一個固定值，我們的目標函數是要找出這些零件排放的最有價值方式，這個價值的評斷可能是基材的使用率，或是零件的優先順序。
5. 二維多量裁切問題 (2D cutting stock problem) 基板數目不設限，多種不同的零件，每種零件有多個數目，而且通常一個基板不敷使用。基板尺寸固定，基板的寬度是一個固定值，基板的高度也是一個固定值，我們的目標函數是要找出最小的基板數目，在這個寬度與高度的範圍內，依照設定的條件要將所有的零件擺放進去。
6. 大量二維裝填問題 (pallet loading problem)：基板數目不設限，一種的零件，零件是大量數目，而且通常一個基板不敷使用。基板尺寸固定，基板的寬度是一個固定值，基板的高度也是一個固定值，我們的目標函數是要找出最小的基板數目，在這個寬度與高度的範圍內，依照設定的條件要將所有的零件擺放進去。

(二) 排版條件

在我們的實務研究中較接近二維裝填問題 (2D bin packing problem)；其影響的條件有：

1. 零件間的距離；
2. 使用者介面需顯示編號；
3. 設定基板之數目；
4. 設定零件之數目；
5. 選擇排置基準位置；
6. 選擇最佳化方式；
7. 零件可旋轉放置否；
8. 零件是否有經驗上優先順序。

切割排版問題是屬於複雜、非線性及變化整數的限制型組合最佳化之問題。這樣的一個非確定性多項式 NP-hard 的問題 (non-deterministic polynomial)，意味此類問題至今仍未找到一個多項式複雜度的決定性演算法，且一般相信沒有多項式複雜度的決定性演算法存在。實際上對板件切割系統之排版而言，欲求得其精確解之唯一方式為完全列舉法 (complete enumeration)，但此法必定可能會消耗很多之電腦計算時間及記憶體，導致求解效率較低而不實用。因此有關自動排版問題，現今許多研究者致力於一些提高求解效率且能使解答品質接近最佳解 (near-optimal solution) 之研究。文獻上可能的方法有，優先順序法 (priority list method)，

此種解法原理簡單且計算迅速，但解答的品質趨於較粗糙，通常較難趨近最佳解。動態規画法 (dynamic programming)，此種解法、計算頗為簡潔、快速且富彈性，但由於需佔用很大的電腦記憶體容量，而且可能會有「維度的詛咒」(curse of dimensionality) 現象，故本法較不適於多物件樣式之解法。整數規画法 (integer programming)，利用作業研究 (operation research) 中的線性規劃 (linear programming) 技巧來求解，但由於此法需作一些線性的假設，而限制解答空間的範圍。分支界限法 (branch-and-bound)，分支界限法能有系統的搜尋有限個可行解 (feasible solution) 之解答空間，利用分支 (branch) 與界限 (bound) 之方式來求得上限 (upper bound) 和下限 (lower bound) 以縮小搜尋範圍，進而求得較佳可行解。但此法在刪除遍離分支的過程中，可能會將真正的最佳解組合狀態丟棄，以致最佳僅求得次佳解。類神經網路 (artificial neural network) 為一種模擬生物神經網路的演算法，籍著學習法則，來修正網路權值及類神經元 (artificial neuron) 界限，以使誤差函數 (error function) 得以最小，亦即使網路的推論運作能夠符合訓練樣本之要求。但此法必需要有大量的學習訓練樣本，和長時間的學習訓練過程；且解的品質受到網路參數的影響頗大。模擬退火法為一種模擬物理學上物質結晶退火程序的一種隨機性最佳化演算法，將問題的解空間表示成粒子的結晶狀態。在物質結晶過程中，如果溫度下降的速度夠慢時，則粒子間活動會達一熱平衡 (thermal equilibrium)，直到冷凍狀態，粒子間有一高密度結晶格結構，使得系統的能量最低，此類比於最佳化問題之最佳解。此法雖可獲得最佳解，但求解的時間較長為其缺點。基因演算法為一種模擬自然界中生物遺傳及進化的行為，將問題的解空間表示成生物學上的染色體 (chromosomes)，依據達爾文「物競天擇，適者生存」的進化論，透過複製 (reproduction)、交配 (crossover) 和突變 (mutation) 的進化機制來繁衍下一代。世代演進的結果，最適者之染色體將存活下來，為此最佳化問題之最佳解。雖然基因演算法具有平行處理之較快速求解能力，但常因編碼和參數設定問題而造成求解誤差，而無法求得真正的最佳解。

我們由定義與之前的描述可知，零件的排置方式 (placement algorithm) 與排列次序最佳化搜尋 (searching algorithm) 是排版問題的兩個最主要的課題；零件排列次序的決定，搭配有效的排置方式，兩個因素的結合就會產生最

佳的一組解，這是個機率的問題；但是如果排列次序對了，沒有排置方式來完成排放的工作，依然無法取的最佳解。但是如果無法找到最佳解，但是我們可以想辦法能找到相對的最佳解，或是接近最佳解的解。本計畫中為了對基板與零件的中空幾何問題作有效的處理，我們在此研究中有三個主要步驟要完成：創新排置方法、目標函數定義與基因演算法三個主要步驟。創新排置方法要能跳脫目前文獻上的方法，另找出一新發法來處理中空問題，而且此方法不會比目前的複雜度高。目標函數的定義關係到目前這組解的品質，現有文獻上的目標函數我們將作修正，以增加實務上的價值。最後在我們嘗試排列的眾多結果中，利用基因演算法來搜尋可能的最佳解。

(三) 目標函數

目標函數的定義要能非常明確地反應出目前解的優劣程度，文獻上常使用的目標函數為常使用在基因演算法中的適應性函數，如式(1)所示，他是裁餘值的倒數，因為裁餘值有可能為零，所以加入一個極小的值以避免奇異性的發生；這方程式表示對每一個群體的染色體，其對應出的適應度值是依據這個適應度函數計算而得。

$$Fitness\ value(\pi) = \frac{1}{(\varepsilon + Trimlossvalue(\pi))} \quad (1)$$

其中 $\varepsilon = 2.2204 \times 10^{-16}$ ，裁餘值如式(2)所示。

$$Trimlossvalue =$$

$$\frac{area\ of\ the\ main\ object - sum\ of\ areas\ of\ placed\ pieces}{area\ of\ the\ main\ object} \quad (2)$$

由上式(2)看來裁餘值必小於 1，而且分子越小代表排入的零件越多，因此裁餘值越小越好，當裁餘值為零時代表零件可完全排滿整個母材。適應性函數值即為裁餘值的倒數，相對地，適應性函數值如果越大表示效果越好。

但此適應性函數有一瑕疵，當兩組排入之零件種類與數目相同時，即使排入次序與位置皆不同，此裁餘值也可相同，如圖 3 所示，這樣無法區分出優劣的比較，或是我們可以視此二者為相同的適應性函數值，這樣的結果在實務上就會有認知上的誤差，因此我們增加一個影響因子：有效影響面積。有效影響面積的定義為被排入之零件的總影響範圍。這個總影響面積越少，代表在相同條件下還可再被利用的面積的機率越大，已排置的零件其緊緻度越高。如上所述，為方便程式規劃，我們定義適應性函數如式(3)，面積影響函數值如式(4)。

$$Fitness\ value = \frac{sum\ of\ areas\ of\ placed\ pieces}{area\ of\ the\ main\ object} \quad (3)$$

$$Effective\ value = \frac{sum\ of\ areas\ of\ placed\ pieces}{total\ effective\ area} \quad (4)$$

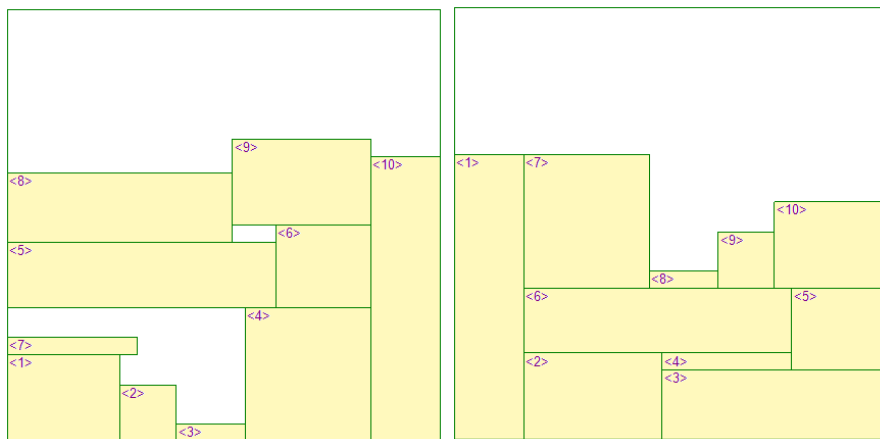


圖 3. 不同的次序排入相同的十個零件

四、創新排置演算法

(一) 排置演算法

文獻上已有許多排置演算法已提出，也有許多文獻探討這些演算法的優缺點，BL (bottom-left) 法為 Jakobs [23] 於 1996 年提出，如圖 4 所示，此方法可解釋為利用滑動的方式來達到定位的決定；例如每個將被排入零件會先在基板的右上方等待，零件優先向下滑動，當遇到阻擋無法向下滑動時，轉而向左移動；當遇到阻擋無法向左移動時，轉而向下滑動；持續這樣的過程一直到既無法再向下滑動，也無法再向左移動時就是這個零件排入的最後位置。IBL (improved bottom-left) 法為 Liu 與 Teng [26] 於 1999 年提出，如圖 5 所示；此方法也可解釋為利用滑動的方式來達到定位的決定；例如每個將被排入零件也會先在基板的右上方等待，零件優先向下滑動，當遇到阻擋無法向下滑動時，轉而向左移動，移動的過程一有向下滑動的機會，立刻轉而向下滑動；當又遇到阻擋無法向下滑動時，轉而向左移動，移動的過程一有向下滑動的機會時，立刻又轉而向下滑動；持續這樣的過程一直到既無法再向下滑動，也無法再向左移動時就是這個零件排入的最後位置。BLF (bottom-left fill) 法為 Hopper 與 Turton [22] 於 2001 年提出，如圖 6 所示；此方法是紀錄每次排入零件後的可被排入點，這些點包括零件的右下角、零件的左上角點、投影在左方的水平高度點與投影在下方的垂直高度點；這些點就成為下一個零件排入的參考點，優先順序為往下優先，若高度一樣再以往左為優先；判斷第一個參考點的空間無法滿足零件的排放時，立刻評估第二個參考點的空間能否滿足零件的排放，若可就定位，不行就往下一個參考點評估。BF (best fit) 法為 Burke [8] 等於 2004 年提出，如圖 7 所示；這個方法在排入一個零件後會利用剩餘寬度空隙，動態判斷下一個優先排入的零件，這個方式常因為要配合剩餘寬度空隙而造成高度的激凸影響，常為了要處理這個現象而多出額外的運算時間。但是這些方法皆無法滿足我們對空隙處理的需求，因此勢必要發展出一個新的排置演算法來滿足空隙處理的演算。我們將採用新的概念演算法，如圖 8 所示，利用面積分割原理，初始可排區域為母代原始基板，當第一個排入的零件確定位置後，基板將被分割成第二步驟的第一代數個較小可排面積，當第二個排入的零件，可排入第一代的可排面積確定位置後，這個被排入面積將被分割成第二代的數個較小可排面積，而且還要與第一代的其他可排面積比對干涉狀況，而歸入在第二代的可排面積內，

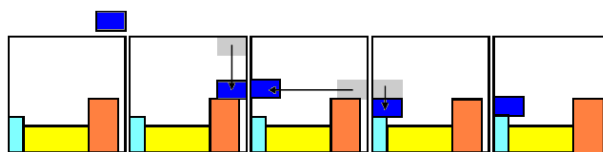


圖 4. BL (bottom-left) 排置演算法則

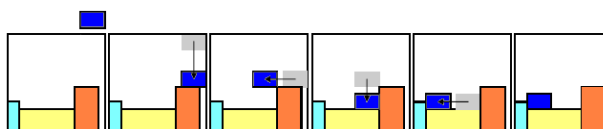


圖 5. IBL (improved bottom-left) 排置演算法則



圖 6. BLF (bottom-left fill) 排置演算法則



圖 7. BF (best-fit) 排置演算法則

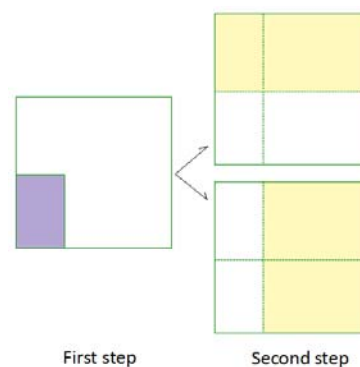


圖 8. AD (area-decomposition) 面積分割排置演算法則

第一代可排面積即失效。依照這個邏輯疊代演算一直到最後一個零件排入為止；這個面積比對分割的干涉情況可有如圖 9 的多種狀況產生，這個作法我們可以馬上得知全部可排的面積，並對每一代的可排面積作排序處理；圖中符號所代表之意義為： p 代表零件， s 代表基材， t 代表上界， b 代表

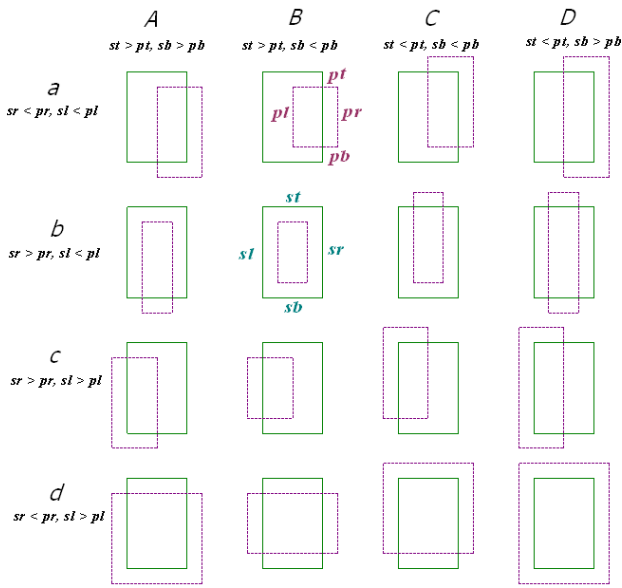


圖 9. 面積分割法的干涉種類

下界， l 代表左界， r 代表右界。本研究所提出來的排置方法與其他種方法的不同處，在於新方法可以有效處理刀具切削所需要的零件間空隙產生的問題，在下一節中我們將針對範例的實驗結果作敘述與討論。

(二) 基因演算法

基因突變和基因交配可產生更適應於環境的後代，經過存優去劣的自然淘汰下，適應性高的基因架構得以保存下來；基因演算法在本質上是一種不倚賴具體問題的直接搜索方法。從許多的文獻中可以發現基因演算法（GA）是非常適用於排程決策分析。GA 是一種多路徑的隨機搜尋最佳化方法，為使用隨機運算來取代確定法則去搜尋合適的解。在搜尋過程中，GA 具有平行搜尋能力，有別於一般傳統的單路徑搜尋方式；致使搜尋效能大大地提昇，且 GA 可以獲得多組解以供使用者選擇。基因演算法是模擬前述生物進化過程的計算模型，它所處理的是染色體（chromosome），或者叫基因行個體（individuals）。一定數量的個體組成了群體（population），也叫集團。群體中個體的數目稱為群體的大小（population size），也叫群體規模。而各個體對環境的適應程度叫做適應度（fitness）。遺傳演算法基本上是一種群體型操作，該操作以群體中的所有個體為對象。選擇（selection）、交配（crossover）和突變（mutation）是基因演算法的三個主要操作算子，它們構成了所謂的基因操作（genetic operation），使演算法具有了其他傳統方法所沒有

的特性。

1. 群體：這是一組問題的可能解，因為群體的大小可隨問題變化，所以通常會多樣測試。
2. 適應性函數：對每一個群體的染色體的適應度值，依據適應性函數計算而得。
3. 選擇：這是從群體中選擇出較適應環境的個體。這些選中的個體用於繁殖下一代。故有時也稱這一操作為再製（reproduction）。
4. 交配：這是在選中用於繁殖下一代的個體中，對兩個不同的個體的相同位置的基因進行交換，從而產生新的個體。
5. 突變：這是在選中的個體中，對個體中的某些基因執行異向轉化，不以編碼互換的方式繁衍，而是以隨機方式抽取染色體中的若干編碼進行改變，便可產生突變。

基因演算法是根據自然生存法則“物競天擇，適者生存”所發展出來的，而基因演算法的基本步驟則是一連串的基因複製、突變、淘汰的進行。基因演算法常用在網路問題或是車輛途程問題搜尋最佳路徑，卻極少用在排版系統中，對於各種搜尋方法而言，基因演算法是兼重效率與效能的搜尋方法，故本研究利用基因演算法應用在自動排版系統以搜尋最佳排版方法。基因演算法運作之基本要素為：

1. 染色體表達法：染色體表達法通常基因的排列順序。
2. 初始群體：是以隨機或根據一些準則挑選出一群初始染色體。
3. 基因運算元：基因運算元有兩種，一為交配運算元，其目的是將母代的特性加以合併，以形成新的子代；另一為突變運算元，其目的則是將染色體之基因任意的改變，以產生不同特性之子代。
4. 適應性函數：根據目標所訂定出之一適應性函數，以評估基因演算法所搜尋結果的優劣。較優之結果則有較好的適應性，並依據各染色體適應性的好壞加以篩選出較佳之染色體。

這些個要素構成了基因演算法的核心內容。經過了許多年的研究發展，基因演算法的應用研究已經從初期的組合優化求解拓展到了許多更新、更多元化的應用方面。並且與其他的進化規劃（evolution programming, EP）以及進化策略（evolution strategy, ES）等進化計算理論日益結合。儘管基因演算法比其他傳統搜索方法有更擅長全局搜索的能力，但是它的局部搜索的能力卻可能嫌不足，基於此未來也會尋求

可能的解決方案。

如圖 10 所示，設計 GA 基因演算法求解問題可大致分成下列步驟進行之：

1. 設計編碼方式，個別染色體表示法。
2. 決定每一代群體規模。
3. 設計適應函數，決定個體適應度的評估標準。
4. 決定挑選與複製方法。
5. 定義交配方式與交配機率。
6. 定義突變方式與突變機率。
7. 決定終止條件。

本文操作第一世代之染色體是用隨機法獲得基因編碼，世代中族群數出現相同染色體的機率非常非常低，一個

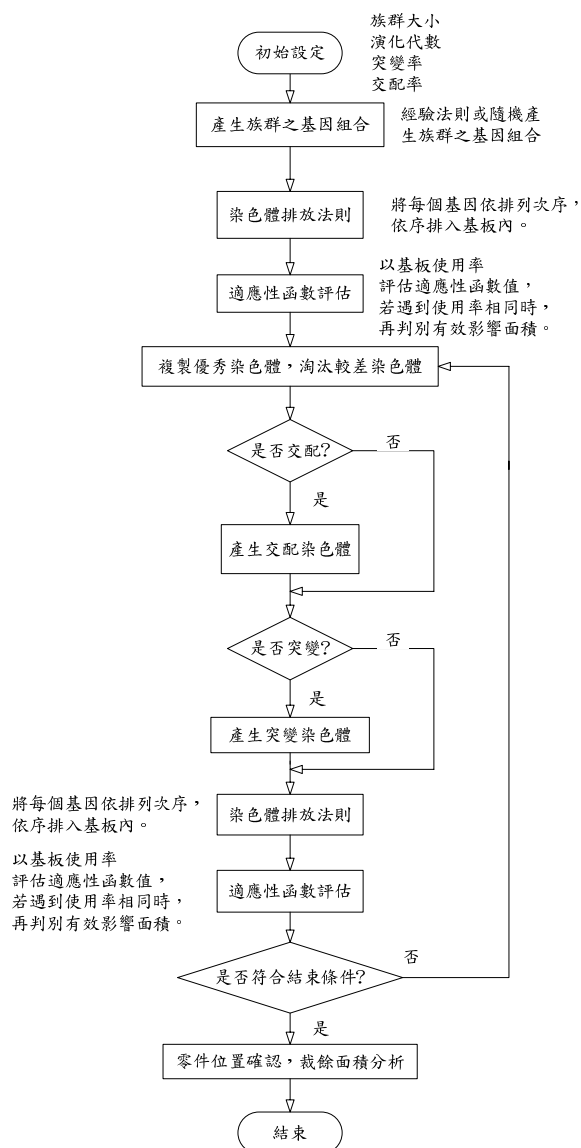


圖 10. 基因演算法之流程示意圖

零件就是一個基因，十七個零件就含有十七個基因，族群數若是 40，代表這個世代會出現 40 個不同的基因組合。子代交配方式則使用父代與母代正向多點交配，父代與母代反向多點交配；突變則使用完全突變，亦即染色體也是用隨機法獲得基因編碼。

五、實驗結果

首先我們利用文獻上 Soke 與 Bingul [4] 的十七個零件的最佳化排版的第一個實驗範例，如圖 11 左所示，零件的形狀與最佳解的位置，編號表示隨機產生之零件編號。

圖 11 右所示為零件按照排列次序，依序排入基板後的位置，這次的測試中，我們使用族群數為 20，演化十個世代，其演算產生出來的材料使用率為 89.44%。圖 12 左所示為零件使用族群數為 40，演化十個世代，其演算產生出來的材料使用率為 86.71%。圖 12 右所示為零件使用族群數為 60，演化十個世代，其演算產生出來的材料使用率為 86.67%。圖 13 左所示為零件使用族群數為 80，演化十個世代，其演算產生出來的材料使用率為 90.27%。圖 13 右所示為零件使用族群數為 100，演化十個世代，其演算產生出來的材料使用率為 90.1%。圖 14 表示這六種不同族群數，演化十個世代，其演算產生出來的材料使用率結果，材料使用率平均為 88.64%，結果顯示族群數的改變並沒有很明顯的差異性，但族群數若是很少，少到比零件數目還要少時，材料使用率明顯不佳的機率就大大增加。

同樣的零件形狀，相同的基板大小尺寸，如果考慮使用加工刀徑為 2mm 的切削刀具，我們即將零件之間隙設定為 2 時，如圖 15 左所示，零件的形狀與最佳解的位置，編號表示隨機產生之零件編號。圖 15 右所示為零件按照排列次序，依序排入基板後的位置，這次的測試中，我們使用族群數為 20，演化十個世代，其演算產生出來的材料使用率為 86.73%。圖 16 左所示為零件使用族群數為 40，演化十個世代，其演算產生出來的材料使用率為 88.93%。圖 16 右所示為零件使用族群數為 60，演化十個世代，其演算產生出來的材料使用率為 87.97%。圖 17 左所示為零件使用族群數為 80，演化十個世代，其演算產生出來的材料使用率為 86.78%。圖 17 右所示為零件使用族群數為 100，演化十個世代，其演算產生出來的材料使用率為 90.18%。圖 18 表示這六種不同族群數，演化十個世代，其演算產生出來的材料使用率結果，材料使用率平均為 88.12%，結果顯示族群數

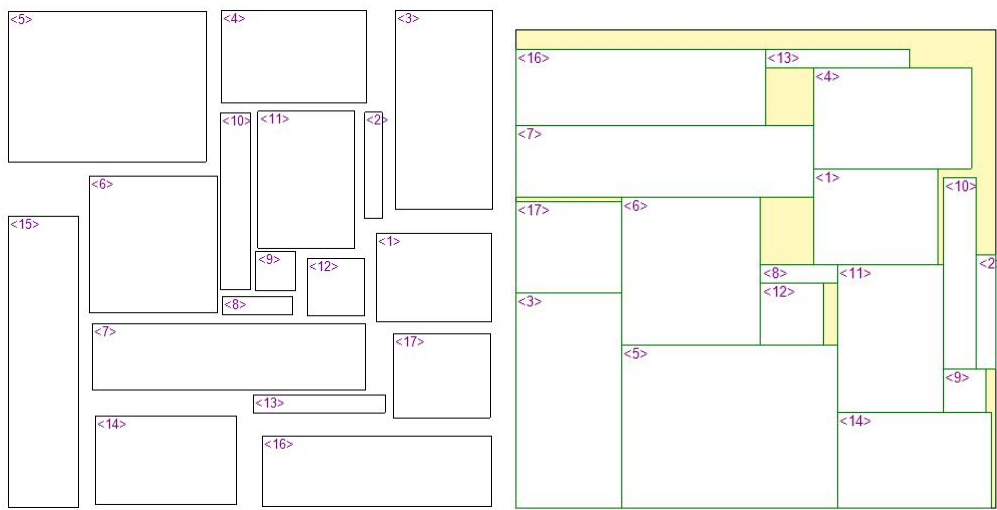


圖 11. 初始零件順序 (左) 與族群數為 20 時之排列完成結果 (右)

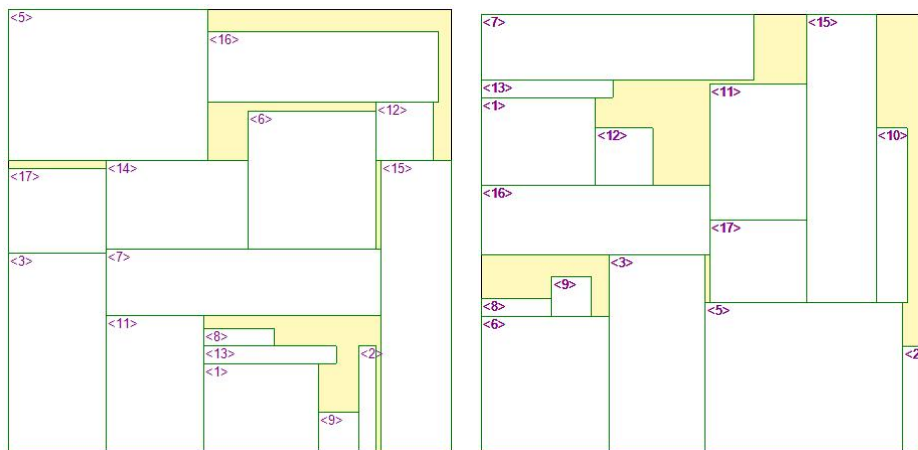


圖 12. 族群數為 40 時之排列完成結果 (左) 與族群數為 60 時之排列完成結果 (右)

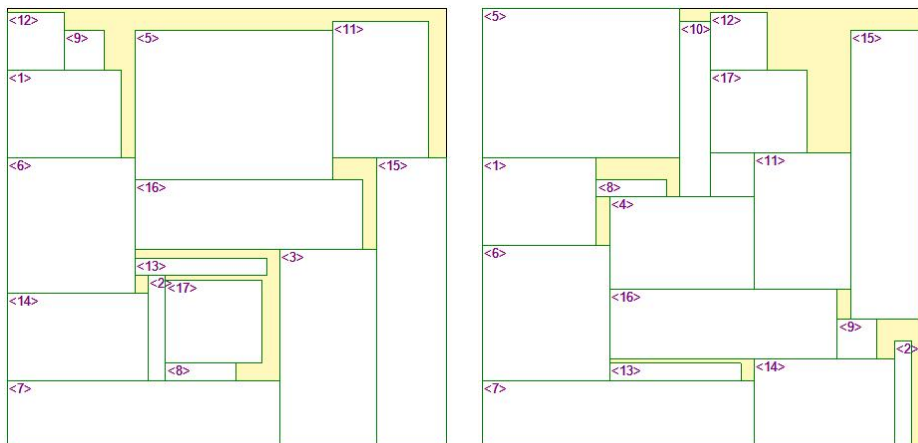


圖 13. 族群數為 80 時之排列完成結果 (左) 與族群數為 100 時之排列完成結果 (右)

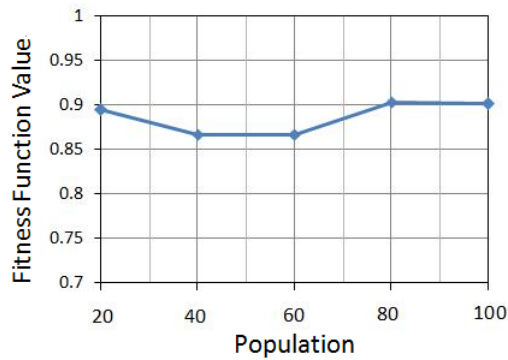


圖 14. 無間隙各種不同族群數演算時代後的結果

的改變並沒有很明顯的差異性，但族群數若是很少，少到比零件數目還要少時，材料使用率明顯不佳的機率就大大增加。

六、結論

本研究的主要目的是針對排版工作在加工切削上的處理，亦即是應用電腦輔助製造之技術，結合學理排版最佳化搜尋分析理論，針對間隙問題在基板與零件間之切割加工建立自主性之製造技術。圖 19 顯示在本系統操作後含間隙之排版結果可產生切削加工路徑之狀況，當選定切削加工用刀徑後，排版間隙也可自動被導入至排版系統，讓所有被切削工件沒有過切的顧慮。以本研究之排置理論未來將可整合以下幾個部分：中空幾何圖形建立、多工件自動化排版、刀具庫管理、加工路徑生成、後處理器架構與產生 CNC 機械加工程式碼。利用瞭解板材空間利用最佳化搜尋之基本理論，整合在大型電腦輔助設計製造軟體。

本文已成功發展出針對加工路徑之自動化排版系統，而且有不錯的功能表現。依照市場機制提出之需求，基板常會

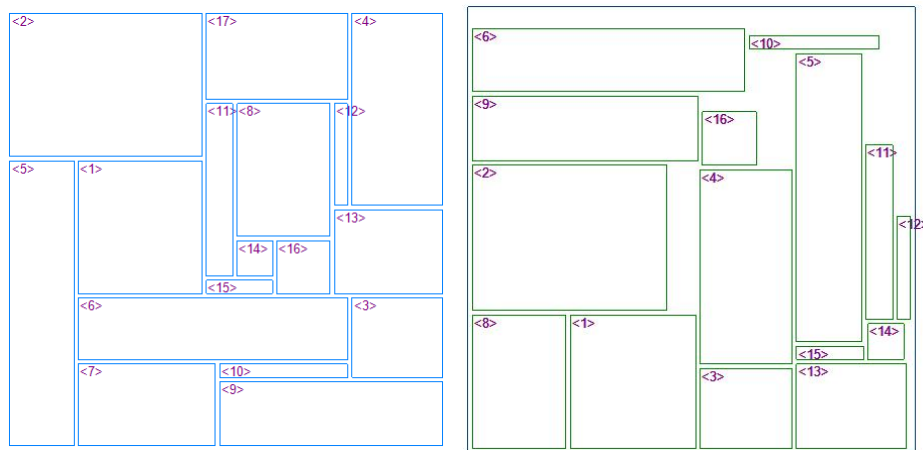


圖 15. 最佳零件順序（左）與族群數為 20 時之排列完成結果（右）

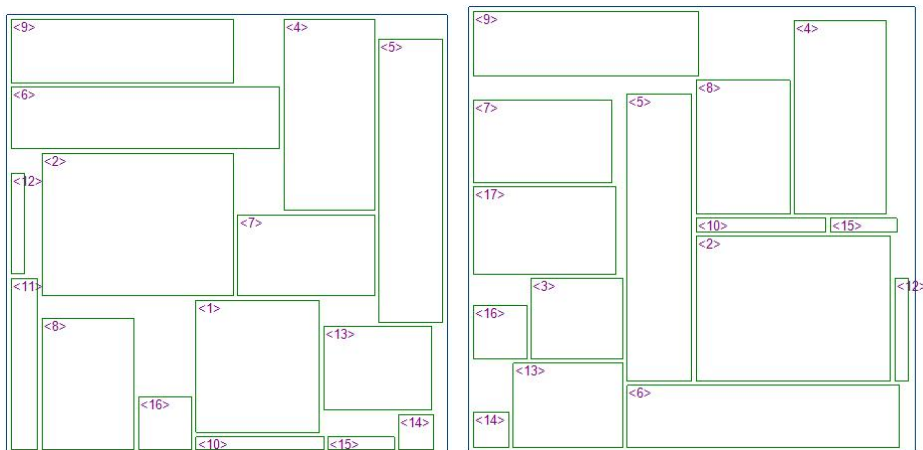


圖 16. 族群數為 40 時之排列完成結果（左）與族群數為 60 時之排列完成結果（右）

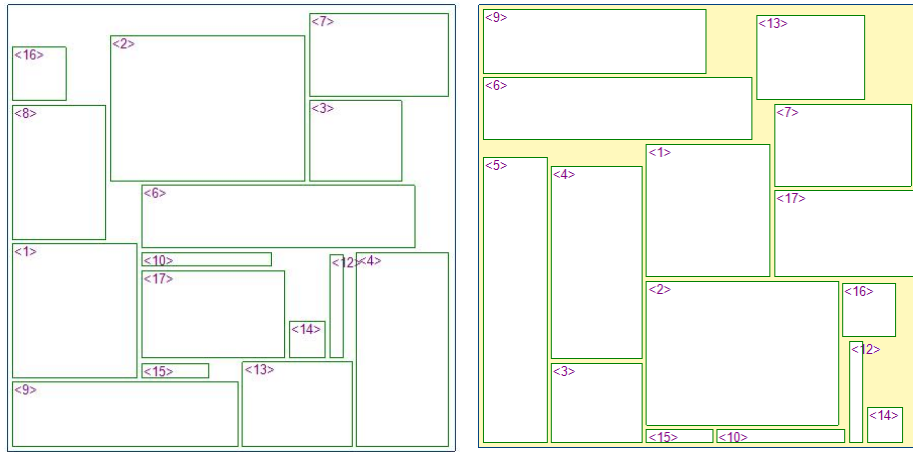


圖 17. 族群數為 80 時之排列完成結果（左）與族群數為 100 時之排列完成結果（右）

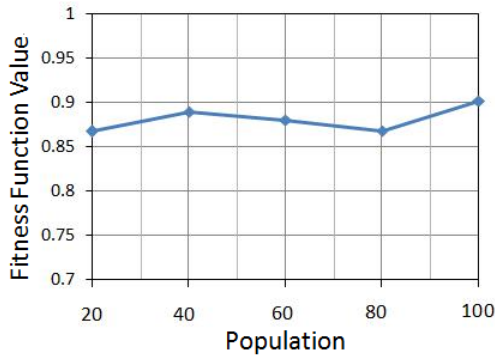


圖 18. 含間隙各種不同族群數演算時代後的結果

有中空的情況，零件也會有中空的情況，在使用排版的過程中就要考慮被挖空的情形，現階段的排版方式將無法滿足這樣的需求，將造成排版作業上的困擾。以作者個人的經驗所及，在眾多的文獻上甚少有學者提到中空之排版問題，或許

這更大大增加了零件位置排放方式的困難度，一定要有一個與目前方式不同的零件位置排放方式來處理中空的問題，否則利用已存在的方法可能難以有重大突破。我們也檢視市場上的排版系統，有些不是效果差強人意，就是功能尚嫌不足，還有一個問題就是沒有獨立的作業系統，這樣會大大地限制在程式開發上的困難，不容易有操控上的優勢，也減低使用者的使用意願。未來將繼續針對矩形零件開發一套自動排版系統，此電腦輔助排版系統可達到處理中空基板與中空零件的工作，也就是零件排入基板後，已排入之零件如果也是中空的狀況，其內部也還可以再排入更小的零件，如此可以將基板的使用率再大幅提高。系統完成後將可直接利用電腦輔助製造產生加工路徑，以利後端 CNC 加工。

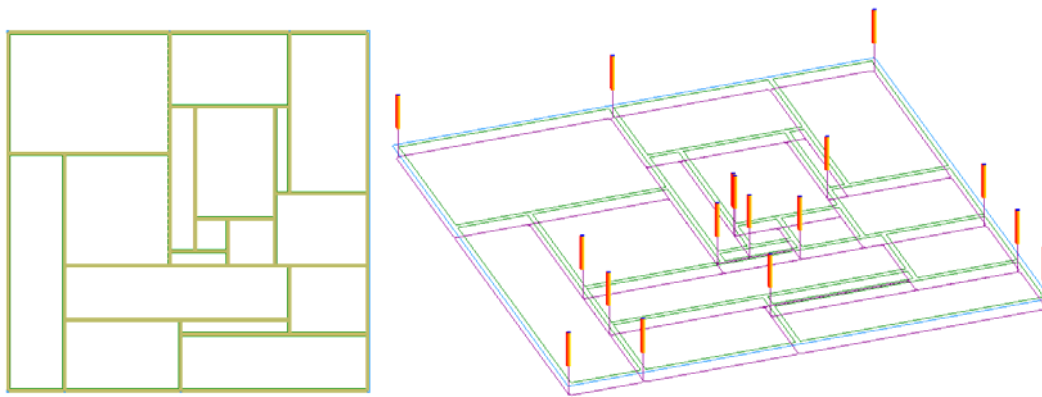


圖 19. 含間隙之排版結果可產生切削加工路徑

誌謝

感謝國科會計劃 NSC96-2622-E-212-002-CC3 在經費上的補助。

參考文獻

- Bazewicz, J., A. M. Salvador and R. Walkowiak (2004) Parallel tabu search approaches for two-dimensional cutting. *Parallel Processing Letters*, 14, 23-32.
- Beasley, J. E. (2004) A population heuristic for constrained two-dimensional non-guillotine cutting. *European Journal of Operational Research*, 156, 601-627.
- Beasley, J. E., N. Meade and T. J. Chang (2003) An evolutionary heuristic for the index tracking problem. *European Journal of Operational Research*, 148, 621-643.
- Bennell, J. A. and K. A. Dowland (1999) A tabu thresholding implementation for the irregular stock cutting problem. *International Journal of Production Research*, 37, 4259-4275.
- Binkley, K. J. and M. Hagiwara (2007) Applying self-adaptive evolutionary algorithms to two-dimensional packing problems using a four corners' heuristic. *European Journal of Operational Research*, 183, 1230-1248.
- Bortfeldt, A. (2006) A genetic algorithm for the two-dimensional strip packing problem with rectangular pieces. *European Journal of Operational Research*, 172, 814-837.
- Boschetti, M. A., E. Hadjiconstantinou and A. Mingozzi (2002) New upper bounds for the two-dimensional orthogonal non guillotine cutting stock problem. *IMA Journal of Management Mathematics*, 13, 95-119.
- Burke, E. K., G. Kendall and G. Whitwell (2004) A new placement heuristic for the orthogonal stock-cutting problem. *Operations Research*, 52(4), 655-671.
- Caprara A. and M. Monaci (2004) On the two-dimensional knapsack problem. *Operations Research Letters*, 32, 2-14.
- Dagli, C. H. and M. Y. Tatoglu (1987) An approach to two dimensional cutting stock problems. *International Journal of Production Research*, 25, 175-190.
- Dori, D. and M. Ben-Bassat (1984) Efficient nesting of congruent convex figures. *Communications of the ACM*, 27(3), 228-235.
- Faina, L. (1999) Application of simulated annealing to the cutting stock problem. *European Journal of Operational Research*, 114, 542-556.
- Fekete, S. P. and J. Schepers (1997) A new exact algorithm for general orthogonal d-dimensional knapsack problems. *Springer Lecture Notes in Computer Science*, 1284, 144-156.
- Fischer, A. D. and C. H. Dagli (2004) Employing subgroup evolution for irregular-shape nesting. *Journal of Intelligent Manufacturing*, 15, 187-199.
- Freeman, H. and R. Shapira (1975) Determining the minimum area encasing rectangle for an arbitrary closed curve. *Communications of the ACM*, 81(7), 409-413.
- Gilmore, P. C. and R. E. Gomory (1961) A linear programming approach to the cutting stock problem - Part I. *Operations Research*, 9, 849-859.
- Gilmore, P. C. and R. E. Gomory (1963) A linear programming approach to the cutting stock problem - Part II. *Operations Research*, 11, 863-888.
- Gomes, A. M. and J. F. Oliveira (2002) A 2-exchange heuristic for nesting problems. *European Journal of Operation Research*, 144, 359-370.
- Goncalves, J. F. (2007) A hybrid genetic algorithm-heuristic for a two-dimensional orthogonal packing problem. *European Journal of Operational Research*, 183(3), 1212-1229.
- Hadjiconstantinou, E. and M. Iori (2007) A hybrid genetic algorithm for the two-dimensional single large object placement problem. *European Journal of Operational Research*, 183(3), 1150-1166.
- Hopper, E. and B. C. H. Turton (1999) A genetic algorithms for a 2D industrial packing problem. *Computers in Engineering*, 37, 375-378.
- Hopper, E. and B. C. H. Turton (2001) An empirical investigation of meta-heuristic and heuristic algorithms for a 2D packing problem. *European Journal of Operational Research*, 128(1), 34-57.
- Jakobs, S. (1996) On genetic algorithms for the packing of polygons. *European Journal of Operational Research*, 88, 165-181.
- Lai, K. K. and W. M. Chan (1997) An evolutionary algorithm for the rectangular cutting stock problem. *International Journal of Industrial Engineering*, 4, 130-139.
- Leung, T. W., C. K. Chan and M. D. Troutt (2003) Application of a mixed simulated annealing-genetic

- algorithm heuristic for the two-dimensional orthogonal packing problem. *European Journal of Operational Research*, 145, 530-542.
26. Liu, D. and H. Teng (1999) An improved BL-algorithm for genetic algorithm of the orthogonal packing of rectangles. *European Journal of Operational Research*, 112, 413-420.
27. Lodi, A., S. Martello and D. Vigo (1999) Approximation algorithms for the oriented two-dimensional bin packing problem. *European Journal of Operational Research*, 112, 158-166.
28. Lodi, A., S. Martello and D. Vigo (2002) Recent advances on two-dimensional bin packing problems. *Discrete Applied Mathematics*, 123, 373-390.
29. Lodi, A., S. Martello and D. Vigo (2004) Models and bounds for two-dimensional level packing problems. *Journal of Combinatorial Optimization*, 8, 363-379.
30. Lodi, A., S. Martello and M. Monaci (2002) The two-dimensional packing problems: A survey. *European Journal of Operational Research*, 141, 3-13.
31. Ma, H. and C. C. Liu (2007) Fast nesting of 2-D sheet parts with arbitrary shapes using a greedy method and semi-discrete representations. *IEEE Transactions on Automation Science and Engineering*, 4(2), 273-282.
32. Ma, H., D. J. Cannon and S. R. T. Kumara (1995) A scheme integrating neural networks for real-time robotic collision detection. Proceedings of IEEE International Conference on Robotics and Automation. Nagoya, Japan.
33. Martello, S., M. Monaci and D. Vigo (2003) An exact approach to the strip packing problem. *INFORMS Journal on Computing*, 15(3), 310-319.
34. Pinol, H. and J. E. Beasley (2006) Scatter search and bionomic algorithms for the aircraft landing problem. *European Journal of Operational Research*, 171, 439-462.
35. Pisinger, D. (2002) Heuristics for the container loading problem. *European Journal of Operational Research*, 141(2), 382-392.
36. Prasad, Y. K. D. V., S. Somasundaram and K. P. Rao (1995) A sliding algorithm for optimal nesting arbitrarily shaped sheet metal blanks. *International Journal Production Research*, 33(6), 1505-1520.
37. Ramesh Babu, A. and N. Ramesh Babu (1999) Effective nesting of rectangular parts in multiple rectangular sheets using genetic and heuristic algorithms. *International Journal of Production Research*, 37(7), 1625-1643.
38. Ramesh Babu, A. and N. Ramesh Babu (2001) A generic approach for nesting of 2-D parts in 2-D sheets using genetic and heuristic algorithms. *Computer-Aided Design*, 33, 879-891.
39. Scheithauer, G. (1999) Lp-based bounds for the container and multi-container loading problem. *International Transactions in Operational Research*, 6, 199-213.
40. Soke, A. and Z. Bingul (2006) Hybrid genetic algorithm and simulated annealing for two-dimensional non-guillotine rectangular packing problems. *Engineering Applications of Artificial Intelligence*, 19, 557-567.
41. Tay, E. H., T. Y. Chong and F. C. Lee (2002) Pattern nesting on irregular-shaped stock using genetic algorithms. *Engineering Applications of Artificial Intelligence*, 15, 551-558.
42. Wascher, G., H. Haussner and H. Schumann (2007) An improved typology of cutting and packing problems. *European Journal of Operational Research*, 183, 1109-1130.
43. Zhang, D., Y. Kang and A. Deng (2006) A new heuristic recursive algorithm for the strip rectangular packing problem. *Computers and Operations Research*, 33, 2209-2217.

收件：97.05.09 修正：97.06.09 接受：97.08.01