Journal of Science and Engineering Technology, Vol. 2, No. 3, pp. 75-83 (2006)

以基因演算法求解單原片方形物件排列問題

吳泰熙 [1]　吳奕樺 [2]　張欽智 [3]

[1] 逢甲大學工業工程與系統管理系

台中市西屯區文華路 100 號

[2] 美國奧本大學工業與系統工程系

Auburn, Alabama 36849, USA

[3] 仁德醫護管理專科學校資訊管理科

苗栗縣後龍鎮溪洲里 7 鄰砂崙湖 79-9 號

摘　要

工業界莫不亟思要如何精減人力以提昇效率、全面降低生產成本。然而尚有部分產業仍利用人工方式來處理原物料切割／排列等相關作業。此類複雜之「物件切割」問題需要有合適之演算法來迅速求得原物料切割計畫。過去文獻中，不乏因為使用遺傳基因演算法（genetic algorithm, GA）而獲致不錯演算績效之研究，因此本研究採用 GA 之演算架構，但繼續加強在上述文獻中尚未被著墨之「菁英策略」，並使用不同之「編碼」設計，以期超越文獻之 GA 演算結果。在例題測試階段，我們首先以部分例題對所提出之 GA 演算法之參數進行實驗，以獲取參數組合建議。接著經由建議之參數組合，對 38 題國際文獻演算例題進行測試，再與國際文獻演算結果進行比較。結果顯示本研究所提出之 GA 演算法，在中、小題型之單原片方形物件排列問題中幾乎皆優於國際文獻最佳結果，某些例題甚至超越文獻結果將近 10% 之使用率，演算績效可謂非常傑出。

關鍵詞：方形物件排列問題，基因演算法

Solving Two-Dimensional Packing Problems by Using a Genetic Algorithm

TAI-HSI WU[1], YI-HUA WU[2] and CHIN-JU CHANG[3]

[1]Department of Industrial Engineering and Systems Management, Feng Chia University

100 Wenhwa Rd., Seatwen, Taichung, Taiwan

[2]Department of Industrial and System Engineering, Auburn University

Auburn, Alabama 36849, USA

[3]Department of Information Management, Jen-Te Junior College of Medicine, Nursing and Management

79-9, Shijou Li, Houlong, Miaoli, Taiwan

ABSTRACT

In this study, a genetic-based algorithm (GA) is proposed for nesting two-dimensional rectangular parts in a material plate. In the literature GA has been adopted for solving the same problem, fairly good results having been obtained. A new method for designing the coding of the chromosomes, accompanied by a proposal for an elitism strategy, is presented in this research, aiming at further extending the problem-solving capability of the traditional GA. Promising computational results are obtained and reported by running examples from the literature.

Key Words**:** two-dimensional packing problem, genetic algorithm, elitism strategy

一、緒論

隨著時代進步、工業快速發展，如何降低生產成本來獲得較高之利潤是各企業極力追求的目標之一。然而觀察時下的產業，仍有不少廠家依舊利用傳統人工方法來處理切割、排列等相關作業，但此舉卻恐造成不必要之物料及成本浪費。因此業者莫不絞盡腦汁，發展出快速的解決之道。「物件切割」問題乃在預先給定的物料原片（material plate）上，剪裁出所需要數量及尺寸之物件，以對物料原片做最有效之利用；而「物件排列」問題是將已知數量及規格之物件，在一個預先給定的空間中做有效排列，達成最高的空間使用率。由於排列／切割兩問題間，存在一材料有效使用與空間有效利用之一體兩面關係，實可以視爲相同的問題，皆屬於組合最佳化（combinatorial optimization）之決策問題，因此後續本文以「物件排列」問題作爲兩類問題之統稱。

對於組合最佳化問題之求解，由於問題結構之複雜，且隨著問題規模增大，解題時間會呈現指數遞增之情形。若以傳統之數學規劃方式求解，雖能保證獲致最佳解，但卻將招致演算耗時無效率、未能及時因應之缺失。因此近年來之研究通常採用能於可接受甚至極短時間內獲致最佳解或近似最佳解之啓發式演算法，特別是萬用型啓發式演算法（meta-heuristic algorithm），例如模擬退火演算法（simulated annealing, SA）、基因演算法（genetic algorithm, GA）、禁忌搜尋法（tabu search, TS）。

基因演算法（GA）是由 Holland 於 1975 年首先提出，爲一根據物競天擇，適者生存之道理所發展出之優選技術，近年來已被廣泛應用於各領域中。它透過選擇物種中具有較好特性的上一親代，並隨機的相互交換親代中彼此基因之資訊及特性，以適合度函數爲引導，希望保留親代中較被喜好之特性並繁衍至子代，以產生較上一親代更爲優秀的子代。如此一再地重複，藉以產生適合度最強的物種，來達到最佳化的目的。由於過去在方形物件排列問題之文獻中，不乏因爲使用 GA 而獲致不錯演算績效之研究，例如 Babu 與 Babu [1] 及 Wu 等人 [25]，因此本研究仍然採用 GA 之演算架構，但繼續加強在上述文獻中尚未被著墨之「菁英策略」（elitism strategy），並使用不同之「編碼」設計，希望透過此二項機制之導入，設計出更好表現之 GA 演算法，以期最後能有效地產生最佳或近似最佳之方形物件排列計畫。

在本研究所探討之方形物件排列問題中，就所提供的方形物料原片之數目而言，可分爲單一原片及多張原片問題。若物件數量過多，無法排入單一原片時，自然需要用到多張原片。多張原片問題的處理方法大多是以數學規劃方式或啓發式解法求解問題：首先將所有需求物件大略地指派到各張物料原片上，然後再進行各單張原片之排列。事實上，多數的文獻仍著重於單原片之排列問題，因此本研究之內容僅就單原片之方形物件排列問題進行探討。

二、文獻回顧與探討

本節對相關文獻進行回顧與探討，包含方形物件排列問題之相關解法及研究。過去文獻，隨著產品種類的不同，如物件在尺寸、形狀、方向、數量上的差異，可衍生出各種不同類型的排列問題，由於本研究對於方形排列問題提出演算法，因此在文獻探討部份將專就方形排列問題之文獻加以描述。

事實上，二維物件排列問題爲一 NP-Hard 之問題 [12]，因此隨著問題愈複雜，解題時間呈現出指數遞增之情形。所以在以往之研究中，大部分仍著重於發展快速的啓發式解法，雖然在解答的精確度上有所犧牲，卻可使解題效率快速凌越最佳解之解題效率。Dagli 與 Tatoglu [9] 及 Babu 與 Babu [1] 提供了二維排列問題之數種問題分類方式。就排列之方式而言，可概分爲啓發式及最佳化：啓發式解法原則上即是透過一些圖形特徵法則（如面積、長、寬、高）來決定圖形置入之次序及置入之位置 [8, 17, 23]；另一種類型之演算法則是透過各需求圖形間事先的聚合（grouping），經過逐步的迭次累積，直到所有的圖形皆被聚合完成 [15, 16]。至於最佳化技術應用於方形排列問題之文獻，依（1）決策目標，及（2）演算效率，分別探討如下。在決策目標部分，大致

可歸納為（a）使用物料原片張數最小化，如 [6, 7]；及（b）使用率最大或浪費量最少 [3, 9]。在演算效率部分，由於排列問題本身為 NP-Hard，若嘗試以最佳化方式來求取使用率最大或浪費量最少之排列方式，由於演算效率低劣，勢必無法應用到實務問題。為了提升排列效率，一系列之研究 [2, 24] 針對演算法之最差情況（worst case analysis）加以分析來評定演算法之好壞。稍後亦有研究提出應當針對演算法之平均表現效率加以分析，而非最差情況。Biro 等人 [5] 利用網路結構之解題效率較佳之特色，將排列問題視為一網路問題，惟演算結果並不如預期來得理想。Chen [7] 建構了一數學模式，用以求解不同大小的方形盒擺放在棧板上，而達成棧板使用量最少之目標。

除了數學規劃方式外，針對排列問題專門量身定做而發展演算法者，例如 Yanasse 等人 [26]，Lamousin 及 Waggenspack [19]。由於萬用啓發式演算法在求解組合最佳化問題中有極為優異之表現，因此也有許多文獻使用這些演算技術來求解此類問題，例如 Dagli 與 Hajakbari [11]、Lutfiyya 與 McMillin [22]、Theodoracatos 與 Grimsley [24]、Lai 與 Chan [18]、Parada 等人 [23] 等人使用模擬退火演算法。Lai 與 Chan [18] 利用 SA 配合 DP （difference process algorithm）擺放邏輯，並在 SA 降溫程序中使用文獻所定義的降溫程序，經實例測試後可以成功地降低原物料的剩餘浪費。Leung 等人 [21] 於 2001 年分別使用 SA 與 GA，再配合 DP 擺放邏輯。到 2003 年時，更將原本的演算法加以改良，提出 SA 與 GA 混合之演算法 [20]，雖然在使用率上有不錯的表現，但相對的在執行時間上也花費了更多的時間。Bennell 與 Dowsland [4] 使用禁忌搜尋演算法；Dagli [10] 使用類神經網路演算法。Jacob [17]、Ismail 及 Hon [14] 使用傳統 GA 演算法；Babu 與 Babu [1] 亦採用 GA 演算法求解方形物件排列問題，將各分片排列至原片上。他們採用左下角優先放置原則，且利用投影座標產生參考點，來記錄各個分片在原片上的排列位置（如圖 1 中在分片被置入原片的過程中，陸續產生了編號 1 號到 14 號等 14 個參考點），以產生出具最佳使用率之排列方式。該研究並透過遺傳基因演算程序，配合有效的複製方法、交配程序和突變機率，經過數百至數千次個迭次演算之後，求得最佳解。

Wu 等人 [25] 針對 Babu 與 Babu [1] 演算法中未有效利用部分空間之缺點，提出一新的分片置放邏輯 IBH，並結合 SA 演算法。他們針對方形物件排列問題進行求解，考慮所有可排列之參考點以求得較精確的排列結果。Wu 等人 [25] 所提出之 IBH 擺放法則，不僅包含了在這之前學者所提出的擺放準則，更加入了“旋轉分片”和“移動分片置入順序”機制，使得原片之利用更加極致。經執行文獻例題後，顯示其結果明顯優於 Babu 與 Babu [1] 之結果，特別是在中大型例題上，此種優勢更為明顯。他們更進一步地利用此方形排列演算法至不規則物件排列問題。其作法乃是先找出各個不規則物件之最小包覆方形，再應用方形排列演算法，排列完成後先去除外觀之包覆方形，再尋求所有可能之緊靠。

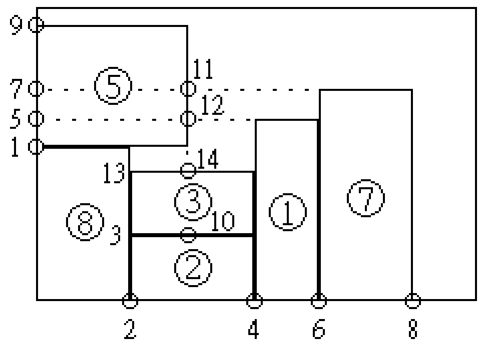

圖 1. 左下優先置放邏輯 [1]

由如上文獻介紹中不難發覺，由於具備廣泛搜尋及融合各代解答之優點，GA 已被廣為使用於各類型物件排列／切割問題之求算。雖然如此，各種研究仍前仆後繼地投入，以改善 GA 之各項運算程序之設計，例如編碼方式（coding）、交配及突變機制等，以期能將 GA 之演算能力發揮至極致。因此針對方形物件排列問題之求解，本研究仍然沿用 GA 之演算架構，但繼續加強在文獻中尚未被著墨之「菁英策略」，並使用不同之「編碼」設計，希望透過此二項機制之導入，設計出一能較文獻有更好表現之 GA 演算法。

三、研究方法

針對本研究所討論之單原片方形物件排列問題，由於 Wu [25] 等人所提出之 IBH 擺放法則（如圖 2）之邏輯，不僅包含了在其之前學者所提出的擺放準則，更加入了「旋轉分片」和「移動分片置入順序」之機制，使得原片之使用更加極致。因此本研究依循 IBH 之擺放法則，並提出一加入「菁英策略」及重新設計之編碼方式。

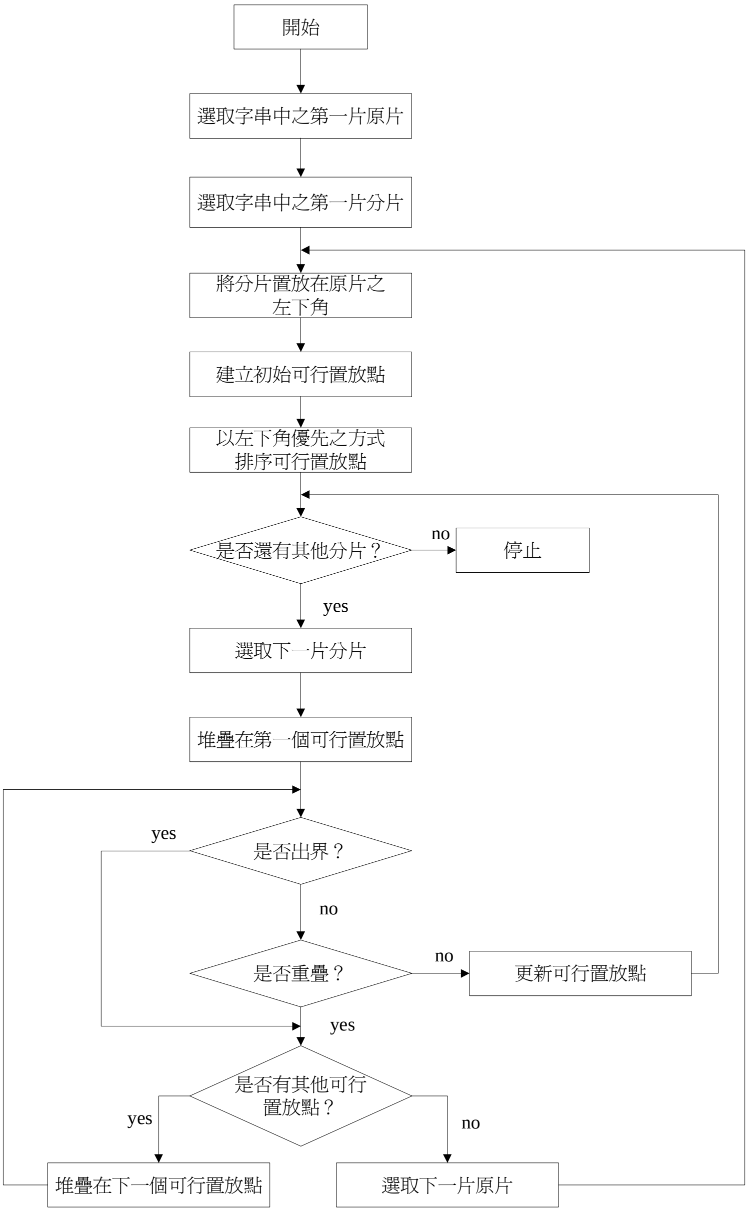

圖 2. IBH 擺放法則 [25]

本研究設計之基因演算法乃將生物界演化的機制應用在搜尋決策問題之最佳解，以下將就 GA 演算法之執行步驟逐一說明：

（一）預處理

在求解方形物件排列問題中，本研究先進行預處理：將所有分片依面積由大至小排列並依序編號；換言之，編號愈小，其面積愈大，愈優先被置入原片內。

（二）母體產生

基因演化過程所需之族群稱爲初始族群，即是所謂的母體；族群成員稱之爲個體（individual）或染色體（chromosome）。產生母體最簡單之方式乃透過隨機選取，意即隨機地從整個搜尋空間中挑選出所需數量之染色體當作最初始之母代。但由於隨機的關係，初始母體內各染色體基因序列之品質往往不一，卻也因此可在演化過程中提供充分之個體差異度，避免發生小範圍內的重組情況，導致結果陷入區域解。

母體大小（population size）之選擇也是影響 GA 績效非常重要之課題之一。若母體太小則演算法將有過早收斂之疑慮，且常會因母體提供之訊息不足而導致產生較差之解答；反之，母體過大則收斂速度將因此變慢而不具效率。雖然母

體大小之決定攸關 GA 演算法之績效甚巨，恰當之母體大小仍須由實驗測試得知。

（三）編碼

在 GA 中，每條染色體即代表對應問題之一組解答，染色體中的基因則表示每個影響因素之猜測值。於此單原片方形物件排列問題中，我們設計染色體中的基因即爲欲置入原片中之分片編號。文獻經驗顯示各分片在擺放的過程當中若允許其作 90 度之旋轉，將可提昇原片使用率。然過去之研究均在基因交配之過程中透過隨機選取少許分片方式讓被選取之分片旋轉 90 度，並非系統化地決定每個分片置入之角度（0 度或旋轉 90 度）。本研究認爲若能將各分片置入之角度（0 度或旋轉 90 度）以基因方式明顯呈現於染色體中，再透過複製、交配及突變等機制，將有可能爲每一分片之置入角度做最佳之決定。圖 3 中染色體內數字爲分片編號，而染色體後端陰影部分表示各分片作 90 度旋轉之後的編號。本例中有 7 片分片，其中編號 1 與 8 爲同一分片，但卻代表不同置入角度，分別爲 0 度及旋轉 90 度。

（四）適合度函數

適合度函數代表遺傳演化系統的性能指標。GA 藉著保留有較高適應函數值的物種與物種間互換基因的方式，一個世代接著一個世代的進化，使系統性能提昇。GA 先根據問題之需求來定義一個適合度函數，以決定各染色體之適應程度，藉以評估該物種是否該生存下來。族群中染色體之適應函數值越佳，代表該染色體的適應程度越高，則被選取參與後代演化之機率越大；反之，適應函數值越差，該染色體被淘汰的機率也越大。若子代族群之適應函數值較母代更佳，即表示子代比母代有更優異之性能表現。於此單原片方形物件排列問題中，適合度函數即爲原片之使用率，即所有需求分片之面積總和除以原片之面積。

（五）複製

複製，是指生物體對自身的複製，複製出來的子代可以跟親代完全一樣；也可以有部份的不同，其目的就是讓母代的特徵不同程度地保留到子代，以免隨著時間的演進，生物演化出來的長處反而消失。進行複製前需先決定複製數目後，再開始進行複製。複製時，新、舊染色體依序由適合度函數最佳的染色體依複製數量開始複製，當複製已達母體數時就停止複製。

由於複製是挑選表現較佳的個體來取代較差的個體，因此這個動作也被稱爲「選擇」（selection）。選擇的方法一般是經由輪盤法（roulette wheel selection）來進行，在每一代的演化過程中，首先依每條染色體的適合度函數值（fitness value）大小來分割輪盤的位置，適合度函數值越好的染色體，在輪盤上佔有的面積比例也越大，當然被挑選進入演化池中之機率也越大。

（六）交配

族群中不同染色體間可以經由隨機交錯各自之數個基因之過程，稱爲交配，以產生新的子代。首先在母體中任選兩條染色體，稱之爲親代，然後隨機在親代字串當中挑出兩點，稱之爲交配點，再將位於兩交配點間之基因進行交換，生成兩個新的染色體，即爲子代，如圖 4 所示。

但並非每個交配池內之染色體都需要交配，因此我們定義一個交配機率（crossover rate）：交配率越高，多元廣度搜尋的速度會越快，但也可能造成原先優良的染色體被替換掉；交配率太低，又容易造成進化的停滯，所以 Babu 與 Babu [1] 將交配率的值設定於 0.5~1 之間，在這一區間作交配率的選取，本研究亦依循其設定。

（七）突變

往往在經過演化過程後，會發生染色體過早收斂而陷入區域解之現象，因此需要透過突變部分染色體內之基因來跳脫區域解，進而邁向最佳解。突變之進行乃是先透過隨機選取一條染色體及其內兩個突變點，再交換兩突變點之基因。突變之數量則交由突變率所控制，突變率也決定演化的快慢。如果突變率一開始設的過大，將使得前幾代很快收斂；在演化後段時，大的突變率反而會造成演化的阻礙，因爲易於突變的性質將使得世代不穩定，無法進行有效的搜尋。

圖 3. 7 片分片之編碼範例

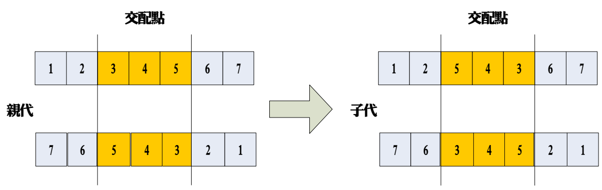

圖 4. 染色體交配示意

（八）精英策略

精英策略即是母代染色體在經過複製、交配與突變運算之後，產生子代新的染色體，而選取其中表現較佳之染色體來取代舊的染色體成爲新的族群，以繼續下一世代的進化，也就是將舊族群中適合度函數值最大者或前幾名的染色體保留下來，其餘的以子代新染色體取代，成爲新的族群。

本研究以 *Pe* 爲菁英策略的選擇機率。舉例來說，當 *Pe* 等於 0.4 時，代表有 40% 母代的染色體保留下來（取自舊族群中適合度函數值前 40% 之染色體）；而其餘的 60% 則以表現較佳的子代染色體來取代，成爲新的族群。每當經過幾個世代時，皆會重複上述的動作，不斷的更新族群中的染色體，透過精英策略之方式來留住表現較佳之染色體，直到演化終了。

（九）終止條件

除了達到事先設定之演化代數，程式將自動結束並輸出最終結果外，若連續 N 次迭代，解答品質皆未改善時，則演算法亦將終止，以免浪費演算時間。

四、演算結果及分析

本研究針對單原片之方形物件排列問題，提出一採用「菁英策略」及重新編碼之 GA 並搭配 IBH 擺放法則求解。此節首先以部分例題對此 GA 演算法之參數進行實驗，以期能獲取最佳參數組合，進而有效率地求算出最佳演算結果。本研究所採用之程式編譯軟體爲 Visual C++ 6.0，測試環境爲 Pentium IV 2.6G 及 RAM 爲 DDR 512M 之個人電腦。

（一）文獻例題說明

文獻中有大量的單原片方形物件排列例題可供測試，本研究所採用之測試例題來源及基本資料如表 1 所示。在所有提供例題中，分片規格是將原片裁剪出各種不同大小及數量的方形物件，由於是將原片完全的利用，所以最佳使用率均爲 100%。在 Hopper 與 Turton [13] 所提供之範例中，分片數目從 16~197 片，包含各種問題規模之題型，因此下節的參數分析中我們將選用其所提供之例題，於本研究測試範例內之編號分別爲 25、26、27，三題皆爲 49 片分片之例題，以求得一組較適合的 GA 參數，並以此參數組合求解本研究所有測試例題。

（二）演算參數設定

在參數方面，由於 Babu 與 Babu [1] 在其文獻中交配率和突變率分別設定爲 0.6 和 0.25 時，可得到較佳解，且 Wu 等人 [25] 在其文獻中亦採用相同之設定，因此，本研究亦採用此參數設定值；而在母體數的多寡分別以（原片數+分片數）、（原片數+分片數）/2、（原片數+分片數）×2 進行分析；迭代數分別選取 25、50 和 100；精英策略的選擇率分別爲 0.2、0.4、0.6 作一分析。

表 1. 測試例題基本資料表

題號	分片數	題號	分片數
1[17]	25	20[13]	25
2[17]	50	21[13]	25
3[9]	16	22[13]	28
4[9]	19	23[13]	29
5[9]	50	24[13]	28
6[9]	50	25[13]	49
7[18,21]	10	26[13]	49
8[18,21]	15	27[13]	49
9[18,21]	20	28[13,21]	73
10[18,21]	10	29[13,21]	73
11[18,21]	10	30[13,21]	73
12[18,21]	15	31[13,21]	97
13[18,21]	15	32[13,21]	97
14[18,21]	20	33[13,21]	97
15[13]	30	34[13,21]	196
16[13]	16	35[13,21]	197
17[13]	17	36[13,21]	196
18[13]	16	37[21]	40
19[13]	25	38[21]	50

本節針對 GA 的母體染色體數、迭代數與精英策略選擇率進行實驗測試和分析，再將演算結果經由 SAS 8.1 軟體分析後得到最佳之參數組合建議。各參數組合之詳細演算結果列示表 2；各參數在不同的水準之下，使用率變化與變異數分析如表 3、表 4。由變異數分析中得知迭代數、母體數與精英策略選取範圍等三種參數，對目標值之表現有顯著差異。我們進一步執行 Duncan 多重檢定，最終建議參數組合爲：迭代數＝100，母體數＝50，精英策略選擇率＝0.4。

（三）演算結果與分析

本節經由實驗分析後的建議參數組合，對 38 題演算例題進行測試。測試過程中對每一演算例進行十次之演算，並記錄十次演算之最佳及平均結果，再與國際文獻演算結果進行比較，如表 5 所示。

觀察表 5 與文獻最佳結果比較，本研究所提出之 GA 演算法，在例題 1~26 中，有 21 個例題之結果優於國際文獻最佳結果，某些甚至超越文獻結果將近 10% 之使用率，演算

表 2. 實驗參數詳細結果

參數*	實驗例題一(#25)		實驗例題二(#26)		實驗例題三(#27)	
	使用率	時間	使用率	時間	使用率	時間
25-25-0.2	0.9419	72.656	0.9434	77.547	0.9524	87.547
25-25-0.4	0.9544	92.655	0.9578	77.198	0.9554	74.951
25-25-0.6	0.9524	79.437	0.9449	69.571	0.9302	95.357
25-50-0.2	0.9554	76.547	0.9626	92.655	0.9569	91.906
25-50-0.4	0.9302	77.719	0.9626	79.437	0.9557	127.930
25-50-0.6	0.9569	148.610	0.9434	76.547	0.9574	71.625
25-100-0.2	0.9499	65.750	0.9524	77.719	0.9375	64.563
25-100-0.4	0.9419	66.484	0.9474	148.610	0.9419	74.860
25-100-0.6	0.9419	70.048	0.9536	71.625	0.9554	102.000
50-25-0.2	0.9554	151.890	0.9474	64.563	0.9554	71.516
50-25-0.4	0.9554	74.547	0.9536	74.860	0.9557	87.542
50-25-0.6	0.9557	83.719	0.9569	102.000	0.9554	68.795
50-50-0.2	0.9574	71.516	0.9474	71.625	0.9302	71.516
50-50-0.4	0.9375	87.542	0.9554	88.547	0.9569	151.890
50-50-0.6	0.9419	68.795	0.9554	67.852	0.9499	74.547
50-100-0.2	0.9585	71.516	0.9557	95.278	0.9419	83.719
50-100-0.4	0.9375	127.930	0.9554	102.320	0.9387	71.516
50-100-0.6	0.9419	71.625	0.9557	99.687	0.9617	87.542
100-25-0.2	0.9387	86.522	0.9536	102.100	0.9536	88.517
100-25-0.4	0.9569	91.906	0.9404	99.109	0.9569	86.497
100-25-0.6	0.9474	127.930	0.9527	118.820	0.9474	77.719
100-50-0.2	0.9554	71.625	0.9479	79.587	0.9554	148.610
100-50-0.4	0.9554	64.563	0.9477	88.597	0.9557	65.750
100-50-0.6	0.9557	74.860	0.9519	67.497	0.9574	66.484
100-100-0.2	0.9574	102.000	0.9471	100.280	0.9375	70.048
100-100-0.4	0.9375	71.625	0.9441	69.547	0.9474	151.890
100-100-0.6	0.9587	88.547	0.9479	66.287	0.9678	120.550

註：* 表迭代數－母體數－精英策略選取範圍

表 3. 使用率變化之變異數分析

變異來源	自由度	平方和	均方	F 值
迭代次數(S)	2	11.2774	5.7854	18.70**
母體染色體數(P)	2	14.6375	7.1987	24.79**
精英策略選擇率(R)	2	16.2488	5.4322	24.38**
S×P	4	0.5004	0.1057	0.61
S×R	4	0.8491	0.3148	0.38
P×R	4	1.1867	0.3867	0.71
S×P×R	8	1.4097	0.2769	0.49
誤差	78	26.8438	0.2897	
總和	104	687.4918		

註：** 表在 α＝0.01 下，自變數在各參數水準間有顯著差異

績效可謂非常傑出。但若觀察分片數超過 50 片例題之演算結果：例題 27~38，則 GA 演算法幾乎皆略遜於國際文獻之執行結果。因此，本研究所提出之 GA 演算法非常適合用於求解中、小題型之單原片方形物件排列問題。

表 4. 參數在不同水準下使用率變化

自變數	樣本數	使用率變化	Duncan Grouping*
迭代次數**			
25	27	0.9472	B
50	27	0.9490	B
100	27	0.9515	A
母體染色體數**			
25	27	0.9472	B
50	27	0.9509	A
100	27	0.9495	A
精英策略選擇率**			
0.2	27	0.9481	B
0.4	27	0.9566	A
0.6	27	0.9452	C

註：* 相同字母表示無顯著差異；
** 表在 α＝0.01 下，自變數在各參數水準間有顯著差異

我們認爲本研究所提出之 GA 演算法中所採用之新的編碼方式，亦即在染色體中系統化地決定每個分片置入之角度（0 度或旋轉 90 度），應當已發揮出預期之效果，而且透過精英策略能留住表現較佳之染色體，故能超越許多文獻演算結果；然而此種編碼方式在中、大題型之問題中，例如 50 分片之例題中，其基因長度爲 100；100 分片之例題中，其基因長度將爲 200，這些過長之染色體將拖垮演算效度及效率，如此觀察也可從表 5 之演算結果比較中得到印證。

五、結論及建議

本研究針對單原片方形物件排列問題求解，以原片使用率最大爲決策目標，建立一有效且迅速之求解方法。由於具備廣泛搜尋及融合各代解答之優點，本研究依然使用 GA 之演算架構，且繼續加強在文獻中尚未被著墨之「菁英策略」，並使用不同之「編碼」設計，希望新的編碼方式搭配菁英策略，能加速 GA 收斂至最終解，獲得最佳或近似最佳之物件排列計畫，並較文獻有更好之演算表現。首先以部分例題對 GA 演算法之參數進行實驗，以期能獲取參數組合建議。接著經由建議之參數組合，對 38 題國際文獻演算例題進行測試，再與國際文獻演算結果進行比較。結果顯示本研究所提出之 GA 演算法，在中、小題型之單原片方形物件排列問題中幾乎皆優於國際文獻最佳結果，某些例題甚至超越文獻結果將近 10% 之使用率，演算績效可謂非常傑出。但在分片

表 5. GA 演算法結果與文獻最佳結果比較

題號	分片數	文獻最佳結果 使用率 (%)	本研究結果 平均使用率 (%)	本研究結果 最佳使用率 (%)	本研究結果 演算時間 (sec)	題號	分片數	文獻最佳結果 使用率 (%)	本研究結果 平均使用率 (%)	本研究結果 最佳使用率 (%)	本研究結果 演算時間 (sec)
1	25	83.470	93.290	95.157	5.512	20	25	94.000	95.300	99.510	4.049
2	20	84.800	92.948	94.268	41.956	21	25	94.000	96.160	96.160	2.197
3	16	80.300	96.000	100.000	95.748	22	28	96.000	98.850	98.850	2.971
4	19	86.400	84.211	84.211	51.908	23	29	96.000	97.090	97.250	8.189
5	50	93.500	90.741	90.741	285.828	24	28	96.000	98.040	98.040	8.423
6	50	91.800	93.825	94.526	375.783	25	49	97.000	94.192	95.238	68.657
7	10	100.000	100.000	100.000	90.313	26	49	97.000	95.440	97.245	91.230
8	15	100.000	99.686	100.000	28.700	27	49	97.000	92.903	95.745	73.215
9	20	100.000	97.157	99.379	73.241	28	73	97.000	95.190	95.240	932.747
10	10	94.800	95.283	96.188	1.583	29	73	97.000	95.700	95.700	635.973
11	10	97.100	93.135	95.586	1.264	30	73	97.000	95.530	95.530	777.175
12	15	92.200	93.797	96.361	3.241	31	97	97.000	94.960	94.960	1048.257
13	15	94.400	94.590	95.372	1.503	32	97	97.000	95.100	95.770	1881.928
14	20	94.600	95.905	97.017	7.069	33	97	97.000	95.480	95.480	1962.082
15	30	N/A	97.794	100.000	12.677	34	196	96.000	93.871	94.094	84217.000
16	16	96.000	98.000	100.000	0.563	35	197	96.000	93.182	93.904	35798.000
17	17	96.000	96.154	96.154	0.547	36	196	96.000	94.875	95.094	71842.000
18	16	96.000	94.118	100.000	0.469	37	40	98.198	95.455	96.042	91.785
19	25	94.000	93.290	95.157	5.512	38	50	97.674	94.989	96.094	272.526

數超過 50 片例題中，GA 演算法幾乎皆略遜於國際文獻之執行結果。因此，本研究所提出之 GA 演算法非常適合用於求解中、小題型之單原片方形物件排列問題。

我們認為本研究所提出之 GA 演算法中所採用之新的編碼方式，亦即在染色體中系統化地決定每個分片置入之角度（0 度或旋轉 90 度），應當已發揮出預期之效果，故能超越許多文獻演算結果；然而此種編碼方式在中、大題型之問題中，過長之染色體將拖垮演算效度及效率。因此後續研究或可針對此種編碼方式提出改善，如何得到效果卻又避免受其害。

參考文獻

1. Babu, A. R. and N. R. Babu (1999) Effective nesting of rectangular parts in multiple rectangular sheets using genetic and heuristic algorithms. *International Journal of Production Research*, 37, 1625-1643.
2. Bartholdi, J. III, J. Vate and J. Zhang (1989) Expected performance of the shelf heuristic for 2D packing. *Operations Research Letter*, 8, 11-16.
3. Beasley, J. E. (1985) An exact two-dimensional non-guillotine cutting tree search procedure. *Operations Research*, 33**,** 49-64.
4. Bennell, J. A. and K. A. Dowsland (1999) A tabu thresholding implementation for the irregular stock cutting problem. *International Journal of Production Research*, 37, 4259-4275.
5. Biro, M. and E. Boros (1984) Network flows and non-guillotine cutting patterns. *European Journal of Operational Research*, 16, 215-221.
6. Chauny, F., R. Loulou, R. Sadones and F. Soumis (1991) A two phase heuristic for the two-dimensional cutting stock problem. *Journal of Operational Research Society*, 42**,** 39-47.
7. Chen, C. S., S. Sarin and B. Ram (1991) The pallet packing for non-uniform box sizes. *International Journal of Production Research*, 29, 1963-1968.
8. Coffman, E. G., M. R. Garey and D. S. Johnson (1984) Approximation algorithms for bin packing- An updated survey. In: *Algorithm Design for Computer Systems Design*, 49-106. G. Ausiello, N. Lucertini and P. Serafini Eds. Springer, Vienna.

9. Dagli, C. H. and M. Y. Tatoglu (1987) An approach to two-dimensional cutting stock problems. *International Journal of Production Research*, 25, 175-190.
10. Dagli, C. H. (1990) Neural networks in manufacturing: Possible impacts on cutting stock problems. Proceedings of Rensselaer's Second International Conference on Computer Integrated Manufacturing, New Jersey, NJ.
11. Dagli, C. H. and A. Hajakbari (1990) Simulated annealing approach for solving stock cutting problem. Proceedings of Rensselaer's Second International Conference on Computer Integrated Manufacturing, New Jersey, NJ.
12. Garey, M. R. and D. S. Johnson (1979) *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman and Company, San Francisco.
13. 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, 34-57.
14. Ismail, H. S. and K. K. B. Hon (1995) The Nesting of 2-dimensional shapes using genetic algorithms. Proceedings of the Institution of Mechanical Engineers Part B-Journal of Engineering Manufacture, Hawaii, HI.
15. Israni, S. S. and J. L. Sanders (1982) Two-dimensional cutting stock problem research: A review and a new rectangular layout algorithm. *Journal of Manufacturing Systems*, 1, 169-182.
16. Israni, S. S. and J. L. Sanders (1985) Performance testing of rectangular parts-nesting heuristics. *International Journal of Production Research*, 23, 437-456.
17. Jacobs, S. (1996) On genetic algorithm for the packing of polygons. *European Journal of Operational Research*, 88, 165-181.
18. Lai, K. K. and W. M. Chan (1997) Developing a simulated annealing algorithm for the cutting stock problem. *Computers and Industrial Engineering*, 32, 115-127.
19. Lamousin, H. J. and W. N. Waggenspack (1997) Nesting of two-dimensional irregular parts using a shape reasoning heuristic. *Computer Aided Design*, 29, 221-238.
20. Leung, T. W., C. K. Chan and D. M. 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.
21. Leung, T. W., C, H. Yung and D. M. Troutt (2001) Applications of genetic search and simulated annealing to the two-dimensional non-guillotine cutting stock problem. *Computers and Industrial Engineering*, 40, 201-214.
22. Lutfiyta, H. and B. Mcmillin (1992) Composite stock cutting through simulated annealing. *Mathematical Computing and Modeling*, 16, 57-74.
23. Parada, V., M. Sepulveda, M. Solar and A. Gomes (1998) Solution for the constrained guillotine cutting problem by simulated annealing. *Computers and Operations Research*, 25, 37-47.
24. Theodoracatos, V. E. and J. L. Grimsley (1995) The optimal packing of arbitrarily-shaped polygons using simulated annealing and polynomial-time cooling schedules. *Computer Methods in Applied Mechanics and Engineering*, 125, 53-70.
25. Wu, T.-H., J.-F. Chen, C. Low and P.-T. Tang (2003) Nesting of two-dimensional parts in multiple plates using hybrid algorithm. *International Journal of Production Research*, 41, 3883-3900.
26. Yanasse, H. H., A. S. I. Zinober and R. G. Harris (1991) Two-dimensional cutting stock with multiple stock sizes. *Journal of Operational Research Society*, 42, 673-683.

收件：**94.10.24**　修正：**94.11.23**　接受：**94.12.16**