

結合啓發式與基因演算法解決 不規則形船體內構件排版問題之研究

邵 揮 洲 陳 建 聲

國立成功大學 造船暨船舶機械工程學系

關鍵詞：船體內構件、排版、啓發式演算法、基因演算法

摘 要

在產品的生產製造過程中，眾多的零組件是在大片的母材上經由排版、切割與加工所產生，如傢俱製造業、切割帆布及皮革的商品、金屬切割業與服飾業...等。排版效率的快慢與母材使用率的多寡直接影響製造之成本。當然，造船業也不例外，且因鋼板的成本較上述的產業來的高且加工不易，所以對於排版問題更值得研究與探討。

本研究的目的是為了解決不規則形船體內構件的排版問題，目標為減短排版設計的時間並增加切割原料的使用率。本研究方法為：先將不規則形船體內構件排版問題簡化成矩形構件排版問題。並提出一個新的啓發式演算法，稱為B-L-S演算法以作為矩形構件排放的方法，並結合基因演算法以尋求最佳的排版設計。本研究完成之後，可提供一個尋求排版最佳化的設計與構件排版的环境，以期能加速初步排版設計的時間，提升排版作業的效率。

前 言

在鋼船的製造過程中，船體內構件是經由設計、放樣、排版、切割所產生。其中，排版效率的快慢與母材使用率的多寡直接影響製造之成本。大體而言，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，蕭惟隆[15]於1997年發展出一套電腦輔助船體內構件排版及數值加控制加工系統，提供船體內構件排版與NC切割的環境。而自動排版方面，可分為兩個方面，一為從矩形原料中切割出矩形外形的構件，另一個為從矩形原料中切割出不規則外形的構件。

本研究中所討論的排版問題為不規則形構件排版問題，所以發展出一套自動排版系統為目標。由於船體內構件是屬於不規則外形的構件，若是要進行自動化排版，則必須要將其外形簡化[2]，簡化的目的乃在於將不規則形構件簡化成多邊形構件。簡化成為多邊形之後，雖然可以直接以外圍於每一個構件的最小

矩形代替多邊形構件，但是此方法會大大的降低切割母材的使用率，改善的方式為先挑選任意兩個多邊形構件並計算出此二構件最緊密結合位置的多邊形構件排版簇(polygon nesting clusters)，接著以最小外圍於排版簇的矩形代替排版簇(rectangle nesting clusters)，如此一來，便將不規則形構件排版問題簡化成矩形構件排版問題。在進行排版工作之前，必須先將構件與切割母材的資料建立，以便利構件排版時所須進行編碼的工作以及方便檢查構件有無重疊的現象產生。

結合啓發式演算法與基因演算法求解，建立完成構件與切割母材資料之後，採用基因演算法為求解最佳化排版設計的工具，蓋因於基因演算法在求解的過程中與問題無關，只須給予目標函數並設定好限制條件，其本身可藉由複製、交配與突變不斷的演化，一直產生更好的子代(結果)。目標函數為切割母材的使用率，並以基因演算法決定出矩形構件放置的程

序。Jakobs, S.[8]在1996年的提出B-L演算法(Bottom-Left algorithm)，並結合基因演算法求解排版設計。Dagli, C.H.與Poshyanonda, P.[3]在1997年提出運用類神經網路系統處理矩形構件排版問題。Ramesh, A.B.與Ramesh, N.B.[13]在1999年提出法結合基因演算法與啟發式排版解決排版不同的矩形構件於多個切割母材上的問題。E. Hopper與B. Turton[7]在1999年將S. Jakobs的B-L演算法改進為B-L-F演算法(Bottom-Left-Filling algorithm)，並產生較高的排版效率。將B-L演算法(Bottom-Left algorithm)加以研究並改進成為B-L-S演算法(Bottom-Left-Sliding algorithm)，作為矩形構件放置的演算法。其中，基因演算法與啟發式演算法為互動關係，因為當基因演算法經由編碼產生矩形構件排版程序之後，接著由啟發式演算法依據基因演算法所傳入的矩形構件排版程序，依序的進行排放矩形構件於切割母材上並產生基因演算法所需的目標函數值，而基因演算法則根據此目標函數值再進行新的一代演化，直至目標函數達到設定值為止。

排版問題與排版流程

在不規則形構件排版問題的研究中，有將構件以特定的方式放置於切割母材上的啟發式演算法，在這一個方面的研究有Albano A.[1]提出改善二維排版的方法。Qu, W.與Sander, J. L.[14]提出一套可以處理複雜構件的排版法。Prasad, Y.K.D.V.[12]提出二維不規則金屬薄板之切割研究。Cheng, C.Y.與Atkinson, J.[2]提出一個改善蒙地卡羅法的啟發式演算法。Li, Zhenyu與Li, Milenkovic[11]在提出緊密與分離演算法。Lee, K.J.和Lee, J.K.[10]提出空間排版法。Lamousin, H.J., Waggenspack, W.N.與Dobson, G.T.[9]研究將複雜的不規則外形的構件圖樣排版至不規則的邊界中。Han, G.C.M.與Ing, S.J.N.[6]在提出使用神經網路與模擬退火處理二維不規則形構件排版問題。

在產品的生產製造過程中，排版效率的快慢與母材使用率的多寡直接影響製造之成本。

如果欲在切割構件時想要有最大的切割母材使用率，則需要事先將構件排版至切割母材上，並且構件與構件之間不可以發生重疊的現象且構件亦不可以超出切割母材的區域。圖1是根據構件外形、切割程序與切割母材的種類所建立的二維構件排版問題分類圖。由於船體內構件的外形屬於不規則形，而切割母材為矩形的鋼板，且船體內構件數量繁多，並非單一塊切割母材所可以完全排放，所以，對於船體內構件排版問題，將其歸類為不規則外形構件與多個矩形切割母材的問題。在本文中，提出以結合啟發式與基因演算法的方式解決排版不規則外形構件於多個切割母材的問題。

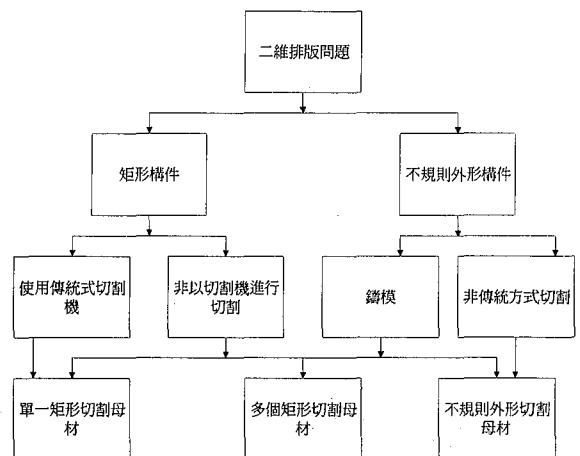


圖1 二維構件排版問題分類圖

船體內構件排版問題為排版不規則外形構件與多個矩形切割母材的問題。在本文中，內構件排版進行的方式步驟如下：

1. 簡化構件外形：將不規則形構件簡化成只含直線元素的多邊形構件。
2. 建立多邊形排版簇：選定任意兩個構件，並找出外圍於此二構件的最小外圍多邊形(minimal convex enclosure polygon) [5]，並以此最小外圍多邊形代替此二構件，稱為多邊形排版簇(polygon nesting cluster)。
3. 建立矩形排版簇：以最小外圍於多邊形排版簇的矩形代替多邊形排版簇，稱為矩形排版簇(rectangle nesting cluster)。

如此一來，便將複雜的不規則外形構件與多個切割母材排版問題簡化成矩形構件與多個切割母材排版問題。將不規則外形構件簡化之後，接著便要進行矩形構件排版。在進行矩形構件排版之前，要先決定出矩形構件排版的順序。以基因演算法替代傳統的試誤法，因為基因演算法為非決定性的最佳化搜尋機制，其求解的過程與問題無關，可以提供出全域性的最佳化解。所以，透過基因演算法，將構件作編

碼(coding)，並透過複製(reproduction)、交配(crossover)與突變(mutation)進行演化(稱為一代演化)，以產生更優良的子代(結果)。有了構件放置的順序之後，接著要討論構件放置的方法，將左下演算法進行改良，提出一個啟發式演算法，稱之為左下滑行演算法(bottom-left-sliding algorithms, B-L-S algorithms)，以增進排版的效率。當以基因演算法決定出矩形構件排版的順序之後，接著由啟發式演算法依據基因演算法所傳入的矩形構件排版程序，依序的進行排放矩形構件於切割母材上並產生基因演算法所需的目標函數值，而基因演算法則根據此目標函數值再進行新一代演化，直至達到目標函數值為止。詳細的排版流程圖可見圖2。

簡化排版問題

由於船體內構件的外形十分的複雜，其所包含的圖形元件可能為點、線、弧...等。因此，若是要進行自動排版的程序，則需要將不規則形船體內構件簡化成多邊形構件[2]。在本研究中，將不規則形船體內構件簡化成多邊形構件的方式如圖4所述，產生的多邊形即為簡化後的多邊形，且此多邊形必定為凸多邊形。

如圖3(a)所示，P為一個不規則形構件，其簡化的步驟如下：

1. 如圖3(b)所示，選定參考點為 $C_0 (X_0, Y_0)$ ，其中， C_0 為X軸座標最大的一個點，以 C_0

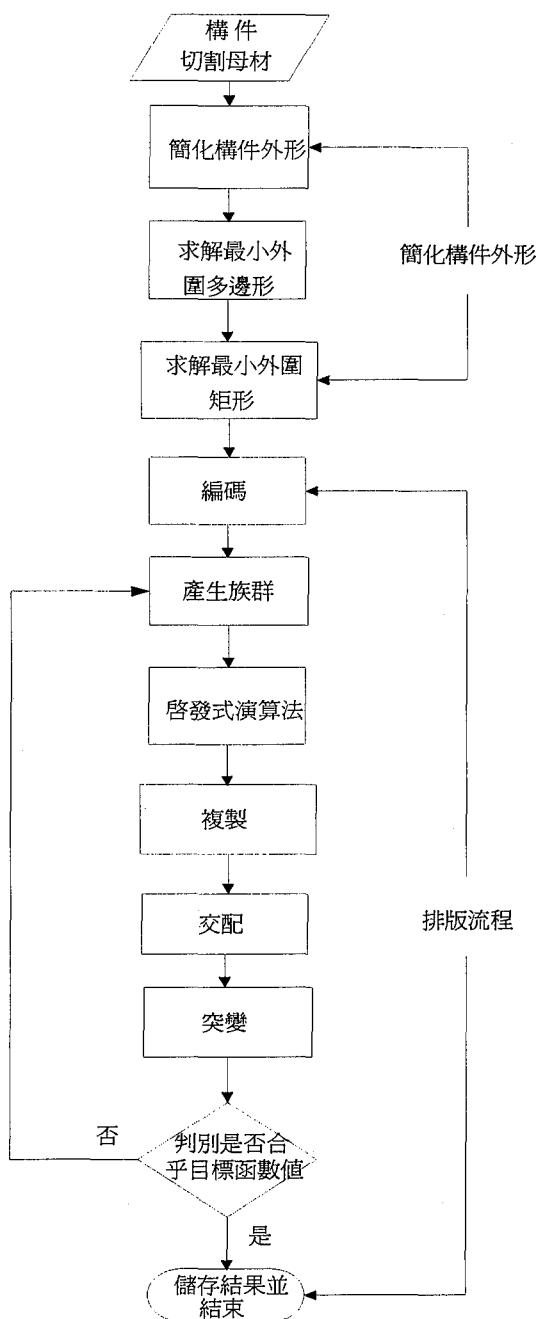


圖2 排版流程圖

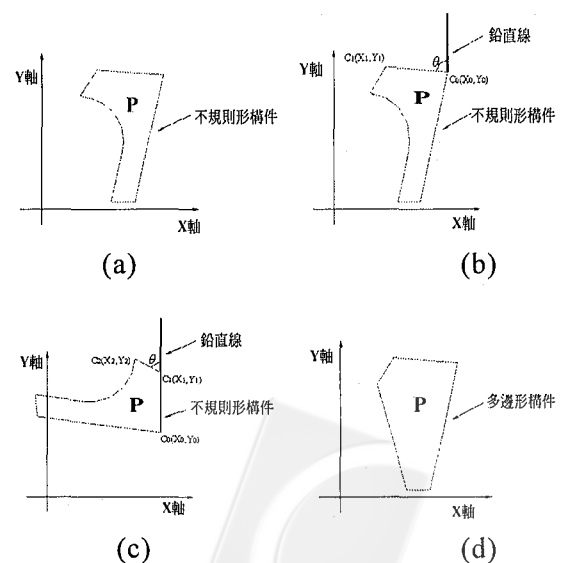


圖3 不規則形船體內構件圖簡化

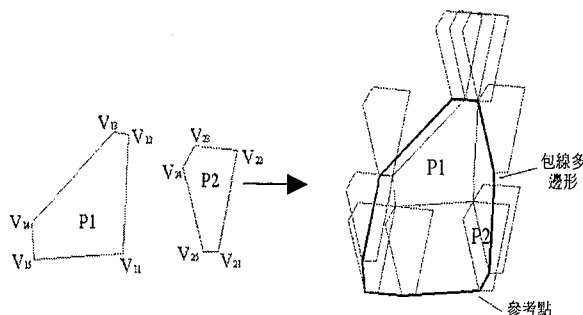


圖4 包線多邊形

為起點，作一條鉛垂線 $\{X=X_0, Y_0 \leq Y < \text{finite}\}$ 。

2. 以 C_0 為圓心，以逆時鐘方向轉動此鉛垂線，一直到碰觸圖形上其他的點為止，紀錄轉動的角度 θ ($0 < \theta < 2\pi$) 與碰觸點 $C_1(X_1, Y_1)$ 。
3. 如圖3(c)所示，以 C_1 為圓心，順時鐘方向轉動整個圖形 θ 角 ($0 < \theta < 2\pi$)，所以 C_1 變成圖形最右邊的一個點。

重複步驟 1. 2. 3.，直到下述的結束條件為止。

結束條件：假設由步驟1. 2. 3.所產生的點為 $C_i \{i=1.2.3....\}$ ，若 $C_{i+1} = C_0$ ，則停止所有的步驟，並將 C_{i+1} 由 C_0 取代，因為 $C_{i+1} = C_0$ 代表所產生的新點 C_{i+1} 即為參考點 C_0 ，表示已繞行圖形一圈，又重回參考點，此時應停止重複步驟1. 2. 3.的動作，並進行下一個步驟。

如圖3(d)所示，將 $C_i \{i=0.1.2.3....\}$ 連接，則產生的多邊形即為簡化後的多邊形，且此多邊形必定為凸多邊形。

將不規則形構件簡化成多邊形構件之後，雖然可經由尋找出外圍於每一個構件的最小矩形予以取代，但是會發生切割損失過大的問題。因此，在本研究中，以 Grinde, R.B. 和 Cavalier, T.M.[5]所提出的尋求外圍於任意兩構件的最小多邊形的方法，建立多邊形排版簇，最後，再以外圍於每一個多邊形排版簇的最小矩形取代多邊形排版簇，並將此最小外圍矩形稱為矩形排版簇。如此一來，不規則形船體內構件排版問題便可以簡化為矩形構件排版問題。

由圖4呈現出決定包線多邊形的方式，以 P_2 繞著 P_1 所產生的包線多邊形(envelope)。包線多邊形的意義為： P_2 繞著 P_1 轉，且 P_2 所有的頂點皆無出現在 P_1 中，亦即是 P_1 與 P_2 無重疊的現象產生，且 P_1 與 P_2 無分離的情況發生，即 P_1 與 P_2 保持點與點、點與線或線與線的接觸關係。

求得包線多邊形之後，我們可以肯定的結論出 P_2 對於 P_1 的最佳的放置位置必位於包線多邊形的某一個位置[5]。因此由尋找外圍面積，並以此決定最佳位置。從圖4中，如果我們將頂點 V_{12} 與頂點 V_{23} 、頂點 V_{15} 與頂點 V_{24} 連接，即可得到外圍於 P_1 與 P_2 的外圍多邊形。在此，我們以外圍於 P_1 與 P_2 的最小多邊形作為判定的準則，當產生了最小的外圍面積時，則我們定義 P_2 的位置為最佳的放置位置。如此一來，不規則形船體內構件排版問題便可以簡化為矩形構件排版問題。

排版演算法

為了有效的將各種不同的矩形排版簇排版至矩形的切割母材上，在本研究中，提出結合啟發式演算法與基因演算法處理矩形構件排版問題，且此二個演算法為互動的關係。

所謂搜尋，係只在一既定之問題條件下，於有限的可行解中找出一特定的答案，此答案可滿足所定之目標函數。而由可行解所構成的集合，稱為解空間(solution space)，目標函數的訂定，則可能為搜尋最大或最小值。傳統中常見的搜尋方法可分為三大類，分別為微積分(calculus)、列舉(enumerative)與隨機(random)。傳統的搜尋法不易求得問題的最佳解或是缺乏效率，故本研究中，以基因演算法作為最佳的搜索工具。基因演算法是由 John Holland 於 1975 所提出的，主要是因其具有特別的生物進化計算模式。它可以表示為一種一般化的搜尋技術，並且具有平行化的特性及避免落入區域最佳化的能力，如同一個搜尋演算法經由自然的選擇及自然的遺傳方式，它結合了生存適合度的字串結構，及透過隨機的交換資訊以搜尋可能的解空間，並且每一個世代都會產生出新的更適合於問題的解集合。

本研究採用以整數型的方式將構件與切割母材進行編碼，以產生排版程序的字串，以整數作為基本單位，不需要再轉換。此種型態適用於排程問題、旅行推銷員問題。在基因演算法中，完成編碼之後，接著便是以隨機的方式產生初始族群，初始族群的大小對於基因演算法搜尋的時間有很大的影響。產生初始族群之後，排版演算法需利用啟發式演算法在切割母材上產生排版結果。

當基因演算法經由編碼產生矩形構件排版程序之後，接著便需要一個排放矩形構件的運算法以排放矩形構件於切割母材上並產生矩形構件的排版結果，並將此結果再回傳至基因演算法中，以作為判定是否符合所求的目標函數值或是再經由複製、交配與突變進行下一代的演化。關於排放矩形構件的方法有許多，最常被使用的矩形構件演算法乃由Jakobs, S. [8]於1996年所提出的B-L演算法(bottom-left algorithm)，其優點乃可以快速的決定出構件放置的位置，且方便進行程式撰寫。在本研究中，提出一個增進B-L演算法的排版效益的啟發式演算法，稱之為B-L-S演算法，茲分述此二演算法如下

B-L 演算法

假設構件數目有 n 個，放置於切割母材上的步驟為：

1. 放置第一個構件：

首先，將第一個構件放置於切割母材上最左下角的位置。

2. 放置第 i 個構件：

依序的放置第 i ($i=2,3,\dots,n$)個構件於切割母材上，放置時，從切割母材的右上方開始，然後盡可能的將構件移動至最下方，接著盡可能的將構件往左邊移動，最後將構件移動至最下方。

B-L-S 演算法

在本研究中，根據S. Jakobs[8]所提出B-L演算法的精神，提出一個改善B-L演算法的啟

發式演算法，稱之為B-L-S演算法(bottom-left-sliding algorithm)，與B-L演算法不同處，及是會將構件滑動，尋找切母材上可排入位置，如此具備B-L演算法優點，更增加其排入構件的位置，其示意圖可見圖5。其方法如下：

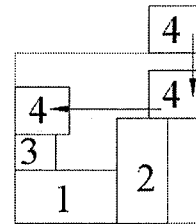


圖5 B-L-S演算法示意圖

首先，將B-L-S演算法中所需要使用到的資料陣列定義：

1. 定義切割母材資料陣列：定義出每一切割母材的長寬。L(切割母材的長度)、B(切割母材的寬度)
2. 定義構件資料陣列：定義出所有構件資料，PLAN(構件長度，構件寬度，排入點的X座標值，排入點的Y座標值)
3. 定義可排入點陣列：定義出於母材上，可排入點的位置與允許排入大小。TP(X座標值，Y座標值，允許排入的長度，允許排入的寬度)
4. 外圍輪廓線陣列：母材的左下邊界線與構件排入後，每一構件的右上邊界線，外圍輪廓線找出可排入點位置。LL(X座標值，Y最小值，Y最大值)、BB(Y座標值，X最小值，X最大值)
5. 內側輪廓線陣列：母材的右上邊界線與構件排入後，每一構件的左下邊界線，內側輪廓線判定可排入點所允許排入構件大小。LX(X座標值，Y最小值，Y最大值)、BX(Y座標值，X最小值，X最大值)

假設構件數目有 n 個，放置於切割母材上的步驟為：

1. 讀取排版資料：
2. 讀取切割母材資料：
3. 設定初始條件：
 - (1) 建立初始外圍輪廓線(母材)。
 - (2) 建立初始內側輪廓線(母材)。
 - (3) 建立初始的構件排入點，即為原點。示意圖可見圖6。
4. 讀取構件資料。
5. 依序將構件排入每一排入點尋找第一順位可排入點。
6. 當確定可排入後，
 - (1) 將資料寫入構件陣列。
 - (2) 產生排入點之對角點POS(X, Y)，以作為投影點。
 - (3) 將此排入點之許可之X,Y值定為(-1,-1)表示已排入。示意圖可見圖7。
7. 新增外圍輪廓線與內側輪廓線。示意圖可見圖8。
8. 將所有POS點對所有外圍輪廓線作投影，求出投影點座標，此投影點座標新增為可排入點。因為由左至下方式排入，因此可排入點必為構件的右上邊界，由此可找出可排入點位置。示意圖可見圖9。
9. 將所有可排入點作排序，並將重複的可排入點剔除。
10. 重新尋找每一排入X Y允許值(已排入點除外)
 - (1) 將所有排入點對最近內側輪廓線作投影。
 - (2) 求出投影垂足距離。
 - (3) 垂足距離為排入X Y允許值。

每一排入點其可允許排放大小，為排入點到母材的右上邊界，或是構件左下邊界距離，因此由此可找出排入構件長寬允許值。示意圖可見圖10

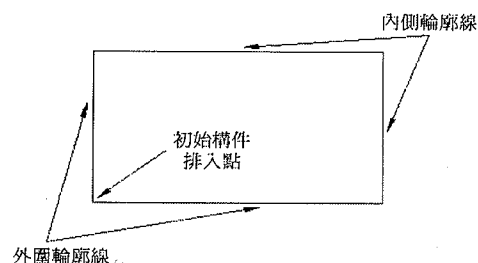


圖6 B-L-S演算法示意圖(一)

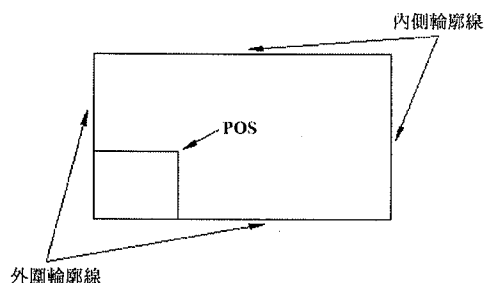


圖7 B-L-S演算法示意圖(二)

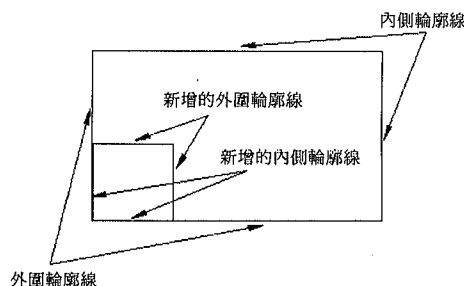


圖8 B-L-S演算法示意圖(三)

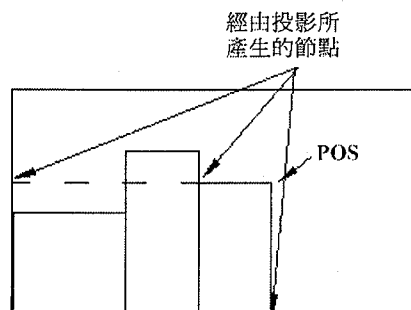


圖9 B-L-S演算法示意圖(四)

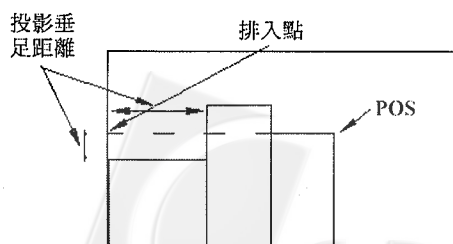


圖10 B-L-S演算法示意圖(五)

- 11.將此構件紀錄為排入切割母材的構件。
- 12.重複步驟5至步驟11即可排完一個切割母材。
- 13.重複步驟2至步驟12即可排完所有構件。

完成構件放置之後，接下來的程序為計算目標函數。當構件排放於切割母材之上時，其所要達成的目標為最大的切割母材使用率(最小切割損失)。在基因演算法中，總是選擇產生較高的目標函數值的字串並經由基因運算子進行運算以作為產生下一個族群的依據。在本研究中，目標函數為F，

其中，Mi為第i個切割母材的面積，Ai為第j個構件的面積

$$F = \sum_{i=1}^n \left[\left(\sum_{j=1}^n A_j \right) / M_i \right]$$

系統驗證

實例測試的目的在於呈現構件在切割母材上的放置結果，程式中的目標函數為切割母材的使用率，目標在於求出最佳的排版設計。

測試實例介紹：從七美輪載客船中選取出29個不規則形船體內構件作為實例測試的對象(如圖11所示)，所有的構件皆由繪圖軟體AutoCAD所繪製產生，並以DXF作為檔案交換格式。

由於本研究中所使用的基因演算法軟體是以Microsoft Excel為運算的背景，所以必須將欲排版的矩形構件資料輸入於Microsoft Excel中，圖12為將七美輪之29個船體內構件以本研究中所提出簡化構件的方法進行簡化後的結果，圖13為將資料建立於Excel後的結果，包括構件名稱、構件面積與構件的點座標。圖14為多邊形排版簇、矩形排版簇、矩形排版簇及其組成構件的資料表，包括矩形排版簇的長度、寬度與面積及組成構件的的名稱與面積。此外，在進行矩形構件排版之前，必須要考慮切割構件時所會發生的變形，所以在本文中，對於每一個矩形構件均留一公分的餘裕，以防止切割時所產生的變形影響到其他的構件切割。

將不規則形構件簡化成矩形排版構件之後，接著便需要將矩形構件進行編碼。在本研

究中，排版問題是屬於排程問題的一種，所以採用整數型的編碼方式。完成矩形構件編碼之後，接著由基因演算法產生排版構件的程序字串並由啟發式演算法進行排放矩形構件於切割母材上並記錄每一個構件放置的順序與所在母材的編號。

在基因演算法中影響結果的因素很多，在本研究中針對不同族群大小Ps、交配機率Pc與突變機率Pm對於切割母材使用率進行最佳化。根據Jakobs, S.[8]及Ramesh,A.B.與Ramesh, N.B.[13]文中所用的交配機率Pc=0.6、突變機率Pm=0.4作為出始的搜尋參數對族群大小(20~300)進行最佳值搜索，以尋找出適合於本案例的族群大小，其結果如表1所示。由表1得知族群大小在Ps=50時，可以得到最大的切割母材使用率。

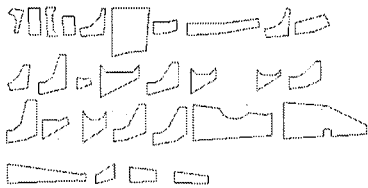


圖11 七美輪之船體內構件圖

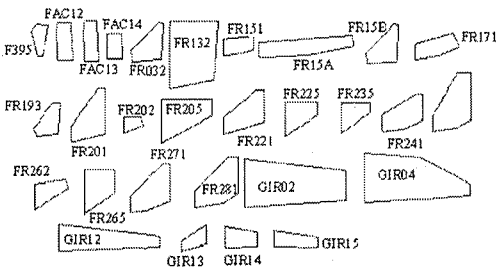


圖12 簡化後的船體內構件圖

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	構件名稱	構件面積	X1	Y1	X2	Y2	X3	Y3	X4	Y4	X5	Y5	X6	Y6
2	FR95	943.2	0	0	20.203	2.278	-35.283	-52.49	-22.622	-55.181	-12.449	-55.177	0	0
3	FR12	2046.211	0	0	-2.503	47.317	-28.586	65.113	-30.029	0	0	0	0	0
4	FR13	1304.494	0	0	-3.271	15.642	-38.749	1.323	28.014	0	0	0	0	0
5	FR14	1594.905	0	0	-1.173	46.447	-51.123	-46.447	-30.028	0	0	0	0	0
6	FR15	258.658	0	0	-5.013	1.528	-49.589	-49.114	-62.494	-69.077	-3.533	-66.511	0	0
7	FR16	1017.171	0	0	35.935	0	59.975	-123.477	-4.472	-88	0	0	0	0
8	FR17	1654.949	0	0	0	26.591	-40	49.728	-40	-10.567	-53.734	-10.248	-53.269	-10.994
9	FR18	4766.949	0	0	-2.943	19.569	-29.669	9.181	-33.669	-20.819	-35.944	-15.644	0	0
10	FR19	2857.652	0	0	-4.946	1.558	-43.313	-30.214	-61.325	-70.156	-3.422	-66.697	0	0
11	FR20	2994.391	0	0	-11.061	22.171	-37.285	-13.467	-37.285	-22.368	-49.516	-15.725	0	0
12	FR21	1913.272	0	0	-6.506	0	-54.062	-46.782	-43.966	-62.517	-5.145	-56.828	0	0
13	FR22	3697.406	0	0	0	33.496	-15	33.496	-10.021	15.543	-69.021	-14.491	-61.696	-14.323
14	FR23	778.404	0	0	-4.472	15.301	-34.376	14.973	-38.376	-15.528	0	0	0	0
15	FR24	7949.869	0	0	0	31.592	-38.692	31.592	-38.692	-11.193	0	0	0	0
16	FR25	2611.597	0	0	0	9.313	-17	9.313	-41.605	4.671	-61.605	-21.139	0	0
17	FR26	2112.934	0	0	0	23.608	-49.943	23.608	-49.943	-17.722	0	0	0	0
18	FR27	1395.654	0	0	0	17.691	-29.674	17.691	-29.674	-16.721	0	0	0	0
19	FR28	3057.176	0	0	0	47.783	-12.5	47.783	-30.023	8.762	-50.678	-21.135	-37.263	-35.961
20	FR29	4700.421	0	0	0	37.634	-27.5	37.634	-26.282	5.817	-16.282	-20.143	-21.854	-12.308
21	FR30	6117.491	0	0	-4.262	15.937	-67.013	1.653	-67.013	-10.524	0	0	0	0
22	FR31	2558.671	0	0	0	-62.025	-59.691	-62.025	-59.691	-39.374	0	0	0	0
23	FR32	3621.638	0	0	0	49.432	-15	49.432	-40.458	4.791	-50.458	-23.229	-47.333	-55.595
24	FR33	4584.467	0	0	0	66.973	-20	66.973	-35.574	12.223	-35.574	-26.797	-27.793	-18.654
25	FR34	1202.875	0	0	0	34.568	-101.143	71.463	-101.143	-11.066	0	0	0	0
26	FR35	1391.109	0	0	0	21.319	-101.071	70.779	-101.071	-21.149	-12.383	0	0	0
27	FR36	6117.237	0	0	0	18.809	-101.142	41.809	-101.142	-1.336	0	0	0	0
28	FR37	1394.694	0	0	0	31.847	-43.827	13.467	-43.827	-11.565	0	0	0	0
29	FR38	1518.312	0	0	0.024	27.735	-46.942	36.315	-46.942	-11.053	0	0	0	0
30	FR39	1594.932	0	0	0	18.591	-32.295	30.526	-32.295	2.818	0	0	0	0

圖13 構件資料表圖

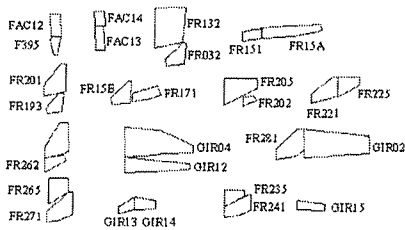


圖14 多邊形排版簇圖

矩形構件的長度														
矩形構件的寬度														
矩形構件的編碼														
切割母材的編碼														
切割母材的寬度														
切割母材的長度														
Parts Nesting														
Parts Data														
Assign to Sheet														
Order														
Blank area														
1	81.6	134.49	105	1	550	350	201	0.67032531	301					
2	76.2882	155.266	113	2	350	300	202	0.31196117	302					
3	69.7926	155.072	102	3					100					
4	34.567	128.117	107	4					100					
5	32.5996	118.985	111	5					100					
6	98.463	181.584	106	6					100					
7	150.373	81.5786	108	7					100					
8	99.5565	88.0099	114	8					100					
9	211.469	141.203	103	9					100					
10	114.185	50.4554	110	10					100					
11	79.3007	84.4415	109	11					100					
12	85.2976	30.5826	112	12					100					
13	284.94	96.5807	101	13					100					
14	145.562	80.6659	115	14					100					
15	241.124	51.0658	104	15					100					

圖15 矩形構件與切割母材編碼圖

表1 族群大小與切割母材使用率之資料表

交配機率Pc=0.6 突變機率Pm=0.4					
族群大小 Ps	20	50	100	200	300
切割母材使用率	0.9886521	0.9934502	0.9919403	0.9905135	0.987679

再以族群數大小Ps=50，突變機率Pm=0.4對各交配機率(Pc=0.4~0.9)進行最佳值搜尋，其結果如表2所示。由表2得知交配機率Pc=0.6時，可以得到最大的切割母材使用率。

表2 交配機率與切割母材使用率之資料表

族群數 =50 突變機率Pm=0.4						
交配機率Pc	0.4	0.5	0.6	0.7	0.8	0.9
切割母材使用率	0.9916534	0.9914256	0.9934191	0.9908207	0.9923494	0.9924395

最後以族群大小Ps=50，交配機率Pc= 0.6對突變機率(Pm=0.1~0.6)進行最佳值的搜尋，其結果如表3所示。由表3得知突變機率Pm=0.25時，可以得到最大的切割母材使用率。

表3 突變機率與切割使用率之資料表

族群數 =50 交配機率Pc=0.6						
突變機率 Pm	0.1	0.15	0.2	0.25	0.3	0.35
切割母材使用率	0.9905135	0.9873574	0.9907375	0.9934502	0.9923705	0.9914594
突變機率 Pm	0.4	0.45	0.5	0.55	0.6	
切割母材使用率	0.9904896	0.9918914	0.9916985	0.9921941	0.9908207	

綜合上述的結果，可以得知當基因演算法的參數選定族群大小Ps=50、交配機率Pc=0.6與突變機率Pm=0.25時，應該會有最佳解的產生。

本研究以歸納出的基因演算法之參數為族群大小為50、交配率為0.6與突變率為0.25進行測試，相關的基本資料如下：

- 系統環境：Microsoft Windows 95作業系統。
- 程式運算背景：Microsoft Excel軟體。
- 基因演算法軟體：Evolver基因演算法軟體。
- 切割母材數目：2個，尺寸為350*500。
- 矩形構件數目：30個，因為船體的構件大多為兩兩對稱，所以在本實例中，將前面所產生的15個矩形排版簇變成30個矩形構件排版簇。
- 族群數大小Ps：50。
- 交配率(Crossover Probability)：0.6。

L M N O									
矩形構件的長度									
矩形構件的寬度									
構件置於母材上的編號號碼									
Sheet	Sheet	Part	Part	Result					
size-length	size-width	size-length	size-width						
201	350	500	211.4615	141.2033	1	29860			
202	550	350	79.3007	84.4415	1	66962			
203			284.9298	96.5807	1	27519			
204			69.7926	155.072	1	108229	104		
205			98.463	181.5837	1	178793	105		
206			81.6	134.4904	1	109744	106		
207			76.2182	155.2657	1	118449	107		
208			241.1242	51.0658	1	123132	108		
209			34.567	128.117	1	442862	109		
210			99.5565	88.0099	2	876156	110		
211			145.5623	80.6659	1	117419	111		
212			32.5996	118.9932	2	38792	112		
213			114.185	50.4554	2	576125	113		
214			150.3725	81.5786	2	122672	114		
215			85.2976	30.5826	1	260162	115		

圖16 啟發式演算法之資料輸出圖

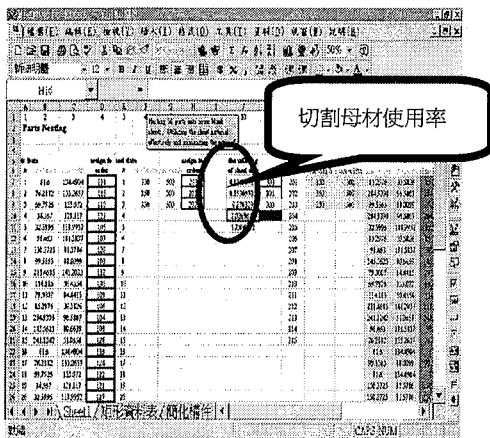


圖17 使用B-L-S運算法求得之最佳排版設計
資料表圖(一)



圖18 使用B-L-S運算法求得之最佳排版設計
資料表圖(二)

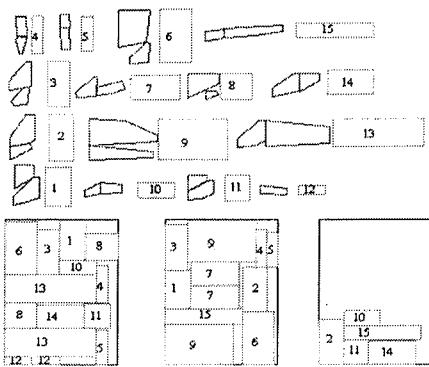


圖19 矩形構件排版簇之排版結果圖

- 收斂條件：設定切割母材使用率在疊代2000次之後，其改變值若無超出1%以上，即達到收斂條件。

經由Evolver求解後，可以看出經由基因演算法將排版設計進行演化之後，可以提高切割母材的使用率。最後達到收斂條件資料整理如下：

族群數：50 交配率：0.6 突變率：0.25			
	使用率	置入構件數	備註
切割母材1	0.896132421	14	已排滿
切割母材2	0.876024805	11	已排滿
切割母材3	0.124961632	5	未排滿

最後，將排版的結果以AutoCAD作為輸出驗證的平台，圖19為矩形構件排版簇之排版結果。

結 論

本論文建立了一系列的程序與方法以建立了船體內構件的排版流程，透過排版問題的純化與分析將船體內構件排版的工作程式化。依照整個研究的過程，歸納幾點結論如下：

1. 構件簡化方面：

對於不規則外形船體內構件排版的問題，提供一個簡化構件外形的方法，以方便後續的排版工作與簡單化排版問題。

2. 啟發式運算法：

構件放置的方法有許多種，其中最常被使用的的方法乃由Jakobs, S.[8]於1996年提出的B-L演算法，本研究根據B-L演算法的基本精神提出B-L-S演算法，其排版效益較B-L演算法為高，可以提高切割母材的使用率。

3. 基因演算法：

在搜尋最佳化的方法中，基因演算法是相當可靠的搜尋工具，在本文中，完成了以基因演算法找尋最佳化的排版設計的程式化，並將排版結果呈現。

未 來 展 望

對於本研究來說是還有很大的發展空間，在此對於本論文完成後未來的相關研究方向與建議敘述如下：

- 突變率(Mutation Probability)：0.25。
- 目標函數：切割母材使用率。

1. 由於船體內構件外形相當複雜，所以需將構件圖形進行簡化，使得構件圖形面積亦增大，造成鋼板(切割母材)的使用率相對的減少，故需改進對構件外形簡化的計算方式，以增加構件間排版的緊密度。
2. 在本研究中，提出一個新的啟發式演算法：B-L-S演算法，其排版效益雖較B-L演算法為高，然而其尚有改善的空間，未來可朝發展更有效率的啟發式演算法的方向研究。
3. 對於鋼板切割路徑的考量，尚未能融入於系統中，無法產生NC碼以進行NC切割，未來可朝此方向進行研究。

參 考 文 獻

1. Albano, A., "A Method to Improve Two-Dimensional Layout," *Computers-Aided Design*, Vol.9, pp.48-52(1977).
2. Cheng, C.Y. and Atkinson, J., "A Comparison of Some Methods for Computer-Aided Nesting of Sheet Components," *Journal of Materials Processing Technology*, Vol.44, pp.311-318 (1994).
3. Cihan, H.D. and Pipatpong, P., "New Approaches to Nesting Rectangular Patterns," *Journal of Intelligent Manufacturing*, Vol.8, pp.177-190 (1997).
4. Cihan, H.D. and Pipatpong, P., "Effective Nesting of Rectangular Parts in Multiple Rectangular Sheets Using Genetic and Heuristic Algorithms," *International Journal of Production Research*, Vol.37, No.7, pp. 1625-1643 (1999).
5. Grinde, R.B. and Tom M. Cavalier, "A New Algorithm for the Minimal-Area Convex Enclosure Problem," *European Journal of Operational Research*, Vol.84, pp.522-538 (1995).
6. Han, G.C.M. and S.J.N. Ing, "Two-Stage Approach for Nesting in Two-Dimensional Cutting Problems Using Neural Network and Simulated Annealing," *Proc. Instn. Mech. Engrs.*, Vol.210, pp. 509-519 (1996).
7. Hopper, E. and B. Turton, "A Genetic Algorithm for a 2D Industrial Packing Problem," *Computer & Industrial Engineer*, Vol.37, pp.375-378 (1999).
8. Jakobs S., "Theory and Methodology on Genetic Algorithms for the Packing of Polygons," *European Journal of Operational Research*, Vol.88, pp.165-181 (1996).
9. Lamousin, H.J., W.N. Waggenspack, Jr. and G.T. Dobson, "Nesting of Complex 2-D Parts within Irregular Boundaries," *Journal of Manufacturing Science and Engineering*, November, Vol.118, pp.615-622 (1996).
10. Lee, K.J. and J.K. Lee, "A Spatial Scheduling System and its Application to Shipbuilding: DAS-CURVE," *Expert System with Applications*, Vol.10, No.3/4, pp.311-324 (1996).
11. Li, Z. and M. Li, "Compaction and Separation Algorithm for Non-Convex Polygons and Their Applications," *European Journal of Operational Research*, 84, pp.539-561 (1995).
12. Prasad, Y.K.D.V., "A Set of Heuristic Algorithm for Optimal Nesting of Two-Dimensional Irregularly Shaped Sheet-Metal Blanks," *Computers in Industry*, Vol. 24, pp.55-70 (1994).
13. Ramesh, A.B. and N.B. Ramesh, "Effective Nesting of Rectangular Parts in Multiple Rectangular Sheets Using Genetic and Heuristic Algorithms," *International Journal of Production Research*, Vol.37, No.7, pp.1625-1643 (1999).
14. Weishuang, Q. and L.S. Jerry, "A Nesting Algorithm for Irregular Parts and Factors Affecting Trim Losses," *INT. J. PROC. RES*, Vol.25, No.3, pp.381-397 (1987).
15. 蕭惟隆, "電腦輔助船體內構件排版及數值控制加工系統", 國立成功大學造船及船舶機械工程研究所碩士論文 (1997)。

STUDY ON NESTING OF IRREGULAR SHAPE SHIP STRUCTURES USING HEURISTIC AND GENETIC ALGORITHMS

H.J. Shaw J.S. Chen

Department of Naval Architecture and Marine
Engineering, National Cheng Kung University
Tainan, Taiwan, ROC

*Keywords: Ship Structures, Nesting, Heuristic
Algorithm, Genetic Algorithm*

ABSTRACT

Cutting of two-dimensional shaped parts from a large sheet with minimum wastage of sheet is an important task in many industries. This task exists in

the textile industry, furniture industry, press working, etc. Certainly, it also exists in shipbuilding industry. We should put more emphasis on nesting problems in shipbuilding industry because the cost of steel sheets is more expensive than materials of other industries described above.

The aim of this thesis is to solve irregular shape ship structures nesting problems in shipbuilding industry. At the beginning, we simplify irregular shape ship structures nesting problems to rectangle nesting problems. We develop a new heuristic algorithm called B-L-S algorithm (bottom-left-sliding algorithm) for placing parts into sheets. We take genetic algorithm as a tool to search the best solution. Finally, we combine heuristic and genetic algorithms to solve nesting problems. After finishing this thesis, we wish to provide a way to solve ship structures nesting problems and improve the efficiency of nesting works.

(Manuscript received Sep. 21, 2001,

Accepted for publication Feb. 25, 2002)

