

應用螞蟻分群模式進行零件家族形成

An Ant-based Clustering Algorithm for Part Family Formation

高有成¹ 李岳倫²

摘要

針對少量多樣化的生產模式，單元製造是最有效率的生產系統。而群組技術是單元製造系統常用的設計方法，其需要一個有效的零件分群方法，來進行製造單元初步設計。集群分析法為常用的零件分群方法之一，搭配特殊的相似係數，能集群相似度高的零件成為零件家族。集群分析法可分為階層式分群與非階層式分群兩種，但階層分群法容易產生鏈結效應，而非階層分群事先需給定零件家族數目。故在本研究中提出一個以人工螞蟻分群模式(AntClust)為基之零件分群演算法，來解決上述問題。本演算法利用螞蟻演算法的特性：群體性與隨機性，使零件分群結果不會在分群過程中過早決定，降低資料本身特性之影響。另外，本演算法不必預先給定群數，可直接利用螞蟻自我組織能力形成自然群組。本演算法已經被開發成軟體系統，並利用零件分群績效指標，對六個文獻案例進行測試，都得到相當好的單元形成績效。

關鍵字：群組技術、零件家族形成、螞蟻演算法、群體智慧

Abstract

Cellular Manufacturing is one of the major applications of group technology. It requires an effective part clustering approach for preliminary manufacturing cell design. One of famous approaches is the cluster analysis method, which uses similarity coefficients and clustering methods to group similarity parts into part families. Clustering methods are divided into two categories: hierarchical and nonhierarchical methods. Hierarchical methods often suffer from chaining effects, while nonhierarchical methods need a predetermined cluster number. The research proposes a part clustering algorithm that is based on an artificial ant clustering model (AntClust). The algorithm utilizes the characteristics of ants, congregation and randomness, to prevent grouping results from being fixed during clustering processes and to reduce the effects of noisy data. Besides, the algorithm doesn't need a predetermined cluster number. The algorithm has the ability of self-organization to form part families naturally. The algorithm has been developed into a software system. Six literature problems were selected to test the proposed algorithm with respect to grouping efficacy. We found that the algorithm is able to obtain better machine cell configurations

¹大同大學資訊經營研究所副教授

²大同大學資訊經營研究所研究生

than other approaches.

Keywords: Group technology, part family formation, ant algorithms, swarm intelligence

1. 簡介

近年來由於全球競爭越來越激烈、產品生命週期逐漸縮短，形成中、小型批量的生產模式漸漸取代了以往大批量生產的模式。而滿足此種中、小批量生產模式的單元製造系統(Cellular Manufacturing Systems, CMS)，及其相關理論，如群組技術(Group Technology, GT)，也因此廣泛地被研究及討論。

群組技術是一種製造的哲學，利用零件之間在設計與生產上的相似性，將相似零件視為同一類別或集結成群，形成零件家族(Part Family)；且對每個零件家族，相對形成一個製造單元(Manufacturing Cells)，使得具有相似特性之零件家族，儘量在同一機器單元中加工，以減少機器之間搬運的浪費。群組技術之效益為降低準備時間(Setup Time)，降低生產流程時間(Production Flow Time)，降低工廠的存貨數量，以及簡化生產流程，所以隨著少量多樣化生產模式的日益重要，群組技術也愈受到重視。單元形成(Cell Formation)是群組技術重要應用之一，也是單元製造系統在初步規劃中決定製造系統佈置時非常重要的步驟。該問題統稱單元形成問題(Cell Formation Problem, CFP)，包括零件分群與機器分群兩部分。

單元形成方法主要可歸類為以下幾種方法(Selim) [16]：1.描述性程序法、2.圖形分割法、3.人工智慧法、4.數學規劃法、5.集群分析法(Cluster Analysis)。其中集群分析法為常見的方法，包括階層法(Hierarchical Methods)與非階層法(Nonhierarchical Methods)。階層法主要缺點在分群過程中，其分群結果的固定化，使得早期錯誤分群的資料無法重新指派，而未能得到最佳組態。另外，階層法並未正式考慮到誤差和變異來源，故易受到異常資料所影響(黃俊英)[2]。所以從工廠蒐集到的機器零件投射矩陣，在經過傳統分群法進行零件群聚分析後，其重排矩陣(re-arranged matrix)之資料分佈並不好，容易造成零件家族與機器單元的分群效益不佳。而非階層法針對上述的缺點進行改進，其中以K組平均法(K-means)為代表。K-means 較不受例外零件存在之影響，且具有重新分群機制的特性，故能較階層式方法求得較佳的分群效果，但是其分群的結果容易受到初始中心點的設計的影響，有可能會找到次佳解而不是最佳解(Selim et al.)[19]。此外在應用K-means方法於CFP問題時，必須預先指定零件家族數，這將對製造單元設計者造成困擾，這也是本方法的缺點之一。

為解決上述問題，本研究提出一個以人工螞蟻分群模式(AntClust)為基之零件分群演算法，作為新的製造單元設計方法。本論文第二節將介紹螞蟻演算法，第三節提出螞蟻零件分群演算法，第四節開發一個螞蟻零件分群系統，第五節進行實例說明與比較性研究，最後一節提出本研究之結論。

2. 螞蟻演算法

過去十幾年來，已經有許多研究學者投入群體智慧(Swarm Intelligence)這個領域，他們藉由觀察自然界生物、昆蟲的生活習性，獲得了很多的啟發。像是昆蟲社會中他們所擁有集體行為(Collective Behavior)、自我組織(Self-Organization)的能力等，皆成為群體智慧研究者爭相研究的主題(Bonabeau et al)[4]。其中人工螞蟻演算法算是較為成功的主題，可分為螞蟻族群最佳化(Ant Colony Optimization, ACO)和螞蟻分群模式(Ant Colony Clustering)兩種模式。前者為一極新的 Meta-heuristic 方法，最常見的是用於解決組合最佳化問題(combinatorial optimization problems)，請見 Dorigo and Stützle [7]之著作，而後者為本研究採用之模式，介紹如下。

Deneubourg et al. [8] 是最早利用螞蟻撿起和放下的兩種簡單行為提出螞蟻分群模式。在該模式中，所有的物件與螞蟻先被隨機的放在 2D 的棋盤內，每一個棋盤格裡最多只能有一個物件。之後，每一隻螞蟻隨機的在棋盤上移動，空手螞蟻會撿起異於週遭環境的物件，而持有物件的螞蟻會放下與週遭環境相似的物件，來達到物件分群的效果。另外 Deneubourg 也賦予人工螞蟻記憶的功能，紀錄著螞蟻最近遇到物件的種類與數量。所以螞蟻記憶中的相同物件數量比例，決定了其目前週遭環境的物件相似程度，並被利用來決定撿起與丟下物件的機率值大小。Deneubourg 模式只能針對兩類物件分群，因此 Lumer and Faieta[11]改進其模式，能夠對多類別物件進行分群，稱之為 LF Model。後來 Monmarche[15]結合 LF Model 和 K-means 方法進行資料分群，並且證明了該混合式分群法比單獨使用螞蟻分群法或 K-mean 分群方法之結果為佳。

最近 Labroche[12][13][14]提出異於 LF Model 的螞蟻分群演算法，叫作 AntClus 模式。他觀察自然界中的螞蟻都擁有一種氣味散佈在表皮，其氣味會根據螞蟻本身以及其所在的環境而不斷改變。當螞蟻彼此遭遇時，假如雙方是相似的，她們將會分享相同的氣味，亦即將彼此視為相同巢穴的成員。其演算法一開始將要分群的物件資料指派給螞蟻，亦即每一隻螞蟻都代表一個物件。再來模擬自然界中的螞蟻兩兩相互接觸，演算法中稱這個過程叫“遭遇”。作者觀察真實螞蟻相互接觸導致氣味改變的行為，提出六種遭遇行為法則，作為兩隻人工螞蟻遭遇時，改變螞蟻各自狀態的依據。在連續隨機遭遇過程中，人工螞蟻呈現自我組織(Self-Organization)的能力：藉由隨機遭遇與彼此交互作用，使得相似的螞蟻最後會擁有同樣氣味，進而達到物件分群的效果。

以上兩種人工螞蟻分群模式皆擁有彈性、穩定性、非中央式控制和自我組織等特性，而這些特性皆可解決複雜的資料分群問題。為了改善傳統群集分析方法的缺點，傅士誠 [1]首先嘗試利用螞蟻分群演算法，來解決製造單元形成問題。其採用的是 LF Model，並且證實 LF Model 較傳統方法得到較好的分群績效值。本研究欲利用另外一種的螞蟻分群模式 AntClus，提出新的零件分群方法，希望和 LF Model 一樣能改進傳統零件分群方法的缺點，並得到最佳的製造單元組態結果。

3. 螞蟻零件分群演算法

本研究以AntClust演算法為基礎，嵌入零件相似係數進而提出新的螞蟻零件分群演算法來解決零件家族形成問題。本演算法分群概念主要分為四個階段：初始階段，遭遇階段，雜群合併階段以及家族形成階段，如圖1所示。以下將以四階段來介紹該演算法流程之步驟。

3.1. 初始階段

初始階段包括三個步驟，主要是對演算法四個參數，包括Label, M , M^+ 和Template，進行初始值設定，我們先介紹四個參數，再說明初始階段的三個步驟。

Label 代表螞蟻身上的氣味，也就表示螞蟻歸屬群之標記， $label = 0$ 表示此螞蟻尚未被歸屬。 M 代表該螞蟻歸屬群大小的估計值，值越大表示該螞蟻的歸屬群中的螞蟻成員越多。當螞蟻進行遭遇過程時，若二者 label 值相同(表同群)，則增加二者之 M 值，否則減少二者之 M 值。 M^+ ：代表該螞蟻歸屬群中，與其相似螞蟻數量多寡的估計值，

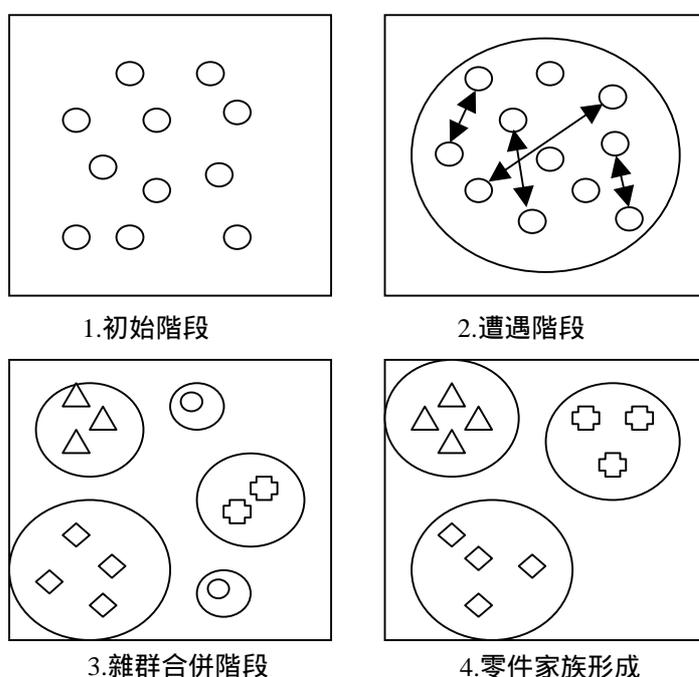


圖1 零件分群演算法四階段

值愈大表該螞蟻所歸屬的群愈穩固。螞蟻進行遭遇過程時，若二者同群且相似，則增加二者之 M^+ 值，否則減少二者之 M^+ 值。

在此 M 及 M^+ 值都介於 0~1 之間，若以 X 代表 M 或 M^+ 值，則增加或減少 X 值的公式分別如公式(1)及公式(2)所示：

$$X_{new} = (1 - \alpha) \times X_{old} + \alpha \tag{1}$$

$$X_{new} = (1 - \alpha) \times X_{old} \tag{2}$$

α ：為一個常數值，此處由Labroche [12]建議 $\alpha = 0.2$ 。

Template 為判斷兩隻螞蟻是否為接受(acceptance)狀態的閾值，兩隻螞蟻互相接受，表示他們是相似的。每隻螞蟻都有其各自的 Template 值，我們以 T_i 表示第 i 隻螞蟻的

$Template$ 值，也是介於 0~1 之間。 $Template$ 值並不是固定的，會隨著遭遇過程依公式(3)不斷調整：

$$T_i = \frac{\overline{Sim(i,...)} + Max(Sim(i,...))}{2} \quad (3)$$

在此， $\overline{Sim(i,...)}$ 是螞蟻 i 和過去所遭遇過的螞蟻相似度之平均值，而 $Max(Sim(i,...))$ 是螞蟻 i 和過去所遭遇過的螞蟻中相似度最大的值。在遭遇過程中的兩隻螞蟻，分別以公式(4)進行接受判斷。當判斷公式(4)為真時，則螞蟻 i 和螞蟻 j 視為接受狀態，表示兩隻螞蟻認定彼此是相似的。

$$(Sim(i, j) > T_i) \wedge (Sim(i, j) > T_j) \quad (4)$$

在此， $Sim(i, j)$ 是螞蟻 i 和螞蟻 j 之間的相似度。而相似係數公式採用 Islam and Sarker [9]所提出之公式(見公式(5))，因其為專門為零件分群所設計的相似係數，較一般的距離公式更能精確的判斷零件間的差別。

$$Sim(i, j) = \frac{a + \sqrt{a \times d}}{a + b + c + d + \sqrt{a \times d}} \quad (5)$$

在此， a ：兩個零件都需要的機器數量； b ：零件 i 需要而零件 j 不需要的機器數量； c ：零件 j 需要而零件 i 不需要的機器數量； d ：兩個零件都不需要的機器數量。

初始階段的三個步驟說明如下：

步驟一 將要分群的每一個零件指派給一隻螞蟻，即 N 個零件就有 N 隻螞蟻。

步驟二 將每一隻螞蟻的四個參數歸零，包括 $Label$ 、 M 、 M^+ 和 $Template$ 。

步驟三 進行 $Template$ 值初始化。初始化過程中，每一隻螞蟻會與其他螞蟻進行 L 次的隨機遭遇，每遭遇一次，就利用公式(3)改變其 $Template$ 值。所以其 $Template$ 值會從零開始不斷調整，調整 L 次後的值就是該隻螞蟻 $Template$ 的初始值。此處 Labroche [13]建議 $L = 75$ 次即可。在本步驟中， $Label$ 、 M 、 M^+ 仍保持為 0。

3.2. 遭遇階段

這一階段中，自全部螞蟻中隨機挑選兩隻螞蟻使其互相遭遇，同時依據利用真實螞蟻特性所推出的六個遭遇法則(見表 1)，改變螞蟻身上的三個參數： $Label$ 、 M 、 M^+ 。而各螞蟻的 $Template$ 值在每次遭遇後依公式(3)繼續調整，學習成一個穩定且適當的值。在表 1 中， $A(i,j)$ 表相遭遇的兩隻螞蟻為接受狀態， $No A(i,j)$ 表相遭遇的兩隻螞蟻為非接受狀態，即彼此不相似。

當法則一發生時，兩個相似且還沒有歸屬群的螞蟻會產生一個新的群，在遭遇階段初期時，法則一會不斷產生只有兩隻螞蟻成員的新群。

當法則二發生時，表示兩隻螞蟻相似，但其中一隻還沒有歸屬群，因此會和有歸屬群的螞蟻歸併為同一群。遭遇階段初期時，可利用法則二來使小群逐漸壯大。

當法則三發生時，表示同一群且相似的螞蟻互相遭遇，故提升螞蟻的 M 及 M^+ 值，使其群更加穩固。

法則四發生時，表示同一群但不相似的螞蟻互相遭遇，故增加這兩隻螞蟻的 M 值，降低 M^+ 值。調整完 M 及 M^+ 後，需將 M^+ 值較小的螞蟻自該群剔除，並將其參數歸零 ($Label=M=M^+=0$ ，但 $Template$ 不需要歸零)。因為 M^+ 值較小的螞蟻，表示該螞蟻在同群中遇到相似螞蟻的機率較低，這種相似度不足的螞蟻，可能因為在遭遇階段初期被不適當的歸屬，或是螞蟻本身為例外零件，和群中大部分螞蟻成員都不太相似，故將該隻螞蟻自該群中剔除。剔除後的螞蟻已和原歸屬群無關聯，所以將 $Template$ 之外的三個參數歸零。

因為 M^+ 表示同群且相似螞蟻的遭遇次數，而 M 表示同群內螞蟻遭遇次數，故法則五發生時，表示相似但不同群的螞蟻互相遭遇，因此必須減少 M 及 M^+ 的值。且因為這兩隻螞蟻是相似的，所以在調整完 M 及 M^+ 後，將 M 值較小的螞蟻歸併到 M 值較大的螞蟻的群內。在遭遇階段中期，此法則可以產生大群逐漸合併小群的效果。

演算法在不斷經由螞蟻的互相遭遇，讓法則三、四、五產生交互作用，將可逐漸形成一群一群穩固且適當的較大群組。最後判斷是否符合迭代次數 N_{Iter} ($N_{Iter} = 250 * N$)。公式中的 250 是由實驗結果建議的係數， N 為螞蟻數目。當符合迭代次數，則進入下一階段；若否，則繼續遭遇循環。

表 1 遭遇期間的六個法則

法則編號	是否接受	螞蟻狀態條件	參數調整
1	A(i,j)	$label_i = label_j = 0$	$label_i = label_j = new$
2	A(i,j)	$label_i = 0 \wedge label_j \neq 0$	$label_i \leftarrow label_j$
3	A(i,j)	$(label_i = label_j) \wedge (label_i \neq 0) \wedge (label_j \neq 0)$	增加 M_i, M_j, M_i^+, M_j^+
4	No A(i,j)	$(label_i = label_j) \wedge (label_i \neq 0) \wedge (label_j \neq 0)$	增加 M_i, M_j , 減少 M_i^+, M_j^+ $label_x \leftarrow 0, M_x \leftarrow 0, M_x^+ \leftarrow 0$ $(x M_x^+ = \text{Min}_{k \in [i, j]} M_k^+)$
5	A(i,j)	$label_i \neq label_j$	減少 M_i, M_j , 將 M 值較小者 $label$ 值變成 M 值較大者 $label$ 值
6	上述情況都不發生		不做任何改變

3.3. 合併雜群階段

在遭遇階段會產生一些只有一個螞蟻成員的群，稱為雜群。雜群形成原因可能是有些螞蟻代表特殊零件，不屬於任何一群，或是早期被歸屬到不適當的群經由法則四而被剔除，因而產生自己為一群的情況。在演算法中我們希望將雜群螞蟻歸併到與其最相似的大群，以降低群組數目。故我們計算非雜群的中心值，該中心值是個概念上的範本零件，具有該零件家族重要特徵。然後依各雜群到各非雜群中心值距離遠近，將視為雜群的螞蟻合併到與其最相似的非雜群內。本階段結束時，將剩下一些完整且相似的群。

3.4. 零件家族形成階段

在此階段中，我們可以讓工廠規劃者有彈性地決定零件群數。一般合併雜群階段產生的群數往往大於工廠規劃者的需求，在此階段我們將會計算每一群的中心值，將兩個中心值較為相似的群合併為一群，如此逐一合併，直到滿足工廠規劃者所需要的群數為止。

4. 螞蟻零件分群系統

本節中將本研究提出的演算法開發成螞蟻零件分群系統，可將其應用於單元形成問題上。在Shafer and Rogers[17]之研究中，曾建議零件家族的集群方法會比機器集群方法更佳，理由為零件家族集群方法在配置重複機器於有需求之單元時，會有較大的彈性，故本研究採用先形成零件家族再進行機台配置的方式。本系統可分成三大模組，包括零件家族形成模組、機台指派模組與群組績效測量模組，底下將介紹此三大模組之運作。

在第一個模組先載入初始矩陣，將零件資料指派給螞蟻，初始化各螞蟻參數後，進入遭遇階段，遭遇階段後產生初步零件分群結果，再來進行雜群合併產生精煉後的零件分群結果，最後依照工廠規劃者所需做零件家族合併，產生符合工廠規劃者所需的理想零件家族數目。

經過零件家族形成模組後會產生最終零件家族，我們將其輸入到系統的機台指派模組進行機器單元的配置，機器單元的配置是利用啟發式的分派法則，其步驟如下所示：

步驟一：讀取零件家族資料，

步驟二：依序指派機器到零件家族，以例外零件最小化規則為基礎進行指派，也就是比較機器在哪一零件家族中的加工次數最多，便會指派至其零件家族。如果加工次數相同時，便在其中選取該機器未加工零件數目最少之零件家族進行指派。

步驟三：當所有零件家族都產生相對應的機器單元，且所有機器都配置完成後，就重排機台零件初始矩陣以形成機台零件重置矩陣，我們將此矩陣輸入到群組績效測量模組。

最後進入群組績效評估模組，本研究將採用四種評估分群效益的準則產生最後績效測量結果，分別為：總聯結力(TBE)、例外元素(EE)、機器利用率(MU)、群組效益(GE)(Shafer and Rogers [18])。除EE指標外，其他指標都是相對越高越好。

5. 案例說明及比較研究

本節將針對本研究所提出的螞蟻零件分群演算法，進行系統的開發，並以文獻實例來驗證整個系統的求解效益，本系統採用Java 語言撰寫開發。

5.1. 案例說明

為了說明本研究所開發的螞蟻零件分群演算法之效益，本研究將採用一個文獻例題進行測試(Stanfel) [20]，此例題亦為第5.3節6個例題中之例題2，共有14種不同類型的機器，24項不同類別的零件，如表2所示。為了說明的方便，在此將根據本系統的三大模組依序作說明。

在零件家族形成模組中，我們將以螞蟻零件分群演算法的四個階段來做說明，首先在第一階段初始階段以表2為螞蟻零件分群系統輸入的初始矩陣，欄數表示零件編號，列數表示機器編號。再來將每一個零件指派給一隻螞蟻，將所有參數歸零($Label=M=M^+=Template=0$)，之後利用公式(3)進行 $Template$ 初始化，其後就進入第二階段遭遇階段。遭遇階段中隨機挑選兩隻螞蟻進行遭遇，依遞迴公式計算共需6000次。在遭遇期間每隻螞蟻的 $Template$ 、 M 及 M^+ 值都不斷的學習調整。最後每個零件將會擁有其歸屬群，我們將 $Label$ 號碼相同的螞蟻聚在一起，形成了初步的零件分群結果，如表3所示。在第三階段合併雜群階段，將雜群合併到最為相近的群中。表3中有三個雜群：群3,4及7，故我們將其合併至其他非雜群，結果如表4所示，故此階段最後所得群數為四群。在第四階段，零件家族形成階段，我們可以讓工廠規劃者有彈性地決定零件群數。在此假設工廠規劃者希望形成四組製造單元，所以本階段不需要進行零件家族合併。

在機器配置模組中，將在零件家族形成模組產生的零件家族(見表4)作為機器配置的依據，並根據例外零件最小化法則進行機台指派，最後可得到重新排列後的輸出矩陣(見表5)。在群組績效評估模組中，利用前階段輸出的重排矩陣來做評估，並透過四種評估法則可以得到此分群之後的效益評估值：TBE值為74、GE值為67.1%、MU值為68.6%、EE值為3.3%。從分群績效評估結果中得知，本研究所提出的方法確實能有效解決單元形成的問題，在各分群效益上得到很好的表現值。

表2 機器與零件的初始關係矩陣

Machines	Componets																							
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1						1	1																	
2							1	1																
3						1	1	1										1					1	
4				1	1																			
5				1	1																1			
6				1																				1
7				1	1																1			1
8	1	1															1		1	1			1	
9	1	1															1		1				1	
10	1	1					1										1		1				1	
11								1	1	1	1		1	1	1									
12								1	1		1		1	1	1									
13					1			1			1	1		1										
14								1		1		1		1										

表3 初步零件分群結果

Group	Components
1	1, 2, 17, 20, 23
2	3, 21, 24
3	4
4	5
5	6, 7, 8, 18
6	9, 10, 11, 12,13, 14,15, 16,22
7	19

表4 精煉後的零件分群結果

Group	Components
1	1,2,17,19,20,23
2	3, 4, 21, 24
3	6, 7, 8, 18
4	9,5,10,11,12,13,14,15,16,22

表5 機台零件重置矩陣

Machines	Componets																							
	24	3	4	21	14	10	22	16	12	9	11	13	15	5	7	8	18	6	20	1	23	17	2	19
6	1	1																						
7	1	1	1																					
5		1	1	1																				
4		1	1	1																				
13								1	1		1	1												
14									1	1	1	1												
11					1	1	1	1	1	1	1		1											
12					1	1	1	1	1	1			1	1										
1																1			1					
3																1	1	1	1			1		
2																1	1							
10															1				1	1		1	1	
9																			1	1	1	1	1	
8																			1	1	1	1	1	1

表6 過去文獻最佳解與本系統做比較

問題	參考文獻	機器 x零件	文獻最佳解				本系統			
			TBE	GE	MU	EE	TBE	GE	MU	EE
1	King [10]	5x7	14.0	82.4	82.4	0.0	14.0	82.4	82.4	0.0
2	Stanfel [20]	14x24	74.0	67.1	68.6	3.3	74.0	67.1	68.6	3.3
3	Burbidge [5]	16x43	143.0	54.3	65.1	23.4	143	54.5(+)	64.2(-)	21.8(+)
4	Shafer and Rogers [18]	20x20	56.0	43.9	57.0	34.5	63(+)	44.4(+)	56.2(-)	32.2(+)
5	Shafer and Rogers [18]	20x20	167.0	36.0	73.0	58.5	173(+)	39.5(+)	76.7(+)	55.1(+)
6	Chandrasekharan[6]	40x100	578	84.0	91.2	8.6	578	84.0	91.2	8.6

5.2 比較研究

為驗證系統對單元形成之成效，本研究將採用6個文獻例題進行測試。矩陣大小從5台機器與7個零件(5X7)至40台機器與100個零件(40X100)不等，已知理想群數為2群至10群，是參考Shafer and Rogers [18]、Askin et al.[3]和Chandrasekharan [6]三篇文獻之整理。利用上述四項群組績效指標，分別算出本研究最終矩陣的求解值與過去文獻最終矩陣的求解值，作一比較與整理，如表6所示。表中數字旁有”+” 號表示有改善，”-” 號表示結果變差，無符號表示和過去求解值相同。針對表6的數據比較中，可以知道本演算法對於較大型的問題，或是矩陣本身資料結構較複雜的問題，能夠比過去文獻方法找到相同或更佳的求解值。

6. 結論

本研究提出新的人工螞蟻零件分群演算法來解決機器單元形成問題。過去單元形成問題若採用傳統階層式分群求解，早期錯分的零件沒有機制可以重新分群，而且容易受零件資料分布不均的問題所干擾；若採用傳統非階層式分群求解，則必須事先給予數個機器單元數目，再以試誤法反覆找出最好的機器單元組態。人工螞蟻分群模式有自我組織能力，所以事先不必給預定的零件家族數目，其又具有群體性與隨機性，使早期被錯誤分群的零件可以在遭遇過程中重新調整分群結果。經過文獻例題測試之結果，證明本論文提出的螞蟻零件分群演算法，確實可以改善傳統方法的缺點，得到與過去文獻相同或更佳的解。未來將與LF Model為基的螞蟻零件分群演算法相互比較，看何者為優。

參考文獻

1. 傅士誠(2004)，「以螞蟻分群演算法設計製造單元」，大同大學資訊經營所碩士論文。
2. 黃俊英(2000)，「多變量分析」，第七版，中國經濟企業研究所。
3. Askin, G., H. C. Steven, B. G. Jeffrey & J. V. Asoo (1991), “A hamiltonian path

-
- approach to reordering the part-machine matrix for cellular manufacturing,” *International Journal of Production Research* , 29(6), pp.1081-1100.
4. Bonabeau, E., M. Dorigo & G. Theraulaz (1999), *Swarm intelligence : from natural to artificial systems*, New York : Oxford University Press.
 5. Burbidge, L. J. (1975), *The introduction of group technology*, New York : John and Sons.
 6. Chandrasekharan, M. P., R. Rajagoplan (1987), “ZODIC-an algorithm for concurrent formation of part-families and machine-cell,” *International Journal of Production Research*, 25(6), pp.835-850.
 7. Dorigo, M. & T. Stützle (2004), *Ant Colony Optimization*, Cambridge, the MIT Press., MA.
 8. Deneubourg, J. L., S. Goss, N., A. Sendova-Franks, C. Detrain & L. Chretien (1991), “The dynamics of collective sorting robot-like ants and ant-like robots,” *Proc. Of the 1st Conf. on Sim. of Adaptive Behavior*, pp.356-363.
 9. Islam, K. Md. S. & B. R. Sarker (2000), “A similarity coefficient measure and machine-parts grouping in cellular manufacturing systems,” *International Journal of Production Research*, 38, pp.699-720.
 10. King, J. R. & V. Nakornchai. (1982), “Machine component group formation in Group Technology--review and extension,” *International Journal of Production Research*, 20, pp.117-133
 11. Lumer, E. & B. Faieta (1994), “Diversity and adaptation in populations of clustering ants,” *Proceeding of the Third International Conference on Simulation of Adaptive Behavior : From Animals to Animats*, MIT Press/ Bradford Books.
 12. Labroche N., N. Monmarche, G. Venturini (2002), “A new clustering algorithm based on the chemical recognition system of ants,” *Proceedings of ECAI 2002*, pp.345–349.
 13. Labroche N., N. Monmarche & G. Venturini (2003), “AntClust:Ant Clustering and Web Usage Mining,” *GECCO 2003*, pp.25–36, Springer-Verlag Berlin Heidelberg.
 14. Labroche N., N. Monmarche, G. Venturini (2003) “Web sessions clustering with artificial ants colonies,” *WWW2003*, May 20-24, Budapest, Hungary. ACM xxx.
 15. Monmarche, N., M. Slimane & G. Venturini (1999), “AntClass: discovery of clusters in numeric data by a hybridization of an ant colony with the Kmeans algorithm,” *Internal Repport, No.213, E3i*, pp.1-21.
 16. Selim, H. M., R. G. Askin & A. J. Vakharia (1998), “Cell formation in group technology : Review Evaluation and directions for future research,” *Computers industrial Engineering*, 34, pp.3-20.
 17. Shafer, S. M. & D. F. Rogers (1993), “Similarity and distance measures for cellular manufacturing. PartI. A survey,” *International Journal of Production Research*, 31, pp.1133-1142.

18. Shafer, S. M. & D. F. Rogers (1993), "Similarity and distance measures for cellular manufacturing. PartII, an extension and comparison," *International Journal of Production Research*, 31, pp.1315-1326.
19. Selim S. Z., M. A. Ismail (1984), "K-means-type algorithms: a generalized convergence theorem and characterization of local optimality," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6, pp.81-87.
20. Stanfel, E. L. (1985), "Machine clustering for economic production, "Engineering costs and production economics," 9, pp.73-81.