邱哲夫、王惠嘉（2019），『結合動態分群人造蜂群演算法之自動化分群系統』，*中華民國資訊管理學報*，第二十六卷，第一期，頁 1-24。

結合動態分群人造蜂群演算法之自動化分群系統

邱哲夫
國立成功大學資訊管理研究所

王惠嘉*
國立成功大學資訊管理研究所

摘要

分群是一種資料探勘技術，它是一種非監督式的學習方法，透過相似度計算，將資料分成不同的群。在分群演算法中，啟發式分群在近年來漸漸受到重視，它指的是運用啟發式演算法或啟發式的概念解決分群問題。相較於目前的一些其方分群方法（如：k-means），啟發式分群似有較好的表現。

一般的分群演算法在實作時，通常需要使用者給予額外的資訊（例如：群數），這些資訊有時候使用者並不容易下決定，因此，若是能讓分群演算法自動決定群數進行分群，對於使用者來說將會更加的便利。鑑於啟發式分群在分群問題的成功，開始有學者嘗試設計可以自動化分群的啟發式分群演算法，像是衍生自基因演算法（genetic algorithm; GA）的 GCUK（genetic clustering for unknown K），以及衍生自粒子群最佳化演算法（particle swarm optimization; PSO）的 MEPSO（Multi-Elitist PSO）。上述這兩種方法雖然可以成功的自動化分群，但卻有著效率不佳的問題，其原因有二：(1)編碼格式設計不佳導致演算法需搜尋之解空間過大，(2)選用之啟發式演算法不一定適合分群問題或其能力不足。

因此，本研究提出一個可以自動化決定群數的自動化分群系統（automatic clustering system; ACS），此系統先使用群數搜尋演算法（cluster range discovery algorithm; CRD）縮減欲搜尋的群數區間，再使用動態分群人造蜂群演算法（dynamic clustering artificial bee colony algorithm; DCABC）進行自動化分群，DCABC 加入了模範策略（model strategy）以克服既有人造蜂群演算法（artificial bee colony algorithm; ABC）的缺點，並透過特別設計的編碼格式，使其可以在分群的時候同時達成決定群數和優化分群品質的功能。

關鍵詞：自動化分群、元啟發式演算法、人工蜂群演算法、模範策略

* 本文通訊作者。電子郵件信箱：hcwang@mail.ncku.edu.tw
2013/05/14 投稿；2014/07/25 修訂；2018/12/18 接受

Chiu, C.F. and Wang, H.C. (2019), 'An automatic clustering system with dynamic clustering artificial bee colony algorithm', *Journal of Information Management*, Vol. 26, No. 1, pp. 1-24.

An Automatic Clustering System with Dynamic Clustering Artificial Bee Colony Algorithm

Che-Fu Chiu
Institute of Information Management, National Cheng Kung University

Hei Chia Wang*
Institute of Information Management, National Cheng Kung University

Abstract

Purpose－This study designs a system that automatically determines the number of groups of clustering. This study refers to the research of past heuristic algorithms and heuristic grouping, and improves the weakness of ABC algorithm, and proposes an exemplary strategy to improve the search performance of the algorithm.

Design/methodology/approach － This study designs an Automatic Clustering System (ACS). ACS uses a Cluster Range Discovery (CRD) algorithm to reduce the search range of cluster number. After that, the ACS uses Dynamic Clustering Artificial Bee Colony algorithm (DCABC) to complete the automatic clustering. DCABC adopts the Model Strategy to overcome the drawback of original artificial bee colony algorithm (ABC). DCABC also designs a brand-new encoding format. Combining this encoding format, DCABC can cluster the data and find the number of clusters simultaneously.

Findings－With the success of meta-heuristic clustering, some researchers tend to design an automatic clustering algorithm with meta-heuristic method. The experiment results show the proposed DCABC can find the suitable cluster number and can have better performance than ABC.

Research limitations/implications－Although the algorithm proposed in

* Corresponding author. Email: hcwang@mail.ncku.edu.tw
2013/05/14 received; 2014/07/25 revised; 2018/12/18 accepted

this study can automatically determine the appropriate number of groups, at the time of initialization, the user must specify the group number interval.

Practical implications – This study proposes an Automatic Clustering System by using a Cluster Range Discovery algorithm to reduce the search range of cluster number. The ACS uses Dynamic Clustering Artificial Bee Colony algorithm to complete the automatic clustering. DCABC adopts the Model Strategy to overcome the drawback of original artificial bee colony algorithm.

Originality/value – Unlike the similar algorithms which needs to assign number of cluster manually. This study proposes an Automatic Clustering System (ACS), which can automatically determine the number of clusters. Combining the designed encoding format, DCABC can cluster the data and find the number of clusters simultaneously.

Keywords: automatic clustering, meta-heuristic clustering, artificial bee colony algorithm, model strategy

壹、導論

在資料探勘領域，分群是個常見的技術（Liao et al. 2012），分群是從未標籤的資料集中，根據資料彼此間的相似度，將資料分配給數個群，使得群內的資料具有高相似度，跨群的資料則具有低相似度。分群常見的技術可以分成以下三種，階層式分群、分割式分群和密度式分群（Hasan et al. 2009）。其中分割式分群擁有較低的運算成本和不錯的分群結果，因此常常被用到大型資料集的分群（Mahdavi et al. 2008）。分割式分群演算法在實作時，通常是用原型基礎分群法（Ji et al. 2012），原型指的是可以象徵整個群的特定資料集合，以 k-means 演算法來說，資料的原型便是該群的群心。此方法將分群問題視為一最佳化問題，目標式為最小化各群資料點到該群群心的距離。k-means 演算法具有易於實作和低運算成本的特性（Laszlo & Mukherjee 2007），但 k-means 演算法有一個嚴重的缺點，就是容易落入區域最佳解（Likas et al. 2003），因此，許多學者開始嘗試其它的分群演算法，啟發式分群便是其中一種（Cowgill et al. 1999）。

啟發式分群具有良好的結果（Das et al. 2009），但過往的研究大多著重於增進分群的品質，對於自動化決定群數則較少涉略（Das et al. 2008）。部份學者嘗試改變編碼格式，讓演算法可以同時具有搜尋群數和優化分群品質的功能（Bandyopadhyay & Maulik 2002; Das et al. 2008），此類演算法雖然可以找出群數，卻由於其編碼方式導致搜尋空間過於複雜亦缺乏合理性，有著效率不彰的問題。另外，未考慮使用更有效率的啟發式演算法，也是一個原因。另外衍生自基因演算法（genetic algorithm; GA）的 GCUK（genetic clustering for unknown K），以及衍生自粒子群最佳化演算法（particle swarm optimization; PSO）的 MEPSO（Multi-Elitist PSO）。上述這兩種方法雖然可以成功的自動化分群，但卻有著效率不佳的問題，其原因有二：(1)編碼格式設計不佳導致演算法需搜尋之解空間過大，進而造成搜尋效率低落；(2)選用之啟發式演算法為較早期的演算法，不一定適合分群問題或其能力不足。近年來各種新興的啟發式演算法相繼出現，在這些演算法中，人造蜂群演算法（artificial bee colony algorithm; ABC）被套用到各種不同類型的問題上，都被證實具有滿意的結果（Akay & Karaboga 2012; Karaboga & Akay 2009; Szeto et al. 2011; Liang et al. 2017; Song et al. 2017）。分群的目的是將資料分成數群，每群代表了有相似特質的群體，此種研究目前常被應用於公司的管理上，例如：銀行信用卡發卡審核，會利用分群方法來決定發卡額度以及是否發卡，也可以用來偵測不正常刷卡行為（Sabau 2012），另外，於客戶關係管理上，許多公司會利用分群將公司客戶分成不同族群，來進行客戶的需求管理（Wei et al. 2013）。

本研究將以人造蜂群演算法為核心，發展一個自動化決定群數的分群系統。

本研究參考過往啟發式演算法和啟發式分群的研究，針對人造蜂群演算法的弱點進行改良，提出模範策略，藉此增進演算法的搜尋效能，再藉由改變解編碼格式，使演算法可以自動推薦群數並最佳化分群結果。考量到過往的自動化啟發式分群方法，一旦給定的群數搜尋區間增加，解編碼長度亦大幅成長，過大的搜尋空間導致自動化分群效率低落，故本研究提出的自動化分群系統將會在進行自動化啟發式分群之前，使用一個群數搜尋演算法找出較為可能的群數範圍，藉此縮減搜尋之解空間以增進分群演算法的效率。

本研究所提出之系統和演算法雖然可以自動化決定適合的群數，但在初始化的時候，需由使用者給定群數搜尋區間，若資料集之最適群數並未在此區間內，則本系統和本演算法可能無法提供最佳品質的分群結果，因此使用者需提供適當的範圍，若在不確定的情況下，可以給予較大的可能範圍。另本研究使用群心作為整個群的原型（prototype），而在相似度的計算上則是使用歐幾里得距離（Euclidean distance），故本研究提出之系統和演算法僅能找出超球面邊界（hypersphere boundary）的群分割，若資料集之最適群分割不符合此種邊界，則無法找出適合該資料集的分群結果。在符合以上兩個限制下，本研究的成果才能有所表現。

貳、文獻探討

本章將針對本研究使用到的概念和技術進行介紹，讓讀者可以擁有適當的基礎知識，並了解相關領域的發展。以下將依序介紹分群、分群評估指標、啟發式演算法、分割式分群和最佳化問題、自動化啟發式分群等技術。

一、分群

分群問題如下所述。設有一資料集$X = \{x_1, x_2, x_3, \dots, x_n\}$，給定一組群集合$C = \{C_1, C_2, C_3, \dots, C_k\}$，滿足以下性質：

1. $C_i \neq \emptyset$ for $\forall i \in \{1,2, \dots, k\}$
2. $C_i \cap C_j = \emptyset$ for $\forall i \neq j$ and $i, j \in \{1,2, \dots, k\}$
3. $\cup_{i=1}^{k} C_i = U$

常見的分群演算法大概可分成以下三種：階層式分群、分割式分群和密度式分群。階層式分群是根據一樹狀圖（dendrogram），將資料由上往下分裂或是由下往上聚合的分群方法。分割式分群通常會指定群的數目，經過重覆的計算和分配最佳化分群分割，再將資料分配給數個群。密度式分群以資料間的密度作為考量，計算相鄰資料的密度是否高於一門檻值，若是則將這些資料歸入一群，再以資料的連結度擴張群的大小。

二、分群評估指標

（一）Davies-Bouldin Index

Davies-Bouldin Index（Davies & Bouldin 1979）是相當常見的分群評估指標，其公式如下：

$$\text{DB Index(Clustering Result)} = \frac{1}{K}\sum_{i=1}^{K} R_i \tag{1}$$

$$R_i = \max_{j,j\neq i}\left\{\frac{S_i+S_j}{M_{i,j}}\right\} \tag{2}$$

$$M_{i,j} = \left\{\sum_{s=1}^{d}\left|z_{i,s} - z_{j,s}\right|^t\right\}^{\frac{1}{t}} \tag{3}$$

$$S_i = \left\{\frac{1}{|C_i|}\sum_{x_q\in C_i}\sum_{s=1}^{d}\left|x_{q,s} - z_{i,s}\right|^t\right\}^{\frac{1}{t}} \tag{4}$$

$M_{i,j}$為兩群心間的距離，S_i為第 i 群之所有元素到群心的平均距離，x_i為資料點，C_i為一個群，z_i為該群的群心，d 為空間維度，K 為群數，t 為一整數參數。Davies-Bouldin Index 的值愈小象徵著分群的品質愈好。

（二）CS Measure

CS Measure（Chou et al. 2004）最初被應用在影像壓縮的處理，但亦可作為分群評估指標，其公式如下：

$$\text{CS(Clustering Result)} = \frac{\frac{1}{k}\sum_{i=1}^{k}\left[\frac{1}{N_i}\sum_{x_i\in c_i}\max_{x_q\in c_i}\{d(x_i,x_q)\}\right]}{\frac{1}{k}\sum_{i=1}^{k}\left[\min_{j\in k,j\neq i}\{d(m_i,m_j)\}\right]} \tag{5}$$

其中 k 為群數，x_i 為資料點，c_i 為一個群，m_i 為該群的群心，N_i 為該群的資料筆數，$d(x,y)$為歐幾里得幾何距離。CS Measure 的值愈小，象徵著較佳的分群品質。

（三）Silhouette Index

Silhouette Index 是由 Rousseeuw（1987）學者所提出，在 Arbelaitz 等（2013）學者的分群評估指標測試中，Silhouette Index 被證實擁有良好的效果，其公式如下：

$$\text{Silhouette Index(Clustering Result)} = \frac{1}{N}\sum_{i=1}^{k}\sum_{j=1}^{n_i}\frac{b(x_j,c_i)-a(x_j,c_i)}{\max\{b(x_j,c_i),a(x_j,c_i)\}} \tag{6}$$

$$a(x_j,c_i) = \frac{1}{n_i}\sum_{i=1}^{n_i} d(x_i,x_j) \tag{7}$$

$$b(x_j, c_i) = \min_{m \in k, m \neq i} \left\{ \frac{1}{n_m} \sum_{i=1}^{n_m} d(x_i, x_j) \right\} \quad (8)$$

其中 N 為總資料數目，k 為群數，c_i 為一個群，n_i 為該群的資料筆數，x_i 為資料點，$d(x,y)$ 為歐幾里得幾何距離。Silhouette Index 的值愈大象徵分群品質愈好。

三、啟發式演算法

啟發式演算法最早是設計來解決最佳化問題的演算法，此類型演算法通常是觀察自然界的生態或物質的物理特性所產生（Oftadeh et al. 2010）。啟發式演算法大多採用隨機性的搜尋，搜尋過程由一個以上的主要策略引導，使演算法的智慧代理人在逐代之間改進搜尋結果，最終趨向全域最佳解。

啟發式演算法一詞最早由 Glover（1986）所提出，該學者同時也是禁忌搜尋法的發明人。但早在此名詞提出之前，諸多與啟發式演算法概念相近的演算法早已誕生。知名度最高的莫過於基因演算法，該演算法由 Holland（1975）所提出，衍生自達爾文的進化論，模仿物種繁殖時基因交配和突變的過程，該演算法亦成為演化式計算的先驅。群智慧是啟發式演算法重要的分支，此類型的演算法觀察自然界具有社會習性的生物，模仿其生態發展而成，其中較著名的有粒子群最佳化演算法、蟻群最佳化演算法等。

雖然啟發式演算法無法保證解的品質，也不保證能夠找出全域最佳解，但有鑑於其找尋可行解（feasible solution）的能力，以及良好的運算時間，因此經常被拿來套用在各種最佳化問題上，著名的像是旅行銷售員問題，此一經典的 NP-Hard 問題，在使用啟發式演算法後，獲得了不錯的成果（Chatterjee et al. 1996）。近年來 k-means 演算法也被證實其分群問題是一個 NP-Hard 問題（Mahajan et al. 2012），許多的分群問題也在套用啟發式演算法後，獲得了良好的效果。

四、分割式分群和最佳化問題

分割式分群最著名的就是 k-means 演算法，該演算法雖然擁有優異的運算速度，但是卻容易落入區域最佳解，此外，過分依賴隨機的初始群心解，同樣使得執行結果相當不穩定。為此，研究人員開始找尋其它能夠兼顧速度並擁有穩定結果的分群演算法。

分割式分群本身可以視為一個最佳化問題，目的在最小化群內資料點到該群群心的距離，計算上使用 Sum of Squared Error（SSE），其目標式如下：

$$\mathrm{J}(w, z) = \sum_{i=1}^{N} \sum_{j=1}^{k} w_{ij} \left\| x_i - z_j \right\|^2 \quad (9)$$

其中 k 代表群數，N 代表資料的筆數，x_i為資料點，z_j為群心，w_{ij}則用來判定該資料點是否屬於該群。z_j和w_{ij}的公式如下：

$$z_j = \frac{1}{N_j}\sum_{i=1}^{N} w_{ij}x_i \tag{10}$$

$$w_{ij} = \begin{cases} 1, & \text{if } x_i \in z_j \\ 0, & \text{if } x_i \notin z_j \end{cases} \tag{11}$$

$$N_j = \sum_{i=1}^{N} w_{ij} \tag{12}$$

啟發式演算法在最佳化問題早已被證實具有不錯的搜尋能力，於是，學者們嘗試使用啟發式演算法套用在分群問題上，研究證實，啟發式分群具有良好的分群效能（Karaboga & Ozturk 2011; Senthilnath et al. 2011）。

五、自動化啟發式分群

既有的分群方法大多需要給定一個群數（例如：k-means）或是指定一個門檻值（例如：階層式分群），才能輸出分群的結果。低維度的資料或許可以透過視覺化的觀察決定群數，可是一旦維度超過三，資料就具有難以觀察性，在缺乏其他先備知識的情形下，群數和相關資訊的取得亦變得相當困難。

分群有為了理解而分群和為了實用而分群（Tan et al. 2006）。對於前者來說，分群的目標是透過資料的原生結構，找出資料間的關係並進行分群，進而透過資料本身的群集分佈，讓使用者可以觀察並找出有用或尚未被察覺的資訊。從這個出發點看來，事先給定群數引導演算法進行特定群數的分群，而不是僅僅透過資料本身的特性推薦群數，顯得不是很合理。

基於啟發式演算法在分群問題的良好成效，有學者設計了可以自動化搜尋群數的啟發式分群演算法，像是修改基因演算法（genetic algorithm; GA）的 GCUK（Bandyopadhyay & Maulik 2002），和修改粒子群最佳化演算法（particle swarm optimization; PSO）的 MEPSO（Das et al. 2008）。以下將介紹這兩種演算法。

（一）GCUK

GCUK 是由 Bandyopadhyay 與 Maulik（2002）所提出，該演算法藉由改變染色體的編碼格式，使演算法可以透過自身的搜尋過程決定群數。雖然 GCUK 可以自動決定群數，但仍有其限制，就是必須給定搜尋群數的上下邊界[K_{min} ,K_{max}]，一般來說下邊界為 2，上邊界則由使用者決定，嚴格來說仍不可視為自動取得適合群數。

假使有一分群問題，其空間維度為 2，且使用者給定 K_{min}=2 和 K_{max}=6，

GCUK 在初始化染色體的時候，會從[K_{min} ,K_{max}]間隨機產生一個數值，象徵該染色體所擁有的群數，之後再根據空間維度隨機產生群心的值（位置）。假設現在有一染色體 C_i 如下所示：

#	(20.4,13.2)	#	(15.8,2.9)	(10.0,5.0)	(22.7,17.7)

圖 1：GCUK 染色體表示法

其中#的部分象徵未啟動的群，以範例來說，此染色體代表一組具有 4 個群的分群結果，其群心位置分別是（20.4,13.2）、（15.8,2.9）、（10.0,5.0）、（22.7,17.7）。

GCUK 在交配的時候使用單點交配，其交配示意圖如圖 2。

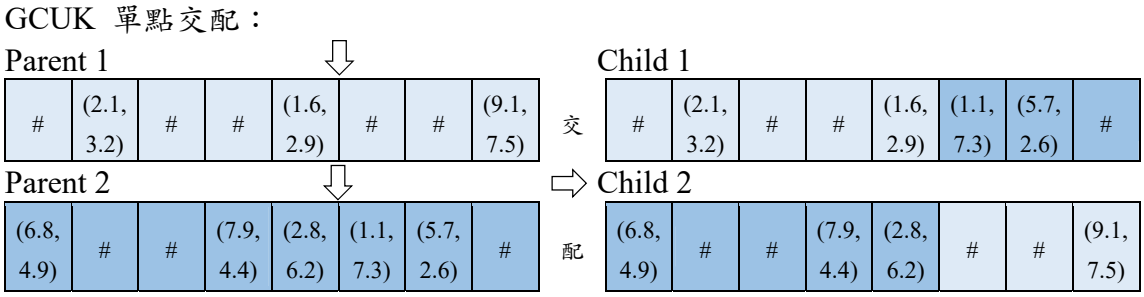

圖 2：GCUK 交配示意圖

圖 2 中親代染色體原先分別為具有 3 個群和 5 個群的分群結果，經過交配後所產生的子代卻變成兩個具有 4 個群的分群結果，GCUK 便是利用此種編碼方式和基因演算法交配時所產生的效應來達成自動化分群的結果。

在適應函數的計算上，GCUK 使用了 Davies-Bouldin index，其公式如下。

$$\text{Fitness}(i) = \frac{1}{\text{DB Index}(\text{Decode}(\text{Chrosome}_i))} \tag{13}$$

GCUK 設定為追求 fitness 的最大化，在最大化 fitness 的同時亦象徵最小化 Davies-Bouldin index，藉此找出較佳的分群結果。

（二）MEPSO

MEPSO 是由 Das 等（2008）學者所提出的自動化啟發式分群演算法，它改善了 PSO 過早收斂的問題，並透過修改粒子表示法，讓 PSO 可以自動化決定群數分群。藉由給定一個群數區間[K_{min} ,K_{max}]，MEPSO 便可以自動找出最佳的群數，此

演算法雖然可以找出群數，但其編碼設計不佳，導致搜尋空間過於複雜亦缺乏合理性，有著效率不彰的問題。

假設有一問題[K_{min} ,K_{max}]=[2,6]，其空間維度為 2，今有一粒子 P_i 如表 1：

表 1：MEPSO 粒子表示法

$T_{i,1}$	$T_{i,2}$	$T_{i,3}$	$T_{i,4}$	$T_{i,5}$	$T_{i,6}$	$M_{i,1}$		$M_{i,2}$		$M_{i,3}$		$M_{i,4}$		$M_{i,5}$		$M_{i,6}$	
0.3	0.6	0.1	0.8	0.9	0.7	1.5	6.8	20.4	13.2	2.6	7.8	15.8	2.9	10.0	5.0	22.7	17.7

前 K_{max} 個值為門檻值 $T_{i,x}$，其範圍在[0,1]之間，只要該門檻值大於 0.5，表示啟動對應的第 x 個群心 $M_{i,x}$，粒子從第 K_{max}+1 個值開始為群心的值，假設此問題的空間維度為 2，為了呈現一個群心，必須使用的格數（值）即為 2 格，而粒子的整體長度為$k + k \times d$。

以粒子 P_i 來說，$T_{i,1}$ 的值為 0.3 小於 0.5，所以並沒有啟動 $M_{i,1}$ 這個群，$T_{i,2}$ 的值為 0.6 大於 0.5，所以啟動了對應的群 $M_{i,2}$={9,12,5,6}，依此類推，可得知粒子 P_i 為一個擁有 4 個群的分群解，其群心位置分別是(20.4,13.2)、(15.8,2.9)、(10.0,5.0)、(22.7,17.7)。

適應函數方面 MEPSO 使用 CS Measure，其公式如下，*eps* 為一極小常數。

$$\text{fitness}(i) = \frac{1}{\text{CS}(\text{Decode}(\text{Particle}_i)) + eps} \tag{14}$$

MEPSO 將 CS Measure 納入適應函數，在最大化 fitness 的同時最小化 CS Measure，進而找出較佳的分群結果。而在編碼方面，把啟動門檻值和群心數值整合進粒子表示法，再透過門檻值判定是否啟動該群，藉由以上設定，MEPSO 便可以自動推薦群數和找出適合的群心位置，完成自動化分群的目的。

參、研究方法

本研究提出一個自動決定群數的自動化分群系統（automatic clustering system; ACS），此系統主要使用改良的人造蜂群演算法進行分群，為了減少演算法搜尋的解空間並增加演算法的效率，本研究會在分群之前，使用一個群數搜尋演算法，找出較為可能的群數區間，進而減少不必要的編碼浪費，並增進後續分群的效率。圖 3 為本研究之 ACS 系統架構圖：以下將針對本研究提出的系統架構和流程以及演算法的細節進行介紹。

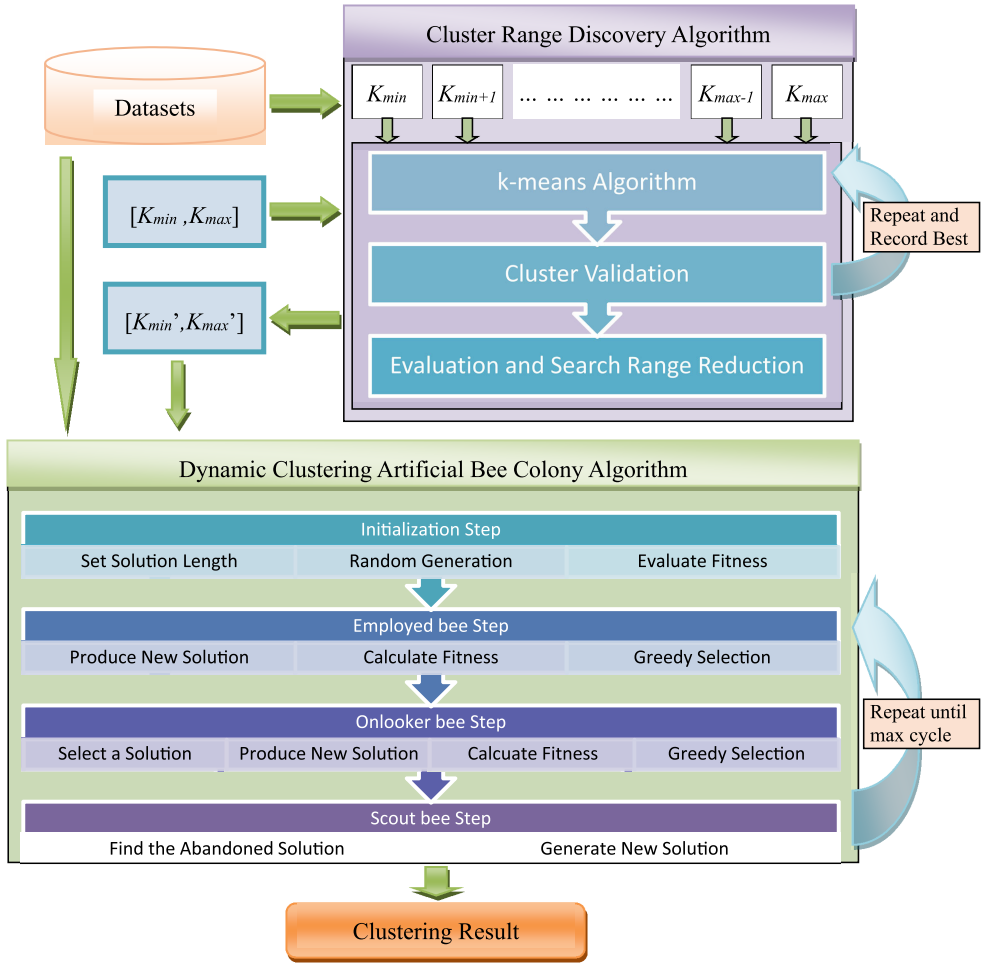

圖 3：ACS 系統架構圖

一、系統流程

針對一個未知群數的資料集，使用者需給定一個群數搜尋上限，之後才能使用本系統，進行自動化搜尋群數和最佳化分群的流程。

本研究先使用群數搜尋演算法（cluster range discovery algorithm; CRD），針對使用者輸入的群數範圍[K_{min} ,K_{max}]進行搜尋，CRD 會找出對於該資料集較為可能的群數，進而達到縮減搜尋空間的目的。CRD 主要會針對使用者輸入的群數範圍，使用 k-means 進行初步的分群，再評估分群結果以縮減群數範圍，之後 CRD 會將此縮減後的群數區間[K_{min}' ,K_{max}']交給動態分群人造蜂群演算法（dynamic clustering artificial bee colony algorithm; DCABC）進行分群，DCABC 主要有三個階段：工蜂階段、觀察蜂階段、偵查蜂階段，三個階段有其不同的搜尋特性和能力，系統會重複進行這三個階段，直到給定的執行代數（iteration），過程中套用

了本研究提出的模範策略，增強了演算法效能，DCABC 由於其特殊的解編碼格式（solution encoding），具有自動化決定群數和優化分群品質的能力，最後 ACS 會輸出分群結果。

二、群數搜尋演算法

群數搜尋演算法（CRD）主要的目的在找出較為可能的群數區間，而不是進行精確的分群。為了達成衡量群數的目的，可行的方法主要有兩種，一種是運用統計的方法，例如：Gap statistic，評估較為可能的群數（Handl & Knowles 2007），或是直接用分群演算法運作於該資料集，藉由給定不同的群數值，獲得各個群數的分群結果，再使用分群評估指標找出較為優異的群數（Arbelaitz et al. 2013）。考量到後續分群演算法的進行和分群評估指標的一致性，本研究決定使用後者。

此方法需要一個低時間成本的分群演算法，在各種分群演算法中，k-means 演算法是一個不錯的選擇。使用 k-means 演算法找尋可能的群數雖然可行，但是 k-means 容易受到不良初始解的影響而產生較差的分群結果（Likas et al. 2003），為此，本研究決定增加 k-means 的執行次數，設給定一執行次數 N，CRD 會在 k-means 階段和 cluster validation 階段重複執行 N 次，並紀錄最佳解和最佳群數，藉由 k-means 每次執行皆會隨機產生初始解的特性，降低不良初始解可能造成的負面影響。

假設最後找出的最佳群數為 k_b，本研究會根據此值擴增搜尋範圍，藉由給定擴增搜尋值 ek，CRD 會產生一新的群數區間[K_{min}',K_{max}']，最後 CRD 會把新的群數區間[K_{min}' ,K_{max}']交給 DCABC 進行後續的分群。

三、動態分群人造蜂群演算法

本研究所提出之動態分群人造蜂群演算法（DCABC），主要修改了人造蜂群演算法（ABC）的兩個部分，第一個部分加入了模範策略（model strategy），此策略讓最佳解不會在偵查蜂的階段被淘汰，並在被拋棄解出現的時候，修改偵查蜂產生新解的方式，新的解將會向最佳解進行學習，藉由以上改變來增進演算法的搜尋效率。第二個部分是修改解編碼的格式，不同於以往給予固定群數的啟發式分群演算法和此類演算法所使用的編碼格式，本研究啟發自 Das 等（2008）學者於自動化啟發式分群研究所提出的編碼概念，在編碼中加入門檻值的概念，以門檻值來決定啟動的群數，藉此達到自動化分群的目的。以下將會先描述人造蜂群演算法，再介紹本研究提出的模範策略和新的解編碼格式，最後會敘述完整的動態分群人造蜂群演算法流程。

（一）人造蜂群演算法

人造蜂群演算法（ABC）是一種群智慧演算法，其概念啟發自蜂群的階級生態和蜜蜂尋找花蜜的行為。ABC 把整個解空間視為覓食空間，每個解的優劣程度象徵著花蜜的多寡，而蜂群會互相交換訊息，以找尋更好的食物來源點。

ABC 將蜜蜂分成三種角色：工蜂、觀察蜂、偵查蜂，工蜂負責隨機產生初始解，紀錄曾經到訪過的最佳位置，並根據其它工蜂的位置進行貪婪搜尋；觀察蜂會獲取所有工蜂的食物來源資訊，再根據機率前往其中一個食物來源，食物來源的品質決定其前往的機率，之後會對該食物來源進行協助搜尋，如果有找到更好的解，便會將此資訊分享給工蜂（即該工蜂移動到更佳解的位置）；偵察蜂負責拋棄不佳的食物來源，當某個工蜂所在之位置（即食物來源）經過多次搜尋卻無法改善解品質的時候，該食物來源點便會被拋棄，偵查蜂會在此時負責引導工蜂前往全新的位置（即在解空間隨機產生新的食物來源）。以上是 ABC 各個角色的詳細概念，下面是 ABC 的運算公式和虛擬碼。在演算法中：

```
1. Give the population number SN, max cycle number MCN, limit number LN
2. Randomly initialize food sources for Employed Bee(EB) FSi, i=1,2,…,SN
3. Evaluate the fitness value for each EB
4. Set cycle = 0
5. While(cycle<MCN){
6.      For each EB{
7.         Produce new solution new_FSi by (17)
8.         Calculate the fitness value of new_FSi
9.         Apply greedy selection process and count the limit Li
10.    }
11.    Calculate Pi according to the fitness value by (18)
12.    For each Onlooker Bee{
13.     Select a food source FSi depending on Pi
14.        Produce new solution new_FSi by (17)
15.        Calculate the fitness value of new_FSi
16.        Apply greedy selection process and count the limit Li
17.     }
18.    For each EB{
19.        If(Li>LN){
20.        Scout Bee move the EB from its food source to another by (19)
21.        }
22.    }
23.    Compare and record the best solution(Food Source)
24.     cycle++
25. }
26. Output the best solution
```

圖 4：Pseudo-code of the ABC algorithm

第 6 行到第 10 行表示產生新的食物來源：

$$new\_FS_i^j = \begin{cases} FS_i^j + \text{Rand}[-1,1] \times \left(FS_i^j - FS_{k=\text{RandInt}[1,SN],k\neq i}^j\right), if\ j = \text{RandInt}[1,D] \\ FS_i^j, otherwise \end{cases} \quad (15)$$

第 11 行到第 17 行表示觀察蜂前往各個食物來源之機率：

$$P_i = \frac{\text{fitness}(FS_i)}{\sum_{j=1}^{SN} \text{fitness}(FS_j)} \quad (16)$$

第 18 行到第 21 行偵查蜂移動工蜂到新的食物來源：

$$FS_i^j = FS_{min}^j + \text{Rand}[0,1] \times (FS_{max}^j - FS_{min}^j),\ j = 1,2,\dots,D \quad (17)$$

若未達穩定最佳解則重新再試，直到最佳情況。

（二）模範策略

人造蜂群演算法的設計上，每個食物來源（即工蜂位置）會在工蜂階段和觀察蜂階段進行搜尋，此搜尋採用貪婪法則，若有找出更佳的解則取代原來的解。在工蜂階段每個食物來源都會擁有一次的搜尋機會，而在觀察蜂階段，則會根據各個食物來源的品質優劣（即適應函數的優劣），決定該食物來源的搜尋次數，品質愈好的解，搜尋次數也會愈多。此種設計是為了加強演算法探鑽的能力，假若該解鄰近全域最佳解，將加速演算法完成最佳化的目標。而偵查蜂的目的則是幫助演算法淘汰不具搜尋價值的解（和其鄰近解空間），故人造蜂群演算法會給定一個限制次數（limit number），假使某個解經過多次連續搜尋卻沒有改善，在偵查蜂的階段，該解便會被淘汰，並隨機產生新的解。

上述設計看似合理，卻可能存在以下問題：一個解的品質愈好，雖然搜尋次數增加，其被淘汰的機率也會增加，假使被淘汰解仍具有搜尋價值，且該解鄰近全域最佳解，則演算法可能錯失了良好的機會。

為了確保被淘汰解真的不再具有任何搜尋價值，演算法通常會設定一個較大的限制次數，可是如此一來，反而讓較差的解不易被淘汰，造成許多搜尋上的浪費。因此，本研究提出了模範策略（model strategy），此策略會保護最佳解，使其不會在偵查蜂的階段因為連續搜尋失敗次數過多而被淘汰。另外，在被淘汰解產生的時候，模範策略會讓被淘汰解向最佳解進行學習，藉此強化最佳解的鄰近空間搜尋程度。

（三）解編碼格式和適應函數

本研究參考 Das 等（2008）學者所提出之 MEPSO 於自動化分群之編碼方式，

並縮減其啟動門檻之編碼空間為一格，詳細如下：

設給定群數搜尋區間[K_{min}' ,K_{max}']，資料空間維度為 D，則解編碼之長度 $L=(D\times K_{max}')+1$，其中第一格象徵群數的啟動門檻值 AT（activated threshold），且 $AT \in [0,1]$，後面的$(D\times K_{max}')$格則依序用來表示各群群心的值。此種編碼方式以 AT 的值判斷啟動了幾個群，其公式和判斷條件如下：

$$activated_i = AT_i \times (K'_{max} - K'_{min} + 1) + K'_{min} \tag{18}$$

$$\text{If the } j \text{ of } centroid_{ij} < activated_i \text{ , then activate the } j\_th \text{ centroid} \tag{19}$$

設有一問題，給定之群數區間[K_{min}' ,K_{max}']=[2,6]，解空間維度為 2，其中一個解編碼 SE_i 數值如表 2：

表 2：DCABC 粒子表示法

AT_i	$Centroid_{i1}$		$Centroid_{i2}$		$Centroid_{i3}$		$Centroid_{i4}$		$Centroid_{i5}$		$Centroid_{i6}$	
0.5	20.4	13.2	15.8	2.9	10.0	5.0	22.7	17.7	1.5	6.8	2.6	7.8

可算出其$activated_i$=0.5*(6-2+1)+2=4.5，根據判斷條件，可得知在 AT_i=0.5 的時候，相當於啟動了前 4 個群 $Centroid_{i1}$ 到 $Centroid_{i4}$。

在適應函數方面，本研究參考 Arbelaitz 等（2013）學者的研究，決定使用 Silhoutte Index 這個分群評估指標作為適應函數，其適應函數公式如下：

$$\text{fitness}(i) = \text{Silhoutte Index}\big(\text{Decode}(SE_i)\big) + 1 \tag{20}$$

由於 Silhoutte Index 的值域為[-1,1]，本研究將求出的值加一，以利演算法整體運算。Silhoutte Index 的值愈大象徵分群品質愈好，故在 DCABC 的運算過程中，設定為追求最大化適應函數，以找出較佳的分群結果。

（四）DCABC 演算法流程

DCABC 演算法流程大致上和 ABC 無異，主要的差異在偵查蜂階段加入了本研究所提出的模範策略。值得一提的是，在 DCABC 的分群過程中，有時會出現解編碼象徵 k 群，但解碼後的群心經過分配點之後，實際擁有的群數小於 k 的情形（即有若干群為空群），當此種情況發生時，考量到演算法搜尋和運作的合理性，DCABC 將會重新隨機產生一組合理的解編碼，並將限制次數歸零，才會繼續進行運算。

肆、實驗結果

本研究取用了 GCUK 和 MEPSO 這兩個比較對象所使用過的 5 個資料集：Iris、Wine、Glass、Segmentation、Breast Cancer Wisconsin (Original)，另外參考 Arbelaitz 等（2013）學者的分群研究，加入了 Ecoli、Ionosphere、Libras Movement 等 3 個資料集，共 8 個資料集，這些資料集皆從 UCI machine learning repository 取得。由於自動化分群所找出來的群數可能會不同於資料集本身給定的類別數，因此需要一個可以比較不同群數分割之間相似度的指標，本研究參考 Arbelaitz et al.（2013）學者的研究決定使用 Adjusted Rand Index（ARI）（Hubert & Arabie 1985）作為衡量指標。

在參數的設定上，為了公平的比較各個演算法間的效能，本研究採用 Das 等（2008）學者在 MEPSO 的研究中使用的參數組，以 50000 次 fitness runtimes 作為各個演算法的比較基準，人口數一律設定為 40，執行代數在 GCUK 和 MEPSO 設定為 1250，在 DCABC 由於一個代數會使用到兩次的 fitness 運算，所以執行代數需除以 2，設定為 625。群數搜尋範圍[K_{min} ,K_{max}]一律參考 MEPSO 的設定設為[2,20]，GCUK 和 MEPSO 其餘的參數則依照該演算法原始研究所給予的值，在 DCABC 的限制次數部份，參考 Yan et al.（2012）學者使用 ABC 進行分群的研究，將其設定為 100 次。

在 ACS 的部分，由於 CRD 演算法運作時會進行分群評估指標的計算，此計算一次之計算量相當於一次適應函數的計算，故在 ACS 運作的時候，當 ACS 執行到 DCABC 的階段時，DCABC 的 fitness runtimes 會減去 CRD 所使用到的分群評估指標次數，以配合整體實驗的測試標準。

一、實驗一：比較 ABC 和 DCABC 在演算法效能上的差異

本實驗主要是為了比較使用模範策略（model strategy）是否能夠改善演算法的效能，兩者之中 ABC 沒有使用模範策略，DCABC 則有使用模範策略。表 3 列出了此次實驗所用到之參數值。

此實驗針對兩個演算法針對每個資料集獨立執行 30 次的最佳適應函數平均值（愈大愈好），本研究對結果進行獨立樣本 t 檢定（95%信賴區間），若該資料集該演算法的結果顯著優於比較對象，則在該數值的前方給予＊字號，表 3 先說明兩個實驗所用的參數。

表 3：實驗一參數設定表

	DCABC	ABC
人口數（Populations）	40	40
執行代數（Iterations）	625	625
限制次數（Limit number）	100	100
群數搜尋範圍（Range of k）	[2,20]	[2,20]
使用模範策略（Model Strategy）	有	無

表 4 中在 8 個資料集中，雖然有 3 個資料集在統計上兩者並無顯著差異，但其他的 5 個資料集 DCABC 的執行結果皆是優於 ABC 且顯著，由此可以看出本研究加入的模範策略可以有效的改善演算法搜尋的效能。

表 4：實驗一結果

Dataset\Algorithm	DCABC	ABC
Iris	1.701421686578	1.701421686578
Wine	*1.673338649557	1.673337889901
Glass	1.652213180025	1.652072946828
Segmentation	1.882588782790	1.882588794406
Breast Cancer Wisconsin (Original)	*1.600763595529	1.599872701088
Ecoli	*1.432990314127	1.431345354134
Ionosphere	*1.340414364063	1.330510826595
Libras Movement	*1.217983723265	1.207645326848

二、實驗二：比較 DCABC、MEPSO 和 GCUK 在分群效能上的差異

表 5 為本研究提出的 DCABC 和比較對象（GCUK 和 MEPSO）針對每個資料集獨立執行 30 次的 ARI 平均值和群數平均值，由於找出的群數不能代表兩種結果的相似度（即便群數相同，群內資料的組成亦有可能不同），故在表格的判讀上和演算法的比較上，仍以 ARI 的數值為主（後續實驗亦同），ARI 愈大表示結果愈好。本研究對 DCABC 和比較對象（GCUK 和 MEPSO）求出之 ARI 數值進行獨立樣本 t 檢定（95%信賴區間），若該演算法在該資料集的結果顯著優於對手，則在該數值的前方給予＊字號。

從表 5（左）可看出，在 8 個資料集中 DCABC 有 5 個資料集顯著優於 GCUK，在 Glass 和 Libras Movement 這兩個資料集兩者並無顯著差異，僅有在

Segmentation 資料集 GCUK 顯著優於 DCABC，但兩者的 ARI 數值皆趨近於 0，所以實際上並不具有討論的價值（因為趨近於 0 象徵兩種分割近似於隨機分布）。

而由表 5（右）可看出，在 8 個資料集中，DCABC 有 3 個資料集顯著優於 MEPSO，在 Iris、Glass、Ionosphere、Libras Movement 這 4 個資料集兩者並無顯著差異，僅在 Segmentation 資料集 MEPSO 顯著優於 DCABC。

表 5：GCUK 和 DCABC 以及 MEPSO 和 DCABC 之分群結果比較表

Dataset\Algorithm		GCUK		DCABC		MEPSO	
Name	k	k	ARI	k	ARI	k	ARI
Iris	3	2.6	0.546	2	*0.567	4.03	0.564
Wine	3	5.13	0.282	2	*0.348	2	0.312
Glass	7	3.6	0.06	2	0.039	3.83	0.064
Segmentation	7	2.63	*0.00002	2	0.000002	2.86	*0.056
BCW (Original)	2	6.86	0.617	2	*0.748	8.03	0.62
Ecoli	8	5.06	0.353	4.43	*0.532	5.63	0.298
Ionosphere	2	6.63	0.09	6.1	*0.144	8.13	0.131
Libras Movement	15	11.73	0.056	2.23	0.062	8.93	0.061

雖然在 Segmentation 資料集 DCABC 呈現了不佳的結果，但整體來說，DCABC 仍較表現傑出，從上述結果可以明顯看出 DCABC 優於 GCUK，而在 MEPSO 的部分，雖然優勢並沒有那麼明顯，但若考量到演算法的執行時間，DCABC 顯然更加具有效率。

圖 5 為各個演算法之於各個資料集經過 30 次獨立運算的執行時間平均值，以執行時間來說，MEPSO 表現都是最差的，而 DCABC 除了在 Segmentation 和 Breast Cancer Wisconsin (Original)這兩個資料集輸給 GCUK，其他的 6 個資料集都是最快的執行時間。從以上數據可以看出，DCABC 不僅擁有優良的分群品質，亦相當具有效率。

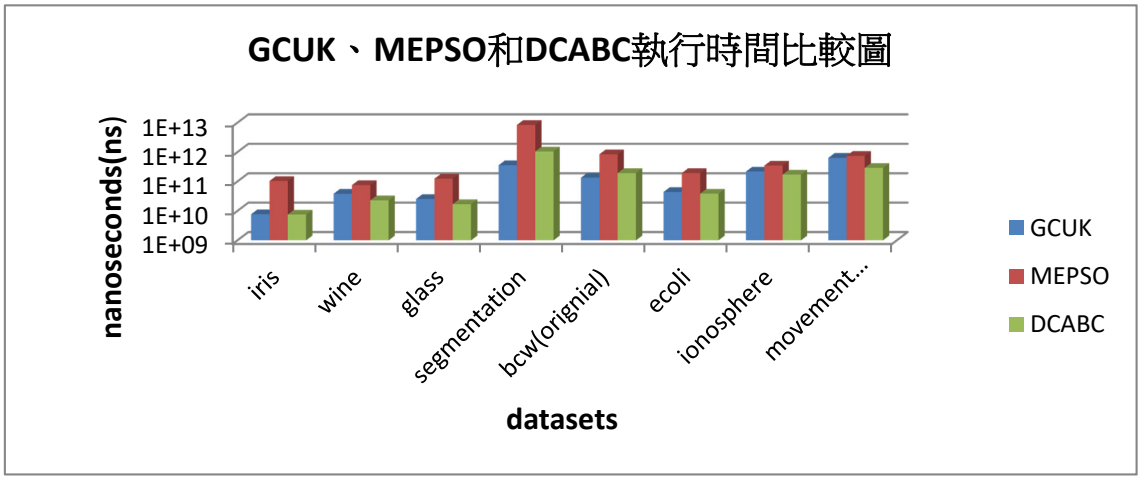

圖 5：GCUK、MEPSO 和 DCABC 執行時間比較圖

三、實驗三：比較 DCABC 與本研究提出改善 DCABC 的 ACS 比較

表 6 為 DCABC 和 ACS 針對每個資料集獨立執行 30 次的 ARI 平均值和群數平均值，本研究對 DCABC 和 ACS 求出之 ARI 數值進行獨立樣本 t 檢定（95%信賴區間），若該資料集該演算法（系統）的結果顯著優於比較對象，則在該數值的前方給予＊字號。

從表 6 可以看出，使用 ACS 讓 Breast Cancer Wisconsin (Original)和 Libras Movement 有了顯著的改善，Segmentation 資料集的部分雖然 ACS 表現比起 DCABC 來得糟，但其實兩者的數值都相當趨近於 0，並不具有討論的意義。

圖 6 為 DCABC 和 ACS 的執行時間比較圖，從這張圖可以看出，除了 Ionosphere 和 Libras Movement 資料集，在其它的資料集，透過事先縮減群數搜尋區間，減少編碼長度，可以有效的降低啟發式分群的執行時間。

表 6：DCABC 和 ACS 之分群結果比較表

Dataset\Algorithm		DCABC		ACS	
Name	k	k	ARI	k	ARI
Iris	3	2	0.567	2	0.567
Wine	3	2	0.348	2	0.354
Glass	7	2	0.039	2	0.03
Segmentation	7	2	*0.000002	2	0.000001
BCW (Original)	2	2	0.748	2	*0.795
Ecoli	8	4.43	0.532	4.96	0.582
Ionosphere	2	6.1	0.144	8.5	0.176
Libras Movement	15	2.23	0.062	14.1	*0.208

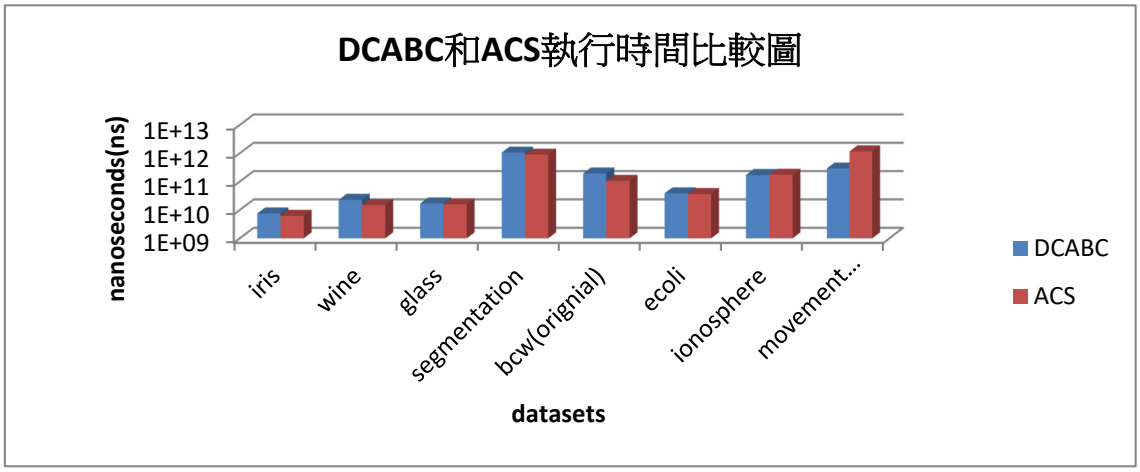

圖 6：DCABC 和 ACS 執行時間比較圖

進而觀察 Ionosphere 和 Libras Movement 這兩個資料集，可以發現 ACS 在這兩個資料集找出的群數比起 DCABC 來得多（特別是 Libras Movement 資料集），故本研究推測，當 CRD 推薦的群數較大的時候，除了縮減的編碼長度減少，後續的分群演算法（即 DCABC）在分群的過程中，為了產生和維持每個解編碼皆是合理且有效的高群數解，將會花費更多的時間，進而導致執行時間的增加。

雖然 Ionosphere 和 Libras Movement 資料集的執行時間增加了，但 ACS 在這兩個資料集的分群品質，確實比起 DCABC 來得優良，且 ACS 在其它資料集的執行時間都獲得降低。總結來說，本研究提出的 ACS 是一個具有效率和效能的分群系統。

伍、討論與結論

傳統的分群方法大多對於使用者具有額外的資訊要求，像是 k-means 演算法必須給定 k 值，階層式分群必須給予門檻值，才能夠完成分群。此種要求增加了使用者的負擔，且在使用者缺乏相關資訊或知識的時候，顯得更加棘手。本研究提出了一個自動化分群系統 ACS 和自動化啟發式分群演算法 DCABC，讓使用者不需對於資料集具有先備知識，便可以直接讓資料集根據其本身的特性（聚合性和離散性）進行自動化分群。

本研究不同於 GCUK 和 MEPSO，提出了另一種的解編碼格式，和它們的主要差異在於，本研究的解編碼格式，所形成之解空間，不會有大量的重複分群解，藉由單純化解空間，可以有效率的提升演算法的搜尋品質。由此可見，解編碼的設計，也是啟發式演算法相當重要的一環。

考量到編碼長度會影響啟發式演算法的搜尋效率，本研究提出了自動化分群

系統（ACS），此系統先使用群數搜尋演算法（CRD）縮減群數搜尋區間，再由DCABC 進行後續的自動化分群。實驗證實，此系統雖然會在 CRD 推薦的群數較大的時候增加執行時間，但在分群結果上也有所提升，且在 CRD 推薦的群數較低的時候，系統的執行時間確實獲得下降。整體來說，ACS 能夠有效的增進分群的效率和效能。

本研究尚有諸多進步的空間，可以在後續的研究上進行改良，以下將針對本研究可能的後續發展方向提出建議。像是運用不同的啟發式演算法或是設計新的啟發式演算法來解決分群問題，再來就是應用本研究提出的模範策略 ABC 之於其他最佳化問題，另外像是 ACS 的系統流程，或許可以加強兩個演算法間的合作性。

最後在研究限制方面，本研究所提出之系統和演算法雖然可以自動化決定適合的群數，但在初始化的時候，仍需由使用者給定群數區間，若資料集之最適群數並未在此區間內，則本系統和本演算法可能無法提供最佳品質的分群結果，因此此方法建議使用者，若在不確定的情況下，可以給予較大的可能範圍。另本研究使用群心作為整個群的原型（prototype），而在相似度的計算上則是使用歐幾里得距離（Euclidean distance），故本研究提出之系統和演算法僅能找出超球面邊界（hypersphere boundary）的群分割，若資料集之最適群分割不符合此種邊界，則無法找出適合該資料集的分群結果。所以僅在符合以上兩個限制下，本研究的成果才能有所表現。

誌謝

本文接受科技部專題研究計畫 MOST107-2410-H-006-040-MY3 補助，謹此致謝。

參考文獻

Akay, B. and Karaboga, D. (2012), ‘A modified Artificial Bee Colony algorithm for real-parameter optimization’, *Information Sciences*, Vol. 192, No. 1, pp. 120-142.

Arbelaitz, O., Gurrutxaga, I., Muguerza, J., Pérez, J. and Perona, I. (2013). ‘An extensive comparative study of cluster validity indices’, *Pattern Recognition*, Vol. 46, N o.1, pp. 243-256.

Bandyopadhyay, S. and Maulik, U. (2002), ‘Genetic clustering for automatic evolution of clusters and application to image classification’, *Pattern Recognition*, Vol. 35, NO. 6, pp. 1197-1208.

Chatterjee, S., Carrera, C. and Lynch, L.A. (1996), ‘Genetic algorithms and traveling

salesman problems', *European Journal of Operational Research*, Vol. 93, No. 3, pp. 490-510.

Chou, C.-H., Su, M.-C. and Lai, E. (2004), 'A new cluster validity measure and its application to image compression', *Pattern Analysis and Applications*, Vol. 7, No. 2 , pp. 205-220.

Cowgill, M.C., Harvey, R.J. and Watson, L.T. (1999), 'A genetic algorithm approach to cluster analysis', *Computers and Mathematics with Applications*, Vol. 37, No. 7, pp. 99-108.

Das, S., Abraham, A. and Konar, A. (2008), 'Automatic kernel clustering with a Multi-Elitist Particle Swarm Optimization Algorithm', *Pattern Recognition Letters*, Vol. 29, No. 5, pp. 688-699.

Das, S., Abraham, A. and Konar, A. (2009), *Metaheuristic Clustering*: Springer.

Davies, D.L. and Bouldin, D.W. (1979), 'A Cluster Separation Measure', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-1, No. 2, pp. 224-227.

Glover, F. (1986), 'Future paths for integer programming and links to artificial intelligence', *Computers & Operations Research*, Vol. 13, No. 5, pp. 533-549.

Handl, J. and Knowles, J. (2007), 'An Evolutionary Approach to Multiobjective Clustering', *IEEE Transactions on Evolutionary Computation*, Vol. 11, No. 1, pp. 56-76.

Hasan, M.A., Chaoji, V., Salem, S. and Zaki, M.J. (2009), 'Robust partitional clustering by outlier and density insensitive seeding', *Pattern Recognition Letters*, Vol. 30, No. 11, pp. 994-1002.

Holland, J.H. (1975), *Adaptation in Natural and Artificial Systems*: University of Michigan Press.

Hubert, L. and Arabie, P. (1985), 'Comparing partitions', *Journal of Classification*, Vol. 2, No. 1, pp. 193-218.

Ji, J., Pang, W., Zhou, C., Han, X. and Wang, Z. (2012), 'A fuzzy k-prototype clustering algorithm for mixed numeric and categorical data', *Knowledge-Based Systems*, Vol. 30, pp. 129-135.

Karaboga, D. and Akay, B. (2009), 'A comparative study of Artificial Bee Colony algorithm', *Applied Mathematics and Computation*, Vol. 214, No. 1, pp. 108-132.

Karaboga, D. and Ozturk, C. (2011), 'A novel clustering approach: Artificial Bee Colony (ABC) algorithm', *Applied Soft Computing*, Vol. 11, No. 1, pp. 652-657.

Laszlo, M. and Mukherjee, S. (2007), 'A genetic algorithm that exchanges neighboring

centers for k-means clustering', *Pattern Recognition Letters*, Vol. 28, No. 16, pp. 2359-2366.

Liao, S.-H., Chu, P.-H. and Hsiao, P.-Y. (2012), 'Data mining techniques and applications – A decade review from 2000 to 2011', *Expert Systems with Applications*, Vol. 39, No. 12, pp. 11303-11311.

Liang, Y., Wan, Z. and Fang, D. (2017), 'An improved artificial bee colony algorithm for solving constrained optimization problems', *International Journal of Machine Learning and Cybernetics*, Vol. 8, No. 3, pp. 739-754.

Likas, A., Vlassis, N. and Verbeek, J.J. (2003), 'The global k-means clustering algorithm', *Pattern Recognition*, Vol. 36, No. 2, pp. 451-461.

Mahdavi, M., Chehreghani, M.H., Abolhassani, H. and Forsati, R. (2008), 'Novel meta-heuristic algorithms for clustering web documents', *Applied Mathematics and Computation*, Vol. 201, No. 1-2, pp. 441-451.

Mahajan, M., Nimbhorkar, P. and Varadarajan, K. (2012), 'The planar k-means problem is NP-hard', *Theoretical Computer Science*, Vol. 442, No. 13, pp. 13-21.

Oftadeh, R., Mahjoob, M.J. and Shariatpanahi, M. (2010), 'A novel meta-heuristic optimization algorithm inspired by group hunting of animals: Hunting search', *Computers and Mathematics with Applications*, Vol. 60, No. 7, pp. 2087-2098.

Rousseeuw, Peter J. (1987), 'Silhouettes: A graphical aid to the interpretation and validation of cluster analysis', *Journal of Computational and Applied Mathematics*, Vol. 20, pp. 53-65.

Sabau, A.S. (2012), 'Survey of clustering based financial fraud detection research', *Informatica Economica*, Vol. 16, No. 1, pp. 110-122.

Senthilnath, J., Omar, S.N. and Mani, V. (2011), 'Clustering using firefly algorithm: Performance study', *Swarm and Evolutionary Computation*, Vol. 1, No. 3, pp. 164-171.

Song, X., Yan, Q. and Zhao, M. (2017), 'An adaptive artificial bee colony algorithm based on objective function value information', *Applied Soft Computing*, Vol. 55, pp. 384-401.

Szeto, W.Y., Wu, Y. and Ho, S.C. (2011), 'An artificial bee colony algorithm for the capacitated vehicle routing problem', *European Journal of Operational Research*, Vol. 215, No. 1, pp. 126-135.

Tan, P.N., Steinbach, M. and Kumar, V. (2006), *Introduction to Data Mining*: Addison Wesley, Boston, MA, USA.

Wei, J., Lee, M., Chen, H. and Wu, H. (2013), 'Customer relationship management in the

hairdressing industry: an application of data mining techniques', *Expert Systems with Applications*, Vol. 40, No. 18, pp. 7513-7518.

Yan, Y., Zhang Y. and Gao, F. (2012), 'Dynamic artificial bee colony algorithm for multi-parameters optimization of support vector machine-based soft-margin classifier', *EURASIP Journal on Advances in Signal Processing*, Vol. 2012: 160. https://doi.org/10.1186/1687-6180-2012-160.