南　　華　　大　　學

資訊管理學系

碩士論文

# 基於反彈機制 KPSO 分群之有效遺漏值推估方法

# An effective missing value estimation approach based on reflexive KPSO clustering

研 究 生：魏岑甄

指導教授：邱宏彬

中華民國 九十七 年 六 月

# 南　華　大　學

## 資訊管理學系

## 碩 士 學 位 論 文

基於反彈機制 KPSO 分群之有效遺漏值推估方法

研究生：(　　)

經考試合格特此證明

口試委員：

李翔詣

指導教授：

系主任(所長)：

口試日期：中華民國　97年　6月　28日

# 誌　　謝

本文承蒙邱宏彬教授之悉心指導，李翔詣教授、謝昆霖教授之指正引導，始能順利完成，謹此致上最誠摯的謝意。撰寫論文中經常麻煩逸瑋學長、俊男學長多次與我討論和提供寶貴的意見，也幸苦了尹汝姐、全男和朋友的幫忙及爸媽、妹妹和室友小俞的打氣加油，在此一併致謝。

感謝我的同學及好友，在論文研究期間給予我精神上莫大之鼓勵；最後，謹以此文獻給我摯愛的所有人，要和你們說一聲幸苦了，謝謝，我愛你們。

# 基於反彈機制 KPSO 分群之有效遺漏值推估方法

學生：魏岑甄　　指導教授：邱宏彬

南　華　大　學　資訊管理學系碩士班

## 摘　　要

目前企業各方面決策幾乎是以歷史資料探勘(Data mining)結果分析爲基礎，故資料庫的完整性則十分重要，若是資料庫中出現過多的遺漏值(Missing value)，則容易使探勘結果可靠性降低，因此遺漏值推估問題成爲許多研究者努力的目標。在早期針對遺漏值推估問題已有許多方法如直接刪除、以平均或眾數回填等，但這樣的方法若遇到含有大量的遺漏值資料時，通常無法提供最後決策有利參考資訊。

本研究在搜尋大量文獻及經過多方討論後，決定改良粒子群最佳化演算法( Particle swarm optimization,PSO)提出整合反彈機制、K-means 及 PSO 之反彈機制 KPSO( hybrid K-means and efficient Particle swarm optimization in clustering ,RKPSO)，對資料作分群後以群內平均回填該

群遺漏值欄位，並與文獻結果比較及多種資料庫作實驗證實本研究方法之可行性。

關鍵詞:資料探勘(Data mining)、遺漏值(Missing value) 、粒子群最佳化演算法( Particle swarm optimization PSO) 、反彈機制 KPSO( hybrid K-means and efficient Particle swarm optimization in clustering ,RKPSO)

# An effective missing value estimation approach based on reflexive KPSO clustering

Student：Tsen-Jen Wei Advisors：Dr. Hung-Pin Chiu .

Department of Information Management
The M.I.M. Program
Nan-Hua University

## ABSTRACT

Nowadays, enterprises' decisions almost are according to the analytic outcomes of past data mining. Hence, it's quite important to keep the solidity of database. In early days , people always used the average、mode to backfill missing value or directly delete the data that includes missing values. However, it's not a very good way to solve this problem. If there are too many missing values within the database it can not offer reliable information.

After searching lots of conferences and discussing with my professor, we try to integrate K-means、efficient in PSO to cluster the data and estimate missing values. At last, we found RKPSO (hybrid K-means and efficient Particle swarm optimization in clustering) better than other ways on through actual database experiment and conferences comparison.

*Keywords: data mining, missing value, Particle swarm optimization (PSO), KPSO( hybrid K-means and efficient Particle swarm optimization in clustering ,RKPSO)*

# 目　　錄

# 表目錄

# 圖目錄

# 第一章 緒論

本章將描述本研究的背景環境、研究動機、研究方法和簡述文章的整體流程架構。

## 第一節 研究背景

近年來，資料探勘(Data Mining)在企業蔚爲一股風潮，愈來愈多的企業想藉由導入資料探勘來提昇經濟利益。所謂的資料探勘簡言之便是能從龐大的數據庫中找出有價值的隱藏事件，再利用人工智能(Artificial Intelligence ,AI)和統計學等探勘技術將資料深入分析，挖掘其中有利的資訊和知識，依據企業問題塑出不同解決模型(Model)，提供決策者作參考依據。舉例而言，餅乾製造商和零售業者可利用資料探勘技術將龐大的顧客資料做篩選、分析、推導以及預測，找出哪種產品組合及促銷帶來的響應率最大，在什麼季節什麼樣的口味最受歡迎等等。換句話說，資料探勘可讓企業了解不同客戶的需求及喜好以作出最佳行銷策略。

如圖 1.1 在資料探勘程序的資料前置處理部份，對於如何有效控制遺漏值(Missing Value)是很重要的問題。而所謂的遺漏值並非空字串(“”)、空白(space)或 0，指的是未定義或不明的值，因爲遺漏值可能會出現在關鍵性資料上而造成資料探勘結果不精確，若資料庫中存有太多遺漏值也易使資料探勘結果失去意義。

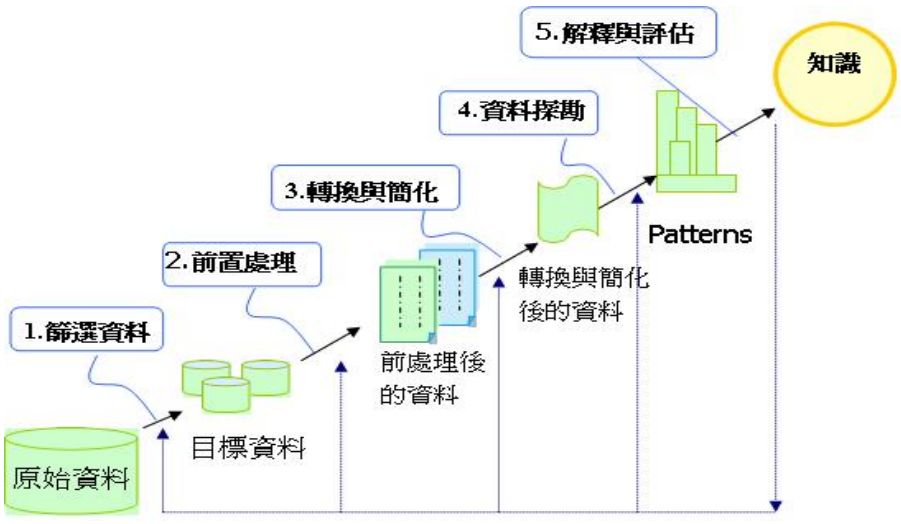


圖 1.1 知識發現程序[9]

## 第二節 研究動機

許多歷史資料的整合、歸納及分析，所用到的資料大多都是從資料庫中所擷取，因此資料庫資料的完整性則十分重要。然而，資料輸入過程有許多因素容易造成資料的遺漏，像是人工輸入時漏填、未存檔時電腦當機等。目前遺漏值在許多研究領域都是一個複雜的問題。以資料探勘來說，遺漏值的存在，可能會造成了下列幾項影響[8]：

1.系統遺失了大量的有用資訊。

2.使系統確定性成份難以把握。

3.過多的遺漏值資料易使探勘過程陷入混亂，導致輸出結果不可靠。

對資料探勘而言，難以通過自身的演算法去完善地處理不完整資料。因此，遺漏的資料需要通過專門的方法進行推估、填補等，以減少資料探勘演算法與實際應用之間的差距，而讓遺漏值的推估更爲精確、方法更爲簡便、執行更爲快速則是許多學者們努力的目標。

## 第三節 研究程序

本研究所探討的是針對遺漏值問題可利用物以類聚的推估特性[1]加以分析，在實驗中利用 PSO (Particle swarm optimization)演算法及改良 PSO 等方法對不同型態資料庫及不同遺漏值存有率下進行實驗，最後依照實驗結果詳加分析和探討。另外本研究以文獻[10]反彈機制 PSO 可避免粒子落入尋解空間外爲基礎，再針對其收斂速度作考量利用 K-means 可加快收斂速度之特性，提出反彈機制加 KPSO( hybrid K-means and Particle swarm optimization in clustering)。期望在反彈機制 PSO 良好的分群效果下能有更快的收斂速度以節省時間成本。研究流程如圖 1.2 所示，下列則爲本研究流程簡要說明:

1. 確立研究問題

經過多方的參考和討論，本研究以遺漏值問題爲主軸，針對其問題特性，尋

求有利的技術及估計方法並加以實驗分析。

2. 確立研究技術

廣閱相關分群文獻後，決定使用在分群上有優異實驗證實的啓發式演算法 PSO-clustering[11]，以物以類聚的特性來推估遺漏值。再以改良 PSO 分群來實驗是否能更縮小遺漏值估計誤差。

3. 收集相關文獻

收集有關遺漏值、PSO、PSO- clustering、K-means 及改良式 PSO 等文獻，同時整理出各種用來解決遺漏值問題的技術，並彙整其優缺。

4. 撰寫程式

將 K-means、PSO- clustering、KPSO- clustering、反彈機制以 Matlab 程式語言撰寫，並且尋找適用的實際資料庫設定人工資料範圍後加以實驗及推估資料。

5. 實驗與數據分析

將 K-means、PSO- clustering、KPSO- clustering、反彈機制加 PSO- clustering、反彈機制加 KPSO- clustering 應用於資料庫分群及回填遺漏值後，分析回填值與真實值的誤差率。

6. 結論與未來研究方向

以推估結果證實，反彈機制加 KPSO- clustering 運用於解遺漏值問題是否能較以往推估方式精準，及在哪些方面還可以進行改良或其它研究方向。

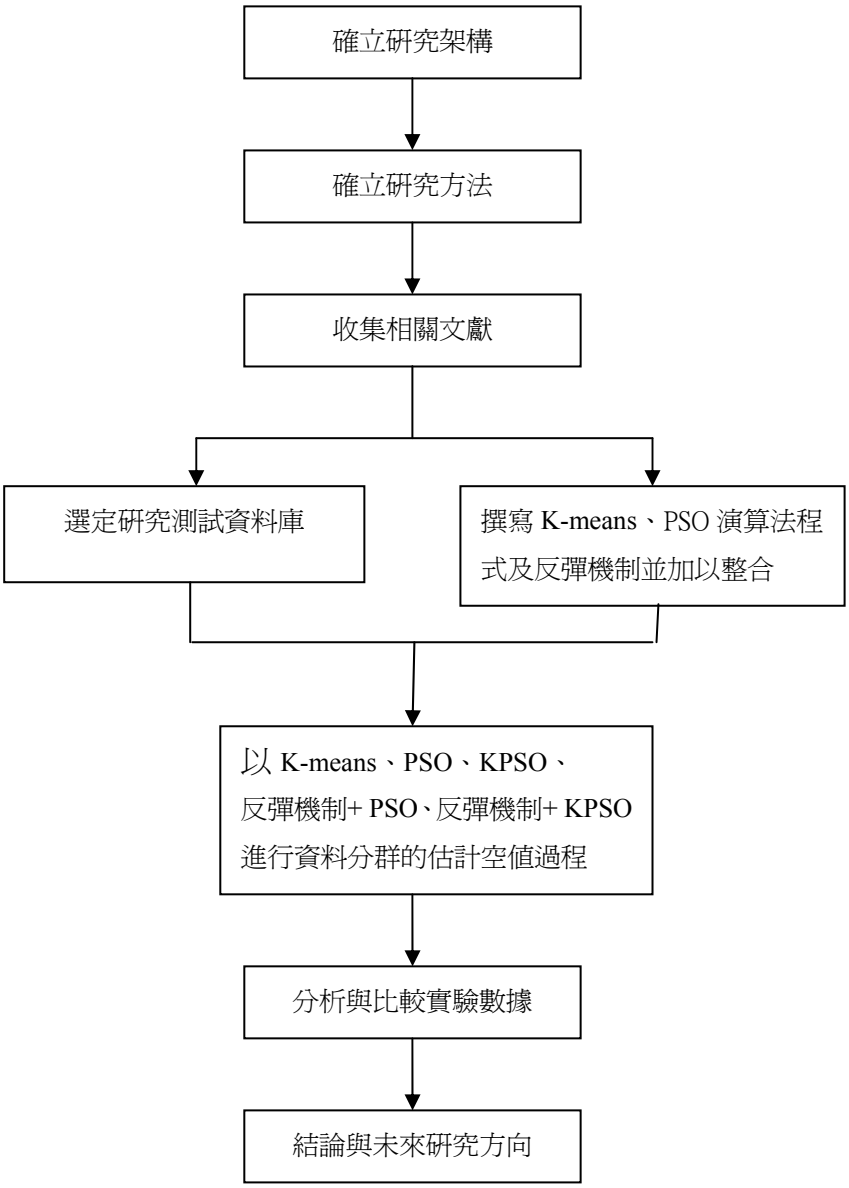


圖 1.2 研究流程圖

第四節 論文架構

本研究其餘章節組織如下:第二章爲文獻回顧及整理研究相關的背景知識，第三章將介紹主要的研究方法以改良式粒子演算法爲基礎之遺漏值推估程序(Estimate null value in relational database system based on PSO-clustering),詳細描述本研究的演算程序。第四章將使用本研究提出的方法進行個案實驗，及使用絕對誤差率平均 MAE(Mean of Absolute Error)[1]爲評估標準與其它估計方法比較，證實本研究方法在估計遺漏值上較其它技術精確。第五章爲研究結論、未來方向以及相關建議等討論。

# 第二章 文獻探討

本章介紹包含二個部份。第一部份包含本研究所用到的相關技術研究文獻，第二部份則此研究所需的相關背景知識，方便讀者更進一步了解本研究的目地、作法及研究方向。

## 第一節 處理遺漏值方法

如表 2.1 針對遺漏值問題在早期就有許多填補的方法，因此估計遺漏值並非十分新穎的題目，現今在不同領域也有許多學者紛紛提出研究方法。下列則簡要介紹目前常用的方法及相關文獻。

多重插補(multiple imputation)[12]是一種資料擴充和統計分析方法，主張用一系列可能的值來替換每一個遺漏值，來反應被替換的遺漏資料的不確定性，再針對每個填補資料集合都以完整資料集的統計方法進行統計分析，最後綜合每個填補資料集的結果，得到最終的統計估計值。然而，多重插補將空缺值視爲隨機樣本，因此所求得的統計估計值可能受到遺漏值的不確定性的影響，加上需要產生大量的插補值及計算過程複雜，因此大幅增加估算的時間和成本。

MVC(missing values completion)引用關連法則(Association Rule)[13]發展出容錯關連規則( Robust Association Rules, RAR)，在資料間的找出滿足一定的信賴度(confidence)與支持度(support)的關聯規則，並利用這些相關規則產生許多遺漏值候選解。但是這僅適用於關聯性較強的資料，若應用在某些資料間關聯薄弱的資料庫則推估效果不佳。

使用機械式學習[14]來處理遺失值，讓電腦從歷史資料中學習，再以其所學習的經驗來預測遺漏值。像是類神經網路的學習演算法，透過不斷的訓練找出輸入及期望輸出間的差異，調整加權值來達到學習的效果。但是，機械式學習必需以大量的學習經驗來建立預測的準確率，若是遇到較特別的個案又要重新學習，在實用上需耗費相當的時間成本。

分割式推估演算法，主要是以分群技術爲基礎將資料分爲 K 個所需叢集後再以所適用之推估摸式進行推估，而目前已有許多分群演算法像是 K-means[15]、K-medoid[16]、螞蟻演算法(Ant Colony Optimization，ACO)、粒子群最佳化演算法(Particle swarm optimization PSO)和禁忌搜尋法(Tabu Search，TS)等方法

如K-means演算法，是利用分群過後群內的平均值回填入該群的遺漏值欄位，而K-medoid演算法則是找出群中最接近群心值的資料點，回填入該群的遺漏值欄位。在文獻[17]中，分別以二種分群技術k-mean和自動分群技術對資料進行分群動作後利用歐氏距離(Euclidean distance)計算含有遺漏值的資料到各群心的距離，將其資料歸入所求距離最小之群集，再以模糊關聯(fuzzy correlation)找出屬性間的關聯程度及用統計計算式，計算屬性間的決定係數找出該遺漏值由其它屬性影響的權重後進行推估。然而，此研究是以自變數去估計依變數，但若遺漏值出現在自變數則無法有效估計。

在文獻[1]中，研究者利用自組織映射圖網路(self-organize mapping ，SOM)將資料進行分群後，以群內平均回填該群遺漏值欄位，其目的在於利用類神經網路的學習讓推估模式在推估過程中記取經驗以達到最佳推估的目的。可是這類的推估模式皆有一弱點，則是如果運用到真實資料庫中需要一定的學習時間，因此遇到不同的資料型態則無法做即時的有效判斷。

表 2.1 處理遺漏值技術分析表

| 方　法 | 特　　色 | 缺　點 |
|---|---|---|
| 刪除法 (Delete) | 直接刪除存有遺漏值的資料，以確保資料完整性。 | 若資料量不大，將造成資料量縮減，導致可分析和挖掘的資訊變少。 |
| 人工比對 (Artificial Alignment) | 以相關測量或題目類比的邏輯推理法，將遺漏值以最有可能出現的答案填補。 | 人易受時.地.心情等內外在因素影響，因此所估出的值並不客觀。 |

| | | |
|---|---|---|
| 固定填補 (Fixed Fill) | 從不含有遺漏值的資料中，將相同屬性欄位以機器式學習(Machine leaing)方法，找出一個固定值填補。 | 雖省時省力，但無法確保值的正確性，可能存有某種程度上的偏差(bias)。 |
| 眾數填補法 (Mode Fill ) | 以資料庫同屬性全部資料中出現次數最多的值當作遺漏值的回填。 | 若屬性資料出現不具有高度重覆性的話，在分析應用上將有其限制。 |
| 平均數填補 (Average Fill) | 以資料庫中不含有遺漏值的同屬性資料計算出平均回填。 | 易受極端值或資料型態分佈影響，導致所求出的平均值有所偏差。 |

第二節 群集分析

群集是將有形的或抽象的物件歸類到相似物件類別的過程，目的在找出同群集資料中的相似性，及各群集之間的差異性，使得同群中資料相似度最大，而各群之資料差異度最大，在這我們介紹兩種集群分析法，分別是階層式分群與分割式分群，下面將簡短介紹階層分群、分割式分群[18]:

一.階層式分群法（hierarchical clustering）:

階層式分群法透過一種巢狀架構的方式，將資料層層分裂或聚合，整個階層式分群法可以由圖 2.1 的樹狀結構來表示。如果採用聚合(Agglomerative)的方式，階層式分群法可由樹狀結構的底部開始；如果採用分裂(Divisive)的方式，則由樹狀結構的頂端開始層層分裂。

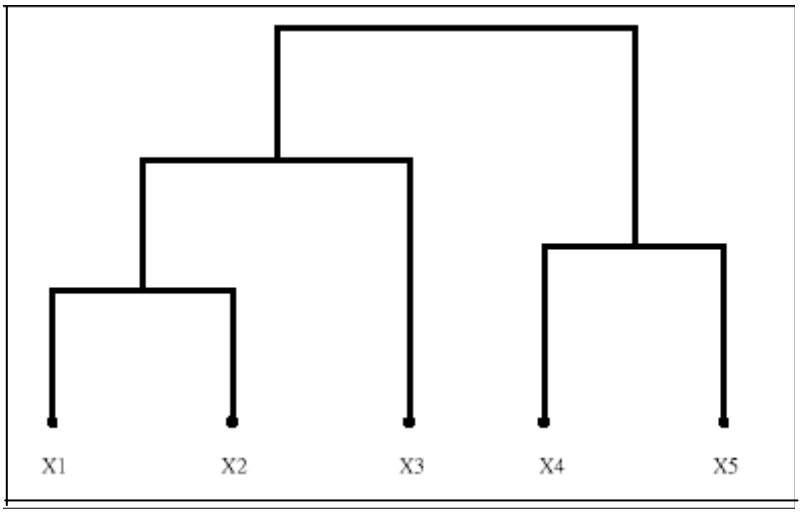


圖 2.1: 階層式分群法樹狀結構

a) 聚合式亦稱爲由下往上(bottom-up)分群法:

階層式聚合演算法由樹狀結構的底部開始層層聚合。一開始我們將每一筆資料視爲一個群聚（cluster），依照距離測量合併最接近的它群，直到全部物件皆合併成一群或是達到所需群數爲止。舉例來說，假設我們現在擁有 n 筆資料，則將這 n 筆資料視爲 n 個群聚，估計步驟如下：

1. 將每筆資料視爲一個群聚 Ci
2. 找出所有群聚間，距離最接近的兩個群聚 Ci、Cj，群與群間距離測量方法有下列幾種，其中 $m_i$ 表示 Cluster $C_i$ 的平均值；d(a,b) 表示兩物件間的距離:

(1)單一連結凝聚演算法(single-linkage agglomerative algorithm)

不同群聚中最接近兩點間的距離 $D_{min}(C_i, C_j)$:

$$D_{\min}(C_i, C_j) = \min_{a \in C_i, b \in C_j} d(a,b) \qquad (2\text{-}1)$$

(2)完整式連結凝聚演算法(complete-linkage agglomerative algorithm)

不同群聚中最遠兩點間的距離 $D_{\max}(C_i, C_j)$:

$$D_{\max}(C_i, C_j) = \max_{a \in C_i, b \in C_j} d(a,b) \qquad (2\text{-}2)$$

(3)平均連結凝聚演算法(average-linkage agglomerative algorithm)

不同群聚中平均兩點間的距離 $D_{avg}(C_i, C_j)$ :

$$D_{avg}(C_i, C_j) = \frac{1}{|C_i||C_j|} \sum_{a \in C_i} \sum_{b \in C_j} d(a,b) \qquad (2\text{-}3)$$

(4)重心連結凝聚演算法(average-linkage agglomerative algorithm)

以群聚重心間的間距爲量測依據 $D_{mean}(C_i, C_j)$ :

$$D_{mean}(C_i, C_j) = |m_i - m_j| \qquad (2\text{-}4)$$

3. 合併 Ci、Cj 成爲一個新的群聚 Cij
4. 假如目前的群聚數目不只一個且多於我們預期的群聚數目，則重複步驟二

b) 分裂式亦稱由上往下(top-down)分群法:

階層式分裂演算法顧名思義由樹狀結構的頂端開始，如同細胞分裂般層層將直徑最大的群聚一分爲二，或者是更多更小的群聚，直到群聚數目達到我們預期的目標。而所謂群聚的直徑 D 指的是一個群聚中，最遠兩點間的距離。假設現在有一個群聚 C，那麼 C 的直徑可以表示如公式(2-5)在介紹階層式分裂演算法之前，我們要先定義某一點 x 與某個群聚 C 的距離 d(x, C)。假設群聚 C 中包含 n 筆資料如公式(2-6)[22]：

:

$$D(C) = \max_{a \in C, b \in C} d(a,b) \qquad (2\text{-}5)$$

$$d(x,C)=\begin{cases}\frac{1}{n}\sum_{y\in C} d(x,y) & ,x\notin C\\ \frac{1}{n-1}\sum_{\substack{y\in C\\ x\neq y}} d(x,y) & ,x\in C\end{cases} \quad (2\text{-}6)$$

階層式分裂演算法步驟如下:

1. 將全部的資料視爲同一個叢聚
2. 在現有的叢聚中，挑出直徑最大的叢聚 C
3. 在 C 中找出最不相似的一點 x，亦即 d(x, C) ≥ max d(y, C) ，其中 y 屬於 C
4. 將 x 由 C 中分裂出來形成新的叢聚 N。假設原有的叢聚 C 中仍剩下的資料稱爲 Mc
5. 重複步驟六、七，直到 C 與 N 都不再變化
6. 計算 Mc 中每筆資料 m 與叢聚 C 的距離及叢聚 N 的距離 d(m, C) , d(m, N)
7. 假如 d(m, C) > d(m, N)，則將 m 由叢聚 C 分裂出來歸入 N
8. 如果目前的叢聚數目仍然少於資料個數且少於我們預期的標準，則回到步驟 2

二.分割式分群法（partitional clustering）:

在使用分割式分群法（partitional clustering）時，我們必須先指定群聚的數目，然後藉著反覆疊代運算，逐次降低一個誤差目標函數的值，直到目標函數不再變化，就達到分群的最後結果。一般而言，分割式分群法的目的是希望盡量減小每個群聚中，每一點與群中心（cluster center）的距離平方差（square error）。假設我們現在擁有一組包含 c 個群聚的資料，其中第 k 個群聚中包含 $nk$ 筆資料（$X_1, X_2, ..., X_{nk}$），此群聚中心爲 $X_k$，則該群聚的平方差 $e_k$ 可以定義如公式(2-7)而這 c 個群聚的總和平方差 E 便是每個群聚的平方差總和如公式(2-8) [18]：

$$e_k = \sum_{i=1}^{n_k} (x_i - x_k)^2 \qquad (2\text{-}7)$$

$$E = \sum_{j=1}^{c} e_j \qquad (2\text{-}8)$$

第三節 K-means 分群技術

集群分析係根據樣本的某些特性之相似程度，將樣本劃分成幾個集群，使同一個集群內的樣本具有高度之同質性，而不同集群間之樣本則具有較高度的異質性。而集群分析依照分類的方式不同可分爲階層式集群分析(Hierarchical Cluster Analysis)及非階層式集群分析(Nonhierarchical Cluster Analysis)，K-means 便是屬於非階層式集群分析裡最常被使用的一種方法[1]。K-means 演算法的演算過程如下：

1.隨機產生初始群體中心點。

2.待分群的資料點爲 n 個；分別爲 $X_i$ ， $i = 1, 2, ..., n$ ，計算資料點到群體中心的歐基里德距離，將資料點歸入距離值最小的群集。

3.以公式(2-9)計算各群內的中心點，並更改爲該群的新中心點。

$$V_j = \frac{1}{N_j} \sum_{X_i \in C_i} X_i \qquad (2\text{-}9)$$

$V_j$:群心　　$C_j$:所屬群集　　N:資料點數　　$X_i$:第 i 個資料點

J=1,…,k

4.重新執行步驟 2，直到群體中心點不再改變，才可結束迴圈。

K-MEANS 分群法之演算流程雖然簡易，卻無法有效處理大量的資料分群數目及解決資料點重疊狀況，因此不能算是一種完善的分群技術。

## 第四節 粒子群最佳化演算法( Particle swarm optimization PSO)

### 一. 粒子分群最佳化演算法:

在粒子群最佳化演算法提出之前便有學者藉由關察動物覓食習慣針對最佳化問題提出演算法，如 Dorigo(1995)[19]提出的螞蟻演算法(Ant Colony Optimization,ACO)，Dorigo 概念爲螞蟻可由蟻穴走一條最短路線到達食物目的地,因爲它們利用之前螞蟻走過的地方所殘留的一種分泌物費洛蒙(pheromone),之後的螞蟻經過時, 就有較高的機率選擇較多螞蟻走過的路徑也就是費洛蒙濃度高的方向,而隨著時間增長,漸漸螞蟻會走同一路線由蟻穴到食物目的地來回路徑(即最短路線), 利用這種自然界的原理已有效率地解一些最佳化問題(Optimization Problems)。

藉由模擬簡單的個體組成的環境與群落以及個體之間的互動行爲過程，利用局部訊息進而推敲不可預測的群體行爲,而螞蟻演算法與粒子群最佳化爲都是基於群智能的演算法，我們將這種相關模擬研究稱爲群體智能（Swarm Intelligence）。粒子群最佳化具有兩個主要的基本概念，其一爲藉由觀察人類的決策過程（Boydand Richerson, 1985）[20]，以人類會相互學習以吸取最佳經驗作爲決策之依據，建立出個體學習與文化傳遞的兩種觀念。另一則是以自然界生物的群體行爲提出一些簡單的法則，將其行爲模組化。Reynolds(1987)[21]提出群體中每個個體行爲可以藉由下列三種向量方式模組化，並且產生複雜的群體行爲:

(1)跟隨離目標最近的個體移動

(2)朝著目標移動

(3)朝群體中心移動

基於上述的兩個基本概念和模擬鳥群飛行的群體行爲，Eberhart and Kennedy[24]發展出粒子群最佳化演算法。粒子群最佳化演算法是 Eberhant 和 Kennedy 以觀察鳥群和魚群覓食行爲而提出，因此我們以鳥群覓食的情形來做推敲，假設有一隻鳥發現食物後發出訊號通知同伴,但對於其它離較遠的鳥只知道訊號大約離自

己多遠卻不能準確的知道食物所在地，因此他們藉由前往目前搜尋周圍區域中距離食物最近的同伴，不斷重覆此步驟直到目的地。

鳥群中的每一隻鳥稱爲「粒子」，這些粒子的位置以$X_{id}$表示，而每一個粒子對於每個維度的飛行速度則以$V_{id}$來表示，其中i指的是第i個粒子，d 則代表粒子所搜尋之空間的維度。如圖2.2在粒子群最佳化中每一個粒子在搜尋中都有自身的速度及方向，並且根據自己過去最佳經驗($p_i$)和群體行爲經驗($p_g$)，逐漸修正自己的速度(v)與位置(x)，使初始隨機散佈於可行解區域的粒子能接近整體最佳目標值的粒子附近，形成粒子群在此一區域進行搜尋，而這一區域可能只是區域最佳解(Local Optimal)，但經由各粒子的搜尋可修正群體最佳值的位置，使所有粒子產生群體效用而逐漸逼近最佳值。因此可將粒子群最佳化歸列成下列幾項特性:

（1）分散式搜尋

（2）具記憶性

（3）元件較少，容易實現

（4）適合在連續的範圍內搜尋

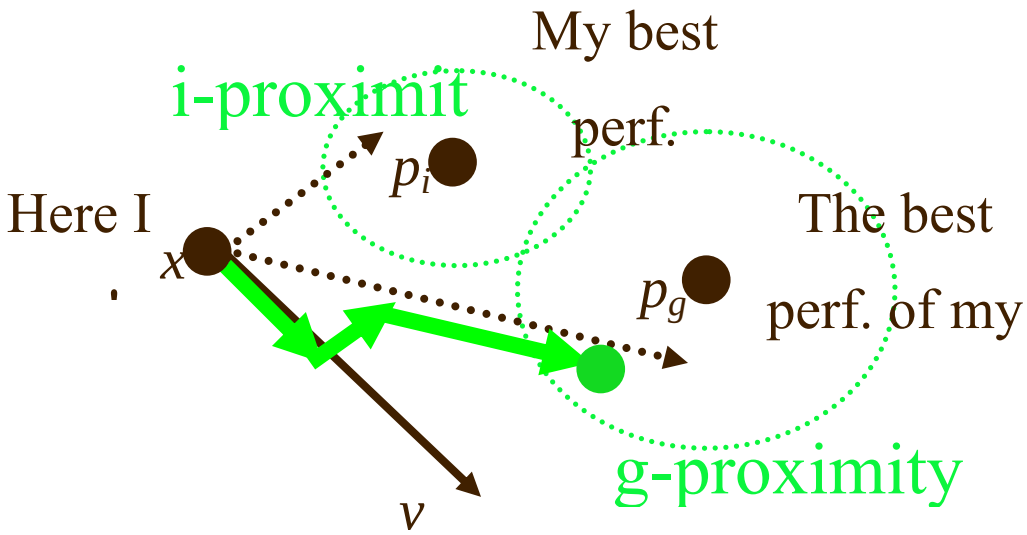


圖2.2 粒子受自我和社會認知之影響圖[10]

二.粒子群最佳化演算法流程

在搜尋空間中每個粒子都有一個對於最佳化問題的適應值（Fitness value），且知道自己目前的最佳適應值和最佳的位置（Particle best value, $p_{id}$ ），這些資訊爲每個粒子自己所擁有的經驗。同時每個粒子也會記錄目前群體最佳值和最佳的位置（Globe best value, $p_{gd}$ ），並且根據自我過去經驗與群體行爲進行機率式的搜尋策略調整。

1. 初始族群中每一個粒子在 n 維空間裡的位置與速度以隨機的方式產生。
2. 針對所設定的目標函數，評估每一個粒子的適應函數值。
3. 適應函數值與粒子本身的最佳函數值記憶比較，而粒子依照個體最佳變數記憶去修正下一次變數搜尋的粒子速度。
4. 個體最佳函數值與群體最佳函數值的最佳化程度作比較，如個體最佳值優於群體最佳值，則修正群體最佳函數值的變數記憶；同時每個粒子依照群體最佳變數記憶來修正下一次變數搜尋的粒子速度。
5. 利用下列核心運作方式改變粒子的速度與位置，即執行PSO 的計算式(2-10)來得到每個粒子每個參數維度新的移動速度，並以計算式(2-11)改變每個粒子的位置。

$$\mathrm{V}_{id}^{\mathrm{New}} = w \times \mathrm{V}_{id}^{\mathrm{old}} + c_1 \times rand() \times (p_{id} - X_{id}^{\mathrm{old}}) + c_2 \times rand() \times (p_{gd} - X_{id}^{\mathrm{old}}) \qquad (2\text{-}10)$$

$$X_{id}^{\mathrm{New}} = X_{id}^{\mathrm{old}} + \mathrm{V}_{id}^{\mathrm{New}} \qquad (2\text{-}11)$$

*rand*() 是介於 0 到 1 之間的亂數，w 爲加速度，產生方式 $[0.5 + (rand()/2)]$ $V_{id}^{old}$ 爲目前的速度， $c_1$ 與 $c_2$ 則是社會與自我認知權重[22]，通常皆設定爲 2[10]， 爲個別粒子紀錄中曾經到達的最佳解， 是全部群體粒子紀錄中曾經到達的最佳解， $X_{id}^{old}$ 表示目前粒子所在的位置，從公式(2-10)中可得更新後的速度 $\mathrm{V}_{id}^{\mathrm{New}}$ ，再利用公式(2-11)便可求得新的群體位置 $X_{id}^{\mathrm{New}}$ 。

6. 重複步驟3 及步驟4 評估函數值以及更新速度及位置，比較是否優於之前所紀錄之全域以及區域最佳解，若較佳則取代之，反之，則以之前紀錄最佳值進行運算。
7. 重複步驟3.至6.直到演算回合數符合演算停止條件，便可求得群體最佳參數。

三.粒子群最佳化分群演算法流程

以圖 2.3 這例子來說，圖 2.3 爲 PSO 初始母體中的一獨立粒子，假設 n=2，K=3，PSO 則代表此爲二維搜解空間及分爲三群，而圖中的粒子則代表 3 個群心[(-4.5,9),(23,15),(3.46,15.23)]其詳細演算法演算流程如以下[11]:

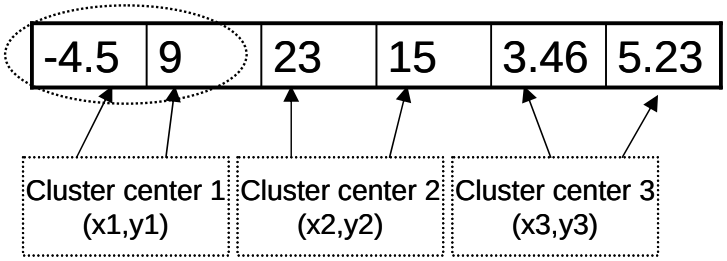


圖2.3 PSO初始母體中的一獨立粒子

1. 設計出所需的評估函數，使得每個粒子所搜尋到的位置組都有一對應的數據值。
2. 決定粒子數目，通常設定爲5N，N可由公式2-12計算而得[10]。隨機產生初始群中所有個體的位置向量 $X_{id}$ 以及移動速度 $V_{id}$ ，此處位置向量 $X_{id}=\{C_{id,j}, j=1,2,...,K\}$ 代表分群群心的集合，總群聚分割數爲 $K$ ，而每一個群聚 $C_{id,j}=(C_{id,j1},C_{id,j2},...,C_{id,jN})$ 均爲維度爲N的向量，並設定PSO 演算法中的所有參數，以及決定迭代次數。

N=維度*資料大小 (2-12)

3. 將粒子的所在位置代入評估函式，每個粒子依傳回評估值，並記錄全體粒子中最佳的位置，以及粒子個別最佳位置。在本研究中則以公式2-13，歐氏距離(Euclidean distance)作爲評估函式，其中$C_{id}$則爲各群群心。

$$D(p_{id}, C_{id}) = \sqrt{\sum_{id}(p_{id} - C_{id})^2}; id = 1, 2, ..., c \qquad (2\text{-}13)$$

4. 執行PSO 的計算式(2-14)來得到每個粒子每個參數維度新的移動速度，並以計算式(2-15)改變每個粒子的位置。

$$V_{id}^{New} = w \times V_{id}^{old} + c_1 \times rand() \times (p_{id} - X_{id}^{old}) + c_2 \times rand() \times (p_{gd} - X_{id}^{old}) \qquad (2\text{-}14)$$

$$X_{id}^{New} = X_{id}^{old} + V_{id}^{New} \qquad (2\text{-}15)$$

5. 重複步驟3 及步驟4 評估函數值以及更新速度及位置，比較是否優於之前所紀錄之全域以及區域最佳解，若較佳則取代之，反之，則以之前紀錄最佳值進行運算。

6. 重複步驟3至5直到演算回合數符合演算停止條件，便可求得群體最佳參數。

PSO 虛擬程式碼如圖 2.4[30]:

For t=1 to Max_iter

$$w_1 = (w_1 - w_2) \times \frac{Max\_iter - t}{Max\_iter} + w_2$$

For i=1 to N // N is the population size

For d=1 to D

$$v_{id}(t) = w_1 v_{id}(t-1) + c_1 r_1 (pbest_{id} - x_{id}(t-1)) + c_2 r_2 (gbest_d - x_{id}(t-1))$$

$$v_{id}(t) = \min(v_{d\max}, \max(-v_{d\max}, v_{id}(t)))$$

// Limit the velocity

$$x_{id}(t) = x_{id}(t-1) + v_{id}(t-1)$$

// Update the position

End For d

If $X_i \in [X_{\min}, X_{\max}]$

```
        Calculate the fitness value of X_i ,Update
        pbest_i and Gbest if needed
    EndIf
 End For i
 Stop if a stop criterion is satisfied
End For t
```

圖 2.4 PSO 虛擬程式碼

四.粒子群最佳化相關發展與應用

Eberhart and Kennedy(1995) [23]提出可使用粒子群最佳化演算法學習類神經網路，在精確分類 XOR 的問題上，能夠在具有 13 個維度的搜尋空間內搜尋函數的最小值，而目前也相當多的學者將 PSO 運用在類神經學習上做研究。Kennedy and Shi (1998)[25]曾比較二進位粒子群最佳化(BPSO)與三種不同的基因演算法，證實二進位粒子群最佳化演算法求解時間短，且在問題搜尋的空間維度變化上較不受影響，因此求解效果佳。在文獻[11]PSO 應用在分群上與 K-means 相比，證明 PSO 在分群上有顯著成效，在文獻[22]則可看出將 K-means 與 PSO 結合後可增加 PSO 最佳化演算法之收斂速度。

目前國內也有許多學者利用 PSO 或改良式 PSO 來解決各種領域的問題: 在文獻[2]應用 PSO 歸類及整合電力負載的用戶，文獻[3]應用 PSO 在多目標條件下進行分類存貨，文獻[4]提出質群演算法(PSO)於多組解方程最佳化問題之研究，文獻[5]提出利用 PSO 演算法探討高速銑削最佳化，文獻[6]以 PSO 爲基礎的臉部偵測系統，文獻[7]基於粒子群最佳化演算法之奈米定位控制系統設計等。

而在文獻[10]中提到雖然PSO在分群問題上有不錯的表現，但是當資料群過大或複雜的時候，則無法達到良好的分群結果。因爲當粒子到達搜尋空間的邊界的時候，他們傾向在那裡停留而不能找到更好的解決途徑移動到其他的方向如圖 2.5(a)。因此，文獻[10]在 PSO 中提議二個反彈方案，將粒子從邊界向其他的方向推動以恢復粒子尋解能力。

第一個方案:

（PSO＋R1)反彈機制是將超出搜尋空間外的粒子拉回尋解空間內。計算反彈的距離是，以$\alpha$乘上尋解空間的最大和最小邊界之間距，尋解空間的最大和最小邊界是由資料集中每一維度最大最小值所構成，而$\alpha$爲介於0到0.25的參數。假若粒子移動超過尋解邊界，則將其反彈尋解空間範圍內的$\alpha$百分比，如圖2.5(b）。

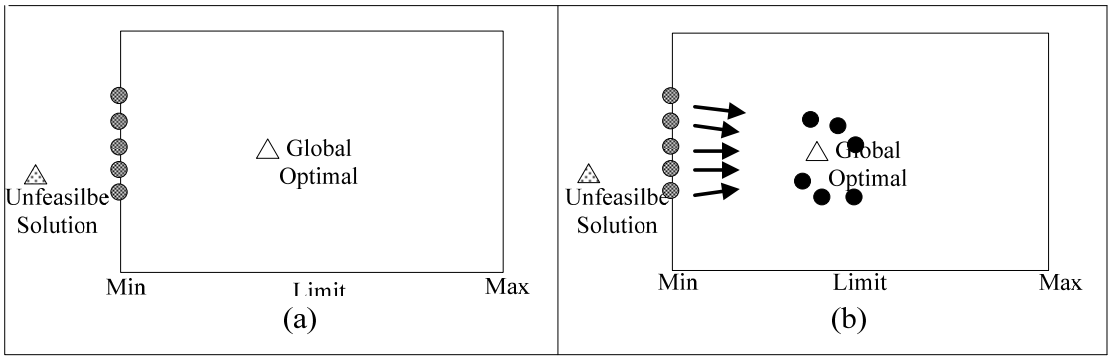


圖2.5:(a)原始PSO (b)PSO加反彈機制

第二個方案:

利用公式(2-16)(2-17)得到粒子新位置$X_{id}^{New}$，其作法是將超出尋解空間的粒子，以加上或減掉一正數值後，乘上一介於0到1間的隨機值拉回尋解空間內，其正數值爲群體最佳值($p_{gd}$)減掉粒子原本位置($X_{id}^{old}$)的絕對值。LB和UB則分別爲資料群集中，每個向量的最小值和最大值而rand則從0至1中隨機選取。

$$X_{id}^{\text{New}}=\begin{cases} X_{id}^{\text{old}}+\left|p_{gd}-X_{id}^{\text{old}}\right|\times rand, if \text{ particle move out of LB} & \text{(2-16)} \\ X_{id}^{\text{old}}-\left|p_{gd}-X_{id}^{\text{old}}\right|\times rand, if \text{ particle move out of UB} & \text{(2-17)} \end{cases}$$

五 K-means分群與PSO分群及改良PSO分群之分析比較表

就目前而言最常拿來做爲分群工具的便是K-means，因此在表2.2則針對K-means分群與啓發式演算法PSO分群作爲分析和比較。而在表2.3，我們則把PSO

及其它改良PSO分群作分析比較以方便了解各改良後PSO其作用和目的爲何。

表2.2K-means分群與PSO分群分析表

| | K-means | PSO |
|---|---|---|
| 分群法類別 | 分割式 | 演化式 |
| 群聚數目 | 自行設定 | 自行設定 |
| 優點 | 簡單、快速 | 較其它啓發式演算法，參數來的少 |
| 缺點 | 隨機初始值完全決定分群結果 | 需不斷演化，若資料複雜或龐大則需要較多演化時間。 |

表2.3PSO分群與改良PSO分群分析表

| | PSO | KPSO | RPSO | RKPSO |
|---|---|---|---|---|
| 群聚數目 | 自行設定 | 自行設定 | 自行設定 | 自行設定 |
| 演算法特色 | 參數少 | 加速 PSO 收斂速度 | 避免 PSO 執行時落入邊界外 | 整合 KPSO 及 RPSO 之優點 |

# 第三章 整合反彈機制與 KPSO 之遺漏值推估模式

本章將介紹利用反彈機制加 KPSO 分群用來估計遺漏值的想法及流程。在本研究中選用文獻[10]中的第二種反彈機制，其原因在於第一種反彈機制需設定參數值，而參數值的設定極有可能會影響實驗結果，因此本研究以第二種反彈機制進行實驗。

## 第一節 改良 PSO 分群演算法

本研究中分別使用K-MEANS、PSO、KPSO、反彈機制+PSO、反彈機制+KPSO實驗以驗證反彈機制+KPSO分群法，是否在不同的條件下，對遺漏值的估計效果都能優於其他四種分群方法。本論文所使用之分群方法，鎖定爲反彈機制+KPSO，期望透過實驗能證實其估計效果較其它技術好，而能對估計遺漏值問題有所貢獻。

如圖3.1所示首先使用K-means分群法加以分群，當所得結果收斂時，再使用反彈機制+PSO演算法，找出最佳解，若K-means所得結果陷入區域最佳解，則反彈機制+PSO將跳出區域最佳解，另尋全域最佳解，若在K-means所得結果，介於全域最佳解之區域內，則PSO演算法，將在範圍內，找尋全域最佳解。也就是因爲有此優點，因此收斂狀況、分群效果與遺漏值估計誤差能優於PSO與反彈機制加PSO。

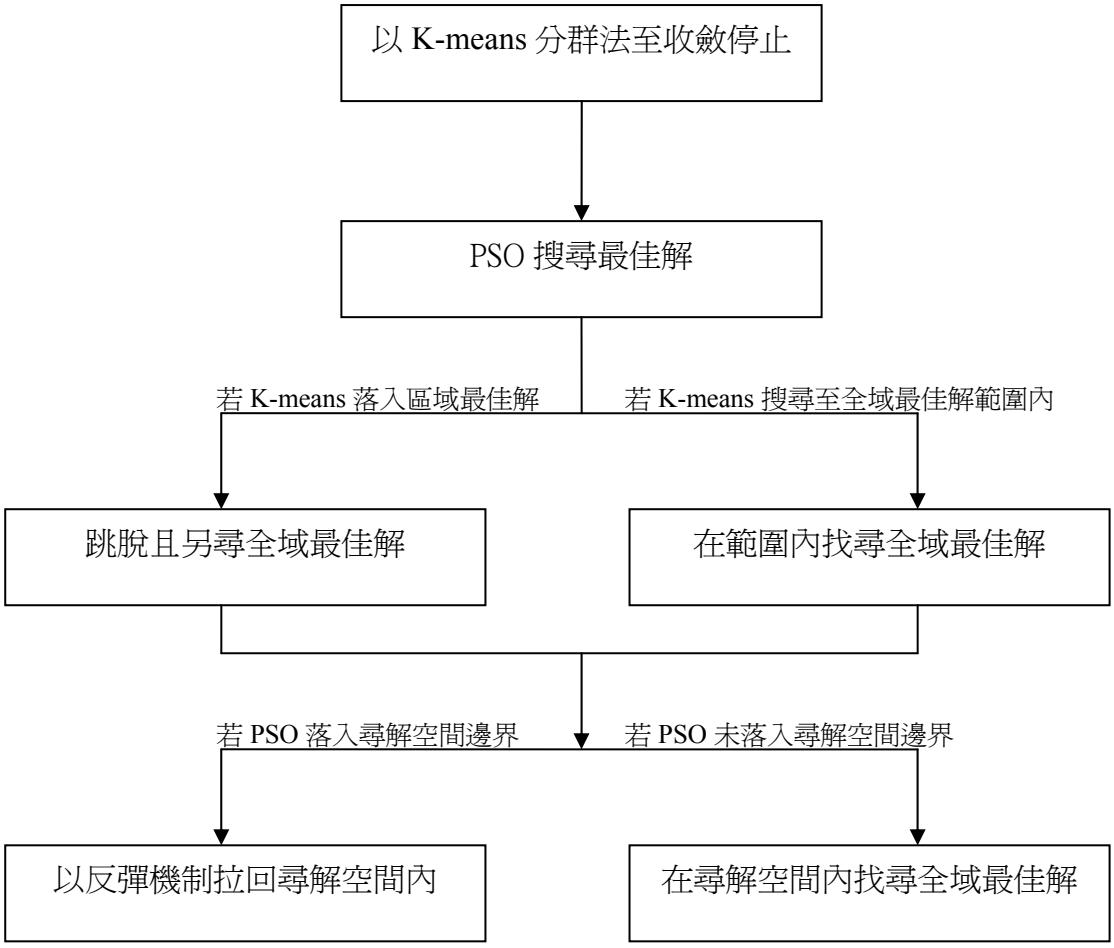


圖3.1 反彈機制+KPSO演算法架構圖

一. 反彈機制+KPSO 分群演算法流程

由文獻[10]中所提出的RPSO分群，可避免粒子陷入邊界以加強分群效果。因此本研究針對K-means分群法可快速收斂[22]，反彈機制可避免落入尋解空間邊界的二項優點整合反彈機制+KPSO演算法，其RKPSO分群演算法流程如圖3.2

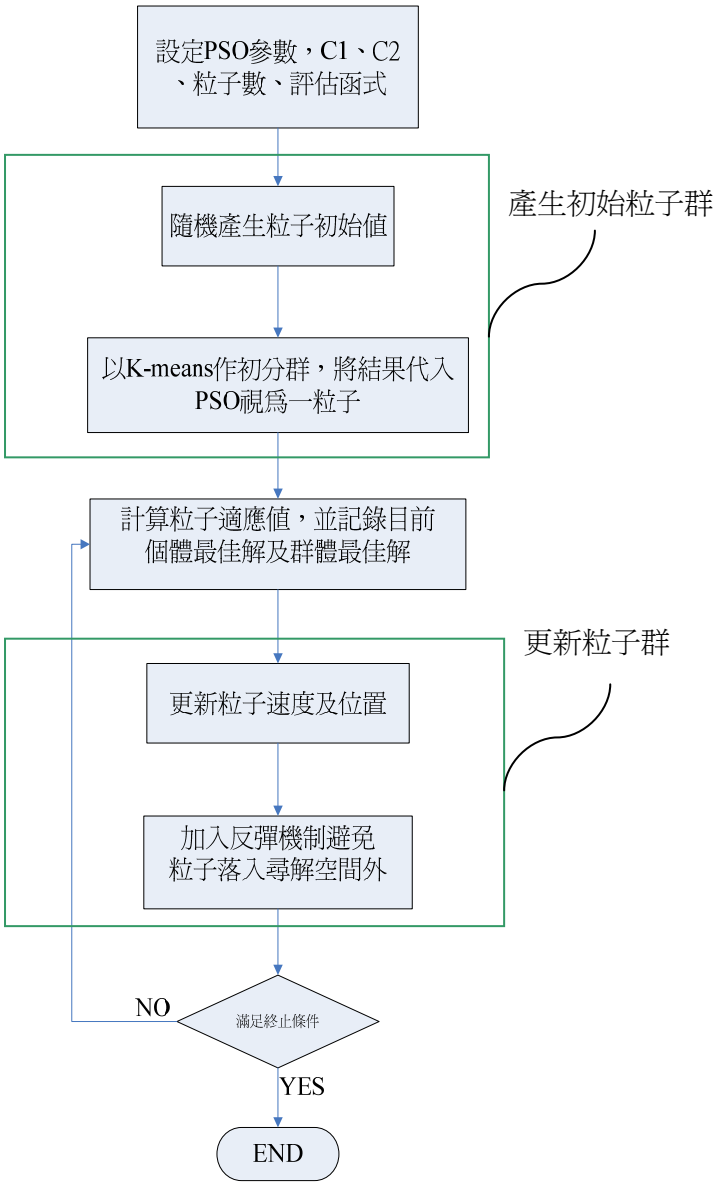


圖 3.2 RKPSO 分群演算法流程

1. 先利用K-means分群法快速收斂之優點，讓資料做初步分群後將分群結果視爲一粒子代入PSO分群演算法內當作PSO分群基礎最佳解。
2. 設計出所需的評估函數，使得每個粒子所搜尋到的位置組都有一對應的數據值。一般來說，分群技術評估績效指標，通常爲計算各群體中心與同一群的資料點距離加總，當距離總和越小則代表分群越成功。公式(3-1)爲分群評估函數Fitness，$x$是待分群資料點，共有$n$個，z是代表分群中心點，共有$K$個，。

$$fitness = \sum \left\| x_j - z_i \right\| \qquad (3\text{-}1)$$

$$j = 1,...,n \qquad i = 1,...,K$$

3. 決定粒子數目，通常設定爲5N，N可由公式3-2計算而得[10]。初始位置及速度以亂數產生，並設定PSO 演算法中的所有參數，以及決定迭代次數。

$$N=K*d \qquad (3\text{-}2)$$

K=群數，d=維度

4. 將粒子的所在位置代入評估函數，每個粒子依傳回評估值，並記錄全體粒子中最佳的位置，以及粒子個別最佳位置。
5. 執行PSO 的計算式(3-3)來得到每個粒子每個參數維度新的移動速度，並以計算示(3-4)改變每個粒子的位置。

$$V_{id}^{New} = w \times V_{id}^{old} + C_1 \times rand() \times (p_{id} - X_{id}^{old}) + C_2 \times rand() \times (p_{gd} - X_{id}^{old}) \qquad (3\text{-}6)$$

$$X_{id}^{New} = X_{id}^{old} + V_{id}^{New} \qquad (3\text{-}7)$$

$rmad()$ 是介於0 到1 之間的亂數，w爲加速度，產生方式爲$[0.5 + (rand()/2)]$，$V_{id}^{old}$ 爲目前的速度，$C_1$ 與 $C_2$ 則是社會與自我認知權重，通常設定爲2[10]，$p_{id}$ 爲個別粒子紀錄中曾經到達的最佳解，$p_{gd}$ 是全部群體粒子紀錄中曾經到達的最佳解，$X_{id}^{old}$ 表示目前粒子所在的位置，從公式(3-3)中可得更新後的速度 $V_{id}^{New}$ ，再利用公式(3-4)便可求得新的群體位置 $X_{id}^{New}$ 。

6. 利用反彈機制公式(3-5)(3-6)[10]，將落入尋解空間邊界解拉回尋解空間內，使其最佳解不至於落入尋解空間外。LB和UB則分別爲資料集中，每個向量的最小值和最大值而rand則是從0至1中隨機選取。

$$X_{id}^{\text{New}} = \begin{cases} X_{id}^{\text{old}} + \left| p_{gd} - X_{id}^{\text{old}} \right| \times rand, if \text{ particle move out of LB} & (3\text{-}5) \\ X_{id}^{\text{old}} - \left| p_{gd} - X_{id}^{\text{old}} \right| \times rand, if \text{ particle move out of UB} & (3\text{-}6) \end{cases}$$

7. 重複步驟4至步驟6 評估函數值以及更新速度及位置，比較是否優於之前所紀錄之全域以及區域最佳解，若較佳則取代之，反之則以之前紀錄最佳值進行運算。
8. 重複步驟4至7直到演算回合數符合演算停止條件，便可求得群體最佳值。

第二節 遺漏值估計方法

從文獻看出將分群應用於遺漏值估計問題上大致可分於下列三種作法:

(a) 分群收斂後直接回填[17]，其作法是將資料分群穩定後，以每一群的平均值回填遺漏值欄位。

(b) 如文獻[1]不斷重覆資料分群穩定後回填，直至回填值穩定才結束估計流程，在此我們稱之爲 Run-based。

(c) 在此我們則將回填步驟置於每一次迭代後，讓回填值跟隨著每一次迭代作更改，在此我們稱之爲 Iteration-based。

(一) Run-based 估計模式:

Run-based 估計模式是以分群穩定後再作回填，其流程如圖 3.3 而詳細說明如下:

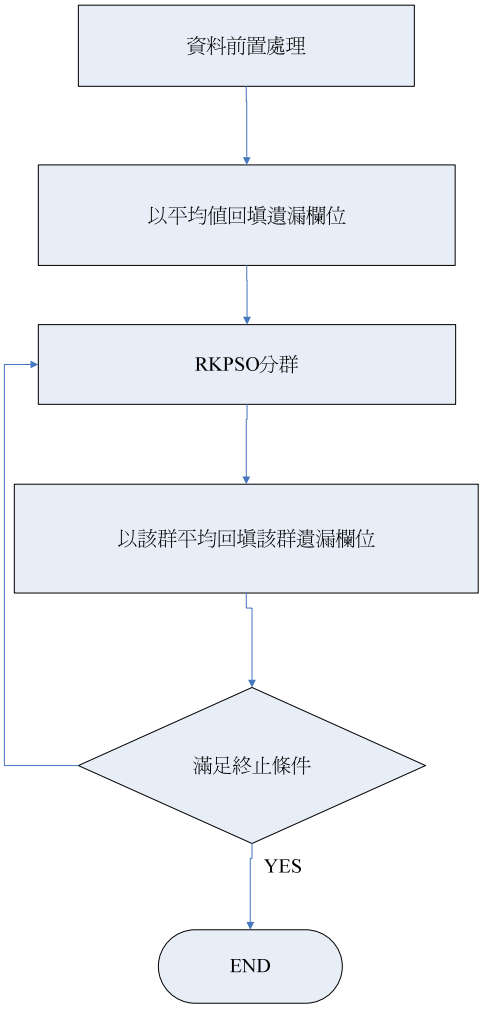


圖 3.3 Run-based 估計模式流程圖

1. 資料前處理

本研究將資料前處理分爲二個部份；(a)去偏極值，因同一資料集中，可能會出現和平均水準落差甚大的值，如果極偏值是因爲某些人爲上的疏失或是無特殊性意義的情況所造成的，若不去除，則可能造成資料展現時的偏差，故在此我們以公式 3-7 將資料標準化後(我們採用 Z 分配)，檢視標準化大於 3 或小於-3 的值已確認無特殊理由所造成後則予以刪去。(b)資料正規化，其目的在於讓每個維度的值都能統一的介於[0，1]之間如圖 3.3，

$$\text{標準化公式}: Z = \frac{(X_i - \bar{X})}{S} \qquad (3\text{-}7)$$

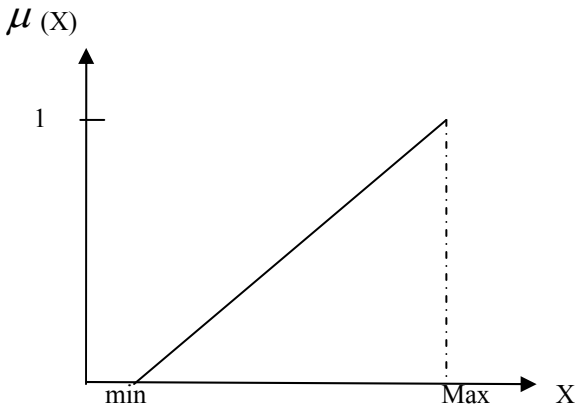


圖 3.4 極值正規化

2. 先以平均值回填於含有遺漏值的欄位。

我們將給予所有標示遺漏值的欄位一個初始值，利用公式 3-8 計算過濾遺漏值後該屬性加總平均值，作爲標示遺漏值欄位的初始估計值。平均值有三個重要特性[1]:

1. 簡化作用:平均值能簡化一群體的所有數值爲一數值。
2. 代表作用:平均值能代表一群體的平均水準。
3. 比較作用:平均值簡化所有數值爲一數值後，以該數值代表群體的平均水準，而便於兩個或兩個以上群體間作比較。

$$\bar{X} = \frac{1}{n}\sum_{k=1}^{n} X_k \qquad (3\text{-}8)$$

$\bar{X}$ :平均值　　n:資料數　　$X_k$ :第 k 筆資料

3. 分別利用 K-MEANS、PSO、KPSO、反彈機制(以下簡稱 R)+PSO、RKPSO 進行分群，直至分群穩定後再執行步驟 4。
4. 檢視其輸出值，過濾原先含有遺漏值的資料，計算該遺漏值所屬群集平均值，回填至遺漏值欄，再重新執行步驟 3 直至回填之估計值穩定則執行步驟 5。
5. 以 MAE [1]作爲實驗評估標準，由 MAE 值之大小，可瞭解預測值與實際

值之離散程度。其值越小代表模式之離散程度越小，其效果亦較佳。公式 3-9 爲 MAE 計算公式，計算估計值($\hat{e}_i$)減掉真實值($O_i$)後絕對值之平均，其 N 爲估計數量，

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left|\hat{e}_i - O_i\right| \qquad (3\text{-}9)$$

(二) Iteration-based 估計模式:

不同於 Run-based 估計模式以分群穩定後再回填的方式，Iteration-based 估計模式是將回值步驟置於分群中的每一次迭代，其流程如圖 3.5 而詳細說明如下:

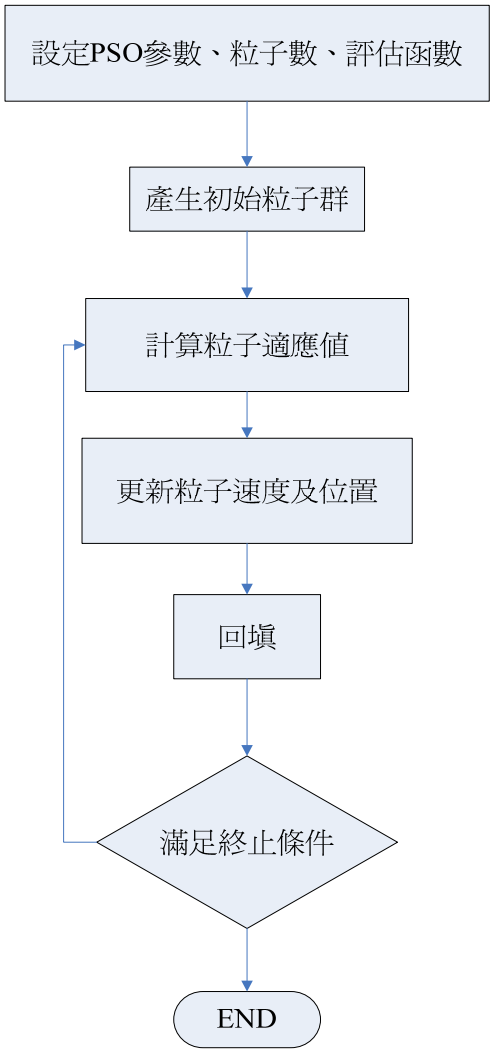


圖 3.5 Iteration-based 估計模式流程圖

1 資料前處理(同 Run-based 估計模式之步驟 1)。

2 先以平均值回填於含有遺漏值的欄位(同 Run-based 估計模式之步驟 2)。

3 分別利用 K-MEANS、PSO、KPSO、反彈機制(以下簡稱 R)+PSO、RKPSO 進行分群，其目的在於實驗 RKPSO 是否能如預期中，以加入 K-means 加速收斂及反彈機制規避落出尋解空間外部這二項特色後能否有較佳的表現。因此我們在實驗部分將記錄 K-MEANS、PSO、KPSO、RPSO、RKPSO 的估計結果並加以分析。

4 檢視每一次迭代後之輸出值，過濾原先含有遺漏值的資料，計算該遺漏值所屬群集平均值，回填至遺漏值欄，再重新執行步驟 3 直至迭代次數結束，本研究預期其回填值及目標函數會在每一次的迭代後收斂。

5 以 MAE [1]作爲實驗評估標準。

## 第三節 RKPSO 分群應用於遺漏值問題之推估範例

本研究延續文獻[1]在估計遺漏值時，所使用的物以類聚想法，針對分群結果會直接影響估計結果的概念下，對分群技術作更進一步的改良，而不同於文獻[1]中以分群穩定後回填再分群再回填的方式，在此我們將回填步驟置於每一次迭代後，讓回填值跟隨著每一次迭代作更改。其作改良動機在於文獻[1]的方法是等分群結果穩定後做回填，直至回填結果穩定才結束迴圈，換句話說若分群內部需執行 100 次迭代，而回填需滿 5 次才穩定則必需執行 500 次的迭代才能得到估計結果。然而，這樣的方式若應用在大型的資料庫上實作是非常費時的。因此我們將回填步驟置於每一次迭代後，讓回填值跟隨著每一次迭代作更改，意即回填值會跟著內部分群的收斂同樣達到收斂的效果。其改良的目的是希望能大幅減短遺漏值估計的速度。本研究選取 PSO 分群爲技術基礎進行實驗研究，其目的在於利用分群結果上有優異表現的 PSO 分群演算法[11]，加強各群群內的內聚力，達到以群內平均填入遺漏值後與其真實值誤差最小

在此我們將以文獻[1]內的半導體銅製程實驗個案來說明本研究之推估流程，及比較文獻[1]所提出的方法Neural Network-based Estimate Missing value Solution(NBEMS)與本研究方法二者之實驗結果。如表3.1此個案資料一共有18筆資料，並隨機假設5%的資料遺漏情況，每筆資料有三個屬性變數，分別是研磨速、均勻度及選擇性，加框部份爲我們隨機假設的遺漏值。

表 3.1 半導體銅製程資料

| 第 n 筆資料 | 研磨速(RR) | 均勻度(NU；%) | 選擇性 (Tan/Cu) |
|---|---|---|---|
| 1 | 294 | 14.3 | 4 |
| 2 | 289 | 15.7 | 4.3 |
| 3 | 314 | 23.2 | 5.6 |
| 4 | 375 | 12.1 | 3.7 |
| 5 | 437 | 8.7 | 4.9 |
| 6 | 498 | 6.5 | 6.1 |
| 7 | 481 | 8.99 | 4.2 |
| 8 | 588 | 11.8 | 4.3 |
| 9 | 660 | 12.4 | 5.3 |
| 10 | 242 | 16.2 | 4.6 |
| 11 | 268 | 26.9 | 4.1 |
| 12 | 340 | 10.5 | 5.3 |
| 13 | 377 | 16.9 | 3.9 |

| | | | |
|---|---|---|---|
| 14 | 434 | 5.06 | 4.7 |
| 15 | 494 | 7.08 | 5.4 |
| 16 | 483 | 8.76 | 5.2 |
| 17 | 580 | 15.1 | 4.6 |
| 18 | 651 | 5 | 5.8 |

1. 資料前處理

如表 3.2 將資料庫資料中所有屬性值調整爲介於[0,1]之間的隸屬函數及標示遺漏值欄位。

表 3.2 正規化半導體銅製程資料

| 第 n 筆資料 | 研磨速(RR) | 均勻度(NU；%) | 選擇性(Tan/Cu) |
|---|---|---|---|
| 1 | 0.1244 | 0.4247 | 0.125 |
| 2 | 0.1124 | 0.4886 | 0.25 |
| 3 | 0.1722 | 0.8311 | 0.7917 |
| 4 | 0.3182 | 0.3242 | 0 |
| 5 | 0.4665 | missing | 0.5 |
| 6 | 0.6124 | 0.0685 | 1 |
| 7 | 0.5718 | 0.1822 | 0.2083 |

| | | | |
|---|---|---|---|
| 8 | 0.8278 | 0.3105 | 0.25 |
| 9 | 1 | 0.3379 | 0.6667 |
| 10 | 0 | 0.5114 | 0.375 |
| 11 | 0.0622 | 1 | missing |
| 12 | 0.2344 | 0.2511 | 0.6667 |
| 13 | 0.323 | 0.5434 | 0.0833 |
| 14 | 0.4593 | 0.0027 | 0.4167 |
| 15 | 0.6029 | 0.095 | 0.7083 |
| 16 | missing | 0.1717 | 0.625 |
| 17 | 0.8086 | 0.4612 | 0.375 |
| 18 | 0.9785 | 0 | 0.875 |

2. 將遺漏值欄位填入初始值:

計算去遺漏值後各屬性之平均，再回填該屬性內所有遺漏值欄位作爲漏值進行分群前的初始值，表 3.3 爲各屬性去遺漏值後之平均。

表3.3半導體銅製程資料各屬性去遺漏值後之平均

| 研磨速(RR) | 均勻度(NU；%) | 選擇性(Tan/Cu) |
|---|---|---|
| 430.7059 | 12.7347 | 4.817 |

3. 以反彈機制+KPSO 進行分群:

為求實驗公平性，本研究將分群數同文獻[1]設為三群。

表 3.4 反彈機制+KPSO 分群結果

| 第 n 群 | 群內資料 |
|---|---|
| 1 | 1、2、3、4、10、11、13 |
| 2 | 7、8、9、17 |
| 3 | 5、6、12、14、15、16、18 |

如表 3.4 利用反彈機制+KPSO 進行分群，再將各群內扣除遺漏值後，各屬性平均回填該群遺漏欄位，表 3.5 為遺漏值回填最後結果。.

表 3.5 遺漏值回填結果

| RR | NU | Tan/Cu |
|---|---|---|
| 475.6667 | 7.15 | 4.35 |

4. 與文獻[1]實驗結果比較

表 3.6 是平均值、K-means、NBEMS[1]、RKPSO 回填結果，有框線的數字為最接近真實值之標示。而表 3.7 則為平均值、K-means、NBEMS、RKPSO 回填結果之誤差比較，有框線的數字為估計最佳之標示。

表 3.6 平均值、K-means、NBEMS、RKPSO 回填結果

| | RR | NU | Tan/Cu |
|---|---|---|---|
| Real data | 483.0000 | 8.7000 | 4.1000 |
| 平均值 | 430.7059 | 12.7347 | 4.8176 |
| K-means | 429.5000 | 9.4863 | 4.6250 |
| NBEMS | 450.6667 | 7.6033 | 4.6250 |
| RKPSO | 475.6667 | 7.15 | 4.35 |

表 3.7 與其它方法推估之誤差比較

| | RR | NU | Tan/Cu | MAE |
|---|---|---|---|---|
| 平均值 | 52.2941 | 4.0347 | 0.7176 | 19.0155 |
| K-means | 53.5000 | 0.7863 | 0.5250 | 18.2704 |
| NBEMS | 32.3333 | 1.0967 | 0.3857 | 11.2719 |
| RKPSO | 7.3333 | 1.55 | 0.25 | 3.0444 |

# 第四章 實驗結果與分析

## 第一節 實驗資料與參數設計

本節將介紹實驗測試資料及相關參數的設定。本實驗以人工資料和實際資料二種資料庫進行實驗，經由資料前處理後，以人工方式隨機產生不同百分比的遺漏值存有率，用來測試在不同遺漏值比例下估計遺漏值的結果。實驗分別以K-means、PSO、KPSO、反彈機制+PSO(以下簡稱RPSO)及反彈機制+KPSO(以下簡稱RKPSO)進行實驗，其目的在於驗證RKPSO分群法，是否在不同的條件下，對遺漏值的估計效果都能優於其他四種分群方法。同時我們也以實驗來探討Run-based和Iteration-based兩種方法的估計結果。

從表4.1可清楚得知本研究的實驗設定，本研究設迭代次數以10N次爲基準，其N爲資料庫的維度乘上群數。實驗總次數爲20次，根據[19]研究顯示進行總和20次的分群，所得的結果再加以平均，是最客觀的數據。而評估分群之績效指標，是採用MAE計算回填後，遺漏值與真實值的平均誤差。

表4.1 實驗設定

| | 實驗設定 |
|---|---|
| 粒子數 | 5N |
| 迭代次數 | 10N |
| 實驗次數 | 20 |
| c1、c2 | 2 |
| 評估指標 | MAE |

## 一. 人工資料

本研究以公式4-1之均勻分配方式，隨機方式產生二維600筆資料的測試資料庫，其資料分佈如圖4.1，再以人工方式分別隨機產生5%、10%、15%、20%的遺漏值存有率，分別利用K-means、PSO、KPSO、RPSO、RKPSO分群技術將人工資料庫分爲四群後，進行遺漏值推估程序。

$$\mu = \begin{bmatrix} m_i \\ 0 \end{bmatrix}, \sum = \begin{bmatrix} 0.50 & 0.05 \\ 0.05 & 0.50 \end{bmatrix} \qquad (4\text{-}1)$$

$$i = 1,...,4 \qquad m_1 = -3, \qquad m_2 = 0, \qquad m_3 = 3, \quad and \quad m_4 = 6,$$

$\mu$=平均向量，$\sum$ =共變矩陣

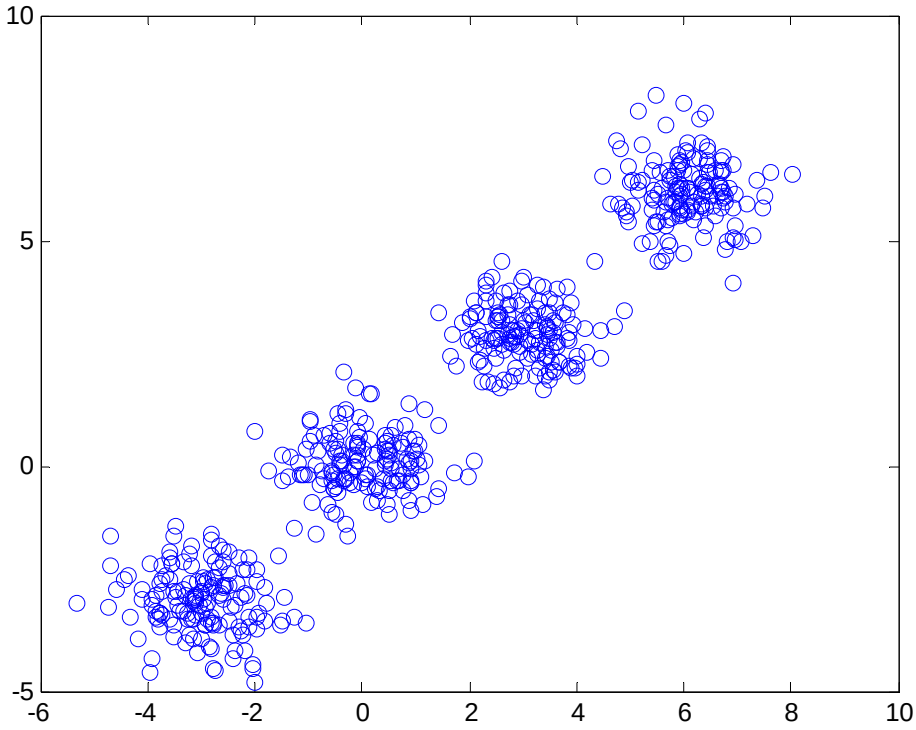


圖4.1 人工資料分佈情況

## 二.實驗資料庫

本研究選取四組由美國加州Irvine 大學資訊與電腦科學系所提供的實際資料庫作爲測試資料[26]，資料庫名稱分別是天然石油(Crude Oil)、蝴蝶花(Iris plants)、玻璃(Glass) 與母音(Vowel)。測試遺漏值存有率爲5%、10%、15%及20%的資料庫情況下遺漏值的估計正確率。表4.2爲上述四組實際資料庫的相關資訊。

表4.2 實際資料庫相關資料表

| | Oil | Iris | Glass | Vowel |
|---|---|---|---|---|
| 資料數目 | 56 | 150 | 214 | 871 |
| 資料維度 | 5 | 4 | 9 | 3 |
| 分群群數 | 3 | 3 | 7 | 6 |

1. 天然石油(Crude Oil)

天然石油資料庫共有56筆資料，可依其資料屬性分爲3群，其中每筆資料皆有5種屬性，分別爲釩(Vanadium)、鐵(Iron)、鈹(Beryllium)、飽和碳氫化合物(Saturated Hydrocarbons)與芳香族碳氫化合物(Aromatic Hydrocarbons)。由於無法繪出3度空間以上的資料分佈情況因此在此大略示意oil資料庫之資料分佈情況，其圖4.2則爲oil資料庫前三維資料分佈圖，而詳細資料如表4.3與表4.4所示。

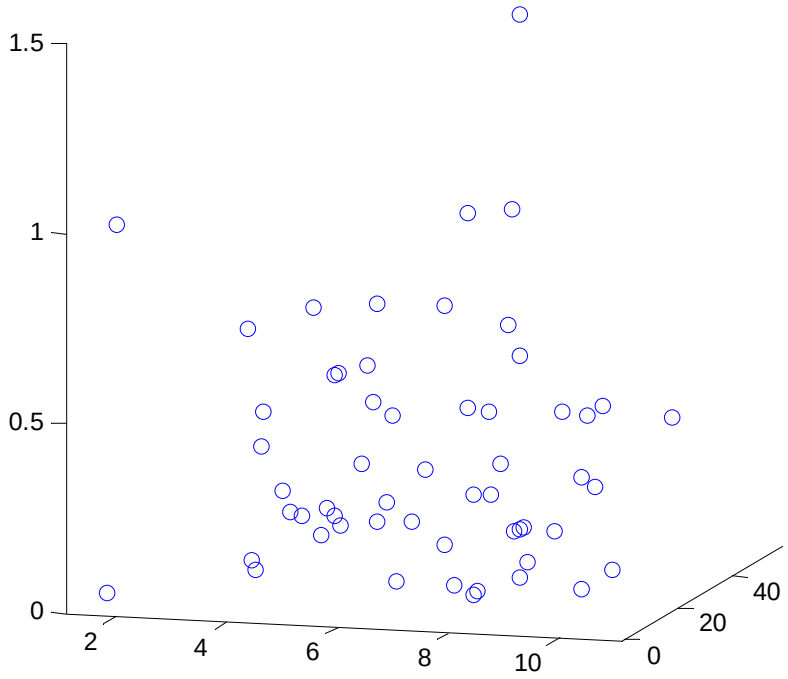


圖4.2 oil資料庫前三維資料分佈圖

表4.3 Crude Oil資料庫屬性分布

| 屬性 | 最大 | 最小 | 平均 | 標準差 |
|---|---|---|---|---|
| Vanadium | 1.2 | 11 | 6.1804 | 2.4026 |
| Iron | 5.6 | 52 | 27.0464 | 11.5014 |
| Beryllium | 0 | 1.5 | 0.3414 | 0.3111 |
| Saturated Hydrocarbons | 3.06 | 9.25 | 5.2911 | 1.3678 |
| Aromatic Hydrocarbons | 2.22 | 13.01 | 6.4336 | 3.1263 |

表4.4 Crude Oil資料庫中各群資料所佔大小與比例

| 群體 | 所佔資料比較 | 所佔資料比例 |
|---|---|---|
| Wilhelm | 7 | 12.5% |
| Sub-Mulinia | 10 | 17.86% |
| Upper | 39 | 69.64% |

2. 鳶尾植物(Iris Plants)

鳶尾植物資料庫共有150筆資料，由Iris Setosa、Versicolour與Virginica3種鳶尾花所組成，其每筆資料皆有萼片(Sepal) 與花瓣(Petal)的長度(Length)、寬度(Width)4種屬性。詳細資料如表4.5與表4.6所示。

表4.5 Iris Plants資料庫屬性分布

| 屬性 | 最大 | 最小 | 平均 | 標準差 |
|---|---|---|---|---|
| Sepal Length | 4.3 | 7.9 | 5.84 | 0.8253 |
| Sepal Width | 2.0 | 4.4 | 3.05 | 0.4321 |
| Petal Length | 1.0 | 6.9 | 3.76 | 1.7585 |
| Petal Width | 0.1 | 2.5 | 1.02 | 0.7606 |

表4.6 Iris Plants資料庫中各群資料所佔大小與比例

| 群體 | 所佔資料比較 | 所佔資料比例 |
|---|---|---|
| Iris Setosa | 50 | 33.33% |
| Versicolour | 50 | 33.33% |
| Virginica | 50 | 33.33% |

3. 玻璃(Glass)

Glass資料庫共有214筆資料，由7種不同的玻璃組成，每筆資料輻包含的9種化學元素。詳細資料如表4.7與表4.8所示。

表4.7 Glass資料庫屬性分布

| 屬性 | 最小 | 最大 | 平均值 | 標準差 |
|---|---|---|---|---|
| RI | 1.51 | 1.5339 | 1.51 | 0.0030 |
| Na | 10.73 | 17.38 | 13.40 | 0.8147 |
| Mg | 0 | 4.49 | 2.68 | 1.4390 |
| A1 | 0.29 | 3.5 | 1.44 | 0.4981 |

| | | | | |
|---|---|---|---|---|
| Si | 69.81 | 75.41 | 72.65 | 0.7727 |
| K | 0 | 6.21 | 0.49 | 0.6507 |
| Ca | 5.43 | 16.19 | 8.95 | 1.4198 |
| Ba | 0 | 3.51 | 0.175 | 0.4961 |
| Fe | 0 | 0.51 | 0.57 | 0.0972 |

表4.8 Glass資料庫中各群資料所佔大小與比例

| 群體 | 所佔資料比較 | 所佔資料比例 |
|---|---|---|
| Float Building Windows | 70 | 32.71% |
| Float Vehicle Windows | 17 | 7.94% |
| Non-Float Building Windows | 76 | 35.5% |
| Non-Float Vehicle Windows | 0 | 0% |
| Containers | 13 | 6.07% |
| Tableware | 9 | 4.20% |
| Headlamps | 29 | 13.55% |

4. 母音(Vowel)

母音資料庫是由3種音頻屬性的印地安語言之母音，共有871筆資料，將分類爲{ $\delta$, a, i, u, e, o}六種母音。圖4.3爲Vowel資料分佈圖，而詳細資料如表4.9與表4.10所示。

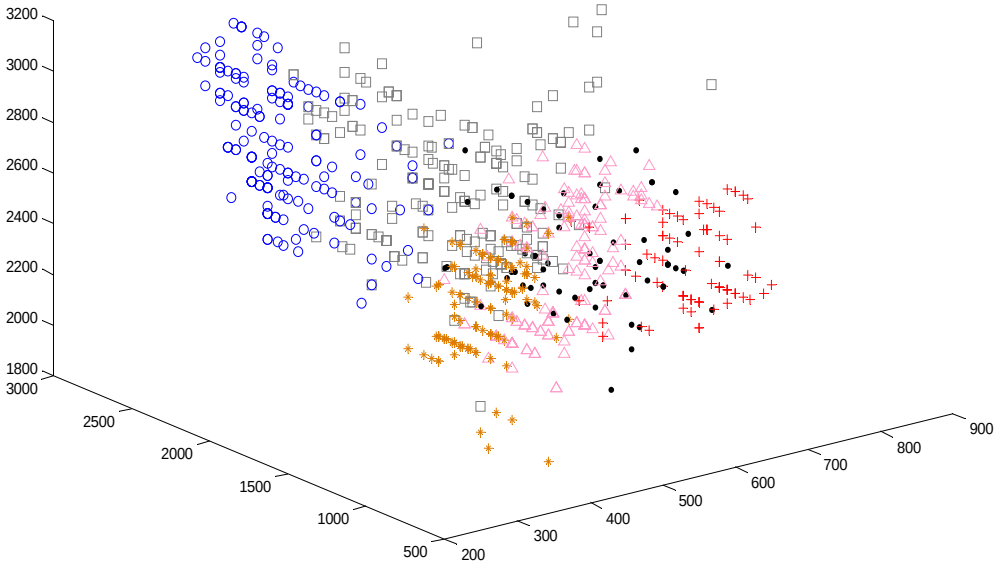


圖4.3 Vowel資料分佈圖

表4.9 Vowel資料庫屬性分布

| 屬性 | 最大 | 最小 | 平均 | 標準差 |
|---|---|---|---|---|
| F1 | 250 | 900 | 470.4822 | 129.148 |
| F2 | 700 | 2550 | 1514.6842 | 507.2898 |
| F3 | 1800 | 3200 | 2561.0218 | 244.4014 |

表4.10 Vowel資料庫中各群資料所佔大小與比例

| 群體 | 所佔資料比較 | 所佔資料比例 |
|---|---|---|
| $\delta$ | 72 | 8.27% |

| | | |
|---|---|---|
| a | 89 | 10.22% |
| i | 172 | 19.57% |
| u | 151 | 17.34% |
| e | 207 | 23.77% |
| o | 180 | 20.67% |

二.實驗結果

(1).收斂情況:如圖 4.4-圖 4.11 分別爲 Iteration-based 和 Run-based 在人工資料庫及實際資料庫之 MAE 與分群適應值之收斂圖。

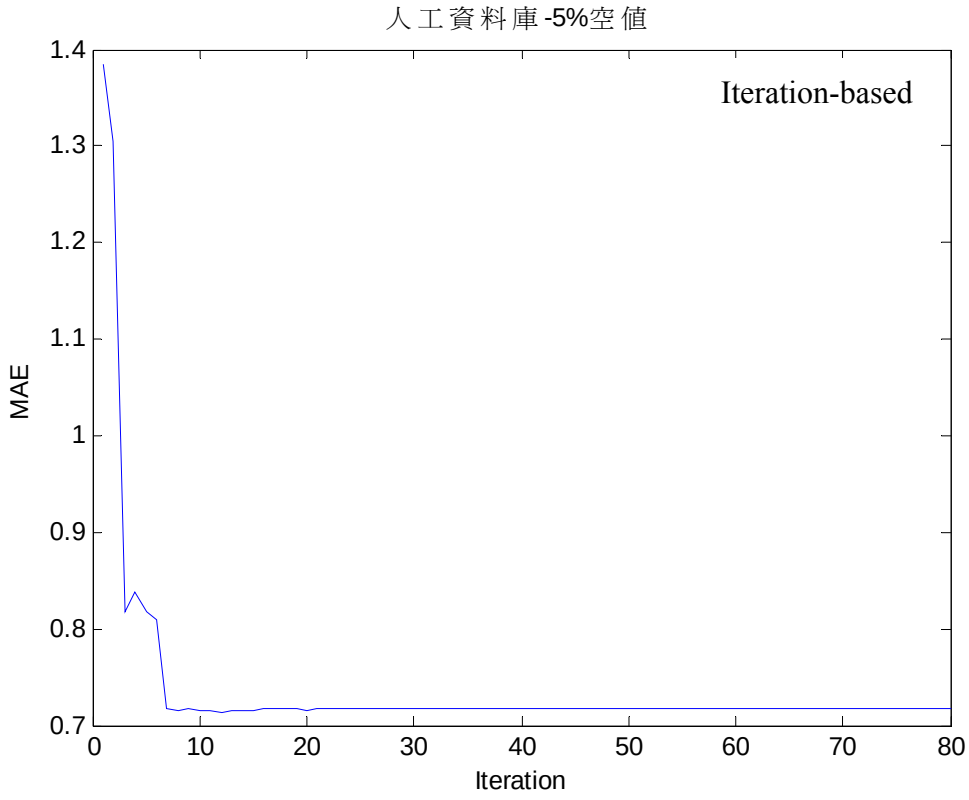


圖 4.4 人工資料庫於 Iteration-based-RKPSO 分群之 MAE 收斂圖

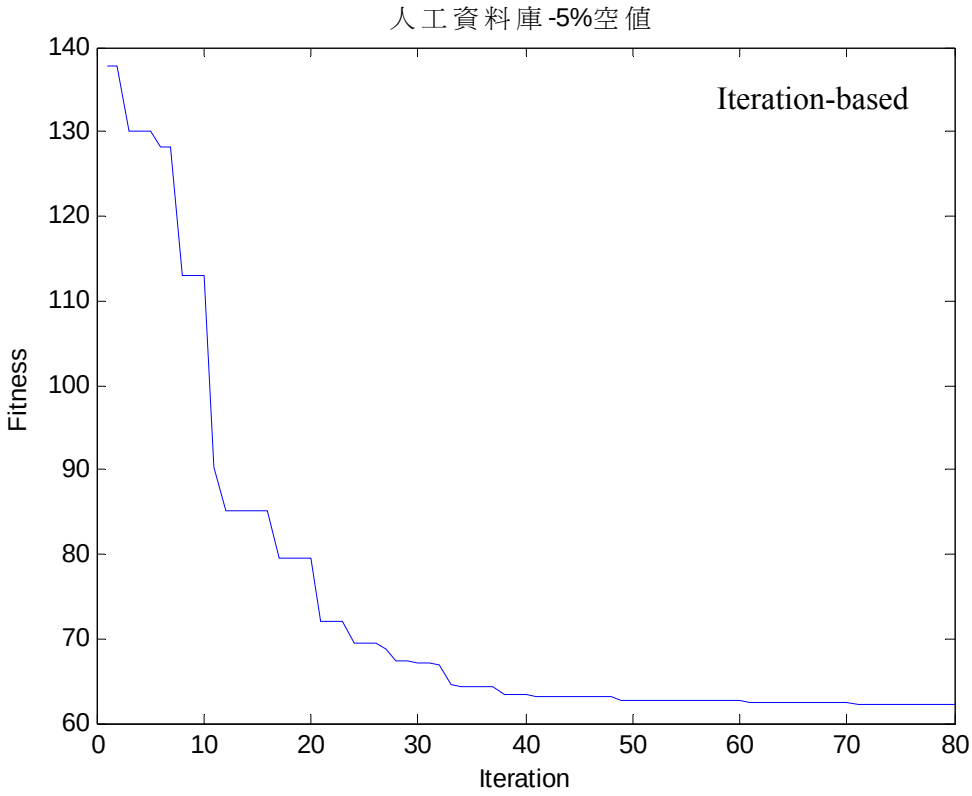


圖 4.5 人工資料庫於 Iteration-based-RKPSO 分群之適應值收斂圖

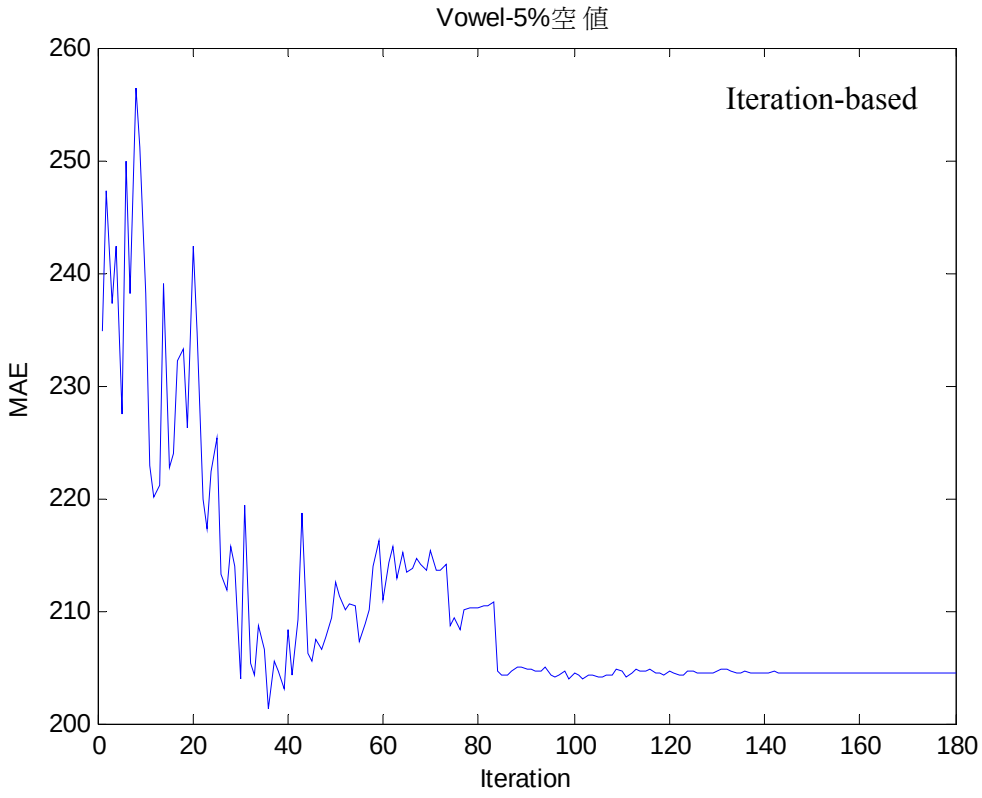


圖 4.6Vowel 於 Iteration-based-RKPSO 分群之 MAE 收斂圖

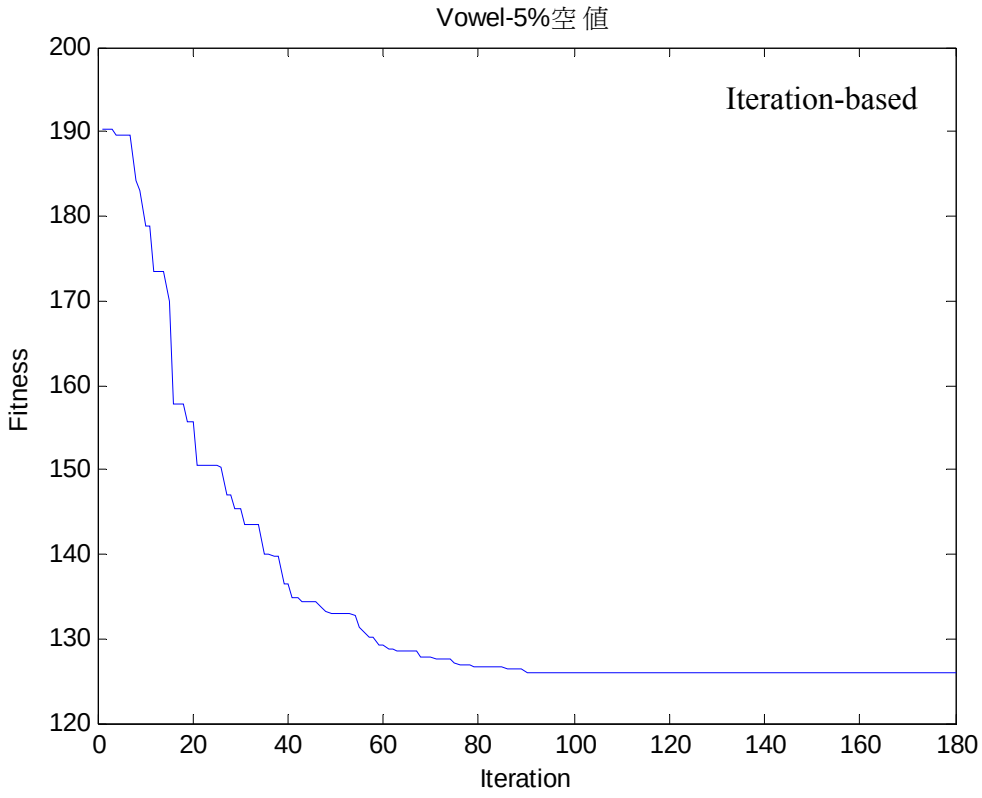


圖 4.7Vowel 於 Iteration-based-RKPSO 分群之適應值收斂圖

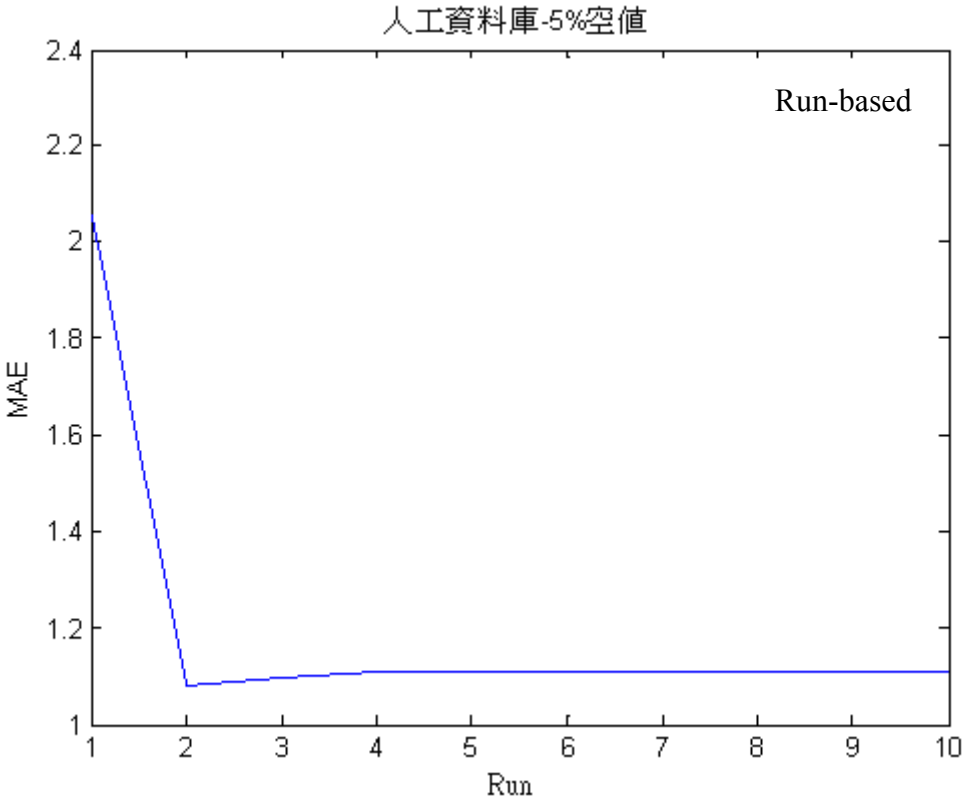


圖 4.8 人工資料庫於 Run-based-RKPSO 分群之 MAE 收斂圖

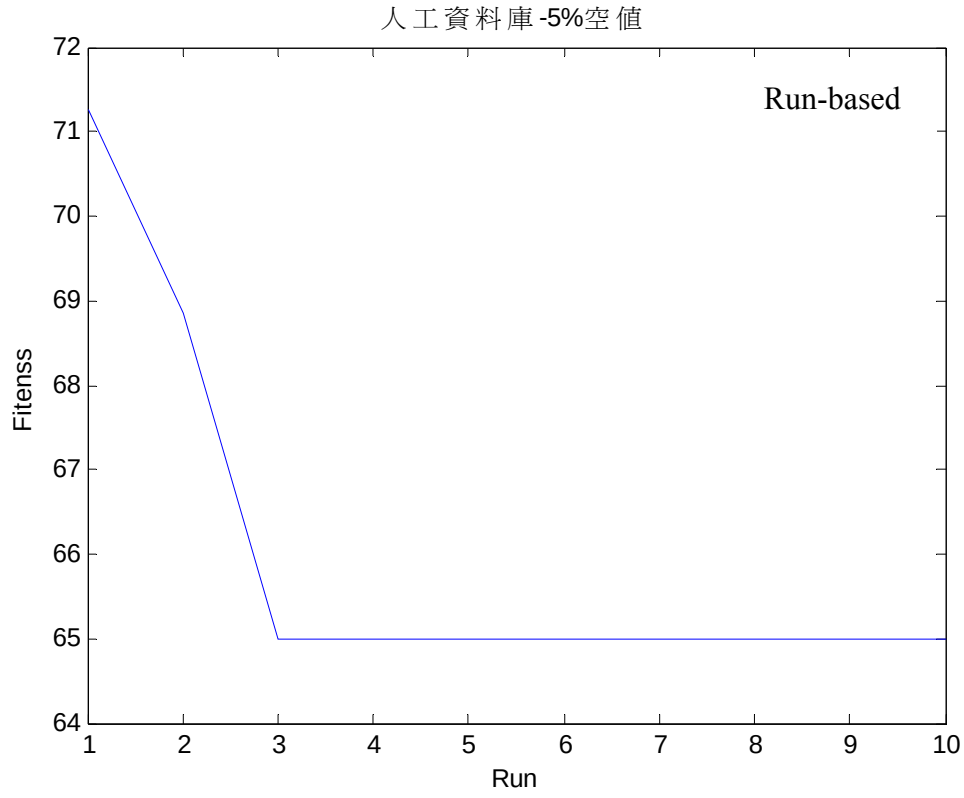


圖 4.9 人工資料庫於 Run-based-RKPSO 分群之適應值收斂圖

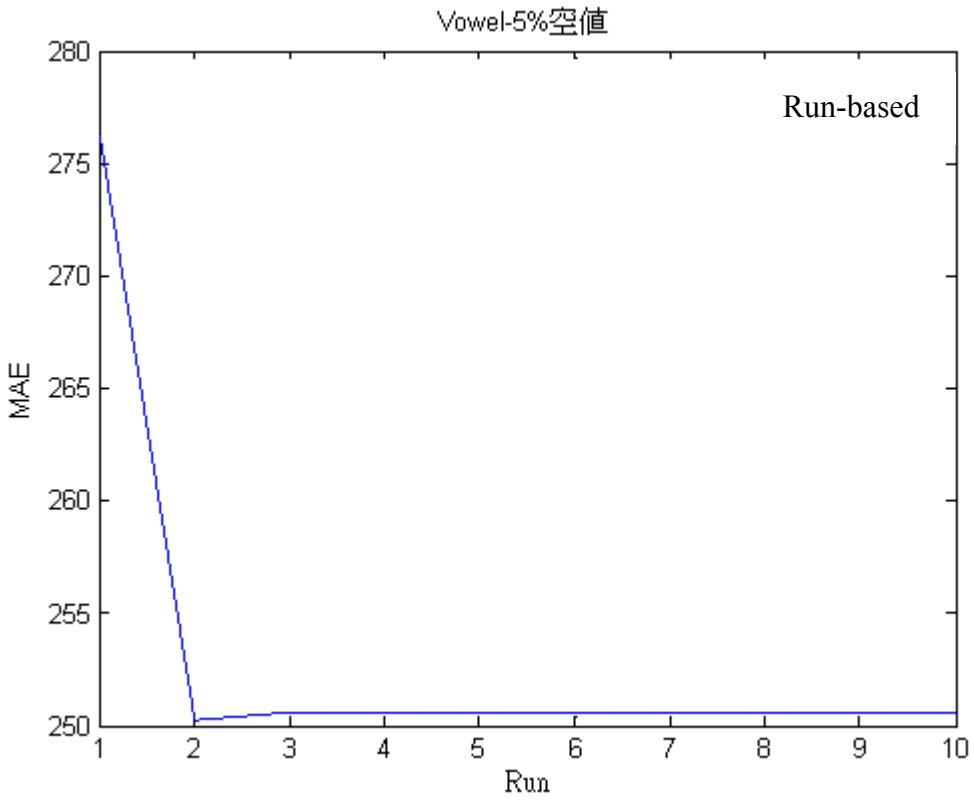


圖 4.10 Vowel 於 Run-based-RKPSO 分群之 MAE 收斂圖

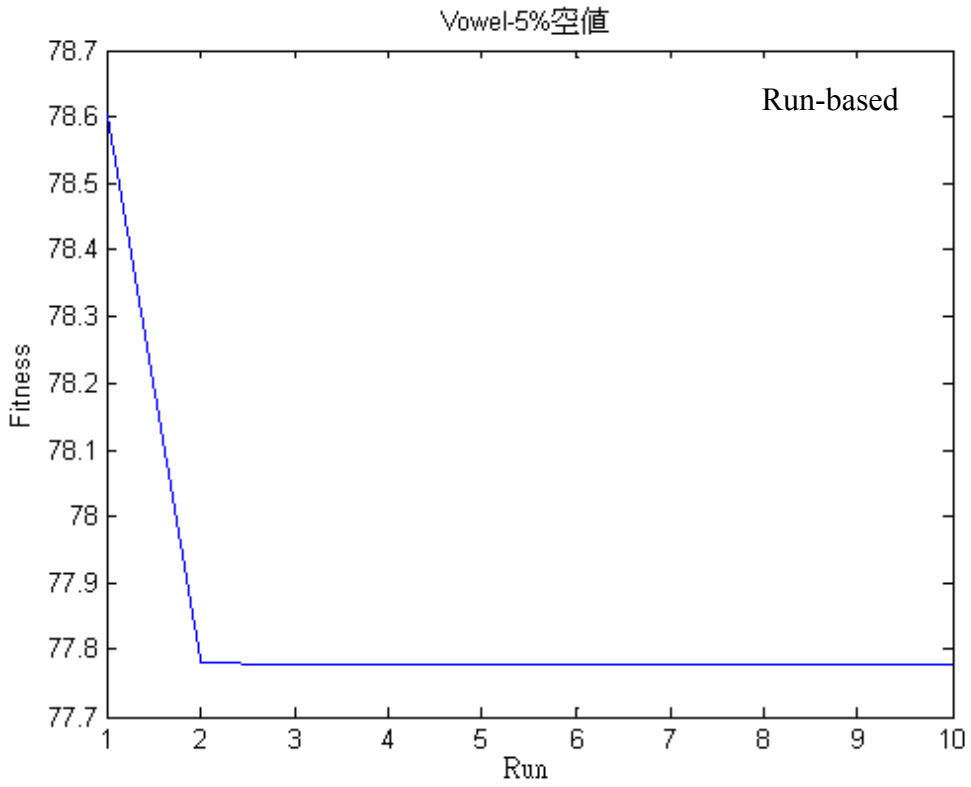


圖 4.11 Vowel 於 **Run-based**-RKPSO 分群之適應值收斂圖

(2)Run-based (R)VS Iteration-based(I)實驗結果:

在表 4.11 至表 4.14 以框線來突顯有較佳估計的值，因此我們可從表 4.11-表 4.14Run-based 和 Iteration-based 在不同方法及不同遺漏值存有率下的估計情況，在表 4.11 和表 4.12 別爲是 **Oil** 及 **Iris** 之估計記錄，我們可以看出在這二個小型資料庫的估計情況是以 Run-based 結果較佳，但隨著遺漏值比率愈高則以 Iteration-based 之估計表現較爲出色，另外表 4.13 和表 4.14 則屬較爲複雜的資料庫，以表中的記錄來看則皆以 Iteration-based 有明顯的最佳表現。

表 4.11 Run-based 和 Iteration-based 在 Oil 資料庫之估計結果比較

| Oil 資料庫 | | | | | | |
|---|---|---|---|---|---|---|
| 遺漏值比率 | Model | K-means | PSO | KPSO | RPSO | RKPSO |
| 5% | R | 0.7191 | 0.668 | 0.5007 | 0.5594 | 0.5002 |
| | I | 0.8415 | 0.7973 | 0.7444 | 0.7708 | 0.7408 |
| B10% | R | 0.7757 | 0.7553 | 0.7468 | 0.7913 | 0.7876 |
| | I | 0.7912 | 0.8178 | 0.8339 | 0.8262 | 0.7911 |
| 15% | R | 0.7465 | 0.7221 | 0.7132 | 0.6832 | 0.7699 |
| | I | 0.8041 | 0.7901 | 0.7653 | 0.8051 | 0.7644 |
| 20% | R | 1.5338 | 1.2145 | 1.5023 | 1.2753 | 1.133 |
| | I | 1.4316 | 1.506 | 1.4124 | 1.384 | 1.3275 |

表 4.12Run-based 和 Iteration-based 在 Iris 資料庫之估計結果比較

| Iris 資料庫 | | | | | | |
|---|---|---|---|---|---|---|
| 遺漏值比率 | Model | K-means | PSO | KPSO | RPSO | RKPSO |
| 5% | R | 0.2809 | 0.2842 | 0.2893 | 0.2785 | 0.2793 |
| | I | 0.2931 | 0.2958 | 0.2975 | 0.3116 | 0.2885 |
| 10% | R | 0.2475 | 0.2362 | 0.2483 | 0.2438 | 0.2188 |
| | I | 0.2533 | 0.2526 | 0.2432 | 0.2441 | 0.2403 |
| 15% | R | 0.3449 | 0.3414 | 0.3398 | 0.3398 | 0.3289 |
| | I | 0.3317 | 0.322 | 0.319 | 0.2983 | 0.2971 |
| 20% | R | 0.3396 | 0.3207 | 0.3229 | 0.3373 | 0.3327 |
| | I | 0.317 | 0.3158 | 0.3169 | 0.296 | 0.2895 |

表 4.13un-based 和 Iteration-based 在 Glass 資料庫之估計結果比較

| Glass 資料庫 | | | | | | |
|---|---|---|---|---|---|---|
| 遺漏值比率 | Model | K-means | PSO | KPSO | RPSO | RKPSO |
| 5% | R | 15.1181 | 15.1994 | 15.1988 | 15.1704 | 15.1675 |
| | I | 0.5163 | 0.5721 | 0.5693 | 0.5696 | 0.4823 |
| 10% | R | 24.2008 | 24.2289 | 24.2297 | 24.1921 | 24.1892 |
| | I | 0.576 | 0.592 | 0.598 | 0.5686 | 0.5609 |
| 15% | R | 21.5387 | 21.5273 | 21.5326 | 21.5209 | 21.5197 |
| | I | 0.5814 | 0.4375 | 0.419 | 0.4188 | 0.4446 |
| 20% | R | 25.1908 | 25.1953 | 25.1954 | 25.1722 | 25.1607 |
| | I | 0.4422 | 0.4692 | 0.4532 | 0.4531 | 0.4393 |

表 4.14n-based 和 Iteration-based 在 Vowel 資料庫之估計結果比較

| Vowel 資料庫 | | | | | | |
|---|---|---|---|---|---|---|
| 遺漏值比率 | Model | K-means | PSO | KPSO | RPSO | RKPSO |
| 5% | R | 301.6598 | 291.9252 | 384.1572 | 327.6887 | 232.3247 |
| | I | 280.9034 | 208.8453 | 210.3507 | 208.7795 | 185.3032 |
| 10% | R | 308.1055 | 247.4305 | 273.349 | 258.6487 | 236.7539 |
| | I | 240.4019 | 215.6348 | 201.172 | 203.7089 | 177.501 |
| 15% | R | 235.1579 | 231.028 | 231.4541 | 288.9942 | 260.0754 |
| | I | 230.465 | 196.2055 | 201.1720 | 182.6086 | 179.7402 |
| 20% | R | 222.9011 | 259.054 | 235.7921 | 244.2691 | 231.4541 |
| | I | 231.1755 | 187.6708 | 187.2437 | 188.0734 | 185.0526 |

(3) )Run-based (R)VS Iteration-based(I)程式執行時間:

爲證實 Iteration-based 執行速度較 Run-based 來的快，我們將上面實驗時間記錄後加以比較，實驗設備爲 Microsoft windows XP Professional Version 2002 Service Pack 2、Genuine Intel (R)CPU、T2250@1.73GHz、1.71GHz2.200GB 的 RAM。在此我們以 RKPSO 之 Run-based 和 Iteration-based 模式分別執行 20%空值存有率下四個資料庫之空值估計，表 4.15 則爲執行 Run-based 和 Iteration-based 一次實驗的時間記錄，我們可從表中明顯看出 Iteration-based 較 Run-based 執行速度快上許多。

表 4.15 Run-based (R)VS Iteration-based(I)程式執行時間

| 單位(秒) | Oil | Iris | Glass | Vowel |
|---|---|---|---|---|
| Run-based | 52.7357 | 91.1195 | 5712.2268 | 2233.1166 |
| Iteration-based | 10.9906 | 18.7254 | 1948.0622 | 466.4176 |

(4) Iteration-based 實驗結果:

本小節將呈現 K-means、PSO、KPSO、R2-PSO 與 R2-KPSO 五種分群方法運用於不同型態資料庫中估計遺漏值的結果在此我們皆以 Iteration-based 估計模式進行實驗。最後利用 MAE 評估方式作爲結果優劣的比較。

1.人工資料庫:

表 4.16 -4.19 分別爲人工資料庫在遺漏值存有率爲 5%、10%、15%及 20%情況下，經由 20 次實驗後，各種分群技術推估遺漏值之 20 次實驗中最好、20 次總實驗平均及 20 次實驗中最差的 MAE 及距離總值紀錄表，表 4.20 則是五種分群技術在上述各種遺漏值存有率情況下進行 20 次實驗後之平均 MAE 比較表，下列分析表中我們以★符號表示該值在該項目中有較好的表現。

在表 4.16 中雖然 RKPSO 估計值標準差較 K-means 大，但以其平均 MAE 而

言 RKPSO 都較其它方法來的好，因此我們可以說 RKPSO 在此人工資料庫且 5% 遺漏值存有率下有較佳的估計表現。

表 4.16 遺漏值存有率 5%之人工資料庫推估記錄

| 人工 5% | K-means | PSO | KPSO | RPSO | RKPSO |
|---|---|---|---|---|---|
| 最好 MAE | 2.0059 | 0.7677★ | 0.8244 | 0.7894 | 0.8203 |
| 平均 MAE | 2.4564 | 0.9586 | 0.9915 | 0.9408 | 0.9234★ |
| MAE 標準差 | 0.1025★ | 0.1592 | 0.1836 | 0.1603 | 0.1651 |
| | | | | | |
| 最好距離 | 71.3341 | 109.3064 | 109.2837 | 109.1963 | 109.1579 |
| 平均距離 | 73.6637 | 110.2291 | 109.942 | 109.6534 | 109.8226 |
| 距離標準差 | 10.1547 | 0.9271 | 1.3354 | 0.6217 | 1.5129 |

表 4.17 中則以 PSO 在最好 MAE 及 MAE 標準差上有較好的表現，而以表現較佳且接近的幾種方法如 PSO、KPSO、KPSO、RPSO 及 RKPSO 來說又以 RKPSO 之估計值最爲穩定。

表 4.17 遺漏值存有率 10%之人工資料庫推估記錄

| 人工 10% | K-means | PSO | KPSO | 、RPSO | RKPSO |
|---|---|---|---|---|---|
| 最好 MAE | 1.7619 | 1.0091★ | 1.0652 | 1.0264 | 1.0091★ |
| 平均 MAE | 1.7702 | 1.1319 | 1.1611 | 1.0965 | 1.0688★ |
| MAE 標準差 | 0.0361★ | 0.0894 | 0.0824 | 0.0617 | 0.0531 |
| | | | | | |
| 最好距離 | 76.5614 | 107.1959 | 107.5757 | 107.2423 | 107.1382 |
| 平均距離 | 79.0037 | 108.3644 | 108.3874 | 107.8472 | 107.5043 |
| 距離標準差 | 10.6460 | 0.7143 | 0.7674 | 0.5326 | 0.3237 |

從表 4.18 可得知，以 PSO 及改良 PSO 在人工資料庫 15%遺漏值存有率下都較 K-means 明顯降低 MAE，而其中又以 RKPSO 之 MAE 來的最好且最爲穩定。

表 4.18 遺漏值存有率 15%之人工資料庫推估記錄

| 人工 15% | K-means | PSO | KPSO | RPSO | RKPSO |
|---|---|---|---|---|---|
| 最好 MAE | 1.6898 | 1.1448 | 1.1073 | 1.0756★ | 1.0782 |
| 平均 MAE | 1.7144 | 1.2051 | 1.1733 | 1.1586 | 1.124★ |
| MAE 標準差 | 0.1071 | 0.0663 | 0.0435 | 0.0410 | 0.0259★ |
| | | | | | |
| 最好距離 | 82.5989 | 105.0699 | 106.5578 | 104.9332 | 104.9092 |
| 平均距離 | 84.8269 | 106.8277 | 105.0551 | 107.0714 | 105.7458 |
| 距離標準差 | 9.7114 | 1.7281 | 1.3621 | 2.0262 | 0.8920 |

表 4.19 中以 RPSO 有較好的 MAE，而 RKPSO 有較好的平均 MAE 其原因在於 RKPSO 之 MAE 標準差較 RPSO 相對的穩定，但以此人工資料庫 20%遺漏值存有率情況下 PSO 相關方法都表現的非常接近。

表 4.19 遺漏值存有率 20%之人工資料庫推估記錄

| 人工 20% | K-means | PSO | KPSO | RPSO | RKPSO |
|---|---|---|---|---|---|
| 最好誤差 | 1.6483 | 1.1571 | 1.1865 | 1.1522 | 1.1255★ |
| 平均誤差 | 1.7631 | 1.2485 | 1.2353 | 1.217 | 1.2084★ |
| MAE 標準差 | 0.3444 | 0.0773 | 0.0492★ | 0.0624 | 0.0608 |
| | | | | | |
| 最好距離 | 88.2039 | 102.8218 | 103.1451 | 102.7793 | 102.6554 |
| 平均距離 | 92.4466 | 104.285 | 104.5812 | 104.0969 | 103.9976 |
| 距離標準差 | 12.7282 | 1.4775 | 0.7028 | 1.0224 | 1.5097 |

表 4.20 則爲上面四種遺漏值存有率在五種分群技術下的估計結果匯總，從此表可以清楚的得知以此人工資料庫而言 RKPSO 有較好的估計能力。

表 4.20 各種遺漏值存有率情況下五種分群技術之 20 次實驗後 MAE 比較表

| 人工 | K-means | PSO | KPSO | RPSO | RKPSO | 誤差最小 |
|---|---|---|---|---|---|---|
| 5% | 2.4564 | 0.9586 | 0.9915 | 0.9408 | 0.9234 | RKPSO |
| 10% | 1.7702 | 1.1319 | 1.1611 | 1.0965 | 1.0688 | RKPSO |
| 15% | 1.7144 | 1.2051 | 1.1733 | 1.1586 | 1.124 | RKPSO |
| 20% | 1.7631 | 1.2485 | 1.2353 | 1.217 | 1.2084 | RKPSO |

2. Crude Oil 資料庫:

表 4.21 -4.24 爲五種分群方法分別在 Crude Oil 資料庫遺漏值存有率爲 5%、10%、15%及 20%情況下，紀錄 20 次實驗中最好、總實驗平均及實驗中最差的 MAE 及距離總值，而表 4.25 則是五種分群技術在上述四種遺漏值存有率情況下進行 20 次實驗後之平均 MAE 比較表。

從表 4.21 來分析，可看出以 PSO 和改良 PSO 所估計出的值都非常接近，因此哪種方法擁有較穩定的估計結果即最小的 MAE 標差，則爲在此最好的估計方法，而 RKPSO 在 MAE 標準差上有較佳的穩定度即較好的估計品質。

表 4.21 遺漏值存有率 5%之 Oil 資料庫推估記錄

| Oil 5% | K-means | PSO | KPSO | RPSO | RKPSO |
|---|---|---|---|---|---|
| 最好 MAE | 0.5514 | 0.6949 | 0.4496★ | 0.6494 | 0.6747 |
| 平均 MAE | 0.8415 | 0.7973 | 0.7444 | 0.7708 | 0.7408★ |
| MAE 標準差 | 0.3563 | 0.1155 | 0.1868 | 0.1227 | 0.1045★ |
| | | | | | |
| 最好距離 | 20.1296 | 19.749 | 19.749 | 19.7489 | 19.7489 |
| 平均距離 | 21.1113 | 19.8547 | 20.3982 | 19.8167 | 19.7938 |
| 距離標準差 | 1.5969 | 0.0834 | 1.2828 | 0.0811 | 0.0750 |

表 4.22 RKPSO 之 MAE 標準差明顯的優於其它種方法，因此我們可說 RKPSO 在 Oil 資料庫 10%遺漏值存有率下有最佳的估計品質。

表 4.22 遺漏值存有率 10%之 Oil 資料庫推估記錄

| Oil 10% | K-means | PSO | KPSO | RPSO | RKPSO |
|---|---|---|---|---|---|
| 最好 MAE | 0.5623★ | 0.6735 | 0.6735 | 0.769 | 0.769 |
| 平均 MAE | 0.7912 | 0.8178 | 0.8339 | 0.8262 | 0.7911★ |
| MAE 標準差 | 0.1161 | 0.0895 | 0.0672 | 0.0743 | 0.0386★ |
| | | | | | |
| 最好距離 | 20.1322 | 19.618 | 19.6182 | 19.6178 | 19.6178 |
| 平均距離 | 20.741 | 20.2368 | 19.8709 | 19.6668 | 19.6465 |
| 距離標準差 | 1.2075 | 1.2859 | 0.7880 | 0.0465 | 0.0357 |

從表 4.21-表 4.23 我們可以察覺到 K-means 在 2%-15%遺漏值存有率下，其最好 MAE 值都優於 PSO 及其它改良 PSO，但因其 MAE 標準差明顯過大及平均 MAE 較差而影響到整體的估計品質，故雖有最好的 MAE 表現則無法提昇平均 MAE 和估計穩定度。

表 4.23 遺漏值存有率 15%之 Oil 資料庫推估記錄

| Oil 15% | K-means | PSO | KPSO | RPSO | RKPSO |
|---|---|---|---|---|---|
| 最好 MAE | 0.5987★ | 0.6602 | 0.6602 | 0.6731 | 0.6731 |
| 平均 MAE | 0.8041 | 0.7901 | 0.7653 | 0.8051 | 0.7644★ |
| MAE 標準差 | 0.1848 | 0.0735 | 0.0770 | 0.0735 | 0.0619★ |
| | | | | | |
| 最好距離 | 20.0818 | 19.5441 | 19.5441 | 19.5441 | 19.5441 |
| 平均距離 | 21.2574 | 19.9604 | 20.1828 | 19.625 | 19.6387 |
| 距離標準差 | 1.5648 | 1.1000 | 1.2924 | 0.0781 | 0.0849 |

以表 4.24 而言，RPSO 及 RKPSO 的表現相當，因其平均 MAE 和 MAE 標準差所差距的值都不算過大因此在 Oil 資料庫 20%遺漏值存有率下，我們可以推測這二種方法的估計品質相當。

表 4.24 遺漏值存有率 20%之 Oil 資料庫推估記錄

| Oil 20% | K-means | PSO | KPSO | RPSO | RKPSO |
|---|---|---|---|---|---|
| 最好 MAE | 1.0098 | 0.909★ | 0.9554 | 0.909★ | 0.909★ |
| 平均 MAE | 1.4316 | 1.506 | 1.4124 | 1.384 | 1.3275★ |
| MAE 標準差 | 0.4127 | 0.3047 | 0.3793 | 0.2086★ | 0.2488 |
| | | | | | |
| 最好距離 | 20.0468 | 19.5051 | 19.505 | 19.5049 | 19.505 |
| 平均距離 | 20.7367 | 19.9712 | 19.8521 | 19.6168 | 19.6543 |
| 距離標準差 | 0.8073 | 1.0499 | 0.7560 | 0.1050 | 0.1192 |

表 4.25 則爲上面四種遺漏值存有率在五種分群技術下的估計結果匯總，從此表可以清楚的得知以 Oil 資料庫而言 RKPSO 有較好的估計能力。

表 4.25 各種遺漏值存有率情況下五種分群技術之 20 次實驗後 MAE 比較表

| Oil | K-means | PSO | KPSO | RPSO | RKPSO | 誤差最小 |
|---|---|---|---|---|---|---|
| 5% | 0.8415 | 0.7973 | 0.7444 | 0.7708 | 0.7408 | RKPSO |
| 10% | 0.7912 | 0.8178 | 0.8339 | 0.8262 | 0.7911 | RKPSO |
| 15% | 0.8041 | 0.7901 | 0.7653 | 0.8051 | 0.7644 | RKPSO |
| 20% | 1.4316 | 1.506 | 1.4124 | 1.384 | 1.3275 | RKPSO |

3.Iris Plants 資料庫:

表 4.26-4.29 爲五種分群方法在 Iris Plants 資料庫含有 5%、10%、15%及 20%遺漏值存有率下的估計誤差及距離總值記錄。而從表 4.29 則是五種分群技術在上述四種遺漏值存有率情況下進行 20 次實驗後之平均 MAE 比較表。

表 4.26 可看出最好 MAE 出現在 PSO 而平均 MAE 及 MAE 標準差表現面好則爲 RKPSO。然而我們察覺到 RKPSO 與 K-means 的差異非常些微，因此 RKPSO 在 Iris 資料庫遺漏值存有率 5%情況下，並沒有非常明顯出色的表現，但即便非常小的差距都有可能在它種情況下造成拉距。

表 4.26 遺漏值存有率 5%之 Iris 資料庫推估記錄

| Iris 5% | K-means | PSO | KPSO | RPSO | RKPSO |
|---|---|---|---|---|---|
| 最好 MAE | 0.2803 | 0.2655★ | 0.2773 | 0.2727 | 0.2797 |
| 平均 MAE | 0.2931 | 0.2958 | 0.2975 | 0.3116 | 0.2885★ |
| MAE 標準差 | 0.0315 | 0.0301 | 0.0296 | 0.0399 | 0.0254★ |
| | | | | | |
| 最好距離 | 29.9967 | 28.9281 | 28.9115 | 28.9109 | 28.911 |
| 平均距離 | 30.9887 | 30.9204 | 30.8633 | 29.5062 | 28.9926 |
| 距離標準差 | 2.3637 | 3.1166 | 3.2702 | 1.6501 | 0.0918 |

表 4.27 我們發現 PSO 及改良 PSO 所估計的最好 MAE 值都非常接近，而最好的平均 MAE 則爲 RKPSO。

表 4.27 遺漏值存有率 5%之 Iris 資料庫推估記錄

| Iris 10% | K-means | PSO | KPSO | RPSO | RKPSO |
|---|---|---|---|---|---|
| 最好 MAE | 0.2373 | 0.1992★ | 0.1992★ | 0.2031 | 0.1992★ |
| 平均 MAE | 0.2533 | 0.2526 | 0.2432 | 0.2441 | 0.2403★ |
| MAE 標準差 | 0.0321 | 0.0564 | 0.0266 | 0.0226★ | 0.0230 |
| | | | | | |
| 最好距離 | 30.5986 | 28.8246 | 28.8294 | 28.822 | 28.8224 |
| 平均距離 | 31.8234 | 30.3159 | 30.246 | 29.1497 | 29.1512 |
| 距離標準差 | 2.4498 | 2.5524 | 2.7234 | 1.1867 | 1.1963 |

表 4.28 中 RKPSO 在最好 MAE、平均 MAE、MAE 標準差中皆有較好的表現。

表 4.28 遺漏值存有率 5%之 Iris 資料庫推估記錄

| Iris 15% | K-means | PSO | KPSO | RPSO | RKPSO |
|---|---|---|---|---|---|
| 最好 MAE | 0.3075 | 0.264★ | 0.264★ | 0.264★ | 0.264★ |
| 平均 MAE | 0.3317 | 0.322 | 0.319 | 0.2983 | 0.2971★ |
| MAE 標準差 | 0.0371 | 0.0387 | 0.0350 | 0.0230 | 0.0137★ |
| | | | | | |
| 最好距離 | 30.5381 | 28.275 | 28.2866 | 28.2745 | 28.2744 |
| 平均距離 | 32.317 | 31.4728 | 30.5348 | 28.7184 | 28.4197 |
| 距離標準差 | 2.7175 | 3.6018 | 3.2205 | 1.1499 | 0.1522 |

表 4.29 PSO 及改良 PSO 所估計的值都非常接近，但 RKPSO 在平均 MAE 上有較好的表現，因此我們可說 RKPSOIris 資料庫在仍有降低 MAE 的作用。

表 4.29 遺漏值存有率 20%之 Iris 資料庫推估記錄

| Iris 20% | K-means | PSO | KPSO | RPSO | RKPSO |
|---|---|---|---|---|---|
| 最好 MAE | 0.3142 | 0.2652★ | 0.2656 | 0.2678 | 0.2656 |
| 平均 MAE | 0.317 | 0.3158 | 0.3169 | 0.296 | 0.2895★ |
| MAE 標準差 | 0.0124 ★ | 0.0327 | 0.0378 | 0.0135 | 0.0185 |
| | | | | | |
| 最好距離 | 31.1729 | 28.2201 | 28.2366 | 28.2143 | 28.2145 |
| 平均距離 | 31.4575 | 30.7272 | 30.878 | 28.3358 | 28.4458 |
| 距離標準差 | 1.2404 | 3.4620 | 3.2494 | 0.1326 | 0.1852 |

表 4.30 則爲上面四種遺漏值存有率在五種分群技術下的估計結果匯總，從此表可以清楚的得知以 Iris 資料庫而言 RKPSO 有較好的估計能力。

表 4.30 四種遺漏值存有率情況下五種分群技術之 20 次實驗後 MAE 比較表

| Iris | K-means | PSO | KPSO | RPSO | RKPSO | 誤差最小 |
|---|---|---|---|---|---|---|
| 5% | 0.2931 | 0.2958 | 0.2975 | 0.3116 | 0.2885 | RKPSO |
| 10% | 0.2533 | 0.2526 | 0.2432 | 0.2441 | 0.2403 | RKPSO |
| 15% | 0.3317 | 0.322 | 0.319 | 0.2983 | 0.2971 | RKPSO |
| 20% | 0.317 | 0.3158 | 0.3169 | 0.296 | 0.2895 | RKPSO |

3.Glass 資料庫:

表 4.31-4.34 爲五種分群方法在 Glass 資料庫含有 5%、10%、15%及 20%遺漏值存有率下的估計誤差及距離總值記錄。而從表 4.35 則是五種分群技術在上述四種遺漏值存有率情況下進行 20 次實驗後之平均 MAE 比較表。

表 4.31 遺漏值存有率 5%之 Glass 資料庫推估記錄

| Glass5% | K-means | PSO | KPSO | RPSO | RKPSO |
|---|---|---|---|---|---|
| 最好 MAE | 0.338★ | 0.5673 | 0.4054 | 0.4632 | 0.3656 |
| 平均 MAE | 0.5163 | 0.5721 | 0.5693 | 0.5696 | 0.4823★ |
| MAE 標準差 | 0.0797 | 0.0067★ | 0.0574 | 0.0375 | 0.0971 |
| | | | | | |
| 最好距離 | 52.048 | 58.5889 | 56.5434 | 48.1661 | 47.4596 |
| 平均距離 | 57.8075 | 66.8559 | 69.23 | 52.3083 | 52.4650 |
| 距離標準差 | 5.1589 | 6.7276 | 7.2449 | 2.9162 | 5.7707 |

表 4.32 遺漏值存有率 10%之 Glass 資料庫推估記錄

| Glass10% | K-means | PSO | KPSO | RPSO | RKPSO |
|---|---|---|---|---|---|
| 最好 MAE | 0.5232 | 0.5607 | 0.546 | 0.5282 | 0.5201★ |
| 平均 MAE | 0.576 | 0.592 | 0.598 | 0.5686 | 0.5609★ |
| MAE 標準差 | 0.0270 | 0.0190★ | 0.0555 | 0.0220 | 0.0210 |

| | | | | | |
|---|---|---|---|---|---|
| | | | | | |
| 最好距離 | 46.5651 | 53.5998 | 53.5723 | 48.1598 | 47.4093 |
| 平均距離 | 53.4726 | 68.7133 | 65.4193 | 54.8518 | 53.2574 |
| 距離標準差 | 0.5232 | 0.5607 | 0.546 | 0.5282 | 0.5201 |

表 4.33 遺漏值存有率 15%之去偏極值 Glass 資料庫推估記錄

| Glass15% | K-means | PSO | KPSO | RPSO | RKPSO |
|---|---|---|---|---|---|
| 最好 MAE | 0.5296 | 0.4121 | 0.4121 | 0.3796 | 0.3786★ |
| 平均 MAE | 0.5814 | 0.4375 | 0.419 | 0.4188★ | 0.4446 |
| MAE 標準差 | 0.0273 | 0.0572 | 0.0043★ | 0.0248 | 0.0298 |
| | | | | | |
| 最好距離 | 49.2907 | 58.2521 | 58.2521 | 46.7018 | 47.8455 |
| 平均距離 | 54.2642 | 69.1114 | 67.0279 | 53.1184 | 53.6011 |
| 距離標準差 | 3.7900 | 8.7997 | 6.8750 | 3.3089 | 3.5091 |

表 4.34 遺漏值存有率 20%之 Glass 資料庫推估記錄

| Glass20% | K-means | PSO | KPSO | RPSO | RKPSO |
|---|---|---|---|---|---|
| 最好 MAE | 0.3848 | 0.4358 | 0.4358 | 0.3738 | 0.3736★ |
| 平均 MAE | 0.4422 | 0.4692 | 0.4532 | 0.4531 | 0.4393★ |
| MAE 標準差 | 0.0447 | 0.0447 | 0.0348 ★ | 0.0474 | 0.0612 |
| | | | | | |
| 最好距離 | 46.4002 | 68.0536 | 58.0497 | 47.9671 | 47.0487 |
| 平均距離 | 56.4017 | 68.483 | 68.7722 | 47.9671 | 52.1480 |
| 距離標準差 | 6.7627 | 6.7627 | 6.5796 | 3.1231 | 3.1782 |

表 4.35Glass 資料庫之 20 次實驗後 MAE 比較表

| Glass | K-means | PSO | KPSO | RPSO | RKPSO | 誤差最小 |
|---|---|---|---|---|---|---|
| 5% | 0.5163 | 0.5721 | 0.5693 | 0.5696 | 0.4823 | RKPSO |
| 10% | 0.576 | 0.592 | 0.598 | 0.5686 | 0.5609 | RKPSO |
| 15% | 0.5814 | 0.4375 | 0.419 | 0.4188 | 0.4446 | RPSO |
| 20% | 0.4422 | 0.4692 | 0.4532 | 0.4542 | 0.4393 | RKPSO |

4.Vowel 資料庫:

表 4.36-4.39 爲五種分群方法在 Vowel 資料庫含有 5%、10%、15%及 20%遺漏值存有率下的估計誤差及距離總值記錄，而表 4.40 則是五種分群技術在上述四種遺漏值存有率情況下進行 20 次實驗後之平均 MAE 比較表。

從表 4.36 中可明顯看出 PSO 及改良 PSO 大幅降低估計遺漏值之 MAE，而其中又以 RKPSO 有明顯最佳表現，與 K-means 之平均 MAE 相比降了近 100 的 MAE。

表 4.36 遺漏值存有率 5%之 Vowel 資料庫推估 MAE

| 5% | K-means | PSO | KPSO | RPSO | RKPSO |
|---|---|---|---|---|---|
| 最好 MAE | 203.8073 | 175.4761 | 106.3816★ | 166.7048 | 132.2301 |
| 平均 MAE | 280.9034 | 208.8453 | 210.3507 | 208.7795 | 185.3032★ |
| MAE 標準差 | 28.3576 | 23.2270★ | 43.3633 | 31.6720 | 26.3104 |
| | | | | | |
| 最好距離 | 130.094 | 125.5044 | 125.6451 | 125.4636 | 125.2994 |
| 平均距離 | 134.1065 | 129.04 | 131.7697 | 127.3583 | 126.9522 |
| 距離標準差 | 6.8455 | 4.1169 | 5.6210 | 1.0903 | 1.2489 |

表 4.37 爲 Vowel 資料庫 10%遺漏值存有率下，RKPSO 除了在最好及平均 MAE 上有較好的表現外，其標準差更是表現優異，大幅提昇估計品質。

表 4.37 遺漏值存有率 10%之 Vowel 資料庫推估 MAE

| 10% | K-means | PSO | KPSO | RPSO | RKPSO |
|---|---|---|---|---|---|
| 最好 MAE | 191.3786 | 199.1835 | 140.001★ | 173.9799 | 157.156 |
| 平均 MAE | 240.4019 | 215.6348 | 201.172 | 203.7089 | 177.501★ |
| MAE 標準差 | 23.6452 | 32.3502 | 27.4687 | 31.8331 | 9.3793★ |
| | | | | | |
| 最好距離 | 131.4516 | 123.7864 | 123.8149 | 123.7642 | 124.1982 |
| 平均距離 | 134.9205 | 126.0875 | 128.4115 | 128.7155 | 125.7768 |
| 距離標準差 | 4.7246 | 3.7907 | 4.3980 | 4.9650 | 0.9806 |

表 4.38 爲 Vowel 資料庫 15%遺漏值存有率下，同 Vowel 資料庫 10%遺漏值存有率，RKPSO 除了在平均 MAE 上亦有較好的表現外，其標準差更是表現優異。

表 4.38 遺漏值存有率 15%之 Vowel 資料庫推估 MAE

| 15% | K-means | PSO | KPSO | RPSO | RKPSO |
|---|---|---|---|---|---|
| 最好 MAE | 181.7731 | 161.217★ | 162.5186 | 266.1911 | 163.2652 |
| 平均 MAE | 230.465 | 196.2055 | 193.7800 | 182.6086 | 179.7402★ |
| MAE 標準差 | 23.0371 | 20.9009 | 18.5286 | 21.6974 | 10.0421★ |
| | | | | | |
| 最好距離 | 132.9401 | 122.3618 | 122.0302 | 122.96 | 122.2013 |
| 平均距離 | 135.4697 | 126.7291 | 125.2722 | 125.014 | 124.7221 |
| 距離標準差 | 1.5952 | 3.7306 | 3.3354 | 1.6705 | 2.0462 |

表 4.39 可看出 KPSO、RPSO 及 RKPSO 都有不錯的估計表現，而以平均來說仍是 RKPSO 估計較佳。

表 4.39 遺漏值存有率 20%之 Vowel 資料庫推估 MAE

| 20% | K-means | PSO | KPSO | RPSO | RKPSO |
|---|---|---|---|---|---|
| 最好 MAE | 206.5176 | 165.1368 | 167.224 | 155.6257★ | 157.1018 |
| 平均 MAE | 231.1755 | 187.6708 | 187.2437 | 188.0734 | 185.0526★ |
| MAE 標準差 | 11.9854★ | 16.5212 | 13.0913 | 22.9056 | 14.8001 |
| | | | | | |
| 最好距離 | 133.7445 | 120.0287 | 121.125 | 119.6461 | 120.6044 |
| 平均距離 | 135.7535 | 123.0951 | 125.4029 | 122.5476 | 123.718 |
| 距離標準差 | 1.4052 | 3.4626 | 5.0447 | 1.9856 | 3.7668 |

表 4.40 爲記錄 5 四種遺漏值存有率情況下五種分群技術之 20 次實驗後平均 MAE，以此表來看我們可得知 RKPSO 在 Vowel 資料庫有較佳的估計表現

表 4.40 四種遺漏值存有率情況下五種分群技術之 20 次實驗後 MAE 比較表

| | K-means | PSO | KPSO | RPSO | RKPSO | 誤差最小 |
|---|---|---|---|---|---|---|
| 5% | 280.9034 | 208.8453 | 210.3507 | 208.7795 | 185.3032 | RKPSO |
| 10% | 240.4019 | 215.6348 | 201.172 | 203.7089 | 177.501★ | RKPSO |
| 15% | 230.465 | 196.2055 | 201.1720 | 182.6086 | 179.7402 | RKPSO |

| | | | | | | |
|---|---|---|---|---|---|---|
| 20% | 231.1755 | 187.6708 | 187.2437 | 188.0734 | 185.0526 | RKPSO |

三. 實驗小結

綜觀上面幾種資料庫之推估實驗結果，雖然在 OIL 及 GLASS 這類對應於資料筆數而言，維度偏高的資料庫其 MAE 仍無法有效大幅降低，但已較原先 K-means 分群技術有較好的表現。而以總實驗結果來說，RKPSO 皆有較好的推估表現，其原因在於它融合了前面幾項技術的優點，尤以人工或 Vowel 資料庫更可明顯看出 MAE 大幅降低，因此由此次研究我們可證實 RKPSO 在降低遺漏值推估誤差上有一定的成效。

# 第五章 結論與未來研究方向

## 一. 研究結論

本論文是以結合K-means、PSO及反彈機制的方法來解決遺漏值估計的問題，期望以K-means快速收斂的特性，PSO跳脫區域最佳解並搜尋全域最佳解之能力和增加PSO尋解效果的反彈機制，加以系統化整合後，能有效的縮小遺漏值估計誤差，並得到以下結論:

1. Iteration-based推估模式在時間上大幅優於Run-based推估模式
2. K-means收斂速度雖快，但易陷入區域最佳解與分錯群組。
3. 反彈機制加上KPSO對於估計遺漏值之平均誤差，明顯優於PSO與K-means。
4. 反彈機制加上KPSO對於資料分群之距離值，不論最佳或平均值，與本研究其他四種分群法相比，皆爲最佳。
5. 分群結果並不如預期的和估計誤差率之間有絕對的關係，其原因是實際資料庫之資料分佈狀況並不一定呈規則性分佈，加上在固定群數下每一群群內的相似差異大，因此分群距離總值較佳，遺漏值估計誤差率不一定低。

## 二. 未來研究方向與應用

本論文使用反彈機制加上 KPSO 分群法，以不斷分群迭代再回填，讓估計值隨著迭代不斷進行演化的方式。本研究從此次實驗中發現二項特點，一是本研究所提出的方法雖大幅的縮短這類演化式估計之時間，但比起統計式分群法，時間上仍有進步的空間若是將其拉進電腦同步處理的方式，則可使本研究方法在時間和降低估計誤差上都有最佳的表現。其二是此次研究所使用的資料庫皆有預先設定群數，但有可能某些資料庫並未確定分群數，因此若是能在此演算法前加上動態分群則可應用到那些未確定群數的資料庫上。

# 參考文獻

## 一、中文參考文獻


[1] 林俊男「應用類神經網路法於遺漏值問題之研究」，南華大學資訊管理學系研究所碩士論文(2005)，嘉義

[2] 張榮芳「電力用戶負載歸類及整合」，國立中山大學電機工程研究所博士論文（2001），高雄。

[3] 葉思緯「應用粒子群最佳化演算法於多目標存貨分類之研究」，元智大學工業工程與管理研究所碩士論文（2003），桃園。

[4] 邱怡瑛「質群演算法(PSO)於多組解方程最佳化問題之研究」，元智大學工業工程與管理研究所碩士論文(2004)，桃園。

[5] 鄧永亟「利用 PSO 演算法探討高速銑削最佳化」，大同大學機械工程學系研究所碩士論文(2004)，台北。

[6] 劉德誠「以 PSO 爲基礎的臉部偵測系統」，長庚大學資訊管理研究所碩士論文(2005)，台北縣。

[7] 林文彬「基於粒子群最佳化演算法之奈米定位控制系統設計」，宜蘭大學電機工程學系研究所碩士論文(2007)，宜蘭。

[8] 丁一賢、陳牧言,資料探勘(2005), 滄海書局，台中。

## 二、英文參考文獻


[9] Fayyad, U.and Irani, K. (1993). "Multi-interval discrimination of c ontinuous valued attributes for classification learning". *In Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*, pp. 1022-1027

[10] Kao, I. W. Tsai .C. Y. and Wang .Y. C. (2007). "An Effective Particle Swarm Optimization Method for Data Clustering"*,The 2007 IEEE International Conference on Industrial Engineering and Engineering Managemen*, pp. 548-552

[11] Chen.C.Y and Ye .F (2004), "Particle Swarm Optimization Algorithm and Its Application to Clustering Analysis"*.International Conference on Networking,Sensing Control*,pp 789-794

[12] Little, R. J. A. and Rubin, D. B. (1987). Statistical analysis with missing data. New York：Wiley.

[13] A. Ragel and B. Cremilleux(1998), “Treatment of Missing Values for Association Rules,” *Proceeding of the Second Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD-98),* pp.258-270.,

[14] Pedreira, C.E and Parente, E. (1995), "Neural Networks with Missing Values Attributes", *IEEE International Conference on Neural Networks,* pp. 3021-3023,

[15] McQueen, J.B.,( 1967)"Some Methods of Classification and Analysis of Multivariate Observations", *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, pp. 281-297.

[16] Kaufman, L. and Rousseeuw, P.J.,( 1990)"Finding groups in data: an Introduction to cluster analysis", John Wiley and Sons.

[17] Cheng, C.H and Wang,J.W (2006) "A new approach for estimating null value in relational database",*Soft Computing*, pp.104-114.

[18] Han ,J and Kamber, M(2000)”Data Mining:Concepts and Techniques”.Morgan Kaufmann,New york

[19] Deneubourg, J. L., Goss, S., Franks, N., Sendova-Franks, A., Detrain, C. and Chretien, L., (1991), “The dynamics of collective sorting robot-like ants and ant-like robots”. *In Proc. of the 1st Conf. on Sim. of Adaptive Behavior*, pp.356-363.

[20] Boyd, R. and P. J. Richerson, 1985, Culture and the Evolutionary Process, University of Chicago Press, Chicago, IL.

[21] Reynolds C W Flocks.Herds and Schools(1987):A Distributed Behaviora

Model[J]. *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*,pp.25～34.

[22] Chen.C.Y and Ye .F , (2003) “K-means Algorithm Based on Particle Swarm Optimization,” *International Conference on Informatics, Cybernetics, and Systems, Taiwan* , pp.1470-1475.

[23] Eberhart, R. C. and Kennedy, J. (1995)” A new optimizer using particle swarm theory”. *Proceedings of the sixth international symposium on micro machine and human science* , pp.39-43.

[24] Kennedy, J.and Eberhart, R. C. (1995) “Particle Swarm Optimization”,*Proceedings of the lEEE International Joint Con-Scrence on Neural Networks*, pp.1942-1948.

[25] Eberhart, R. C., Shi, Y. H. (1998) “Comparison Between Genetic Algorithms and Particle Swarm Optimization” *Lecture Notes in Computer Science*, pp. 611-616

[26] ftp://ftp.ics.uci.edu/pub/machine-learning-datasets/