改良式粒子族群最佳化用於資料分群

莊麗月[1]　　邱國鑫[2]　　楊正宏[2,3]
[1]義守大學化學工程系 chuang@isu.edu.tw
[2]國立高雄應用科技大學電子工程系 tpvuu@yahoo.com.tw
[3]稻江科技暨管理學院網路系統學系 chyang@cc.kuas.edu.tw

摘要

資料分群(data clustering)依據資料集中的特徵將擁有相同類別的資料歸類成群，並找出各類別的中心點，以簡化資料複雜性。本研究提出改良式粒子族群最佳化演算法(Particle Swarm Optimization, PSO)進行資料分群，為防止 PSO 快速收斂導致中心點落入區域最佳解，本研究結合互補式策略(Complementary)增加粒子多樣性，進而避免粒子族群因多次迭代後粒子之間相似度過高而使族群陷入區域最佳解(CPSO)。本研究使用六筆資料(UCI Repository)進行測試，並與相關文獻之演算法做比較，實驗結果表示本研究方法能較其他方法找到較短的群集內距離總和與較低的錯誤率。

關鍵詞：資料分群、粒子族群最佳化演算法、互補式

1. 前言

群集分析主要是將資料中有相關聯的資料集合一起[3]，而所有資料分出來的各個群合起來即為群集(Cluster)[1]。群集分析時並沒有事先指定類別，純粹依資料的相似性來識別，利用數學函數來運算，進而找出中心點，所以視為非監督式學習(Supervised learning)。

分群分為分割式分群法與階層式分群法兩種，分割式分群法是利用離群中心最短距離來計算，主要找出大小相似且形狀為圓形的群集，例如常用 K-means 演算法[7]。K-means 於 1967 年由 MacQueen 提出，被廣泛應用於分群技術，其原理是設 K 個群集為中心點開始分群，最終將分群的資料分成 K 個集合，其優點是容易且高效率[2]，缺點是群集中心容易受偏移值影響而落入區域最佳解[8]。另一種是階層式分群法，利用密集度導向，主要找出任意形狀的群集，例如常用 BIRCH[10]、Chameleon[5]演算法。本研究即利用分割式分群法進行分群，利用離群中心最短距離將資料集中的資料歸屬到所屬的群中。

本研究使用改良式粒子族群最佳化—CPSO。粒子族群最佳化 (Particle Swarm Optimization, PSO) [6]是一種以族群搜尋為基礎的演算法，粒子會依自己過去的經驗及族群共同經驗進行移動，進而快速提升各粒子的適應值。PSO 目前用於以下研究，例如系統設計、函數最佳化[9]、分類、型樣識別、機器人應用、生物系統模擬、排程、決策制定及路由選擇、神經網路訓練、網路安全、模擬和識別等。粒子族群最佳化經過多次迭代後，粒子與 *Gbest* 之距離過近時會造成移動距離變小，導致粒子落入區域最佳解，為改善此問題，本研究加入互補式來協助粒子跳脫區域最佳解，藉此增加獲得更佳中心點的機會。

本研究使用 UCI Repository 的六筆真實資料進行分群測試，資料包含 Crude Oil、Contraceptive Method Choice (CMC)、Wine、Breast Cancer、Vowel 與 Iris Plants，利用這些資料驗證本研究方法是否可以找出每個群的最佳解(中心點)及最低錯誤率。實驗證明本研究方法能優於其他文獻所提出的方法(K-means、NM-PSO、K-PSO、K-NM-PSO、

PSO[4])，例如在 Iris 資料集中本研究能找到最短距離 96.66 且錯誤率只有 10%，而在 Cancer 資料集中錯誤率更小到 3.51%，所以證明本研究能找出最短的群集內距離總和與最低的錯誤率。

2. 方法

2.1 粒子族群最佳化 (Particle Swarm Optimization, PSO)

粒子族群最佳化[6]於 1995 年由 James Kennedy 和 Russell Eberhart 所提出，其概念源自鳥群、魚群覓食時的特性所發展出的一種具有群體智慧的最佳化演算法。在粒子族群最佳化中，每個粒子均代表一個解(即鳥群裡的每隻鳥)。

粒子族群最佳化主要的流程是將粒子族群分散於解空間內，透過粒子族群在此空間中進行搜尋並將搜尋到的最好資訊傳遞給整個粒子族群，粒子族群會依據這些訊息與自身經驗來改變移動方向，使整個族群往好的區域搜尋，進而於整個解空間中找出最佳解。粒子族群最佳化和傳統最佳化演算法的不同在於粒子族群最佳化是屬於一個多目標搜尋法，在同一時間內有多個粒子同時進行搜尋的動作，其中多個粒子稱之為族群。粒子族群最佳化透過不斷的迭代來改變搜尋位置，以求得問題的最佳解。

粒子族群最佳化利用資料集中群的數量 N 與資料維度 D 隨機產生粒子(Particle)在一限定範圍內(X)，再利用亂數為粒子產生一速度(V)，粒子依自我最佳經驗(*Pbest*)、族群最佳經驗(*Gbest*)與原始粒子移動速度(V)三個項目，將找出一個活動空間並且隨機落在空間內，此速度為粒子的新速度。當粒子移動到新位置即計算此位置的適應值，若此適應值優於 *Pbest*，此 *Pbest* 將被新適應值取代，成為粒子的自我最佳經驗。當全部粒子算出適應值後選出最好的 *Pbest*，若此 *Pbest* 優於 *Gbest* 則取代。依此循環，粒子將移動到最佳解(中心點)。圖 1 為該演算法之流程。

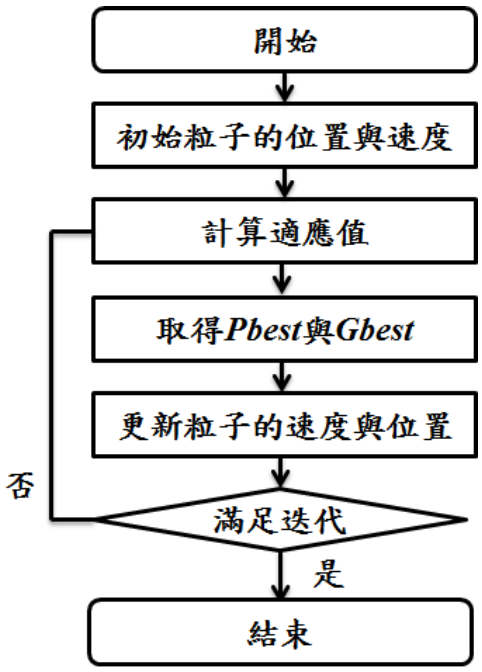

圖 1 PSO 流程圖

2.1.1 粒子編碼

分群資料被定義在一個 D 維的空間中，首先為了防範粒子初始化時隨機產生超出資料範圍，因此必須先找出該資料的各維度邊界，設定 Max、Min 值來限制每顆粒子(Particle)的位置。依資料的類別數而定隨機產生 N 個群集中心點與 $3N$ 個粒子，其第 N 個粒子位

置為 $X_n=(x_{n1},x_{n2},x_{n3},...,x_{nD})$ 與速度 $V_n=(v_{n1},v_{n2},v_{n3},...,v_{nD})$，且其範圍接限制在 $[X_{\min},X_{\max}]^D$，粒子的編碼為各中心點相串，所以其維度為 $D\times N$。例如 $X_{\max}$ 均為 5、$X_{\min}$ 均為 0，2 維的空間中與 3 個群集，隨機產生群集中心點 P1(1, 0)、P2(3, 2)、P3(5, 4)，粒子編碼即為(1, 0, 3, 2, 5, 4)。

2.1.2 判斷適應值

辨別資料點為何群的方法為離群中心最短距離(分割式分群法)，故本研究採用歐基里德距離(Euclidean Distance)做為計算粒子適應值(Fitness value)的評估方法。將群集中心點與隸屬於同一群集的資料點之距離做加總，加總的值越小則代表結果越好，此距離總和為粒子之適應值，如公式(1)，此公式為歐基里德距離，其中 X 代表 i 個群集中心點，Z 代表 j 個資料點。本研究另一種評估準則為錯誤率，即資料被分配到不正確群集中的比率，如公式(2)，A_i與B_i 表示第i個資料點所屬的群集與分群後所屬的群集，若 A_i與B_i 相同表示分群正確，以0表示，反之為1。錯誤率越低，表示分群效果越好。

$$\mathrm{F}=\sum\left\|X_i-Z_j\right\|, i=1,...,k,\ j=1,...,n \tag{1}$$

$$ER=\left(\sum_{i=1}^{n}(if\ \ (A_i=B_i)\ then\ 0\ else\ 1)\div \mathrm{n}\right)\times 100 \tag{2}$$

2.1.3 粒子移動位置

Pbest 為每顆粒子從過去到目前最佳的適應值，*Gbest* 為每次迭代中選出最好的 *Pbest*，也就是全體群集到目前為止的最佳適應值 *Gbest*，將每顆粒子的 *Pbest*、*Gbest* 帶入公式(3)，粒子隨機移動至範圍內產生新的位置與產生新的速度，如圖 2。粒子到新的位置即返回步驟 2.1.2.算出其適應值，直到滿足迭代次數。

公式(3)中，w 為粒子移動距離的慣性權重值，c_1、c_2 為 *Pbest* 與 *Gbest* 的學習因子，本研究使用文獻上提供的值 2。r_1、r_2 為範圍在 0 到 1 之間隨機產生的亂數，x_{id}^{old} 與 v_{id}^{old} 為粒子更新前的位置及速度，v_{id}^{new} 與 x_{id}^{new} 為更新後的位置及速度。

$$v_{id}^{new}=w\times v_{id}^{old}+c_1\times r_1\times\left(Pbest_{id}-x_{id}^{old}\right)+c_2\times r_2\times\left(Gbest_{id}-x_{id}^{old}\right) \tag{3}$$

$$x_{id}^{new}=x_{id}^{old}+v_{id}^{new} \tag{4}$$

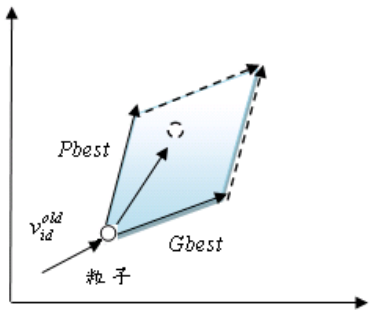

圖 2 粒子移動示意圖

2.2 互補式(Complementary)

由於粒子族群最佳化經過多次迭代後，若粒子與 *Gbest* 之距離過近，會使每次移動距離將變小，導致粒子容易落入區域最佳解[8]，為了使粒子跳脫此區域，本研究提出互補式方法來改善粒子族群最佳化的缺點。

互補是利用數學互為補角的概念，若兩角合為 180 度即為互補。互補後粒子位置為未互補前之對應位置，如公式(5)為互補式，將解空間邊界的最大值($X_{\max}$)加上最小值($X_{\min}$)再減去目前位置(x_{id}^{old})，即為粒子新產生的互補位置(x_{id}^{new})。舉例說明，假定各維度之邊界最大值($X_{\max}$)均為 5、各維度之最小值($X_{\min}$)均為 0，粒子位置為(1, 0, 3, 2, 5, 4)，互補後粒子位置為(4, 5, 2, 3, 0, 1)。

$$x_{id}^{new} = \left(X_{\max} + X_{\min}\right) - x_{id}^{old} \tag{5}$$

本研究以最佳解(*Gbest*)連續重複 10 次無提升來判定粒子族群落入區域最佳解[8]，以便執行互補動作。由於粒子與 *Gbest* 之距離過近導致適應值無法提升，因此以族群中適應值較差的50%個粒子進行互補，使其脫離區域最佳解，並試圖搜尋區域最佳解以外的全域最佳解。

CPSO 執行流程如以下步驟：

步驟1. 取得邊界值 $X_{\max}$ 與 $X_{\min}$，在範圍內產生每個粒子的位置及速度，粒子的位置即為每個群集之中心點。

步驟2. 利用距離將離群中心最短距離之資料點歸屬於群集中心點。

步驟3. 計算每個粒子之適應值。

步驟4. 每個粒子的適應值個別與目前之 *Pbest* 相比，若適應值比目前 *Pbest* 佳，適應值即取代 *Pbest*。

步驟5. 計算出此次迭代最好的 *Pbest*，若此 *Pbest* 優於 *Gbest*，則 *Pbest* 取代 *Gbest*。

步驟6. 利用公式(3)、公式(4)，計算出每個粒子的新位置與速度。

步驟7. 重複執行步驟 2 到步驟 6，直到滿足迭代次數。

3. 結果與討論

3.1 資料集

本研究採用 Crude Oil、Contraceptive Method Choice (CMC)、Wine、Breast Cancer、Vowel 與 Iris Plants 作為測試資料，以上六組是由美國加州大學資訊與電腦科學系提供的實際資料集(ftp://ftp.ics.uci.edu/pub/machine-learning-data-bases/)資料詳細內容如下：

3.1.1 Crude Oil (*n* = 56, *d* = 5, *k* = 3)

Crude Oil 為天然石油資料，每筆資料中皆含有五種屬性，分別是釩(Vanadium)、鐵(Iron)、鈹(Beryllium)、飽和碳氫化合物(Saturated Hy-drocarbons)與芳香族碳氫化合物(Aromatic Hydrocarbons)；天然石油資料庫分成三種，分別為 7 筆 Wilhelm、11 筆 Sub-Mulnia 與 38 筆 Upper。

3.1.2 Contraceptive Method Choice (*n* = 1473, *d* = 9, *k* = 3)

Contraceptive Method Choice 為避孕器資料，其中包含九種屬性，經由分群可將資料歸納為不使用(No-use) 629 筆、長期使用(Long-term) 334 筆以及短期使用(Short-term)

510 筆。

3.1.3 Wine (n = 178, d = 13, k = 3)

Wine 為葡萄酒資料，其中包含 13 種屬性；葡萄酒種類共有 3 種，分別是 class1 (59 筆)、class2 (71 筆)與 class3 (48 筆)。

3.1.4 Breast Cancer (n = 683, d = 9, k = 2)

Breast Cancer 為乳癌資料，扣除遺失資料，共整理出 683 筆資料，其中包含 9 種屬性，是由良性細胞(Benign)以及惡性細胞(Mali-gnant)兩種所組成。

3.1.5 Vowel (n = 871, d = 3, k = 6)

Vowel 為母音資料庫，是由 871 筆印地安語言之母音資料所構成，其中包含 3 種音頻屬性，並將之分類為六種母音，分別為 δ (72 筆)、a (89 筆)、i (172 筆)、u (151 筆)、e (207 筆)與 o (180 筆)。

3.1.6 Iris Plants (n = 150, d = 4, k = 3)

Iris Plants 為鳶尾植物資料，其中包含了萼片(Sepal)與花瓣(Petal)的長度(Length)、寬度(Width)等四種屬性；鳶尾植物資料庫是由 Iris Setosa (50 筆)、Versicolour (50 筆)與 Virginica (50 筆)這 3 種鳶尾花 (150 筆)的種類所組成。

表一： 資料詳細內容

資料集	維度	群集數	資料數目(括號內為各類別之資料數目)
Crude Oil	3	5	56 (7, 11, 38)
Contraceptive Method Choice	3	9	1473 (629, 334, 510)
Wine	3	13	178 (59, 71, 48)
Breast Cancer	2	9	683 (444, 239)
Vowel	6	3	871 (72, 89, 172, 151, 207, 180)
Iris Plants	3	4	150 (50, 50, 50)

3.2 實驗結果

為公平起見，本研究之參數與各演算法之參數一致。設定參數 D 為資料維度與類別的乘積，再 $X_{\max}$ 與 $X_{\min}$ 的範圍內隨機產生 $3D$ 個粒子，分別對六個多維度資料集進行 $10D$ 次的迭代(依許多文獻表示 $10D$ 為效率很好的迭代次數)，總共做 20 次實驗。

本研究提出互補式粒子族群最佳化應用於分群問題，並與其他五種演算法(K-means、PSO、NM-PSO、K-PSO、K-NM-PSO)[4]進行比較，比較結果如表二所示。本研究使用平均值、標準差、最佳值三種型態進行比較，平均值為實驗 20 次的群集內最短距離總和之平均，最佳值為實驗 20 次裡出現最短的群集內距離總和，利用標準差了解演算法的穩定性，標準差值越大表示每次結果的差異越大，標準差值越小表示每次結果越變化越小，故標準差越小越好。藉由表二實驗結果得知，PSO 較諸多演算法之結果劣，其原因是因為族群多次迭代後粒子與 *Gbest* 之距離過近使得移動距離變小，使得粒子陷入區域最佳解而無法找尋此區域外的可行解。為了預防此情況發生本研究使用互

補式來協助跳離區域最佳解。雖然 K-means 演算法的演算速度快，但其結果與一般 PSO 一樣，當 K-means 運算到最後容易陷入區域最佳解而無法跳離。本研究使用 PSO 加入互補式進行比對，在 Iris、Crude Oil、CMC、Cancer 平均值均優於其他七種演算法，且穩定效果明顯，也能有效找出 CMC、Cancer 之最佳值，而在 Vowel 平均值能有效優於 K-means[7]。表示 CPSO 效果更加強大且穩定性好。

表三為 CPSO 與五種演算法對資料集進行分群後的錯誤率比較，平均值為實驗 20 次錯誤率的平均，最佳值為錯誤率最低的值，利用標準差了解穩定性。由於一般 PSO 容易落入區域最佳解[8]，導致無法尋得全域最佳解，實驗結果得知，加入互補式之 PSO 能跳離區域最佳解使錯誤率有效降低。以 Iris、Cancer、Vowel 與 CMC 的平均值來說，能有效降低錯誤率且優於其他演算法[4]，甚至 Iris、Cancer 與 CMC 三種資料集中能準確的運算出群集，使每次分群時結果一致。表示 CPSO 能有效的降低錯誤率、更準確的分辨群集且更穩定。

表二： 五種分群演算法的群集內距離總和比較表

資料集	評估標準	K-means	NM-PSO	K-PSO	K-NM-PSO	PSO	CPSO
Vowel	平均值 (標準差) 最佳值	159242.87 (916) 149422.26	151983.91 (4386.43) 149240.02	149375.70 (155.56) 149206.10	**149141.40** (120.38) **149005.00**	168477.00 (3715.73) 163882.00	155191.79 (4177.74) 150913.56
Iris	平均值 (標準差) 最佳值	106.05 (14.11) 97.33	100.72 (5.82) 96.66	96.76 (0.07) 96.66	96.67 (0.008) 96.66	103.51 (9.69) 96.66	**96.66** (0.0001) 96.66
Crude Oil	平均值 (標準差) 最佳值	287.36 (25.41) 279.20	277.59 (0.37) 277.19	277.77 (0.33) 277.45	277.29 (0.095) **277.15**	285.51 (10.31) 279.07	**277.22** (0.019) 277.21
CMC	平均值 (標準差) 最佳值	5693.60 (473.14) 5542.20	5563.40 (30.27) 5537.30	5532.90 (0.09) 5532.88	5532.70 (0.23) 5532.40	5734.20 (289.00) 5538.50	**5532.18** (0.00) **5532.18**
Cancer	平均值 (標準差) 最佳值	2988.30 (0.46) 2987	2977.70 (13.73) 2965.59	2965.80 (1.63) 2964.50	2964.70 (0.15) 2964.50	3334.66 (357.66) 2976.30	**2964.39** (0.0001) **2964.39**
Wine	平均值 (標準差) 最佳值	18061.00 (793.21) 16555.68	16303.00 (4.28) **16292.00**	16294.00 (1.70) **16292.00**	16293.00 (0.46) **16292.00**	16311.00 (22.98) 16294.00	**16292.65** (0.511) 16292.18

註：粗體表示六種分群演算法執行相同資料集後，群集內距離的平均值與最佳值為八種分群演算法中最小之方法。

表三： 五種演算法的錯誤率比較表

資料集	評估標準	K-means (%)	NM-PSO (%)	K-PSO (%)	K-NM-PSO (%)	PSO (%)	CPSO (%)
Vowel	平均值 (標準差) 最佳值	44.26 (2.15) 42.02	41.96 (0.98) 40.07	42.24 (0.95) 40.64	41.94 (0.95) 40.64	44.65 (2.55) 41.45	**41.19** (1.77) **37.08**
Iris	平均值 (標準差) 最佳值	17.80 (10.72) 10.67	11.13 (3.02) **8.00**	10.20 (0.32) 10.00	10.07 (0.21) 10.00	12.53 (5.38) 10.00	**10.00** (0.00) 10.00
Crude Oil	平均值 (標準差) 最佳值	24.46 (1.21) **23.21**	24.29 (0.75) **23.21**	24.29 (0.92) **23.21**	**23.93** (0.72) **23.21**	24.64 (1.73) **23.21**	25.62 (0.87) 25.00
CMC	平均值 (標準差) 最佳值	54.49 (0.04) 54.45	54.47 (0.06) 54.38	**54.38** (0.00) 54.38	**54.38** (0.054) 54.31	54.41 (0.13) **54.24**	**54.38** (0.00) 54.38
Cancer	平均值 (標準差) 最佳值	4.08 (0.46) 3.95	4.28 (1.10) 3.66	3.66 (0.00) 3.66	3.66 (0.00) 3.66	5.11 (1.32) 3.66	**3.51** (0.00) **3.51**
Wine	平均值 (標準差) 最佳值	32.12 (0.71) 29.78	28.48 (0.27) **28.09**	28.48 (0.40) **28.09**	**28.37** (0.27) **28.09**	28.71 (0.27) **28.09**	28.54 (0.35) **28.09**

註：粗體表示六種分群演算法執行相同資料集後，錯誤率的平均值與最佳值為八種分群演算法中最低之方法。

4. 討論

本研究中之結果可從表三中發現，除 Vowel、Wine 外都能找到最短距離，最低標準差，且最低平均值。而從表三中發現，Vowel 利用 K-NM-PSO 為群集內距離總和最短的演算法，但是錯誤率最佳值卻是群集內距離總和較長的互補式 PSO 較低；Crude Oil 的錯誤率較高，但是群集內距離總和卻比其他演算法短，因為實際資料集的資料分佈不一定成規則分佈，若有資料群重疊，則可能出現此情形，如圖 4，灰色地帶為重疊部分，粒子只能找距離中心點最近之資料點作歸屬動作，而導致錯誤率上升，所以群集內距離總和越短錯誤率不一定越低[4]。

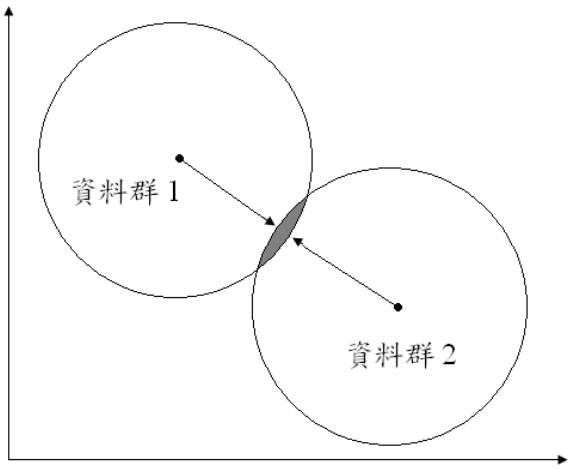

圖 3 資料群重疊示意圖

雖然 K-means 容易且高效率的時間複雜度，但是容易受偏移值影響群集中心與容易落入區域最佳解，而且無法跳離，使得錯誤率變高。而 K-NM-PSO 是利用 K-means 演算法、Nelder-Mead 演算法加 PSO 的結合，其能改善 PSO 的收斂速度，進而找到全域最佳解，但是當區域解太多時容易落入區域最佳解，只能利用緩慢的 PSO 脫離區域最佳解，且經過多次迭代後，若粒子與 *Gbest* 之距離過近時每次移動距離變小，導致粒子容易落入區域最佳解，所以本研究加入 CPSO。

解決方法如圖 4，當多次迭代中，*Gbest* 之最佳適應值連續 10 次無提升時，即判定粒子落入區域最佳解，此時進行互補動作，將粒子族群中適應值較差的 50%個粒子進行互補，如圖 4.1 所示，假設 $X_{\max}$ 為 10、$X_{\min}$ 為 0，挑選後 50%之粒子 S_1=[5,8]、S_2=[6,5]、S_3=[9,5]，經過公式(5)互補後為 C_1=[1,5]、C_2=[1,3]、C_3=[2,1]，如圖 4. 2，粒子即跳離區域最佳解，且持續受 *Gbest* 影響移動為 C_1=[3,6]、C_2=[3,4]、C_3=[3,3]，如圖 4.3，若移動後粒子的適應值 *Gbest*[6,4]優於 *Gbest*[7,7]，新適應值即取代 *Gbest*，使得粒子能有機會朝全域最佳解移動，如圖 4.4。藉由此循環能有效且快速的跳離區域最佳解，使演算法能更快找到正確的全域最佳解。

互補式 PSO 受收斂問題影響並不明顯，PSO 或 K-means 等演算法迭代一定次數後便開始加速收斂，導致後期迭代的適應值無法提升。而互補式 PSO 卻不會出現此問題，粒子會因為適應值重複而進行互補的關係，所以適應值將有機會改變而不會因為適應值無法提升而收斂。

實驗結果得知，互補式 PSO 在群集內距離總和與錯誤率的比較上能有較佳的成效，表示本研究方法優於文獻之演算法。

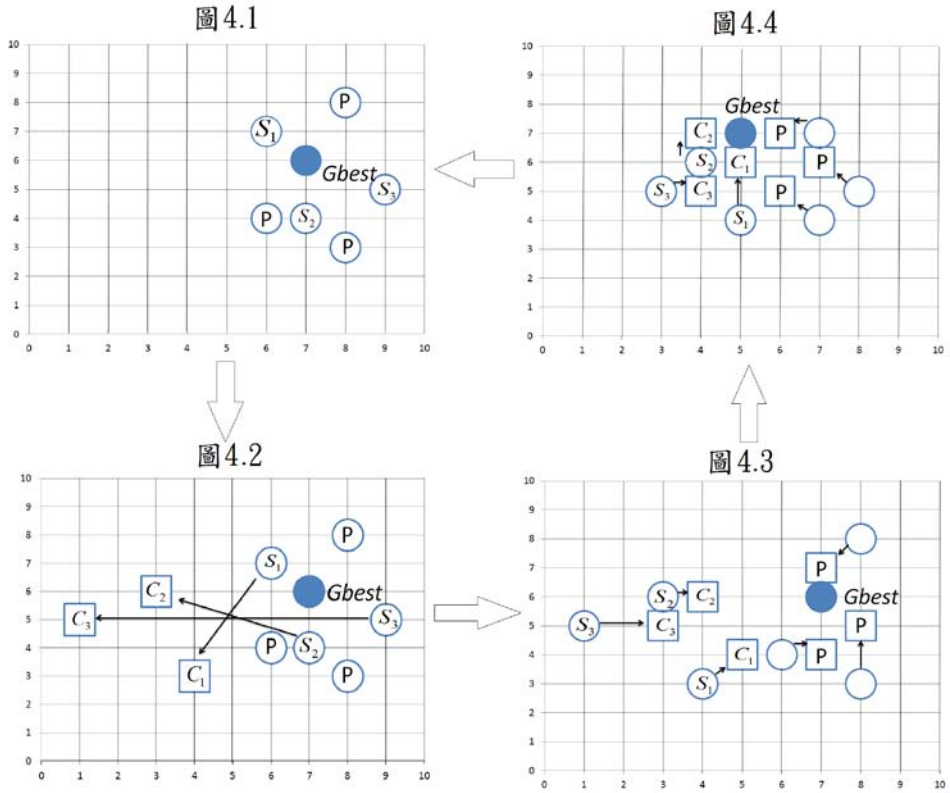

圖 4 互補式驗算法示意圖

5. 結論

本研究利用粒子族群最佳化進行初步的粒子分群，但我們發現此方法在一定的迭代次數後，粒子族群會呈現收斂狀態，稱為落入區域最佳解，而 PSO 中沒有解決這個問題的機制，因此加入了互補式來解決這個問題。使用互補式 PSO 後粒子可以有效率的跳離區域最佳解，利用互補，粒子將移動到對應之位置，粒子將有機會往正確的最佳解移動。在結果中我們比較了 CPSO 與其他的演算法，可以發現此方法確實能夠使粒子跳脫區域最佳解，使粒子搜尋到更好的解，而與其他方法比較也是大幅降低錯誤率與較短的群集內距離總和。結果證明本研究提出之方法優於其他方法，進而助於資料簡化。綜合以上實驗結果，互補式 PSO 是能有效跳離區域最佳解進而找到最佳解，且錯誤率低的演算法，即使有些資料集的群集內距離總和較其他演算法長，但是其錯誤率卻是較低的，證明互補式 PSO 較其他演算法優。

參考文獻

1. Anderberg, M.R. “Cluster analysis for applications”, Academic Press, New York 1973.
2. Chen, C.Y. and Ye, F. “Particle swarm optimization algorithm and its application to clustering analysis”, 2004 IEEE International Conference on Networking, Sensing and Control 2004, pp.789-794.
3. Han, J. and Kamber, M. “Data mining: concepts and techniques”, Morgan Kaufmann, 2000.
4. Kao, Y.T. Zahara, E. and Kao, I.W. “A hybridized approach to data clustering”, Expert Systems with Applications (Vol. 34) 2008, pp.1754-1762.
5. Karypis, G. Eui-Hong, H. and Kumar, V. “Chameleon: hierarchical clustering using dynamic modeling”, Computer (Vol. 32) 1999, pp.68-75.
6. Kennedy, J. and Eberhart, R.C. “Particle swarm optimization”, Proceedings of the IEEE International Conference on Neural Networks (Vol. 4) 1995, p.1942-1948.
7. MacQueen, J.B. “Some methods for classification and analysis of multivariate observations”, Proceedings of the Fifth Berkeley Symp. Math. Stat. Prob. 1967, pp. 281-297.
8. Selim, S.Z. and Ismail, M.A. “K-Means-Type Algorithms: A Generalized Convergence Theorem and Characterization of Local Optimality”, IEEE Transactions on Pattern Analysis and Machine Intelligence (Vol. 6) 1984, pp.81-87.
9. Shi, Y. Eberhart, R.C. “A modified particle swarm optimizer”, Proceedings of IEEE International Conference on Evolutionary Computation, Anchorage 1998. pp. 69-73.
10. Zhang, T. Ramakrishnan, R. and Livny, M. “BIRCH: An Efficient Data Clustering Method for Very Large Databases”, SIGMOD 1996, pp.103-114.

Improved Particle Swarm Optimization for data clustering

Li-Yeh Chuang[1] Guo-Sin Ciou [2] Cheng-Hong Yang[2,3]
[1]Dept. of Chemical Eng I-Shou University Kaohsiung, chuang@isu.edu.tw
[2]Dept. of Electronic Eng National Kaohsiung University of Applied Sciences Kaohsiung, tpvuu@yahoo.com.tw
[3]Dept. of Network Systems Toko University Chiayi, chyang@cc.kuas.edu.tw

Abstract

Data clustering use data set to group with the same category and classify, and find all groups center for simplify the data complexity. This study proposes Particle Swarm Optimization (PSO) for data clustering. For preventing PSO occur fasts convergence cause center trap to optimal solution region. We combine complementary strategy to Increase particle's diversity to avoid particles trapping to optimal solution region, because if particles iteration too much time combining particles too smear. This study use six really data (UCI Repository) to test and compare with other literature. The results indicate this study can find shorter intra-cluster distances and lower error rate than the other algorithm.

Keywords: Data clustering, Particle Swarm Optimization, Complementary.