

# 結合模糊分群與粒子群最佳化演算法求解旅行銷售員問題

## Hybrid Fuzzy C-means Clustering and Particle Swarm Optimization Algorithms for Traveling Salesman Problems

馮玄明

國立金門技術學院資訊工程系  
hmfeng@kmit.edu.tw

廖國隆

國立金門技術學院電資研究所  
evan910@gmail.com

### 摘要

旅行銷售員問題 (Traveling Salesman Problem, TSP) 已被證實為一 NP-Complete 問題，當節點數量越多所形成的解空間會越複雜。本文以結合粒子群最佳化演算法 (Particle Swarm Optimization, PSO)、轉換空間 (Transfer Space, TS) 與模擬退火 (Simulated Annealing, SA) 之 PSO-TS-SA 演算法為基礎，並且透過模糊分群演算法 (Fuzzy C-means Clustering, FCM) 來降低求解大型旅行銷售員問題之困難度。經過實驗與相關文獻進行比較，結果顯示本文所提出結合模糊分群與粒子群最佳化演算法的方法，確實有助於降低求解運算時間並達到相對較高之準確度。

關鍵詞：旅行銷售員問題、粒子群最佳化演算法、模擬退火演算法、模糊分群演算法。

### Abstract

It is true that the normal TSP can be proved as the well-known NP-Complete (Non-Deterministic Polynomial) path searching problems. The more cities' nodes number will cause more complex traveling path problems. This paper combines the Particle Swarm Optimization (PSO), Transfer Space (TS) and Simulated Annealing (SA) to build the PSO-TS-SA algorithm. The Fuzzy C-means Clustering (FCM) algorithm is determined to reduce the complexity of large scale traveling cities. From the experiments on several TSP, the proposed hybrid fuzzy C-means clustering and particle swarm optimization Algorithms achieve more accuracy in the lower cost of computation time.

Keywords: Traveling Salesman Problem; Particle Swarm Optimization; Simulated Annealing; Fuzzy C-means Clustering.

### 1. 前言

旅行銷售員問題 (Traveling Salesman Problem, TSP) 為一典型最佳化的問題。此方面的應用相當廣泛，如汽油引擎噴嘴之配置[20]、衛星導航GPS之路徑規劃[23]、飛機航班之排程[17]等，皆可運用旅行銷售員問題的概念來求解。

TSP求解困難度會隨著節點數量之增加而大幅上升，因此陸續有學者以分群等相關方法，希望藉此提高最佳化演算法求解大型旅行銷售員問題的效能。如Mulder和Wunsch[25]提出結合類神經 (Neural) 分類法與傳統區域搜尋法 (Local Search)，提升PSO求解大型旅行銷售員問題的速度。Zuo和Xiong[8] 提出以區域最佳群聚之螞蟻演算法[18]，將原本高維度 (high-dimension) 問題轉換成低維度 (low-dimension) 問題，求解中國旅行銷售員問題可求得最佳解。Ding[6] 等人提出 Two-Level Genetic Algorithm (TLGA) 演算法，解決大型旅行銷售員問題，實驗結果顯示相較於傳統遺傳演算法 (Genetic Algorithm)[13]，TLGA 在求解之效能較佳。Zaeri[19] 等人提出結合群聚分析 (cluster analysis) 與旅行銷售員問題，解決 NDP 問題 (newspaper production/distribution problem)，嘗試分配設備至請求節點與在每一群聚依照旅行路徑進行路由。陳彥霖[3] 提出一種改良式整數排列粒子群最佳化演算法 (Integer PSO, IPSO)，結合模糊適應共振理論 (Fuzzy ART) 來處理大型旅行銷售員問題。姚敦瀚[1] 將模糊分群 (Fuzzy C-Mean) 的機制與 Improved PSO 結合，降低對於大型旅行銷售員問題的複雜度，減少粒子搜尋的時間。馮玄明[4] 等人提出 PSO-TS-SA 演算法，以轉換空間處理 PSO 實數編碼問題，並且結合具有跳脫區域最佳解特性的模擬退火演算法，讓粒子有能力繼續朝向不同區域進行搜尋。

本文提出一種 PSO-TS-SA & FCM 演算法，

主要核心可分為：分群(Clustering)、融合(Merge)、區域搜尋(Local Search)共三個部分。以模糊分群演算法(Fuzzy C-means Clustering, FCM)[12]將 TSP 問題拆解成數群聚，各群聚先以 PSO-TS-SA 演算法求解。但最終目的是形成一個完整 TSP 可行解，故運用連結分群結合區域搜尋之方法，以主群聚(Main Cluster)做為融合起始點，將各群聚逐一融合。為了減少群聚融合後路徑交叉之現象，透過區域搜尋法改善融合後的路徑，期望能夠取得更佳之可行解。

## 2. 研究方法

### 2.1 旅行銷售員問題定義

旅行銷售員問題(Traveling Salesman Problem, TSP)為一典型最佳化的問題，已被證實為一 NP-complete 問題[9]。在 1859 年，由愛爾蘭數學家 William Rowan Hamilton 率先提出環遊世界的問題(around the world)：假設有 20 個城市，是否存在一種走法可一次走完這 20 個城市[22]？約於 1930 年，Karl Menger 首先以數學的方式來研究旅行銷售員問題，以及由 Hassler Whitney 在普林斯頓大學的研討會中提出旅行銷售員問題之名詞[5]。TSP 基本定義為給定 n 個城市以及城市與城市之間的距離，銷售員必須拜訪完所有的城市，且花費的路徑成本要能達到最短。旅行路徑會有  $(n-1)!/2$  種之組合，當城市數量越多，所形成之解空間會相當複雜。

TSP 的數學規劃模式[1][2][16]如下：

$$\text{Minimize } \sum_{i=1}^n \sum_{j=1}^n C_{ij} x_{ij} \quad (1)$$

$$\text{subject to } \sum_{i=1}^n x_{ij} = 1 \quad (j=1, \dots, n) \quad (2)$$

$$\sum_{j=1}^n x_{ij} = 1 \quad (i=1, \dots, n) \quad (3)$$

$$X = (x_{ij}) \in S \quad (4)$$

$$x_{ij} = 0 \text{ or } 1 \quad (i, j=1, \dots, n) \quad (5)$$

$$\text{Cost}(C_1, C_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (6)$$

$$S = \left\{ (X_{ij}) : \sum_{i \in Q} \sum_{j \notin Q} X_{ij} \geq 1 \right. \\ \left. \text{for every nonempty proper subset } Q \text{ of } N \right\} \quad (7)$$

公式(1)代表總成本最小化的目標式，其中 n 代表節點數量， $C_{ij}$  為節點 i 至節點 j 的成本，如公式(6)以歐幾里得距離(Euclidean distance)做為計算的方法。 $X_{ij}$  為決策變數，如公式(5)，當  $X_{ij}$  為 1 的時候，表示會經過節點 i 至節點 j，若  $X_{ij}$  為 0 則否。而公式(2)、(3)限制每一個節點只能被拜訪一次，公式(4)當中的 S 代表避免產生子迴路的限制式(subtour-breaking constraints)，其說明如公式(7)，在解集合 X 當中，每個特定子集合 Q 內的節點，至少要有一個節點與 Q 以外的其它節點相連。

### 2.2 PSO-TS-SA 演算法

在 2008 年，由馮玄明[4]等人提出一種 PSO-TS-SA 演算法。將傳統粒子群最佳化演算法(Particle Swarm Optimization, PSO)[10]，以轉換空間(Transfer Space, TS)[26]處理 PSO 實數編碼問題，並且結合具有跳脫區域最佳解特性的模擬退火演算法(Simulated Annealing, SA)[24]，讓粒子有能力繼續朝向不同區域進行搜尋。本文將以 PSO-TS-SA 演算法，做為 TSP 分群處理後各群聚求解之方法。

### 2.3 模糊分群演算法

本文採用模糊分群演算法(Fuzzy C-means Clustering, FCM)[12]做為分群的方法，FCM 是一種根據 C-means[11][21]演算法所延伸的版本，以模糊邏輯的概念，期望能夠提升資料分群之效果[14][15]。FCM 與 C-means 主要差異是在於，FCM 加入模糊的概念，其資料點 x 不再絕對的屬於任何群聚，則是一個介於 0 至 1 之間的數字，表示 x 隸屬於某個群聚的程度。例如分群數目為 c( $c_1, c_2, \dots, c_c$ )，其資料總共包含 n 個點( $x_1, x_2, \dots, x_n$ )，可透過一個  $c \times n$  的矩陣 U，表示每個資料點隸屬於每個群聚的程度。以其中一點  $x_j$  為例，它隸屬於每一個群聚的程度總和會等於 1，如公式(8)所示。

$$\sum_{i=1}^c u_{ij} = 1, \forall j = 1, 2, \dots, n \quad (8)$$

根據矩陣 U，可將目標函數(Object Function) J，定義成公式(9)。

$$J(U, c_1, c_2, \dots, c_c) = \sum_{i=1}^c J_i \quad (9)$$

$$= \sum_{i=1}^c \sum_{j=1}^n (u_{ij})^m \text{dist}(c_i, x_j)^2$$

其中  $m$  代表的是權重係數，其範圍介於 1 至無窮大，而  $\text{dist}(c_i, x_j)$  為  $c_i$  與  $x_j$  之間的距離函數，一般是採用歐幾里得距離 (Euclidean distance)。

為了符合公式(8)的條件，可依據公式(9)定義新的目標函數  $J_{\text{new}}$ ，如公式(10)。

$$J_{\text{new}}(U, c_1, c_2, \dots, c_c, \lambda_1, \lambda_2, \dots, \lambda_n)$$

$$= J(U, c_1, c_2, \dots, c_c) + \sum_{j=1}^n \lambda_j (\sum_{i=1}^c u_{ij} - 1) \quad (10)$$

$$= \sum_{i=1}^c \sum_{j=1}^n (u_{ij})^m \text{dist}(c_i, x_j)^2 + \sum_{j=1}^n \lambda_j (\sum_{i=1}^c u_{ij} - 1)$$

其中  $\lambda_j$  為相對於公式(8)當中， $n$  組限制的拉格朗日乘數 (Lagrange multipliers)。

為了將  $J_{\text{new}}$  最佳化，針對各個傳入之參數分別進行微分，得到以下公式(11)、公式(12)。

$$c_i = \frac{\sum_{j=1}^n (u_{ij})^m x_j}{\sum_{j=1}^n (u_{ij})^m} \quad (11)$$

$$u_{ij} = \frac{1}{\sum_{k=1}^c \left(\frac{d_{ij}}{d_{kj}}\right)^{\frac{2}{m-1}}} \quad (12)$$

根據上述定義，FCM 的步驟如下[12]：

- step1. 初始化歸屬值矩陣  $U$ ，每行每列的數值範圍介於 0 到 1 之間，且必須滿足公式(8)。
- step2. 根據公式(11)，計算各群之群聚中心  $c_i$ 。
- step3. 根據公式(9)，計算目標函數  $J_{\text{now}}$ 。如果  $J_{\text{now}}$  已經小於某個臨界值 (threshold)，或分群改良效果 ( $J_{\text{pre}} - J_{\text{now}}$ ) 過小，則結束該演算法。
- step4. 根據公式(12)，計算新的矩陣  $U$ ，並且回到 Step2。

## 2.4 連結分群

TSP 節點經過分群之後，將會被拆解成數群，但最終目的是形成一個原 TSP 可行解，因此必須將每一群聚做適當的連結。以姚敦瀚所提出的融合規則，做為本文連結分群之方法，其融合規則如下[1]：

1. 以節點數目最多的群聚定義為主群聚 (Main

Cluster)，且做為融合起始點，而距離主群聚最接近的群聚，會當作下一個融合對象，透過此種方式可避免群聚的跨越式融合。

2. 兩群聚融合連接點為兩群聚距離最短的節點，將融合連接點定義為  $C1\_End\_Node$  與  $C2\_Start\_Node$ 。
3. 融合連接點  $C1\_End\_Node$  及  $C2\_Start\_Node$ ，因為旅行方式可能為順時針或逆時針的走法，而融合連接點有 2 個，共有 4 種排列組合，故選取路徑成本較短的組合作為融合連結點，並且將融合連接點定義為  $C1\_Start\_Node$  與  $C2\_End\_Node$ 。

詳細融合運作過程，如圖 1 所示。其中圖 1(a) 的部分，將節點分為 3 群後，各群聚先透過最佳化演算法進行求解，以節點數目最多之群聚作為主群聚 (Main Cluster)，如  $C1$ 。 $C1$  中的節點 5 與  $C2$  中的節點 8 最為接近，故選擇  $C2$  當作下一個融合的對象。圖 1(b) 的部分，將融合連接點  $C1\_End\_Node$  與  $C2\_Start\_Node$ ，分別定義為節點 5 與節點 8。當  $C1\_End\_Node$ 、 $C2\_Start\_Node$  決定好之後，尚需決定  $C1\_Start\_Node$  與  $C2\_End\_Node$ 。由於  $C1\_End\_Node$  可為順時針或逆時針的旅行順序，故  $C1\_Start\_Node$  有節點 4 與節點 6 可供選擇； $C2\_Start\_Node$  的部分也是如此，而  $C2\_End\_Node$  有節點 7 與節點 9 可供選擇。因此會有 4 種排列組合，分別為：(4, 7)、(4, 9)、(6, 7)、(6, 9)。選取 4 種排列組合當中，距離成本最少的組合，如節點 4 與節點 9 距離最短，則分別定義為  $C1\_Start\_Node$  及  $C2\_End\_Node$ ， $C1$  與  $C2$  融合後之結果如圖 1(c) 所示。以  $C2\_End\_Node$  來判斷下一個進行融合的對象，其中  $C3$  節點 12 最為接近  $C2\_End\_Node$ ，故  $C3\_Start\_Node$  定義為節點 13，同樣的有順時針及逆時針旅行順序問題， $C3\_End\_Node$  有節點 13、節點 16 可供選擇，由於  $C1\_Start\_Node$  已固定，故只有 2 種可能的排列組合：(4, 13)、(4, 16)。以節點 4 與節點 16 距離最短，故將  $C3\_End\_Node$  定義為節點 16。最後融合完成之結果如圖 1(d) 所示。

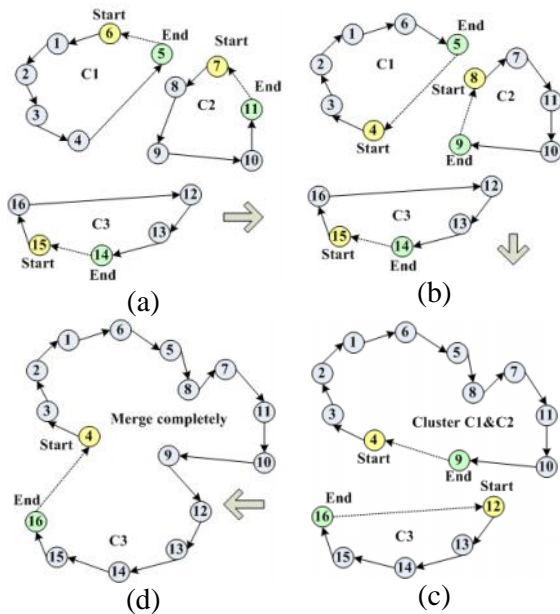


圖 1: 群聚融合運作過程

- 各群以最佳化演算法求解
- 群聚 C1 與 C2 城市節點距離最接近
- 將群聚 C1 與 C2 融合
- 融合完成

## 2.5 連結分群結合區域搜尋

根據姚敦瀚[1]所提出的三項融合規則，以節點數目最多的群聚作為主群聚(Main Cluster)，並且當作起初點，選擇最接近之群聚作為下一個融合對象，如此依序融合可以避免跨越式的融合。但此種作法，當所有群聚皆融合完畢之後，得到的可行解可能會產生旅行路徑交叉之現象。為了盡可能減少旅行路徑交叉的情況，本文以模擬退火演算法針對融合後之可行解進行區域搜尋，試圖取得較佳的可行解，運作過程如下圖 2 所示。

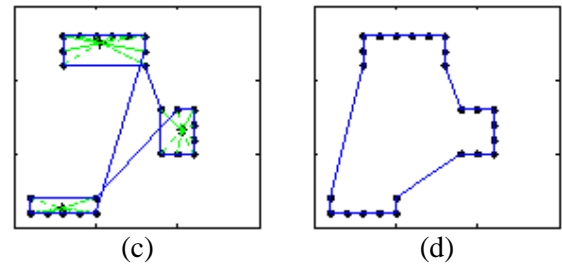
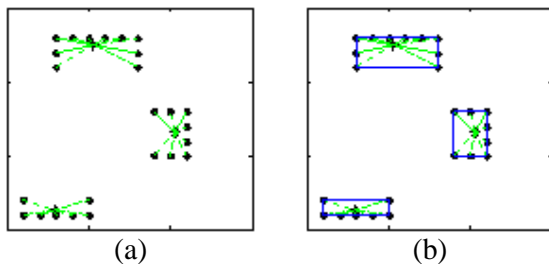


圖 2: 連結分群結合區域搜尋

- 將城市資料分成 3 群
- 各群聚使用 PSO-TS-SA 演算法求解
- 融合完成的距離成本 = 53.3743
- 以 SA 區域搜尋後的距離成本 = 39.3818

## 2.6 PSO-TS-SA & FCM 系統實現流程設計

綜合上述所提到的方法，首先使用模糊分群演算法將 TSP 之節點分群，降低各群聚之搜尋解空間及運算量，接著各群聚以 PSO-TS-SA 演算法進行求解，會形成數個子迴路。再以融合規則將各群聚逐一連結，直到融合完成後會形成一個完整 TSP 可行解。為了減少融合後路徑交叉的現象，會透過模擬退火演算法進行區域搜尋，試圖改善融合後之路徑以取得更佳的可行解。故本文整合上述之方法，提出一種新的方法，稱為 PSO-TS-SA & FCM 演算法。整個演算法運作的流程，主要可分為三個步驟：分群(Clustering)、融合(Merge)、區域搜尋(Local Search)。

PSO-TS-SA & FCM 詳細運作流程與步驟說明如下：

- step1. 設定群聚數目，以模糊分群演算法(FCM)針對城市節點進行分群，將群聚定義為 C1, C2, ..., CC。
- step2. 各群聚透過 PSO-TS-SA 演算法求解，並且將求得之旅行路徑定義為 C1\_Path, C2\_Path, ..., CC\_Path。
- step3. 以城市節點數目最多的群聚作為主群聚(Main Cluster)及融合起始點。
- step4. 與主群聚最為接近的群聚，當作下一個融合對象。假設主群聚為 C1，且 C1 與 C2 的城市節點最為接近，則將 C1\_Path 和 C2\_Path 的旅行路徑順序進行調整，然後定義融合連接點為 C1\_End\_Node 與 C2\_Start\_Node。

- step5. 當 C1\_End\_Node 和 C2\_Start\_Node 確定後，以四種交叉排列組合，選取距離成本最少者作為 C1\_Start\_Node 與 C2\_End\_Node，接著將 C1 與 C2 融合成為一條旅行路徑，定義為 Cluster C1 & C2。
- step6. 判斷所有群聚是否皆已融合完成，如果已經融合完成，將 Cluster C1 & C2 定義為 Path，且跳至 step10；否則，進入 step7 繼續融合的工作。
- step7. 以 C2\_End\_Node 連接到其它尚未融合的群聚。假設 C2\_End\_Node 最為接近 C3 的城市節點，則將 C3\_Path 之旅行順序進行調整，並且定義融合連接點為 C3\_Start\_Node。
- step8. 當 C3\_Start\_Node 確定後，尚需決定 C3\_End\_Node。由於主群聚 C1\_Start\_Node 已確定，故只剩下兩種排列組合，同樣地選取距離成本最少者作為 C3\_End\_Node，接著將 Cluster C1 & C2 與 C3 融合成為一條旅行路徑，定義為 Cluster C1 & C2 & C3。
- step9. 判斷所有群聚是否皆已融合完成，如果已經融合完成，將 Cluster C1 & C2 & C3 定義為 Path，且進入 step10；否則跳至 step7，改用 C3\_End\_Node 連接到其它尚未融合的群聚，繼續融合的工作。
- step10. 當所有群聚皆已融合完成後會產生一條旅行路徑 Path，再以模擬退火進行區域搜尋，試圖減少旅行路徑交叉現象，力求較佳之可行解，經過模擬退火後的旅行路徑定義為 Use\_SA\_Path。
- step11. 判斷 Use\_SA\_Path 是否小於 Path，如果是，代表經過模擬退火區域搜尋後旅行路徑有獲得改善，並且將 Use\_SA\_Path 指定為 Final\_Path；否則將 Path 指定為 Final\_Path。
- step12. 輸出最終旅行路徑 Final\_Path，本演算法運作結束。

### 3. 實驗

#### 3.1 實驗說明

本文實驗測試所使用的資料，取自於

TSPLIB[7]中的城市坐標點資料，分別為 Pr76、Ch130、Pcb442 三個檔案。電腦硬體方面，採用 Intel Celeron 1.8GHz 中央處理器，主記憶體為 1GB；電腦軟體方面，程式開發工具採用 Matlab R2007b。首先測試 PSO-TS-SA 演算法結合模糊分群之求解能力，並與相關文獻進行效能比較。

#### 3.2 實驗參數設定

表 1、表 2 為 PSO-TS-SA 的參數設定，測試三種城市坐標資料，皆使用相同的參數。

表 1 粒子相關參數設定

粒子相關參數設定：			
粒子數目 P	50	迭代次數 I	200
慣性權重 W	1	慣性權重遞減率 e	0.7
最大速度 V <sub>max</sub>	10	最小速度 V <sub>min</sub>	-10
學習因子 C <sub>1</sub>	2	學習因子 C <sub>2</sub>	2
隨機亂數 R <sub>1</sub>	[0,1]	隨機亂數 R <sub>2</sub>	[0,1]

表 2 模擬退火相關參數設定

模擬退火相關參數設定：	
每一溫度下運算次數 MaxTrialN	NumCity * 5
每一溫度下接受次數 MaxAcceptN	NumCity * 0.5
初始溫度 T	10
停止溫度 Stop_T	0.005
冷卻率 r	0.5
註：NumCity 為城市數量，分別為 76、130、442	

#### 3.3 實驗結果與分析

將群聚數目分為 1、2、4、8 群，本文所提出之 PSO-TS-SA & FCM 演算法求解能力如表 3 所示。欄位 NumCluster 代表群聚數目，Time 為運算所需時間，Err 為誤差率 %。

表 3 PSO-TS-SA & FCM 之求解能力分析

	Algorithms	NumCluster	Time(s)	Err(%)
Pr76	PSO-TS-SA	1	29.438	7.37
	PSO-TS-SA & FCM	2	9.531	5.11
		4	9.422	12.08
		8	7.531	19.34

Ch130	PSO-TS-SA	1	39.859	6.40
	PSO-TS-SA	2	23.281	4.97
	& FCM	4	26.125	11.31
Pcb442	PSO-TS-SA	8	26.234	19.19
	PSO-TS-SA	1	103.297	14.59
	PSO-TS-SA	2	66.734	13.69
	& FCM	4	57.641	11.08
		8	53.813	19.54

從表 3 可得知，相較於單群，透過分群的方式進行求解，其運算時間會有明顯縮減。如 Pr76 問題，以單群求解大約需要 30 秒左右的時間，而當群聚分為兩群時，運算時間縮減至約 10 秒左右。實驗數據中可得知，求解 Pr76 與 Ch130 問題，當分為 2 群時，其誤差率相較於單群時來得低，而 Pcb442 問題，則是以分成 4 群時，誤差率相對較低。在求解過程不一定會隨著分群數目之增加，而使得求解誤差率隨之下降。

針對 Ch130 問題以本文所提方式分為 2 群，各重要步驟模擬後結果，分別展現在圖 3 至圖 6 所示。首先將節點數目分成 2 群，如圖 3；接著各群聚先以 PSO-TS-SA 演算法進行求解，如圖 4；將 2 群聚融合在一起，如圖 5；最後再以模擬退火進行區域搜尋，如圖 6。

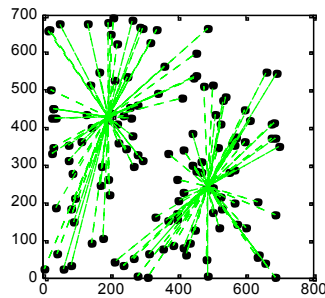


圖 3: Ch130 分為 2 群

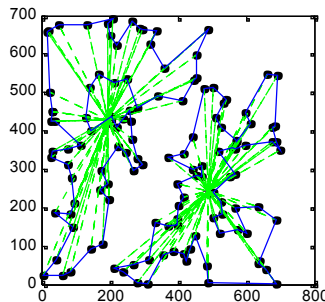


圖 4: Ch130 每群分別使用 PSO-TS-SA 求解

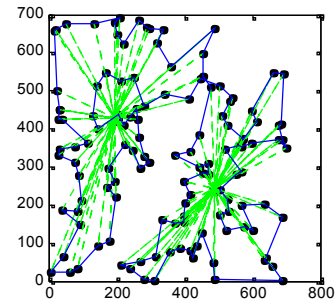


圖 5: Ch130 融合結果(距離=6426.9651)

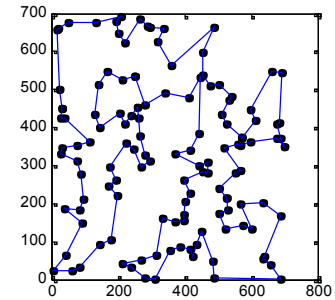


圖 6: Ch130 使用 SA 改善(距離=6413.7616)

表 4 為 PSO-TS-SA & FCM 與姚敦瀚[1]所提出的 Improved PSO & FCM 以及陳彥霖[3]所提出的 IPSO & Fuzzy ART，針對求解 Pr76、Ch130、Pcb442 問題其誤差率之比較。在 Pr76 問題的部分，以 Improved PSO & FCM 分為 2 群時誤差率較低。而 Ch130 問題，則是以 PSO-TS-SA & FCM 當群聚數目為 2 時誤差率相對較低。在 Pcb442 問題，PSO-TS-SA & FCM 以分成 4 群時，求解誤差率相較於 IPSO & Fuzzy ART 分為至 6 群時來得低。

表 4 PSO-TS-SA & FCM 與相關文獻效能比較

	Algorithms	NumCluster	Err(%)
Pr76	PSO-TS-SA	1	7.37
	PSO-TS-SA & Fuzzy C-means	2	5.11
	Improved PSO & Fuzzy C-means	2	4.19
Ch130	PSO-TS-SA	1	6.40
	PSO-TS-SA & Fuzzy C-means	2	4.97



	Improved PSO & Fuzzy C-means	5	7.53
Pcb442	PSO-TS-SA	1	14.59
	PSO-TS-SA & Fuzzy C-means	4	11.08
	IPSO & Fuzzy ART	6	11.57

#### 4. 結論

實驗結果顯示，在求解 Ch130 和 Pcb442 問題部分，相較於 Improved PSO & FCM、IPSO & Fuzzy ART，本文所提出的 PSO-TS-SA & FCM 演算法擁有相對較佳之求解能力。城市節點數量若依本文分成若干群，其運算時間會因分群事先處理而有明顯縮減，但是求解誤差率並不一定會隨著群聚數目的增加而降低。如何合適分群數將是未來極可能影響解誤差率重要因子，值得再三研究。

經過本文實驗證實，適當的分群有助於增加最佳化演算法之求解效益，降低求解誤差率、運算時間複雜度。

#### 參考文獻

- [1] 姚敦瀚，粒子尋優法效能改進及其在旅行商問題之應用，碩士論文，國立成功大學航空太空工程學系碩博士班，台南，2008。
- [2] 陳建緯，大規模旅行推銷員問題之研究：鄰域搜尋法與巨集啟發式解法之應用，碩士論文，國立交通大學運輸工程與管理系，新竹，2001。
- [3] 陳彥霖，粒子群最佳化演算法求解旅行者推銷員問題，碩士論文，國立台北科技大學工業工程與管理研究所，台北，2007。
- [4] 馮玄明、廖國隆，「結合模擬退火之粒子群最佳化演算法解決旅行銷售員問題」，第十三屆人工智慧與應用研討會(TAAI2008)，宜蘭，2008。
- [5] A. Schrijver, K. Aardal, G.L. Nemhauser, and R. Weismantel, "On the History of Combinatorial Optimization (till 1960)," in Handbook of

Discrete Optimization, ed, pp. 1-68, 2005.

- [6] C. Ding, Y. Cheng, and M. He, "Two-Level Genetic Algorithm for Clustered Traveling Salesman Problem with Application in Large-Scale TSPs," Tsinghua Science & Technology, vol. 12, pp. 459-465, 2007.
- [7] G. Reinelt. TSPLIB. Available: <http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/tsp/>
- [8] H.H. Zuo and F.L. Xiong, "Novel ant colony algorithm based on local optima clustering and its application in Chinese traveling salesman problem," Tongji Daxue Xuebao/Journal of Tongji University, vol. 32, pp. 142-144, 2004.
- [9] H.P. Christos and S. Kenneth, Combinatorial optimization: algorithms and complexity: Prentice-Hall, Inc., 1982.
- [10] J. Kennedy and R. Eberhart, "Particle swarm optimization," in Proceedings of IEEE International Conference on Neural Networks, USA, pp. 1942-1948, 1995.
- [11] J. Makhoul, S. Roucos, and H. Gish, "Vector quantization in speech coding," Proceedings of the IEEE, vol. 73, pp. 1551-1588, 1985.
- [12] J.C. Bezdek, "Fuzzy Mathematics in Pattern Classification," PhD thesis, Applied Math. Center, Cornell University, Ithaca, 1973.
- [13] J.H. Holland, Adaptation in Natural and Artificial Systems. Ann Arbor: University of Michigan, 1975.
- [14] J.-S.R. Jang, C.-T. Sun, and E. Mizutani, Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence: MATLAB Curriculum Series, Upper Saddle River, NJ, Prentice Hall, 1997.
- [15] J.-S.R. Jang. Data Clustering and Pattern Recognition. Available: <http://neural.cs.nthu.edu.tw/jang/books/dcpr/>
- [16] L. Bodin, B. Golden, A. Assad, and M. Ball, "Routing and Scheduling of Vehicles and Crews: The State of the Art," Computers & Operations Research, vol. 10, pp. 63-211, 1983.
- [17] L. Chih-Chung and D. Guang-Feng, "Using Ant Colony Optimization Algorithm to Solve Airline Crew Scheduling Problems," in Natural

- Computation, 2007. ICNC 2007. Third International Conference on, pp. 797-804, 2007.
- [18] M. Dorigo, V. Maniezzo, and A. Colomi, "Ant system: optimization by a colony of cooperating agents," *Proceedings of IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 26, pp. 29-41, February 1996.
- [19] M.S. Zaeri, J. Shahrabi, M. Pariazar, and A. Morabbi, "A combined spatial cluster analysis - traveling salesman problem approach in location-routing problem: A case study in Iran," in *IEEE International Conference on Industrial Engineering and Engineering Management*, pp. 1599-1602, 2007.
- [20] P. Johanns, T. Lowe, and R. Plante, "Selection and sequencing heuristics to reduce variance in gas turbine engine nozzle assemblies," *European Journal of Operational Research*, vol. 132, pp. 490-504, 2001.
- [21] P.R. Krishnaiah and L.N. Kanal, *Classification, pattern recognition, and reduction of dimensionality*. Amsterdam; New York; New York, N.Y.: North-Holland Pub. Co.; Sole distributors for the U.S.A. and Canada, Elsevier Science Pub. Co., 1982.
- [22] R. Balakrishnan and K. Ranganathan, *A textbook of graph theory*. New York: Springer, 2000.
- [23] S. Hussain Aziz and D. Peter, "Effective Heuristics for the GPS Survey Network of Malta: Simulated Annealing and Tabu Search Techniques," *Journal of Heuristics*, vol. 7, pp. 533-549, 2001.
- [24] S. Kirkpatrick, C.D. Gelatt, and M.P. Vecchi, "Optimization by Simulated Annealing," *Science*, vol. 220, pp. 671-680, May 1983.
- [25] S.A. Mulder and D.C. Wunsch, "Using adaptive resonance theory and local optimization to divide and conquer large scale traveling salesman problems," in *Proceedings of the International Joint Conference on Neural Networks, USA*, pp. 1408-1411, 2003.
- [26] W. Pang, K.-P. Wang, C.-G. Zhou, L.-J. Dong, M. Liu, H.-Y. Zhang, and J.-Y. Wang, "Modified particle swarm optimization based on space transformation for solving traveling salesman problem," in *Proceedings of 2004 International Conference on Machine Learning and Cybernetics, China*, pp. 2342-2346, 2004.