

演化式群蟻最佳化演算法應用於 大型旅行銷售員問題

劉振隆	杜宜蓉	蕭天輝	邱泰毓
義守大學資訊管理 學系教授 jlliu@isu.edu.tw	立德管理學院應用 資訊研究所研究生 an850.du@msa.hinet.net	立德管理學院行銷 管理學系講師 sth@mail.leader.edu.tw	義守大學資訊管理 學系碩士班研究生 caster5210@hotmail.com

摘要

本文提出一種演化式群蟻最佳化演算法，利用基因演算法求解群蟻最佳化演算法之最佳化參數組合，並導入群蟻最佳化演算法中應用於大型旅行銷售員問題，期能得到最佳的路徑搜尋結果。首先，以群蟻最佳化演算法計算基因演算法所需之環境適應值，然後執行基因演算法最佳化過程後決定出群蟻最佳化演算法之最佳參數組合。計有 10 個參數，包含螞蟻數量 m 、利用與探勘機率之權衡比重 q_0 、費洛蒙揮發參數 α 、距離參數 β 、區域更新參數 ρ_{local} 、全域更新參數 ρ_{global} 、相關費洛蒙之一個固定參數值 Q 、候選名單數 l 、疊代數 h 以及重複運算的次數 i 等。分析從 100、200 與 225 個節點數的大型旅行銷售員標準案例，經由演化群蟻最佳化演算法後歸納出一組最佳化參數組合，並應用於超過 300 節點數的大型旅行銷售員問題上。

關鍵詞：演化式群蟻最佳化演算法、基因演算法、最佳化參數組合、大型旅行銷售員問題。

Abstract

This study proposes an evolving ant colony optimization (EACO), which employs a genetic algorithm (GA) to find the best set of parameters employed in the ant colony optimization (ACO), and apply to large traveling salesman problems (TSP) for the purpose of obtaining optimal searching tours. There are ten designed parameters include the number of ant m , three weighting factors q_0 , α and β , local and global evaporation coefficients ρ_{local} and ρ_{global} , parameter of pheromone Q , number of table list l , number of iteration h , and repeated run number i . The benchmark cases of large traveling salesman problem (TSP) with

100, 200 and 225 nodes are computed to deduce a set of optimal parameters through the evolving ACO. Moreover, the suggested parameter set obtained from the evolving ACO is applied to large traveling salesman problems with over 300 nodes.

Keywords: Evolving ant colony optimization, genetic algorithm, best set of parameters, large traveling salesman problems.

1. 簡介

基因演算法(Genetic Algorithm, GA)與群蟻最佳化演算法(Ant Colony Optimization, ACO)都是屬於啟發式演算法(Heuristic Algorithm)，其中群蟻最佳化演算法是由真實螞蟻的生物行為啟發而來的，目前此演算法在達成求解推論組合最佳化上已成為一種高性能之可行方法。雖然群蟻最佳化演算法在求解分散式組合最佳化問題上是相當有效的，但這種演算法在搜尋過程含有許多影響演算法性能的隱式和顯式參數。因此，群蟻最佳化演算法若能配合適當挑選出來的合適參數將可獲得優越的計算效能[1]。

群蟻最佳化演算法，是由 Dorigo *et al.* 在 1996 年提出的一種啟發式演算法(Heuristic Method) [2-5]，主要應用於搜尋最短路徑之問題上，啟發方式則是模擬自然界中群蟻在覓食(Foraging)時為有效搜尋路徑所產生的溝通合作模式而成。螞蟻移動時會分泌一種稱為「費洛蒙」(Pheromone)之化學物質拖曳物(Trails)，將使螞蟻選擇要走某一路徑的機率與該路徑上已遺留之費洛蒙量濃度成正比。換句話說，越多螞蟻走過同一條路時所遺留下之費洛蒙的量將會增加，而此堆積越多費洛蒙的路徑上又會吸引越多的螞蟻行走這條路徑，到最後大部分的螞蟻都會選擇走這同一條路徑，形成正向回饋(Positive Feedback)的強化學習(Reinforcement Learning)模式。因此，以這個模式來看，比較短的路徑所需行經花費的時間

比較短，很容易累積比較多之費洛蒙量，因而吸引比較多的螞蟻，最後大部分的螞蟻將沿著最短路徑行走，據此演算法即可求得最佳解。

如圖1所示，如果在一群螞蟻行經的路程上置放一個障礙物形成原行走路徑分岔為二條路線可供行走時，螞蟻面臨到要選擇其中一條路線的機率和遺留費洛蒙量是有非常密切的關係，螞蟻在越短路線所通過的時間較短，來回行走此路徑的螞蟻數量就會增多，導致所殘留的費洛蒙量是相對的高，最後將讓更多的螞蟻選擇行走此較短的路線。

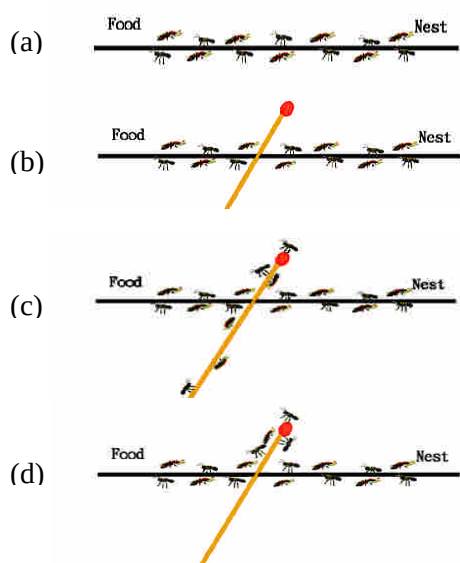


圖 1 螞蟻由巢穴搜尋食物之示意圖，(a)沒有障礙物原始行走路徑，圖(b)置放一個障礙物，圖(c)螞蟻開始搜尋最佳路徑，圖(d)最佳路徑搜尋完成

在圖 1(a)中，螞蟻由巢穴(右邊)螞蟻窩出發往左邊覓食，剛開始眾多螞蟻沿著直線行進。圖 1(b) 顯示往食物的路上置放一個障礙物(Obstacle)，造成岔路的情況發生，亦造成有二條不等距離的路線。剛開始時，螞蟻對於選擇這兩條路徑行走的機率是相同的，但選擇往上方路線(即較短路徑)之螞蟻可以比較快速通過障礙物到達食物的所在地。圖 1(c)顯示出因為障礙物的阻擋導致路徑長短不一致，所以選擇往上方路線的螞蟻來通過障礙物並取得食物往回走到分岔路時的數量較多，所留下的費洛蒙也比較多；反之，往下方路線因為行經此路徑的螞蟻比較少，故所留下的費洛蒙量也比較少。在正向回饋模式作用下，經過多次疊代後，大部分的螞蟻將趨向選擇往上方最短路線行走(如圖 1(d))。因此，群蟻最佳化演算法的優點為(1)確實回饋：能夠快速地發現新

的起始解，(2)分散式計算：亦就是所謂的多點搜尋，期避免過早收斂(Premature Convergence)，(3)使用積極的貪婪法則(Greedy Method)：能在發展起始解的時候，較早發現可接受解(Feasible Solution)。因此，目前群蟻最佳化演算法最常被使用來搜尋最佳路徑(或網路路由)的問題上。然而，雖然群蟻最佳化演算法展現了良好的搜尋最佳路徑的能力，但是此演算法的參數在設定上容易影響最佳解的正確性，亦有可能衍生出造成過早收斂及演算時間過長等問題。

為求得群蟻最佳化演算法之最佳參數組合，可行的方式為將原始參數作為變數，結合基因演算法作為最佳化解子(Optimizer)來求出最佳參數值。基因演算法[6-11]是由 John Holland 在 1970 年提出，採用達爾文的「進化論」為主，以「物競天擇，適者生存」(Survival Theory)法則，模擬自然界中生物的進化過程，在進化的過程中將不適合在競爭環境生存的生物個體淘汰，留存具有較佳環境適應值(Environmental Fitness)的生物個體。Goldberg[8]利用電腦的模擬方式是以隨機產生初始族群(Population)，再以物競天擇法則來淘汰不好的物種(Individual)，而留下適合的物種，再經由搜尋的方式來產生下一代新的族群。它的搜尋方式主要是以選擇(Selection)、交配(Crossover)與突變(Mutation)三個操作子(Operator)以及採用精英策略(Elitist Strategy)來作為搜尋問題空間的依循原則，產生新的下一代，據此一直反覆進行演化將可得到適應環境的下一代。因此，基因演算法能夠避免在最佳化過程中陷入區域最佳解，進而具有搜尋全域最佳解的能力。

本文之研究目地為結合群蟻最佳化演算法於基因演算法內，以群蟻最佳化演算法提供環境適應值的計算，利用基因演算法可在全域產生出最佳參數的特性，將求解出來之最佳化參數導入群蟻最佳化演算法上，此方法稱之為演化式群蟻最佳化演算法(Evolving Ant Colony Optimization, EACO)[1]。此方法之建立期能在短時間內快速搜尋最佳路徑，進而增進群蟻最佳化演算法以更高的效率搜尋出最佳路徑，本文將以大型旅行銷售員問題[12, 13]作為演算法之驗證實例，以確定本文方法之準確性與可靠性。

2. 研究方法

2.1 群蟻最佳化演算法之流程與結構

群蟻最佳化演算法其演算結構以及演算虛擬程式碼如下：

Pseudo-code: ACO algorithm

```

Begin
  While (ACO has not been stopped) do
    Ants generation ();
    Ants activity();
    Pheromone evaporation();
    Elitist ant chosen();
    Best path search();
  End;
End;

```

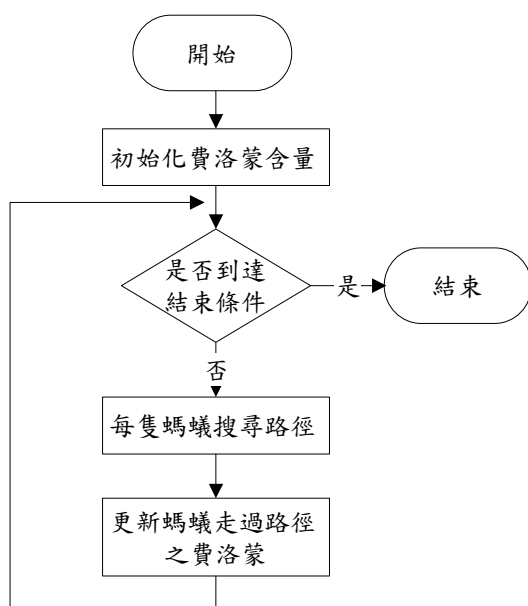


圖 2 群蟻最佳化演算法整體流程圖

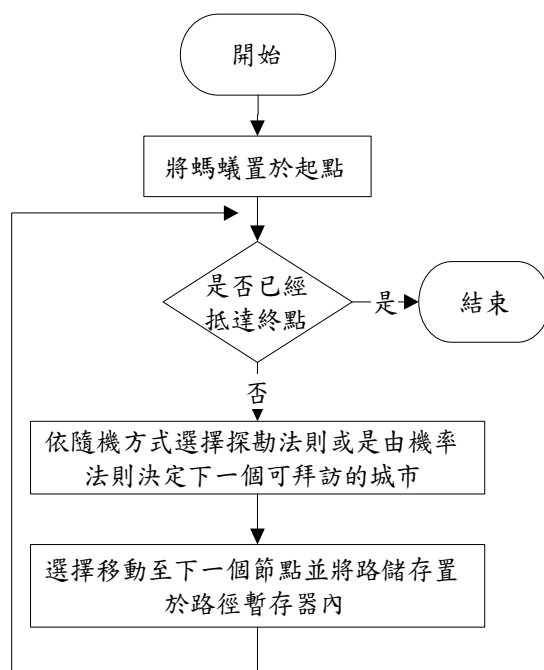


圖 3 群蟻最佳化演算法中單一螞蟻演算流程

如圖 2 與圖 3 所示之以群蟻最佳化演算法求解最佳化問題的過程中，整個演算法的終止條件(Stopping Criteria)與限制條件如下：

- (1) 執行固定的疊代次數：不管搜尋結果如何，演算法則執行固定的回合數後運算停止。
- (2) 限制螞蟻又走回到已拜訪過的節點。
- (3) 如果螞蟻已經處於「邊緣點」，即代表已經沒有任何相鄰的節點可拜訪，就必須強迫此隻螞蟻結束其拜訪工作。

另外，其他相關顯著影響群蟻最佳化演算法之因素表示如下：

- (1) 整個系統的螞蟻總數量。
- (2) 費洛蒙揮發率(ρ)。
- (3) 拜訪路徑的表示方式。
- (4) 其他作為考量選擇機率的啟發函數(如距離)。
- (5) 影響選擇機率的因素間的關係比值(使用者自訂之控制參數 α 與 β)。

2.1.1 機率法則

在螞蟻進行尋找最佳路徑時，係利用一隨機機率進行下一節點的選擇，Dorigo *et al.* 考慮到此機率與費洛蒙強度以及兩節點 r 與 s 間距離的倒數(稱為可見度 η ，Visibility)相關，其設定探勘(Exploration)機率方程式表示如下：

$$P_k(r, s) = \begin{cases} \frac{[\tau(r, s)]^\alpha \cdot [\eta(r, s)]^\beta}{\sum_{u \in T_k(r)} [\tau(r, u)]^\alpha \cdot [\eta(r, u)]^\beta}, & \text{if } s \in T_k(r) \\ 0, & \text{else} \end{cases} \quad (1)$$

方程式(1)中的 $P_k(r, s)$ 為螞蟻 k 在節點 r 選擇節點 s 做為下一個將拜訪節點的機率，其中 $s \in T_{k(r)}$ 為 s 與 r 有相連而且還未拜訪過的節點集合。 $\tau(r, s)$ 為目前分佈在節點 r 至節點 s 路徑上的費洛蒙含量， $\eta(r, s)$ 為節點 r 與 s 間距離的倒數， α 與 β 用以決定費洛蒙濃度與距離之間的相對權重值。

2.1.2 費洛蒙更新

螞蟻在行經的路徑上會留下費洛蒙，稱為費洛蒙拖曳量(Pheromone Trail)，在有高含量費洛蒙拖曳量的路徑上會與沒有螞蟻走過或費洛蒙已揮發完的路徑有明顯區隔。各節線費洛蒙的更新方式如下：

$$\tau_{rs}(t+1) = (1 - \rho) \cdot \tau_{rs} + \sum_{k=1}^k \Delta \tau_{rs}^k \quad (2)$$

其中 $\tau_{rs}(t)$ 及 $\tau_{rs}(t+1)$ 為在時間 t 至 $t+1$ 時，節點 r 到節點 s 間路徑上的費洛蒙分佈值。

$$\Delta\tau_{rs}^k = \begin{cases} Q/L^k & , \text{if } r \text{ and } s \in L^k \\ 0 & , 0 \end{cases} \quad (3)$$

其中 L^k 為螞蟻 k 所完成的路徑總長， Q 為一固定參數值，一般設定為 100， ρ 為費洛蒙揮發度，其中 $0 < \rho < 1$ 。

2.1.3 利用法則

群蟻最佳化演算法為改良群蟻系統(Ant System)之搜尋效能，增加了利用(Exploitation)機制，其表示式如下：

$$s = \begin{cases} \arg \max\{[\tau(r,u)]^\alpha \cdot [\eta(r,u)]^\beta\} & , \text{if } q \leq q_0 \\ S & , \text{else} \end{cases} \quad (4)$$

其中 $\tau(r,u)$ 是 r 節點至 u 節點路徑上的費洛蒙含量， $\eta(r,u)$ 是 r 節點與 u 節點路徑長度的倒數， α 與 β 是關係著 $\tau(r,u)$ 以及 $\eta(r,u)$ 的權重值。 q 是一個常態分佈於 $[0,1]$ 之間的隨機變數， q_0 是一個參數且 $0 < q_0 < 1$ ，一般設定值為 0.9，而 S 則是依據機率公式(1)選擇下一個所要拜訪的節點。依據上式，若隨機亂數值 $q > q_0$ ，此為探勘法則(Exploration Rule)。若 $q \leq q_0$ 則直接選取具有最大吸引力的節點(費洛蒙濃度高、距離長度短)，故其行為較偏向利用法則(Exploitation Rule)。

2.1.4 加入區域更新法則

只要有螞蟻走過的路徑，即會改變路徑上的費洛蒙量；也就是假設當螞蟻 k 從節點 r 移動至節點 s 時就需要使用「區域更新法則(Local Updating Rule)」來對此路徑上的費洛蒙含量進行更新。區域更新的主要精神在於避免產生一個強勢的路徑，吸引所有的螞蟻選擇該路徑，無法進行適當的探勘新路徑動作，而影響所得解答之精確性。

$$\tau(r,s) \leftarrow (1-\rho) \cdot \tau(r,s) + \rho \cdot \Delta\tau(r,s) \quad (5)$$

其中 $\Delta\tau(r,s) = \tau_0$ ， τ_0 是原始費洛蒙濃度，通常設定為： $\tau_0 = (NL_{mn})^{-1}$ 其中 N 為節點總數； $0 < \rho < 1$ 是費洛蒙揮發參數。

2.1.5 更新費洛蒙之全域更新法則

針對每次疊代數取得最佳解答的路徑改變其費洛蒙量；也就是指所有人工螞蟻各自完成拜訪其路徑後，則須藉由「全域更新法則(Global Updating Rule)」來進行最佳路徑上之費洛蒙含量更新。全域更新的精神則在於對好的解答給予獎賞，以引導螞蟻依據這個路徑進行利用及近一步的探勘其他新路徑。

$$\tau(r,s) \leftarrow (1-\alpha) \cdot \tau(r,s) + \alpha \cdot \Delta\tau(r,s) \quad (6)$$

其中

$$\Delta\tau_k(r,s) = \begin{cases} (L_{gb})^{-1} & , \text{if } (r,s) \in \text{global best tour} \\ 0 & , \text{else} \end{cases}$$

式中 α 為費洛蒙揮發參數(與 ρ 相同)，而 L_{gb} 為該疊代中最佳路徑長度。

2.2 基因演算法之流程

圖 4 為基因演算法流程，一開始先設定初始參數，如個體數(Number of Individual)、交配率(Probability of Crossover)、突變率(Probability of Mutation)及演化代數(Number of Evolutionary Generation)等。再隨機產生初始的族群並且根據問題的特性產生合適函數(Fitness Function)分別計算出初始值的環境適應值(Fitness Value)，繼而依序執行選擇、交配及突變運算以及精英策略(Elitist Strategy)，最後，再判斷是否已達到終止條件。說明如下：

(1)隨機產生初始值

- (i) 個體數：同一時間在解空間搜尋的個體數量。
- (ii) 交配率：預期每代做交配運算的個體數目。
- (iii) 突變率：預期每代會做突變運算的基因數目。

(2)計算合適值：計算每個個體的適應值。

(3)三個運算子：

- (i) 選擇：選擇運算子是依據個體的適應值決定被複製到下一代的機率。
- (ii) 交配：利用目前所擁有的個體找到更好的個體，其隨機選擇兩個個體並隨機決定切點做基因的交換。
- (iii) 突變：使有機會去探索搜尋空間，在預設參數時必要注意不宜過高，以免過早收斂。

(4) 精英策略：複製最佳個體(Best Individual)，確保此個體能在下一世代中繼續存活。

- (5) 終止條件：判斷是否已達到了預設之疊代數，若條件滿足則停止運算。詳細之流程圖，如下圖4所示：

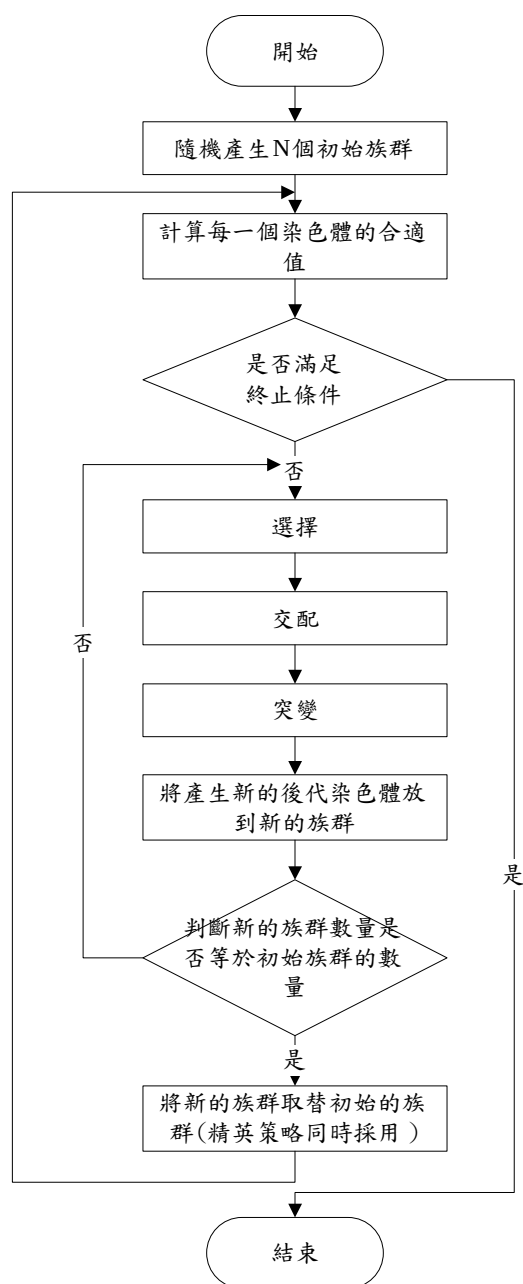


圖 4 基因演算法流程圖

2.3 以基因演算法演化群蟻最佳化演算法參數之個體編碼方式及流程

群蟻最佳化演算法目前已成功地應用於旅行銷售員問題上，並證實所找出的最佳路徑結果優於其他已知之搜尋法。由於群蟻最佳化演算法的參數無法精確掌控，同時群蟻最佳化演算法有許多的參數，求解值及求解性能會隨著參數的不同而有所不同，所以參數設定是非

常重要的。本文計分析 10 個參數，分別為： m 代表是螞蟻數量； q_0 表示使用利用機率與探勘機率之權衡比重。 α 代表是費洛蒙揮發權衡參數。 β 代表是距離權衡參數。 ρ_{local} 是代表區域更新的參數。 ρ_{global} 是代表全域更新的參數。 Q 是代表方程式(3)之固定參數值， l 是代表候選名單數(Number of Candidate List)， h 是代表疊代數，而 i 是代表 Run 的次數， n 代表節點數。

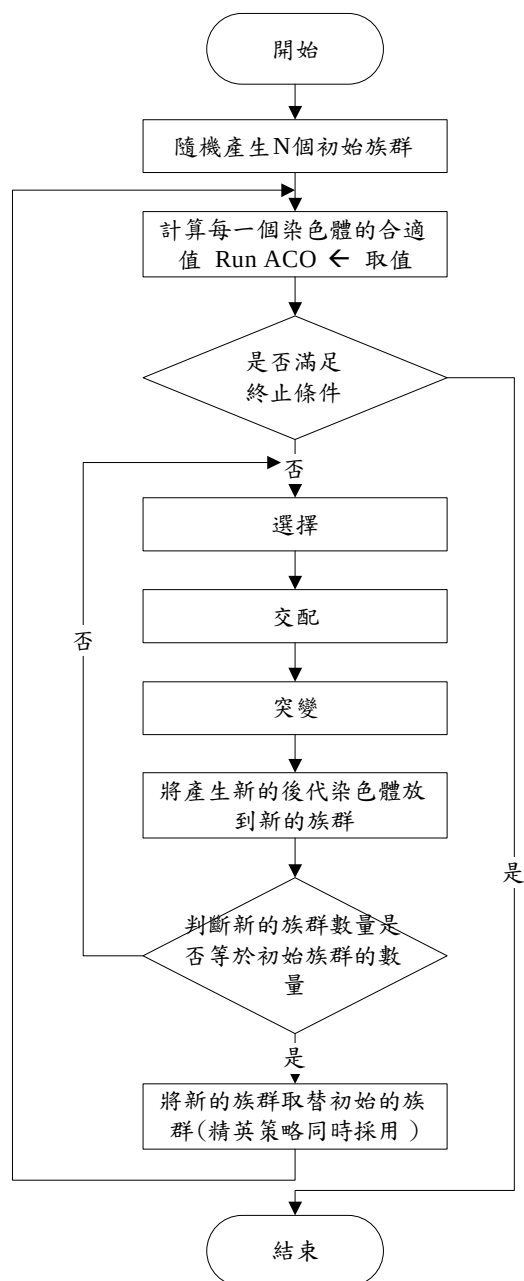


圖 5 基因演算法演化群蟻最佳化演算法參數流程圖

本文使用個體編碼方式(Coding Type of

Individual)，是以實數型基因演算法 (Real-Valued GA, RGA)。每一條完整染色體直接以問題的變數值組成，不需要再轉換成二進位。一般而言，實數型基因演算法在處理高維度和需要高精確度的數值問題時會比二進位編碼方式(Binary Coding)來的有效率，很適合處理連續型變數問題。基因演算法演化群蟻最佳化演算法參數流程圖，如圖 5 所示。由流程圖可看出基因演算法所需之環境適應值是由群蟻最佳化演算法求解提供。

2.4 基因演算法之適應值函數

本文基因演算法之目標函數(Objective Function) F 定義如下：

$$\text{Maximize } F = 1/L \quad (7)$$

L 代表群蟻最佳化演算法中螞蟻族群所行經之完整路徑距離。在本文中基因演算法中使用 10 個群蟻最佳化演算法中所使用之參數作為設計變數，分別為螞蟻數量 m 、利用與探勘機率之權衡比重 q_0 、費洛蒙揮發參數 α 、距離參數 β 、區域更新參數 ρ_{local} 、全域更新參數 ρ_{global} 、相關費洛蒙之一個固定參數值 Q 、候選名單數 l 、疊代數 h 以及重複運算的次數 i 等。這 10 個設計變數之設計範圍，如表 1 所示。

表 1 演化參數及參數之設計範圍

Parameter 1	$1 \leq m \leq 2*n$
Parameter 2	$0.0 \leq q_0 \leq 1.0$
Parameter 3	$0.0 \leq \alpha \leq 5.0$
Parameter 4	$0.0 \leq \beta \leq 10.0$
Parameter 5	$0.0 \leq \rho_{local} \leq 1.0$
Parameter 6	$0.0 \leq \rho_{global} \leq 1.0$
Parameter 7	$0.0 \leq Q \leq 100.0$
Parameter 8	$0 \leq l \leq 9$
Parameter 9	$1 \leq h \leq 500$
Parameter 10	$1 \leq i \leq 10$

3. 實驗結果

3.1 最佳參數組合分析

本文採 TSPLIB 題庫國際例題 [13] KroA100、KroA200 及 TSP225 等大型旅行銷售員問題作為測試例，供推論出 10 個參數值的選定參考資料，並將演化式群蟻最佳化演算法及傳統群蟻最佳化演算法的計算效能互做比較。傳統群蟻最佳化演算法的參數值(如表 2)以及經由演化式群蟻最佳化演算法分析三個案例後之最佳各參數值(表 3-5)表列如下：

表 2 傳統群蟻最佳化演算法參數值 (KroA100、KroA200 及 TSP225)

Parameter 1	$m=10$
Parameter 2	$q_0=0.9$
Parameter 3	$\alpha=1.0$
Parameter 4	$\beta=2.0$
Parameter 5	$\rho_{local}=0.1$
Parameter 6	$\rho_{global}=0.1$
Parameter 7	$Q=100.0$
Parameter 8	$l=20$
Parameter 9	$h=1000$
Parameter 10	$i=5$

表 3 演化式群蟻最佳化演算法參數值 (KroA100)

Parameter 1	$m=19$
Parameter 2	$q_0=0.43938$
Parameter 3	$\alpha=2.87592$
Parameter 4	$\beta=6.76445$
Parameter 5	$\rho_{local}=0.35363$
Parameter 6	$\rho_{global}=0.84680$
Parameter 7	$Q=51.64027$
Parameter 8	$l=7$
Parameter 9	$h=386$
Parameter 10	$i=4$

表 4 演化式群蟻最佳化演算法參數值 (KroA200)

Parameter 1	$m=41$
Parameter 2	$q_0=0.58992$
Parameter 3	$\alpha=1.00894$
Parameter 4	$\beta=4.77123$
Parameter 5	$\rho_{local}=0.13802$
Parameter 6	$\rho_{global}=0.64327$
Parameter 7	$Q=69.0966$
Parameter 8	$l=9$
Parameter 9	$h=412$
Parameter 10	$i=4$

表 5 演化式群蟻最佳化演算法參數值 (TSP225)

Parameter 1	$m=29$
Parameter 2	$q_0=0.60366$
Parameter 3	$\alpha=3.82165$
Parameter 4	$\beta=5.52234$
Parameter 5	$\rho_{local}=0.21545$
Parameter 6	$\rho_{global}=0.70207$
Parameter 7	$Q=49.1409$
Parameter 8	$l=7$
Parameter 9	$h=381$
Parameter 10	$i=4$

3.1.1 KroA100 測試例驗證結果

演化群蟻最佳化演算法執行之 KroA100 解, 4 次重複 Runs 中最佳路徑長度為 21285.44 (整數解為 21282), 路徑規劃為 39 - 30 - 48 - 100 - 41 - 71 - 14 - 3 - 43 - 46 - 29 - 34 - 83 - 55 - 7 - 9 - 57 - 20 - 12 - 27 - 86 - 35 - 62 - 60 - 77 - 23 - 98 - 91 - 45 - 32 - 11 - 15 - 17 - 59 - 74 - 21 - 72 - 10 - 84 - 36 - 99 - 38 - 24 - 18 - 79 - 53 - 88 - 16 - 94 - 22 - 70 - 66 - 26 - 65 - 4 - 97 - 56 - 80 - 31 - 89 - 42 - 8 - 92 - 75 - 19 - 90 - 49 - 6 - 63 - 1 - 47 - 93 - 28 - 67 - 58 - 61 - 51 - 87 - 25 - 81 - 69 - 64 - 40 - 54 - 2 - 44 - 50 - 73 - 68 - 85 - 82 - 95 - 13 - 76 - 33 - 37 - 5 - 52 - 78 - 96 - 39, 最短路徑如圖 6(a)所示。另外, 傳統蟻行最佳化系統執行 KroA100 解(圖 6(b))之最短路徑長度為 21397.58 (整數解為 21394), 路徑規劃為 51 - 87 - 58 - 67 - 28 - 93 - 47 - 1 - 63 - 6 - 49 - 90 - 19 - 75 - 92 - 8 - 42 - 89 - 31 - 80 - 56 - 97 - 4 - 65 - 26 - 66 - 70 - 94 - 22 - 16 - 88 - 53 - 79 - 18 - 24 - 38 - 99 - 36 - 84 - 10 - 72 - 21 - 74 - 59 - 17 - 15 - 11 - 32 - 45 - 91 - 98 - 23 - 77 - 60 - 62 - 35 - 86 - 27 - 12 - 20 - 57 - 9 - 7 - 55 - 83 - 34 - 29 - 46 - 43 - 3 - 14 - 71 - 41 - 100 - 48 - 30 - 39 - 96 - 78 - 52 - 5 - 37 - 33 - 76 - 13 - 95 - 82 - 85 - 68 - 73 - 50 - 44 - 2 - 54 - 40 - 64 - 69 - 81 - 25 - 61 - 51。

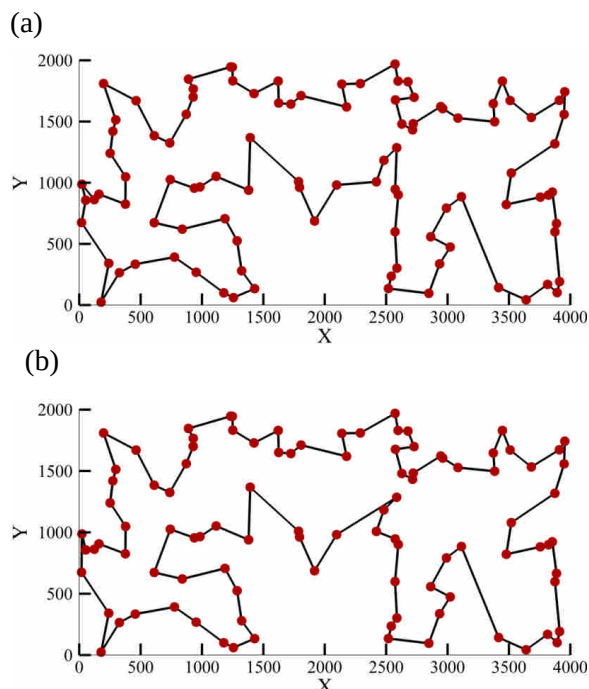


圖 6 以(a) 演化式 ACO 與(b)傳統 ACO 執行 KroA100 之路徑結果

目前已知 TSPLIB 題庫國際例題測試 KroA100 之標準解為 21282, 演化式群蟻最佳

化演算法結果顯示和標準解誤差值為 0%, 而傳統群蟻最佳化演算法結果顯示與標準解誤差值為 5.3%。兩者之路徑搜尋運算收斂圖, 如圖 7 所示, 縱軸為搜尋路徑長度平均值, 由圖顯示演化式群蟻最佳化演算法於 386 次疊代後即可達到最佳解。

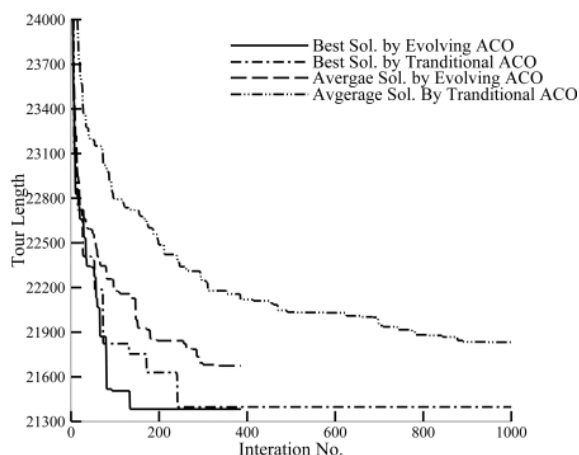


圖 7 演化式 ACO 與傳統 ACO 之 KroA100 收斂比較圖

3.1.2 KroA200 測試例驗證結果

演化式群蟻最佳化演算法執行 KroA200 解(圖 8(a))之最短路徑長度為 21285.44 (整數解為 21282), 路徑規劃為 190 - 49 - 18 - 110 - 29 - 184 - 37 - 179 - 153 - 66 - 119 - 19 - 99 - 92 - 10 - 175 - 36 - 57 - 74 - 100 - 156 - 33 - 45 - 197 - 81 - 97 - 104 - 165 - 166 - 96 - 126 - 87 - 52 - 11 - 84 - 48 - 170 - 122 - 116 - 188 - 44 - 63 - 194 - 51 - 16 - 118 - 124 - 138 - 9 - 78 - 82 - 7 - 199 - 26 - 61 - 136 - 32 - 24 - 159 - 174 - 121 - 172 - 46 - 12 - 147 - 40 - 132 - 111 - 117 - 115 - 53 - 1 - 85 - 145 - 191 - 27 - 198 - 123 - 15 - 13 - 79 - 160 - 162 - 64 - 20 - 55 - 42 - 135 - 186 - 127 - 112 - 120 - 47 - 31 - 67 - 177 - 65 - 80 - 77 - 158 - 193 - 128 - 167 - 30 - 68 - 169 - 35 - 2 - 181 - 125 - 161 - 151 - 187 - 157 - 107 - 109 - 6 - 54 - 75 - 183 - 155 - 8 - 22 - 134 - 129 - 146 - 103 - 143 - 17 - 25 - 90 - 34 - 58 - 141 - 171 - 200 - 98 - 114 - 88 - 148 - 28 - 39 - 38 - 71 - 130 - 72 - 83 - 62 - 185 - 168 - 50 - 139 - 86 - 196 - 56 - 152 - 178 - 5 - 105 - 43 - 137 - 133 - 176 - 113 - 195 - 182 - 94 - 95 - 91 - 150 - 173 - 23 - 144 - 70 - 76 - 102 - 164 - 140 - 21 - 154 - 89 - 41 - 59 - 3 - 73 - 189 - 131 - 180 - 142 - 69 - 108 - 14 - 192 - 60 - 101 - 4 - 163 - 93 - 106 - 149 - 190。而傳統蟻行最佳化系統執行 KroA200 解之最短路徑長度為 29862.70 (整數解為 29860), 路徑規劃(圖 8(b))為 141 - 171 - 200 - 88 - 148 - 28 - 39 - 38 - 71 - 130 - 72 - 83 - 62 -

185 - 168 - 173 - 23 - 150 - 91 - 94 - 95 - 50 - 139 - 86 - 196 - 56 - 152 - 178 - 5 - 105 - 43 - 137 - 133 - 176 - 113 - 195 - 182 - 76 - 70 - 144 - 102 - 164 - 140 - 21 - 154 - 89 - 41 - 59 - 3 - 73 - 189 - 69 - 142 - 131 - 180 - 156 - 100 - 74 - 57 - 36 - 33 - 45 - 197 - 81 - 97 - 104 - 165 - 166 - 96 - 126 - 87 - 52 - 11 - 84 - 48 - 170 - 44 - 153 - 66 - 119 - 19 - 92 - 10 - 175 - 99 - 14 - 192 - 108 - 60 - 101 - 4 - 163 - 93 - 106 - 149 - 190 - 49 - 18 - 110 - 29 - 184 - 37 - 179 - 16 - 63 - 188 - 116 - 122 - 194 - 51 - 118 - 124 - 138 - 9 - 78 - 82 - 7 - 199 - 26 - 61 - 136 - 32 - 24 - 159 - 174 - 121 - 172 - 46 - 12 - 147 - 40 - 111 - 117 - 115 - 53 - 1 - 132 - 85 - 145 - 191 - 27 - 198 - 123 - 15 - 13 - 79 - 160 - 162 - 64 - 20 - 55 - 42 - 135 - 186 - 127 - 112 - 120 - 47 - 31 - 67 - 177 - 65 - 80 - 77 - 158 - 193 - 128 - 167 - 30 - 68 - 169 - 35 - 2 - 181 - 125 - 161 - 151 - 187 - 157 - 107 - 109 - 6 - 54 - 75 - 155 - 183 - 22 - 134 - 8 - 17 - 25 - 143 - 90 - 34 - 103 - 146 - 129 - 114 - 98 - 58 - 141。

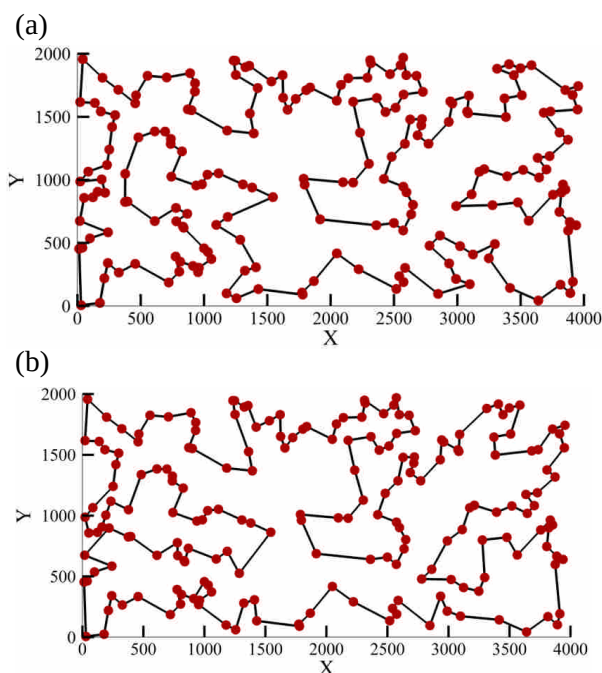


圖 8 以(a) 演化式 ACO 與(b)傳統 ACO 執行 KroA200 之路徑結果

TSPLIB 題庫國際例題測試標準解為 29368，演化式群蟻最佳化演算法結果顯示和標準解誤差值為 0%，傳統群蟻最佳化演算法結果顯示與標準解之誤差值為 1.7%。傳統式與演化式群蟻最佳化演算法兩者之路徑搜尋運算收斂圖，如圖 9 所示，由圖顯示演化式群蟻最佳化演算法比傳統之群蟻最佳化演算法快速達到收斂，演化式群蟻最佳化演算法利用

基因演算法取得 10 個最佳參數後套用於群蟻最佳化演算法中於 412 次疊代後即可快速達到收斂，同時搜尋之最佳路徑優於傳統式群蟻最佳化演算法。由此顯現在群蟻最佳化演算法中適當之參數設定將決定演算法之計算效能。

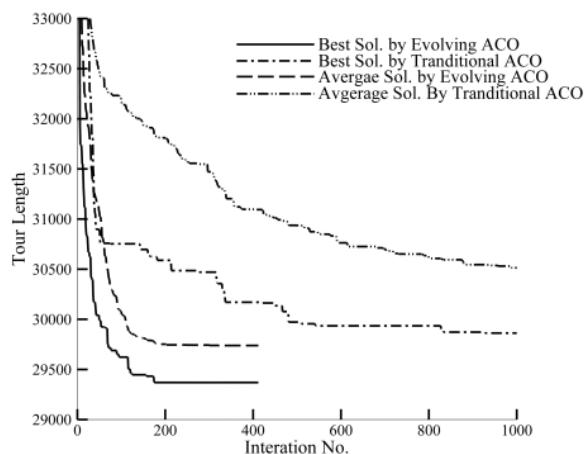


圖 9 演化式 ACO 與傳統 ACO 於 KroA200 收斂比較圖

3.1.3 TSP225 測試例驗證結果

演化群蟻最佳化演算法執行 TSP225 解之最短路徑長度為 3859.78 (整數解為 3913)，路徑規劃(圖 10(a))為 136 - 137 - 138 - 139 - 140 - 141 - 142 - 201 - 143 - 144 - 145 - 146 - 147 - 148 - 149 - 150 - 151 - 152 - 153 - 154 - 155 - 156 - 157 - 159 - 160 - 161 - 162 - 163 - 158 - 213 - 164 - 165 - 166 - 167 - 214 - 212 - 168 - 169 - 170 - 171 - 172 - 173 - 174 - 176 - 177 - 178 - 179 - 180 - 181 - 182 - 184 - 185 - 186 - 187 - 188 - 27 - 189 - 205 - 191 - 192 - 196 - 193 - 218 - 194 - 195 - 197 - 198 - 200 - 1 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10 - 11 - 12 - 13 - 14 - 15 - 16 - 17 - 18 - 19 - 203 - 20 - 21 - 22 - 23 - 24 - 208 - 25 - 26 - 204 - 28 - 29 - 30 - 202 - 206 - 31 - 3235 - 33 - 34 - 36 - 37 - 38 - 39 - 40 - 41 - 42 - 43 - 44 - 46 - 45 - 48 - 224 - 199 - 133 - 190 - 225 - 47 - 2 - 207 - 49 - 51 - 50 - 52 - 53 - 54 - 55 - 56 - 57 - 58 - 59 - 60 - 61 - 62 - 63 - 64 - 65 - 66 - 67 - 68 - 69 - 70 - 71 - 72 - 73 - 74 - 75 - 76 - 216 - 219 - 217 - 77 - 78 - 79 - 80 - 81 - 221 - 82 - 83 - 84 - 85 - 86 - 210 - 87 - 88 - 89 - 90 - 91 - 92 - 93 - 94 - 209 - 95 - 96 - 97 - 98 - 99 - 100 - 101 - 102 - 103 - 104 - 220 - 105 - 106 - 107 - 108 - 109 - 110 - 111 - 112 - 113 - 114 - 115 - 223 - 116 - 117 - 118 - 119 - 120 - 175 - 121 - 122 - 123 - 124 - 125 - 126 - 127 - 128 - 129 - 222 - 130 - 211 - 131 - 132 - 215 - 134 - 135 - 183 - 136。而傳統蟻行最佳化系統執行 TSP 範例之 TSP225 解，最短路徑長度為

3961.45 (整數解為 3959) , 路徑規劃(圖 10(b)) 為 55 - 56 - 57 - 58 - 59 - 60 - 61 - 62 - 63 - 64 - 65 - 66 - 67 - 68 - 69 - 70 - 71 - 72 - 73 - 74 - 75 - 76 - 216 - 219 - 217 - 77 - 78 - 79 - 81 - 221 - 82 - 83 - 84 - 85 - 86 - 210 - 87 - 88 - 89 - 90 - 91 - 92 - 93 - 94 - 209 - 95 - 80 - 96 - 97 - 98 - 99 - 100 - 101 - 102 - 103 - 104 - 220 - 105 - 106 - 107 - 108 - 109 - 110 - 111 - 112 - 113 - 114 - 115 - 223 - 116 - 118 - 119 - 175 - 121 - 122 - 123 - 124 - 25 - 126 - 127 - 128 - 129 - 222 - 130 - 211 - 131 - 132 - 215 - 134 - 135 - 183 - 136 - 137 - 138 - 139 - 140 - 141 - 142 - 201 - 143 - 144 - 145 - 146 - 147 - 148 - 149 - 152 - 153 - 154 - 155 - 156 - 157 - 159 - 160 - 161 - 162 - 163 - 158 - 213 - 164 - 65 - 166 - 167 - 151 - 150 - 214 - 212 - 168 - 169 - 170 - 171 - 172 - 174 - 176 - 177 - 178 - 179 - 180 - 181 - 173 - 182 - 184 - 20 - 185 - 186 - 187 - 117 - 188 - 27 - 189 - 205 - 191 - 192 - 196 - 193 - 218 - 194 - 46 - 195 - 197 - 4 - 198 - 3 - 200 - 1 - 5 - 6 - 7 - 8 - 9 - 10 - 11 - 12 - 13 - 14 - 15 - 16 - 17 - 18 - 19 - 203 - 20 - 21 - 22 - 23 - 24 - 208 - 25 - 26 - 204 - 28 - 29 - 30 - 202 - 206 - 31 - 32 - 5 - 33 - 34 - 36 - 37 - 38 - 39 - 40 - 41 - 42 - 43 - 44 - 45 - 48 - 224 - 199 - 133 - 190 - 225 - 47 - 2 - 207 - 49 - 51 - 50 - 52 - 53 - 54 - 55 。

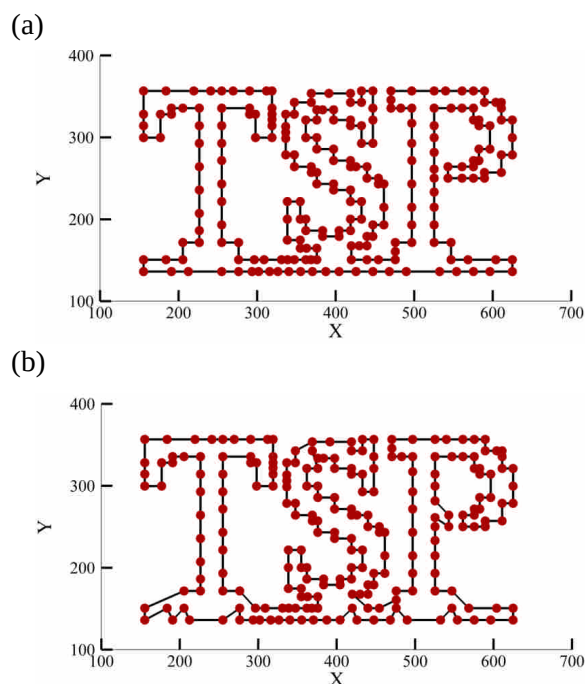


圖 10 以(a) 演化式 ACO 與(b)傳統 ACO 執行 TSP225 之路徑結果

TSPLIB 題庫國際例題測試標準解：3916，演化式群蟻最佳化演算法結果顯示和標準解誤差值為 0%，傳統群蟻最佳化演算法結

果顯示比標準解誤差值為 2.2%。兩者之路徑搜尋運算收斂圖，如圖 11 所示，由圖顯示演化式群蟻最佳化演算法比傳統之群蟻最佳化演算法快速達到收斂，演化式群蟻最佳化演算法於 381 次疊代後即達收斂。

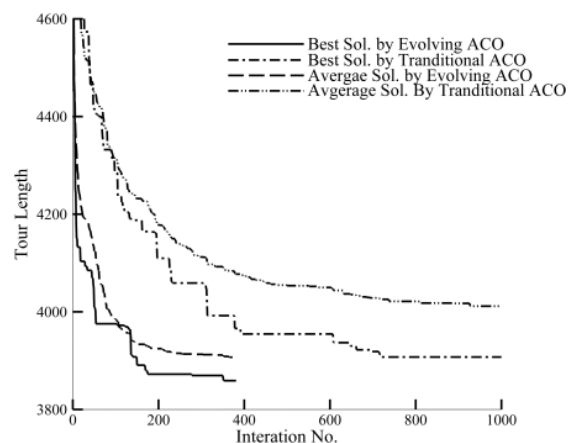


圖 11 演化式 ACO 與傳統 ACO 之 TSP225 收斂比較圖

表 6 演化式 ACO 與傳統 ACO 之結果比較表

TSP	演算法	參考解	最佳解	誤差值	疊代數
KroA100	傳統 ACO	21282	21394	5.3%	475
	演化式 ACO		21282	0%	386
KroA200	傳統 ACO	29368	29867	1.7%	830
	演化式 ACO		29368	0%	412
TSP225	傳統 ACO	3916	3959	2.2%	710
	演化式 ACO		3913	0%	381

進一步將 KroA100、KroA200 及 TSP225 三個測試例之傳統與演化式群蟻最佳化演算法結果比較分析，如表 6 所示。由結果可見演化式群蟻最佳化演算法可大幅度改善傳統群蟻最佳化演算法之搜尋結果及搜尋效能。另外，由表 2 至表 5 比較可知較佳之螞蟻數量約為 30，此與 Dorigo *et al.* [2]所建議之值相當，而 q_0 比一般值 0.9 還低，此結果與 Botee 與 Bonabeau [1]之結果一致，最佳化之 α 與 β 兩個值比較時， β 的值需比 α 大，此結果符合可

見度 η (Visibility) 的權衡重要性比費洛蒙 τ (Pheromone) 來的高，此與群蟻最佳化演算法原創者 Dorigo 之看法一致。另外，較大的區域與全域的費洛蒙揮發係數 (ρ_{local} 與 ρ_{global} 兩者值大於 0.1) 將有利於加速搜尋，此結果與 Botee 與 Bonabeau [1] 之結果一致，方程式 (3) 中之 Q 值低於 100 較佳，候選名單 l 約 7-9 以及重複執行 4 次即可得到很好的搜尋結果。一般而言，不同的案例計算時群蟻最佳化演算法中之參數應當有所不同，此結果亦由本文提出之演化式群蟻最佳化演算法的計算結果得到印證，因此，本文提出之演化式群蟻最佳化演算法可依各種不同之應用問題自行調整出群蟻最佳化演算法之所需最佳化參數，同時均可獲致最佳化的結果。

3.1.4 建議之群蟻最佳化演算法參數值分析

本文先以 Botee 與 Bonabeau 所提出的演化參數 [1] (如表 7)，以 TSP 之 KroA100 的案例作分析比較。採用 Botee 與 Bonabeau 所提的演化參數所執行的結果為 24521.35 (整數: 24520)，由圖 12 得知最佳解及收斂速度比演化式群蟻最佳化演算法來的差。此部份驗證 Botee 與 Bonabeau 所提出的演化參數，僅使用小型節點數 (Oliver30) 之單一案例演化之參數組合，對於大型節點數量的案例結果並不是最佳的。

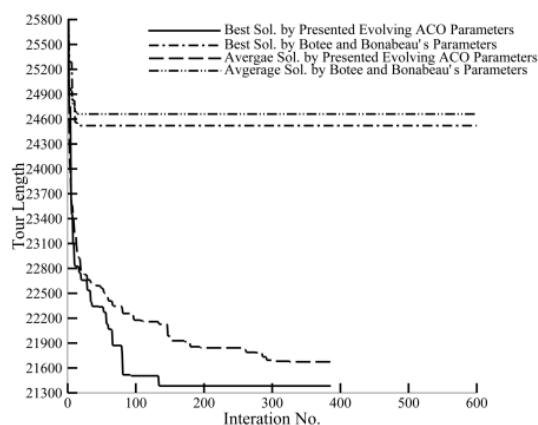


圖 12 Botee 與 Bonabeau 所提出的演化參數組合與本文計算 KroA100 路徑搜尋之收斂圖

表 7 Botee 與 Bonabeau 建議之群蟻最佳化演算法參數值 [1]

Parameter 1	$m=14$
Parameter 2	$q_0=0.384665$
Parameter 3	$\alpha=0.370642$
Parameter 4	$\beta=6.683146$

Parameter 5	$\rho_{local}=0.302831$
Parameter 6	$\rho_{global}=0.308385$
Parameter 7	$Q=78.042267$
Parameter 8	$l=1$

本文進一步將前述 3 個大型 TSPLIB 案例所演化出的最佳參數，以均值落點分析，約略可得到之建議之最佳參數組合，如表 8 所示。

表 8 本文建議之群蟻最佳化演算法參數值

Parameter 1	$m=30$
Parameter 2	$q_0=0.6$
Parameter 3	$\alpha=3$
Parameter 4	$\beta=6$
Parameter 5	$\rho_{local}=0.2$
Parameter 6	$\rho_{global}=0.7$
Parameter 7	$Q=61$
Parameter 8	$l=8$
Parameter 9	$h=400$
Parameter 10	$i=4$

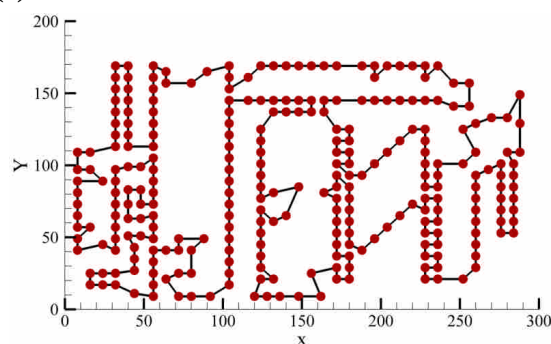
3.2 最佳參數組合驗證大型旅行銷售員問題

採用表 8 之最佳參數組合應用於 300 節點附近之大型旅行銷售員問題上，分析案例包括 A280、Pr299 以及 Lin318 等。同時將結果與採用傳統式參數之結果作比較，以驗證這些參數的可靠性。

3.2.1 A280 大型旅行銷售員案例分析

演化式群蟻最佳化演算法之最佳參數執行 A280 之最短路徑長度為 2603.64 (整數解為 2595)；而傳統蟻行最佳化系統執行之最短路徑長度為 2647.86 (整數解為 2638)。參考 TSPLIB 例題測試標準解為 2579，顯示演化式群蟻最佳化演算法之最佳參數結果和標準解誤差值僅為 0.6%，而傳統群蟻最佳化演算法結果之誤差值為 2.3%。兩者之搜尋路徑圖與運算收斂圖，分別表示於圖 13 與 14，由結果顯示演化式群蟻最佳化演算法之最佳參數比傳統之群蟻最佳化演算法快速達到收斂。

(a)



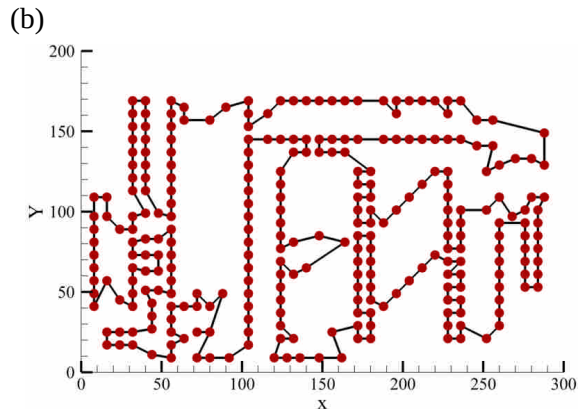


圖 13 (a)最佳參數與(b)傳統參數執行 A280 之路徑結果

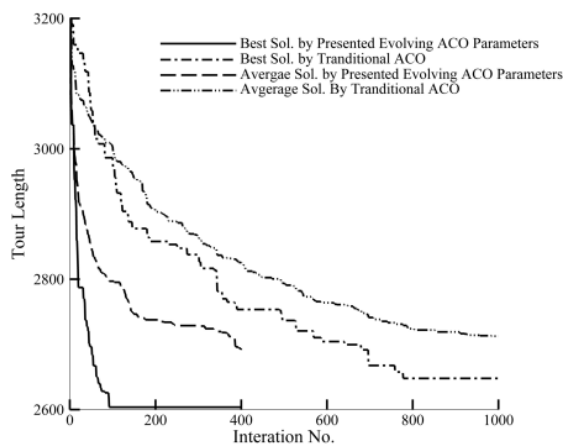


圖 14 演化式 ACO 最佳參數與傳統 ACO 於 A280 路徑搜尋運算收斂圖

3.2.2 Pr299 大型旅行銷售員案例分析

採用最佳參數於群蟻最佳化演算法執行 Pr299 解之最短路徑長度為 48814.21 (整數解為 48809)；而傳統參數之最短路徑長度為 49659.63 (整數解為 49655)。由 TSPLIB 例題測試標準解為 48191，可計算出採用本文提出之最佳參數組合於群蟻最佳化演算上時其結果顯示和標準解誤差值為 1.3%，而採用傳統參數與標準解誤差值為 3.0%。兩者之搜尋路徑圖與運算收斂圖分別表示於圖 15 與 16，由結果顯示演化式群蟻最佳化演算法之最佳參數比傳統之群蟻最佳化演算法快速達到收斂。

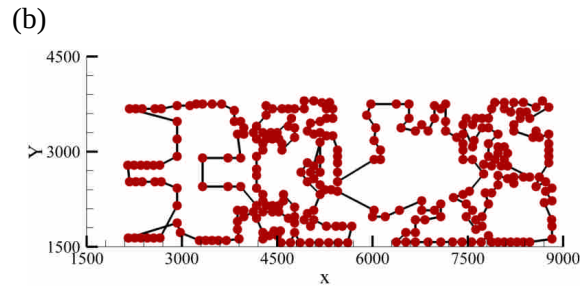
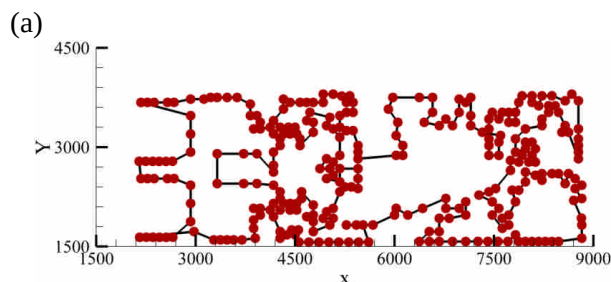


圖 15 (a)最佳參數與(b)傳統參數執行 Pr299 之路徑結果

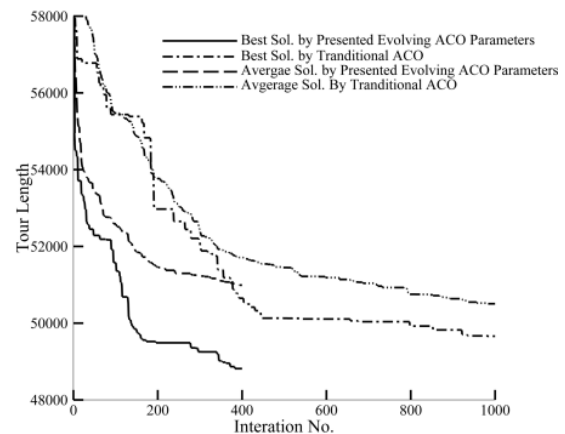
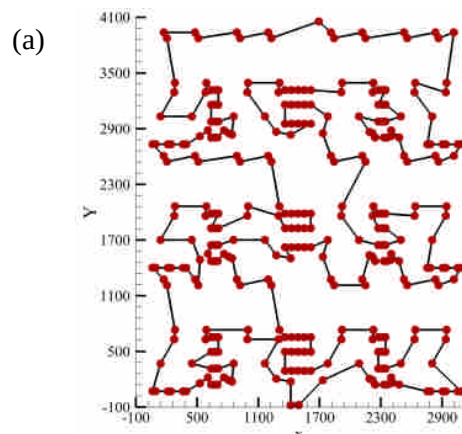


圖 16 演化式 ACO 最佳參數與傳統 ACO 於 Pr299 路徑搜尋運算收斂圖

3.2.3 Lin318 大型旅行銷售員案例分析

採用最佳參數於群蟻最佳化演算法執行 Lin318 解其最短路徑長度為 43014.06(整數解為 43002)；而傳統參數之最短路徑長度為 43769.96(整數解為 43758)。參考 TSPLIB 例題測試標準解為 42029，可計算出採用本文提出之最佳參數組合於群蟻最佳化演算上時其結果顯示和標準解誤差值為 2.3%，而採用傳統參數與標準解誤差值為 4.1%。兩者之搜尋路徑圖與運算收斂圖分別表示於圖 17 與 18，由結果顯示演化式群蟻最佳化演算法之最佳參數比傳統之群蟻最佳化演算法快速達到收斂。



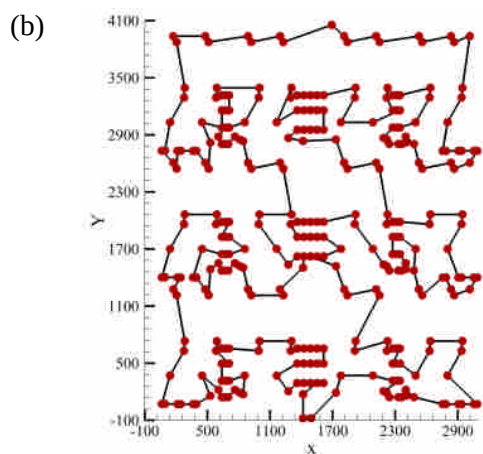


圖 17 (a)最佳參數與(b)傳統參數執行 Lin318 之路徑結果

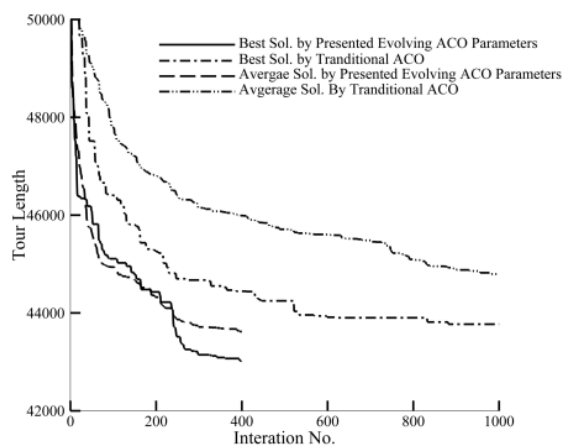


圖 18 演化式 ACO 最佳參數與傳統 ACO 於 Lin318 路徑搜尋運算收斂圖

表 9 為經由演化群蟻最佳化演算法後歸納出的一組最佳化參數，應用於 300 節點數之大型 TSP 問題 A280、Pr299 與 Lin318，由表中數據顯示結果本文提出之最佳化參數表現優於傳統參數。在此表中包含了參考解、最佳解以及標準解誤差值。

表 9 最佳參數與傳統參數組合應用群蟻最佳化演算法於 A280、Pr299 與 Lin318 之比較

例題	參數組	參考解	最佳解	誤差值
A280	傳統參數組	2579	2638	2.3%
	本文建議之參數組		2595	0.6%
Pr299	傳統參數組	48191	49655	3.0%

	本文建議之參數組		48809	1.3%
Lin3185	傳統之參數組	42029	43758	4.1%
	本文建議之參數組		43002	2.3%

4. 結論

本文所提出利用基因演算法求解全域最佳化參數導入群蟻最佳化演算法，能適當地設定參數值及加快求解的收斂速度。求解流程以群蟻最佳化演算法求解之結果作為基因演算法之環境適應值(亦即目標函數值)，繼而操作基因演算法的選擇、交配與突變三個操作子以及採用精英策略，經由數次疊代後即可求得群蟻最佳化演算法中的各個最佳參數值。本文將此演化式群蟻最佳化演算法應用在旅行銷售員問題以測試本文所提出之方法的可靠性與準確性，採用 TSPLIB 題庫國際例題測試例 Kroa100、KroA200 與 TSP225 作為實例運算。由結果得知，演化式群蟻最佳化演算法所得到的最佳解與 TSPLIB 題庫國際例題標準解比較誤差均為 0%，同時演化式群蟻最佳化演算法求解的收斂速度亦均比傳統群蟻最佳化演算法來得快，顯見演化式群蟻最佳化演算法可顯著地提升最短路徑搜尋的性能。本論文也將 Botee 與 Bonabeau 以單一例題演化得到之最佳參數應用於 KroA100 做為實驗參考，驗證結果單一演化參數並無法適用所有例題。所以，本研究亦將上述 3 個 TSPLIB 例題測試中所演化出的最佳參數加以分析推論後，得到一組最佳參數趨勢值應用於測試例 A280、Pr299 與 Lin318 三個大型的旅行銷售員問題測試，結果顯示 A280、Pr299 與 Lin318 案例與目前最佳解之誤差均在 2.3%以內，而傳統蟻行最佳化系統所得到最佳解與參考解誤差均大於 2.3%以上，在收斂速度上也比採用傳統參數來的快速，顯見採用演化式蟻行最佳化系統推演出來的參數組可顯著地提升最短路徑搜尋的性能。因此，本論文之演化式群蟻最佳化演算法應為一大有可為的最佳化路徑搜尋方法，同時所提出之最佳參數組合應可應用於更大型之旅行銷售員問題上。

參考文獻

- [1] Botee, H.M. and Bonabeau, E., "Evolving Ant Colony Optimization," *Adv. Complex*

- Systems, No. 1, pp. 149-159, 1998.
- [2] Dorigo, M., Maniezzo, V., and Colomi, A., "The Ant System: Optimization by a Colony of Cooperation Agents," *IEEE Transactions of Systems, Man and Cybernetics*, Part B, Vol. 26, No.2, pp. 29-41, 1996.
 - [3] Dorigo, M., and Gambardella, L.M., "Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem," *IEEE Transactions on Evolutionary Computation*, Vol. 1, No.1, pp. 53-66, 1997.
 - [4] Dorigo, M., *Ant Colony Optimization*, Massachusetts Institute of Technology, pp. 25-117, 2004.
 - [5] Dorigo, M., Bonabeau, E., and Theraulaz, G., "Ant Algorithms and Stigmergy," *Future Generation computer Systems*, Vol. 16, No. 8, pp. 889-914, 2000.
 - [6] Eiben, A.E. and Smith, J.E., *Introduction to Evolutionary Computing*, Springer-Verlag Berlin, pp. 37-69, 2003.
 - [7] Gillett, B. and Miller, L., "A Heuristic Algorithm for the Vehicle Dispatch Problem," *Operational Research*, Vol. 22, No. 2, pp. 340-349, 1974.
 - [8] Goldberg, D.E., *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading Massachusetts, 1989.
 - [9] Holland, J.H., "Genetic Algorithm and the Optimal Allocation of Trials," *SIMA J. Computing*, Vol. 2, No. 2, pp.88-105, 1973.
 - [10] Holland, J.H., *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, 1975.
 - [11] Michalewicz, Z., *Genetic Algorithms + Data Structures = Evolution Programs*, Springer-Verlag Berlin, 1996.
 - [12] Sengoku, H. and Yoshihara, I., "A Fast TSP Solution using Genetic Algorithm," *Information Processing Society of Japan 46th National Conference*, pp. 121-125, 1993.
 - [13] Library of Traveling Salesman Problems, Interdisciplinary Center for Scientific Computing, <http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/index.html>