運輸計劃季刊
第四十五卷 第三期
民國一〇五年九月
頁 189～頁 216

Transportation Planning Journal
Vo1. 45 No. 3
September 2016
PP. 189 ～216

以和諧演算法為基礎之混合全域搜尋法求解最小凹型成本轉運問題[1]

HYBRID GLOBAL SEARCH ALGORITHM BASED ON HAMONY SEARCH OPTIMIZATION FOR MINIMUM CONCAVE COST TRANSSHIPMENT PROBLEMS

顏上堯 ShangyaoYan [2]
林至康 Chih-Kang Lin [3]
劉向邦 Xiang-Bang Liu [4]

(105 年 2 月 19 日收稿，105 年 8 月 17 日第 1 次修改，105 年 9 月 26 日定稿)

摘 要

在實務上，貨物運送的單位成本常隨數量的增加而遞減，其成本函數曲線為凹形，而此類問題可定式為含凹形節線成本之最小成本網路流動問題，但此問題屬於NP-hard問題，故難在有限時間內求得大型問題的最佳解。新近的和諧搜尋演算法目前在各領域的問題求解上效果頗佳，但尚未發現有應用於含凹形節線成本最小成本網路流動問題，緣此，本研究以和諧搜尋演算法為基礎，並結合粒子群演算法、螞蟻族群演算法、門檻值接受法與凹形成本網路啟發解法之特點，以節線及路徑為基礎發展一混合式全域搜尋法，以有效求解含凹形節線成本之最小成本網路流動問題。為測試本研究演算法在不同規模及參數的網路問題之求解績效，本研究設計一隨機

1. 本研究承行政院科技部專題研究計畫部分補助 (計畫編號：NSC- 104-2221-E-008-031-MY3)，特此致謝。作者亦感謝二位審查委員寶貴的修正意見，使本文更臻完整與嚴謹。
2. 國立中央大學土木工程學系教授 (聯絡地址：320 桃園市中壢區五權里中大路 300 號 中央大學土木工程學系；電話：03-4227151 分機 4070；E-mail:t320002@cc.ncu.edu.tw)。
3. 中華大學運輸科技與物流管理學系助理教授。
4. 國立中央大學土木工程學系碩士。

網路產生器產生大量隨機網路，並測試遺傳演算法、門檻值接受法、大洪水法、類螞蟻族群演算法及粒子群演算法，以評估本研究演算法之求解績效。測試結果顯示本研究演算法求解品質良好，可提供實務界求解此類網路運送問題之參考。

關鍵詞：和諧搜尋演算法；凹形節線成本；最小成本網路流動問題；全域搜尋

ABSTRACT

In practice, the unit cost for transporting freight usually decreases as the amount of freight increases. Hence, in actual operations the transportation cost function can usually be formulated as a concave cost function, causing the transshipment problem an NP-hard problem. The harmony search (HS), a global search algorithm, has led to good results in many applications. Since there has not yet been any research applying HS to minimum concave cost network flow problems, we employ HS, coupled with the techniques of PSO, ACS and TA, to develop a global search algorithms for efficiently solving minimum concave cost network flow problems. Finally, to evaluate our algorithms we designed a network generator to create a sufficient number of problem instances. To evaluate our algorithm, we also tested the recently designed TA, GDA, GA, ACS and PSO that solve minimum concave cost network flow problems. The results show that the developed algorithms performed well in the tests.

Key Words: *Harmony search; Concave arc cost; Minimum cost network flow problem; Global search*

一、緒　論

以往學界或實務界在最小成本網路流動問題的定式 (formulation) 上，常將各節線之單位運送成本予以簡化為固定值，即以線性成本來定式運送成本，其主要原因在於簡化後的線性成本較易於求解。然而，此假設常與許多實際貨物之運送成本不相符合，實際上單位運送成本常隨著運送貨物的增加而減少，亦即經濟學上常見之經濟規模特性。由於貨物的運送成本主要受到物品種類不同與種類多寡、運送方式、運送距離、以及運送數量的影響，實務上高固定成本或低變動成本的運送方式，例如鐵路與水運，通常貨物運送的每單位成本隨貨物量的增加而減少，故此類的經濟規模特性，貨物運送成本函數常可定式為凹形成本 [1,2]，以符合實務上的運輸型態。

然而，含凹形節線成本之最小成本網路流動問題，則因在網路流動規劃問題中屬於 NP-hard 問題 [3]，故難在有限時間內求得大型問題的最佳解。故學術上求解含凹形節線成本之運輸問題，可分為三大方向進行處理，現分述如下：

一、直接將問題形式簡化為線性問題，將網路節線的單位運送成本簡化為固定值，並以線性規劃的方式求解。例如 Zangwill [4] 運用動態規劃法，將多元貨品 (multi-commodity) 簡化為單一貨品類型所建立的單一起點至單一迄點模式進行求解；Rech 與 Barton [5] 曾以運送單一貨物種類從 m 個起點到 n 個終點的非凸性分段線性網路 (non-convex piecewise linear network)，運用分枝定限法來求得下限解；Gallo 與 Sandi [6] 曾從凹形成本網路最佳化的觀點，由網路上給定的端點解，利用最佳解落在可行解端點的特性，將原問題簡化後以窮舉方式求解；Gallo 等人 [7] 曾以單一起點至多個迄點的最小成本網路流動問題，用分枝定限法找尋全域最佳解，同時用線性鬆弛原問題的限制式以求得下限解。基本上此類求解方法在問題規模擴大時，受限於 NP-hard 問題特性，將無法於有限的時間內求得最佳解，故實用性受到限制。

二、將非線性問題分成數個子問題，以線性近似法求解，此類求解方法多搭配整數規劃之技巧進行求解。例如：Nourie 與 Guder [8] 曾對於 piecewise linear 之凹形成本運輸問題，將成本分成生產成本與運送成本兩部分，再以限制進入基底 (restricted-entry basis) 的方式改良單體法 (simplex method)，並反覆求解線性問題以找出區域最佳解；然而，透過此類簡化後的線性成本，常難以反應實際成本，而不易求得近似最佳解；而以線性近似法求解凹形成本的轉運問題，當問題規模變大時，則傳統整數規劃模式將因問題複雜度增加而難以求解。

三、利用改良式區域搜尋法 (local search) 或全域式演算法 (global search) 等巨集啟發式演算法進行問題求解。所謂巨集啟發式演算法乃是以傳統啟發式方法為核心架構，結合高階的搜尋策略 (meta-strategies)，使其能跳脫局部最佳解的束縛而找到更好的解 (Osman 與 Kelly [9])，因此多數高階搜尋策略大多取材自物理化學、生物自然、人類思考之行為或邏輯。諸如，模擬退火法 (simulated annealing) (Kirkpatrick 等人 [10])、遺傳演算法 (genetic algorithm, GA) (Goldberg [11])、禁制搜尋法 (tabu search, TS) (Glover [12,13])、門檻值接受法 (threshold accepting, TA) (Dueck 與 Scheuer [14])、大洪水法 (great deluge algorithm, GDA) 與記錄更新法 (record-to-record travel) (Deuck [15])、類神經網路 (neural network) (Reeves [16])、噪音擾動法 (noising method) (Charon 與 Hurdy [17]、韓復華與卓裕仁 [18])、兩極跳躍法 (flip-flop method) (韓復華等人 [19])、搜尋空間平滑法 (search space smoothing) (Gu 與 Huang[20])、粒子群最佳化演算法 (partical swarm optimization, PSO) (Kennedy 與 Spears [21])、螞蟻族群最佳化 (ant colony optimization, ACO) (Dorigo 與 Gambardella [22])、變動鄰域搜尋法 (variable neighborhood search, VNS) (Hansen 與 Mladenovic [23])、和諧搜尋演算法 (harmony search, HS) (Geem 等人 [24]) 等。此等新近演算法，相較以往的啟發式演算法，能有效的跳脫局部最佳解，使得搜尋範圍更大，因而提高求得最佳解 (或較佳解) 的可能性，因此在不少的應用中，具有良好的成果。

依據過去經驗，由於巨集啟發式演算法具有在合理時間內求解大規模問題之優點，且啟發解的品質通常也可被接受，因此近年來許多學者均以巨集啟發式演算法求解含凹形節

線成本之最小成本網路流動問題。例如 Yan 與 Luo [1,2] 曾利用新近組合最佳化的方法，如模擬退火法 (SA)、禁制搜尋法 (TS)、門檻值接受法 (TA) 等鄰近搜尋法，針對單純之運輸問題，發展求解演算法，且經過測試及比較傳統啟發解後，發現效果頗佳。然而，此等鄰近搜尋法在求解含凹形節線成本之最小成本網路流動問題上，會面臨退化的問題，造成搜尋上的低效率，且是否能有效的搜尋全域求得最佳解 (或近似最佳解)，往往不得而知。有鑑於此，Yan 等人 [25] 更進一步利用遺傳演算法，並改良編碼的方式，以發展一求解演算法，經測試後，發現其求解效果較上述鄰近搜尋法為佳。另外顏上堯等人 [26] 與 Yan 等人 [27] 則利用粒子群最佳化 (PSO) 演算法之觀念，分別發展 PB-APSO 與 AB-APSO 兩種全域式搜尋演算法，並針對問題特性提出可行解產生方式、速度更新公式、門檻值個體最佳解與門檻值全體最佳解更新方式、速度範圍限制策略，經過 49 組參數測試後，發現此演算法求解績效比以往之研究更佳。另 Yan 等人[28]亦利用新近的螞蟻演算法之搜尋概念為基礎，並結合 GA、TA 與凹形成本網路啟發解法之特點，發展一類螞蟻演算法 (analogous ant colony system, AACS)，經測試後，發現其求解效果頗佳。因此，可得知利用混和式全域搜尋演算法求解含凹形節線成本之最小成本網路流動問題，均可獲得非常不錯的求解績效。

和諧搜尋演算法 (harmony search, HS) 為一新發展之巨集啟發式演算法，此演算法由 Geem 等人 [24] 提出。和諧搜尋演算法以音樂即興演奏為基礎，模擬數名演奏家之和聲行為進行求解，在演算法求解過程中各樂手即代表欲求解問題之決策變數，各種樂器都有各自的音域範圍，即決策變數之可行範圍，透過演算法的和弦組合與調音機制，將較優可行解放入和弦記憶空間 (harmony memory, HM) 中，直至符合演算法之停止機制，即可獲得原問題較佳的可行解。此演算法架構簡明易實現，亦容易配合其他演算法進行求解，因此，能針對問題特性加入有利於問題求解之步驟。到目前為止，許多學者亦將此演算法運用在多種不同領域之問題，並且得到頗佳的結果。例如 Geem 等人 [24] 首先應用和諧搜尋演算法於管線網路設置問題，在滿足水資源流量需求及最小化頸輸送管的情形下，以和諧搜尋演算法求解管線網路設置總成本最小化；之後 Geem 等人 [29] 又利用和諧搜尋演算法求解桁架問題，測試規模最大至 200 根桁架，並與 GA 等演算法進行範例測試與比較，結果發現和諧搜尋演算法求解結果與效率均較 GA 等演算法佳。另 Geem 等人 [30] 利用和弦搜尋演算法求解 TSP，並以全中國為例，求解旅行員最佳路徑。之後李亮等人 [31] 利用和弦搜尋演算法求解土坡臨界滑動面，其演算法無論用於簡單或複雜的土坡，都能有效尋找到最危險的滑動面。李亮等人 [32] 亦將和弦搜尋演算法應用於土坡穩定分析中，以尋找更小的安全係數，均獲得不錯的成效。Wang 等人 [33] 利用和弦搜尋演算法之求解架構，針對無暫存區流程型工廠排程問題 (flow shop scheduling problem with blocking constraint) 之特性，發展出 3 種混合式和弦求解演算法進行問題求解。Afkhami 等人 [34] 針對傳統組合優化中最大團問題（maximum clique problem, MCP）的特性，提出一個修正式的二元和諧演算法 (binary harmony search, BHS) 進行問題求解。Hosseini 等人 [35] 則針對貨物運送問題之特性，將供應商直接送至顧客與透過已排定發車時間、排定路程、取送貨點車輛 (milk run

logistics) 送貨等兩種不同性質之網路進行結合，針對此路網提出一個整數規劃模式，並發展一套和諧求解演算法進行問題求解。

由上述文獻可得知和諧搜尋演算法除了廣泛的應用在結構工程計算上，另外亦已應用在銷售員旅行問題 (TSP)、管線網路設置問題與含額外限制式之最小化問題上，並且都能得到良好的結果。然而，以往卻未發現有利用和諧搜尋演算法求解凹形成本網路流動問題，有鑑於此，本研究針對含凹形節線成本之最小成本網路流動問題特性，以和諧搜尋演算法為基本架構，引用和諧搜尋演算法中以和弦記憶空間為母體 (harmony memory consideration) 之全域搜尋法與調音機制 (pitch adjusting) 之區域搜尋法重新組成新可行解，並加入粒子群演算法 (PSO) 速度更新策略、螞蟻族群演算法 (ACS) 費洛蒙更新策略、門檻接受策略 (TA) 與凹形成本網路啟發解法之演算特色，針對凹形成本網路流動問題之特性，發展一套適合凹形節線成本轉運問題之混合型類和諧搜尋演算法 (analogous harmony search, AHS)。最後，為測試比較本研究求解績效，本研究參考 Yan 等人 [25] 針對本問題所發展的遺傳演算法 (GA)、門檻值接受法 (TA)、大洪水法 (GDA) 與禁制搜尋法 (TS)、顏上堯等人 [26] 與 Yan 等人 [28] 針對本問題所發展的粒子群演算法、及 Yan 等人 [27] 所發展的螞蟻族群演算法之測試結果，以測試分析本研究演算法之求解績效。

二、問題描述與求解演算法設計

2.1 問題定式及特性

基本上，成本函數是指在技術水平不變的條件下，成本與產出之間的相互關係，可反應出成本隨產量變化而規律變化的關係。一般而言，具規模經濟的凹型成本函數，可分為包含固定成本與變動成本的凹型成本函數及不含固定成本 (僅具有變動成本) 的凹型成本函數兩大類 (如圖 1 所示)，其中包含固定成本與變動成本的凹型成本函數，更進一步分為固定費率用的凹型成本 (fixed charge cost)、凹型成本 (concave cost) 與分段凹型成本 (piecewise concave cost) 3 類；而不含固定成本 (僅具有變動成本) 的凹型成本函數，則分為凹型成本 (concave cost) 與分段凹型成本 (piecewise concave cost) 兩類。本研究探討的對象為含凹形節線成本之最小成本網路流動問題 (minimum concave cost network flow problem, MCNFP)，至於其他類之凹型成本函數問題，雖仍可以本研究演算架構求解，但其測試與績效可在未來再進行深入探討。

本研究探討的最小凹型成本網路流動問題，其一般性的網路含有多個起點、終點與轉運點，具有方向性或不具方向性之節線。由於具節線流量上限之網路將可透過網路修正方式，轉換成不具節線流量上限之網路 (Ahuja 等人 [36])，另不具方向性節線網路亦可透過網路修正方式，轉換為具方向性節線網路，因此，本研究為適度簡化演算法之發展，將針對單一貨物運送且節線流量上限為無窮大的有方向性網路求解。以下針對具方向性之含凹形成本最小成本網路流動問題之數學模式作一描述，且探討其相關理論特性，以為後續演算法設計之依據。

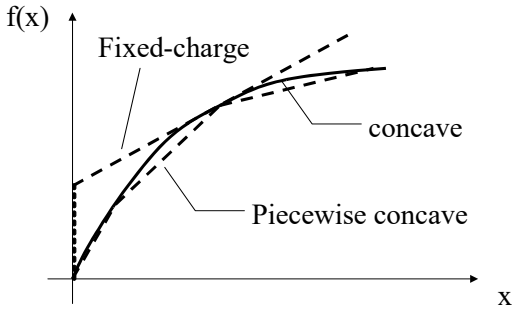

圖 1　凹形成本函數示意圖 (Yan **與** Luo [2])

依據上述定義，令 $G=(N,A)$ 為一有方向性之網路，其中 N 為所有節點之集合，A 為節線集合，則含凹形節線成本的最小成本網路流動問題或轉運問題可定式為：

$$\text{Min} \quad \sum_{j\in A} f_{ij}(x_{ij}) \tag{1}$$

$$\text{S.T.} \quad \sum_{j} x_{ij} - \sum_{k} x_{ki} = b_i, \quad \forall i \in N \tag{2}$$

$$x_{ij} \geq 0, \quad \forall (i,j) \in A \tag{3}$$

其中，目標式 (1) 表示總成本最小化，而節線 ij 的成本函數 $f_{ij}(x_{ij})$ 為凹形成本型態，x_{ij} 為節線 ij 之運送量。限制式 (2) 為節點流量守恆限制式，當 i 點為供給點時，$b_i > 0$；為需求點時，$b_i < 0$，其餘各點 (為轉運點) $b_i = 0$。限制式 (3) 為節點流量非負限制式。此外，為確保此最小成本網路流動問題之可行性，本研究設定總供給量等於總需求量，故於本研究中所有測試的網路均必須滿足 $\sum b_i = 0$ 之條件。

此種轉運問題的定義為將所有貨物從所有供給點，透過所有供需點及轉運點，運送到所有需求點，並使花費的運送成本最小化。本研究為便於演算法之發展，對於問題形式做了下列假設：

1. 所有節線流量均介於 0 到無窮大之間，亦即節線流量無上限。另外，二節點間可能存在來回方向之節線，且兩方向節線成本不一定相同。
2. 本研究旨在探討啟發式解法應用在含凹形節線成本運輸問題上的適用性，故為簡化問題複雜度並方便測試起見，茲參考 Yan 與 Luo [2] 及 Yan 等人 [25] 中節線成本的設計方式，假設節線 ij 之運送總成本為 $f_{ij} = c_{ij}\sqrt{x_{ij}}$，單位運送成本為 $f_{ij} = c_{ij}/\sqrt{x_{ij}}$，其中 x_{ij} 為流量，c_{ij} 為成本參數。至於其他類凹形成本函數的測試，可為後續研究。
3. 令節點總供給量等於節點總需求量，以確立網路流量守恆。

4. 不考慮節點成本 (node cost)，亦即貨物運送經過節點時不增加其成本。

本研究網路問題屬於非線性的最小凹形成本網路流動問題，因網路中各節線成本均為凹形成本，故目標函數將形成一凹形成本函數 (Ahuja 等人 [36])，而在參考 Yan 與 Luo [2] 類似的證明後，可得知各端點為一區域最佳解，且此凹形成本函數轉運問題的最佳解必落在可行解區域 (feasible region) 的一端點上。此外，以往學術上在求解一般線性最小成本網路流動問題時，大都利用伸展樹 (spanning tree) 之觀念求解，因此將此觀念應用至本研究問題時，可得知在凹形成本網路中，一個可行伸展樹將對應唯一的一組路段解流量，其滿足供給點與需求點間之運送。綜合上述，本問題具有下列特性 (Yan 等人 [28])：

1. 最佳解落在端點；
2. 端點解為整數解，且對應唯一的一組路段解流量。

本研究網路問題為非線性規劃問題 (NLP)，若透過鄰近搜尋法尋求最佳解，則理論上需搜尋目前解的所有線性鄰近解，如此需搜尋的數量將變得相當龐大。但根據上述最佳解為端點解及整數解的特性下，本研究網路問題的可行解區域將可從廣闊的線性解集合濃縮至部分的端點解集合，而此等端點解散落在線性解集合之情況，則與一般整數規劃問題的離散解散落在線性解集合中的特性相似，故可利用一般求解整數線性規劃問題 (ILP) 之組合最佳化的啟發式解法進行求解。

2.2 類和諧搜尋演算法之設計

本問題在數學上可歸類為 NP-hard 問題，其傳統求解最佳化方法的求解時間，將隨問題規模的增大可能呈指數的增加，故學界至今大都發展啟發解以求解實際的大型問題。另一方面，由於傳統的和諧演算法雖具有架構簡明、操作容易等優點，但針對求解含凹形節線成本之最小成本網路流動問題時，因其可行解為一伸展樹 (spanning tree)，故於產生和弦組合時，並無法直接使用傳統和諧演算法中的調音機制以產生新的和絃組合。此外，傳統的和諧演算法亦有深度搜尋較弱的缺點，故本研究將針對本問題之特性，加入 PSO、ACS 與 TA 的部分機制，強化傳統和諧演算法在求解含凹形節線成本最小成本轉運問題之效率與品質。其加入 PSO、ACS 與 TA 的部分機制說明如下：

1. 粒子群演算法 (PSO) ：此演算法視每個可行解為一個體，而利用每次求解過程中目前族群中的最佳解與以往經歷過的最佳解，調整個體的位移，因此具有快速向較佳解靠近的優勢；但本演算法只和族群中的目前最佳個體與歷史最佳記錄作比較，較易喪失跳脫局部最佳解的能力，故本研究將採用粒子群演算法的速度更新公式，並配合所設定的和弦記憶考慮機率 (harmony memory considering rate；HMCR) 值，進行伸展樹中的節線更新工作，以產生較佳的新可行解。
2. 螞蟻族群演算法 (ACS)：本演算法較適合設計於求解巡迴推銷員問題 (travelling salesman problem, TSP) 問題 (Dorigo 與 Gambardella [22])，故此演算法架構較不適合求解含凹形節線成本之最小成本轉運問題；但螞蟻族群演算法中的「開發」 (exploitation)

與「探索」 (exploration) 兩種搜尋機制，對於區域更新與全域更新具有頗佳的求解效率 (Yan 等人 [27])，故本研究將採用螞蟻族群演算法的費洛蒙更新公式，並結合和諧演算法中的調音機制 (pitch adjusting)，以更新和弦記憶空間。

3. 門檻值接受法 (TA)：本演算法為一改善起始解的交換法，其主要精神在於讓較差之可行解能有機會於後續的求解步驟中，在可容許變差的門檻值內，跳出局部最佳解的空間，進而搜尋更佳的新可行解；但此演算法只侷限於區域搜尋與交換，無法進行更有效率的廣度搜尋，故本研究將採用 TA 機制，並配合和諧演算法的和弦記憶空間調整，讓較差之可行解有一定機率能保留於和弦記憶空間中，有機會於後續過程中產生更佳的可行解。

綜上所述，本研究將發展以和諧搜尋演算法為基礎架構並結合 PSO、ACS、TA 之類和諧搜尋演算法，再比較其與 Yan 等人 [25] 之 GA、GDA、TA，顏上堯等人 [26] 與 Yan 等人 [28] 兩種不同之粒子群最佳化演算法 (APSO)，以及 Yan 等人 [27] 之類螞蟻族群演算法 (AACS) 的求解績效。

2.2.1 演算法步驟

因和諧搜尋演算法擁有簡單、明確的架構，然而在求解含凹形節線成本之最小成本網路流動問題時，此架構則顯得較為鬆散，故本研究在和諧搜尋演算法之架構下，依循問題特性與需求加入 PSO、ACS、TA 等部分機制，並進行改良，發展一適合求解凹形節線成本之最小成本網路流動問題之演算法。其求解流程如下：首先，利用隨機與凹形成本特性方式產生足夠的初始群體，並依照其成本值由小到大進行排序，選擇較佳的初始解加入和弦記憶空間中並將之填滿。接著，將和弦記憶空間中各組解之節線，依序利用 PSO 機率更新公式進行節線機率更新，再以此機率值決定各組欲加入之非伸展樹節線，當加入的節線達到設定之數量後，則各組皆得一新可行解。此時可依據所設定的和弦記憶考慮機率 (harmony memory considering rate；HMCR) 值，由和弦記憶空間中產生數對較佳的供需節點對，並依循原路徑進行配送，其餘未配送完成之需求量則由隨機方式完成配送，若新可行解組成方式以和弦記憶空間為母體，則進行變數調整策略。此策略以和弦演算法之調音機制 (pitch adjusting) 為基礎，結合螞蟻演算法依循費洛蒙濃度選擇路徑之機制，選出數條未選取之節線，以調音機率值 (pitch adjusting rate, PAR) 決定該節線是否加入可行伸展樹中。待完成變數調整策略後，再進行鄰近搜尋法，即可得到一組全新的解。將此新可行解目標值與和弦記憶空間中之解比較，若新解目標值優於和弦記憶空間中任一解，則將此新可行解加入和弦記憶空間中，並將原和弦記憶空間中最後一組解刪除。為跳離區域最佳解，讓母體更加廣泛，本研究也在此階段加入 TA 機制，讓較差之可行解有一定機率能保留於和弦記憶空間中，以期於後續步驟中能有機會產生更佳的新可行解。最後，重覆上述步驟直到終止條件滿足。本演算法之詳細求解步驟如下 (其演算法架構如圖 2 所示)：

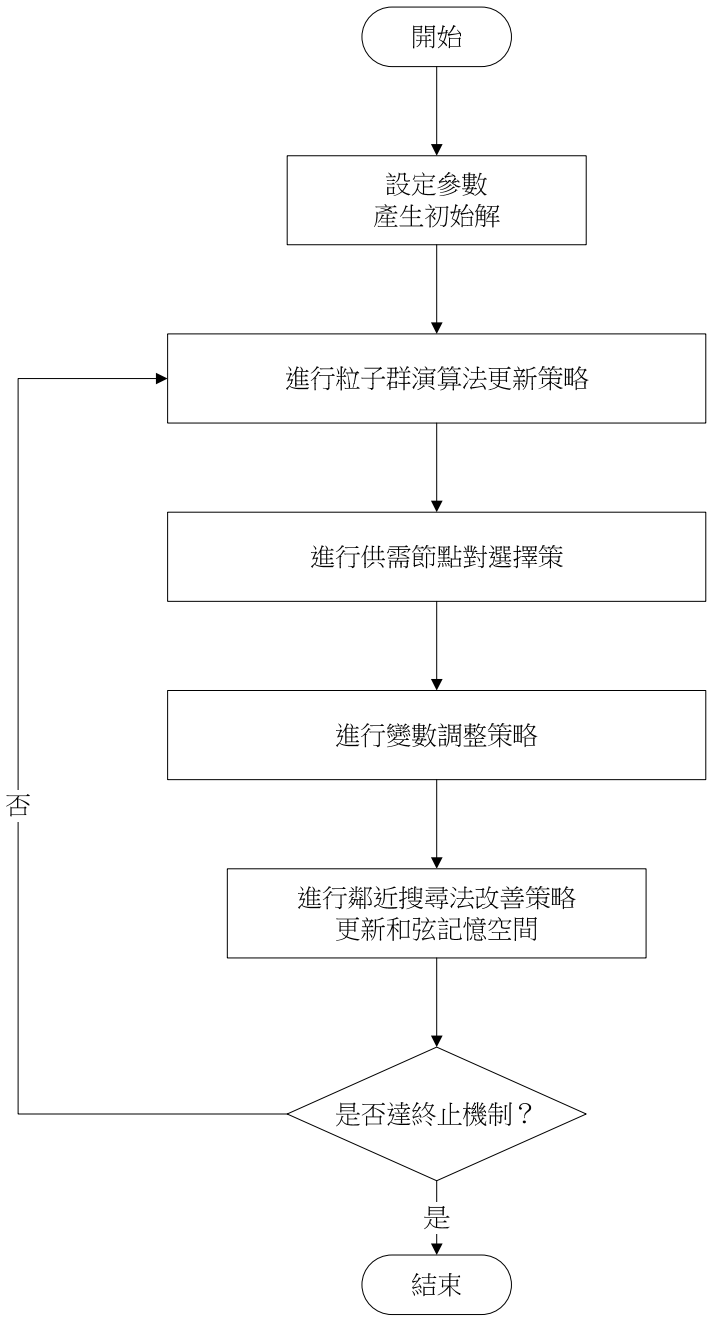

圖 2　類和諧搜尋演算法 (AHS) 整體架構流程圖

步驟 1　設定所有參數。

步驟 2　進行 PSO 加入節線策略 (詳見 2.2.3 節)，以 PSO 速度更新公式選擇節線加入，分別更新和弦記憶空間中各組可行解。各組解更新方式皆相同，以下僅說明一組的

更新步驟。

2.1 利用 PSO 機率更新公式 (詳見 2.2.3 節) 及全體與個體暫時最佳解更新各組解內各節線機率值。

2.2 設定增加至上一回合可行伸展樹的節線數量為 N_a 。

2.3 將節線集合分成伸展樹與非伸展樹兩節線集合。

2.4 以上一回合伸展樹為基礎，從非伸展樹節線集合內挑選節線加入此伸展樹。其節線挑選步驟如下：

2.4.1 記錄和弦記憶空間中各組可行伸展樹每一節線總選取次數。

2.4.2 依照步驟 2.4.1 統計，分別給予各節線一機率值，各節線機率值為所有節線選取總次數除以由第一條節線累積至該節線選取總次數，和弦記憶空間中選取次數越多，其機率越大。

2.4.3 產生一 0 至 1 的隨機值 r，檢視此隨機值落點，以決定選擇的節線。

2.4.4 針對此節線隨機產生 0 至 1 之隨機值 V_i。

2.4.5 判斷是否 $S(v_i)=\dfrac{1}{1+(e^{-V_i})}>r$ ？若是，則進行步驟 2.4.6；否則回到步驟 2.4.3。

2.4.6 將節線 i 加入可行伸展樹中，判斷加入節線數是否等於 N_a ？若是，則進行步驟 2.5；否則進行步驟 2.4.3。

2.5 利用流量推擠法 (Yan 與 Young [37]) 將路徑推成一可行伸展樹。

步驟 3　進行供需節點對選擇策略 (詳見 2.2.4 節)，產生一組新可行解。

3.1 記錄和弦記憶空間中各組可行伸展樹其供需節點對。

3.2 產生一 0 至 1 的隨機值 Ran，若 $Ran \le HMCR$ ，則進行 3.3；否則進行 3.4。

3.3 從母體中選擇數對供需節點對與其路徑分配流量。

3.3.1 設定欲從母體中選取的供需節點對數 N_p。

3.3.2 產生新的虛擬節線成本。

3.3.3 給予所有選取過之供需節點對一機率值，其機率為所有供需節點對選擇總次數除以累加至該供需節點對之選擇次數。和弦記憶空間中選擇的次數越多，則機率越大。

3.3.4 隨機產生 0 至 1 之機率值 r，以此隨機值檢視此隨機值落點，決定欲選擇加入之供需節點對。

3.3.5 利用步驟 3.3.2 產生的虛擬節線成本決定該供需節點對之路徑。

3.3.6 判斷加入的供需節點對是否等於 N_p？若是，則進行步驟 3.3.7；若否，則進行步驟 3.3.4。

3.3.7 將剩餘未分配之流量以隨機的方式分配完成，其使用之節線成本亦為虛擬節線成本。

3.3.8 利用流量推擠法將路徑推成一可行伸展樹。

3.4 跳脫母體搜尋新的可行解。

3.4.1 產生 0 至 1 的隨機值 r，若 $r \le 0.5$ ，則進行步驟 3.4.2；若 $r > 0.5$ ，則進行步驟 3.4.3。

3.4.2 以隨機方式產生一組新的初始解，進行步驟 3.4.4。

3.4.3 依凹型成本特性產生一組新的初始解，進行步驟 3.4.4。

3.4.4 利用流量推擠法將路徑推成一可行伸展樹。

3.4.5 利用鄰近搜尋法 (詳見 2.2.7 節) 找出區域最佳解。

步驟 4 進行變數調整策略 (詳見 2.2.5 節)，針對供需節點對選擇策略所產生的可行解進行微調。

4.1 設定欲調整節線數 N_t。

4.2 將節線分成伸展樹與非伸展樹兩集合。

4.3 利用費洛蒙更新策略更新各節線之費洛蒙濃度。

4.4 產生 0 至 1 的隨機值 r ，若 $r \le Q_0$ （Q_0 為自行設定之參數值，$0 \le Q_0 \le 1$），則進行步驟 4.5；若 $r > Q_0$ ，則進行步驟 4.6。

4.5 選擇費洛蒙濃度與節線成本參數乘積最大之節線加入，進行步驟 4.7。

4.6 進行狀態轉移策略，選擇非伸展樹節線加入，進行步驟 4.7。

4.7 判斷調整的節線是否等於 N_t？若是，進行步驟 4.8；若否，則進行步驟 4.3。

4.8 利用流量推擠法將路徑推成一可行伸展樹。

4.9 利用鄰近搜尋法找出區域最佳解。

步驟 5 更新和弦記憶空間。

步驟 6 演算法是否達到結束機制 (詳見 2.2.8 節)？若是，則結束演算法；若否，則回到步驟 2。

上述 AHS 演算法中所包含初始解產生策略、粒子群演算法更新策略、供需節點對選擇策略、變數調整策略、門檻接受法改善策略、鄰近搜尋法改善策略與演算法終止機制等相關要素，如下所述。

2.2.2 初始解產生策略

由於和諧搜尋演算法包含全域搜尋與區域搜尋法兩部分，在初始群體方面需要產生出多個可行解，為使往後搜尋之新可行解不被侷限於一區域，初始群體產生時期望能廣泛的散佈於可行解範圍內。因此，本研究參考 Yan 等人 [25] 提出兩種啟發式可行解產生策略以產生初始群體，分別為針對凹形成本特性產生初始解與隨機方式產生初始解。在初始群體組成中，本研究綜合兩種初始解產生方式，以機率值決定由上述任一種方式產生初始解。本研究經過多次的測試後，將其比例控制在依凹型成本特性產生與隨機配對各半的狀態。

2.2.3 粒子群演算法更新策略

為增加尋找較佳可行解之成效與效率，本研究在此階段採用粒子群演算法中的速度更新公式，用以更新各節線機率與更新和弦記憶空間內的各組可行解。在更新各節線機率方面，為了讓 PSO 速度更新公式適用於凹型成本之網路問題，本研究參考 Yan 等人 [27] 所發展與以往不同之機率值之更新方式。其作法如下：

步驟 1：將各粒子之節線集合，分成伸展樹與非伸展樹上之兩個節線集合。

步驟 2：若為非伸展樹之節線集合，則依全體與個體最佳解之各變數值，並以式 (4) 進行機率值之更新。

步驟 3：若為伸展樹之節線集合，則除依全體與個體最佳解之各變數值外，並依目前與前一回合目標函數之比值，以式 (5) 進行機率值之更新。

$$v_{id}^{new} = v_{id}^{old} + c_1 * rand_1() * p_{id} + c_2 * rand()_2 * g_d \tag{4}$$

$$v_{id}^{new} = \omega * v_{id}^{old} + c_1 * rand_1()(p_{id} - x_{id}) + c_2 * rand()_2(g_d - x_{id}) + (1-w) * \frac{F_i^{old}}{F_i^{new}} * v_{id}^{old} \tag{5}$$

其中，i：第 i 個粒子，即第 i 組可行解。

d：第 d 個搜尋空間的維度，即表示第 d 條節線 (或最短路徑)。

ω：慣性權重值，$0 \le \omega \le 1$。

v_{id}^{new}：在第 i 個粒子內，第 d 條節線 (或最短路徑) 更新後之機率值。

v_{id}^{old}：在第 i 個粒子內，第 d 條節線 (或最短路徑) 目前之機率值。

x_{id}：在第 i 個粒子內，第 d 條節線 (或最短路徑) 目前的使用情況。

p_{id}：在第 i 個粒子的個體暫時最佳解內，第 d 條節線 (或最短路徑) 的使用情況。

g_d：在全體暫時最佳解內，第 d 條節線 (或最短路徑) 的使用情況。

F_i^{old}：第 i 組可行解目前的目標值。

F_i^{new}：第 i 組可行解更新後的目標值。

c_1、c_2：學習因子。

$rand_1()$、$rand_2()$：介於 0 和 1 之間的隨機變數。

而在更新和弦記憶空間中各組解更新方面，本研究在演算法每一回合開始時，皆利用 PSO 加入節線方法將和弦記憶空間中各組解更新。更新完成後，再以更新過之可行解為母體進入供需節點對選擇策略，以期未來能產生較佳可行解。此階段和弦記憶空間中的各組可行解將同時進行下述步驟：利用上述粒子群機率更新公式以更新各節線之機率，並決定欲加入之節線數，接著以機率值方式選取非伸展樹上之節線，再利用邏輯特函數決定此節線是否加入可行伸展樹中；若是，則將此節線加入可行伸展樹；若否，再選擇下一條節線進行判斷。重覆上述步驟，直到已加入之節線數到達先前設定欲加入之節線數，再利用流

量推擠法將迴圈消除，形成更新後之可行伸展樹。

2.2.4 供需節點對選擇策略

在類和諧搜尋演算法架構中，當已存一組和弦記憶空間之解當母體後，將依序選出各決策變數之值，其選擇方式可由母體中已選擇之決策變數選出，或由母體中未出現之數值選出，再以這些選擇出之決策變數組合成一組新的解。然而，凹形成本問題之決策變數值僅為 1 或 0，故本研究將以此架構為基礎，發展由母體中選出供需節點對之選擇策略。由於選擇策略的目的為產生一組新的可行解，故本研究在演算法中以和弦記憶空間中所有可行解為母體，以機率值方式挑選出數對供需節點對，當完成挑選之供需節點對達設定數量，其餘未配送之貨物則以隨機選擇供需節點對方式繼續完成運送。若新可行解是在此機制下產生，則再進行下個階段之變數調整策略。

此外，在選擇較佳供需節點對後，為同時保留較佳路徑，本研究將改變原有之節線成本，以虛擬節線成本尋找後續節線路徑。虛擬節線成本產生的方式如下：記錄和弦記憶空間中各節線被選取的次數，該節線新的節線成本則為原始節線成本減掉該節線在和弦記憶空間中被選取次數，如此將能反應出各節線之優劣。另當虛擬節線成本小於 0 時，為避免之後路徑選擇時形成負成本迴圈，因此強制該節線成本為 0。此路徑選擇方式除可選擇成本較少的節線外，亦能使貨物盡量由同一節線運送，以符合凹型成本之特性，進而降低貨物運送成本，期得到更好之可行解。

2.2.5 變數調整策略

在組成新的可行解後，若此可行解是經由和弦記憶空間中之元素組合而成的，則再進行變數調整策略，其目的是為了進行全域搜尋後之區域搜尋。本研究之變數調整策略是以和諧搜尋演算法中調音機制為基礎進行改良，在一般的和諧搜尋演算法中，調音機制分為兩種，一為針對連續型的變數，另一則是針對離散型的調音機制。然而一般的調音機制並不適用於凹形成本問題，因此本研究為更加有效率換得較佳節線以加到可行伸展樹中，故採用螞蟻演算法中循費洛蒙更新濃度之方法，以找出較佳之交換節線。

以往螞蟻演算法之研究大都採用線性機率模式以定義狀態轉移策略 (如參考 Dorigo 與 Gambardella [22])，並且存在「開發」 (exploitation) 與「探索」 (exploration) 兩種搜尋機制，因此本研究將狀態轉移策略融合到和諧搜尋演算法之調音機制中，在設計上也保留了「開發」與「探索」兩種機制，並引用費洛蒙濃度決定需調整之節線。然而，為了避免無上限的費洛蒙使後續螞蟻尋找的路線陷入區域解，本演算法中也模擬了自然界中費洛蒙消散的自然現象，在費洛蒙更新公式中增加了費洛蒙的消散機制。此機制能增加探索其他路徑的機會，讓可行解範圍更加廣泛。費洛蒙蒸散率對費洛蒙濃度影響極大，若蒸散率太高，將失去費洛蒙累積應有的功能，若蒸散率太低，則之前搜尋的節線費洛蒙濃度會過高，容易陷入區域解。綜合上述，本研究參考 Yan 等人 [28] 針對凹形成本流量越大會使成本越低之特性發展之費洛蒙更新公式，並針對整個伸展樹，與以往針對路徑或路段更新有極大

不同。式 (6) 為全域費洛蒙更新公式，目的為依流量大小給予不同的費洛蒙值更新，增加其節線上之費洛蒙濃度。式 (7) 為區域費洛蒙更新公式，目的為透過流量修正讓流量較高的節線減少節線上費洛蒙濃度。

$$\tau(r,s)=(1-\alpha-\mu)\cdot\tau(r,s)+\alpha\cdot(L_{best})^{-1}+\mu\cdot x_{ij}^{2} \tag{6}$$

其中，α ：費洛蒙衰退參數，$0\le\alpha\le1$。

L_{best} ：所有螞蟻搜尋出之可行解中的最佳解。

μ ：避免加入流量而造成的數值誤差參數。

$$\tau(r,s)=(1-\rho-\mu)\cdot\tau(r,s)+\rho\cdot\tau_0+\mu\cdot x_{ij}^{2} \tag{7}$$

其中，ρ ：費洛蒙衰退參數，$0\le\rho\le1$。

τ_0 ：費洛蒙之起始值。通常設定 $\tau_0=(n\cdot L_{nn})^{-1}$，$n$ 為總節點數，L_{nn} 乃是以初始解所得到之目標值。

μ ：避免加入流量而造成的數值誤差參數。

由於本凹型成本網路問題之決策變數為 0 或 1 變數，在變數調整策略下，本研究將決策變數由非伸展樹節線 (變數值為 0) 調整成伸展樹節線 (變數值為 1)。由於可行伸展樹若加入一條新的節線，將會使可行伸展樹產生一迴圈，因此可使用流量推擠法將之推擠成可行伸展樹。其調音機制步驟如下：首先，利用式 (6) 與式 (7) 之費洛蒙更新公式計算各節線費洛蒙濃度，將節線分為新可行解中伸展樹節線集合與非伸展樹節線集合，並產生一隨機值，決定選擇用「開發」 (exploitation) 或「探索」 (exploration) 機制。若選擇「開發」機制，則選擇費洛蒙濃度與節線成本參數之乘積最高的非伸展樹節線加入可行伸展樹；若選擇「探索」機制，則隨機選擇一非伸展樹節線，並進行該節線是否加入可行伸展樹之判斷。之後，重覆上述步驟直到已調整的節線達到設定量，則停止變數調整機制。變數調整完成後即完成本演算法中區域搜尋之部分。

2.2.6 門檻接受法改善策略

此演算法在產生一組新可行解後，會判斷此可行解是否能儲存於和弦記憶空間中。然而，若和弦記憶空間中的解太過相近，則可能使新產生的可行解陷入區域解。為避免演算法太快陷入區域解，本研究在產生一組新的可行解後，引用 TA 門檻值的觀念，讓較差的可行解也有機率能夠加入到和弦記憶空間中；其門檻接受法改善策略之步驟如下：首先設定起始門檻值 θ 為一初始百分比，當搜尋到之可行解目標值小於和弦記憶空間中最後一組解之目標值×(1+θ) 時，則將新的可行解插入和弦記憶空間中。經過一定回合數後，若目標值無改善時，則降低 θ 值，再重複上述過程，直到 θ 為 0 後，均以 $\theta=0$ 進行個體最佳解的更新。此作法期能避免搜尋快速陷入區域最佳解之情況，提升演算法求解之績效。

2.2.7 鄰近搜尋法改善策略

由於本研究演算法中的「加入節線策略」、「供需節點對選擇策略」與「變數調整策略」都屬於隨機性之搜尋，故有可能在尚未搜尋到區域最佳解 (local optimum) 時，即跳脫至另一可行解區域。有鑑於此，本研究配合鄰近搜尋法，提升演算法之搜尋深度，以有效改善可行解之品質。在鄰近搜尋法的設計上，本研究以隨機選取的方式選擇非伸展樹的節線加入可行伸展樹中，並在加入的節線可得到更好的解時，進行可行伸展樹的更新與取代，直到鄰近搜尋的回合數結束為止。當選擇加入一條非伸展樹的節線後，將形成有迴圈的伸展樹，此時可利用流量推擠法 (Yan 與 Young [37]) 以推導出可行伸展樹。

2.2.8 演算法終止機制

一般演算法終止機制可分為固定回合數、固定運算時間與連續未改善回合數等。以往處理含凹形成本最小成本網路問題時，則以固定 500 回合 (Yan 等人 [25,27,28] 及顏上堯等人 [26]) 為演算法終止條件。此終止機制能控制演算法進行時間，卻也抑制了演算法繼續求得更佳解之可能性。本研究為增加求得較佳解的機會，將終止機制設計為基本回合數 500 回合，若 450 回合後還能改善目前解，則再增加 100 回合，直到連續 100 回合未改善目前解為止。

三、測試分析

為測試本研究所提出類和諧演算法之求解績效，本研究選取 Yan 等人 [25] 發展的隨機網路產生器所產生的 12 種網路類型問題，作為本次測試之基礎。本研究使用 C++語言撰寫程式並測試作為演算法實證，測試環境為 Intel Core 2 Quad CPU Q6600 2.4GHz，1Gb RAM 之個人電腦。此外，本研究參考 Yan 等人 [25] 所發展之 GA 與其他區域式搜尋演算法、Yan 等人 [27] 所發展 AB-APSO 粒子群最佳化演算法，及 Yan 等人 [28] 所發展之類螞蟻族群演算法，以作為測試本研究 AHS 演算法求解績效之比較。

各演算法參數部分，本研究之 AHS 共有初始群體個數 (M_n)、初始群體型態 (M_t)、和弦記憶空間大小 (*HMS*)、和弦記憶空間使用率 (P_{HM})、和弦記憶空間考慮率 (*HMCR*)、變數調整數量 (N_{aa})、PSO 演化次數 (N_{pe})、PSO 加人節線數 (N_{pa})、粒子初始速度 (V_i)、PSO 速度更新策略-慣性權重值 (ω)、PSO 速度更新策略-學習因子 1 (c_1)、PSO 速度更新策略-學習因子 2 (c_2)、ACS 費洛蒙更新策略-權衡因子 (Q_0)、ACS 費洛蒙更新策略-全域更新蒸發率 (α)、ACS 費洛蒙更新策略-區域更新蒸發率 (ρ)、門檻更新策略-起始門檻值 (T_p)、門檻更新策略-門檻下降參數 (T_{pk}) 與鄰近搜尋範圍 (N_e) 等 18 種參數；HS 方面共有和弦記憶空間大小 (*HMS*)、和弦記憶空間考慮率 (*HMCR*)、和弦記憶空間使用率 (P_{HM})、鄰近搜尋範圍 (N_e) 等 4 種參數；粒子群最佳化演算法方面，總共有粒子數 (P)、凹形初始解比率 (P_m)、PSO 速度更新策略-慣性權重值 (ω)、PSO 速度更新策略-學習因子 1 (c_1)、PSO

速度更新策略-學習因子 2 (c_2)、門檻全體更新策略-起始門檻值 (T^g)、門檻全體更新策略-數列長度 (T_k^g)、門檻全體更新策略-未變動回合數 (T_l^g)、門檻個體更新策略-起始門檻值 (T^p)、門檻個體更新策略-數列長度 (T_k^p)、門檻全體更新策略-未變動回合數 (T_l^p)、輪盤分組數 (N_r) 與鄰近搜尋範圍 (N_e) 等 13 種參數；AACS 總共有人工螞蟻搜尋數 (N_t)、濃度更新設定參數值 (Q_0)、費洛蒙全域衰退參數 (α)、費洛蒙距離參數 (β)、費洛蒙區域衰退參數 (μ)、費洛蒙更新率 1 (θ_1)、費洛蒙更新率 2 (θ_2)、搜尋範圍 (N_a)、較佳路徑保留率 (P_t)、鄰近搜尋率 (P_c)、回合較佳解保留率 (P_b) 與較佳解取代門檻 (e) 等 14 種參數；GA 共有群體大小 (I_k)、優良解複製率 (U_{rep})、加入群體率 (U_s)、直接搜尋率 (D_r)、突變率 (C_r)、交換節線率 (A_{cr})、搜尋範圍 (N) 與輪盤區分組 (RW_i) 等 8 種參數；而各區域搜尋法共有門檻更新策略-起始門檻值 (T_p)、門檻值數列長度 (T_h)、候選節線數 (S)、未改善次數 (Δ)、回合數 (C_t) 與禁制串列長度 (T_f) 等 6 種參數 (Yan 等人 [25, 27, 28])。

3.1 AHS 求解策略

AHS 以和諧搜尋演算法為主要架構，再結合 PSO 與 ACS 中較好的機制所發展之求解演算法。AHS 演算法中各機制皆有其控制參數，因此需針對各類參數設定適當的值，方能求得較佳解。如和弦記憶空間大小 (HMS)、粒子初始速度 (V)、供需節點對選擇比例 (P_r) 等參數均會影響求解品質。本研究經過多次的測試與經驗的累積，先找出一組求解績效不錯的參數組合，以作為後續測試之設定參數。其中初始群體數量 (M_n) 為 200、初始群體組成型態 (M_t) 為 0.5、和弦記憶空間大小 (HMS) 為 15、和弦記憶空間使用率 (P_{HM}) 為 1.5、和弦空間使用率 ($HMCR$) 為 0.9、變數調整次數 (N_{aa}) 為 10、PSO 演化次數 (N_{pe}) 為 3、PSO 加入節線數 (M_{pa}) 為 3、粒子初始速度 (V_i) 為 20、PSO 慣性權重值 (ω) 為 0.7、學習因子 (c_1, c_2) 為 2 與 3、ACS 權衡因子 (Q_0) 為 0.8、全域更新蒸發率 (α) 為 0.2、區域更新蒸發率 (ρ) 為 0.4、起始門檻值 (T_p) 為 0.3%、門檻值下降參數 (T_{pk}) 為 20、鄰近搜尋範圍 (N_e) 為 200。此外，本研究希冀以較佳的組合參數，進行後續演算法間之績效比較，故以上述的參數值 (設定為 AHS-2 組合方案) 為基礎進行延伸，針對各個參數進行參數測試，共設計出 45 種參數組合進行網路測試，其各種求解策略之參數組合如表 1 所示。舉例而言，第 1 個求解策略「初始群體個數 (M_n)」在最初的參數組合 (AHS-2) 設定數量為 200，故可調整數量值，將初始群體個數 (M_n) 設為 100 的參數組合定為 AHS-1 (其餘參數值仍同於 AHS-2)，即為第 2 個求解策略，而將初始群體個數 (M_n) 設為 300 的參數組合定為 AHS-3 (即為第 3 個求解策略) 進行測試比較，而在「初始群體型態 (M_t) 」方面，最初的參數組合 (AHS-2) 設定值為 0.5，故可調整數量值，將初始群體型態 (M_t) 設為 0 的參數組合定為 AHS-4 (其餘參數值仍同於 AHS-2)，即為第 4 個求解策略，而將初初始群體型態 (M_t) 設為 1.0 的參數組合定為 AHS-5 (即為第 5 個求解策略) 進行測試比較，其他求解策略亦依照此方式產生後進行測試比較。值得注意的是，本研究所做測試僅針對含凹形成本最小成本網路流動問題，至於其他類問題則需另外測試並以驗證。

表 1　各種求解策略參數組合之彙總表

求解策略	參數名稱	參數組合
初始群體個數	M_n	AHS-1、AHS-2、AHS-3
初始解群體型態	M_t	AHS-2、AHS-4、AHS-5
和弦記憶空間大小	HMS	AHS-2、AHS-6、AHS-7
和弦記憶使用率	P_{HM}	AHS-2、AHS-8、AHS-9、AHS-10、AHS-11
和弦記憶空間考慮率	$HMCR$	AHS-2、AHS-12、AHS-13
變數調整次數	N_{aa}	AHS-2、AHS-14、AHS-15
PSO 演化次數	N_{pe}	AHS-2、AHS-16、AHS-17
PSO 加入節線數	N_{pa}	AHS-2、AHS-18、AHS-19
PSO 粒子初始速度	V_i	AHS-2、AHS-20、AHS-21、AHS-22、AHS-23
PSO 速度更新策略-慣性權重值	ω	AHS-2、AHS-24、AHS-25
PSO 速度更新策略-學習因子 1	c_1	AHS-2、AHS-26、AHS-27、AHS-28
PSO 速度更新策略-學習因子 2	c_2	AHS-28、AHS-29、AHS-30、AHS-31
ACS 費洛蒙更新策略-權衡因子	Q_0	AHS-2、AHS-32、AHS-33
ACS 費洛蒙更新策略-全域更新蒸發率	α	AHS-2、AHS-34、AHS-35
ACS 費洛蒙更新策略-區域更新蒸發率	ρ	AHS-2、AHS-36、AHS-37
門檻值更新策略-起始門檻值	T_p	AHS-2、AHS38、AHS-39
門檻值更新策略-門檻值下降參數	T_{pk}	AHS-2、AHS-40、AHS-41
鄰近搜尋法	N_e	AHS-2、AHS-42、AHS-43
最佳參數組合		AHS-44、AHS-45

此外，本研究亦以顏上堯等人 [26] 及 Yan 等人 [27] 利用 PSO 演算法所求得之目前最佳解為基礎，在各種不同的網路類型情況下，同時進行各方案所求得最小目標值與平均目標值的比較。其中在網路類型方面，本研究參考 Yan 等人 [25]，以節點數 100、150、300、500 為基礎，搭配 3 種控制供需點密度 D=0.2、0.5、0.8 之參數，共形成 12 種網路類型進行測試 (100_02 表示 100 個節點數與供需點密度為 0.2)，而各網路類型目前最佳解如表 2 所示；另測試結果將以誤差百分比 (誤差百分比＝(所得的最佳目標值–文獻目前最佳解)／文獻

目前最佳解) 表示，其目的在評估各方案於不同網路時的測試績效，而誤差百分比越小，代表該方案在該網路所得到的求解績效越好。

表 2　各網路類型目前最佳解

網路類型	目前最佳解	演算法 (文獻)
100_02	30847.87	AB-APSO　(Yan 等人[28])
100_05	46966.13	AB-APSO　(Yan 等人[28])
100_08	53664.02	AB-APSO　(Yan 等人[28])
150_02	42015.22	AB-APSO　(Yan 等人[28])
150_05	74066.99	AB-APSO　(Yan 等人[28])
150_08	77158.77	AB-APSO　(Yan 等人[28])
300_02	65402.45	PB-APSO　(顏上堯等人[26])
300_05	117064.49	AB-APSO　(Yan 等人[28])
300_08	177059.67	AB-APSO　(Yan 等人[28])
500_02	121283.34	AB-APSO　(Yan 等人[28])
500_05	218377.86	AB-APSO　(Yan 等人[28])
500_08	304001.68	AB-APSO　(Yan 等人[28])

資料來源：顏上堯等人 [26] 及 Yan 等人 [27]。

本研究利用上述所產生的 45 組參數組合，針對 12 種網路類型分別進行求解測試後，發現其中 4 組參數組合 AHS-12, AHS-39, AHS-43 與 AHS-45 之求解績效較其他組合為佳，即在所有測試網路的求解結果中，此 4 組參數組合在 12 個網路類型的求解測試中，均至少有 4 個網路類型測試結果的誤差百分比小於零 (其中 AHS-12 誤差百分比小於零次數為 5 次，AHS-39 為 4 次，AHS-43 為 5 次，AHS-45 亦為 4 次)，顯示此 4 組參數組合具有極佳的問題求解效果 (誤差百分比數值請參閱表 3)，有鑑於此，本研究將以 AHS-12, AHS-39, AHS-43 與 AHS-45 等 4 個參數組合，作為本研究所提出 AHS 求解演算法之測試方案，並與其他演算法進行求解績效之比較分析。

3.2 AHS 與 APSO、AACS、GA 及各區域搜尋法之求解績效比較

本研究針對含凹形節線成本與具方向性之網路問題，提出全域式搜尋演算法之求解策略與表現較佳之參數組合後，將更進一步測試本研究所提出 AHS 求解演算法與 HS、APSO、AACS、GA 及各區域搜尋法之求解績效。本研究將以前述參數測試結果較佳的 AHS-12, AHS-39, AHS-43 與 AHS-45 等 4 個參數組合，作為 AHS 求解演算法參與績效測試之方案。關於其他測試演算法的選取，在 GA 與各區域演算法部分，本研究選取 Yan 等人 [25] 針對本問題所發展的全域與區域搜尋演算法，其中全域演算法為遺傳演算法

(GA)，區域搜尋法則是包括了門檻值接受法 (TA)、大洪水法 (GDA)、結合禁制搜尋策略門檻值接受法 (TTA)、結合禁制搜尋策略大洪水法 (TGDA) 等 4 種；在 ACS 演算法部分，本研究將選取 Yan 等人 [28] 針對本問題所發展的 AACS 全域搜尋演算法；在 PSO 演算法方面，則選取 Yan 等人 [27] 針對本問題所發展之 AB-APSO 全域搜尋演算法，值得一提的是，由於顏上堯等人 [26] 所發展 PB-APSO 的測試績效，普遍較 Yan 等人 [28] 所發展 AB-APSO 為差，故為測試方便，本研究將以 Yan 等人 [28] 所發展 AB-APSO 演算法為基礎進行測試。此外，為了解純 HS 演算法對於本問題之求解績效，本研究亦對 HS 進行測試。在演算法設計方面只單純使用 HS 之原架構，並加入鄰近搜尋法。因此，本研究將以上述 8 種演算法，作為與本研究 AHS 演算法進行求解績效比較之對象。

關於上述各演算法測試方案的選取上，在全域式求解演算法方面，本研究將依據文獻中針對 GA、ACS 與 PSO 進行測試後所產生的「最佳解參數組合」、「最佳平均解參數組合」與「最佳回合數參數組合」，作為該演算法進行此次測試的 3 個方案 (亦即每個全域式演算法均有 3 個測試方案)，如此將可在相同的基礎上，與本研究所發展 AHS 演算法進行績效測試比較；而在區域式求解演算法方面，由於文獻中顯示 TA、TTA、GDA 與 TGDA 的測試結果較為不佳，故本研究選取文獻測試結果中產生的「最佳解參數組合」，作為該演算法進行測試的唯一方案。

在測試方面，本研究除比較各方案之最佳解外，亦對各方案 30 次求解之平均解進行評估。因本研究演算法終止機制不同於上述文獻之演算法以 500 回合為終止機制，而是以連續未改善回合數為終止機制，因此在大規模網路時求解回合數常超過 500，進而增加求解時間。為在相同基礎下進行測試績效之比較，本研究將其演算法之中止機制，改跟本研究演算法的中止機制一樣，且若求解時間達本研究演算法所提出方案最佳解之最長求解時間時，亦終止演算法。同時針對各演算法之測試方案均測試 30 次，並選取 30 次測試中表現最佳的結果 (誤差百分比最低)，作為其演算法之最佳解求解結果，亦將 30 次求解結果進行平均，分別與文獻中目前最佳解進行比較，如此比較的結果可在較相同的基準上。

各求解演算法最佳解求解結果列於表 3、平均求解結果列於表 4，而本研究所提出 4 組 AHS 方案於各網路類型之求解時間如表 5 所示。由表 3 知，各演算法不同方案求解 30 次所得最佳解求解績效以 AHS-45 最佳，其平均誤差為–0.125，在求解 150_05 以下之網路 AHS 4 個參數組合方案皆能達到文獻之最佳解，另 AHS-12 在求解 150_08、300_02、500_02、500_05 與 500_08 時都能找到優於文獻中目前之最佳解；AHS-39 在求解 300_05、300_08、500_02 與 500_08 時都能找到優於文獻中目前之最佳解；AHS-39 在求解 150_08、300_05、300_08、500_02 與 500_08 時都能找到優於文獻中目前之最佳解；AHS-45 在求解 300_02、500_02、500_05 與 500_08 時都能找到優於文獻中目前之最佳解。在其他演算法方面，APSO 在求解時間拉長的情況下，亦有機會改善較大網路之解，但改善幅度卻比 AHS 小。AACS、GA 與 4 種區域搜尋法之求解績效較差，AACS 與 GA 只能在求解小網路問題時有機會求得最佳解，而區域搜尋則無法在網路規模 100 以上之問題求得最佳解。HS 求解績效雖較 AACS 與 GA 佳，但在多數網路卻無法突破 APSO 之求解結果。

表 3　各演算法最佳解之求解結果

網路類型	AHS-12	AHS-39	AHS-43	AHS-45	HS	APSO-1	APSO-2	APSO-3	AACS-1
100_02	0.000%	0.000%	0.000%	0.000%	0.000%	0.000%	0.000%	0.000%	1.053%
100_05	0.000%	0.000%	0.000%	0.000%	0.195%	0.000%	0.000%	0.000%	0.016%
100_08	0.000%	0.000%	0.000%	0.000%	0.000%	0.000%	0.000%	0.000%	1.148%
150_02	0.000%	0.000%	0.000%	0.000%	0.000%	0.000%	0.000%	0.000%	0.000%
150_05	0.000%	0.000%	0.000%	0.000%	0.119%	0.000%	0.000%	0.243%	3.988%
150_08	-0.005%	0.096%	-0.005%	0.218%	0.300%	0.396%	0.416%	0.182%	2.157%
300_02	-0.380%	0.556%	0.681%	-0.429%	0.462%	1.035%	0.566%	1.637%	2.210%
300_05	0.237%	-0.712%	-0.317%	0.234%	0.690%	0.575%	0.773%	0.890%	3.197%
300_08	0.912%	-0.002%	-0.003%	0.241%	2.120%	1.277%	0.982%	0.800%	4.185%
500_02	-1.126%	-1.212%	-1.407%	-0.779%	-0.375%	1.816%	1.820%	2.636%	5.012%
500_05	-0.434%	0.548%	0.634%	-0.120%	3.206%	0.835%	0.798%	1.013%	8.259%
500_08	-0.584%	-0.298%	-0.214%	-0.865%	0.742%	-0.223%	-0.585%	0.077%	7.404%
平均誤差	-0.115%	-0.085%	-0.053%	-0.125%	0.622%	0.476%	0.398%	0.623%	3.219%

表 3　各演算法最佳解之求解結果 (續)

網路類型	AACS-2	AACS-3	GA-1	GA-2	GA-3	TA	TTA	GDA	TGDA
100_02	0.215%	1.971%	1.408%	2.010%	1.436%	2.016%	2.520%	1.822%	3.807%
100_05	3.416%	1.131%	2.582%	1.066%	0.538%	3.300%	1.874%	1.483%	3.254%
100_08	0.184%	0.000%	2.715%	0.000%	0.183%	3.465%	4.430%	2.287%	2.820%
150_02	0.900%	1.215%	2.447%	0.000%	2.182%	3.041%	2.931%	5.074%	4.170%
150_05	2.515%	3.954%	3.099%	3.986%	1.120%	4.161%	7.313%	7.288%	5.825%
150_08	0.782%	2.176%	0.949%	1.814%	1.289%	6.835%	3.443%	4.530%	5.241%
300_02	10.286%	7.340%	1.703%	3.329%	2.370%	4.591%	4.465%	3.803%	10.453%
300_05	4.597%	3.732%	3.765%	3.101%	2.680%	6.027%	7.018%	10.279%	7.051%
300_08	6.048%	5.754%	5.907%	2.910%	3.993%	10.090%	7.952%	6.904%	9.773%
500_02	7.027%	4.722%	0.392%	2.487%	0.786%	8.825%	9.969%	3.608%	5.015%
500_05	5.581%	8.189%	4.799%	1.570%	3.227%	6.685%	4.718%	5.023%	9.206%
500_08	6.652%	7.199%	3.963%	2.739%	3.846%	6.250%	6.643%	4.241%	7.446%
平均誤差	4.017%	3.949%	2.811%	2.084%	1.971%	5.441%	5.273%	4.695%	6.172%

表 4　各演算法平均解之求解結果

網路類型	AHS-12	AHS-39	AHS-43	AHS-45	HS	APSO-1	APSO-2	APSO-3	AACS-1
100_02	0.177%	0.230%	0.529%	0.103%	0.195%	0.190%	0.363%	0.885%	3.830%
100_05	0.501%	0.529%	0.454%	0.320%	0.533%	0.480%	0.580%	0.729%	3.933%
100_08	0.098%	0.474%	0.263%	0.097%	0.346%	0.397%	0.610%	0.777%	6.083%
150_02	0.210%	0.497%	0.336%	0.199%	0.297%	0.633%	0.456%	0.757%	4.770%
150_05	0.749%	1.121%	0.769%	0.485%	1.181%	1.004%	1.277%	1.208%	8.185%
150_08	0.823%	1.092%	0.860%	0.598%	1.057%	1.367%	1.212%	1.080%	2.404%
300_02	2.188%	2.137%	2.241%	1.665%	2.317%	3.549%	2.441%	2.739%	5.951%
300_05	1.784%	1.888%	2.224%	1.767%	2.088%	2.492%	2.705%	2.276%	9.678%
300_08	2.088%	2.256%	1.966%	1.846%	3.142%	2.333%	2.750%	2.108%	7.875%
500_02	0.965%	0.993%	1.254%	1.034%	1.386%	4.205%	4.823%	5.495%	7.121%
500_05	3.098%	3.323%	2.850%	2.172%	5.228%	2.547%	3.038%	2.505%	8.596%
500_08	2.610%	2.293%	2.412%	2.256%	3.830%	2.071%	2.132%	2.103%	5.272%
平均誤差	1.274%	1.403%	1.346%	1.045%	1.800%	1.772%	1.866%	1.889%	6.141%

表 4　各演算法平均解之求解結果 (續)

網路類型	AACS-2	AACS-3	GA-1	GA-2	GA-3	TA	TTA	GDA	TGDA
100_02	2.857%	6.488%	3.165%	6.488%	3.895%	3.080%	5.310%	3.506%	5.765%
100_05	6.041%	0.908%	3.484%	3.841%	4.338%	5.247%	4.956%	4.817%	6.530%
100_08	2.157%	1.212%	4.955%	2.138%	1.797%	6.475%	9.604%	4.283%	5.686%
150_02	4.515%	2.578%	3.987%	1.982%	4.332%	7.502%	7.770%	8.142%	8.422%
150_05	5.086%	5.706%	4.476%	6.907%	2.393%	10.146%	15.133%	13.428%	11.915%
150_08	3.854%	2.834%	2.897%	2.695%	3.013%	12.295%	6.807%	9.165%	13.005%
300_02	8.787%	10.249%	4.571%	5.813%	3.959%	9.629%	14.192%	8.353%	16.850%
300_05	10.392%	10.447%	4.933%	4.167%	5.218%	13.728%	16.581%	16.042%	12.500%
300_08	9.338%	9.846%	6.514%	6.981%	7.900%	16.534%	12.145%	12.116%	19.554%
500_02	8.253%	7.601%	2.566%	5.038%	2.342%	15.550%	15.387%	7.979%	12.863%
500_05	9.581%	6.343%	6.221%	5.277%	6.007%	15.963%	12.346%	10.267%	11.243%
500_08	7.575%	8.289%	6.971%	5.663%	5.192%	14.034%	16.486%	8.284%	13.850%
平均誤差	6.536%	6.042%	4.562%	4.749%	4.199%	10.848%	11.393%	8.865%	11.515%

表 5　AHS 於各種網路類型下之求解時間

網路類型	AHS 組合方案							
	AHS-12		AHS-39		AHS-43		AHS-45	
	誤差百分比	求解時間（秒）	誤差百分比	求解時間（秒）	誤差百分比	求解時間（秒）	誤差百分比	求解時間（秒）
100_2	0.000%	77.77	0.000%	70.66	0.000%	97.00	0.000%	142.19
100_5	0.000%	119.47	0.000%	86.45	0.000%	124.25	0.000%	182.00
100_8	0.000%	115.52	0.000%	102.84	0.000%	142.80	0.000%	209.36
150_2	0.000%	106.53	0.000%	94.42	0.000%	133.16	0.000%	193.81
150_5	0.000%	140.64	0.000%	121.56	0.000%	167.33	0.000%	248.78
150_8	-0.005%	175.88	0.096%	164.49	-0.005%	197.24	0.218%	286.22
300_2	-0.380%	221.33	0.556%	196.89	0.681%	266.83	-0.429%	397.39
300_5	0.237%	392.30	-0.712%	462.27	-0.317%	364.27	0.234%	618.98
300_8	0.912%	406.16	-0.002%	315.68	-0.003%	742.28	0.241%	645.23
500_2	-1.126%	423.44	-1.212%	433.69	-1.407%	449.84	-0.779%	872.24
500_5	-0.434%	1043.38	0.548%	883.69	0.634%	652.38	-0.120%	1590.14
500_8	-0.584%	1024.47	-0.298%	823.69	-0.214%	1289.89	-0.865%	3095.36
平均	-0.115%	353.91	-0.085%	313.03	-0.053%	385.60	-0.125%	706.81

在各方案平均解部份如表 4 所示，其中 AHS-45 方案平均誤差表現最佳為 1.045%，而 HS 平均誤差為 1.800%；APSO 3 種方案平均誤差為 1.772% 至 1.889%之間；AACS 3 種方案平均誤差為 6.042%至 6.536%之間；GA 之 3 種方案平均誤差為 4.199%至 4.749%之間；4 種區域搜尋平均誤差則達 8.865%至 11.515%之間。此結果顯示在相同求解時間之基礎下，AHS 不論在最佳解或平均解皆能優於其他演算法。

經由上述結果分析，可發現本研究演算法所採用測試的 4 種參數組合 (AHS-12, AHS-39, AHS-43 與 AHS-45)，均可在不同的網路類型下達到文獻目前最佳解或是優於文獻目前最佳解，顯示本演算法的演算架構及所設計的 4 種參數組合對於本問題的求解，具有良好的尋優效果。另雖然每個參數組合無法在所有的網路類型下達到文獻目前的最佳解(例如:AHS-12 在 300_05,300_08 網路類型的求解)，但觀察其最佳解的平均誤差與平均解結果時，均可發現仍優於其他啟發式解法的平均誤差與平均解，故可說明本演算法在求解此類問題時，具有較優的求解品質，可兼顧求解品質的最佳性與可靠性。

3.3 收斂分析

為評估 AHS 於大型網路求解收斂之情況，本研究以參數方案組合 AHS-2 為例說明，至於其他參數組合方案的結果類似，不再贅述。在測試方案中每一回合皆輸出兩種數值，第一為回合中的最佳目標值 f_g，第二為回合中的平均目標值 $\bar{f}$ ，利用此數值與本研究所找

到最佳解 F_g 進行探討，以討論在大型網路但不同網路密度下的收斂情況，故本研究以 AHS 在不同網路與供需節點密度下，將此 3 種數值與運算回合的關係，繪成大網路高密度圖 (500_08，如圖 3 所示)、大網路中密度圖 (500_05，如圖 4 所示) 與大網路低密度圖 (500_02，如圖 5 所示) 作為比較。

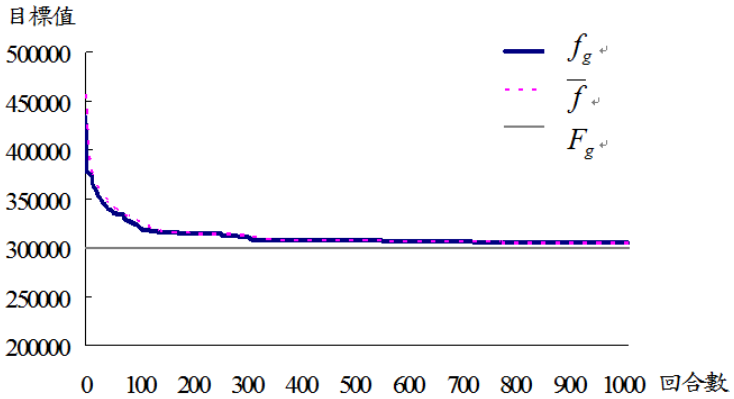

圖 3　大網路高密度 (500_08) 之求解收斂圖

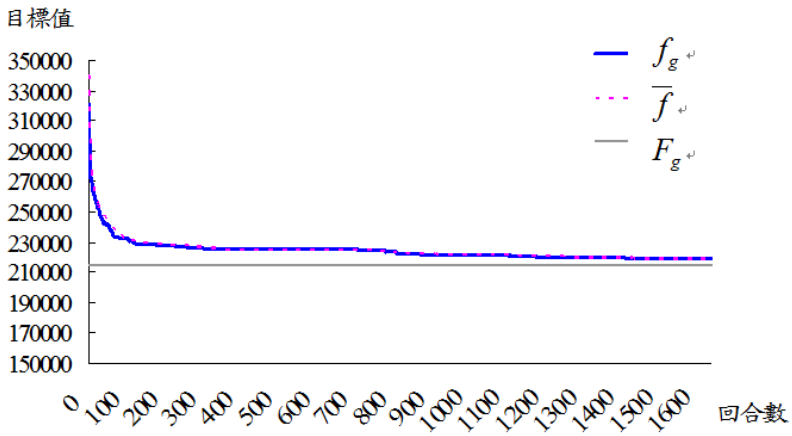

圖 4　大網路中密度 (500_05) 之求解收斂圖

由上列各圖可知，當 AHS 求解不同網路密度之問題時，前幾回合皆能快速減少最小目標值與平均目標值，如圖 3 之大網路高密度問題，第 10 回合最小目標值即由 434,860 降至 375,238，和弦記憶空間中平均目標值亦由 458,164 降至 384,040。而後至 150 回合，最小目標值與平均目標值下降速度漸趨緩和。150 回之後則以相當緩慢的速度下降，同時平均目標值亦越來越接近最小目標值，顯示此搜尋已接近區域最佳解。300 回合後最小目標值突然以較大幅度下降，顯示演算法已成功跳出區域範圍，找到更佳的可行解。之後平均

目標值亦因和弦記憶空間不斷取代與更新，緩慢往最小目標值靠近。此後目標值下降幅度皆有限，但因皆能在 100 回合內進行最小目標值之更新，因此演算回合一直延續到 1,011 回為止。另圖 4 大網路中密度問題與圖 3 大網路高密度問題相似，在演算法前 100 回合最小目標值與平均目標值快速下降，之後則以相當緩和的幅度減少，但因連續的 100 回合內最小目標值皆能成功地降低，因此演算回合延續至 1,637 回。而圖 5 大網路低密度問題之最小目標值，則在前 50 回合快速下降，平均目標值亦因最小目標值的不斷更新而降低，50 回合後最小目標值之下降減緩，至 250 回合已降至接近本研究所得之最佳解，同時平均目標值亦漸漸與最小目標值重合，至 357 回合，平均目標值已等於最小目標值，顯示演算法已搜尋至區域最佳解。大網路低密度之問題因能快速往本研究最佳解收斂，因此演算法在基本 500 回合內結束，求解時間亦較短。

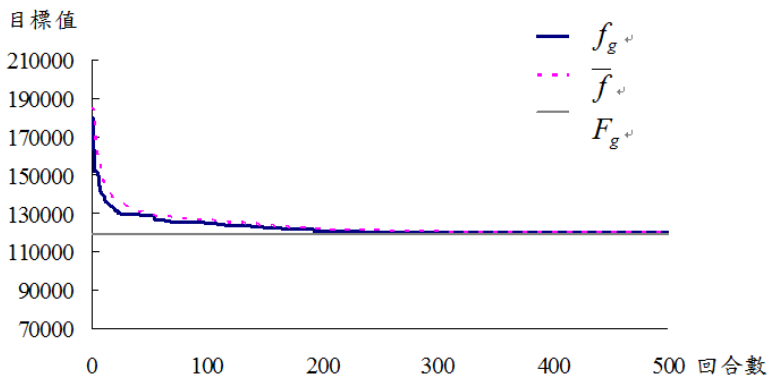

圖 5　大網路低密度 (500_02) 之求解收斂圖

上述 3 種密度之網路問題求解時均有最小目標值突然下降之情況，顯示本研究演算法能有效地跳離區域最佳解，進而搜尋到更好的解。此外，當利用 AHS-2 求解大網路高密度問題時，雖能往本研究求得之最佳解收斂，但仍與本研究所找到最佳解有 1.69%之求解間距，此求解間距隨著網路規模與網路密度之減小而降低，顯示網路規模越小或網路密度較小時，AHS 能有較好的求解績效。平均目標值方面，因本研究以和弦記憶空間之各組平均解為平均目標值，和弦記憶空間之解又隨著演算法進行不斷地更新，將較好的解加入，取代較差解，因此各組平均解會逐漸往最小目標值收斂。若最小目標值無法再下降時，數回合後平均目標值將趨近於最小目標值。

四、結論與建議

本研究以和諧搜尋演算法為基礎，再輔以 PSO、ACS、TA 與網路啟發式解法的相關機制，發展類和諧搜尋演算法，以求解實務上常見的含凹形節線成本最小成本轉運問題。

本研究並參考 Yan 等人 [25] 所提出求解此問題之 GA、TA、GDA 等演算法、Yan 等人 [28] 之類粒子族群最佳化演算法 (APSO) 及 Yan 等人 [27] 之類螞蟻族群演算法 (AACS) 作為求解績效評估之對照組，其測試結果顯示，本研究所提出的類和諧 (AHS) 最佳化演算法，於求解各規模的網路類型時，不僅在最佳解的平均誤差有較佳之表現，亦於比較平均解的平均誤差上有較佳的求解績效，此結果可供後續相關研究選取求解演算法時之參考。

此外，本研究假設目標函數之運送總成本為 $c_{ij}\sqrt{x_{ij}}$ 的凹形成本函數及節線流量無上限，於後續研究時可放鬆相關限制而進行更深入的探討。例如，若設定之成本函數形式與本研究不同時，可以相似之演算法流程進行求解；而具節線流量上限之網路可透過網路修正方式，轉換成不具節線流量上限之網路進行求解，或修訂伸展樹的設計及流量推擠法，以考量具節線流量上限的作法以進行求解。最後在運輸成本的處理上，由於目前業者均將相關費用以線性攤提的方式入會計帳處理，且因資料取得不易，故本研究是以產生測試範例方式 (共 12 組測試範例)，進行本問題的測試分析，但經本研究測試結果，除已證實演算法的穩健性 (robustness) 外，且求解效率亦相當佳，因此相關實務的成本函數建立及演算法的測試，建議可納入後續研究進行實務性的實證分析。

參考文獻

1. Yan, S. and Luo, S. C., “A Tabu Search-based Algorithm for Concave Cost Transportation Network Problems”, *Journal of the Chinese Institute of Engineers*, Vol. 21, 1998, pp. 327-335.
2. Yan, S. and Luo, S. C., “Probabilistic Local Search Algorithms for Concave Cost Transportation Network Problems”, *European Journal of Operational Research*, Vol. 117, 1999, pp. 511-521.
3. Garey, M. R. and Johnson, D. S., Computers and Intractability: A Guide to the Theory of NP-Completeness, WH Freeman and Co., San Francisco, 1979.
4. Zangwill, W. I., “Minimum Concave Cost Flows in Certain Networks”, *Management Science*, Vol. 14, 1968, pp. 429-450.
5. Rech, P. and Barton, L. G., “A Non-Convex Transportation Algorithm”, E. M. Beale, Ed., *Applications of Mathematical Programming Techniques*, American Elsevier Publishing, New York, 1970.
6. Gallo, G. and Sandi, C., “Adjacent Extreme Flows and Application to Min Concave Cost Flow Problems”, *Networks*, Vol. 9, 1979, pp. 95-121.
7. Gallo, G., Sandi C., and Sodini, C., “An Algorithm for the Min Concave Cost Flow Problem”, *European Journal of Operation Research*, Vol. 4, 1980, pp. 248-255.
8. Nourie, F. J. and Guder, F., “A Restricted-Entry Method for a Transportation Problem with Piecewise-Linear Concave Cost”, *Computer and Operations Research*, Vol. 21, 1994, pp. 723-733.
9. Osman, I. H. and Kelly, J. P., “Meta-Heuristics: An Overview”, *Meta-Heuristics: Theory*

and Applications, Kluwer Academic Publishers, Boston, London, Dordrecht, 1996, pp. 1-21.

10. Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P., "Optimization by Simulated Annealing", *Science*, Vol. 220, 1983, pp. 671-680.
11. Goldberg, D. E., Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, Reading, MA, 1989.
12. Glover, F., "Tabu Search, Part I", *ORSA Journal on Computing*, Vol. 1, No. 3, 1989, pp. 190-206.
13. Glover, F., "Tabu Search- Part II", *ORSA Journal on Computing*, Vol. 2, No. 1, 1990, pp. 4-32.
14. Dueck, G. and Scheuer, T., "Threshold Accepting: A General Purpose Optimization Algorithm Appearing Superior to Simulated Annealing", *Journal of Computational Physics*, Vol. 90, 1990, pp. 161-175.
15. Dueck, G., "New Optimization Heuristics: The Great Deluge Algorithm and the Record-to-Record Travel", *Journal of Computational Physics*, Vol. 104, 1993, pp. 86-92.
16. Reeves, C. R., "Improving the Efficiency of Tabu Search for Machine Sequencing Problems", *Journal of the Operation Research Society*, Vol. 44, No. 4, 1994, pp. 375-382.
17. Charon, I. and Hurdy, O., "The Noising Method: A New Method for Combinatorial Optimization", Operations Research Letters, Vol. 14, 1993, pp. 133-137.
18. 韓復華、卓裕仁，「門檻接受法、成本擾動法與搜尋空間平滑法在車輛路線問題之應用研究與比較分析」，**運輸學刊**，第 9 卷，第 3 期，民國 85 年，頁 103-129。
19. 韓復華、陳國清、卓裕仁，「成本擾動法在 TSP 問題之應用」，中華民國第 2 屆運輸網路研討會論文集，國立中央大學土木工程學系，民國 86 年，頁 283-292。
20. Gu, J. and Huang, X., "Efficient Local Search with Search Space Smoothing: A Case Study of the Traveling Salesman Problem (TSP)", *IEEE Transaction on Systems, Man and Cybernetics*, Vol. 24, 1994, pp. 728-739.
21. Kennedy, J. and Spears, W., "Matching Algorithms to Problems: An Experimental Test of the Particle Swarm and Some Genetic Algorithms on the Multimodal Problem Generator", *IEEE World Congress on Computational Intelligence, IEEE*, 1998, pp. 74-77.
22. Dorigo, M. and Gambardella, L. M., "A Study of Some Properties of Ant-Q", Proceedings of PPSN IV-Fourth International Conference on Parallel Problem Solving from Nature, Springer-Verlag, Berlin, 1996, pp. 656-665.
23. Hansen, P. and Mladenovic, N., "Variable Neighborhood Search", *Computers and Operations Research*, Vol. 24, 2007, pp. 1097-1100.
24. Geem, Z. W., Kim, J. H., and Loganathan, G. V., "A New Heuristic Optimization Algorithm: Harmony Search", *Simulation*, Vol. 76, No. 2, 2001, pp. 60-68.
25. Yan, S., Juang, D. H., Chen, C. R., and Lai, W. S., "Global and Local Search Algorithms for Concave Cost Transshipment Problems", *Journal of Global Optimization*, Vol. 33, No. 1, 2004, pp. 123-156.
26. 顏上堯、李旺蒼、施佑林，「路徑基礎類粒子群最佳化演算法於求解含凹形節線成本最小成本轉運問題之研究」，**運輸計劃季刊**，第 36 卷，第 3 期，民國 96 年，頁 393-423。
27. Yan, S., Shih, Y. L., and Wang, C. L., "An Ant Colony System-Based Hybrid Algorithm for

Square Root Concave Cost Transshipment Problems", *Engineering Optimization*, Vol. 42, Issue 11, 2010, pp. 983-1001.

28. Yan, S., Shih, Y. L., and Lee, W. T., "A Particle Swam Optimization-Based Hybrid Algorithm for Minimum Concave Cost Network Flow Problems", *Journal of Global Optimization*, Vol. 49, No. 4, 2011, pp. 539-559.
29. Geem, Z. W., Lee, K. S., Lee, S. H., and Bae, K. W., "The Harmony Search Heuristic Algorithm for Discrete Structural Optimization", *Engineering Optimization*, Vol. 37, No. 7, 2005, pp. 663-684.
30. Geem, Z. W., Tseng, C. L., and Park, Y., "Harmony Search for Generalized Orienteering Problem: Best Touring in China", *Advances in Natural Computation*, Vol. 3612, 2005, pp. 741-750.
31. 李亮、遲世春、褚雪松，「基於修復策略的改進和聲搜索演算法求解土坡非圓臨界滑動面」，**中國岩土力學**，第 27 卷，第 10 期，民國 95 年，頁 1714-1718。
32. 李亮、遲世春、鄭榕明、林皋，「一種新型遺傳演算法及其在土坡任意滑動面確定中的應用」，水利學報，第 38 卷，第 2 期，民國 96 年，頁 157-162。
33. Wang, L., Pan, Q. K., and Tasgetiren, M. F., "Minimizing the Total Flow Time in a Flow Shop with Blocking by Using Hybrid Harmony Search Algorithms", *Expert Systems with Applications*, Vol. 37, No. 12, 2010, pp. 7929-7936.
34. Afkhami, S., Ma'rouzi, O. R., and Soleimani, A., "A Binary Harmony Search Algorithm for Solving the Maximum Clique Problem", *International Journal of Computer Applications*, Vol. 69, No. 12, 2013, pp. 38-43.
35. Hosseini, S. D., Shirazi, M. A., and Taghi Fatemi Ghomi, S. M., "Harmony Search Optimization Algorithm for a Novel Transportation Problem in a Consolidation Network", *Engineering Optimization*, Vol. 46, 2014, pp. 1538-1552.
36. Ahuja, R. K., Maganti, T. L., and Orlin, J. B., *Network Flows, Theory, Algorithms, and Applications*, Prentice Hall, Englewood Cliffs, 1993.
37. Yan, S. and Young, H. F., "A Decision Support Framework for Multi-Fleet Routing and Multi-Stop Flight Scheduling", *Transportation Research*, Vol. 30A, 1996, pp. 379-398.