

行政院國家科學委員會專題研究計畫 成果報告

應用螞蟻演算法求解時間窗收卸貨問題之研究：整合時間窗分割技術 研究成果報告(精簡版)

計畫類別：個別型
計畫編號：NSC 96-2221-E-216-018-
執行期間：96年08月01日至97年07月31日
執行單位：中華大學運輸科技與物流管理學系

計畫主持人：張靖

計畫參與人員：博士班研究生-兼任助理人員：歐世明
博士班研究生-兼任助理人員：莊子駿
博士班研究生-兼任助理人員：林靜芬

處理方式：本計畫可公開查詢

中華民國 97年09月26日

行政院國家科學委員會補助專題研究計畫 成果報告
 期中進度報告

應用螞蟻演算法求解時間窗收卸貨問題之研究：整合時間窗
分割技術(1/2)

Integrate Ant Colony System and Time Window Partitioning
Strategy to Solve the Pick and Delivery Problem With Time
Window (1/2)

計畫類別： 個別型計畫 整合型計畫

計畫編號：NSC 96 - 2221 - E - 216 - 018

執行期間：96年8月1日至97年7月31日

計畫主持人：張靖

計畫參與人員：林靜芬、莊子駿、歐世明

成果報告類型(依經費核定清單規定繳交)： 精簡報告 完整報告

本成果報告包括以下應繳交之附件：

- 赴國外出差或研習心得報告一份
- 赴大陸地區出差或研習心得報告一份
- 出席國際學術會議心得報告及發表之論文各一份
- 國際合作研究計畫國外研究報告書一份

處理方式：除產學合作研究計畫、提升產業技術及人才培育研究計畫、
列管計畫及下列情形者外，得立即公開查詢

涉及專利或其他智慧財產權， 一年 二年後可公開查詢

執行單位：中華大學運輸科技與物流管理學系

中華民國九十七年九月二十四日

摘要

本研究利用時窗分割 (Partitioning)與整數化(Discretization) 策略將時窗收卸貨問題 (Pickup and Delivery Problem with Time Windows , PDPTW) 轉換為無時窗的近似 PDP(Similar Pickup and Delivery Problem, SPDP), 即得以求解 PDPTW 時不用考慮時窗。在轉換過程中進一步提出問題規模精簡策略與關連式旅行網路結構表, 大量減少轉換過程中所衍生出來的新的限制式與決策變數的數量。研究中利用 Lau and Liang(2002)的方法, 修改國際標準題庫 VRPTW 例題, 產生具有相同最適解 (The Optimal Solution) 之 PDPTW 例題進行測試。本研究首先提出 SPDP 之數學規劃模式, 利用小規模例題, 透過 LINGO 來驗證, 當時窗切割夠細的話, 轉換所得之 SPDP 與原 PDPTW 之最適解是相同的, 即求解時窗切割夠細的 SPDP 即可求得原 PDPTW 之最適解; 其次利用較大規模問題, 透過啟發式演算法來驗證, 顯示求解 SPDP 較直接求解原 PDPTW, 其解的精確度平均提升 7.88%, 求解時間大幅減少達 88.10%, 因此本研究確定先將 PDPTW 轉換為 SPDP 求解較直接求解 PDPTW, 確實可以在較短的時間內求出品質較佳的近似最佳解。

關鍵詞: 時窗收卸貨問題、收卸貨問題、時窗分割策略、巨集啟發式演算法

Abstract

The main goal of this research is to provide a new concept to solve a Pickup and Delivery Problem with Time Window (PDPTW) efficiently and accurately. In order to achieve this goal, a PDPTW is transferred to be a new similar PDP (SPDP) without time window by the Time Window Partitioning and Discretization Strategy. Every time window of each pickup or delivery point is partitioned as many equal-length sub time window. Besides, only one of all sub time windows of the pickup or delivery point can be served. The SPDP is formulated as a 0-1 integer programming

model in this paper. The optimal solution obtained by LINGO of the transferred SPDP is equal to the optimal solution of the original PDPTW when the time window is partitioned small enough, i.e. the SPDP is the same as PDPTW when the length of the sub time window is short enough. However, the size of the transferred SPDP is much bigger than the original PDPTW because a lot of new decision variables and constraints are produced. Since these additional derived decision variables and constraints will make computation inefficient, we also design a preprocessing procedure to reduce problem size of the SPDP, e.g. the redundant decision variables and constraints, and a relation structured asymmetric travel cost matrix to avoid searching the infeasible solutions. There are 18 Solomon benchmark VRP problems transferred to be PDPTW problems by the method developed by the Lau and Liang (2002). In order to show our contribution, we developed a simple Meta-Heuristic algorithm to solve both PDPTW and SPDP. According to the computation results, we can improve the accuracy about 7.88% and save the computation time about 88.1%.

Keywords: Pickup and Delivery Problem with Time Windows, Time Window Partitioning Strategy, Meta-Heuristic algorithm

一、前言

時窗收卸貨問題(Pickup and Delivery with Time Windows , PDPTW) 已知屬於 NP-hard 問題(Fabián and Pérez, 2005), 運算時間會隨著例題規模增加而呈指數性的成長, 無法在合理的時間內求得精確解, 因此近年許多屬於 NP-hard 問題皆已朝向巨集啟發式演算法 (Meta-Heuristics)的方向發展, 在滿足各需求服務的時窗內, 以總旅行成本、或配合總車輛數最少作為路徑構建時的主要目標函數, 因此如何提出一個有效的 PDPTW 啟發式演算法是值得重視的。

本研究提出不同的求解概念, 建議利用時窗切割與整數化策略將 PDPTW 時窗限制

除去並轉換為接近無時窗的 PDP，不僅可以延用原有 PDP 相關的啟發式演算法，同時可避免所有與服務時間相關的運算過程，搜尋效率可以提高，但是轉換所得的近似 PDP(Similar PDP, SPDP) 將不同於傳統 PDP，而且會衍生許多新的限制式與決策變數，本研究進一步提出問題規模精簡策略與關連式旅行網路結構表來克服這些新生的障礙。

二、文獻回顧

從過去 PDPTW 相關研究的解法上可以發現，可以求出精確解(Exact Solution)的辭書搜尋法、分枝定限法、分枝截面法等求解效率皆不佳，且僅止於求解較小型的問題(Dumas *et al.*, 1991; Kohl, 1995; Lu and Dessouly, 2004; Lau and Liang, 2002)，因此多數學者皆已朝向巨集啟發式演算法(Meta-Heuristics)的方向發展，原因在於巨集啟發式演算法能夠跳脫求解過程中陷入局部最佳解的困境，與改善傳統啟發式演算法的效率。

2.1 PDPTW 精確解演算法回顧

Psarafis (1980, 1983)針對單一車種的 PDPTW，提出一個複雜度為 $O(n^23^n)$ 的動態可程式演算法，求取精確解，目標函數為求取最小的顧客服務不便(例如服務延遲)，但僅能求解很小規模的問題，最大需求量必須小於 10。Sexton and Bodin (1985^a, 1985^b)也針對相同的問題，提出組合式演算法求取精確解，該方法將單一車種 PDPTW 中，有關於路線規劃方面的問題，轉換為整數規劃問題；而指派作業方面的問題則轉換為線性問題，再利用演算法依序個別求解，其可以在 18 秒內有效求解出需求規模介於 7~20 之間的 PDPTW。目前所發展可應用於求解 PDPTW 精確解之演算法仍無法有效降低求解時間，且在合理時間內可求得精確解之問

題規模亦不超過 30 個顧客。

2.2 PDPTW 啟發式演算法回顧

Nanry and Barnes (2000) 提出反應式禁忌演算法(Reactive Tabu Search)，以最小旅行成本為車輛指派與路徑規劃目標，配合違反時窗與車輛超載兩種懲罰成本函數機制，考量與違反時窗限制條件，求解 PDPTW，問題規模為 100 個節點的問題，與 Kohl (1995) 所求得的近似最佳解相比較，平均差距僅 0.01%，且求解時間平均減少 65%。Lau and Liang (2002) 發展兩階段啟發式演算法求解 PDPTW，第一階段利用類似掃描法的方式，針對收貨點進行路線分群，然後再將卸貨點加入對應群中最適合的路線位置，第二階段亦配合利用禁忌搜尋法進行路線改善作業，測試需求為 100 對左右規模的問題，可求得不錯的近似解。Bent and Van Hentenryck (2006) 亦提出兩階段式演算法，研究中提及降低使用的車輛數是降低總旅行成本的關鍵之一，因此第一階段主要目標為減少車輛總數，第二階段為減少總路線成本，同時也再次減少車輛數。演算法核心則改為模擬退火法(Simulated Annealing, SA)與大型鄰域搜尋法(Large Neighborhood Search, LNS)。該演算法改善了國際題庫中 200 與 600 個顧客標竿解達 47% 與 76%，為求解績效極佳的啟發式演算法之一，但求解問題規模較大時相對求解時間亦相對增加。

以上學者們所提出的各式啟發式演算法，皆可求得高品質的近似最佳解，但是反觀學者們設計各式演算法的目的與求解過程，多是針對如何跳出區域最佳解、如何增加搜尋廣度與搜尋深度為目標，但是針對時窗限制所提出的啟發式演算法，卻相對較少。

2.3 考量時窗限制發展之演算法回顧

實際生活中所會面臨到的各式路徑規劃問題，極大多數都會存在服務時間限制，但是相較於過去學者們提出求解考慮時間窗限

制之各式路徑規劃問題的啟發式演算法，以時間窗限制或服務時間為基礎所發展的啟發式演算法則相對較少。Wang(2000)將過去學者們以時間窗限制為基礎所發展的啟發式演算法，依服務時間處理方式區分為三種：

1. 考量時間限制

該類型處理方式為在可行路徑構建過程中，同時且優先考慮時間限制，相關啟發式演算法有 Dantzig-Wolf 分解法 (Dror, 1994)，該分解法近似於 Lagrangian 鬆弛法，是一種可以求得近似最佳解的方式，藉由鬆弛較複雜的限制式，將原問題簡化，另外尚有 Kolen, *et al.* (1987) 針對動態問題所提出的特定區域鬆弛法(State Space Relaxation)。

2. 時間限制式鬆弛法

該類型方法首先鬆弛問題中所有包含時間的限制式，然後進行求解，並藉由每次求解回合中最後產生的不可行解，作為下一求解回合中時間分割的依據，然後再利用網絡鬆弛法(Network Relaxation Methods)求解時間鬆弛後的問題，如此反覆求解。該種啟發式演算法的求解績效較 Dantzig-Wolf 分解法差，但是卻優於 Lagrangian 鬆弛法 (Desrosiers *et al.*, 1995)。

3. 時間分割法 (Appelgren (1969, 1971); Levin(1971) ; Swersey and Ballard(1984); Graharn and Nuttle (1986))

時間分割法將具有連續時間的作業點，分割為一群具有不同子時間的數個作業點，並各自視為一個獨立的決策變數。Graharn and Nuttle (1986) 針對時間分割法的績效進行測試，發現該方式所求得的近似最佳解品質極佳，但卻造成求解時間過長。Wang (2000) 應用時間分割法求解 mTSP 問題，將各作業時間分割成數個子時間，並將各個子時間視

為一項新的作業，再配合利用由大到小的分割時間長度，產生差距愈來愈小的最佳解上、下限值，漸漸逼近至找出最佳解的方式。

綜上所述，時間分割法的觀念尚未使用於 PDPTW，因此，本研究提出時間分割與整數化策略，將 PDPTW 轉換成 SPDP，再利用 PDPTW 車輛行駛路線卸貨作業地點必須在收貨作業地點之後的特性，設計問題規模精簡策略與關連式旅行網路結構表來解決轉換 SPDP 後問題規模倍增的缺點。

三、PDPTW 轉換 SPDP

本研究將 PDPTW 轉換為 SPDP，期望求解 SPDP 時可以完全忽略各需求作業的服務時間限制。轉換作業主要包括 (1) PDPTW 中作業點分割與整數化；(2) SPDP 中各子作業點服務時間計算；(3)建立 SPDP 子作業間的旅行成本矩陣；以及(4)精簡 SPDP 問題規模(意即精簡 SPDP 子作業旅行成本矩陣)等作業步驟。

1. PDPTW 作業點分割與整數化

鑑於實務問題中各作業點的服務時間窗大小不盡相同，並且作業點之間的原始時間窗區間差異可能過大，若使用固定數量進行分割作業，將使得部分子作業時間窗極小，徒增問題規模，以及部分子作業的時間窗偏大，徒增車輛閒置時間，增加搜尋最佳解的困難，因此研究中採用固定的時間長度的策略來進行時間窗分割，如圖 1 所示。

作業點 J_u 分割出了 f_u 個時段，計算方式如式(18)所示，需針對 PDPTW 中的每一個作業點，逐個進行分割，因此分割所需時間將隨著 PDPTW 問題規模的大小呈線性趨勢關係。

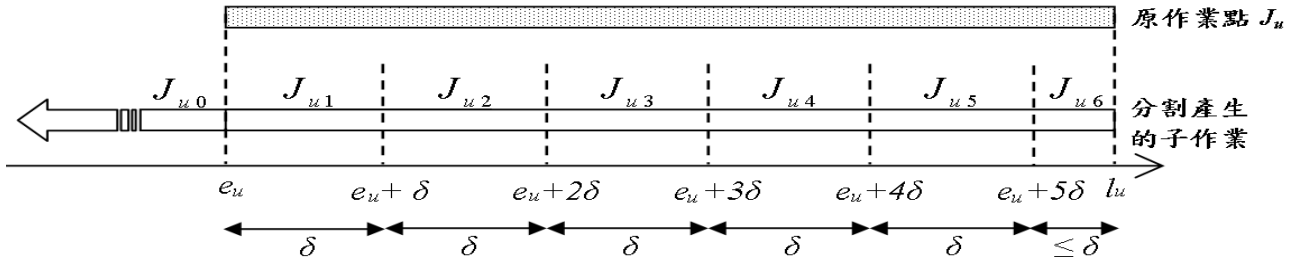


圖 1 固定時間值分割時窗示意圖

$$f_u = \left\lceil \frac{(l_u - e_u)}{\delta} \right\rceil, \quad \forall u \in V \quad (18)$$

作業點 J_u 所分割出的各個時段，分別視為一項新的子作業，子作業 J_{ui} 即表示作業點 J_u 所分割出的第 i 個子作業；步驟 5 中額外增加虛擬子作業的目的，為允許車輛在各作業點的最早開始服務時間前到達。若是由分割作業點 J_u 後所產生的，則以 J_{u0} 表示，服務時窗固定為 $[S, e_u]$ ，最早開始服務時間固定為車輛離開場站的最早時間(開工時間)，最晚完成時間則等於子作業 J_{u1} 的最早開始服務時間。

2. SPDP 子作業點服務時窗計算

作業點被服務的時間，主要是由車輛所服務的前一個作業點的離開時間與該作業點本身的服務時窗限制推算而得，子作業的服務時間計算方式當然也相同，計算方式如式(19)與(20)，分別為子作業 J_{ui} 的最早開始服務時間(e_{ui})與最晚完成時間(l_{ui})。

$$e_{ui} = \text{Max}(e_u, e_u + \delta(i-1)), \quad i=1,2,\dots,f_u, \quad \forall u \in V \quad (19)$$

$$l_{ui} = \text{Min}(e_u + (\delta \cdot i), l_u), \quad i=1,2,\dots,f_u, \quad \forall u \in V \quad (20)$$

3. 建立 SPDP 子作業間旅行成本矩陣

建立旅行成本矩陣前，需先利用整數化策略產生 SPDP 中無時窗限制的作業點決策變數與路段決策變數。然後藉由子作業旅行成本矩陣，將子作業時窗限制刪除，故構建

子作業旅行成本矩陣時需採用嚴謹離開時間法則，方能確保任兩個子作業間連續服務時，必定不會違反各自的時窗限制，意即車輛離開時間點一律使用該子作業的最晚完成時間。

(1) 增設虛擬的子作業 J_{u0}

由於離開子作業的時間點一律使用最晚完成時間，為確保作業點 J_u 最早能在 e_u 時開始服務，因此需新增一個虛擬的子作業 J_{u0} ，否則可能出現車輛離開場站後，沒有可以服務的子作業，以及可能因為車輛等待時間減少而增加了車輛使用數量的情況發生。

(2) 建立子作業旅行成本矩陣

子作業旅行成本矩陣為針對兩兩子作業間、以及子作業與場站間，所構建的旅行成本矩陣，是一個非對稱式的矩陣。矩陣中存在旅行成本的路段表示依序服務路段所連結的兩個子作業時，絕對不會違背該兩個子作業的服務時窗限制，利用子作業旅行成本矩陣求解時，將可以完全不需要考慮跟時窗有關的運算，即將 PDPTW 轉換為 SPDP。矩陣中所存在的所有路段皆必須滿足兩項最基本的原則：

原則 1：各作業地點必須僅服務一次

原則 2：滿足子作業服務時窗限制

車輛到達時間必須在子作業服務時窗範圍內，服務完成後的

離開時間亦不可超過子作業的最晚離開時間，如式(21)與(22)所示。無法滿足此兩項限制條件的子作業連結路段，皆可從矩陣中刪除。

$$l_{ui} + t_{ui,vj} \geq e_{vj} \quad (21)$$

$$l_{ui} + t_{ui,vj} + s_{vj} \leq l_{vj} \quad (22)$$

SPDP 中雖減少了後續運算過程中所有與服務時間有關的步驟，但是子作業的總數量卻倍增於原作業點的數量，減少的時窗限制運算時間，無法確保其可以彌補問題規模增加所造成的限制式增量與運算時間，因此進一步設計問題規模精簡策略，將可行解中不可能出現的路段，以及前述轉換作業所產生的部份不可能被服務的子作業，從問題中刪除。

4. 精簡 SPDP 問題規模

子作業旅行成本矩陣中仍存在部分 PDPTW 可行解中不可能出現的路段，例如不滿足作業點收、卸貨服務順序的路段、不滿足車容量限制的路段、不滿足車輛起迄場站時須為空車等等，這些路段在求解模式中都會使問題規模增加的決策變數，也是造成求解時間增加的無效運算步驟，因此本研究進一步增設 6 條規模精簡原則(原則 3 至原則 8)。

原則 3：同一訂單的收貨作業點必須在卸貨作業點前被服務

原則 4：任兩連續服務作業所裝載的貨物量不得超過車容量的限制

原則 5：車輛離開與回到場站時必須為空車

當子作業旅行成本矩陣中違反原則 3、4 與 5 的路段刪除後，我們檢視子作業旅行成本矩陣中每個子作業間的連結關係，發現由於部分決策變數的刪除，會造成新的無效決

策變數產生，因此再設計原則 6、原則 7 與原則 8，進一步再反覆刪除這些因為部分決策變數被刪除而新生的無效決策變數。

原則 6：若某一收貨作業僅可以從場站到達，則該收貨作業能直接到達的卸貨作業，必為且僅有與其相對應的卸貨作業

原則 7：若某一卸貨作業僅能直接回到場站，可到達該卸貨作業的收貨作業，必為且僅有與其相對應的收貨作業(原則 7 與原則 6 相呼應，作業執行所需時間將與 SPDP 問題規模呈現平方趨勢關係。)

原則 8：子作業點應符合流量守恆定理子作業旅行成本矩陣中違反原則 3~7 的路段皆被刪除後，發現部分子作業亦隨之出現「路徑中斷」現象，這類型子作業文中後續稱之為無用子作業。當車輛行至無用子作業點後，就沒有任何其他子作業點可再進行服務，也無法回到場站；或是車輛無法從任何子作業或場站行駛至無用子作業。無用子作業不可能存在於可行解，當然可從矩陣中刪除，同時也可將所有與無效子作業相連的路段也刪除。由於又有部份路段被刪除掉，所以也可能有新的無用子作業再次出現，因此需反覆確認與執行刪除作業，直到矩陣中確定沒有無用子作業為止。

值得注意的是，求解 SPDP 時，仍須考慮各個不連續服務的子作業間的收、卸貨順序關係、累積車容量限制與一組作業必須由同一車輛服務等的限制條件。

四、模式設計與測試結果

本研究構建 SPDP 的整學規劃模式，模式中之作業與路段決策變數為應用經時窗分割策略與問題規模精簡策略處理過之有效決策變數。並撰寫 LINGO 程式進行求解測試，以印證求解 SPDP 等於求解 PDPTW。

4.1 數學規劃模式

本研究中不允許服務延遲，且車輛服務各作業點時皆採用嚴謹離開時間法則，因此目標式中僅需要考慮最小化的使用車輛數與路徑成本，如式(23)所示，只要 P（車輛固定成本方式）夠大就可以滿足最少車輛數為目標式主要目標，最短旅行成本為次要目標，SPDP 數學規劃模式中使用的符號定義於表 9。另外，作業中服務等待時間最小化之目標，採用時窗分割技術進行控制，並未加入目標式中，當分割時窗時使用的 δ 值愈小，作業中的服務閒置時間將愈少。

$$\text{Min} \quad \sum_{k=1}^m \sum_{u=0}^{2n} \sum_{i=0}^{f(u)} \sum_{v=0}^{2n} \sum_{j=0}^{f(v)} c_{ui,vj} X_{ui,vj}^k + P \cdot \sum_{k=1}^m \sum_{u=1}^{2n} \sum_{i=0}^{f(u)} X_{0,ui}^k \quad (23)$$

模式限制式方面，由於子作業旅行成本矩陣中存在旅行時間的路段，必定滿足原則 1 至原則 8，因此，後續求解時就僅需考量不連續作業間的服務順序關係以及車容量限制條件，在服務延遲時間方面則嚴格控制不允許服務延遲狀況出現。數學規劃模式限制式詳列於下，模式中使用的集合、決策變數與參數則定義於表 9 中。

$$\sum_{k=1}^m \sum_{i=0}^{f(u)} Z_{ui}^k = 1, \quad u=1,2,\dots,2n \quad (24)$$

$$\sum_{i=0}^{f(u)} Z_{ui}^k - \sum_{j=0}^{f(u+n)} Z_{u+n,j}^k = 0, \quad u=1,2,\dots,n, \quad k=1,2,\dots,m \quad (25)$$

$$\sum_{k=1}^m \sum_{i=0}^{f(u)} l_{ui} \times Z_{ui}^k - \sum_{k=1}^m \sum_{j=0}^{f(u+n)} l_{u+n,j} \times Z_{u+n,j}^k \leq 0, \quad u=1,2,\dots,n \quad (26)$$

$$\sum_{k=1}^m \sum_{i=0}^{f(u)} \sum_{v=0}^{2n} \sum_{j=0}^{f(v)} X_{ui,vj}^k = 1, \quad u=1,2,\dots,2n, \quad (27)$$

其中

$$\begin{cases} \text{若 } u=1,2,\dots,n, \text{ 則 } v \neq 0 \text{ 且 } v \neq u \\ \text{若 } u=n+1,n+2,\dots,2n, \text{ 則 } v \neq u-n \text{ 且 } v \neq u \end{cases}$$

$$\sum_{k=1}^m \sum_{u=0}^{2n} \sum_{i=0}^{f(u)} \sum_{j=0}^{f(v)} X_{ui,vj}^k = 1, \quad v=1,2,\dots,2n, \quad (28)$$

其中

$$\begin{cases} \text{若 } v=1,2,\dots,n, \text{ 則 } u \neq v+n \text{ 且 } u \neq v \\ \text{若 } v=n+1,n+2,\dots,2n, \text{ 則 } u \neq v \text{ 且 } u \neq 0 \end{cases}$$

$$\sum_{v=0}^{2n} \sum_{j=0}^{f(v)} X_{vj,ui}^k = Z_{ui}^k = \sum_{v'=0}^{2n} \sum_{j'=0}^{f(v')} X_{ui,v'j'}^k, \quad k=1,2,\dots,m, \quad u=1,2,\dots,2n, \quad (29)$$

其中

$$\begin{cases} \text{若 } u=1,2,\dots,n, \text{ 則 } v \neq v+n, v \neq u, \text{ 且 } v' \neq 0, v' \neq u \\ \text{若 } u=n+1,n+2,\dots,2n, \text{ 則 } v \neq 0, v \neq u, \text{ 且 } v' \neq u-n, v' \neq u \end{cases}$$

$$\sum_{u=1}^n \sum_{i=0}^{f(u)} X_{0,ui}^k = \sum_{v=n+1}^{2n} \sum_{j=0}^{f(v)} X_{vj,0}^k, \quad k=1,2,\dots,m \quad (30)$$

$$\sum_{u=1}^n \sum_{i=0}^{f(u)} X_{0,ui}^k \leq 1, \quad k=1,2,\dots,m \quad (31)$$

$$y_v = y_v \left(1 - \sum_{i=0}^{f(u)} \sum_{j=0}^{f(v)} X_{ui,vj}^k\right) + \sum_{i=0}^{f(u)} \sum_{j=0}^{f(v)} X_{ui,vj}^k (q_v + y_u), \quad k=1,2,\dots,m, \quad v=0,1,\dots,2n, \quad u=0,1,\dots,2n, \quad (32)$$

其中

$$\begin{cases} \text{若 } v=1,2,\dots,n, \text{ 則 } u \neq v+n, \text{ 且 } u \neq v \\ \text{若 } v=n+1,n+2,\dots,2n, \text{ 則 } u \neq 0, \text{ 且 } u \neq v \end{cases}$$

$$0 \leq y_u \leq Q, \quad u=1,2,\dots,2n \quad (33)$$

表 9 轉換所得之 PDP 問題數學規劃模式中符號定義

N	訂單集合, $n = N $, 表示訂單數量, 集合中的元素為 J_u
M	車輛集合, $m = M $, 表示可使用的車輛數
Q	車容量限制
P	使用車輛的固定成本, 相較於各車輛旅行距離而言, 是一個較旅行成本相對大的常數值。
u	變數下標, 表示作業點編號
v	變數下標, 表示作業點編號
J_u	表示編號為 u 的作業點, 當 $u=0$ 時為場站; $u = 1, \dots, n$ 時表示收貨作業點; $n = n+1, \dots, 2n$ 為卸貨作業點, 其中收貨作業點 J_u 所對應的卸貨作業點為 J_{u+n} 。該變數為集合變數, 集合中的元素為作業點所分割出的子作業。
q_u	作業點 u 的運載量, 當 $1 \leq u \leq n$ 時, $q_u \geq 0$, 反之 $q_u \leq 0$, 另外 $q_0=0$
$f(u)$	作業點 u 所分割出的子作業總數
J_{ui}	作業點 u 所分割出的第 i 個子作業, $i = 0, 1, \dots, f(u)$
$c_{ui,vj}$	子作業 J_{ui} 與子作業 J_{vj} 間的旅行成本
e_{ui}, l_{ui}	子作業 J_{ui} 的服務時窗, e_{ui} 最早開始服務時間, l_{ui} 為最晚完成服務時間 (或稱最晚離開時間)
Z_{ui}^k	若車輛 k 有服務 J_{ui} , 則 $Z_{ui}^k = 1$, 反之 $Z_{ui}^k = 0$, 另由於場站為車輛的起迄點, 因此 $Z_0^k = 1$
$X_{ui,vj}^k$	若車輛 k 從 J_{ui} 行駛至 J_{vj} (行經該路段), 則 $X_{ui,vj}^k = 1$, 反之 $X_{ui,vj}^k = 0$
y_u	車輛離開作業點 u 時的負載量, $y_0 = 0$

限制式(24)控制每各作業點僅能被一輛車服務一次, 式(25)控制每一個訂單的收送

貨作業必須由同一輛車完成服務, 式(26)控制收卸貨作業的服務順序, 式(27)(28)限制每一個作業點只能有一輛車駛離與到達, 式(29)為流量守恆現制式, 式(30)控制場站為車輛的起迄點, 式(31)則限制車輛僅能離開與回到場站至多一次, 式(32)與(33)則為車容量限制。

4.2 模式分析與實例驗證

國際標準題庫中具有精確解完整資料的 PDPTW 搜尋困難, 因此本研究應用 Lau and Liang (2002) 所提出的標準作業步驟, 配合國際標準題庫 VRPLIB 中之 VRPTW 問題(題目: r103), 轉換產生作業點數量為 10 與 20 之 PDPTW 各 5 題, 共 10 題具有精確解的 PDPTW 測試例題, 測試結果如表 12 所示。

SPDP 所求得之精確解路徑成本與路徑, 與原 PDPTW 的精確解路徑成本與路徑完全相同, 且經轉換所得之 SPDP 模式, 皆可在合理時間內求得原 PDPTW 的精確解, 以及模式中決策變數與限制式總數量皆明顯少於 PDPTW, 平均減少 47.01%。

表 12 精確解求解績效比較表

問題名稱	最佳解路徑成本	最佳解路徑	PDPTW		SPDP			
			決策變數數量 (A)	限制式數量 (B)	有效決策變數 (C)	限制式數量 (D)	求解時間	決策變數減少率% (B-D)/B%
r103_10_1	179	0-1-6-2-3-8-7-0 0-5-4-10-9-0	246	2036	505	1144	00:22	43.81%
r103_10_2	132.79	0-5-10-0 0-3-1-4-8-9-2-6-7-0	246	2036	614	1184	00:23	41.85%
r103_10_3	198.3	0-5-10-0 0-3-1-6-4-2-9-8-7-0	246	2036	656	1254	00:34	38.41%
r103_10_4	104.98	0-3-8-2-5-7-10-1-4-6-9-0	246	2036	534	1689	00:59	17.04%
r103_10_5	197.79	0-5-3-8-4-10-9-0 0-1-6-2-7-0	246	2036	350	929	00:11	54.37%
平均							00:29	39.10%
r103_20_1	246.23	0-1-4-3-11-13-14-2-12-0 0-8-7-9-19-17-18-6-5-16-15-0 0-10-20-0	736	7046	2001	2839	04:12	59.71%
r103_20_2	222.72	0-3-13-10-20-9-8-19-18-0 0-4-7-6-14-46-17-5-15-0 0-2-1-12-11-0	736	7046	2731	3264	07:36	53.68%
r103_20_3	258.32	0-9-10-19-20-0 0-7-6-4-8-14-5-17-18-16-15-0 0-1-2-12-3-13-11-0	736	7046	1876	2684	04:47	61.91%
r103_20_4	294.72	0-2-4-1-12-3-11-13-14-0 0-6-5-15-8-7-16-17-18-0 0-9-10-19-20-0	736	7046	3106	3179	07:28	54.88%
r103_20_5	234.73	0-4-3-13-2-12-1-11-14-0 0-6-7-5-17-16-15-0 0-8-10-9-20-18-19-0	736	7046	3515	3914	09:06	44.45%
平均							06:37	54.92%
平均								47.01%

五、巨集啟發式演算法設計與測

試結果

本研究所提之轉換作業，雖可大幅減少無效的路段決策變數，但對於作業決策變數(子作業)卻不易大幅降低，因此演算法若是配合子作業旅行成本矩陣進行求解，將無法有效突顯決策變數大量削減後的績效，有鑑於此，本研究另設計關連式旅行網路結構表，各別記錄每一個子作業與其他子作業間的連結關係，即可減少演算法求解過程中，每次搜尋回合中需重覆考慮的子作業數量。茲將關連式旅行網路結構表、巨集啟發式演算法架構與測試結果分述於後。

5.1 關連式旅行網路結構表

本研究設計關連式旅行網路結構，各別記錄每一個子作業與其他子作業間的連結關係，如表 13 所示，可進一步減少啟發式演算法求解過程中，每次搜尋回合中需重覆考慮的子作業數量。關連式旅行網路結構表使用方式條列如下。

表 13 關連式旅行網路結構表

作業點 (J_u)	子作業 (J_{ui})	-	(J_{ui}) 可前進的子作業(J_{vj})		-	可到達(J_{ui})的子作業(J_{vj})			
			(J_{ui}) 可前進的作業(J_v)			可到達(J_{ui})的作業(J_v)			
J_0	J_0	-	J_{10}	J_{20}	-	J_{31}	J_{32}	J_{46}	J_{47}
			J_1	J_2		J_3	J_4		
J_1	J_{10}	-	J_{20}	J_{31}	-	J_0			
			J_2	J_3		J_0			
J_2	J_{20}	-	J_{32}	J_{46}	-	J_0	J_{10}		
			J_3	J_4		J_0	J_1		
J_3	J_{31}	-	J_0	J_{46}	-	J_{10}			
			J_0	J_4		J_1			
	J_{32}	-	J_0	J_{47}	-	J_{20}			
			J_0	J_4		J_2			
J_4	J_{46}	-	J_0		-	J_{20}	J_{31}		
			J_0			J_2	J_3		
	J_{47}	-	J_0		-	J_{32}			
			J_0			J_3			

較求解 PDPTW 時可省略掉與服務時間相關

的限制，選擇加入路徑中的作業，

1. 在新產生的路徑中加入作業

首先產生一條尚未指派作業的新路徑，($J_0 \rightarrow J_0$)，然後選擇一個場站 J_0 可前進的、尚未被服務過的、且成本增量最小的作業加入新路徑中。從表 13 可知，場站 J_0 可前進的作業有作業 1 與作業 2。

(1) 若選擇加入作業 1，從表 13 可知， J_0 可前進到收貨作業點 J_1 的子作業為 J_{10} ，($J_0 \rightarrow J_{10} \rightarrow J_0$)，子作業 J_{10} 可前進到卸貨作業點 J_3 的子作業 J_{31} ，而 J_{31} 亦可回到場站 J_0 ，($J_0 \rightarrow J_{10} \rightarrow J_{31} \rightarrow J_0$)，因此，作業 1 可加入新路徑中。

(2) 若選擇加入作業 2，從表 13 可知， J_0 可前進到收貨作業點 J_2 的子作業為 J_{20} ，($J_0 \rightarrow J_{20} \rightarrow J_0$)，子作業 J_{20} 可前進到卸貨作業點 J_4 的子作業 J_{46} ，而 J_{46} 亦可回到場站 J_0 ，($J_0 \rightarrow J_{20} \rightarrow J_{46} \rightarrow J_0$)，因此，作業 2 也可加入新路徑中。

(3) 加入作業 1 的成本增量較小，因此選擇路徑($J_0 \rightarrow J_{10} \rightarrow J_{31} \rightarrow J_0$)進行後續指派作業。

2. 在已指派作業的路徑中，加入新作業

由於關連式旅行網路結構表中已清楚的表示了各作業間的連結關係，因此在已指派作業的路徑中，加入新作業時，可以以路徑中已指派的作業做為選擇新加入作業的依據，而不需逐一考慮所有尚未指派的作業是否可以加入路徑中。

或是選擇進行交換的作業時，亦可將結構表中未連結的作業忽略考慮，大幅減少了

應用啟發式演算法搜尋最佳解過程中許多重複且無效的步驟。請注意很重要的一點，關連式旅行網路結構表中並不記錄子作業間旅行成本的資料，如此可避免旅行成本資料重複記錄。演算法求解 SPDP 時仍延用原 PDPTW 中所使用的傳統旅行矩陣，記錄各作業點之間的旅行成本資料。

5.2 巨集啟發式演算法架構

演算法執行架構如圖 3 所示，路徑構建法採用最小成本插入法，路徑改善法為節點交換法，再配合巨集啟發式演算法之門檻接受法(Threshold Accepting, TA)輔助跳脫區域最佳解。

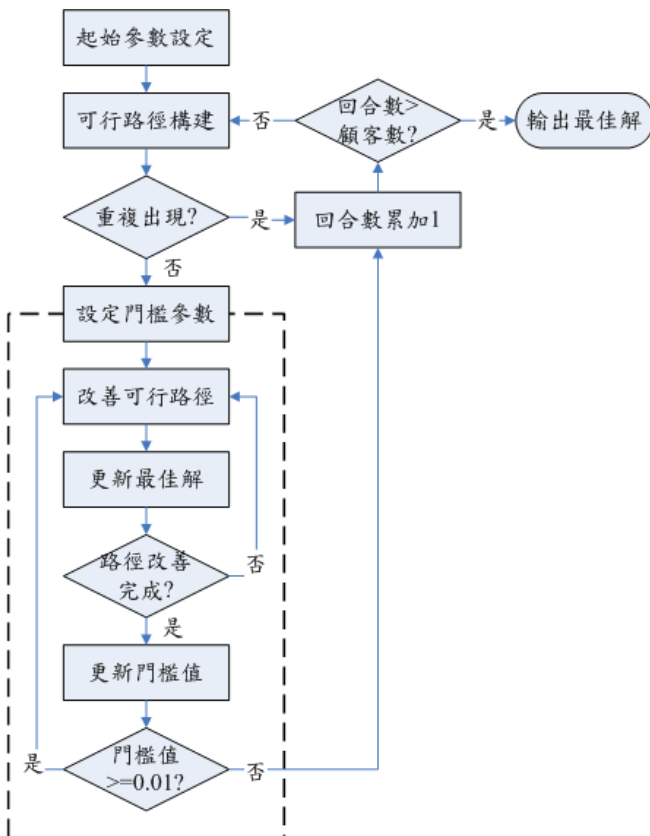


圖 3 巨集啟發式演算法測試架構圖

5.3 測試結果

本章節主要測試目的在測試轉換作業是否可以有效提升演算法求解 PDPTW 的績效，因此研究僅設計結構簡單的巨集啟發式演算法架構，並應用該演算法，在相同求解

回合數與時間限制條件下，比較直接求解 PDPTW 所得的近似解，與求解經過轉換處理後所產生的 SPDP 所得的近似解間的品質與求解時間，以評估本研究所提出的轉換作業是否可以有效提升求解績效。

1. 測試例題

測試例題為利用 4.2.1 小節中所敘述的方式，將國際標準題庫 VRPLIB 中規模為 100 個節點的 VRPTW 問題 6 題，分別轉換產生 3 個 PDPTW 例題，共 18 題測試例題。

2. 門檻收斂法最適參數測試

門檻收斂法中的門檻值與門檻長度為主要控制參數，目前已證實門檻的存在卻時能提升求解品質，但門檻的產生並無一定的規則(Penna, 1995)，且學者 Dueck and Scheuer (1990) 提出 TA 時亦未明確說明如何設定起始門檻，他們所使用的門檻值約為最佳成本的 0.25%。Toussaint and Golden (1994) 使用起始解成本的 1%、2%、2.5%，韓復華等人 (1997) 及韓復華與楊智凱 (1996) 使用起始解成本的 0.2% 至 2%，楊智凱 (1995) 則使用起始解成本的 0.4% 至 0.8%。門檻數列長度為門檻數列中門檻值的個數，目前文獻中對門檻數列長度亦未有明確的建議值，Dueck and Scheuer (1990) 使用的數列長度為 30，韓復華與王國琛 (2000) 使用 30 與 60 兩種長度，而韓復華與王國琛 (2002) 研究中提出門檻數列長度之大小對於求解效率並不敏感。

有鑑於此本研究針對起始門檻值設定 7 種參數值進行測試，分別為各回合起始解成本的 0.25%、0.5%、0.75%、1%、1.5%、2%、與 2.5% 等，門檻數列長度等於 30 進行測試，結果顯示起始門檻值比例等於 1.5% 時所求得之近似最佳解品質較佳，即以此做為後續例題測

表 16 轉換作業績效測試結果

題目名稱	最佳解 (A)	問題 轉換 時間	求解 SPDP			直接求解原 PDPTW			近似最佳 解差異 (C-B)/B%	求解時間 差異 (T2-T1)/T2%
			近似最 佳解 (B)	求解 時間 (T ₁ 秒)	與最佳 解差異 (B-A)/A%	近似最 佳解 (C)	求解 時間 (T ₂ 秒)	與最佳 解差異 (C-A)/A%		
R103_1	1292.68	0.80	1337.47	4.85	3.46%	1339.34	29.94	3.61%	0.14%	83.80%
R103_2	1292.68	0.80	1328.33	4.75	2.76%	1350.50	29.54	4.47%	1.67%	83.92%
R103_3	1292.68	0.80	1348.24	4.25	4.30%	1414.24	25.74	9.40%	4.90%	83.49%
R112_1	982.14	0.80	1053.92	8.25	7.31%	1132.60	53.54	15.32%	7.47%	84.59%
R112_2	982.14	0.80	1005.70	8.55	2.40%	1129.61	52.54	15.02%	12.32%	83.73%
R112_3	982.14	0.80	998.64	7.95	1.68%	1123.52	50.44	14.40%	12.50%	84.24%
R208_1	726.82	1.84	748.71	10.20	3.01%	863.15	116.982	18.76%	15.29%	91.28%
R208_2	726.82	1.52	741.39	11.24	2.00%	842.90	96.735	15.97%	13.69%	88.38%
R208_3	726.82	2.00	824.21	11.99	13.40%	944.31	146.822	29.92%	14.57%	91.84%
R210_1	939.37	1.52	983.95	7.65	4.75%	1107.53	80.734	17.90%	12.56%	90.53%
R210_2	939.37	1.52	981.27	6.21	4.46%	1058.56	62.784	12.69%	7.88%	90.12%
R210_3	939.37	1.52	983.95	7.03	4.75%	1050.53	69.834	11.83%	6.77%	89.94%
RC204_1	798.46	1.60	835.97	9.84	4.70%	887.32	109.544	11.13%	6.14%	91.02%
RC204_2	798.46	1.68	868.29	9.52	8.75%	890.00	105.674	11.46%	2.50%	90.99%
RC204_3	798.46	1.52	835.97	8.28	4.70%	865.59	89.014	8.41%	3.54%	90.70%
RC205_1	1297.65	1.36	1352.36	8.71	4.22%	1361.60	79.122	4.93%	0.68%	88.99%
RC205_2	1297.65	1.36	1359.42	9.13	4.76%	1506.02	85.362	16.06%	10.78%	89.30%
RC205_3	1297.65	1.36	1352.36	9.11	4.22%	1465.48	83.022	12.93%	8.36%	89.03%
解平均 差異%					4.76.0%			13.01%	7.88%	88.10%

試時使用的起始門檻參數數值。

3. 轉換作業績效測試

測試結果如表 16 所示，表中 SPDP 求解時間(T1)已包含問題轉換所需時間。各問題轉換作業(包括時窗分割、問題規模精簡、以及關連式旅行網路結構表建立)所花費的時間，處理 100 節點規模的問題，處理時間均不超過 2 秒，處理之計算機等級為 Intel Centrino Core 2 Duo (2.0GHZ), 2.00G DDR RAM。求解結果與最佳解間平均差異雖有 5%，但作為本研究進行轉換作業提升啟發式演算法績效測試則已足夠。由表中明顯可以看出轉換作業配合使用本研究所提出的關連式旅行網路結構表，確實可大幅提升 PDPTW 的求解績效，解品質平均提升 7.88%，求解時間更是大幅減少了 88.10%。改善績效明顯卓越，期望本研究成果對後續學者進行 PDPTW 相關研究應有不少助益。

六、研究結論

本研究結論如下：

1. 本研究提出時窗分割與整數化策略、問題規模精簡策略以及關連式旅行網路結構表，成功地將 PDPTW 轉換為 SPDP，也就是在求解過程中不再需要考慮時窗的限制。以本研究測試例題而言，將問題規模為 100 節點的 PDPTW 轉換 SPDP，耗時皆小於 2 秒，平均需時僅 1.31 秒，即顯示本研究所提出的構想是有價值的，因此未來求解 PDPTW 則可以利用既有文獻的 PDP 啟發式演算法。
2. 本研究將 PDPTW 轉換所得的 SPDP 與原來的 PDP 差異很小，過去學者針對 PDP 所設計之演算法，可以很容易應用來求解 SPDP。
3. 研究中設計之問題規模精簡策略，確實可

以大幅減少求解時需考慮的決策變數數量。經本研究測試，求解小規模問題精確解方面，SPDP 限制式總量，較 PDPTW 限制式總量，平均減少 47.01%。

4. 研究構建 SPDP 數學規劃模式，撰寫 LINGO 程式進行求解，發現當時窗分割時段足夠小時，SPDP 之精確解即為 PDPTW 的精確解。
5. 大規模問題測試結果顯示，應用本研究所提出的轉換作業與關連式旅行網路結構表，較直接求解原 PDPTW 而言，可提升求解品質平均 7.88%，與減少求解時間達 86.16%，確實可以在較短的時間內求出品質較佳的近似最佳解，有效提升求解效率。期望本研究成果對後續學者進行 PDPTW 相關研究應有不少助益。

七、計畫成果自評

本計畫第一年成果，目前已發表期刊論文 2 篇，如下：

1. 李泰琳、張靖「應用時窗分割與整數化策略簡化時窗收卸貨問題之研究」，運輸學刊 (TSSCI)，第二十卷，第三期，pp.313-350，民國 97 年 9 月。
2. 李泰琳、張靖「調適型導引螞蟻演算法求解時窗收卸貨問題之研究」(送審運輸計劃季刊，TSSCI)。

本研究符合預期結果。

參考文獻

- 劉佩玲 (2006)，在考量資源限制下國際快遞業服務中心位置選擇，國立高雄第一科技大學運籌管理所碩士論文。
- 韓復華、王國琛 (2002)，「巨集啟發式解法在求解大規模旅行推銷員問題之應用」，運輸學刊，第十四卷第二期，頁 1-14。
- 楊智凱 (1995)，以門檻接受法改善 TSP 與 VRP 路網成本之研究，國立交通大學土木研究所運工管組碩士論文。

- 韓復華、王國琛 (2000), 「巨集啟發式解法在求解大規模旅行推銷員問題之研究」, *中華民國第五屆運輸網路研討會論文集*, 頁 195-204。
- 韓復華、楊智凱 (1996), 「門檻接受法在 TSP 問題上之應用」, *運輸計畫季刊*, 第二卷第二十五期, 頁 163-188。
- 韓復華、楊智凱、卓裕仁 (1997), 「應用門檻收斂法求解車輛路線問題之研究」, *運輸計畫季刊*, 第二卷第二十六期, 頁 253-280。
- Appelgren, L. H. (1969), "A Column Generation Algorithm for a Ship Scheduling Problem," *Transportation Science*, Vol. 3, Iss. 1, pp. 53-68.
- Appelgren, L. H. (1971), "Integer Programming Methods for a Vessel Scheduling Problem," *Transportation Science*, Vol. 5, Iss. 1, pp. 62-74.
- Bent, R. and Van Hentenryck, P. (2006), "A Two-Stage Hybrid Algorithm for Pickup and Delivery Vehicle Routing Problems with Time Windows," *Computers and Operations Research*, Vol. 33, Iss. 4, pp. 875-893.
- Desrosiers, J., Dumas, Y., Solomon, M. M. and Sournis, E. (1995), *Time Constrained Routing and Scheduling*, Handbooks in Operations Research and Management Science: Volume 8: Network Routing, Chapter 2, pp.35-139.
- Dror, M (1994), "Note on the Complexity of the Shortest Path Models for Column Generations in the VRPTW," *Operations Research*, Vol. 42, Iss. 5, pp. 977-978.
- Dueck, G. and Scheuer, T. (1990), "Threshold Accepting: A General Purpose Optimization Algorithm Appearing Superior to Simulated Annealing," *Journal of Computational Physics*, Vol. 90, Iss. 1, pp. 161-175.
- Dumas Y., Desrosiers J. and Soumis F. (1991), "The Pickup and Delivery Problem with Time Windows," *European Journal of Operational Research*, Vol. 54, Iss. 1, pp. 7-22.
- Fabián J. and Pérez L. (2005), "A Meta-heuristic Applied for a Topologic Pickup and Delivery Problem with Time Windows Constraints," *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, Vol. 3516, pp. 924-928.
- Graham, D. and Nuttle, H. L. W. (1986), "A Comparison of Heuristics for a School Bus Scheduling Problem," *Transportation Research Part B*, Vol. 20, Iss. 3, pp. 175-182.
- Haibing, L. and Andrew, L. (2003), "A Metaheuristic for the Pickup and Delivery Problem with Time Windows," *International Journal on Artificial Intelligence Tools*, Vol. 12, Iss. 2, pp.173-186.
- Kohl, N. (1995), Exact Methods for Time Constrained Routing and Related Scheduling Problems, Ph.D. Dissertation, Technical University of Denmark, Denmark.
- Kolen, A. W. J., Rinnooy Kan, A. G. H. and Trienekens, H. W. J. M. (1987), "Vehicle Routing with Time Windows," *Operations Research*, Vol. 35, Iss. 2, pp. 266-273.
- Lau, H. C. and Liang, Z. (2002), "Pickup and Delivery with Time Windows - Algorithms and Test Case Generation," *International Journal on Artificial Intelligence Tools*, Vol. 11, No. 3, pp. 455-472.
- Levin, A. (1971), "Scheduling and Fleet Routing Methods for Transportation System," *Transportation Science*, Vol. 5, Iss. 3, pp. 232-255.
- Lu, Q. and Dessouly, M. (2004), "An Exact Algorithm for the Multiple Vehicle Pickup and Delivery Problem," *Transportation Science*, Vol. 38, Iss. 4, pp. 503-514.
- Mitrovic-Minic, S. and Laporte, G. (2004), "Waiting Strategies for The Dynamic Pickup and Delivery problem with Time Windows," *Transportation Research Part B*, Vol. 38, Iss. 7, pp. 635-655.
- Nagy, G. and Salhi, S. (2005), "Heuristic Algorithms for Single and Multiple Depot

Vehicle Routing Problems with Pickups and Delivery,” *European Journal of Operation Research*, Vol. 162, Iss. 4, pp.126-142.

- Nanry W. P. and Barnes J. W. (2000), “Solving the Pickup and Delivery Problem with Time Windows Using Reactive Tabu Search,” *Transportation Research Part B*, Vol.34, Iss. 2, pp.107-121.
- Penna, T. J. P. (1995), “Traveling Salesman Problem and Tsallis Statistics,” *Physical Review E*, Vol. 51, Iss. 1, pp. R1-R3.
- Psarafis, H. (1980), “A Dynamic Programming Solution to the Single Vehicle Many to Many Immediate Request Dial-a-Ride Problem,” *Transportation Science*, Vol.14, Iss. 2, pp.130-54.
- Psarafis, H. (1983), “An Exact Algorithm for the Single Vehicle Many-to-Many Immediate Request Dial-a-Ride Problem,” *Transportation Science*, Vol. 17, Iss. 4, pp. 351-361.
- Savelsbergh, M. W. P. and Solomon, M. (1995), “The General Pickup and Delivery Problem,” *Transportation Science*, Vol. 29, Iss. 1, pp.17-29.
- Sexton, T. R. and Bodin, L. D. (1985), “Optimizing Single Vehicle Many-to-Many Dial-a-Ride Problem with Desired Delivery Time: I Scheduling,” *Transportation Science*, Vol.19, Iss. 4, pp.378-410.
- Sexton, T. R. and Bodin, L. D. (1985), “Optimizing Single Vehicle Many-to-Many Dial-a-Ride Problem with Desired Delivery Time: II Routing,” *Transportation Science*, Vol.19, Iss. 4, pp.411-435.
- Swersey, A. and Ballard, W. (1984), “Scheduling School Buses,” *Management Science*, Vol. 30, No. 7, pp. 844-853.
- Toussaint, K. J. and Golden, B. L. (1994), “Exchange Heuristics to Improve the Clarity of Base/Time Plots,” *Computers and Operations Research*, Vol. 21, Iss. 5, pp. 573-586.
- Wang, X. (2000), Algorithms and Strategies for Dynamic Carrier Fleet Operations for an Application to Local Trucking