

以啟發式演算法求解單一場站多車種 同時收送貨之車輛途程問題

A Heuristic Algorithm for Single Depot Vehicle Routing Problem with Simultaneous Pickup and Delivery

朱經武 Ching-Wu Chu¹

周偉禮 Wei-Li Chou²

摘要

由配送中心運送貨物給顧客是配銷管理者每日所面臨的決策問題，如何有效率將貨物運送給顧客對配銷管理而言，是一項很重要的決策，因為運輸成本在配銷各項成本中所佔之比例很高，除此之外，速度也是一完善物流系統不可或缺的服務。

本研究中以單一配送中心為研究對象，並以實務中多車種同時收送貨車輛途程問題 (Vehicle Routing Problem with Simultaneous Pickup and Delivery: VRPSPD) 為探討情境，研究之主要目的為在於滿足顧客需求及考慮公司成本下，發展一演算法，幫助配銷管理者解決規劃運送路線問題。研究中建構數學規劃模式與發展啟發式演算法，其中數學規劃模式僅適合小型問題求解與驗證演算法精確度之用。經測試比較後顯示啟發式演算法之效率與準確度均十分良好。

關鍵詞：同時收送貨車輛途程問題、啟發式演算法、0-1 整數規劃、運籌

ABSTRACT

How to efficiently delivery goods to customers from a depot is a daily and an important decision for the logistics managers, because the transportation cost accounts for a large portion of the distribution cost. Furthermore, the speed of delivery is also an indispensable element of a sound logistics system.

A single-depot vehicle routing problem with simultaneous pickup and delivery is studied in this paper. The main purpose of this paper is to develop a heuristic algorithm facilitating the logistics managers in planning the delivery routes under the consideration of customers' demands and operating costs. Both the mathematical model

¹國立台灣海洋大學航運管理系教授 E-mail:cwchu@mail.ntou.edu.tw

²陽明海運股份有限公司工程部工程師 E-mail:wesleychou@yml.com.tw

and heuristic algorithm are developed in this paper. The mathematical model is suitable for small size problem and developed for comparing the accuracy with the heuristic algorithm. From the empirical results, we know that the heuristic algorithm performs well in terms of efficiency and accuracy.

Keywords: VRP with simultaneous pickup and delivery, Heuristics, 0-1 integer programming, Logistics

壹、前言

由配送中心運送貨物給顧客是配銷管理者每日所面臨的決策問題，如何有效率將貨物運送給顧客對配銷管理而言，是一項很重要的決策，因為運輸成本在配銷各項成本中所佔之比例很高，除此之外，速度也是一完善物流系統不可或缺的服務，例如目前全球前幾大個人電腦公司（IBM, DELL）均努力達到 983 模式，亦即 98%產品將於下單三日內送達顧客手中。

現實生活中之專業物流公司擁有場站（配銷中心），同時其車隊亦為多種不同噸位之車輛組合而成，專業物流公司的配送問題即為單一場站多車種車輛途程問題。目前之相關研究均侷限於僅考量送貨或回程收貨之情境，但從日常生活觀察中可以發現一些同時收送貨之情形。例如，配送貨物至賣場時，賣場會要求將空瓶、退貨、裝運貨物之籃框、棧板等順道運回。此種情境即演變為單一場站多車種同時收送貨問題（Vehicle Routing Problem with Simultaneous Pickup and Delivery, VRPSPD）。

理想的狀況下，配銷管理者應完整的分析一組運送者、車種分配和路線選擇，進而發展出一運送計畫，也就是最適解（exact procedure）。不過這項分析將十分複雜，由於車輛途程問題的計算複雜度屬於 NP-Hard 的問題，解決一車輛途程問題，需要許多的計算資源，若以數學規劃求解，不可能在合理時間內獲得結果，因此在可接受時間內，求得一高精準度的演算法協助配銷管理者解決上述問題有其必要性。

探討車輛途程問題（Vehicle Routing Problem）之相關文獻眾多，大部分以討論啟發式演算法為主。因為以數學規劃求解會遭遇圍圈破解問題。啟發式演算法可以廣泛地分成傳統與萬用兩大類 Laporte 等（2000）。傳統演算法發展於 1960 至 1990 間。一般又可分成：

- (1) 途程建構(tour building heuristics)
- (2) 途程改良(tour improvement heuristics)
- (3) 兩階段法(two-phase method)
- (4) 不完全最佳化方法(incomplete optimization method)等四大類。

最常被應用的途程建構啟發式方法為 Clarke and Wright (1964) 的節省法。已經有多位學者修改 Clarke and Wright 的節省法，例如，Gaskell (1967) and Yellow (1970) 分別引進一修改節省法的概念，其中節省值的計算，可藉由改變兩節點間旅行成本而得。

途程改良的演算法是由 Lin (1965) 提出，Lin and Kernighan (1973) 將其運用於旅行銷售員問題。Christofides and Eilon (1969) 修改途程改良演算法，並將其運用於車輛途程問題。兩階段法以 Gillette and Miller (1974)、Christofides 等 (1979) 之著作最為有名。不完全最佳方法是一種樹結構搜尋，在 Christofides (1979) 中有詳細的報告。

萬用演算法發展於過去十年間，以下僅回顧禁忌搜尋(Tabu Search)，因為禁忌搜尋是所有萬用演算法當中，應用於車輛途程問題最成功與最有效的方法。Osman (1993)、Taillard (1993)、Gendreau 等 (1994)、Rochat and Taillard (1995)、Xu and Kelly (1996)、Rego and Roucairol (1996) 等，上述這些作者將禁忌搜尋應用於不同類型之車輛途程問題，均獲得滿意的結果。

回程收貨之車輛途程問題 (Vehicle Routing Problem with Backhaul) 是 VRP 問題的延伸，其定義：車輛由配送中心出發，載貨分送給各需求點，送完所有貨後，回程再到各需求點收貨後才回配送中心。這一類型問題的需求點分為兩個集合，一為送貨需求點的集合，另一個為收貨需求點的集合。在這個問題中，限制送貨需求點與取貨需求點不能混合作業，也就是說車輛在每個途程中必須先服務送貨需求點後，才可以服務收貨需求點。其考慮重點是避免在收貨或送貨後必須重新整理貨櫃之安排，來空出搬運與裝載貨物的空間。

在許多文獻中雖提到回程收貨之車輛途程問題，但探討同時收送貨之文獻並不多見。Schruben and Clifton (1968) 提出類似節省法之解法將其運用於農產品同時送貨及收貨的情形，其問題僅有六個需求點，每個載貨量皆不同。Psarafits (1980) 考慮一動態即時的 dial-and-ride 問題，其研究背景為一部車與 14 個需求點。Min (1989) 求解每個需求點均有收送貨之圖書、錄影帶配送問題，研究中包含兩部相同車種與 22 位顧客。Salhi and Nagy (1999) 針對 Christofides (1979) 之車輛途程問題作修改，用以測試含同時收送貨之問題，其研究背景為多場站、多車種、顧客數目介於 50 至 249 之間。近期 Nagy and Salhi (2005) 提出啟發式演算法求解單一和多場站同時收送貨之問題，其研究的目標是為同時及混合收送貨問題兩者共同建立一個有效的啟發式探討，進而延伸到多場站收送貨問題的解決方案，且排除先送貨再收貨問題，研究中使用許多測試範例，求解效果良好。

本研究異於前述文獻，在於考慮不同車種之同時收送貨車隊規劃，並改良 Clarke and Wright 方法，引進車輛之固定成本；最佳化過程中，考慮多種改善程序

之執行順序，可導致更良好的結果。本研究以單一配送中心為研究對象，考慮不同車種下之同時收送貨車輛途程問題，主要研究目的為發展一啟發式演算法，協助運銷管理者降低營運成本，增加競爭優勢。

本文組織結構如下，第二節建構整數規劃模式，啟發式演算方法於第三節詳加介紹，並於第四節執行範例測試與分析，最後提出結論與建議。

貳、整數規劃模式

為簡化問題與研究，遂有下列之假設：

1. 僅有一配送中心（Depot），且車輛運輸起始點與終點皆在配送中心。
 2. 每一位顧客的送貨量與收貨量為已知且不超過任一車輛的裝載量(包括重量與體積)。
 3. 每一位顧客只能由一車輛服務，且所有顧客需求必須被滿足。
 4. 考慮貨運公司針對每位顧客同時進行收送貨的情況。
 5. 允許收貨量大於送貨量的情境。
 6. 車輛運輸成本，可分為固定成本與變動成本兩部分。固定成本包含諸如車輛的保險、折舊與人員薪資等，變動成本與運送距離成正比(\$/每公里)，例如油資。
- 整數規劃數學模式敘述如下：

$$\text{Min } z = \sum_k^m FC_k + \sum_i^n \sum_j^n \sum_k^m C_{ijk} X_{ijk}$$

subject to

$$\sum_{k=1}^m Y_{1k} \leq m \quad (1)$$

$$\sum_{k=1}^m Y_{ik} = 1 \quad (i = 1, 2, \dots, n) \quad (2)$$

$$\sum_{\substack{j=1 \\ i \neq j}}^n X_{ijk} = Y_{ik} \quad (i = 1, 2, \dots, n; \quad k = 1, 2 \dots, m) \quad (3)$$

$$\sum_{\substack{j=1 \\ i \neq j}}^n X_{jik} = Y_{ik} \quad (i = 1, 2, \dots, n; \quad k = 1, 2 \dots, m) \quad (4)$$

$$\sum_{\substack{j=1 \\ i \neq j}}^n X_{ijk} = \begin{cases} U_{ik}, & i = 2, 3, \dots, n \\ V_{ik}, & k = 1, 2, \dots, m \end{cases} \quad (5)$$

$$\sum_{\substack{j=1 \\ i \neq j}}^n X_{ijk} = \begin{cases} U_{ik}, & i = 2, \dots, n \\ V_{ik}, & k = 1, \dots, m \end{cases} \quad (6)$$

$$\sum_{i=2}^n a_i U_{ik} \leq Q_k \quad (k = 1, 2, \dots, m) \quad (7)$$

$$Z_{ik} - a_j U_{jk} + b_j V_{jk} \leq Z_{jk} + M(1 - X_{ijk})$$

(i = 1, 2, \dots, n; j = 1, 2, \dots, n; k = 1, 2, \dots, m; i \neq j) \quad (8)

$$Z_{ik} \leq M \sum_{\substack{j=1 \\ i \neq j}}^n X_{ijk} \quad (i = 2, 3, \dots, n; k = 1, 2, \dots, m) \quad (9)$$

$$Z_{ik} \leq Q_k \quad (i = 1, 2, \dots, n; k = 1, 2, \dots, m) \quad (10)$$

$$\begin{cases} Y_{ik} - U_{ik} = 0, \\ Y_{ik} - V_{ik} = 0, \end{cases} \quad (i = 1, 2, \dots, n; k = 1, 2, \dots, m) \quad (11)$$

$$\forall X_{ijk}, Y_{ik}, U_{ik}, V_{ik}, Z_{ik} \in \{0, 1\}$$

以下就模式的變數、符號與意義做一說明。

m : 代表車輛數。

a_i : 表顧客 i 所需送貨數量。

b_i : 表顧客 i 收貨數量。

Q_k : 表車輛 k 的承載量。

FC_k : 表車輛 k 的固定成本。

C_{ijk} : 表車輛 k 由顧客 i 到顧客 j 間的運送成本。

Z_{ik} : 表車輛 k 離開 i 地時，所載運的貨物數量。

$$X_{ijk} = \begin{cases} 1, & \text{車輛 } k \text{ 從顧客 } i \text{ 行駛至顧客 } j \\ 0, & \text{其他} \end{cases}$$

$$U_{ik} = \begin{cases} 1, & \text{顧客 } i \text{ 接受車輛 } k \text{ 的送貨服務} \\ 0, & \text{其他} \end{cases}$$

$$V_{ik} = \begin{cases} 1, & \text{顧客 } i \text{ 接受車輛 } k \text{ 的收貨服務} \\ 0, & \text{其他} \end{cases}$$

$$Y_{ik} = \begin{cases} 1, & \text{顧客 } i \text{ 需求由車輛 } k \text{ 的服務} \\ 0, & \text{其他} \end{cases}$$

$$i = 1, 2, \dots, n; j = 1, 2, \dots, n; k = 1, 2, \dots, m$$

其中 1 代表配送中心。

目標函數的目的為求取總運送成本最低。

限制式(1)確保由配送中心出發的車輛數目小於等於所有車輛數目。

限制式(2)確保每一位顧客僅受一輛車服務。

限制式(3)確保一地只允許一車離開。

限制式(4)確保一地只允許一車進入。

限制式(5)、(6)實際配送限制。

限制式(7)各車欲離開配送中心時的承載量小於等於車容量。

限制式(8)、(9) 各車裝卸均衡限制式，離開顧客時的承載量小於等於車容量限制式，其中 M 為一非常大之正數。

限制式(10) 各車離開該顧客時的車容量限制。

其中限制式(8)原應為 $X_{ijk} = 1$ ，則 $Z_{ik} - a_j U_{jk} + b_j V_{jk} = Z_{jk}$ 。但這條件性等式要符合線性規劃的規格，所以須將上式轉化為(8)有效不等式，以求得線性整數最佳解。

參、啟發式演算法

本研究之啟發式演算法，可分成二大程序：(1) 建構起初解程序 (2) 最佳化程序。

3.1 建構起初解程序

為建構可行起初解，本程序包含三個子程序：(1) Clarke-Wright 節省法 (2) 搬移與插入程序(Move and Insert Procedure) (3) 檢查 (Check) 與重新安排程序(Rearrange Procedure)。

3.1.1 Clarke-Wright 節省法

因 Clarke-Wright 節省法求解速度最為快速，為大部分研究車輛途程學者建構起始解時所採用之演算法，本研究亦引用此法，其執行步驟如下：

- (1) 形成從公司運送到單一顧客之路線與計算所需成本之總和。
- (2) 計算合併任兩位顧客 i 與 j 可節省之成本， S_{ij} 。

$$S_{ij} = t_{io} + t_{oj} - t_{ij}$$

其中 S_{ij} = 合併顧客 i 與 j 需求量運送時，所能節省之成本

t_{io} = 從顧客 i 運送至公司所需之成本

t_{oj} = 從公司運送至顧客 j 所需之成本

t_{ij} = 在顧客 i 與顧客 j 之間所需之成本

- (3) 依 S_{ij} 由大至小排序。
- (4) 將成本最大節省之二位顧客合併，並檢視其需求量（送貨量）是否符合車容量限制。假若可行，則進行合併；若不可行，則檢視下一個次大節省成本。
- (5) 繼續進行步驟（4），直到所有具有正的節省成本均列入考量為止。

3.1.2 搬移與插入程序

由於 Clarke-Wright 節省法有可能無法將所有顧客安排在路線中，所以本程序主要目的是將所有顧客安排在所有路線中。

本程序之執行步驟簡述如下：

- (1) 首先檢視是否有顧客被排除於路線中。假若是，則進行下一步驟；假若否，則不執行本程序，直接進行 3.1.3 節之重新安排程序。
- (2) 檢視被排除於路線中顧客 x 之需求， $q(x)$ ，掃瞄路線 r 之剩餘載貨量， $unload(r)$ ，如果 $q(x) \leq unload(r)$ ，則將顧客 x 插入路線 r 之中，並更新 $unload(r) = unload(r) - q(x)$ ；如果 $q(x) > unload(r)$ ，則繼續掃瞄其他路線，直到所有路線均被檢視完畢。
- (3) 若顧客 x ，已被安排於路線中，則進行步驟（5）；若顧客 x 未被安排於路線中，則進行步驟（4）。
- (4) 檢視任兩條路線 J 與 L ，並掃瞄 J 路線上任一顧客 k ，若 $INSERT_Capacity \geq 0$ 與 $MOVE_Capacity \geq 0$ 成立，則將顧客 x 插入路線 J 中，並將顧客 k 搬移至路線 L 中，順便更新兩路線車輛剩餘容量， $unload(J)$ 與 $unload(L)$ ，接著執行步驟（5）。

其中 $INSERT_capacity = unload(J) - q(x) + q(k)$

$$MOVE_capacity = unload(L) - q(k)$$

若 $INSERT_capacity$ 與 $MOVE_capacity \geq 0$ 不成立，更新顧客 k ，繼續執行步驟（4）直至路線 J 之所有顧客均被檢視完畢。若顧客 x 仍未被安排

至路線內，更新路線 J 與 L，繼續執行步驟（4），直至所有可能之路線組合均被考慮過。

- (5) 檢查是否還有顧客被排除於路線中，假若是，則進行步驟（2）；假若否，則停止。

3.1.3 檢查與重新安排程序

本研究允許收貨量大於送貨量，依據 3.1.2 所產生之路徑，有可能為不可行，因為前述二程序均以需求量（送貨量）為考量設計程式。檢查程序用以檢查路徑之可行性，若產生不可行之路徑則執行重新安排程序，否則直接進行 3.2 之最佳化程序。

檢查程序為一簡單的程序，用以檢查所有路徑之可行性。令 $arrival(x)$ 與 $leave(x)$ 分別代表卡車到達與離開顧客 x 時之載貨量。其中 $arrival(x)$ 與 $leave(x)$ 可計算如下：

$$leave(x) = arrival(x) - q(x) + p(x)$$

$$arrival(x) = leave(x-1)$$

符號 $q(x)$ 與 $p(x)$ 分別代表顧客 x 之送貨量與收貨量； $x-1$ 代表路徑中顧客 x 的前一位顧客。

重新安排程序為一程序將服務顧客之順序重新安排，以便產生可行路線。令 $reduce_load = q(x) - p(x)$ ，其中 $reduce_load(x)$ 為服務顧客 x 時，車輛所能減少之載貨量。

重新安排程序之執行步驟簡述如下：

- (1) 依據 3.1.2 之執行結果，分別於各路線計算每位顧客之 $reduce_load(x)$ 。
- (2) 各路線依據 $reduce_load(x)$ 重新由大至小排列。
- (3) 步驟（2）所對應之結果即為起始解。

3.2 最佳化程序

在獲得起初解之後便執行本程序，其中包含一連串的路線內改善程序、路線間改善程序與搜尋程序。

3.2.1 路線內改善

考慮將同一路線內任意不相鄰的兩路段交換，這項程序類似 Kindervater 與 Savelsbergh (1997) 敘述之程序。其步驟如圖 1 所示，交換兩路段係藉由將路段 (m, p) 與 (n, q) 取代原路段 (m, n) 與 (p, q) 。首先計算路段交換之改善值 $cost(m, n) + cost(p, q) - cost(m, p) - cost(n, q)$ 。若改善值大於零，則更新路段，否則繼續交換其他任

意不相鄰的兩路段直至路線內之所有路段皆被考慮完畢。

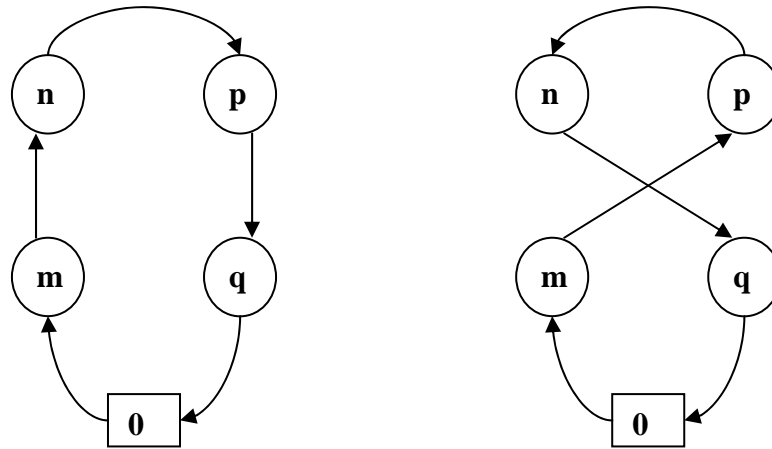


圖 1 路線內任意不相鄰兩路段交換示意圖

3.2.2 路線間改善

考慮交換不同路線間之顧客，可分成 1 位顧客交換、2 位顧客交換與 4 位顧客交換，其步驟分別如圖 2、圖 3 與圖 4 所示。圖 2 中，1 位顧客交換，係將路線 a 的 m 顧客插入至路線 b，藉由將路段 (l, m)、(m, n)、(p, q) 移除，取代成路段 (l, n)、(p, m)、(m, q)。首先計算路線交換之改善值 $cost(l,m) + cost(m,n) + cost(p,q) - cost(l,n) - cost(p,m) - cost(m,q)$ 。若符合(1)改善值大於零，(2)路線 a 的 m 顧客需求加上路線 b 目前的總載貨量不得超過車容量，若滿足上述兩條件則將路線 a 的 m 顧客插入至路線 b 的顧客 p 與 q 之間。

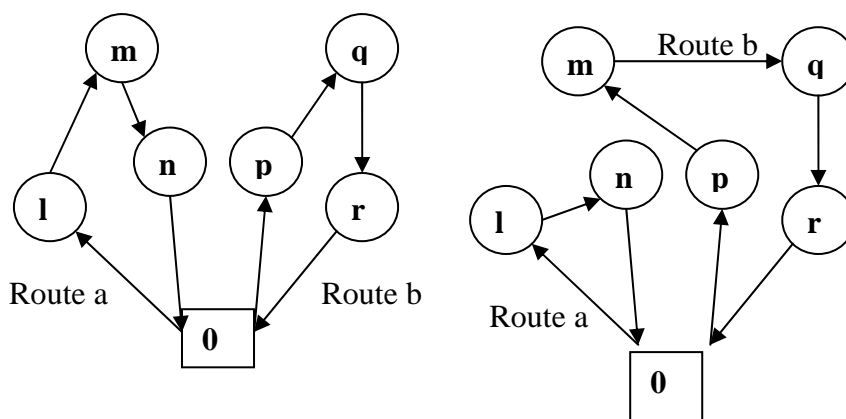


圖 2 路線間 1 位顧客交換示意圖

圖 3 中，2 位顧客交換，係將路線 a 的 m 顧客與路線 b 的 q 顧客交換，藉由將

路段 (l, m) 、 (m, n) 、 (p, q) 、 (q, r) 移除取代成爲路段 (l, q) 、 (q, n) 、 (p, m) 、 (m, r) 。首先計算路線交換之改善值 $\text{cost}(l,m)+\text{cost}(m,n)+\text{cost}(p,q)+\text{cost}(q,r)-\text{cost}(l,q)-\text{cost}(q,n)-\text{cost}(p,m)-\text{cost}(m,r)$ 。若符合(1)改善值大於零，(2)交換後各路線的總載貨量不得超過車容量，若滿足上述兩條件則將路線 a 的 m 顧客與路線 b 的 q 顧客交換。

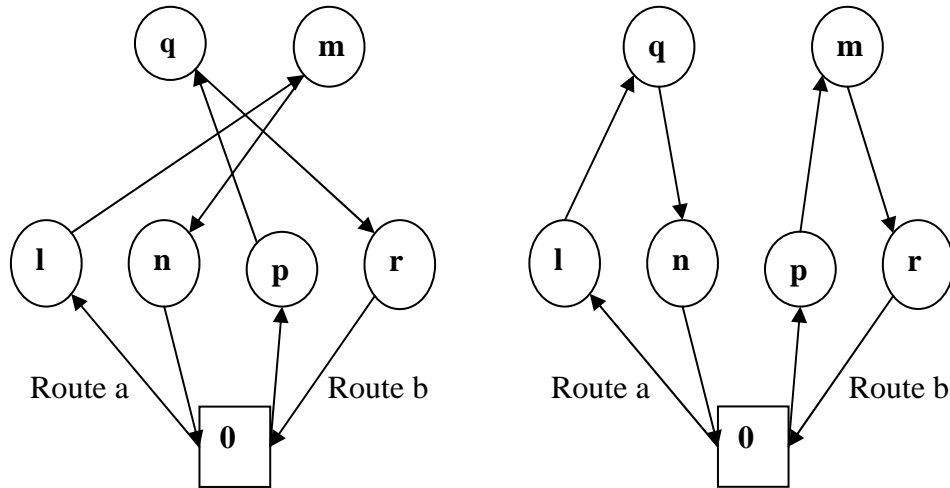


圖 3 路線間 2 位顧客交換示意圖

圖 4 中，4 位顧客交換，係將路線 a 相鄰兩位顧客 l、m 與路線 b 相鄰兩位顧客 q、r 交換，藉由將路段 (k, l) 、 (m, n) 、 (p, q) 、 (r, s) 移除取代成爲 (k, q) 、 (r, n) 、 (p, l) 、 (m, s) 首先計算路線交換之改善值 $\text{cost}(k, l)+\text{cost}(m, n)+\text{cost}(p, q)+\text{cost}(r, s)-\text{cost}(k, q)-\text{cost}(r, n)-\text{cost}(p, l)-\text{cost}(m, s)$ ；若符合(1)改善值大於零(2)交換後各路線的總載貨量不得超過車容量，若滿足上述兩條件，則將路線 a 的顧客 l、m 與路線 b 的兩位顧客 g、r 交換。

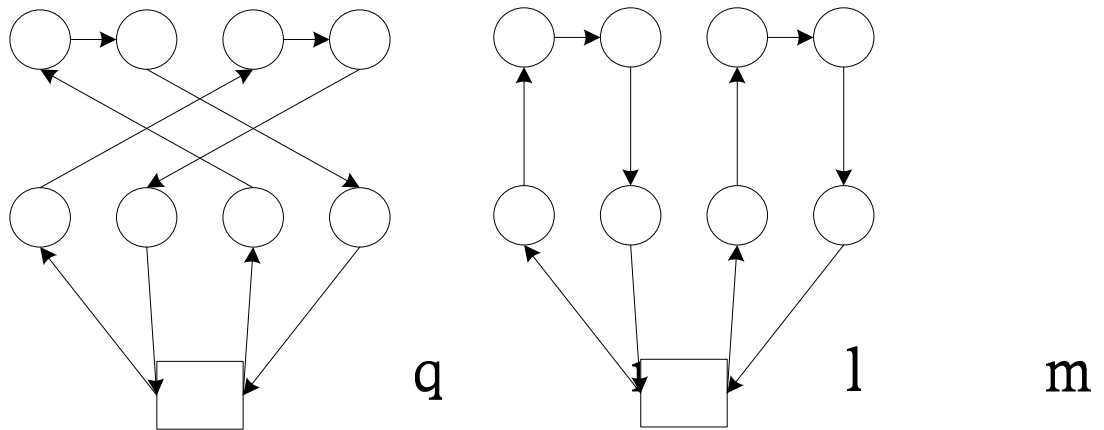


圖 4 路線間 4 位顧客交換示意圖

3.2.3 搜尋

由廣泛的測試瞭解，執行路線內與路線間改善程序之先後順序會影響求解的品質，因此設計一迴路安排不同順序之路線內與路線間改善。一旦獲得更佳解，便立刻更新路線與更新改善程序之順序，直至所有改善程序的排列組合均被執行完畢，執行過程中，前述之檢查程序應用於所有搜尋之結果，避免產生不可行路線。

肆 範例說明

Route b

本節主要的目的是依據第二節與第三節所建立的整數規畫法與啟發式演算法，來驗證演算法解題精確度與解題效率。4.1 節敘述測試例題的蒐集，4.2 節提出求解的結果。

4.1 測試例題之蒐集與說明

本研究所採用的測試例題取自 droopy.imag.fr 的 EILLIB 題庫加以修改而成，均屬滿足三角不等式的平面座標而各題需求點的座標則大致為均勻分佈，每位顧客之送貨量取自原題庫，每位顧客之收貨量由隨機亂數產生，其範圍為 0 與送貨量兩倍之間。以下將本研究所利用的範例資料整理如表 1，詳細之需求點座標與收送貨量請參閱附錄：

表 1 測試例題相關資料

測試範例編號	車輛容量	自有車輛 固定成本	相關變動成本
測試範例 1-1	50, 30	450, 250	\$1.5/per mile
測試範例 1-2	50, 30	450, 250	\$1.5/per mile
測試範例 1-3	50, 30	450, 250	\$1.5/per mile
測試範例 1-4	50, 30	450, 250	\$1.5/per mile
測試範例 2-1	150, 100	1200, 800	\$1.5/per mile
測試範例 2-2	150, 100	1200, 800	\$1.5/per mile
測試範例 2-3	150, 100	1200, 800	\$1.5/per mile
測試範例 2-4	150, 100	1200, 800	\$1.5/per mile
測試範例 3-1	150, 100, 100	1200, 800, 800	\$1.5/per mile
測試範例 3-2	150, 100, 100	1200, 800, 800	\$1.5/per mile
測試範例 3-3	150, 100, 100	1200, 800, 800	\$1.5/per mile
測試範例 3-4	150, 100, 100	1200, 800, 800	\$1.5/per mile
測試範例 4-1	11000, 10000	40000, 38000	\$1.5/per mile
測試範例 4-2	11000, 10000	40000, 38000	\$1.5/per mile
測試範例 4-3	11000, 10000	40000, 38000	\$1.5/per mile
測試範例 4-4	11000, 10000	40000, 38000	\$1.5/per mile
測試範例 5	4500, 4500, 4500 4500, 4000, 4000	3000, 3000, 3000, 3000, 2000, 2000	\$1.5/per mile
測試範例 6	230, 230, 220, 220	1100, 1100, 1000, 1000	\$1.5/per mile
測試範例 7	240, 240, 230, 230, 220, 220	1200, 1200, 1100, 1100, 1000, 1000,	\$1.5/per mile
測試範例 8	800, 700	4500, 4000	\$1.5/per mile

4.2 求解結果

為測試啟發式演算法之效率與準確度，本研究以 FORTRAN 程式語言編寫啟發式演算法，商業套裝軟體 LINGO 執行整數規劃模式，並針對 20 個範例來測試啟發式演算法在實務上應用之可行性。表 2 彙整啟發式演算法與部分整數規劃模式的執行結果，並分別從電腦演算時間、總成本兩方面來比較啟發式演算法與數學規

劃的效率與準確度，限於篇幅本研究省略啟發式演算法與整數規劃兩種不同方法求得之車輛配送路線。

從表 2，可觀察這 16 個範例以啟發式演算法求解的運算時間最長不超過 0.12 秒鐘，顯示其執行效率，而數學規劃的執行時間，則隨著問題的變大，大幅增加。這種現象是預料中的，因為問題本身有 NP-hard 的性質，一般以數學規劃求解車輛途程問題，顧客數目侷限於 25 位以內，所以數學規劃僅適用於小型的問題。關於總成本方面，由表 2，可觀察啟發式演算法與數學規劃所獲得的成本，其差距不大，最大值為 6.87%，最小值為 0%，所有範例平均誤差在 1.66% 以內，其中 3 個範例啟發式演算法獲得最佳解，表現令人滿意，由這 16 個範例，得知本研究所發展的啟發式演算法可獲得近似最佳解或最佳解。由表 2 已知啟發式演算法的執行時間十分快速，為進一步測試演算法對較大型問題之求解效率，表 3 彙總範例 5~8 範例啟發式演算法的執行時間，在這四範例中顧客數目分別為 32 位、50 位、75 位與 100 位。由表中顯示隨著問題變大，啟發式演算法之電腦執行時間大抵呈線性增加，求解速度十分快速，例如範例 8 中，100 位顧客的執行時間亦僅需 1.93 秒。

綜合以上的題庫的測試，本研究所發展的啟發式演算法，在所追求的準確度與效率兩目標的表現都十分良好。

表 2 範例 1-1~範例 4-4 的總成本與電腦執行時間

測試範例編號	數學規劃		啟發式演算法		誤差百分比(%)
	總成本	執行時間 (秒)	總成本	執行時間 (秒)	
測試範例 1-1	941.5	0.4	941.5	0.02	0.00%
測試範例 1-2	937	0.4	937	0.023	0.00%
測試範例 1-3	917.5	0.4	917.5	0.02	0.00%
測試範例 1-4	917.5	0.4	980.5	0.01	6.87%
測試範例 2-1	2337.5	215	2492	0.017	6.61%
測試範例 2-2	2372	206	2373.5	0.063	0.06%
測試範例 2-3	2259.5	211	2274.5	0.04	0.66%
測試範例 2-4	2252	214	2265.5	0.037	0.60%
測試範例 3-1	3149.5	879	3211	0.12	1.95%
測試範例 3-2	3173.5	904	3221.5	0.053	1.51%
測試範例 3-3	3361	912	3533.5	0.086	5.13%
測試範例 3-4	3232	896	3286	0.053	1.67%
測試範例 4-1	78762	1546	79033.5	0.076	0.34%
測試範例 4-2	79149	1621	79180.5	0.071	0.04%
測試範例 4-3	79747.5	1578	79767	0.073	0.02%
測試範例 4-4	79792.5	1634	80706	0.074	1.14%

表 3 範例 5 ~範例 8 的車輛配送路線、總成本與電腦執行時間

測試範例編號	車輛配送路線	總成本	執行時間
測試範例 5	路線一: 1- 8- 29- 28- 31- 6- 7- 30- 33- 1 路線二: 1- 4- 27- 1 路線三: 1- 26- 24- 22- 25- 23- 1 路線四: 1- 32- 2- 19- 17- 20- 16- 18- 1 路線五: 1- 15- 12- 3- 13- 14- 1 路線六: 1- 9- 10- 11- 21- 5- 1	18467.5	0.19
測試範例 6	路線一: 1- 39- 10- 50- 7- 44- 28- 49- 37- 36- 11- 20- 41- 42- 14- 5- 18- 1 路線二: 1- 23- 3- 2- 27- 32- 29- 30- 22-35- 48- 38- 46- 34- 40- 1 路線三: 1- 15- 19- 43- 45- 16- 47- 33- 13- 12- 17- 6- 1 路線四: 1- 51- 4- 9- 8- 24- 25- 26- 21- 31- 1	5769	0.31
測試範例 7	路線一: 1- 67- 76- 3- 64- 70- 62- 61- 73- 74- 75- 72- 68- 71- 69- 66- 65- 1 路線二: 1- 57- 59- 55- 2- 53- 52- 50- 54- 60- 49- 58- 51- 48- 1 路線三: 1- 39- 46- 40- 37- 38- 44- 36- 42- 56- 43- 45- 47- 41- 1 路線四: 1- 29- 32- 35- 28- 31- 26- 30- 63- 27- 33- 34- 1 路線五: 1- 15- 22- 17- 19- 24- 20- 14- 16- 21- 23- 18- 25- 1 路線六: 1-12- 10- 6- 8-13- 9- 7- 11- 4- 5- 1	10540.5	0.55
測試範例 8	路線一: 1- 50- 88- 94- 82- 96- 55- 76- 59- 80- 67- 62-101- 98- 64- 97- 84- 74- 99- 52- 57- 53- 70- 93- 79- 61- 89- 75- 54- 91- 56- 92- 71- 81- 68- 63- 86- 73- 60- 85-100- 78- 58- 65- 77- 51- 90- 72- 83- 87- 66- 95- 69- 1 路線二: 1- 40- 32- 15- 6- 10-24- 28- 29- 35- 31- 13- 46- 39- 22- 17- 12- 36- 19- 2- 38- 9- 26- 44- 3- 8- 37- 7- 20- 30- 47- 11- 25- 18- 42- 21- 14- 4- 43- 5- 41- 16- 34- 45- 27- 33- 49- 23- 48- 1	14146	1.93

伍、結論與建議

本研究以單一場站同時收送貨問題為基礎，考慮多車種，在顧客需求已知並允許收貨量大於送貨量之情境下，規劃出一個追求總成本最小的車輛途程安排。研究中建立一整數規劃模式與測試例題並發展一啓發式演算法以解決單一場站多車種同時收送貨之車輛途程問題。將整個研究結果與建議，歸納敘述如下：

- 一、現實生活中之專業物流公司配送中心，其車隊由多種不同噸位之車輛組合而成，顧客之需求亦已不僅侷限於送貨，實際配送環境已包含同時收送貨情境，本研究發展一啓發式演算法完整的分析一組運送者、車種的分配和路線選擇，進而發展出一運送計劃，有其實用性。
- 二、因收送貨車輛途程問題具有 NP-hard 的性質，由本研究發展的整數規劃求解此問題，雖然可獲得最佳解，但是相當耗費人力與時間，僅適用於小型問題。
- 三、本研究啓發式演算法與數學規劃所獲得的成本，其差距不大，最大值為 6.87%，最小值為 0%，所有範例平均誤差在 1.66% 以內，測試結果十分理想。
- 四、好的啓發式演算法除了需有精確性之外，也需具有求解的效率，從範例中，得知啓發式演算法，其執行時間，均十分快速，測試範例中具有 100 位顧客的問題，亦僅需要 1.93 秒即能快速獲解，求解效率相當優異。
- 五、未來研究方向可朝發展一整合以多場站多車種同時收送貨車輛途程問題之啓發式演算法，使其更貼近現今配送環境並更具實用性。

參考文獻

1. Christofides, N. and Eilon, S. "An Algorithm for the Vehicle Dispatching Problems," *Operational Research Quarterly*, 20: 309-318, 1969.
2. Christofides, N., "Exact Algorithms for the Vehicles Routing Problems, Based on Spanning Tree and Shortest Path Relaxations," *Math. Prog.*, Vol. 20: 255-282, 1981.
3. Christofides, N., Mingozzi, A., and Toth, P., "The Vehicle Routing Problem," *Combinatorial Optimization*, Wiley, Chichester, pp. 315-338, 1979.
4. Clarke, G. and Wright, J., "Scheduling of Vehicles from a Central Depot to a Number of Delivery Points," *Operational Research*, 12: 568-581, 1964.
5. Gaskell, T.J., "Basis for Vehicle Fleet Scheduling," *Operational Research Quarterly*, 18: 281-295, 1967.
6. Gendreau, M., Hertz, A., and Laporte, G., "A Tabu Search Heuristic for the Vehicle Routing Problem," *Management Science* 40: 1276-1290, 1994.
7. Kindervater, G.A.P., and Savelsbergh, M.W.P., "Vehicle Routing: Handling Edge Exchanges. In: Aarts, E.H., Lenstra, J.K. (Eds.)," *Local Search in Combinatorial Optimization*. Wiley, Chichester, 1997.
8. Laporte, G., Nobert, Y. and Desrochers, M., "Optimal Routing Under Capacity and Distance Restriction," *CACM*, 33: 1015-1073, 1985.

9. Laporte, G., Gendreau, M., Potvin, J-Y, and Semet, F., "Classical and Modern Heuristics for the Vehicle Routing Problem," *International Transactions in Operational Research*, 7: 285-300, 2000.
10. Lin, S., "Computer Solution of the Traveling Salesman Problem," *Bell System Technology Journal*, 44: 2245-2269, 1965.
11. Lin, S. and Kernighan, B., "An Effective Heuristic Algorithm for the Traveling Salesman Problem," *Operational Research*, 21: 498-516, 1973.
12. Min, H., "The Multiple Vehicle Routing Problem with Simultaneous Delivery and Pick-up Points," 23: 377-386, 1989.
13. Nagy, G. and Salhi, S., "Heuristic Algorithms for Single and Multiple and Depot Vehicle Routing Problem with Pickups and Deliveries," *European Journal of Operational Research*, 162: 126-141, 2005.
14. Osman, I.H., "Metastrategy Simulated Annealing and Tabu Search Algorithm for the Vehicle Routing problem," *Annals of Operations Research* 41: 421-451, 1993.
15. Psaraftis, H.N., "A Dynamic Programming Solution to the Single Vehicle Many-to-many Immediate Request Dial-a-ride Problem," *Transportation Science*, 14: 130-154, 1999.
16. Rego, C., Roucairol, C., "A Parallel Tabu Search Algorithm Using Ejection Chains for Vehicle Routing Problem," In: Osman, I.H., Kelly, J.P. (Eds.), *Meta-Heuristics: Theory and Applications*, Kluwer, Boston, 1996.
17. Rochat, Y., and Taillard, É.D., "Probabilistic Diversification and Intensification in Local Search for Vehicle Routing," *Journal of Heuristics* 1: 147-167, 1995.
18. Salhi, S., and Nagy, G., "A Cluster Insertion Heuristic for Single and Multiple Depot Vehicle Routing Problem with Backhauling," *Journal of the Operational Research Society*, 50: 1034-1042, 1999.
19. Schruben, L., and Clifton, R., "The Lockset Method of Sequential Programming Applied to Routing Delivery and Pickup Trucks," *American Journal of Agricultural Economics*, 50: 854-867, 1968.
20. Taillard, É.D., "Parallel Iterative Search Methods for Vehicle Routing Problem," *Networks* 23: 661-673, 1993.
21. Yellow, P., "A Computational Modification to the Savings Method of Vehicle Scheduling," *Operational Research Quarterly*, 21: 281-283, 1970.
22. Xu, J., and Kelly, J.P., "A Network Flow-based Tabu Search Heuristic for the Vehicle Routing Problem," *Transportation Science* 30: 379-393, 1996.

附 錄

測 試 範 例 1-1

節點	座標(X, Y)	送貨量	收貨量
1	30 40	0	0
2	21 47	30	9
3	17 63	16	25
4	31 62	9	11
5	52 33	21	7

測 試 範 例 1-2

節點	座標(X, Y)	送貨量	收貨量
1	30 40	0	0
2	21 47	19	6
3	17 63	23	10
4	31 62	11	14
5	52 33	5	6

測 試 範 例 1-3

節點	座標(X, Y)	送貨量	收貨量
1	40 30	0	0
2	37 52	9	2
3	49 49	21	9
4	52 64	15	19
5	20 26	19	22

測 試 範 例 1-4

節點	座標(X, Y)	送貨量	收貨量
1	40 30	0	0
2	37 52	20	6
3	49 49	23	10
4	52 64	12	15
5	20 26	5	6

測 試 範 例 2-1

節點	座標(X, Y)	送貨量	收貨量
1	30 40	0	0
2	37 52	7	2
3	49 49	30	13
4	52 64	16	20
5	20 26	9	10
6	40 30	21	5
7	21 47	15	15
8	17 63	19	1
9	31 62	23	10
10	52 33	11	0
11	51 21	5	5

測 試 範 例 2-2

節點	座標(X, Y)	送貨量	收貨量
1	30 40	0	0
2	37 52	30	9
3	49 49	21	9
4	52 64	19	24
5	20 26	15	18
6	40 30	16	4
7	21 47	29	29
8	17 63	26	2
9	31 62	37	16
10	52 33	16	4
11	51 21	12	13

測試範例 2-3

節點	座標(X, Y)		送貨量	收貨量
1	30	40	0	0
2	21	45	18	5
3	45	35	26	11
4	55	20	11	14
5	33	34	30	36
6	50	50	26	7
7	55	45	37	37
8	26	59	16	1
9	40	66	12	5
10	55	65	31	9
11	35	51	8	8

測試範例 2-4

節點	座標(X, Y)		送貨量	收貨量
1	30	40	0	0
2	21	45	18	5
3	45	35	11	4
4	55	20	21	27
5	33	34	19	22
6	50	50	15	4
7	55	45	16	16
8	26	59	29	2
9	40	66	26	11
10	55	65	16	3
11	35	51	12	13

測試範例 3-1

節點	座標(X, Y)		送貨量	收貨量
1	40	40	0	0
2	22	22	18	5
3	36	26	26	11
4	21	45	11	14
5	45	35	30	36
6	55	20	21	5
7	33	34	19	19
8	50	50	15	1
9	55	45	16	7
10	26	59	29	0
11	40	66	26	28
12	55	65	37	9
13	35	51	16	6
14	62	35	12	13
15	62	57	31	2
16	62	24	8	12

測試範例 3-2

節點	座標(X, Y)		送貨量	收貨量
1	40	40	0	0
2	22	22	26	8
3	36	26	11	4
4	21	45	30	38
5	45	35	21	25
6	55	20	19	5
7	33	34	15	15
8	50	50	16	1
9	55	45	29	12
10	26	59	26	0
11	40	66	37	40
12	55	65	16	4
13	35	51	12	4
14	62	35	31	34
15	62	57	29	1
16	62	24	10	16

測試範例 3-3

節點	座標(X, Y)		送貨量	收貨量
1	40	40	0	0
2	15	30	37	11
3	25	30	16	7
4	20	50	12	15
5	10	43	31	37
6	55	60	8	2
7	30	60	19	19
8	20	65	20	1
9	50	35	13	5
10	30	25	15	0
11	15	10	22	24
12	30	5	28	7
13	10	20	12	4
14	5	30	6	6
15	20	40	27	1
16	15	60	14	22

測試範例 3-4

節點	座標(X, Y)		送貨量	收貨量
1	40	40	0	0
2	15	30	16	5
3	25	30	29	12
4	20	50	26	33
5	10	43	37	44
6	55	60	16	4
7	30	60	12	12
8	20	65	31	2
9	50	35	8	3
10	30	25	19	0
11	15	10	20	22
12	30	5	13	3
13	10	20	15	5
14	5	30	22	24
15	20	40	28	1
16	15	60	12	19

測試範例 4-1

節點	座標(X, Y)		送貨量	收貨量	節點	座標(X, Y)		送貨量	收貨量	節點	座標(X, Y)		送貨量	收貨量
1	266	235	0	0	9	230	262	150	6	17	318	252	100	4
2	295	272	125	3	10	249	268	110	20	18	329	224	250	40
3	301	258	84	3	11	256	267	410	451	19	267	213	120	5
4	309	260	60	7	12	265	257	225	5	20	275	192	600	60
5	217	274	500	60	13	267	242	300	11	21	303	201	500	64
6	218	278	300	8	14	259	265	250	27	22	208	217	175	1
7	282	267	175	17	15	315	233	500	3	23	326	181	75	13
8	242	249	350	2	16	329	252	150	24					

測試範例 4-2

節點	座標 (X, Y)		送貨量	收貨量	節點	座標 (X, Y)		送貨量	收貨量	節點	座標 (X, Y)		送貨量	收貨量
	X	Y				X	Y				X	Y		
1	266	235	0	0	9	230	262	150	66	17	318	252	750	373
2	295	272	400	128	10	249	268	450	20	18	329	224	140	224
3	301	258	120	527	11	256	267	700	770	19	267	213	100	443
4	309	260	40	51	12	265	257	550	140	20	275	192	500	500
5	217	274	80	96	13	267	242	650	251	21	303	201	250	324
6	218	278	200	553	14	259	265	300	330	22	208	217	170	161
7	282	267	750	750	15	315	233	130	84	23	326	181	110	208
8	242	249	150	116	16	329	252	700	112					

測試範例 4-3

節點	座標 (X, Y)		送貨量	收貨量	節點	座標 (X, Y)		送貨量	收貨量	節點	座標 (X, Y)		送貨量	收貨量
	X	Y				X	Y				X	Y		
1	266	235	0	0	9	322	437	150	664	17	296	418	130	647
2	292	495	700	224	10	323	433	150	20	18	261	384	700	112
3	298	427	400	175	11	324	433	250	275	19	297	410	750	332
4	309	445	400	519	12	323	429	160	409	20	315	407	140	140
5	307	464	200	240	13	314	435	450	174	21	314	406	500	124
6	336	475	900	249	14	311	442	700	770	22	321	391	170	161
7	320	439	600	600	15	304	427	550	35	23	326	181	110	208
8	321	437	750	58	16	293	421	650	104					

測試範例 4-4

節點	座標 (X, Y)		送貨量	收貨量	節點	座標 (X, Y)		送貨量	收貨量	節點	座標 (X, Y)		送貨量	收貨量
1	266	235	0	0	9	322	437	150	664	17	296	418	130	647
2	292	495	120	385	10	323	433	150	20	18	261	384	700	112
3	298	427	40	17	11	324	433	250	275	19	297	410	750	332
4	309	445	80	103	12	323	429	160	409	20	315	407	140	140
95	307	464	200	240	13	314	435	450	174	21	314	406	300	419
6	336	475	900	249	14	311	442	700	770	22	321	391	600	56
7	320	439	600	600	15	304	427	400	26	23	326	181	100	189
8	321	437	750	58	16	293	421	300	480					

測試範例 5

節點	座標(X, Y)		送貨量	收貨量	節點	座標(X, Y)		送貨量	收貨量
1	292	495	0	0	18	297	410	550	880
2	298	427	700	224	19	315	407	650	288
3	309	445	400	175	20	314	406	200	200
4	307	464	400	519	21	321	391	400	519
5	336	475	1200	1440	22	321	398	300	28
6	320	439	40	11	23	314	394	1300	2469
7	321	437	80	80	24	313	378	700	334
8	322	437	2000	155	25	304	382	750	1125
9	323	433	900	398	26	295	402	1400	412
10	324	433	600	300	27	283	406	4000	6000
11	323	429	750	825	28	279	399	600	48
12	314	435	1500	383	29	271	401	700	113
13	311	442	150	58	30	264	414	400	400
14	304	427	250	275	31	277	439	400	167
15	293	421	1600	104	32	290	434	1200	298
16	296	418	450	720	33	319	433	40	55
17	261	384	700	348					

測試範例 6

節點	座標(X, Y)		送貨量	收貨量	節點	座標(X, Y)		送貨量	收貨量
1	30	40	0	0	27	27	68	7	10
2	37	52	7	2	28	30	48	15	1
3	49	49	30	13	29	43	67	14	2
4	52	64	16	20	30	58	48	6	6
5	20	26	9	10	31	58	27	19	7
6	40	30	21	5	32	37	69	11	2
7	21	47	15	15	33	38	46	12	16
8	17	63	19	1	34	46	10	23	34
9	31	62	23	10	35	61	33	26	3
10	52	33	11	10	36	62	63	17	4
11	51	21	5	5	37	63	69	6	10
12	42	41	19	4	38	32	22	9	2
13	31	32	29	11	39	45	35	15	5
14	5	25	23	25	40	59	15	14	3
15	12	42	21	1	41	5	6	7	10
16	36	16	10	16	42	10	17	27	35
17	52	41	15	7	43	21	10	13	20
18	27	23	3	4	44	5	64	11	10
19	17	33	41	18	45	30	15	16	22
20	13	13	9	9	46	39	10	10	3
21	57	58	28	36	47	32	39	5	6
22	62	42	8	10	48	25	32	25	44
23	42	57	8	15	49	25	55	17	22
24	16	57	16	7	50	48	28	18	1
25	8	52	10	15	51	56	37	10	17
26	7	38	28	8					

測 試 範 例 7

節 點	座 標 (X, Y)		送 貨	收 貨	節 點	座 標 (X, Y)		送 貨	收 貨	節 點	座 標 (X, Y)		送 貨	收 貨
1	40	40	0	0	27	41	46	18	27	53	54	38	19	8
2	22	22	18	5	28	55	34	17	1	54	55	57	22	24
3	36	26	26	11	29	35	16	29	4	55	67	41	16	2
4	21	45	11	14	30	52	26	13	13	56	10	70	7	11
5	45	35	30	36	31	43	26	22	9	57	6	25	26	5
6	55	20	21	5	32	31	76	25	6	58	65	27	14	26
7	33	34	19	19	33	22	53	28	39	59	40	60	21	6
8	50	50	15	1	34	26	29	27	40	60	70	64	24	28
9	55	45	16	7	35	50	40	19	2	61	64	4	13	10
10	26	59	29	10	36	55	50	10	2	62	36	6	15	10
11	40	66	26	28	37	54	10	12	1	63	30	20	18	21
12	55	65	37	9	38	60	15	14	4	64	20	30	11	5
13	35	51	16	6	39	47	66	24	8	65	15	5	28	53
14	62	35	12	13	40	30	60	16	4	66	50	70	9	16
15	62	57	31	2	41	30	50	33	49	67	57	72	37	18
16	62	24	8	12	42	12	17	15	19	68	45	42	30	33
17	21	36	19	9	43	15	14	11	17	69	38	33	8	16
18	33	44	20	32	44	16	19	18	10	70	50	4	11	3
19	9	56	13	5	45	21	48	17	23	71	66	8	3	16
20	62	48	15	15	46	50	30	21	7	72	59	5	1	5
21	66	14	22	28	47	51	42	27	37	73	35	60	6	1
22	44	13	28	2	48	50	15	19	34	74	27	24	10	7
23	26	13	12	22	49	48	21	20	25	75	40	20	20	11
24	11	28	6	2	50	12	38	5	0	76	40	37	8	2
25	7	43	27	40	51	15	56	22	37					
26	17	64	14	4	52	29	39	12	5					

測試範例 8

節點	座標 (X, Y)		送貨	收貨	節點	座標 (X, Y)		送貨	收貨	節點	座標 (X, Y)		送貨	收貨
1	35	35	0	0	35	65	55	14	1	69	56	39	36	57
2	41	49	10	3	36	63	65	8	1	70	37	47	6	2
3	35	17	7	3	37	2	60	5	3	71	37	56	5	7
4	55	45	13	16	38	20	20	8	2	72	57	68	15	25
5	55	20	19	22	39	5	5	16	5	73	47	16	25	30
6	15	30	26	7	40	60	12	31	7	74	44	17	9	1
7	25	30	3	3	41	40	25	9	13	75	46	13	8	8
8	20	50	5	2	42	42	7	5	6	76	49	11	18	2
9	10	43	9	3	43	24	12	5	8	77	49	42	13	22
10	55	60	16	0	44	23	3	7	2	78	53	43	14	19
11	30	60	16	17	45	11	14	18	25	79	61	52	3	2
12	20	65	12	3	46	6	38	16	5	80	57	48	23	10
13	50	35	19	7	47	2	48	1	1	81	56	37	6	8
14	30	25	23	25	48	8	56	27	48	82	55	54	26	8
15	15	10	20	1	49	13	52	36	46	83	15	47	16	28
16	30	5	8	12	50	6	68	30	3	84	14	37	11	2
17	10	20	19	9	51	47	47	13	22	85	11	31	7	12
18	5	30	2	3	52	49	58	10	4	86	16	22	41	45
19	20	40	12	5	53	27	43	9	4	87	4	18	35	48
20	15	60	17	17	54	37	31	14	15	88	28	18	26	7
21	45	65	9	11	55	57	29	18	2	89	26	52	9	9
22	45	20	11	1	56	63	23	2	3	90	26	35	15	24
23	45	10	18	34	57	53	12	6	1	91	31	67	3	4
24	55	5	29	13	58	32	12	7	13	92	15	19	1	2
25	65	35	3	4	59	36	26	18	5	93	22	22	2	1
26	65	20	6	1	60	21	24	28	33	94	18	24	22	3
27	45	30	17	25	61	17	34	3	2	95	26	27	27	43

測 試 範 例 8(續)

節 點	座 標 (X, Y)		送 貨 量	收 貨 量	節 點	座 標 (X, Y)		送 貨 量	收 貨 量	節 點	座 標 (X, Y)		送 貨 量	收 貨 量
28	35	40	16	1	62	12	24	13	2	96	25	24	20	2
29	41	37	16	2	63	24	58	19	22	97	22	27	11	1
30	64	42	9	9	64	27	69	10	0	98	25	21	12	2
31	40	60	21	8	65	15	77	9	17	99	19	21	10	3
32	31	52	27	6	66	62	77	20	35	100	20	26	9	14
33	35	69	23	32	67	49	73	25	12	101	18	18	17	6
34	53	52	11	16	68	67	5	25	27					