Traveling Salesperson Problem (TSP)

$$\text{Minimize } Z = \sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij} x_{ij}$$

st

$$\sum_{j=1}^{n} x_{ij} = 1 \quad i = 1, 2, \dots, n$$

$$\sum_{i=1}^{n} x_{ij} = 1 \quad j = 1, 2, \dots, n$$

$$x_{ij} = (0, 1)$$

$$\sum_{ij \in S} x_{ij} \leq |S| - 1$$

$$x_{ij} = \begin{cases} 1, & \text{if city } j \text{ is reached from city } i. \\ 0, & \text{otherwise} \end{cases}$$

distance matrix

	1	2	3	4	5
1		2	3	4	5
2	2		6	7	8
3	3	6		9	10
4	4	7	9		11
5	5	8	10	11	

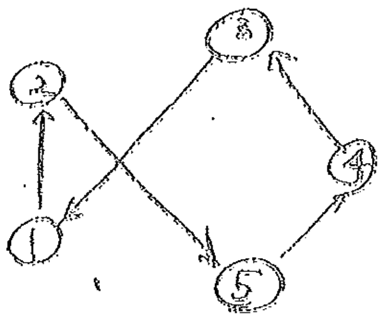
$$\text{Minimize } Z = 2x_{12} + 3x_{13} + 4x_{14} + 5x_{15}$$
$$+ 2x_{21} + 6x_{23} + 7x_{24} + 8x_{25}$$
$$+ 3x_{31} + 6x_{32} + 9x_{34} + 10x_{35}$$
$$+ 4x_{41} + 7x_{42} + 9x_{43} + 11x_{45}$$
$$+ 5x_{51} + 8x_{52} + 10x_{53} + 11x_{54}$$

subject to

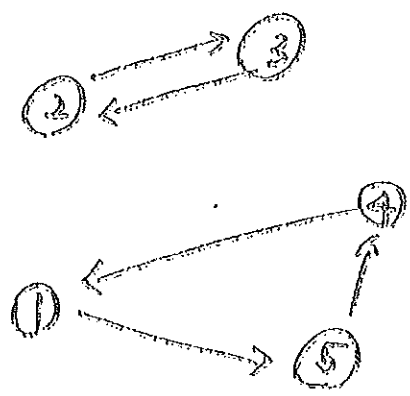
$$x_{12} + x_{13} + x_{14} + x_{15} = 1$$
$$x_{21} + x_{23} + x_{24} + x_{25} = 1$$
$$x_{31} + x_{32} + x_{34} + x_{35} = 1$$
$$x_{41} + x_{42} + x_{43} + x_{45} = 1$$
$$x_{51} + x_{52} + x_{53} + x_{54} = 1$$

$$x_{21} + x_{31} + x_{41} + x_{51} = 1$$
$$x_{12} + x_{32} + x_{42} + x_{52} = 1$$
$$x_{13} + x_{23} + x_{43} + x_{53} = 1$$
$$x_{14} + x_{24} + x_{34} + x_{54} = 1$$
$$x_{15} + x_{25} + x_{35} + x_{45} = 1$$

$$\forall x_{ij} \in (0, 1).$$



Tour solution



Subtour solution

$X_{12} = X_{25} = X_{54} = X_{43} = X_{31} = 1$

$X_{23} = X_{32} = X_{15} = X_{54} = X_{41} = 1.$

subtour breaking

$$X_{23} + X_{32} \le 1$$

$$X_{15} + X_{51} + X_{45} + X_{54} + X_{41} + X_{14} \le 2.$$

The Optional Stop TSP

$$y_j = \begin{cases} 1 & \text{if city } j \text{ is visited} \\ 0, & \text{otherwise} \end{cases}$$

V_j : the value of visiting city j

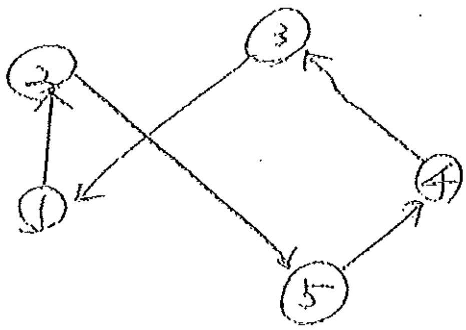
$$\text{Min} \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij} - \sum_{j=1}^{n} V_j y_j$$

$$\sum_{i \neq j} x_{ij} = y_j$$

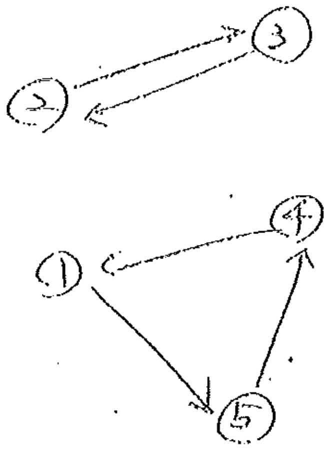
$$\sum_{k \neq j} x_{jk} = y_j$$

$$\sum_{i,j \in S} x_{ij} \le |S| - 1$$

$X_{12} = X_{25} = X_{54} = X_{43} = X_{31} = 1$



Tour Solution



Subtour solution

$X_{23} = X_{32} = X_{15} = X_{54} = X_{41} = 1$

Subtour breaking

$X_{23} + X_{32} \leq 1$

$X_{15} + X_{51} + X_{45} + X_{14} + X_{41} + X_{54} \leq 2$

☆ 会随问题加大变得越困难，呈出指数的变化，求解之速度快慢，会因人而异（功力）。所以当问题变大时，i.e. 50 城市以上，一般皆以启发式演算法求解。

	1	2	3	4	5
1	0	132	217	164	58
2	132	0	290	201	79
3	217	290	0	113	303
4	164	201	113	0	196
5	58	79	303	196	0

the nearest-neighbor heuristic (NNH) for TSP

可從任一節點開始，在未被拜訪的城市中，選擇城市為目前城市距離最近的。

1－5－2－4－3－1 optimal（距離 668）

3－4－1－5－2－3 距離 704 非最佳

一常見受歡迎的 NNH 為計算所有節點開始之距離，再從中找出最佳解。

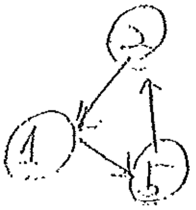
the cheapest insertion (CIH) for TSP

從任一節點開始，擇一與目前節點最近的節點。(i, j) 將 (i, j) 替換成 (i, k) 與 (k, j) 其中 k 為不在 (i, j) 中，k 使增加之距離最短（計算方式 $C_{ik}+C_{kj}-C_{ij}$）。一開始 (1, 5) 和 (5, 1)

	k=2 或 3 或 4		
(1,5)	(1,2)-(2,5)	$C_{12}+C_{25}-C_{15}=153$ *	任選一
(1,5)	(1,3)-(3,5)	$C_{13}+C_{35}-C_{15}=462$	
(1,5)	(1,4)-(4,5)	$C_{14}+C_{45}-C_{15}=302$	
(5,1)	(5,2)-(2,1)	$C_{52}+C_{21}-C_{51}=153$ *	
(5,1)	(5,3)-(3,1)	$C_{53}+C_{31}-C_{51}=462$	
(5,1)	(5,4)-(4,1)	$C_{54}+C_{41}-C_{51}=302$	

任何 X_{ij} 包含 subtour 將會違反上述限制式

例如 $X_{15}=X_{52}=X_{21}=X_{43}=X_{34}=1$ 包含兩個 subtour



選擇不包含城市1之 subtour

$$u_3 - u_4 + 5X_{34} \le 4$$
$$u_4 - u_3 + 5X_{43} \le 4$$

加總兩式產生 $5X_{34} + 5X_{43} \le 8$

⇒ 不可能產生 $X_{34} = X_{43} = 1$

1. Any set of X_{ij}'s containing a subtour will be infeasible (that is, they violate the constraint)

2. Any set of X_{ij}'s that forms a tour will be feasible (there will exist a set of u_j's that satisfy the constraint)

令 t_i：為城市 i 為路徑中的位置 (拜訪順序)

令 $u_i = t_i$ 將會滿足限制式

例如 1 − 3 − 4 − 5 − 2 − 1

$u_1=1$ $u_3=2$ $u_4=3$ $u_5=4$ $u_2=5$

城市位置

例如考慮 $x_{ij}=1$，例如 x_{52}

對應 $u_5-u_2+5x_{52}\le 4$

城市2緊接在城市5之後，則 $u_5-u_2=-1$

$-1+5x_{52}\le 4$　$5x_{52}\le 5$ 成立。

考慮 $x_{ij}=0$，例如 $x_{32}=0$，

$$u_3-u_2+5x_{32}\le 4 \quad \because x_{32}=0$$

$$u_3-u_2\le 4 \quad \because u_3\le 5，且 u_2\ge 2$$

$5-2=3$，∴ u_3-u_2 不可能超過3，

亦滿足上述限制式

$$u_i-u_j+p\,x_{ij}\le p-1 \quad \text{for } 2\le i\ne j\le n \quad O(n^2)$$

subtour elimination constraints (subtour prevention constraint

新成 (1, 2)(2, 5)-(5, 1) k=3 成本

(1, 2)	(1, 3)-(3, 2)	$C_{13}+C_{32}-C_{12} = 375$
(1, 2)	(1, 4)-(4, 2)	$C_{14}+C_{42}-C_{12} = 233$ *
(2, 5)	(2, 3)-(3, 5)	$C_{23}+C_{35}-C_{25} = 514$
(2, 5)	(2, 4)-(4, 5)	$C_{24}+C_{45}-C_{25} = 318$
(5, 1)	(5, 3)-(3, 1)	$C_{53}+C_{31}-C_{51} = 462$
(5, 1)	(5, 4)-(4, 1)	$C_{54}+C_{41}-C_{51} = 302$

(1, 4)-(4, 2)-(2, 5), (5, 1) k=3

(1, 4)	(1, 3)-(3, 4)	$C_{13}+C_{34}-C_{14} = 166$ *
(4, 2)	(4, 3)-(3, 2)	$C_{43}+C_{32}-C_{42} = 202$
(2, 5)	(2, 3)-(3, 5)	$C_{23}+C_{35}-C_{25} = 514$
(5, 1)	(5, 3)-(3, 1)	$C_{53}+C_{31}-C_{51} = 462$

(1, 3)(3, 4)-(4, 2)-(2, 5)-(5, 1)

optimal

The Traveling Salesman Problem

A Guided Tour of Combinatorial Optimization

Edit by

E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan.

D. B. Shmoys. John Wiley & Sons Ltd 1985

Evaluation of Heuristics

The following three methods have been suggested for evaluating heuristics:

1. Performance guarantees
2. Probabilistic analysis
3. Empirical analysis

1. gives a worst-case bound on how far away from optimality a tour constructed by the heuristic can be. (complexity). NNH 很差.

2. a heuristic is evaluated by assuming that the location of cities follows some known probability distribution.

$$\frac{\text{Expected length of the path found by the heuristic}}{\text{Expected length of an optimal tour}}$$

越接近 1，越好

例如城市是独立且均匀，uniformly distributed on a cube of unit length, width and height.

3. Empirical analysis: heuristics are compared to the optimal solution for a number of problems for which the optimal tour is known.

A Comparison of the growth of some polynomial functions to that of certain exponential functions

Function	Approximate values		
polynomial function n	10	100	1,000
$n \log n$	33	664	9966
n^3	1000	1,000,000	10^9
$10^6 n^8$	10^{4}	10^{22}	10^{33}
2^n	1024	1.27×10^{30}	1.05×10^{301}
$n^{\log n}$	2099	1.93×10^{13}	$2.89 \times 10^{[illegible]}$
$n!$	3628800	10^{158}	4×10^{2567}

call an algorithm "good" when it is sufficiently efficient to be useable in practice, (if its worst-case complexity is bounded by a polynomial function of n.

O (big-O notation) to express the running time function

$O(n \log n)$ $O(n^3)$ 等.

整合以自有車隊運送或選擇貨運公司服務與車輛運送路線最佳解之研究

朱經武

國立台灣海洋大學航運管理系

摘要

運銷管理者所面臨的決策問題很多，如何選擇一合適的業者運送貨物對運銷管理而言，是一很重要的決策，因為運輸成本在運銷各項成本中所佔之比例最高。基本上有兩種運輸方式可供業者選擇，一種是以自有車隊運送，另者是以貨運公司運送。選擇貨運公司運送，只需將貨物托運即可。以自有車隊運送允許以同一車合併(consolidate)許多不同的運送（shipment），去不同的目的地，所以運輸成本通常是行駛距離的函數。為了確保運送能滿足公司的成本與服務之目標，運銷管理者需要決定的是合併（consolidate）那些運送在同一車和規劃運送路線。

本研究構建一最佳解的數學模式，並編寫程式以便產生數學模式求解，協助運銷管理者解決問題。

前言

航運管理涵蓋範圍甚廣，舉凡與海上運送、航空運輸、陸上運銷（Physical Distribution）相關的問題，均為航運管理探討之範疇。

運銷為航運管理重要的一環，運銷管理者所面臨的決策問題很多，如何選擇一合適的運輸方式或業者運送貨物對運銷管理而言，是一很重要的決策，因為運輸成本在運銷各項成本中所佔之比例最高。

基本上有兩種運輸方式可供業者選擇，一種是以自有車隊運送，另者是以貨運公司運送。以自有車隊運送允許以同一車合併許多不同的運送，到達不同的目的地，所以運輸成本通常是行駛距離的函數。為了確保運送能滿足公司的成本與服務之目標，運銷管理者需要決定的是合併那些運送在同一車和規劃運送路線。而車隊的司機者只是去執行運銷管理者的運送計劃。

貨運公司通常個別的處理每一公司的運送，其運費通常是貨物種類、目的地、大小的函數。貨運公司，負責規劃每次從運送地至目的地的路線，運銷管理者只需將貨物交與給他們即可。市場上常見的新竹貨運、與大榮貨運屬於上述情形。

在上述以自有車隊運送或以貨運公司運送的情境下，運銷管理需要回答下列三項問題：

1. 運送是要以自有車隊運送、委託貨運公司運送或同時採用二者。
2. 如果是以自有車隊運送，則需規劃如何將不同的運送合併在同一車，使得總運輸成本最低。
3. 如果是同時採用二者，則需規劃那些運送將由貨運公司處理與如何將其餘不同的運送合併在同一車，使得總運輸成本最低。

上述的第二項問題，是一傳統的車輛運送路線問題（Vehicle routing problem），要解決一車輛運送路線問題，需要許多的計算資源。第三項問題為車輛運送路線問題再加上選擇貨運公司將使問題更加複雜。

本研究主要目的為構建一可獲得最佳解的數學模式，協助運銷管理者解決問題。

相關文獻

探討本研究這類型問題的國外文獻僅有兩篇，而其子問題-車輛運送路線問題（Vehicle Routing Problem）則有許多相關文獻。

R. Ballou and A. Chowdhury(1980),結合一個別演算法（a separate algorithm）和一路線安排節省演算法（a vehicle routing savings algorithm）。個別演算法需要（enumerate）列舉所有可能的運送，而路線安排節省演算法只是一個模組用來計算節省成本。

John Pooley(1992)基於 Clarke and Wright 的演算法，加入成本限制（例如，行車時間）及邏輯限制。邏輯限制包括作業及環境的障礙，例如車輛目的地停靠次數，顧客裝載貨物不希望與其競爭者的貨物同車，每車的裝載量等，提出其修改的演算法。

本研究子問題車輛運送路線問題（Vehicle Routing Problem）的解決方法依據 Laporte 大約可分成兩大類（一）最佳解（exact algorithms）演算法（二）啟發式演算法（heuristic algorithms）。

(一）最佳解演算法基本尚可分成三大類:

1.直接樹搜尋法（direct tree search method）
2.動態規劃（dynamic programming）
3.整數規劃（integer linear programming）

在 Laporte , G., and Nobert, Y, "Exact algorithms for the vehicle routing problem," Surveys in Combinatorial Optimization, North-Holland, Amsterdam (1987):147-184. 中有詳細的介紹。

（二）啟發式演算法

啟發式演算法有許多種，僅敘述三種專為解決車輛運送路線問題而設計之演算法。

1. The Clarke and Wright Algorithm

Clarke and Wright 最早提出解決 VRP 的啟發式方法，這方法的（complexity）複雜度是 0（n2 log n）。使用適當的資料結構可以降低複雜性（Nelson et al.1985 和 Paessens,1988）

2. The Sweep Algorithm

Gillett 和 Miller 提出 the sweep algorithm，他們利用（θi, ρi）代表運送節點 i，其中θi 為第 i 節點在座標上的角度，ρi 第 i 節點的長度，其方法對任意節點 i 指定θi=0 ，計算節點 1 至 i 的θ，並將θ由小至大排列，在未被指派的節點中，選取最小的θ並且分配裝載，再以 Traveling Salesman Problem 銷售員問題求解相對應的路線。

3.Tabu Search algorithm

Gendreau, Hert and Laporte （1991）提出 tabu Search 禁制搜尋法，tabu Search 構建一連串的解，然後執行改進步驟，由演算法連續產生的路線未必可行，詳細之步驟參考 Laporte。Tabu Search 已經成功的被應用到一些作業研究文獻所敘述車輛運送路線問題，計算的及結果被證實是目前發展出來最好的演算法之一。

國內研究並沒有與本研究相同之文獻，而其子問題車輛運送路線問題之研究亦不多見，近三年內主要文獻如下：

韓復華、卓裕仁（1996）採用新近發展的門檻接受法，噪音擾動法與搜尋空間平滑法等，啟發式演算法並結合傳統交換型演算法，設計多種解題方法求解車輛運送路線問題。

韓復華等人（1997）應用門檻接受法求解車輛運送路線問題，並以國際網路題庫中選取 11 題標竿測試例題，進行測試，總平均誤差為 1.71%，測試例題中規模最大的 200 點題目，更優於文獻報導的已知最佳結果。

陳春益、林正章、高玉明（1997）以國內某大路線貨運公司貨物排程問題為例進行實例研究，利用 Dantizg-Wolfe 分解演算法，求解屬多元商品網路流量問題之貨物排程模式，並將營運網路予以電腦化，據以構建貨物排程模式。

李宗儒、翁基華（1998）以整數規劃構建解決車輛運送路線問題，其中並將員

工負荷平衡的考量因素，納入車輛運送路線規劃模式中。

李宗儒、翁基華（1997）以整數規劃構建解決車輛運送路線問題，並將其應用於農產運銷。

模式構建

為簡化問題與研究，遂有下列之假設：

1. 僅有一(Depot)倉庫，且自有車隊車輛運輸起始點與終點皆在倉庫。
2. 每一位顧客的訂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. 只考慮純送貨的情況。
5. 自有車隊的車輛運輸成本，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：

$$\min z = \sum_{k}^{m} FC_k + \sum_{i}^{n}\sum_{j}^{n}\sum_{k}^{m} C_{ijk}X_{ijk} + \sum_{i}^{n} CL_i L_i$$

subject to

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

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

$$\sum_{i}^{n} q_i Y_{ik} \le Q_k \quad (\quad k = 1,\cdots,m) \quad (3)$$

$$\sum_{j}^{n} X_{ijk} = Y_{ik} \quad (i = 1,\cdots,n\ ;\ k = 1,\cdots,m) \quad (4)$$

$$\sum_{j}^{n} X_{jik} = Y_{ik} \quad (i = 1,\cdots,n\ ;\ k = 1,\cdots,m) \quad (5)$$

$$\sum_{i\ j\in s} X_{ijk} \le |S| - 1 \quad \text{for all } S \subseteq \{2,\cdots,n\}$$

$$(k = 1,\cdots,m) \quad (6)$$

以下就模式的變數、符號與意義做一

$$X_{ijk} \in \{0,1\};\quad Y_{ik} \in \{0,1\};\quad L_i \in \{0,1\}$$

$$(i = 1,\cdots,n\ ;\ j = 1,\cdots,n;\ k = 1,\cdots,m)$$

說明。

i:顧客 $i = 1,\ldots,n$ (其中 1 代表倉庫)

j:顧客 $j = 1,\ldots,n$ (其中 1 代表倉庫)

k:車輛 $k = 1,\ldots,m$

$$X_{ijk} = \begin{cases} 1 & \text{自有 車輛k在到達顧客i 處之後，即至顧客j} \\ 0 & \text{其他} \end{cases}$$

$$L_i = \begin{cases} 1 & \text{顧客i 的需求 ，由貨運公司運送} \\ 0 & \text{其他} \end{cases}$$

$$Y_{ik} = \begin{cases} 1 & \text{顧客i 的需求 ，由自有 車輛k運送} \\ 0 & \text{其他} \end{cases}$$

FC_k:自有車輛 k 的固定成本

C_{ijk}:自有車輛 k 從顧客 i 到顧客 j 之間的成本

CL_i:運送公司服務顧客 i 的成本

q_i: 顧客 i 的需求

Q_i: 自有車輛 i 的裝載量

S:預防或破解內圍路線之限制式集合

目標函數的目的為整合以自有車隊運送與選擇貨運公司服務下，求取總成本最低。

限制式(1)確保派給倉庫車輛數目等於所有自有車輛數目。

限制式(2)確保每一位顧客只能派給一輛車。

限制式(3)確保每一輛車所裝載的貨物不超過每一輛車的裝載量。

限制式(4)與限制式(5)確保每一輛車在拜訪過一位顧客後，同時亦會從一位顧客處離開。

限制式(6)為防止內圍路線。

範例分析

假設倉庫有二輛車，車輛的基本資料，每一顧客的需求，顧客與顧客間之距離與貨運公司的運送成本如下:

表 1. 車輛之裝載量與成本資料

倉庫中心車輛	車輛裝載量(單位)	固定成本($)	變動成本($/公里)
1	4	300	1.5
2	4	300	1.5

表 2. 顧客與顧客間之距離　　　單位:公里

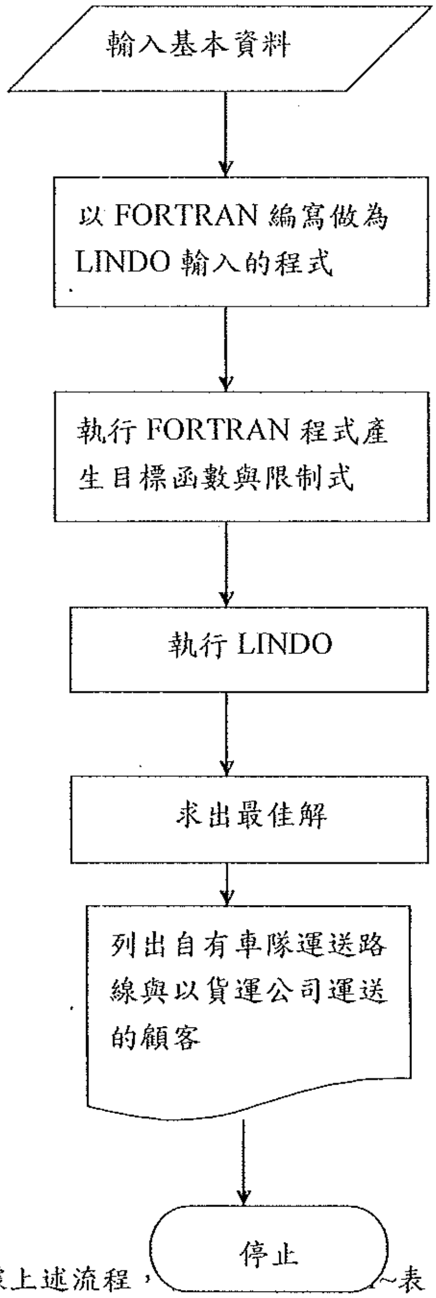
顧客	1	2	3	4	5	6	7
1	0	28	21	14	17	18	22
2	28	0	47	36	25	20	35
3	21	47	0	26	37	30	20
4	14	36	26	0	15	31	34
5	17	25	37	15	.0	29	39
6	18	20	30	31	29	0	16
7	22	35	20	34	39	16	0

表 3. 顧客需求量與貨運公司運送之成本

顧客編號	需求量	以貨運公司運送之運送成本
2	2	168
3	3	126
4	1	84
5	1	102
6	2	108
7	1	132

欲解 0-1 整數規劃，專業的套裝軟體有 OSL，CPLEX 與 LINDO 等。本研究採用 LINDO。雖然 LINDO 程式的輸入十分容易，但是本研究所探討這類型的問題，其變數與限制式的數目，往往超過數百或上千個，一一鍵入將十分浪費時間與人力且容易出錯。

為解決此一問題，以 FORTRAN 語言編寫一程式，程式會自動讀入基本資料(顧客與顧客間之距離、自有車輛之裝載量與成本資料、顧客需求量與以貨運公司運送之成本)，程式即可產生符合模式結構的目標函數與限制式的檔案，再以 LINDO 處理這個檔案，即可獲得最佳解，其流程如下:

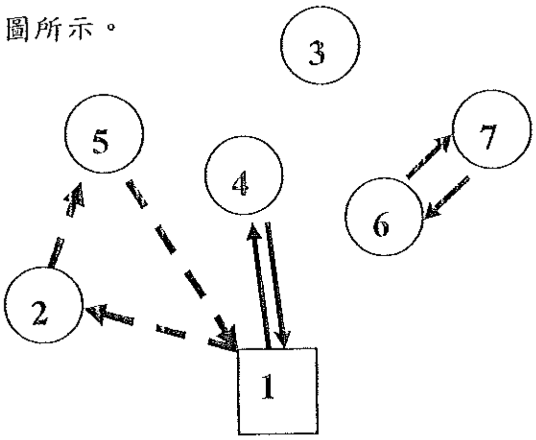


依據上述流程，~表 3 的基本資料並且儲存於不同檔案，然後執

行 FORTRAN 程式則自動產生如附錄的檔案以便供 LINDO 使用。最後執行 LINDO 即可求解，其結果摘要如下：

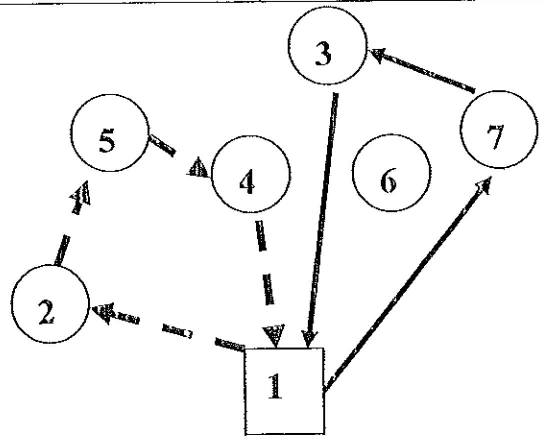
目標函數值 279	
X141=1	Y11=1
X411=1	Y12=1
X671=1	Y22=1
X761=1	Y41=1
X122=1	Y52=1
X252=1	Y61=1
X512=1	Y71=1
L3=1	其餘變數為 0

上述結果中的目標函數值 279，並未包括所有自有車輛的固定成本(300×2)，因為求解時一般不將常數納入，只需在求解後加回即可，所以全部的運送成本應為 879。可惜的是上述結果產生兩組內圍路線如下圖所示。



其中實線代表第一車；虛線代表第二車，很明顯第一車產生兩組內圍路線，為了破解內圍路線於是將新的限制式 X161+X171+X461+X471≥1 加入於附錄的檔案再度求解，其結果與對應的路線，摘要如下：

目標函數值 283.5	
X171=1	Y11=1
X731=1	Y12=1
X311=1	Y71=1
X122=1	Y31=1
X252=1	Y22=1
X542=1	Y52=1
X412=1	Y42=1
L6=1	其餘變數為 0



很明顯所有內圍路線均已被破解，目前的解即為最佳解，有最低的運送成本 $883.5。

結論

本研究構建一最佳解的數學模式並且編寫程式，協助運銷管理者解決問題。

未來研究方向，可以本研究的數學模式為基礎，逐步擴充，使其更臻於完善，其項目如下：

1. 本研究求解之執行時間，隨著顧客數目增加會有顯著的增加，發展啟發式演算法，將可減少求解時間，有其實用性。
2. 本研究考慮純送貨情況，如有需求可加入時窗(Time windows)的限制式，符合實際需求。
3. 結合地理資訊系統，提供道路資訊可節省更多的人力與時間。

參考文獻

1. R. Ballou and A. Chowdhury, "MSVS: A Generalized Vehicle Routing Algorithm," The Logistics and Transportation Review 16 (1980): 72-85.
2. G. Clarke and J.W. Wright, "Scheduling of vehicles from a central depot to a number of delivery points," Operations Research 12 (1964): 568-581.
3. M. Gendreau, A. Hertz, and G. Laporte (1991), "A tabu search heuristic for the vehicle routing problem", Publication # 777, Centre de recherche sur les transports, Montre'al.
4. B. Gillett and L. Miller,"A heuristic algorithm for the vehicle dispatch problem, "Operations Research 22 (1974): 340-349.
5. J. N. Hooker (1994), "Needed: An Empirical Science of Algorithms," Operations Research, vol.42, No. 2.: 201-212.
6. G. Laporte, and Y. Nobert, "Exact algorithms for the vehicle routing problem," Surveys in Combinatorial Optimization, North-Holland, Amsterdam (1987):147-184.
7. D. Nelson, K.E. Nygard, J.H.Griffin, and W.E. Shreve, "Implementation techniques for the vehicle routing problem, "Computer & Operations Research 12 (1985): 273-283
8. H. Paessens , "The savings algorithm for the vehicle routing problem, " European Journal of Operations Research 34 (1988): 336-344.
9. John Pooley, "A Vehicle Routing Algorithm For Less-Than-Truckload vs. Multiple-stop Truckload Problem," Journal of Business Logistics, 13(1992): 239-258.
10. 李宗儒、翁基華（1998）「多目標車輛途程整數規劃模式之研究」，農產運銷論叢，3：197-210。
11. 李宗儒、翁基華（1997）「配銷系統之車輛途程問題之研究，於農產運銷的應用-以規劃農會超市宅配送為例」，台灣土地金融季刊，34（1）；147-160。
12. 陳春益、林正章、高玉明（1997）「路線貨運業貨物排程模式之研究」，運輸計劃季刊，26（2）：327-352。
13. 韓復華、卓裕仁（1996）「門檻接受法、噪音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，9（3）：113-144。
14. 韓復華、楊智凱、卓裕仁（1997）「應用門檻接受法求解車輛路線問題之研究」，運輸計劃季刊，26（2）：253-280。

附錄

```
MIN    0.0X111 +   42.0X121 +   31.5X131 +   21.0X141 +   25.5X151 +   27.0X161 +   33.0X171 +
  42.0X211 +    0.0X221 +   70.5X231 +   54.0X241 +   37.5X251 +   30.0X261 +   52.5X271 +
  31.5X311 +   70.5X321 +    0.0X331 +   39.0X341 +   55.5X351 +   45.0X361 +   30.0X371 +
  21.0X411 +   54.0X421 +   39.0X431 +    0.0X441 +   22.5X451 +   46.5X461 +   51.0X471 +
  25.5X511 +   37.5X521 +   55.5X531 +   22.5X541 +    0.0X551 +   43.5X561 +   58.5X571 +
  27.0X611 +   30.0X621 +   45.0X631 +   46.5X641 +   43.5X651 +    0.0X661 +   24.0X671 +
  33.0X711 +   52.5X721 +   30.0X731 +   51.0X741 +   58.5X751 +   24.0X761 +    0.0X771 +
  42.0X212 +    0.0X222 +   70.5X232 +   54.0X242 +   37.5X252 +   30.0X262 +   52.5X272 +
  31.5X312 +   70.5X322 +    0.0X332 +   39.0X342 +   55.5X352 +   45.0X362 +   30.0X372 +
  21.0X412 +   54.0X422 +   39.0X432 +    0.0X442 +   22.5X452 +   46.5X462 +   51.0X472 +
  25.5X512 +   37.5X522 +   55.5X532 +   22.5X542 +    0.0X552 +   43.5X562 +   58.5X572 +
  27.0X612 +   30.0X622 +   45.0X632 +   46.5X642 +   43.5X652 +    0.0X662 +   24.0X672 +
  33.0X712 +   52.5X722 +   30.0X732 +   51.0X742 +   58.5X752 +   24.0X762 +    0.0X772 +
168.0L2     + 126.0L3     +   84.0L4     + 102.0L5     + 108.0L6     + 132.0L7
SUBJECT TO
Y11 + Y12 = 2
Y21 + Y22 + L2 = 1
Y31 + Y32 + L3 = 1
Y41 + Y42 + L4 = 1
Y51 + Y52 + L5 = 1
Y61 + Y62 + L6 = 1
Y71 + Y72 + L7 = 1
2Y21 + 3Y31 + 1Y41 + 1Y51 + 2Y61 +     1Y71 <=    4
2Y22 + 3Y32 + 1Y42 + 1Y52 + 2Y62 +     1Y72 <=    4
0X111 + 1X211 + 1X311 + 1X411 + 1X511 + 1X611 + 1X711 - Y11 =0
1X121 + 0X221 + 1X321 + 1X421 + 1X521 + 1X621 + 1X721 - Y21 =0
1X131 + 1X231 + 0X331 + 1X431 + 1X531 + 1X631 + 1X731 - Y31 =0
1X141 + 1X241 + 1X341 + 0X441 + 1X541 + 1X641 + 1X741 - Y41 =0
1X151 + 1X251 + 1X351 + 1X451 + 0X551 + 1X651 + 1X751 - Y51 =0
1X161 + 1X261 + 1X361 + 1X461 + 1X561 + 0X661 + 1X761 - Y61 =0
1X171 + 1X271 + 1X371 + 1X471 + 1X571 + 1X671 + 0X771 - Y71 =0
0X112 + 1X212 + 1X312 + 1X412 + 1X512 + 1X612 + 1X712 - Y12 =0
1X122 + 0X222 + 1X322 + 1X422 + 1X522 + 1X622 + 1X722 - Y22 =0
1X132 + 1X232 + 0X332 + 1X432 + 1X532 + 1X632 + 1X732 - Y32 =0
1X142 + 1X242 + 1X342 + 0X442 + 1X542 + 1X642 + 1X742 - Y42 =0
1X152 + 1X252 + 1X352 + 1X452 + 0X552 + 1X652 + 1X752 - Y52 =0
1X162 + 1X262 + 1X362 + 1X462 + 1X562 + 0X662 + 1X762 - Y62 =0
1X172 + 1X272 + 1X372 + 1X472 + 1X572 + 1X672 + 0X772 - Y72 =0
0X111 + 1X121 + 1X131 + 1X141 + 1X151 + 1X161 + 1X171 - Y11 =0
1X211 + 0X221 + 1X231 + 1X241 + 1X251 + 1X261 + 1X271 - Y21 =0
1X311 + 1X321 + 0X331 + 1X341 + 1X351 + 1X361 + 1X371 - Y31 =0
1X411 + 1X421 + 1X431 + 0X441 + 1X451 + 1X461 + 1X471 - Y41 =0
1X511 + 1X521 + 1X531 + 1X541 + 0X551 + 1X561 + 1X571 - Y51 =0
1X611 + 1X621 + 1X631 + 1X641 + 1X651 + 0X661 + 1X671 - Y61 =0
1X711 + 1X721 + 1X731 + 1X741 + 1X751 + 1X761 + 0X771 - Y71 =0
0X112 + 1X122 + 1X132 + 1X142 + 1X152 + 1X162 + 1X172 - Y12 =0
1X212 + 0X222 + 1X232 + 1X242 + 1X252 + 1X262 + 1X272 - Y22 =0
1X312 + 1X322 + 0X332 + 1X342 + 1X352 + 1X362 + 1X372 - Y32 =0
1X412 + 1X422 + 1X432 + 0X442 + 1X452 + 1X462 + 1X472 - Y42 =0
1X512 + 1X522 + 1X532 + 1X542 + 0X552 + 1X562 + 1X572 - Y52 =0
1X612 + 1X622 + 1X632 + 1X642 + 1X652 + 0X662 + 1X672 - Y62 =0
1X712 + 1X722 + 1X732 + 1X742 + 1X752 + 1X762 + 0X772 - Y72 =0
END
INTE 112
```

Available online at www.sciencedirect.com
SCIENCE DIRECT®
European Journal of Operational Research 165 (2005) 657–667
EUROPEAN JOURNAL OF OPERATIONAL RESEARCH
www.elsevier.com/locate/dsw

Discrete Optimization

A heuristic algorithm for the truckload and less-than-truckload problem

Ching-Wu Chu *

Department of Shipping and Transportation Management, National Taiwan Ocean University, 2, Pei Ning Road, Keelung, Taiwan

Received 11 December 2002; accepted 22 August 2003
Available online 25 March 2004

Abstract

The delivery of goods from a warehouse to local customers is an important and practical problem of a logistics manager. In reality, we are facing the fluctuation of demand. When the total demand is greater than the whole capacity of owned trucks, the logistics managers may consider using an outsider carrier.

Logistics managers can make a selection between a truckload (a private truck) and a less-than-truckload carrier (an outsider carrier). Selecting the right mode to transport a shipment may bring significant cost savings to the company.

In this paper, we address the problem of routing a fixed number of trucks with limited capacity from a central warehouse to customers with known demand. The objective of this paper is developing a heuristic algorithm to route the private trucks and to make a selection of less-than-truckload carriers by minimizing a total cost function. Both the mathematical model and the heuristic algorithm are developed. Finally, some computational results and suggestions for future research are presented.

Keywords: Vehicle routing; Heuristics; 0–1 integer programming; Logistics

1. Introduction

The delivery of goods from a warehouse to local customers is an important and practical problem of a logistics manager. In many sectors of the economy, transportation costs amount for a fifth or even a quarter (lumber, wood, petroleum, stone, clay, and glass products) of the average sales dollars [1].

Logistics managers can make a selection between a truckload (a private truck) and a less-than-truckload carrier (an outsider carrier). A private truck allows a company to consolidate several shipments, going to different destinations, in a single truck. A less-than-truckload carrier usually assumes the responsibility for routing each shipment from origin to destination. The freight charged by a less-than-truckload carrier is usually much higher than the cost of a private truck. Selecting the right mode to transport a shipment may yield significant cost savings to the company.

Our motivation for this study stems from observations on a local logistics company. This

* Tel.: +886-2-24622192x3407; fax: +886-2-24631903.
E-mail address: cwchu@mail.ntou.edu.tw (C.-W. Chu).

0377-2217/$ - see front matter
doi:10.1016/j.ejor.2003.08.067

company owns different types of trucks and main business of this company is delivering foods and beverages to wholesalers. Since the business hours of the wholesalers are fixed, the delivery time window constraint is not major concern for this company. But, this company is facing the fluctuation of demand within a year. When the demands are greater than the whole capacity of owned trucks during the peak season, there are two ways to deal with this situation. One is asking truck drivers to work overtime; the other is using the outsider carriers. Since the overtime cost is much higher than that of using an outsider carrier, the logistics managers may consider using an outsider carrier.

In this paper, we address the problem of routing a fixed number of trucks with limited capacity from a central warehouse to customers with known demand. The objective of this paper is developing a heuristic algorithm to route the private trucks and to make a selection of less-than-truckload carriers by minimizing a total cost function.

The literature on vehicle routing problem has been concerned almost exclusively with heuristics. Several families of heuristics have been proposed for the VRP. These can be broadly classified into two main classes: classical heuristics developed mostly between 1960 and 1990, and meta-heuristics whose growth has occurred in the last decade [2]. In general, the classical heuristics are of four types: (1) tour building heuristics, (ii) tour improvement heuristics, (iii) two-phase method, (iv) incomplete optimization methods.

The most often mentioned tour building heuristics is the Clarke and Wright method [3]. There have been many modifications to the basic Clarke and Wright method. Gaskell [4] and Yellow [5] independently introduced the concept of a modified savings given by $S_{ij} - \theta C_{ij}$ where is θ a scalar parameter. One can change emphasis on the cost of travel between two nodes by varying θ.

The tour improvement heuristics are based on Lin [6] and Lin–Kernighan [7] heuristics for the traveling salesman problem. Christofides and Eilon [8] have modified this heuristic for vehicle routing problem. Two-phase methods include those of Gillett and Miller [9] and Christofides et al. [10]. The example of a heuristic based on incomplete optimization is the tree-search method reported in [10].

The meta-heuristics, presented below, is restricted to tabu search methods since these have been proved the most successful meta-heuristic approach. Over the past decade, tabu search have been applied to the VRP by several authors. Osman [11], Taillard [12], Gendreau et al. [13], Rochat and Taillard [14], Xu and Kelly [15] and Rego and Roucairol [16] all obtained quite satisfactory results.

Very little research has examined the problem of selecting between a less-than-truckload and truckload carrier. Ball et al. [17] consider a fleet planning problem for long-haul deliveries with fixed delivery locations and an option to use an outside carrier. Agarwal [18] considers the static problem with a fixed fleet size and an option to use an outside carrier. Klincewicz et al. [19] develop a methodology to address the fleet size planning and to route limited trucks from a central warehouse to customers with random daily demands.

In general, our research described here differs from previous fleet planning or vehicle routing in that it modifies the Clarke and Wright method by shifting from distance to cost and also incorporates the fixed cost of different types of trucks into the model; it allows the permutations of the three improvement procedures that will result in best results; it simultaneously considers the determination of routing a heterogeneous fleet vehicles and the selection of less-than-truckload carriers; it also presents a mathematical model for solving the problem.

This paper is organized as follows. Next section formulates the mathematical model for our problem. Section 3 presents the heuristic algorithm. Some computational results are reported in Section 4. Finally some concluding remarks and suggestions for future research are provided in Section 5.

2. Mathematical model

To simplify the analysis, we formulate our mathematical model based on the following assumptions:

1. We consider one warehouse system; all trucks start at the warehouse and return back to the warehouse.
2. The requirements of all the customers are known; the requirement of each customer cannot exceed the truck capacity;
3. Each customer is served by one truck (either by the private truck or the less-than-truckload carrier); the requirements of all the customers must be met.
4. We restrict ourselves to delivery only.
5. The cost of operating the truck fleet consists of fixed cost and variable cost. Principal cost items in fixed cost include personnel, insurance, and truck depreciation. The main item of variable cost is fuel. It is usually proportional to the distance of truck traveled.

In the following we present an integer programming model and relevant notations:

$$\min\ z = \sum_{k}^{m} \mathrm{FC}_k + \sum_{i}^{n}\sum_{j}^{n}\sum_{k}^{m} C_{ijk}X_{ijk} + \sum_{i}^{n} \mathrm{CL}_i L_i$$

subject to

$$\sum_{k}^{m} Y_{0k} = m \qquad , \tag{1}$$

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

$$\sum_{i}^{n} q_i Y_{ik} \leqslant Q_k \quad (\qquad k = 1, \dots, m), \tag{3}$$

$$\sum_{j}^{n} X_{ijk} = Y_{ik} \quad (i = 1, \dots, n;\ k = 1, \dots, m), \tag{4}$$

$$\sum_{j}^{n} X_{jik} = Y_{ik} \quad (i = 1, \dots, n;\ k = 1, \dots, m), \tag{5}$$

$$\sum_{ij \in s} X_{ijk} \leqslant |S| - 1 \text{ for all } S \subseteq \{2, \dots, n\} \quad (k = 1, \dots, m), \tag{6}$$

$$X_{ijk} \in \{0,1\}; \quad Y_{ik} \in \{0,1\}; \quad L_i \in \{0,1\} \quad (i = 0, \dots, n;\ j = 0, \dots, n;\ k = 1, \dots, m),$$

i: $\{i = 0, \dots, n\}$, the index set of customers (let the index 0 denote the warehouse);
j: $\{j = 0, \dots, n\}$, the index set of customers;
k: $\{k = 1, \dots, m\}$, the index set of trucks;
n: the number of customers;
m: the number of trucks;

$$X_{ijk} = \begin{cases} 1 & \text{if truck } k \text{ travels from customer } i \text{ to customer } j, \\ 0 & \text{otherwise,} \end{cases}$$

$$L_i = \begin{cases} 1 & \text{if the demand of customer } i \text{ is satisfied by the less-than-truckload carrier,} \\ 0 & \text{otherwise,} \end{cases}$$

$$Y_{ik} = \begin{cases} 1 & \text{if the demand of customer } i \text{ is satisfied by the private vehicle } k, \\ 0 & \text{otherwise,} \end{cases}$$

FC_k: fixed cost of private truck k;
C_{ijk}: the cost of truck k traveling from customer i to customer j;
CL_i: the cost charged by the less-than-truckload carrier for serving customer i;
q_i: the demand of customer i;
Q_i: the capacity of private truck i.

The objective is to route the private trucks and to make a selection of less-than-truckload carriers by minimizing a total cost function.

Constraint (1) ensure that all trucks have been assigned to customers.
Constraint (2) ensure that each customer is served either by the private truck or the less-than-truckload carrier.
Constraint (3) are the truck capacity constraints.
Constraints (4) and (5) ensure that a truck arrives at a customer and also leaves that location.
Constraint (6) serve as subtour-breaking constraints.

3. Heuristic algorithm

In this section we describe an algorithm, called TL–LTL, for solving the vehicle routing and the selection of less-than-truckload carriers problem. The heuristic algorithm can be decomposed into three main steps. In the following we describe algorithm TL–LTL by examining its main steps separately.

3.1. Selection step

The first step of algorithm TL–LTL requires the selection of a group of customers, who will be served by the less-than-truckload carriers. In this step, we will check if the total demand is greater than the whole capacity of owned trucks. If the answer is not, we will skip this step and implement next step directly.

In order to minimize the total cost, we have to design a procedure that can achieve this goal. In reality, the freight charged by the less-than-truckload carrier is usually higher than the cost handled by a private truck. It is obvious that we should order the customers in ascending order based on the freight charged by the less-than-truckload carrier and choose the customers with the lowest cost.

The detail for selecting the customers is described as follows:

(1) Calculate the total demand for all customers.
(2) Calculate the whole capacity of owned trucks.
(3) If the total demand for all customers is greater than the whole capacity of owned trucks, go to step (4) otherwise skip this procedure.
(4) Subtract the whole capacity of own trucks from the total demand for all customers, which is the unsatisfied truck capacity.
(5) Order the customers in ascending order based on the freight charged by the less-than-truckload carrier. Starting at top of the list, do the following.
(6) Sum up the demand of each customer until the total demand is greater than the unsatisfied truck capacity. The corresponding customers will be served by the less-than-truckload carrier; the remaining customers in the list will be served by private trucks and will be used for constructing initial solution.

3.2. Initial solution construction

The Clarke and Wright's savings algorithm is used to solve this problem by making two modifications. The first modification to the algorithm is a shift in criterion from distance to cost. The second modification of the Clarke and Wright formulation is a change in the savings calculation.

The mathematical relationship of the savings of linking two customers is a function of the mix of a less-than-truckload carrier and a private truck that serve customers. There are three possible mixes serving a pair of customers: (1) two less-than-truckload carriers; (2) a private truck and a less-than-truckload carrier; (3) two private trucks.

Before explaining the revised savings calculation, we list the relevant notations as follows:

S_{ij} = savings from consolidating shipments to customer i and j into the same truck.
LTL_i = the total cost charged by the less-than-truckload carrier for serving customer i.
TL_{ij} = the total cost of a private truck that travels from warehouse to customer i, then from customer i to customer j and finally returns back to warehouse.
$\mathrm{FC}(Z)$ = the fixed cost of the smallest truck that can serve a demand of Z.
d_{ij} = the distance from customer i to customer j.
v = the cost of traveling a mile for private truck (\$/per mile).

Fig. 1 illustrates the revised savings calculation from linking two customers under each of the three possible mixes.

The detail for constructing the initial solution is described as follows:

(1) Calculate the savings for all pairs customers based on revised savings scenario 1 in Fig. 1.
(2) Order the savings in descending order. Starting at top of the list, do the following.
(3) Find the feasible link in the list which can be used to extend one of the two ends of the currently constructed route.

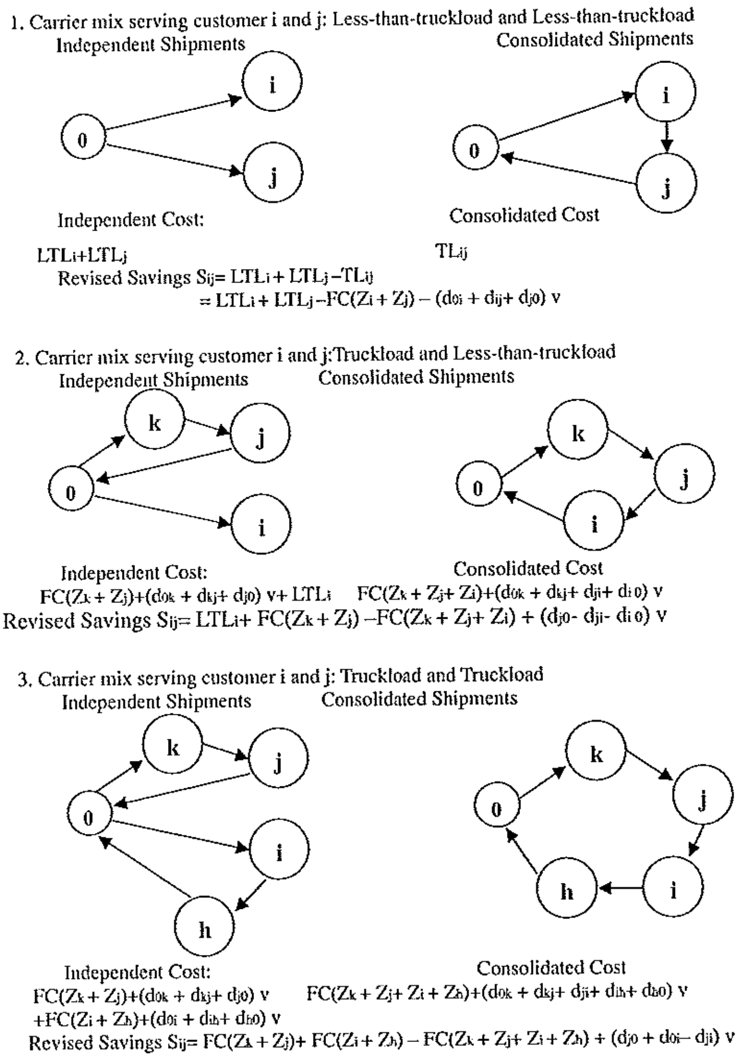


Fig. 1. Savings calculation from consolidating two customers.

(4) If the route cannot be expanded further, terminate the route. Choose the first feasible link in the list to start a new route.
(5) Repeat Steps (3) and (4) until no more links can be chosen.
(6) Output all the temporary single-customer routes (served by the less-than-truckload carriers) and multi-customer routes.
(7) Calculate the savings for single-customer routes based on revised savings scenario 2 in Fig. 1.
(8) Order the savings in descending order. Starting at top of the list, do the following.
(9) Find the feasible link in the current multi-customer routes which can be used to extend the route.
(10) If the route cannot be expanded further, terminate the route.
(11) Repeat Steps (9) and (10) until no more links can be chosen.
(12) Output all the routes.

3.3. Refining procedure

A refining procedure is applied to the solution obtained through the initial solution step. This procedure is composed of a succession of intra-route and inter-route arc exchanges.

3.3.1. Intra-route improvement

Each route is improved by applying a refining procedure which considers all the feasible exchanges of two arcs belong to the route (the so-called intra-route two-exchanges [20]). The procedure is similar to those described in Christofides and Eilon [8] and Kindervater and Savelsbergh [21]. Given a route, a two-exchange is obtained by replacing arcs (m,n) and (p,q) with arcs (m,p) and (n,q), as illustrated in Fig. 2.

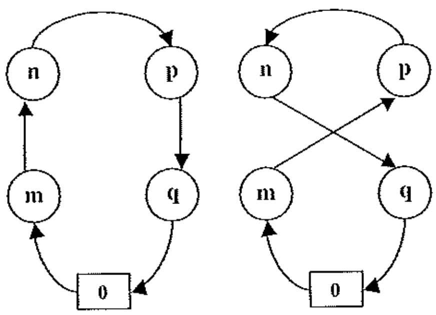


Fig. 2. Example of intra-route two-exchanges.

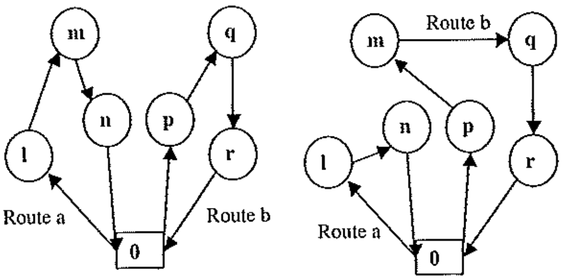


Fig. 3. Example of inter-route one-exchange.

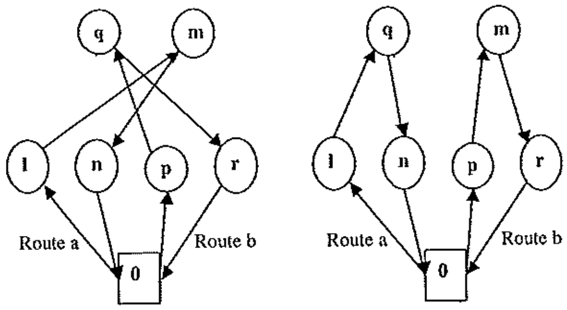


Fig. 4. Example of inter-route two-exchanges.

3.3.2. Inter-route improvement

In this step, a set of routes is obtained by using further local search procedures. These procedures are based on the so-called inter-route one-exchange and two-exchanges, illustrated in Figs. 3 and 4, respectively.

For each node m, belonging to route a, the one-exchange corresponding to its insertion after node p, belonging to route b, is obtained by removing arcs (l,m), (m,n) and (p,q), and replacing them with arcs (l,n), (p,m) and (m,q), as illustrated in Fig. 3.

For each node m, belonging to route a, the two-exchange corresponding to its exchange with node q, belonging to route b, is obtained by removing arcs (l,m), (m,n), (p,q) and (q,r), and replacing them with arcs (l,q), (q,n), (p,m) and (m,r), as illustrated in Fig. 4.

3.3.3. Search procedure

A search procedure is designed in searching for a better solution. From the results of extensive experiments which are not shown here, we know that the implementation sequence of intra-route and inter-route improvement procedure might have impacts on the quality of solution.

The improvement procedures mentioned above include intra-route two-exchanges, inter-route one-exchange and two-exchanges. The possible permutations of three different improvement procedures are only six, so a loop procedure consisting of arranging the possible sequences of intra-route and inter-route improvement is applied on the solution obtained in the initial solution construction phase. The purpose of this loop proce-

Table 1
Results for five test problems

Test problem		Routes	Total cost	CPU[a] time	% larger than the best solution
1	Heuristic algorithm	1-3-5-4-1 1-6-1 Customer 2 is served by LTL	387.5	3.14	0
	Mathematical model	1-3-5-4-1 1-6-1 Customer 2 is served by LTL	387.5	1.1	
2	Heuristic algorithm	1-9-8-7-5-11-1 1-2-4-3-10-1 Customer 6 is served by LTL	631	4.58	7.67
	Mathematical model	1-4-3-10-11-5-1 1-2-9-8-7-1 Customer 6 is served by LTL	586	34.5	
3	Heuristic algorithm	1-3-2-4-11-10-1 1-12-15-8-13-1 1-16-6-14-9-7-1 Customer 5 is served by LTL	900	5.88	0
	Mathematical model	1-8-12-4-2-3-1 1-7-13-10-11-1 1-9-15-14-16-6-1 Customer 5 is served by LTL	900	578	
4	Heuristic algorithm	1-17-16-4-3-2-7-14-10-6-5-9-1 1-22-20-19-21-15-18-23-12-8-13-1 Customer 11 is served by LTL	1681.5	8.42	1.81
	Mathematical model	1-20-23-21-19-15-18-16-17-4-3-2-8-10-13-1 1-7-14-12-6-5-9-22-1 Customer 11 is served by LTL	1651.5	830	
5	Heuristic algorithm	1-19-24-9-15-30-28-27-29-1 1-11-12-13-10-18-8-14-17-16-1 1-20-21-23-7-26-25-2-6-5-4-1 Customers 3 and 22 are served by LTL	1917	11.06	0.86
	Mathematical model	1-3-5-2-1 1-22-18-17-27-29-28-26-25-23-21-1 1-4-6-7-30-16-14-8-10-15-9-13-12-11-1 Customers 19, 20 and 24 are served by LTL	1900.5	2406	

[a] All times are in seconds; the results were obtained on a PC running at 2000 MHz.

dure is in a sense to similar to the tabu search method to escape from a local minimum. Once a better solution is found after finishing improvement phase, the best solution record is updated. We repeat the above improvement processes until all possible permutations of three different improvement procedures have been implemented.

4. Computational results

In this section, we summarize our computational results on five test problems. The detailed data associated with five examples are given in Appendix A. The solutions produced by the heuristic algorithm are compared with the optimal results from the mathematical model. The heuristic algorithm was written in FORTRAN language and the mathematical model was solved using the software LINDO version 6.1. Both of them were implemented on a PC with a 2000 MHz processor. Computational results on five test problems are reported in Table 1.

For the first and the third test problems, our heuristic algorithm obtains the optimal solution. As shown in Table 1, both the mathematical model and the heuristic algorithm yield the same total cost \$387.5 and \$900, respectively. The only difference between two approaches in the third test problem is in that each approach arranges customers in different routes and in different sequences.

Computationally, exact algorithm for the VRP is restricted to solving problems of only up to about 25 customers. For five test problems, the solution time of mathematical model increased quickly with problem size. On the other side, our heuristic algorithm required very little time to solve the problem. Every problem took only a few seconds. The CPU time of test problems is not very sensitive to problem size.

In order to test whether the solution time of our algorithm is not sensitive to larger size of problem, we have solved additional three test problems with the customer size of 51, 76 and 101, respectively. Because the VRP is very difficult to solve with mathematical model even for relatively small size instances, only the average computation times to run the heuristic are reported. These results are presented in Table 2. Though the solution's time increased with problem size, it is obvious that the solution's time increase gradually without rapid growth.

Table 2
Results for larger size of test problems

Test problem[a]	CPU[b] time
1 [E-n51-K5]	27.84
2 [E-n76-K7]	84.48
3 [E-n101-K8]	192.48

[a] Test problem 1 and 3 can be found in Christofides and Eilon [8]; test problem 2 can be found in Gillett and Miller [9].
[b] All times are in seconds; the results were obtained on a PC running at 2000 MHz.

5. Conclusions

The delivery of goods from a warehouse to local customers is an important and practical problem of a logistics manager. In this paper, we develop both the mathematical model and the heuristic algorithm for solving the less-than-truckload and truckload problem. Some computational results are presented. Our heuristic algorithm obtains the optimal or near-optimal solutions in an efficient way in terms of time and accuracy.

As for further research, a wide range of test problems should be performed. It would be interesting to see if other intelligent optimization techniques, such as tabu search, genetic algorithms, simulated annealing and neural networks, can be modified to solve this problem and even provide better results.

Acknowledgements

The author wishes to thank a referee for his valuable comments. This work was partially supported by the National Science Council of Taiwan under grant NSC 89-2416-H-019-004.

Appendix A

Details of problem 1

No.	x	y	Q	LTL
1	35	35	0	0
2	41	49	10	90
3	35	17	7	108
4	55	45	13	132
5	55	20	19	150
6	15	30	26	120

Warehouse co-ordinates(35,35);
Customer demands (q) in cwt.

Vehicle	Capacity	Fixed Cost
1	40 cwt	60
2	30 cwt	50

The variable cost for private Vehicles is $1.5/per mile

Details of problem 2

No.	x	y	q	LTL
1	30	40	0	0
2	37	52	7	78
3	49	49	30	126
4	52	64	16	192
5	20	26	9	102
6	40	30	21	84
7	21	47	15	66
8	17	63	19	156
9	31	62	23	132
10	52	33	11	138
11	51	21	5	168

Warehouse co-ordinates(30,40);
Customer demands (q) in cwt.

Vehicle	Capacity	Fixed Cost
1	75 cwt	120
2	65 cwt	100

The variable cost for private vehicles is $1.5/per mile

Details of problem 3

No.	x	y	Q	LTL
1	40	40	0	0
2	22	22	18	150
3	36	26	26	84
4	21	45	11	114
5	45	35	30	42
6	55	20	21	150
7	33	34	19	54
8	50	50	15	84
9	55	45	16	90
10	26	59	29	138
11	40	66	26	156
12	55	65	37	174
13	35	51	16	72
14	62	35	12	132
15	62	57	31	162
16	62	24	8	162

Warehouse co-ordinates(40,40);
Customer demands (q) in cwt.

Vehicle	Capacity	Fixed Cost
1	110 cwt	150
2	100 cwt	140
3	90 cwt	130

The variable cost for private vehicles is $1.5/per mile

Details of problem 4

No.	x	y	Q	LTL
1	266	235	0	0
2	295	272	125	282
3	301	258	84	246
4	309	260	60	294
5	217	274	500	372
6	218	278	300	384
7	282	267	175	210
8	242	249	350	162
9	230	262	150	270
10	249	268	1100	222
11	256	267	4100	198
12	265	257	225	132
13	267	242	300	42
14	259	265	250	180
15	315	233	500	294
16	329	252	150	390
17	318	252	100	324
18	329	224	250	378
19	267	213	120	132
20	275	192	600	258
21	303	201	500	300
22	208	217	175	360
23	326	181	75	480

Warehouse co-ordinates (266,235);
Customer demands (q) in cwt

Vehicle	Capacity	Fixed Cost
1	4500 cwt	250
2	4000 cwt	200

The variable cost for private Vehicles is $1.5/per mile

Details of problem 5

No.	x	y	q	LTL	No.	x	y	q	LTL
1	162	354	0	0	21	180	360	300	114
2	218	382	300	372	22	159	331	1500	138
3	218	358	3100	336	23	188	357	100	156
4	201	370	125	252	24	152	349	300	66
5	214	371	100	324	25	215	389	500	378
6	224	370	200	384	26	212	394	800	384
7	210	382	150	330	27	188	393	300	276
8	104	354	150	348	28	207	406	100	408
9	126	338	450	234	29	184	410	150	360
10	119	340	300	270	30	207	392	1000	348
11	129	349	100	198					
12	126	347	950	216					
13	125	346	125	222					
14	116	355	150	276					
15	126	335	150	240					
16	125	355	550	222					
17	119	357	150	258					
18	115	341	100	288					
19	153	351	150	54					
20	175	363	400	90					

Warehouse co-ordinates (162,354);
Customer demands (q) in cwt

Vehicle	Capacity	Fixed Cost
1	4500 cwt	250
2	4000 cwt	200
3	3500 cwt	180

The variable cost for private Vehicles is $1.5/per mile

References

[1] L.M. Schneider, New era in transportation strategy, Transportation Strategy (1985) 118–126.
[2] G. Laporte, M. Gendreau, J.-Y. Potvin, F. Semet, Classical and modern heuristics for the vehicle routing problem, International Transactions in Operational Research 7 (2000) 285–300.
[3] G. Clarke, J. Wright, Scheduling of vehicles from a central depot to a number of delivery points, Operational Research 12 (1964) 568–581.
[4] T.J. Gaskell, Basis for vehicle fleet scheduling, Operational Research Quarterly 18 (1967) 281.
[5] P. Yellow, A computational modification to the savings method of vehicle scheduling, Operational Research Quarterly 21 (1970) 281.
[6] S. Lin, Computer solution of the traveling salesman problem, Bell System Technology Journal 44 (1965) 2245–2269.
[7] S. Lin, B. Kernighan, An effective heuristic algorithm for the traveling salesman problem, Operational Research 21 (1973) 498–516.
[8] N. Christofides, S. Eilon, An algorithm for the vehicle dispatching problems, Operational Research Quarterly 20 (1969) 309–318.
[9] B. Gillett, L. Miller, A heuristic algorithm for the vehicle dispatch problem, Operational Research 22 (1974) 340–349.
[10] N. Christofides, A. Mingozzi, P. Toth, The Vehicle Routing Problem, Combinatorial Optimization, Wiley, Chichester, 1979, pp. 315–338.
[11] I.H. Osman, Metastrategy simulated annealing and tabu search algorithm for the vehicle routing problem, Annals of Operations Research 41 (1993) 421–451.
[12] É.D. Taillard, Parallel iterative search methods for vehicle routing problem, Networks 23 (1993) 661–673.
[13] M. Gendreau, A. Hertz, G. Laporte, A tabu search heuristic for the vehicle routing problem, Management Science 40 (1994) 1276–1290.
[14] Y. Rochat, É.D. Taillard, Probabilistic diversification and intensification in local search for vehicle routing, Journal of Heuristics 1 (1995) 147–167.
[15] J. Xu, J.P. Kelly, A network flow-based tabu search heuristic for the vehicle routing problem, Transportation Science 30 (1996) 379–393.
[16] C. Rego, C. Roucairol, A parallel tabu search algorithm using ejection chains for vehicle routing problem, in: I.H. Osman, J.P. Kelly (Eds.), Meta-Heuristics: Theory and Applications, Kluwer, Boston, 1996.
[17] M.O. Ball, A. Golden, A. Assad, L.D. Bodin, Planning for truck fleet size in the presence of a common-carrier option, Decision Sciences 14 (1983) 103–120.
[18] Y.K. Agarwal, Vehicle routing with limited fleet and common carrier option, presented at TIMS/ORSA Joint National Meeting, Boston, 1985.
[19] J.G. Klincewicz, H. Luss, M.G. Pilcher, Fleet size planning when outside carrier service are available, Transportation Science 24 (1990) 169–182.
[20] P. Toth, D. Vigo, A heuristic algorithm for the symmetric and asymmetric vehicle routing problems with backhauls, European Journal of Operational Research 113 (1999) 528–543.
[21] G.A.P. Kindervater, M.W.P. Savelsbergh, Vehicle routing: Handling edge exchanges, in: E.H. Aarts, J.K. Lenstra (Eds.), Local Search in Combinatorial Optimization, Wiley, Chichester, 1997.