

國立交通大學
運輸科技與管理學系碩士班

碩士論文

運用粒子群最佳化解決多場站之收送貨問題

A Particle Swarm Optimization Approach for
the Multi-depot Vehicle Routing Problem with
Pickup and Delivery

研究生：李忠憲

指導教授：王晉元

中華民國一百年七月

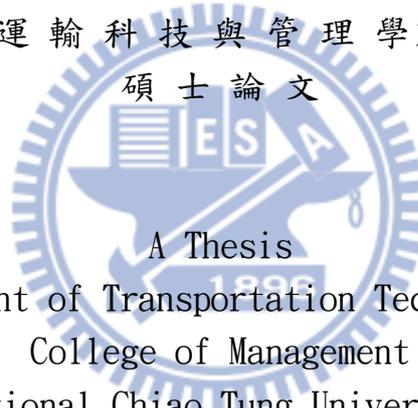
應用粒子群最佳化解決多場站之收送貨問題

A Particle Swarm Optimization Approach for the Multi-depot Vehicle Routing
Problem with Pickup and Delivery

研究生：李忠憲
指導教授：王晉元

Student : Chung-Hsien Li
Advisor : Jin-Yuan Wang

國立交通大學
運輸科技與管理學系
碩士論文



A Thesis
Submitted to Department of Transportation Technology and Management
College of Management
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of
Master
in

Transportation Technology and Management

July 2011

Hsinchu, Taiwan, Republic of China

中華民國一百年七月

應用粒子群最佳化解決多場站之收送貨問題

學生：李忠憲

指導教授：王晉元

國立交通大學運輸科技與管理學系碩士班

摘要

本研究主要針對多場站之收送貨問題(Multi-depot Vehicle Routing Problem with Pickup and Delivery, MDPDP)定式數學模式與發展一套有效的解法。在數學模式部分，本研究參考過去所發表過MDVRP與PDPTW之數學模式，加入考量收送貨的優先限制，建構一個MDPDP之數學模式。優先限制為收送貨問題特性之一，對於每個顧客而言，必須先服務該顧客之收貨後才能服務該顧客之送貨。在演算法部分，本研究以粒子群最佳化(PSO)為基礎，設計一般PSO與GLNPSO兩種演算法來求解MDPDP，比較兩種演算法之求解績效，並提出兩種新的慣性權重更新法則。在編碼方式的設計上，參考Wu and Wang[28]，產生適合於本研究之編碼方式。最後本研究參考Ropke and Pisinger[15]之測試例題，加以修改作為本研究之測試範例。測試結果指出本研究之兩種演算法在求解小範例之績效沒有顯著差異，求解較大規模之問題時，GLNPSO求解績效較優。

關鍵字：多場站收送貨問題，粒子群最佳化，優先限制

A Particle Swarm Optimization Approach for the Multi-depot Vehicle Routing Problem with Pickup and Delivery

Student : Chung-Hsien Li

Advisor : Dr. Jin-Yuan Wang

Department of Transportation Technology and Management
National Chiao Tung University

ABSTRACT

The goals of this research are to develop a mathematical model and an effective solution technique for the Multi-depot vehicle routing problem with pickup and delivery(MDPDP). Our model is based on the published MDVRP and PDPTW formulations plus the proposed precedence constraints. The precedence constraints mean the pickup tasks must be fulfilled prior to the delivery tasks for each customer. Two particle swarm Optimization based heuristic algorithms, general PSO and GLNPSO, are developed for solving MDPDP. The uniqueness of our solution techniques is the mechanism of encoding and decoding of a solution, which is a modification of that of Wu and Wang [28]. Testing examples are generated from the existing benchmark instances. Numerical experiments show that these two algorithms perform equally well for small problems. However, GLNPSO is more effective for solving larger problems.

Keywords: MDPDP, Particle Swarm Optimization, precedence constraints

誌謝

對於我的指導教授王晉元老師兩年來的指導，不管是身教或言教，都確確實實影響到我，舉凡像是看paper之技巧、寫論文之邏輯、程式語言之觀念以及做人處事的道理，讓我了解到所謂一日為師，終生為父之意義。其實跟學長姐相較之下，我的資質駑鈍，很謝謝老師對我有耐心，感激您對我的所有敦敦教誨。

感謝論文期中報告審查委員黃寬丞老師與姚明忠老師，於論文期中報告給予之寶貴建議與評論，讓我對論文期中報告後續研究小心謹慎；感謝論文口試委員中華大學蘇昭銘老師以及本系的韓復華老師，於口試前花時間審閱此論文，並給予許多寶貴建議，使本論文能夠更加完善，在此致上十二萬分的謝意。

感謝實驗室的學長姐們，尤其是學姐一姐、Dasiy、小惠、阿尼基以及HOHO，在我剛開始學程式時遇到之瓶頸給予我鼓勵跟方向，讓我論文得以順利完成。也謝謝陪伴著我的同學們，尤其是馬岱、小朱、阿勇、阿胖、凱開和我組成之GOGOBOY團體，想當初我們還一起作詞作曲和聯誼夜遊樣樣都來，讓我研究所兩年中有歡笑與扶持。也謝謝學弟妹們，有你們的加入，讓我們在繁忙又有壓力的碩二給予熱鬧與生機。最後也謝謝跟我同個LAB一起畢業的阿勇，有你在的地方，總會有歡笑；更謝謝妮妮、趙哥、鱷魚、八賴、+1和劉丸，在我的論文上幫助不少，我會銘記在心。

最重要的是感謝我的姐姐與媽媽，我們家境不好，但你們仍為了讓我順利完成學業，不辭辛苦的工作以及照顧久病的父親，一天睡不上幾個小時，即使家裡有事情，也不願意讓我知道太多而過於擔憂，只要我好好完成論文的順利畢業，真的很謝謝您們，因為有妳們，才会有這本論文，未來我也不會辜負妳們。

最後，誠心的感謝我認識的所有人，並與您們分享完成論文的喜悅與成就感。

李忠憲 謹誌

中華民國一百年七月 於新竹交大

目錄

中文摘要	I
英文摘要	II
誌謝	III
目錄	IV
表目錄	VI
圖目錄	VIII
第一章 緒論	1
1.1 研究動機	1
1.2 研究目的	2
1.3 研究範圍與相關名詞說明	2
1.4 研究流程	3
第二章 文獻回顧	5
2.1 MDPDP 相關文獻之回顧	5
2.2 多場站車輛路線問題(MULTI-DEPOT VEHICLE ROUTING PROBLEM, MDVRP)....	5
2.3 收送貨問題(PICKUPS AND DELIVERIES PROBLEM, PDP)	7
2.4 粒子群最佳化(PARTICLE SWARM OPTIMIZATION, PSO)	11
2.4.1 粒子群最佳化之發展背景	11
2.4.2 粒子群最佳化演算法之說明	11
2.5 小結	15
第三章 模式建立與演算法設計	16
3.1 多場站收送貨問題之數學模式	16
3.1.1 符號說明	16
3.1.2 數學模式	17
3.2 粒子群集最佳化演算法	18
3.2.1 演算法之架構說明	18
3.3 範例說明	22
第四章、範例測試	25
4.1 測試範例	25
4.2 參數測試	26
4.2.1 粒子數	28
4.2.2 迭代數	30
4.2.3 慣性權重值	32
4.2.4 一般 PSO 學習因子(cp 、 cg)	34
4.2.5 GLNPSO 學習因子(cp 、 cg 、 cl 、 cn)	36
4.2.6 小結	38
4.3 範例測試	38

五、結論與建議	45
5.1 結論	45
5.2 建議	45
六、參考文獻	47
附錄一 各組參數測試之詳細結果	51



表目錄

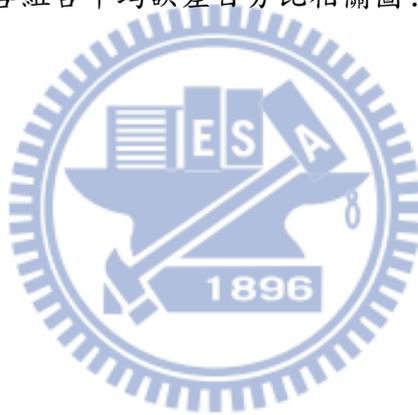
表 2-1 MDPDP 之求解方法相關文獻整理	10
表 3-1 小範例之成本表.....	23
表 4-1 50 個顧客訂單範例之部分資料	26
表 4-2 測試粒子數多寡之參數組	26
表 4-3 最佳目標值誤差百分比與平均目標值誤差百分比之示意表	27
表 4-4 測試粒子數最佳目標值誤差百分比與求解時間.....	28
表 4-5 測試粒子數平均目標值誤差百分比與求解時間.....	29
表 4-6 測試迭代數多寡之參數組	30
表 4-7 測試迭代數最佳目標值誤差百分比與求解時間.....	30
表 4-8 測試迭代數平均目標值誤差百分比與求解時間.....	31
表 4-9 測試慣性權重值之參數組	33
表 4-10 測試慣性權重值之最佳目標值誤差百分比與求解時間.....	33
表 4-11 測試慣性權重值之平均目標值誤差百分比與求解時間.....	33
表 4-12 測試一般 PSO 學習因子之參數組	35
表 4-13 測試一般 PSO 學習因子之值最佳目標值誤差百分比與求解時間.....	35
表 4-14 測試一般 PSO 學習因子之平均目標值誤差百分比與求解時間.....	35
表 4-15 測試一般 PSO 學習因子之參數組	37
表 4-16 測試 GLNPSO 學習因子之最佳目標值誤差百分比與求解時間.....	37
表 4-17 測試 GLNPSO 學習因子之平均目標值誤差百分比與求解時間.....	37
表 4-18 用來評估演算法績效之測試範例資料.....	39
表 4-19 一般 PSO 之演算結果與最佳解之誤差表.....	39
表 4-20 GLNPSO 之演算結果與最佳解之誤差表	39
表 4-21 一般 PSO 求解測試範例之結果.....	40
表 4-22 GLNPSO 求解測試範例之結果	41
表 4-23 粒子群最佳化求解測試範例之顧客訂單數差距百分比.....	42
表 4-24 粒子群最佳化求解測試範例之地理分佈誤差百分比.....	43
表 4-25 粒子群最佳化求解測試範例之平均車輛數.....	43

表 4-26 粒子群最佳化求解測試範例在不同地理分佈下之平均車輛數.....	44
附表 1-1 測試粒子數之各組最佳目標值誤差百分比與求解時間.....	51
附表 1-2 測試粒子數之各組平均目標值誤差百分比與求解時間.....	52
附表 1-3 測試迭代數之各組最佳目標值誤差百分比與求解時間.....	53
附表 1-4 測試迭代數之各組平均目標值誤差百分比與求解時間.....	54
附表 1-5 測試慣性權重值之各組最佳目標值誤差百分比與求解時間.....	55
附表 1-6 測試慣性權重值之各組平均目標值誤差百分比與求解時間.....	55
附表 1-7 一般 PSO 學習因子之各組最佳目標值誤差百分比與求解時間.....	56
附表 1-8 一般 PSO 學習因子之各組平均目標值誤差百分比與求解時間.....	56
附表 1-9 GLNPSO 學習因子之各組最佳目標值誤差百分比與求解時間.....	57
附表 1-10 GLNPSO 學習因子各組平均目標值誤差百分比與求解時間.....	57



圖目錄

圖 1-1 巡迴路徑之優先順序與聯結限制示意圖[35]	2
圖 1-2 研究流程圖	4
圖 2-1 收送貨有無配對之 VRPPD 分類[35]	8
圖 2-2 粒子群最佳化之演算流程	13
圖 3-1 演算法流程圖	18
圖 4-1 粒子數與各組合平均誤差百分比趨勢圖	29
圖 4-2 迭代數與各組合平均誤差百分比相關圖	32
圖 4-3 慣性權重值與各組合平均誤差百分比相關圖	34
圖 4-4 一般 PSO 學習因子與各組合平均誤差百分比相關圖	36
圖 4-5 GLNPSO 學習因子與各組合平均誤差百分比相關圖	38



第一章 緒論

1.1 研究動機

在現今的社會中，供應鏈的觀念早已廣為企業界所熟知與接受，尤其是想在競爭日趨激烈的環境下求生存。供應鏈涉及搜源、採購及物流等所有活動的規劃與管理，並整合企業內部，甚至是企業之間的供應與需求管理，完善的供應鏈管理往往能夠為業者在同業的競爭壓力下取得競爭優勢，並間接降低企業之成本，進而提高了效率與品質[37]。

由上述得知，物流乃是供應鏈其中的一環，由於近年來商業流程縮短、連鎖店興起以及市場採薄利多銷策略，使得物流逐漸受到重視。另一方面，由物流所衍生的研究，像是車輛排程問題(Vehicle Routing Problem, VRP)也因而在國內外都有廣泛的探討。VRP 起初僅考慮單一場站之配送，但隨著顧客需求以及全球趨勢影響，單一場站已無法滿足現今社會之需要，因此衍生出多場站之車輛途程問題(Multi-Depot Vehicle Routing Problem, MDVRP)。

在物流服務中，根據顧客之要求將貨物從指定點收貨再送之另一指定點，這類問題稱之為收送貨問題(Pickup and Delivery Problem, PDP)，此亦為 VRP 之延伸，PDP 與 VRP 不同在於前者所服務的顧客必定同時包含一收貨作業與一送貨作業，且分別有不同之作業地點，並必須以同一車輛進行服務，且需先完成收貨作業才可進行送貨作業；而後者其顧客需求僅為收貨作業或收貨作業之服務。綜合以上所述，結合 MDVRP 與 PDP 之特性，衍生出多場站之收送貨問題(Multi-Depot Pickup and Delivery Problem, MDPDP)。

MDPDP為現今社會中常見之運輸型態，諸如宅配業等，但相關研究甚少。此外MDPDP屬於NP-hard問題[11][22]，故很難在有限時間內找到問題之最佳解，近年來有許多屬於NP-hard問題皆透過如模擬退火法(Simulated Annealing, SA)、禁制搜尋法(Tabu search)等啟發式演算法來求解，其求解效果不差，在實務上被廣為應用。

粒子群最佳化發展十幾年，在啟發式解法中相對其他方法是屬於較新穎的，所以近年來受到相當廣泛的研究(多目標最佳化、類神經網路以及生物識別安全系統)[20]。研究亦發現其用於大規模之問題時，求解之效率與解的品質在標竿範例的表現上都不差[17][30]。

因此本研究將針對較少被研究之MDPDP，參考過去文獻並根據問題定義來建構一數學模式，以粒子群最佳化為基礎下，設計兩種啟發式演算法，對於MDPDP產生在不違反車輛路徑限制與收送貨順序限制之情況下滿足顧客訂單需求的解。

1.2 研究目的

本研究的目的為先建構MDPDP之數學模式，並設計一套粒子群最佳化來求解，在參數測試中，針對粒子群最佳化之參數提出新的更新方式，以範例測試之方式，期能對於各範例皆能找出一組路徑成本最低之配送路徑，最後檢討本研究之演算法績效，供未來研究之進一步探討。

1.3 研究範圍與相關名詞說明

本小節首先定義MDPDP問題相關名詞，再說明本研究所探討的範圍，MDPDP問題相關名詞定義如下：

(1)顧客訂單：公司接獲顧客的訂單後，即產生一個收貨作業與一個送貨作業的配對，完成服務此訂單的收送貨後才算完成此筆顧客需求。

(2)場站：每部車輛之起迄點，在本研究中車輛之起迄點為相同。

(3)路徑：每部車輛從場站出發後，最終再回到場站的路徑稱為路徑。

(4)優先限制(Precedence Constraints)：針對每一顧客需求，收貨作業的服務順序必優先於送貨作業前，如圖1-1所示。

(5)聯結限制(Paring Constraints)：每一顧客需求之收送貨作業均由同一輛車服務。

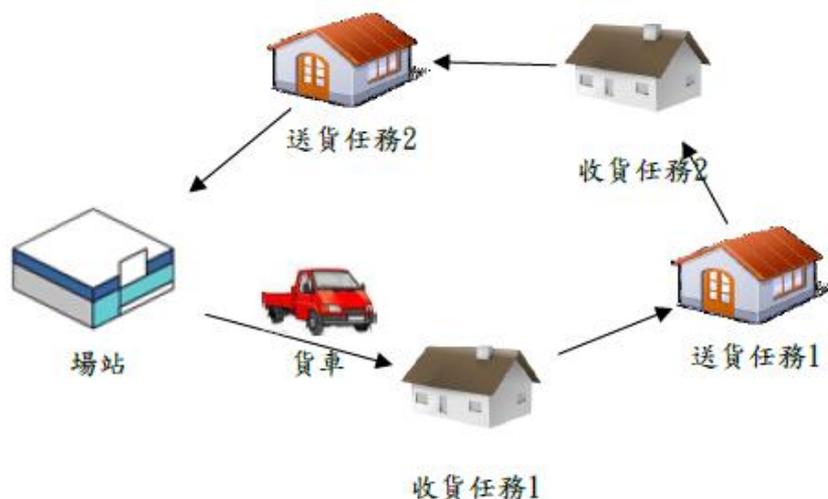


圖 1-1 巡迴路徑之優先順序與聯結限制示意圖[35]

本研究所探討之多場站收送貨問題，其場站位置、場站數量皆已知，場站沒有容量限制，每個場站所擁有的車輛數目已知，車輛僅考慮單一車種，且皆有相同的容量限制，每部車輛所行駛之途徑皆構成一條封閉式的服務路線。

在需求點方面則是分成收貨點與送貨點，同一組的收貨作業需於送貨作業前完成，並由同一車輛服務。因此本研究之MDPDP可描述為：需求包含收貨作業與送貨作業兩部分，車輛由場站空車出發，在不違反車輛容量限制下服務一連串之顧客訂單，一顧客訂單可分為收貨和送貨，且同一訂單之收貨作業必須在送貨作業前先完成，最後再空車回到場站，並尋求總旅行距離成本最小的車輛路徑組合。其問題特性如下：

- (1)每部車由某一場站出發，最終會回到同一場站。
- (2)網路上所有節點需被滿足。
- (3)每個顧客需求包含一個收貨作業與一個送貨作業，須由同一車輛進行服務，且只能服務一次。
- (4)每部車輛所服務之總容量不能超過車輛容量限制。
- (5)一需求指派給一車輛進行服務，則車輛的行駛路線必須包含該需求的收貨作業與送貨作業，且收貨作業的服務順序必須於送貨作業前。
- (6)車輛於需求點完成後立即離開，並前往下一個需求點。
- (7)每部車輛沒有行駛路線長度之限制。

1.4 研究流程

本研究之流程為先界定問題，接著回顧MDPDP之相關文獻與粒子群最佳化之相關研究，並建構一能夠滿足本研究之多場站收送貨問題的數學模式以及應用粒子群最佳化於MDPDP。最後再透過試驗範例，根據其分析結果，提出結論與建議，如圖1-2所示。

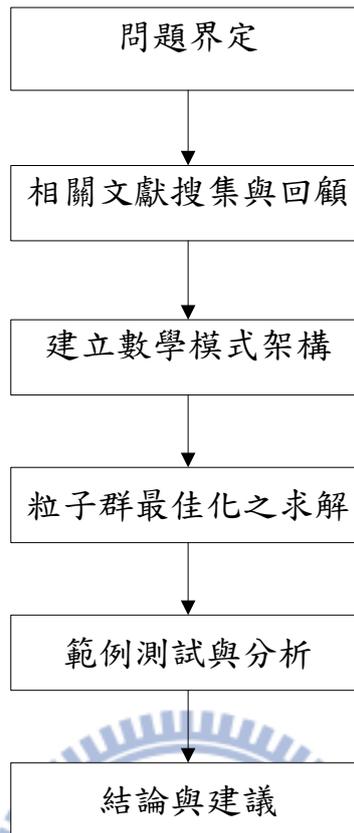


圖 1-2 研究流程圖

第二章 文獻回顧

本研究之課題為 MDPDP，由於過去相關文獻甚少，因此在文獻回顧上除了探討 MDPDP 外，另外分別從多場站之車輛路線問題(Multi-Depot Vehicle Routing Problem, MDVRP)與收送貨問題(Pickups and Deliveries Problem, PDP)進行說明，並介紹啟發式演算法之中的粒子群最佳化(Particle Swarm Optimization, PSO)的演算法。

首先在第一節回顧 MDPDP 之相關文獻，接著在第二節回顧 MDVRP 之相關文獻，在第三節回顧 PDP 之相關文獻，第四節說明本研究所使用之啟發式解法-粒子群最佳化之概念以及其相關應用與發展，最後第五節為文獻回顧小結。

2.1 MDPDP 相關文獻之回顧

本節主要針對 MDPDP 進行回顧，該問題特性已於第一章說明，由於大部分文獻是針對單一場站之收送貨為研究主題，僅 Ropke and Pisinger[15]之研究內容包含多場站收送貨，與本研究較相關，因此在本節回顧 MDPDP 時著重該篇文獻。

Ropke and Pisinger[15]針對 MDPDPTW 與 PDPTW 為研究課題，以 Large Neighborhood Search(LNS)演算法為基礎發展一新演算法(Adaptive Large Neighborhood Search, ALNS)，傳統 LNS 在搜尋解的過程中，解的插入(Insertion)與移出(Removal)僅各配合一種演算法，然而 ALNS 在解的插入與移出會使用數種演算法，最後透過 350 題以上 PDPTW 之標竿範例來評估其求解績效，測試結果發現有一半以上之例題透過該演算法所求得之解比過去文獻上之解較好。

此外，該文獻修改標竿範例產生出適合於 MDPDPTW 之例題，並使用傳統 LNS 以及 ALNS 來分別求解 MDPDPTW，發現其所提出之演算法(ALNS)不管在求解時間上或是求解品質，亦優於傳統 LNS。

2.2 多場站車輛路線問題(Multi-Depot Vehicle Routing Problem, MDVRP)

Dantzig 與 Ramser 於 1959 年率先提出 VRP(Vehicle Routing Problem)為一網路 $G=(V, A)$ ， V 為節點(Vertex)的集合， $V=\{0,1,2, \dots, n\}$ ，節點 0 代表場站， $1,2, \dots, n$ 為服務的顧客，共有 $n+1$ 個節點， A 為所有節線(Arc)集合，即 $A=\{(i,j) \in V, i \neq j\}$ 。在各節點需求為已知下，期望找出能滿足所有需求點且總成

本/總距離能最小的路徑組合，且需滿足以下情況：

- (1)每部車由場站出發，最終仍會回到原出發場站；
- (2)網路上所有顧客需求都需被滿足；
- (3)每位顧客只能由一部車輛進行一次服務；
- (4)每部車輛所服務之顧客總需求量不能超過車輛容量限制。

多場站車輛路線問題由VRP延伸而來，簡稱MDVRP(Multi-Depot Vehicle Routing Problem)。此類問題的定義如下：考慮多個場站，場站位置與數量已知，而場站沒有容量限制，每個場站皆擁有多部車輛，每部車輛有容量限制，僅考慮單一車種，每部車輛皆形成一條封閉式服務路線，需求點的數量、位置與需求量為已知，並為一對稱性網路，其需滿足之情況與單一場站之VRP相同。

MDVRP之數學模式與符號說明如下，是由Golden et al.於1983年所提出之模型：

n :所有節點集合，其中 $1\sim m$ 為場站， $m+1\sim n$ 為需求點。

m :場站數

NV :總車輛數

Q_k :車輛 k 的容量上限

T_k :車輛 k 的最長路線時間

d_i :需求點 i 的需求量

t_i :車輛服務需求 i 所需耗費的時間

t_{ij} :車輛由需求點 i 到需求點 j 行駛的時間

c_{ij} :車輛由需求點 i 到需求點 j 所需耗費的配送成本

x_{ij}^k :車輛 k 是否由需求點 i 行駛到需求點 j ，且 $i \neq j$ ，若是則為1，否為0

$S: \{(x_{ij}): y_i - y_j + N \times X_{ij} \leq N - 1, \text{ for } m + 1 \leq i \neq j \leq N, y_i \text{ and } y_j \in \mathbb{R}\}$ ，為避免發生子迴路限制式。

$$\text{Min } \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^{NV} c_{ij} x_{ij}^k \quad (1)$$

ST:

$$\sum_{i=1}^n \sum_{k=1}^{NV} x_{ij}^k = 1 \quad j = m + 1 \sim n \quad (2)$$

$$\sum_{j=1}^n \sum_{k=1}^{NV} x_{ij}^k = 1 \quad i = m + 1 \sim n \quad (3)$$

$$\sum_{i=1}^n x_{ip}^k - \sum_{j=1}^n x_{pj}^k = 0 \quad k = 1 \sim NV; p = m + 1 \sim n \quad (4)$$

$$\sum_{i=1}^n d_i (\sum_{j=1}^n x_{ij}^k) \leq Q_k \quad k = 1 \sim NV \quad (5)$$

$$\sum_{i=1}^n t_i \sum_{j=1}^n x_{ij}^k + \sum_{i=1}^n \sum_{j=1}^n t_{ij} x_{ij}^k \leq T_k \quad k = 1 \sim NV \quad (6)$$

$$\sum_{i=1}^m \sum_{j=m+1}^n x_{ij}^k \leq 1 \quad k = 1 \sim NV \quad (7)$$

$$\sum_{p=1}^m \sum_{i=m+1}^n x_{ip}^k \leq 1 \quad k = 1 \sim NV \quad (8)$$

$$X \in S \quad (9)$$

$$x_{ij}^k \in \{0,1\} \quad (10)$$

上述數學規劃式中，目標函數(1)為最小化配送成本；限制式(2)(3)表除場站以外的需求點只得由一部車輛服務一次；限制式(4)表除場站以外的需求點，進出該需求點的車輛必須為同一輛；(5)限制車輛所服務道之需求點的需求量不得超過車輛容量限制；(6)限制車輛的服務時間與行駛時間總和不得超過時間限制；限制式(7)(8)限制每部車輛最多被使用一次；限制式(9)為避免子迴圈之發生；限制式(10)為 0-1 變數限制。

關於求解 MDVRP 方面之相關文獻甚多，本研究舉出兩個為例，Renaud, et al.[14]使用禁制搜尋法(Tabu search)解決多場站車輛路線問題，透過 23 個標竿範例之測試，發現該啟發式演算法為當時所有啟發式演算法中求解之品質為最優。Thangiah et al. [21]使用基因演算法(Genetic Algorithms)去求解多場站車輛路線問題，並配合群集方法(clustering method)，在 23 個標竿範例之測試中，發現 11 個品質比原先更優之解。

2.3 收送貨問題(Pickups and Deliveries Problem, PDP)

收送貨問題是為一般化收送貨問題之一類，一般化收送貨問題簡稱 GPDP(General Pickups and Deliveries)，其目標函數為在不違反最大車輛行駛距離與最大容量限制之下，找到其最小化總路徑成本，從配送方式可分為兩種模式 [12]：車輛載貨從場站出發將貨物運送給送貨點顧客，再至收貨點顧客收取貨物運回場站，此種將貨物從場站至場站的配送方式為「回程取貨車輛途程問題 (Vehicle Routing Problems with Backhauls, VRPB)」。若車輛從場站出發至顧客指定地點收取貨物，將貨物送到顧客指定地點送貨，此種將貨物從顧客點直接配送至另一顧客點之配送方式為收送貨車輛途程問題 (Vehicle Routing Problems with Pickups and Deliveries, VRPPD)。

若從收送貨點是否配對，VRPPD 可再細分兩種模式 [12]，如圖 2-1 所示。若運送皆為相同商品，從一收貨點取貨後，可送至任一送貨點，此為收送貨點無配對之收送貨問題(Pickup and Delivery Vehicle Routing Problem, PDVRP)。另收送貨點有配對之收送貨問題，若運送為人則分類為撥召問題(Dial-A-Ride Problem, DARP)，若運送為物品則分類為有配對的收送貨問題簡稱收送貨問題 (Pickup and Delivery Problem, PDP)，此即為本研究之課題。

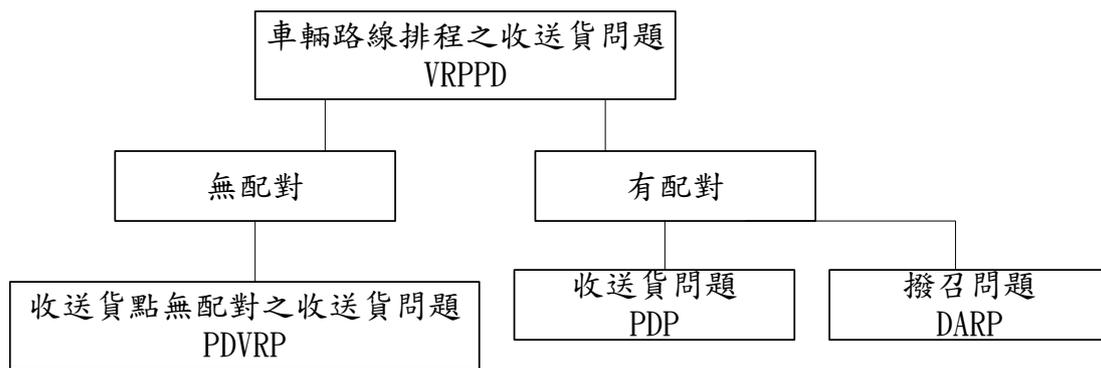


圖 2-1 收送貨有無配對之 VRPPD 分類[35]

已知 PDP 亦是 GDPD 的延伸問題，但有一些不同之限制，如優先限制 (precedence)：顧客需求(訂單)為一組收送貨作業，收貨與送貨作業可交錯服務，且同一組的收貨作業(Pickup)需在送貨作業(delivery)前完成；聯結限制(pairing)：每個貨運需求的收送貨作業均由同一輛車執行。

PDP之顧客需求為一組收送貨作業，不僅是收貨或送貨，且同一組的收貨作業需於送貨作業前完成服務，並由同一輛車服務。因此PDP問題可描述為：需求包含收貨作業與送貨作業兩部分，車輛由場站空車出發，在不違反車輛容量限制下服務一連串之需求，且同一需求之收貨作業必須在送貨作業前先完成，最後再空車回到場站，並尋求總旅行距離成本最小的車輛路徑組合。根據以上所述，PDP問題定義如下：

- (1)各車輛的起迄點須為同一場站。
- (2)每個顧客需求包含一個收貨作業與一個送貨作業，須由同一車輛進行服務，且只能服務一次。
- (3)一顧客需求指派給一車輛進行服務，則車輛的行駛路線必須包含該顧客需求的收貨作業與送貨作業，且收貨作業的服務順序必須於送貨作業前。
- (4)各車輛的載貨量無論何時皆不能超過車容量上限。
- (5)車輛於需求點完成後立即離開，並前往下一個需求點。
- (6)網路上所有節點必須被滿足。

本研究以[11]所提出之PDP數學模式為例，符號說明如下：

n : 訂單數

0 : 起點，代表所有車輛出發之地點。

$2n+1$: 迄點，代表所有車輛回去之地點。

N^+ : 收貨點作業之集合， $N^+ = \{1, \dots, n\}$

N^- : 送貨點作業之集合， $N^- = \{n+1, \dots, 2n\}$

N : 所有作業點的集合

q_i : 需求點 i 的貨物量。若 $i \in N^+$ ， q_i 為正數；若 $i \in N^-$ ， q_i 為負數；若

$$i = 2n + 1, q_i = 0$$

K : 所有車輛的集合

Q_k : 車輛 k 的容量上限

c_{ij}^k : $k \in K, i, j \in V, i \neq j$, 車輛 k 從作業點 i 到 j 的旅行成本

t_{ij}^k : $k \in K, i, j \in V, i \neq j$, 車輛 k 從作業點 i 到 j 的旅行時間

s_i : 作業點 i 所需的服務時間

V : 所有節點的集合

A : 所有節線的集合

PDP數學規劃模式中的決策變數如下：

x_{ij}^k : $k \in K, i, j \in V, i \neq j$, 車輛 k 從作業點 i 到作業點 j , $x_{ij}^k = 1$; 否則 $x_{ij}^k = 0$

L_i^k : 車輛 k 離開作業點 i 時累積的承載量

T_i^k : 車輛 k 開始服務作業點 i 的時間

PDP之定式如式(1)~(11)所示

$$\text{Min} \sum_{k \in K} \sum_{(i,j) \in A} c_{ij}^k x_{ij}^k \quad (11)$$

Subject to

$$\sum_{k \in K} \sum_{j: (i,j) \in A} x_{ij}^k = 1 \quad \forall i \in N \quad (12)$$

$$\sum_{j: (0,j) \in A} x_{0j}^k = 1 \quad \forall k \in K \quad (13)$$

$$\sum_{i: (i,2n+1) \in A} x_{i,2n+1}^k = 1 \quad \forall k \in K \quad (14)$$

$$\sum_{i: (i,j) \in A} x_{ij}^k - \sum_{i: (j,i) \in A} x_{ji}^k = 0 \quad \forall k \in K, j \in N \quad (15)$$

$$\sum_{j: (i,j) \in A} x_{ij}^k - \sum_{j: (n+i,j) \in A} x_{n+i,j}^k = 0 \quad \forall k \in K, i \in N \quad (16)$$

$$x_{ij}^k (L_i^k + q_i - L_j^k) = 0 \quad \forall k \in K, (i,j) \in A \quad (17)$$

$$T_k^i \leq T_{n+i}^k \quad \forall k \in K, \forall i \in N^+ \quad (18)$$

$$x_{ij}^k (T_i^k + s_i + t_{ij}^k - T_j^k) = 0 \quad \forall k \in K, (i,j) \in A \quad (19)$$

$$\max\{0, q_i\} \leq L_i^k \leq \min\{Q^k, Q^k + q_i\} \quad \forall k \in K, i \in V \quad (20)$$

$$x_{ij}^k \in \{0,1\} \quad \forall k \in K, (i,j) \in A \quad (21)$$

上述數學規劃式中，目標函數(11)為最小化總旅行成本；限制式(12)限制每一顧客需求(包含一收貨作業和一送貨作業)被同一車輛服務，且恰好一次；限制式(13)(14)表每輛車從場站出發，最後回到場站；限制式(15)確保流量守恆；(16)聯結限制，即同一組收送貨作業要同台車服務；(17)確保服務完此點的車容額等於上一點服務完的車容量加上要服務此點的需求量；(18)確保優先限制，即車輛服務一顧客需求先收貨再送貨；(19)任一項作業點完成後立即前往下一個作業點服務，不會出現等待情形；(20)確保車輛服務的過程中不超過車容量限制；(21)決策變數，若車輛從作業點 i 到需求點 j , $x_{ij}^k = 1$; 否則 $x_{ij}^k = 0$ 。

由於PDP文獻較少，接下來將針對各種不同之收送貨問題進行求解方法回顧，Catay[2]發展一新節省法運用於螞蟻演算法(ant colony algorithm)求解多場站之同時收送貨問題(Multi-depot Vehicle Routing Problem with Simultaneous Pickup and Delivery, MDVRPSPD)，同時收送貨問題亦為收送貨問題之一種，每個顧客需求皆需進行收貨與送貨之任務，與本研究之收送貨問題不同在於單一節點可同時進行收貨與送貨，而本研究之單一節點僅進行收貨或送貨之限制，故參考依據有限。其透過標竿範例測試，雖然對於大規模問題時，其求解速度較慢，但其解之品質不差，甚至優於標竿範例。

Bigang et al.[1]使用兩階段之混合啟發式解法求解收送貨問題，首先利用模擬退火法(simulated annealing Algorithm)減少路線數，接著再透過禁制搜尋法減少總旅行成本。其針對顧客訂單數為 100 到 600 之標竿範例進行測試，當顧客訂單數在 100、200 以及 600 下，不僅求解速度快，且部份解之品質優於標竿範例。

Goksal et al.[4]利用粒子群最佳化(Particle swarm Optimization)與變動鄰域搜尋法(variable neighborhood descent algorithm)求解同時收送貨問題，透過標竿範例測試，發現該混合演算法優於粒子群最佳化、螞蟻演算法以及基因演算法。

Wang et al.[26]針對多場站車輛路線之收送貨問題，在僅考慮其收送貨為回程取貨(backhauling)之限制下，發展一個結合基因演算法與禁制搜尋演算法(Tabu search)之啟發式解法，其目標式所求得之最短路徑優於其他啟發式解法。

Kachivichayanukul and Sombuntham[19]運用加入社會學習因子之粒子群最佳化(GLNPSO)，並在考慮時間窗的限制下，解決多場站收送貨問題。透過標竿範例之測試，發現當顧客需求有群集的現象時，求解之品質不差，且對於不同的編碼方式，解的品質也有所差異。

本研究將 MDPDP，以及多場站車輛路線問題與一般收送貨問題之求解方法的相關文獻，依其問題型態與求解方法整理如下表所示，可見得對於該類似問題，大多以啟發式演算法求解。

表 2-1 MDPDP 之求解方法相關文獻整理

	年份	作者	求解方法
MDPDP	2005	Ropke and Pisinger	ALNS
多場站車輛路線問題	1994	Renaud, Laporte and Boctor	禁制搜尋法
	2010	Thangiah and Salhi	基因演算法
一般化收送貨問題	2010	Catay	螞蟻演算法
	2009	Bigang and Changshi	混合啟發式解法
	2010	Goksal, Altiparmak and Karaoglan	混合啟發式解法

	2009	Wang ,Sun and Ren	混合啟發式解法
	2010	Kachivichayanukul and Sombuntham	粒子群最佳化

2.4 粒子群最佳化(Particle swarm Optimization, PSO)

粒子群最佳化由 Eberhart and Kennedy[6]提出，主要概念是源自於對動物群體行為之研究，以下針對粒子群最佳化的發展背景、運作法及相關發展應用來加以介紹說明：

2.4.1 粒子群最佳化之發展背景

在自然界中，許多生物的生活行為都會呈現群體的活動，因此在 90 年代時，有一股利用自然界生物的群體行為去做最佳化研究的潮流。例如 Dorigo[6]提出螞蟻演算法(Ant Colony Optimization, ACO)，在螞蟻演算法中每隻螞蟻皆會在其移動的路徑上留下費洛蒙，以供其他螞蟻藉著費洛蒙的濃度來判別哪一條路徑是離食物(目標值)最近的。

粒子群最佳化亦是由生物之群體活動所觀察而來的，其具有兩個主要的基本概念，其一概念是藉由觀察人類的決策過程(Boyd and Richerson, 1985)[6]，提出人類會使用兩個重要的資訊來做決策：一個是人們自己所擁有的經驗，根據自己的經驗去選擇哪個決策是目前為止較好的；另一個則是根據其他人所擁有的經驗來選擇到目前為止最好的決策，也就是每個個體在做決策時都會依照自己的經驗和其他人的經驗來決定。另一概念即是對自然界生物的群體行為提出一些簡單的法則來將其模組化，如水中的魚群和空中的鳥群等都可以使用這些模式來模組化。基於上述的兩個基本概念，Eberhart and Kennedy[6]提出粒子群體最佳化之概念。

2.4.2 粒子群最佳化演算法之說明

粒子群最佳化主要是基於模擬鳥群的行為所構想出來的，想像一群鳥隨機在某一區域內搜尋食物，而此一區域只有一個地方有食物，不過所有的鳥都不知道食物的確切位置，只知道目前距離食物還有多遠，此時最有效的辦法即是搜尋目前距離食物最近的鳥之鄰近區域。

因此在粒子群最佳化中，搜尋空間中的每一隻鳥都代表著一組解，在此稱這些鳥為「粒子」，每一個粒子的位置都以 θ_{ih} 表示，其中 i 代表第 i 個粒子， h 則代

表該粒子第 h 個維度，每個粒子對於每個維度的飛行速度則以 V_{ih} (第 i 個粒子之第 h 個維度之速度) 來表示。另外所有的粒子都會有一個對於問題的適應值 (fitness value)，每個粒子皆會知道自己目前的最佳適應值和最佳的位置，稱之為每個粒子目前各自最佳值 (Particle best value; P_{ih})，簡稱個體最佳值，這些資訊就像是每個粒子自己所擁有的經驗一樣。同時每個粒子也會知道在群體中到目前為止的最佳值與最佳的位置，稱之為群體最佳值 (Global best value; P_{gh})，這類資訊就像是其他粒子所擁有之經驗一樣。

每一階段的粒子皆是透過個體經驗與群體經驗之修正導引，使初始隨機散布於可行解區域的粒子能在運算過程中逐漸接近整體最佳目標值的粒子附近，形成粒子群在此一區域進行搜尋，並透過群體最佳值之效用而逐漸逼近最佳值。

粒子群最佳化之演算流程如圖 2-2 所示，Eberhart and Kennedy [6] 針對流程中之更新步驟提出每一階段之粒子的位置與速度更新法則，如下所示，粒子第 $(\tau + 1)$ 次運算階段之速度會依據各粒子目前第 τ 次運算階段之速度、與個體最佳解之距離及與群體最佳解之距離來更新，再依據更新之速度調整目前粒子之位置，更新每個粒子的搜尋方向與距離。

$$V_{ih}(\tau + 1) = V_{ih}(\tau) + c_p u (P_{ih} - \theta_{ih}(\tau)) + c_g u (P_{gh} - \theta_{ih}(\tau))$$

$$\theta_{ih}(\tau + 1) = \theta_{ih}(\tau) + V_{ih}(\tau + 1)$$

其中， i ：第 i 個粒子。

h ：第 h 個維度。

τ ：第 τ 次運算階段

P_{ih} ：第 i 個粒子的個體最佳解內，粒子在第 h 個維度的位置。

P_{gh} ：在群體最佳解內，粒子在第 h 個維度的位置。

$V_{ih}(\tau + 1)$ ：第 $(\tau + 1)$ 次的運算階段時，第 i 個粒子在第 h 個維度的速度。

$\theta_{ih}(\tau + 1)$ ：第 $(\tau + 1)$ 次運算階段時，第 i 個粒子在第 h 個維度的位置。

$V_{ih}(\tau)$ ：第 τ 次運算階段之第 i 個粒子在第 h 個維度的速度。

$\theta_{ih}(\tau)$ ：第 τ 次運算階段之第 i 個粒子在第 h 個維度的位置。

c_p 、 c_g ：學習因子。

u ：介於 0 與 1 之間的隨機亂數。

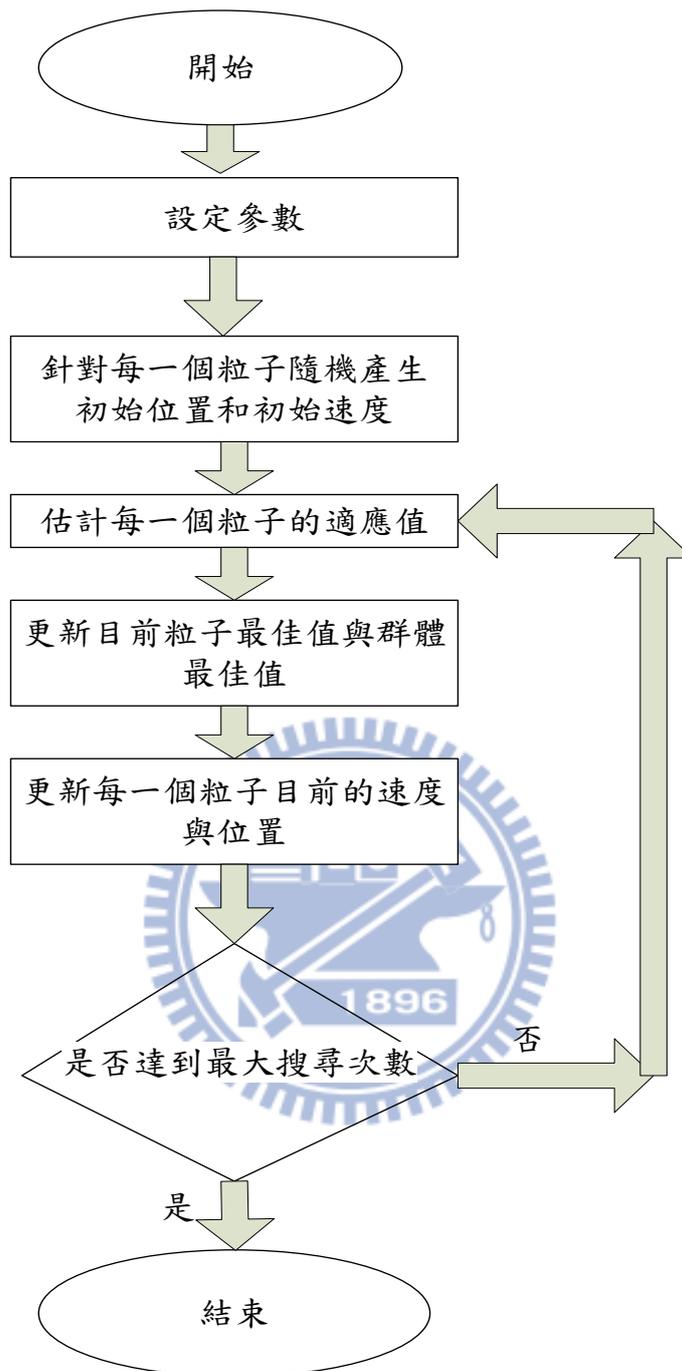


圖 2-2 粒子群最佳化之演算流程

對於粒子的位置與速度更新法則，Shi and Kennedy[31]提出慣性權重法 (Inertia Weight Method)，將慣性權重值(W)引入基本的更新法則內，藉由這個慣性權重值使粒子群在大範圍的搜尋之後，能降低粒子的速度，迫使粒子進入一個較好的區域搜尋，期望以更有效率的方法找到更好的可行解。除了慣性權重法，Shi and Kennedy[31]亦提出最大速度法(max V Method)以及收縮係數法 (Constriction Factor Method)等，如下所示。

(1) 慣性權重法：

$$V_{ih}(\tau + 1) = W V_{ih}(\tau) + c_p u(P_{ih} - \theta_{ih}(\tau)) + c_g u(P_{gh} - \theta_{ih}(\tau))$$

$$\theta_{ih}(\tau + 1) = \theta_{ih}(\tau) + V_{ih}(\tau + 1)$$

其中， W ：慣性權重值，一般為 0.9-0.4 之間。

Shi and Eberhart[31]認為慣性權重值可配合總運算次數(iterations)來調整，如下所示：

$$W(\tau) = W(T) + \frac{\tau - T}{1 - T} [W(1) - W(T)]$$

其中， $W(\tau)$ ：第 τ 階段之慣性權重值。

T ：總運算次數。

(2) 最大速度法：設一可忍受之最大速度(V_{max})與最小速度($-V_{max}$)，在不超過這個速度範圍內進行速度之更新。

$$V_{ih}(\tau + 1) = V_{ih}(\tau) + c_p u(P_{ih} - \theta_{ih}(\tau)) + c_g u(P_{gh} - \theta_{ih}(\tau))$$

$$\theta_{ih}(\tau + 1) = \theta_{ih}(\tau) + V_{ih}(\tau + 1)$$

If $V_{ih}(\tau + 1) > V_{max}$, $V_{ih}(\tau + 1) = V_{max}$

Else if $V_{ih}(\tau + 1) < -V_{max}$, $V_{ih}(\tau + 1) = -V_{max}$

其中， V_{max} ：設定的最大速度。

(3) 收縮係數法：每次之更新幅度隨著迭代數的增加而逐漸減少。

$$V_{ih}(\tau + 1) = K \times [V_{ih}(\tau) + c_p u(P_{ih} - \theta_{ih}(\tau)) + c_g u(P_{gh} - \theta_{ih}(\tau))]$$

$$\theta_{ih}(\tau + 1) = \theta_{ih}(\tau) + V_{ih}(\tau + 1)$$

$$K = \frac{2}{|2 - \mu - \sqrt{\mu^2 - 4\mu}|}$$

$$\mu = c_p + c_g , \mu > 4$$

粒子之速度更新法則是粒子群最佳化的概念之一，因此相關研究不間斷，Pongchairerks and Kachitvichyanukul[31]提出更新法則中的速度除了會受到個體最佳值與群體最佳值的影響外，亦會受到區域最佳值(Local best value; P_{ih}^L)與鄰近最佳值(Near neighbor value; P_{ih}^N)的影響，此粒子群最佳化簡稱為GLNPSO。區域最佳值乃是每個個體最佳值與其鄰近之個體最佳值中，擁有最佳適應值之粒子；而鄰近最佳值是指該階段的某粒子與其他粒子間擁有最大的FDR(fitness-distance ratio)者稱之。根據上述，更新法則修正成如下所示，多了

$c_l u(P_{ih}^L - \theta_{ih}(\tau))$ 與 $c_n u(P_{ih}^N - \theta_{ih}(\tau))$ ，在更新速度上較先前為精準，根據文獻

指出，該更新法則之求解績效較優。

$$V_{ih}(\tau + 1) = V_{ih}(\tau) + c_p u(P_{ih} - \theta_{ih}(\tau)) + c_g u(P_{gh} - \theta_{ih}(\tau)) \\ + c_l u(P_{ih}^L - \theta_{ih}(\tau)) + c_n u(P_{ih}^N - \theta_{ih}(\tau))$$

$$\theta_{ih}(\tau + 1) = \theta_{ih}(\tau) + V_{ih}(\tau + 1)$$

其中， c_l 、 c_n ：學習因子。

P_{ih}^L ：第 i 個粒子的區域最佳解內，粒子在第 h 個維度的位置。

P_{ih}^N ：第 i 個粒子的鄰近最佳解內，粒子在第 h 個維度的位置。

雖然粒子之速度與位置更新法則不斷提出，但大部份之文獻在使用粒子群最佳化時是以慣性權重法為更新法則[3][24][27][28][31][36]，因此本研究亦是使用慣性權重法為更新法則，並比較僅考慮個體最佳值與群體最佳值之一般 PSO 和加入考慮鄰近最佳值與區域最佳值之 GLNPSO 之求解績效。

2.5 小結

綜合上述，有以下結論：

1. 對於多場站車輛路線問題與收送貨問題該類問題，大都以啟發式演算法來求解。
2. 多場站車輛路線問題與收送貨問題皆已受到討論與延伸研究多年，相關數學式業已成熟，然而多場站車收送貨問題相關之研究甚少，因此所能引用之文獻有限。
3. 文獻上之一般化收送貨問題鮮少以收送貨點有配對之收送貨問題為對象，因此本研究針對受到較少注目之多場站收送貨點有配對之收送貨問題進行研究。
4. 粒子群最佳化為近十幾年來興起之啟發式解法，除了演算法本身之更新法則不斷逐年調整，粒子群最佳化之應用也日益廣泛[20]。由於粒子群最佳化尚未應用於本研究課題，因此本研究透過較新穎之求解方法來求解，期以獲得不錯的成效。

第三章 模式建立與演算法設計

粒子群最佳化主要是利用分散式的搜尋觀念與資訊的分享與導引來求解問題，從過去文獻上也發現此演算法對於大規模問題時，比其他一些演算法所求解之品質較好且運算時間上較有效率[17][30]。故本研究以粒子群最佳化為基礎，設計兩種演算法，並對於多場站收送貨問題產生在不違反車輛路徑限制與收送貨順序限制之情況下滿足顧客需求的解。

本章節先針對多場站之收送貨問題建構一新數學模式，設計兩種以粒子群最佳化(Particle Swarm Optimization)為基礎之演算法(一般 PSO 與 GLNPSO)，並配合一小範例進行試算。首先，3.1 節對 MDPDP 進行數學模式建構；接著 3.2 節針對本研究之課題所運用之粒子群集法之流程進行介紹，3.3 節則協以範例說明。

3.1 多場站收送貨問題之數學模式

根據過去文獻，本研究參考 MDVRP 與 PDP 之數學模式，研擬出多場站收送貨問題之數學模式，符號與限制式說明如下：

3.1.1 符號說明

數學模式符號說明：

m : 節點總個數，其中 $1 \sim 2n$ 為需求點， $2n+1 \sim m$ 為場站

n : 訂單數

NV : 總車輛數

N^+ : 收貨點作業之集合， $N^+ = \{1, \dots, n\}$

N^- : 送貨點作業之集合， $N^- = \{n+1, \dots, 2n\}$

N : 所有需求點的集合

q_i : 需求點 i 的貨物量。若 $i \in N^+$ ， q_i 為正數；若 $i \in N^-$ ， q_i 為負數；

若 $i = 2n+1 \sim m$ ， $q_i = 0$

Q_k : 車輛 k 的容量上限

P_k : 場站 P 可使用之車輛數目

c_{ij}^k : $k \in K$ ， $i, j \in V$ ， $i \neq j$ 車輛 k 從需求點 i 到 j 的旅行成本

y_i : 每一迴路中被車輛服務的節點編號 i

決策變數：

x_{ij}^k : $i \neq j$ ，若車輛 k 從節點 i 到節點 j ， $x_{ij}^k = 1$ ；否則 $x_{ij}^k = 0$

L_i^k : 車輛 k 離開節點 i 時累積的承載量

T_i^k : 車輛 k 服務作業點 i 的順序編號

3.1.2 數學模式

多場站之收送貨問題建構模式如式(1)~(13)所示

$$\text{Min } \sum_{i=1}^m \sum_{j=1}^m \sum_{k=1}^{NV} c_{ij}^k x_{ij}^k \quad (22)$$

ST:

$$\sum_{i=1}^m \sum_{k=1}^{NV} x_{ij}^k = 1 \quad j = 1 \sim 2n \quad (23)$$

$$\sum_{j=1}^m \sum_{k=1}^{NV} x_{ij}^k = 1 \quad i = 1 \sim 2n \quad (24)$$

$$\sum_{i=1}^m x_{ip}^k - \sum_{j=1}^m x_{pj}^k = 0 \quad k = 1 \sim NV; p = 1 \sim 2n \quad (25)$$

$$\sum_{k=1}^{NV} \sum_{i=1}^n \sum_{j=2n+1}^m x_{ij}^k = 0 \quad (26)$$

$$\sum_{k=1}^{NV} \sum_{i=2n+1}^m \sum_{j=n+1}^{2n} x_{ij}^k = 0 \quad (27)$$

$$\sum_{i=2n+1}^m \sum_{j=1}^n x_{ij}^k \leq 1 \quad k = 1 \sim NV \quad (28)$$

$$\sum_{i=n+1}^{2n} \sum_{j=2n+1}^m x_{ij}^k \leq 1 \quad k = 1 \sim NV \quad (29)$$

$$\sum_{j=1}^n x_{ij}^k - \sum_{j=n+1}^{2n} x_{ji}^k = 0 \quad k = 1 \sim NV; i = 2n + 1 \sim m \quad (30)$$

$$\sum_{j=1}^m x_{ij}^k - \sum_{j=1}^m x_{n+i,j}^k = 0 \quad k = 1 \sim NV; i = 1 \sim n \quad (31)$$

$$x_{ij}^k (L_i^k + q_j - L_j^k) = 0 \quad k = 1 \sim NV; i, j = 1 \sim m \quad (32)$$

$$T_i^k \leq T_{n+i}^k \quad \forall i \in N^+; k = 1 \sim NV \quad (33)$$

$$\sum_{j=1}^m X_{ij}^k \leq T_i^k \leq 2n \sum_{j=1}^m X_{ij}^k \quad k = 1 \sim NV; i = 1 \sim 2n \quad (34)$$

$$x_{ij}^k (T_i^k + 1 - T_j^k) = 0 \quad k = 1 \sim NV; i, j = 1 \sim 2n \quad (35)$$

$$\max\{0, q_i\} \leq L_i^k \leq \min\{Q^k, Q^k + q_i\} \quad k = 1 \sim NV; i = 1 \sim m \quad (36)$$

$$\sum_{k=1}^{NV} \sum_{j=1}^n x_{ij}^k \leq P_k \quad i = 2n + 1 \sim m \quad (37)$$

$$y_i - y_j + m \times x_{ij}^k \leq m - 1 \quad 1 \leq i \neq j \leq 2n; \forall y_i, y_j \in \mathbb{R} \quad (38)$$

$$x_{ij}^k \in \{0,1\} \quad k = 1 \sim NV; i, j = 1 \sim m; i \neq j \quad (39)$$

上述數學規劃式中，目標函數(22)為最小化總旅行成本；限制式(23)(24)限制除場站外的需求點只得由一部車輛服務一次；限制式(25)確保流量守恆；限制式(26)(27)表任何一部車輛皆不會從收貨點直接回到場站或是從場站直接出發至送貨點；限制式(28)(29)表每輛車從場站出發，最後回到場站；限制式(30)表車輛會回到與起點相同的場站；限制式(31)聯結限制，即同一組收送貨作業要同台車服務；限制式(32)確保服務完此點的車容額等於上一點服務完的車容量加上要服務此點的需求量；限制式(33)確保優先限制，即同一顧客需求需先收貨再送貨；限制式(34)表車輛服務作業點之順序編號之範圍，假若所有訂單由同一輛車來服務，則根據限制式(35)可知最大之順序編號為 $2n$ ；限制式(35)是用來配合限制式(34)，其表示車輛服務作業點之順序編號差距為 1；限制式(36)確保車輛服務的過程中不超過車容量限制；限制式(37)為各場站可使用車輛數目之限制；限制式(38)避免子迴路發生的限制式；限制式(39)決策變數，若車輛從需求點 i

到需求點 j , $x_{ij}^k = 1$; 否則 $x_{ij}^k = 0$ 。

3.2 粒子群集最佳化演算法

3.2.1 演算法之架構說明

下圖為一般 PSO 與 GLNPSO 應用於多場站之收送貨問題之方法流程，差別僅在於更新過程，接下來將針對該流程進行詳細的說明。

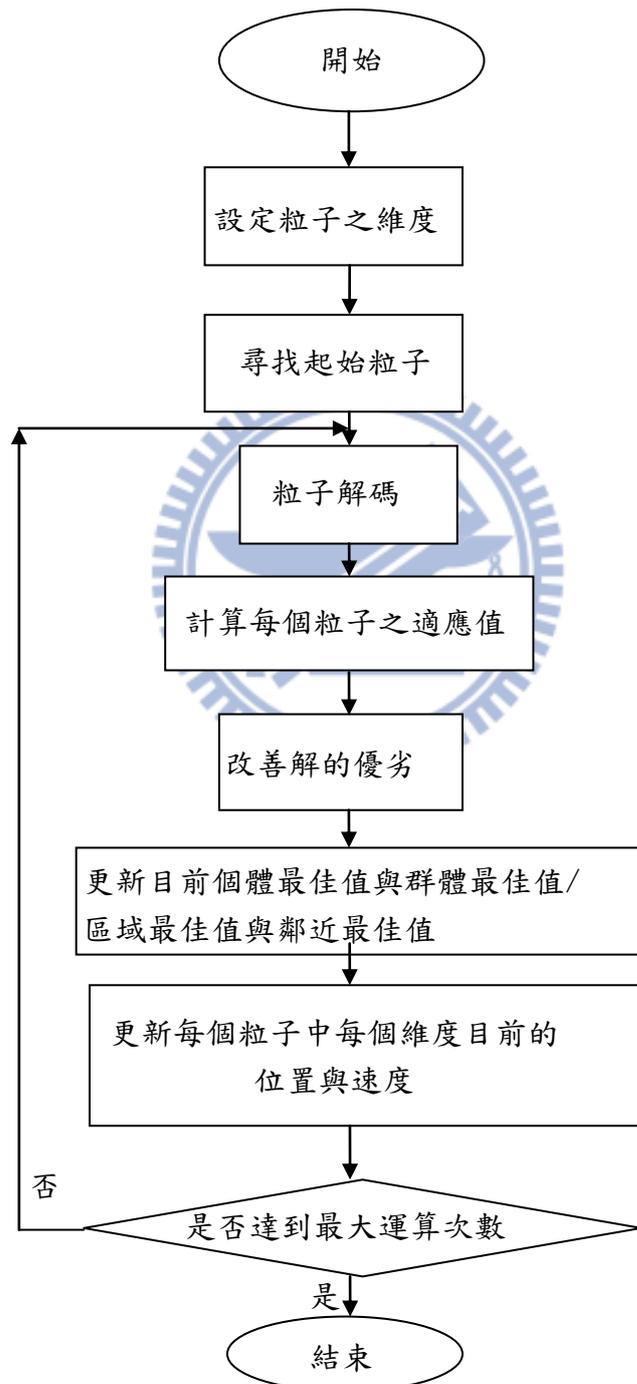


圖 3-1 演算法流程圖

1. 設定粒子之維度

對於演算法 PSO 而言，如何透過粒子表達一個解是重要的關鍵之一，一個粒子的維度代表一個解的長度，由於一個粒子代表著一個解，因此粒子與解之間需要一些轉換的方式，這兩者之間的轉換稱之為編碼(解→粒子)與解碼(粒子→解)。而維度的長度越長，需要的轉換時間會越多，則整個演算法的運算時間都會越長，因此設定一個適當的粒子維度，以及具有解的代表性非常重要，而本研究對於 MDPDP 僅根據顧客訂單數來決定粒子之維度，以避免維度之過長。

根據顧客訂單數，來決定粒子之維度，假設目前有 n 個顧客訂單數，則粒子的維度為 n 維，而每個粒子都代表著一個解。

2. 尋找起始粒子

決定好粒子的維度後，接下來即是找尋起始粒子，也就是起始解。本研究起始粒子之每個維度之數值皆是介於 0~總車輛數之間之亂數所構成，整數部分表示該訂單被服務之車輛，亦為車輛編號，當整數部分相同時，小數部分之大小排序則表示訂單被該車輛服務之優先順序。

3. 粒子解碼

由於從粒子本身較難直接看出解的情況，因此需透過解碼的方式得知，解碼過程如下：

- (1) 根據整數值進行分組，相同者為一組，可分成 $\{0\}, \{1\} \cdots \{NV-2\}, \{NV-1\}, \{NV\}$ 組集合。
- (2) 對於各組之數值，根據小數值之大小進行排列，形成顧客訂單之服務順序。針對各組集合取一最小值放置最前面，再針對各組集合取一次小值置於第二位，以此類推，直至所有組合內之數值皆排序完成。
- (3) 將顧客訂單一一分解成收、送貨兩點(送貨點之編號本研究設定為收貨點+顧客訂單數)。若顧客訂單編號為 X ，則其收-送貨編號為 $X-(X+n)$ 。
- (4) 對於整數值所代表之車輛編號，找出其對應之場站。
- (5) 建構車輛路線，對於各組集合皆可得到：場站- $X_1-(X_1+n)$ - $X_2-(X_2+n)$ -...- $X_n-(X_n+n)$ -場站之路線。

假設六個顧客訂單、兩個場站之 MDPDP 問題，每個場站所擁有的車輛分別為 3 台，某起始粒子如下：

顧客訂單	1	2	3	4	5	6
粒子之每個維度	2.3	3.5	5.8	2.6	5.1	5.4

A 場站所擁有之車輛編號為：0、1、2

B 場站所擁有之車輛編號為：3、4、5

按照上述解碼方式，根據整數值進行分組，得到以下三組：
 (2.3, 2.6), (3.5), (5.8, 5.1, 5.4)，然後在每個分組內，按照小數部分由小到大排列，得到以下結果：(2.3, 2.6), (3.5), (5.1, 5.4, 5.8)，將該數值映射到對應之顧客訂單，並分解成收、送貨，本研究設定之收貨編號等於顧客訂單之編號，送貨編號為收貨編號加總顧客訂單數，因此對應到各顧客訂單之(收-送)為(1-7, 4-10), (2-8), (5-11, 6-12, 3-6)，以及找出整數值(車輛編號)所對應之場站，車輛編號為 2，對應之場站為 A；車輛編號為 3 和 5，對應之場站為 B，即可得到所對應之路徑如下：

第一條路徑：A-1-7-4-10-A

第二條路徑：B-2-8-B

第三條路徑：B-5-11-6-12-3-9-B

由於粒子的維度與顧客的訂單數相當，解碼時只需進行一次排序與取整運算，在更新粒子與面對大規模問題時，勢必可節省運算時間。

4. 計算每個粒子之適應值

對於所得到之初始路線，將其代入目標式($\sum_{i=1}^m \sum_{j=1}^m \sum_{k=1}^{NV} c_{ij}^k x_{ij}^k$)計算其適應值，本研究之目標式是以成本最低為目的，則運算後所得到的適應值越低越好。

5. 改善解的品質

對解碼後所得到的解，在不違反限制式的條件下針對每一條路線個別使用 2-opt 進行路線內交換，透過這種微調的方式，對所得的解進行改善。

6. 更新目前個體最佳值與群體最佳值/區域最佳值與鄰近最佳值

個體最佳值：將每次運算階段的解帶入目標式中計算適應值，目標式的結果是越小越好，則適應值的結果也是越小越趨近最佳解。而每階段後每個粒子會判斷此次階段的適應值是否比本身之前所有階段的適應值小，如果比先前更小則此階段粒子之適應值會更新為粒子之個體最佳值，而每個粒子皆有一個個體最佳

值。

群體最佳值：群體最佳值與粒子最佳值之概念相同，主要的差別在於粒子最佳值是粒子與粒子自己本身的比較，而群體最佳解是粒子與其他所有粒子的比較，也就是說群體最佳值是至目前為止所有粒子的最佳值，因此只會有一個群體最佳值。

區域最佳值：乃指數個粒子中最好之個體最佳值，本研究根據過去文獻[19]，以每 5 個粒子為一個區域，找出每區域中較佳之個體最佳值。

鄰近最佳值：對於每個粒子，找出與其他粒子間擁有最大之 FDR 之粒子則稱為鄰近最佳值，因此每個粒子在每次迭代皆會有一個鄰近最佳值。

7. 更新每個粒子中每個維度目前的位置與速度

本研究對於各粒子所設定之初始速度為 0，初始位置即是起始粒子之數值，往後在進行粒子之更新時，必須也進行更新位置與速度，更新方式如下：

一般 PSO：

$$V_{ih}(\tau + 1) = W(\tau)V_{ih}(\tau) + c_p u(P_{ih} - \theta_{ih}(\tau)) + c_g u(P_{gh} - \theta_{ih}(\tau))$$

$$\theta_{ih}(\tau + 1) = \theta_{ih}(\tau) + V_{ih}(\tau + 1)$$

GLNPSO：

$$V_{ih}(\tau + 1) = W(\tau)V_{ih}(\tau) + c_p u(P_{ih} - \theta_{ih}(\tau)) + c_g u(P_{gh} - \theta_{ih}(\tau))$$

$$+ c_l u(P_{ih}^L - \theta_{ih}(\tau)) + c_n u(P_{ih}^N - \theta_{ih}(\tau))$$

$$\theta_{ih}(\tau + 1) = \theta_{ih}(\tau) + V_{ih}(\tau + 1)$$

If $\theta_{ih}(\tau + 1) > \theta^{max}$, then

$$\theta_{ih}(\tau + 1) = \theta^{max}$$

$$V_{ih}(\tau + 1) = 0$$

If $\theta_{ih}(\tau + 1) < \theta^{min}$, then

$$\theta_{ih}(\tau + 1) = \theta^{min}$$

$$V_{ih}(\tau + 1) = 0$$

其中

τ ：第 τ 次運算階段

$V_{ih}(\tau + 1)$ ：第 $\tau + 1$ 次運算階段時，第 i 個粒子的第 h 個維度之速度

$V_{ih}(\tau)$ ：第 τ 次運算階段時，第 i 個粒子的第 h 個維度之速度

$w(\tau)$ ：第 τ 次運算階段時之惰性權重值

$\theta_{ih}(\tau + 1)$ ：第 $\tau + 1$ 次運算階段時，第 i 個粒子的第 h 個維度之位置

$\theta_{ih}(\tau)$ ：第 τ 次運算階段時，第 i 個粒子的第 h 個維度之位置

P_{ih} ：第 i 個個體最佳值的第 h 個維度

P_{gh} ：到目前為止之群體最佳值的第 h 個維度

P_{ih}^L ：第 i 個粒子的區域最佳解內，粒子在第 h 個維度的位置。

P_{ih}^N ：第 i 個粒子的鄰近最佳解內，粒子在第 h 個維度的位置。

c_p ：個體最佳值之常數

c_g ：群體最佳值之常數

c_l ：區域最佳值之常數

c_n ：鄰近最佳值之常數

u ：介於 0 與 1 之間的隨機亂數

θ^{\max} 與 θ^{\min} ：為粒子之移動範圍

8. 是否達到最大運算次數

判斷搜尋次數是否已達到所設定之最大搜尋次數，如果是則結束搜尋，否則回到步驟「粒子解碼」持續進行演算至達到最大搜尋次數。

3.3 範例說明

本範例設定有三個訂單數，每個顧客訂單之收貨量分別為 3、4、4；送貨量為 -3、-4、-4；場站數為二，每個場站各擁有兩部車，每部車之容量皆為 8。在 PSO 參數之設定上，假設最大運算次數(T)為 10、惰性權重(W)最初由 0.9 至最終為 0.4、隨機亂數(u)為 0.7、粒子最佳值(c_p)之常數為 0.8、群體最佳值之常數(c_g)為 0.7，初始速度(ω)為 0，本研究以兩個粒子作為範例進行運算。

步驟一：設定粒子之維度

顧客訂單數為粒子之維度，本範例顧客訂單數為三，因此每個粒子之維度亦為三，如下所示：

--	--	--

步驟二：車輛編號設定

每個場站擁有兩部車，因此車輛編號 0、1 為 A 場站，車輛編號 2、3 為 B 場站。

步驟三：尋找起始粒子

本研究是透過隨機方式而得，兩個粒子各三個維度皆產生 0 到 4(總車輛數)間之數值，假設所得之結果如下：

1.6	1.5	3.4
2.4	2.1	2.9

步驟四：粒子解碼

根據前小節粒子解碼之方式，我們可將第一個粒子排序成(1.5, 1.6)、(3.4)，第二個粒子則排序成(2.1, 2.4, 2.9)，接著將每個顧客訂單分解成收、送貨以及找出其整數值所對應之場站，可得到初始路線，其結果如下：

第一個粒子:A-2-5-1-4-A 與 B-3-6-B 兩條路線

第二個粒子:B-2-5-1-4-3-6-B 一條路線

步驟五：計算每個粒子之適應值

下表為收、送貨與場站間之成本表(列為起點，行為迄點)，可由此表計算出各粒子之適應值，可知第一個粒子之適應值為 $(3+20+14+5+15)+(2+6+2)=67$ ；第二個粒子之適應值為 $1+20+14+5+20+6+2=68$ 。

表 3-1 小範例之成本表

	1	2	3	4	5	6	A	B
1	0	10	12	5	11	20	16	20
2	1	0	10	12	20	20	30	30
3	12	30	0	21	20	6	40	20
4	12	20	20	0	1	2	15	30
5	14	12	40	10	0	5	2	13
6	20	20	12	20	20	0	14	2
A	12	3	14	45	12	40	0	50
B	20	1	2	30	20	20	50	0

步驟六：改善解的品質

本研究所使用之改善法為路線內之 2-opt，以第一個粒子為例，其第一條路線經過交換可以發現，A-2-1-5-4-A 較 A-2-5-1-4-A 為優，因為其適應值 $(3+1+11+10+15=40)$ 小於原先之 $(3+20+14+5+15=57)$ ，故接受此交換，以此類推第二個粒子 B-2-1-5-4-3-6-B 較 B-2-5-1-4-3-6-B 為優，因其適應值 $(1+1+11+10+20+6+2=51)$ 小於原先之 68，故亦交換。

步驟七：更新目前個體最佳值與群體最佳值(以一般 PSO 為例)

在 T(運算次數)=1 時，各粒子之適應值皆為各粒子之個體最佳值，亦即第一個粒子之個體最佳值為 $40+(2+6+2)=50$ ，第二個粒子之個體最佳值為 51，而群體最佳值則為第一個粒子之適應值 50。

步驟八：更新每個粒子中每個維度目前的位置與速度

在進行 T=2 之前，必須先更新粒子之位置與速度。根據前一小節所述之方法，本範例粒子之速度 ω 與位置 θ 更新如下：

$$\omega_{11}(2) = 0.9 * 0 + 0.8 * 0.7(1.6 - 1.6) + 0.7 * 0.7(1.6 - 1.6) = 0$$

$$\theta_{11}(2) = 1.6 + 0 = 1.6$$

$$\omega_{12}(2) = 0.9 * 0 + 0.8 * 0.7(1.5 - 1.5) + 0.7 * 0.7(1.5 - 1.5) = 0$$

$$\theta_{12}(2) = 1.5 + 0 = 1.5$$

$$\omega_{13}(2) = 0.9 * 0 + 0.8 * 0.7(3.4 - 3.4) + 0.7 * 0.7(3.4 - 3.4) = 0$$

$$\theta_{13}(2) = 3.4 + 0 = 3.4$$

$$\omega_{21}(2) = 0.9 * 0 + 0.8 * 0.7(2.4 - 2.4) + 0.7 * 0.7(1.6 - 2.4) = -0.392$$

$$\theta_{21}(2) = 2.4 - 0.392 = 2.008$$

$$\omega_{22}(2) = 0.9 * 0 + 0.8 * 0.7(2.1 - 2.1) + 0.7 * 0.7(1.5 - 2.1) = -0.294$$

$$\theta_{22}(2) = 2.1 - 0.294 = 1.806$$

$$\omega_{23}(2) = 0.9 * 0 + 0.8 * 0.7(2.9 - 2.9) + 0.7 * 0.7(3.4 - 2.9) = 0.245$$

$$\theta_{23}(2) = 2.9 + 0.245 = 3.145$$

由上得知，T=2 時，粒子之速度更新為：

0	0	0
0	0	0

0	0	0
-0.392	-0.294	0.245

粒子之位置之更新如下所示，可以發現第二個粒子在最初是由同一輛車所服務，經過交換改善法與 PSO 運算，三個顧客訂單轉變為皆由不同車輛所服務。此後再根據新的粒子之位置進行解碼等重複步驟，直到達到最大運算次數。

1.6	1.5	3.4
2.4	2.1	2.9

1.6	1.5	3.4
2.008	1.806	3.145

第四章、範例測試

此章節主要為驗證本研究所使用之粒子群最佳化應用於MDPDP之求解效果，共可分為三個小節來說明。首先在第一節介紹用來驗證求解績效之測試範例來源及問題特色，接著在第二節決定適合於本研究課題之粒子群最佳化之參數，最後於第三節進行範例之測試，並使用套裝軟體求解小範例之測試，與本研究之演算法進行比較，來驗證本演算法之求解績效，再比較一般 PSO 與 GLNPSO 之求解績效。

本研究使用 Microsoft Visual Studio 編譯器以 vb.net 程式語言撰寫演算法，另外利用套裝軟體 CPLEX 9.0 所提供之求解功能求解 MDPDP 之數學模式，於 2.00GB RAM、Windows XP Professional SP2 作業平台的個人電腦上執行測試。

4.1 測試範例

從文獻回顧已知本研究之 MDPDP 相關文獻甚少，與本研究課題較相近之測試範例僅有 Li and Lim 之標竿例題以及 Ropke and Pisinger[15]之例題。由於 Li and Lim 之標竿例題不考慮多場站且具有時間窗限制，若要引用來測試本研究課題實有實行性上之難度，因此本研究以 Ropke and Pisinger[15]針對 MDPDPTW 之問題所產生之範例作為測試範例之依據，並進一步修改成適合於本研究之測試範例。

Ropke and Pisinger[15]以 Solomon VRPTW 之標竿範例為基礎，產生適用於 MDPDPTW 之範例例題，該問題特性除了為多場站並具有時間窗外，另外會依據(1)路線型態、(2)顧客型態與(3)地理分佈來分類，茲說明如下：

- (1) 路線型態：路線可分為起迄點相同與起迄點不同。
- (2) 顧客型態：其將顧客型態分為兩類，一類為一般顧客，皆可由任何車輛來服務；另一類為特殊顧客，該類顧客只能被固定之車輛來服務。
- (3) 地理分佈：節點可分為均勻分佈、聚集分佈以及半聚集分佈。聚集分佈為將所有節點分成十個群聚處，半聚集乃為 50%的節點數為均勻分佈；另 50%則分佈於十個群聚處。

本研究之 MDPDP 不考慮時間窗限制以及特殊顧客之服務限制，因此將關於該兩項條件之資料刪除，如表 4-1，該表為從 Ropke and Pisinger[15] 50 個顧客訂單之範例所擷取的部分資料，本研究將(起始時間窗)、(結束時間窗)、(服務時間)與(特定車輛服務)之欄位皆刪除，另外(前一服務點)與(後一服務點)之兩欄位主要是針對收送貨問題中同一顧客訂單之收送貨必須由同輛車來服務之限制，本研究已在求解方法中考量此點，因此亦一併刪除，成為適用於本研究之例題，共 48

題。有鑒於台灣物流業屬於小規模公司，服務之顧客訂數鮮少一次多於 250 個顧客訂單，因此本研究在不考慮 500 個顧客訂單之範例測試下，測試範例共 36 題，並由 50、100、250 個顧客訂單所構成，意即每種顧客訂單數為 12 題，每 12 題中，均勻分佈、半聚集分佈以及聚集分佈各佔 4 題。

表 4-1 50 個顧客訂單範例之部分資料

節點編號	X 軸座標	Y 軸座標	需求	起始時間窗	結束時間窗	服務時間	前一服務點	後一服務點	特定車輛服務
0	538	102	15	2406	2945	180	-1	1	-1
1	755	194	-15	3537	4283	153	0	-1	
2	183	645	12	1145	2199	175	-1	3	-1
3	52	542	-12	2504	3299	145	2	-1	

註 1：在(前一服務點)之欄位，若為-1，則表該節點為收貨點。

註 2：在(後一服務點)之欄位，若為-1，則表該節點為送貨點。

註 3：在(特定車輛服務)之欄位，若為-1，則表該節點為所有車輛皆可服務，若無值，表該點為送貨點。

4.2 參數測試

本研究演算法內之參數有慣性權重(W)、學習因子(c_p 、 c_g)、粒子數與迭代數均會影響解之品質，而 4.2 節之目的為測試不同組合之參數，以找到最適合於本研究之參數值。首先根據過去之文獻[7][18][31]，提出一組績效不錯之參數組(粒子數, 迭代數, 慣性權重, c_p , c_g) = (30, 500, 0.9-0.4, 2, 2)，再以此組參數為基礎，針對各項參數進行更深入之測試，延伸出各種不同的參數組合。舉例來說，本研究在測試粒子數之多寡時，將固定其他參數(迭代數、慣性權重、 c_p 與 c_g)之值，針對不同的粒子數提出不同之參數組合，如表 4-2 所示。

表 4-2 測試粒子數多寡之參數組

	組合 1	組合 2	組合 3	組合 4	組合 5
粒子數	30	10	20	40	50
迭代數	500	500	500	500	500
c_p	2	2	2	2	2
c_g	2	2	2	2	2
慣性權重	0.9- 0.4	0.9- 0.4	0.9-0.4	0.9-0.4	0.9-0.4

在測試之範例上，本研究針對顧客訂單數為 50 與 100 進行參數測試，不同顧客訂單數之三種地理分佈型態各取一範例做測試，亦即共六題測試參數之範例，以找出優良求解品質之參數組合。在測試參數之過程中，針對各組參數組合測試十次，以十次測試中最好的一次為該參數組合之最佳目標值，本研究並以各參數組合所求得的最小目標值與配合各參數組合所得的最小目標值相比，以最佳目標值誤差百分比表示；以各參數組合所求得的最小目標值與配合各參數組合所得的最小目標值相比，以平均目標值誤差百分比表示，如表 4-3 所示，其目的在評估各組合在不同範例下的實證績效，最佳目標值誤差百分比與平均目標值誤差百分比越小表示該參數組合在該網路求解的效果越佳。

表 4-3 最佳目標值誤差百分比與平均目標值誤差百分比之示意表

	組合 1	組合 2	……	組合 10	組合 11	所有組合 下之最佳 目標值
目標值 1	27916.12	26464.4	……	25663.95	27334.82	22843.19
目標值 2	29355.68	24140.67	……	26707.15	24366.37	
目標值 3	30440.07	31389.84	……	27274.97	27370.75	
目標值 4	29960.98	26924.73	……	22843.19	27634.16	
目標值 5	27334.06	25545.83	……	28358.44	23685.3	
目標值 6	29004.18	29719.12	……	26238.01	25738.07	
目標值 7	27642.49	25887.74	……	24366.58	28896.43	
目標值 8	28395.63	26765.84	……	24574.72	23332.36	
目標值 9	27510.95	30445.72	……	23205.87	27568.9	
目標值 10	28398.95	31913.32	……	24250.2	28798.19	
最佳目標值	27334.06	24140.67	……	22843.19	23332.36	
平均目標值	28595.91	27919.72	……	25348.31	26472.53	
最佳目標值誤差百分比	19.66%	5.68%	……	0.00%	2.14%	
平均目標值誤差百分比	25.18%	22.22%	……	10.97%	15.89%	

差百分比						
------	--	--	--	--	--	--

註：最佳目標值誤差百分比=

$$\frac{\text{各組合之最佳目標值} - \text{所有組合下最佳目標值}(22843.19)}{\text{所有組合下最佳目標值}(22843.19)} \times 100\%$$

平均目標值誤差百分比=

$$\frac{\text{各組合之平均目標值} - \text{所有組合下最佳目標值}(22843.19)}{\text{所有組合下最佳目標值}(22843.19)} \times 100\%$$

接下來分別針對粒子群個數、演算法迭代數、慣性權重值、學習因子四個小節進行參數測試。在兩種顧客訂單數、三種地理分佈，共6種之測試後，其結果過於繁雜無法一同討論分析，因此分段討論於4.2.1節至4.2.4節中。另外，在各小節測試結果的列表中，若測試範例一欄內僅以顧客訂單數表示，則表示各測試範例的誤差百分比為在三種不同地理分佈情況下測試結果的平均值，以此方式突顯各參數組合在不同顧客訂單數下其求解結果的差異性。

4.2.1 粒子數

粒子數即代表每迭代數中的可行解數量，在許多粒子群最佳化之相關文獻中，對於其研究問題均探討粒子數多寡對於求解時間與效果的相關性，並提出適合的粒子數。根據過去之文獻[7]，發現粒子數在10至50時，其求解結果令人滿意，因此本研究於此小節提出粒子數量分別為10、20、30、40、50之五組參數組進行測試，如4.2節之表4-2，以驗證粒子數多寡對於求解時間與結果之影響。並配合顧客訂單數為50與100之範例例題與節點分佈為均勻分佈、半聚集分佈、聚集分佈之三種地理分佈，共六種規模進行測試。求解結果與求解時間列於表4-4與表4-5，詳細的求解結果與求解時間附於書後附錄二。

表 4-4 測試粒子數最佳目標值誤差百分比與求解時間

	組合	組合 2	組合 3	組合 1	組合 4	組合 5
顧客 訂單 數	測試之粒子數	10	20	30	40	50
50	誤差百分比	22.23%	13.11%	16.00%	10.71%	10.70%
	求解時間	60.57	126.02	232.75	257.28	336.44
100	誤差百分比	29.47%	19.36%	23.53%	13.49%	15.31%

	求解時間	129.25	500.50	605.50	969.93	1433.13
	平均誤差百分比	25.85%	16.23%	19.77%	12.10%	13.01%
	平均求解時間 (秒)	94.91	313.26	419.13	613.60	884.78

表 4-5 測試粒子數平均目標值誤差百分比與求解時間

	組合	組合 2	組合 3	組合 1	組合 4	組合 5
顧客訂 單數	測試之粒子 數	10	20	30	40	50
50	誤差百分比	37.45%	32.43%	27.51%	22.69%	25.41%
	求解時間	57.67	123.16	183.68	255.37	300.80
100	誤差百分比	39.06%	34.94%	32.17%	30.54%	28.49%
	求解時間	157.13	352.09	487.12	764.15	996.53
	平均誤差百 分比	38.26%	33.68%	29.84%	26.62%	26.95%
	平均求解時 間(秒)	107.40	237.63	335.40	509.76	648.67

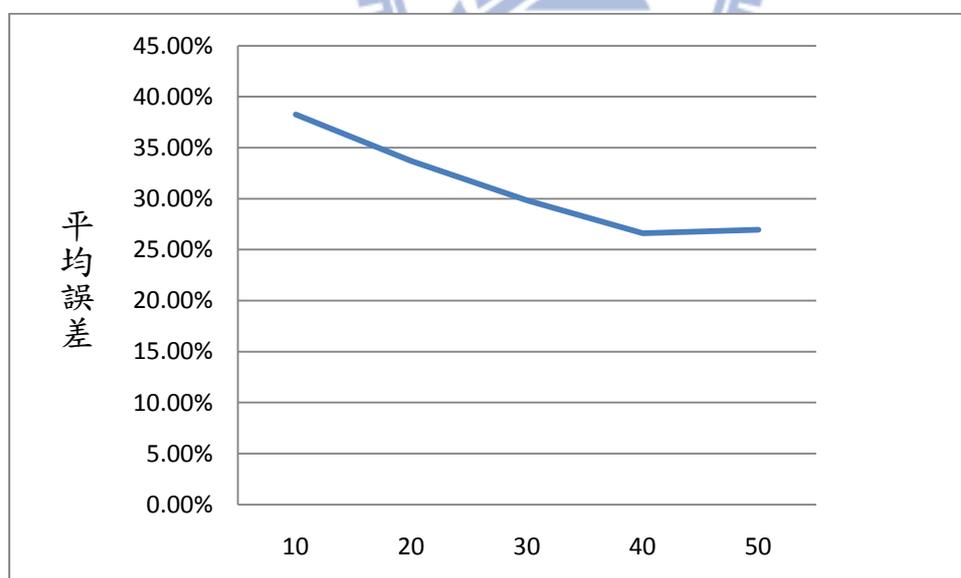


圖 4-1 粒子數與各組合平均誤差百分比趨勢圖

由測試結果可知，當粒子數越多所花費的時間也越長，同時隨著顧客訂單越多，其求解時間亦隨著顧客訂單數而大幅上升。在表 4-4 中，可知粒子數越多，所得到的最佳目標值誤差百分比雖有逐漸降低，但並不保證誤差一定會降低，

就整體而言，粒子數越多，最佳目標值誤差百分比有下降之趨勢；在表 4-5 中，可知粒子數與所得到的平均目標值平均誤差百分比具有負相關，當粒子數越多時，平均目標值平均誤差百分比越小。當粒子數=40 時，最佳目標值平均誤差百分比與平均目標值平均誤差百分比皆有最小值。

粒子數越多時，其求解品質應更好，但根據測試之結果，求解績效較好之粒子數為 40，不過根據圖 4-1 發現在粒子數為 40 或 50 時，所得到之平均誤差百分比幾乎已趨於收斂，但隨著顧客訂單數、粒子數以及迭代數越大，其求解時間會越長，故在該兩種粒子數求解品質差異不多下，本研究選擇 40 為演算法之粒子數。

4.2.2 迭代數

根據過去文獻[4][28]，粒子群最佳化用於VRP或是同時收送貨問題時，其迭代數大部分設計至多1000次，就能獲得品質不錯的解。由於粒子群最佳化尚未應用於本研究課題，而本研究為測試多少迭代數之粒子群最佳化用於收送貨問題能得到較好的求解績效，嘗試將迭代數延伸至1500次，因此其迭代數分別為100、500、1000與1500共4組參數組來測試，如表4-6。配合4.1節所提之顧客訂單為50與100之測試例題與、均勻分佈、半聚集分佈以及聚集分佈三種地理分佈，共六種例題進行測試。其求解結果與時間列於表4-7與4-8，詳細的求解結果與求解時間附於書後附錄二。

表 4-6 測試迭代數多寡之參數組

	組合 6	組合 1	組合 7	組合 8
粒子數	30	30	30	30
迭代數	100	500	1000	1500
c_p	2	2	2	2
c_g	2	2	2	2
慣性權重	0.9-0.4	0.9-0.4	0.9-0.4	0.9-0.4

表 4-7 測試迭代數最佳目標值誤差百分比與求解時間

	組合	組合 6	組合 1	組合 7	組合 8
顧客訂單數	測試之迭代數	100	500	1000	1500
50	誤差百分比	44.72%	16.00%	3.72%	2.34%
	求解時間	46.86	232.75	388.53	585.37

100	誤差百分比	40.77%	23.53%	5.49%	6.89%
	求解時間	115.04	605.50	2448.38	3768.49
	平均誤差百分比	42.74%	19.77%	4.61%	4.62%
	平均求解時間(秒)	80.95	419.12	1418.46	2176.93

表 4-8 測試迭代數平均目標值誤差百分比與求解時間

	組合	組合 6	組合 1	組合 7	組合 8
顧客訂單數	測試之迭代數	100	500	1000	1500
50	誤差百分比	55.88%	27.51%	17.20%	13.57%
	求解時間	35.74	183.67	397.94	612.59
100	誤差百分比	50.95%	32.17%	18.56%	19.98%
	求解時間	100.26	487.12	1525.40	3074.59
	平均誤差百分比	53.41%	29.84%	17.88%	16.77%
	平均求解時間(秒)	68.00	335.40	961.67	1843.59

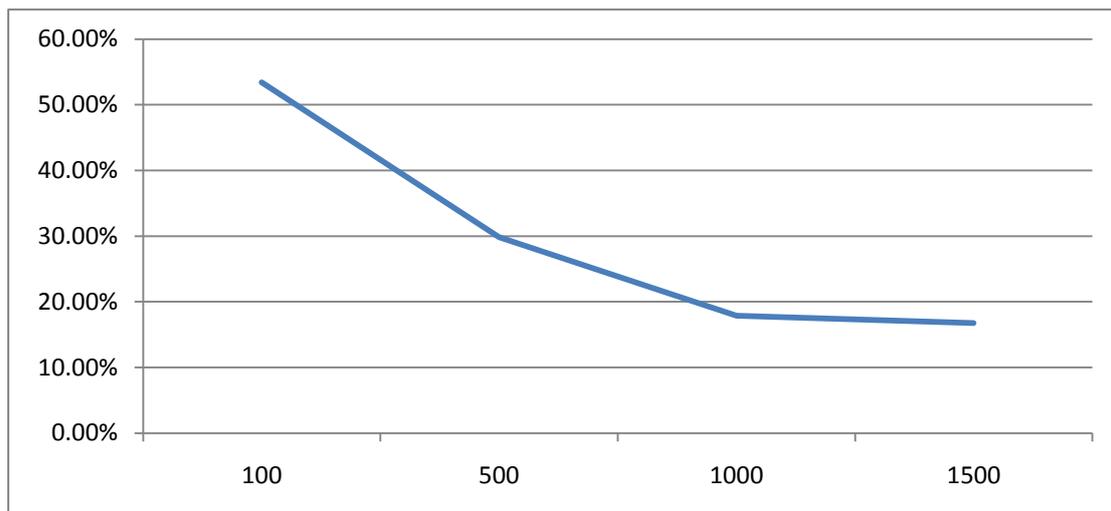


圖 4-2 迭代數與各組合平均誤差百分比相關圖

經過測試後，從表 4-7 與表 4-8 可發現組合 8 在顧客訂單數為 50 時所求得之最佳目標值誤差百分比與平均目標值誤差百分比皆較小，但在顧客訂單數為 100 時所求得之最佳目標值誤差百分比與平均目標值誤差百分比皆以組合 7 為較小。

但從圖 4-2 可以發現迭代數為 1000 與 1500 時，其實所得到之平均誤差百分比幾乎已趨於收斂，而隨著顧客訂單數、粒子數以及迭代數越大，其求解時間會越長，故在該兩種迭代數求解品質差異不多下，本研究選擇 1000 為演算法之迭代數。

4.2.3 慣性權重值

在粒子群最佳化之速度更新公式中，慣性權重值與學習因子能導引粒子搜尋方向與控制粒子搜尋的範圍，因此速度更新公式參數設計的好壞對於求解的品質影響甚大。本研究針對此兩項速度更新公式相關參數進行測試，並將其結果與分析分述至 4.2.3 與 4.2.4。

粒子群最佳化之速度更新公式中，慣性權重值會影響該粒子目前位置對於下個位置之移動多寡。根據過去文獻，Shi and Eberhart[31]建議慣性權重值可依據總運算次數來決定，因此本研究根據[18]所提出之方法：慣性權重值從 0.9 隨著總運算次數而逐次降低至 0.4 來作為測試慣性權重值之方案之一。

另外，本研究提出兩種慣性權重值之更新方法，方法一為慣性權重值從 0.9 開始遞減，遞減之原則為每 100 次迭代數就下降某固定數值，最低降至 0.4，本研究測試慣性權重值所使用之迭代數為 500，因此設定為每 100 次迭代數就下降 0.1。方法二為當群體最佳值每更新一次時，慣性權重值就下降 0.05。

根據過去文獻，發現慣性權重值設定不宜過低，本研究所提出之兩種更新方法，其慣性權重值最低僅到 0.4。上述三組參數組合如表 4-9 所示，並配合顧客訂單數為 50 與 100 的範例例題與節點分佈為均勻分佈、半聚集分佈、聚集分佈三種地理分佈，共六種規模進行測試，求解結果與求解時間列於表 4-10 與 4-11，詳細的求解結果與求解時間附於書後附錄二。

表 4-9 測試慣性權重值之參數組

	組合 1	組合 9	組合 10
粒子數	30	30	30
迭代數	500	500	500
c_p	2	2	2
c_g	2	2	2
慣性權重	0.9-0.4	方法一	方法二

表 4-10 測試慣性權重值之最佳目標值誤差百分比與求解時間

	組合	組合 1	組合 9	組合 10
顧客訂單數	測試之慣性 權重	0.9-0.4	方法一	方法二
50	誤差百分比	16.00%	9.82%	2.99%
	求解時間	232.75	385.18	264.02
100	誤差百分比	23.53%	7.26%	1.78%
	求解時間	605.50	2388.85	1164.55
	平均誤差百 分比	19.77%	8.54%	2.39%
	平均求解時 間(秒)	419.12	1387.02	714.28

表 4-11 測試慣性權重值之平均目標值誤差百分比與求解時間

	組合	組合 1	組合 8	組合 9
顧客訂單數	測試之慣性 權重	0.9-0.4	方法一	方法二
50	誤差百分比	27.51%	27.20%	17.26%
	求解時間	183.67	273.66	277.40

100	誤差百分比	32.17%	18.82%	15.00%
	求解時間	487.12	2167.69	1088.31
	平均誤差百分比	29.84%	23.01%	16.13%
	平均求解時間(秒)	335.40	1220.67	682.86

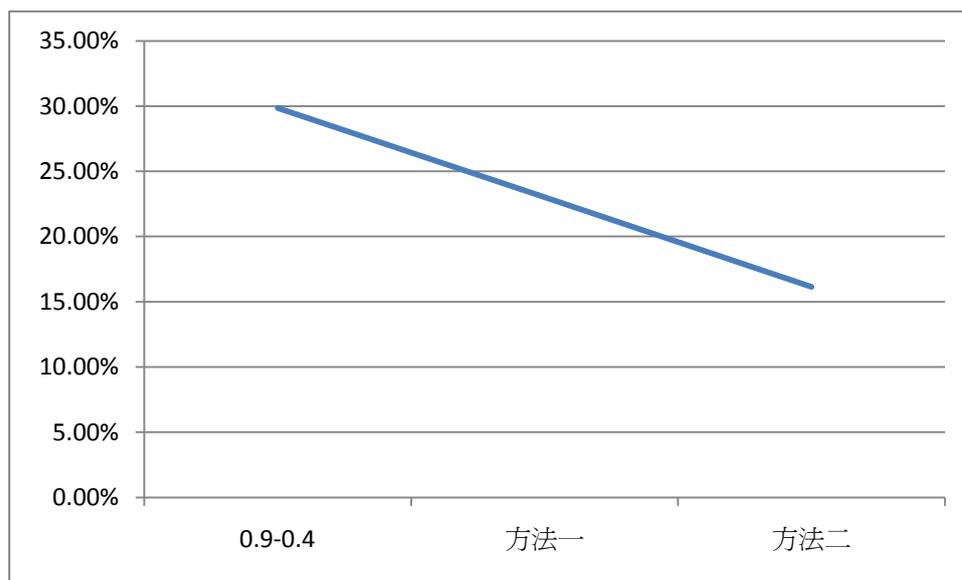


圖 4-3 慣性權重值與各組合平均誤差百分比相關圖

經過測試後，從表4-10與4-11可發現組合10在任一顧客訂單數所求得之最佳目標值誤差百分比與平均目標值誤差百分比皆較小。在本研究之目的為求得績效不錯的解，本研究選擇方法二為本演算法之慣性權重值。

4.2.4 一般 PSO 學習因子(c_p 、 c_g)

粒子群最佳化之速度更新公式中，學習因子(c_p 、 c_g)引導各粒子往區域與全域最佳解方向搜尋，使粒子群能廣泛的搜尋可行解區域，並能夠逐漸收斂至較佳的可行解區域。其中參數 c_g 引導各粒子往所有粒子中的最佳解方位搜尋，而 c_p 則讓粒子能夠根據自身的經驗，搜尋自己經驗中最佳解的可行解區域，使粒子能廣泛的搜尋整個可行解區域。

學習因子 c_g 設定較高時，能使粒子群體迅速收斂至最佳解的區域進行搜尋，其收斂速度較快，但也限制了粒子跳脫區域最佳解區域而搜尋其它空間的可能性，因此較容易陷入區域最佳解；若給予參數 c_p 較高的數值，則可使粒子能有效的搜尋各可行解區域，但其收斂速度較慢。為找出最適合本研究問題特性之學習因子，

本研究根據過去文獻[18][23][29]，針對三種學習因子之組合提出三組參數組合如表 4-12 進行參數測試，其(c_p 、 c_g)分別為(2, 2)、(1.4962, 1.4962)、(2.8, 1.3)，並配合顧客訂單數為 50 與 100 的範例例題與節點分佈為均勻分佈、半聚集分佈、聚集分佈三種地理分佈，共六種規模進行測試。求解結果與求解時間列於表 4-13 與 4-14，詳細的求解結果與求解時間附於書後附錄二。

另外，本小節旨在區分哪種學習因子之組合較好，並非在求取好的求解成果，因此本研究將測試之迭代數從 500 降至 100，以利求解效率。

表 4-12 測試一般 PSO 學習因子之參數組

	組合 1	組合 11	組合 12
粒子數	30	30	30
迭代數	100	100	100
c_p	2	1.4962	2.8
c_g	2	1.4962	1.3
慣性權重	0.9-0.4	0.9-0.4	0.9-0.4

表 4-13 測試一般 PSO 學習因子之值最佳目標值誤差百分比與求解時間

	組合	組合 1	組合 11	組合 12
顧客訂單數	測試之學習因子	$c_p: 2$ $c_g: 2$	$c_p: 1.4962$ $c_g: 1.4962$	$c_p: 2.8$ $c_g: 1.3$
50	誤差百分比	44.72%	24.72%	46.76%
	求解時間	46.85	28.70	31.69
100	誤差百分比	40.77%	30.82%	44.10%
	求解時間	115.04	61.58	78.32
	平均誤差百分比	42.74%	27.77%	45.43%
	平均求解時間(秒)	80.95	45.14	55.01

表 4-14 測試一般 PSO 學習因子之平均目標值誤差百分比與求解時間

	組合	組合 1	組合 11	組合 12
顧客訂單數	測試之學習因子	$c_p: 2$ $c_g: 2$	$c_p: 1.4962$ $c_g: 1.4962$	$c_p: 2.8$ $c_g: 1.3$
50	誤差百分比	55.88%	41.46%	60.90%

	求解時間	35.74	30.08	29.69
100	誤差百分比	50.95%	41.46%	54.15%
	求解時間	100.26	62.33	76.16
	平均誤差百分比	53.41%	41.46%	57.53%
	平均求解時間(秒)	68.00	46.20	52.93

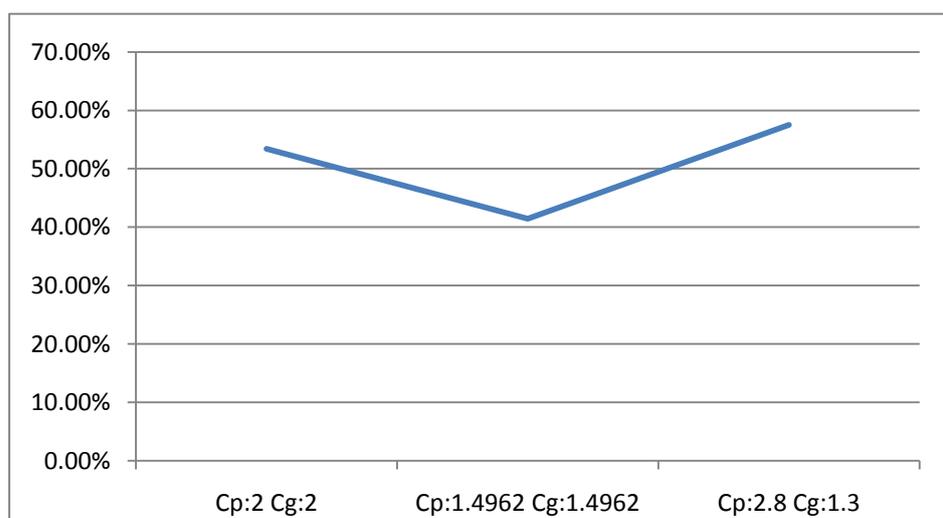


圖 4-4 一般 PSO 學習因子與各組合平均誤差百分比相關圖

經過測試後，從表 4-13 與 4-14 可發現雖然組合 12 在顧客訂單數為 50 時其平均目標值誤差百分比為最低，但組合 11 在任一顧客訂單數所求得之最佳目標值誤差百分比與顧客訂單數為 100 時之平均目標值誤差百分比皆較小。再加上如表 4-14，其平均求解時間為最少，在本研究之目的為求得績效不錯的解，本研究選擇 $c_p: 1.4962$ 、 $c_g: 1.4962$ 為一般 PSO 學習因子之值。

4.2.5 GLNPSO 學習因子(c_p 、 c_g 、 c_l 、 c_n)

本研究根據過去文獻，針對三種學習因子之組合提出三組參數組合如表 4-15 進行參數測試，其(c_p 、 c_g 、 c_l 、 c_n)分別為(2, 2, 1.5, 1.5)、(0.5, 0.5, 1.5, 1.5)、(1, 0, 1, 2)，並配合顧客訂單數為 50 與 100 的範例例題與節點分佈為均勻分佈、半聚集分佈、聚集分佈三種地理分佈，共六種規模進行測試。求解結果與求解時間列於表 4-16 與 4-17，詳細的求解結果與求解時間附於書後附錄二。

另外，本小節旨在區分哪種學習因子之組合較好，並非在求取好的求解成果，因此本研究將測試之迭代數從 500 降至 100，以利求解效率。

表 4-15 測試一般 PSO 學習因子之參數組

	組合 13	組合 14	組合 15
粒子數	30	30	30
迭代數	100	100	100
c_p	2	0.5	1
c_g	2	0.5	0
c_l	1.5	1.5	1
c_n	1.5	1.5	2
慣性權重	0.9-0.4	0.9-0.4	0.9-0.4

表 4-16 測試 GLNPSO 學習因子之最佳目標值誤差百分比與求解時間

	組合	組合 13	組合 14	組合 15
顧客訂單數	測試之學習因子	$c_p:2$ $c_l:1.5$ $c_g:2$ $c_n:1.5$	$c_p:0.5$ $c_l:1.5$ $c_g:0.5$ $c_n:1.5$	$c_p:1$ $c_l:1$ $c_g:0$ $c_n:2$
50	誤差百分比	22.23%	33.64%	24.70%
	求解時間	352.4896	144.4479	204.6563
100	誤差百分比	19.10%	26.86%	30.26%
	求解時間	961.401	579.0885	591.8542
	平均誤差百分比	20.67%	30.25%	27.48%
	平均求解時間(秒)	656.9453	361.7682	398.2552

表 4-17 測試 GLNPSO 學習因子之平均目標值誤差百分比與求解時間

	組合	組合 13	組合 14	組合 15
顧客訂單數	測試之學習因子	$c_p:2$ $c_l:1.5$ $c_g:2$ $c_n:1.5$	$c_p:0.5$ $c_l:1.5$ $c_g:0.5$ $c_n:1.5$	$c_p:1$ $c_l:1$ $c_g:0$ $c_n:2$
50	誤差百分比	35.21%	51.93%	49.89%
	求解時間	297.101	143.0297	191.0327
100	誤差百分比	27.96%	36.72%	36.71%
	求解時間	946.51	562.687	648.1867
	平均誤差百	31.59%	44.33%	43.30%

	分比			
	平均求解時間(秒)	621.8055	352.8583	419.6097

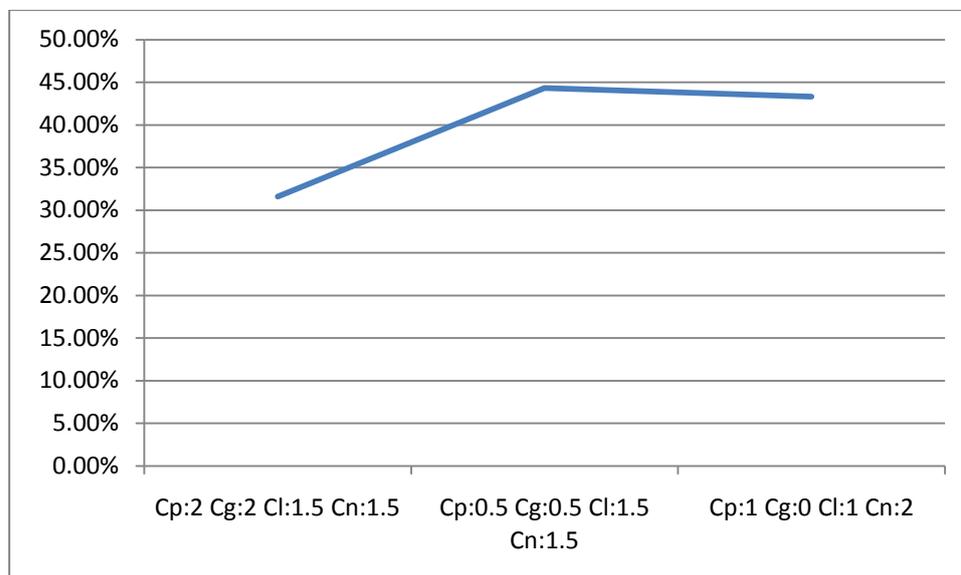


圖 4-5 GLNPSO 學習因子與各組合平均誤差百分比相關圖

經過測試後，從表4-16與4-17可發現組合13之求解時間較多，但求解品質較優，在本研究之目的為求得績效不錯的解，本研究選擇 $c_p: 2$ 、 $c_g: 2$ 、 $c_l: 1.5$ 、 $c_n: 1.5$ 為GLNPSO學習因子之值。

4.2.6 小結

經4.2.1至4.2.5各組參數測試後，本研究之演算法參數值分別為粒子數=40、迭代為=1000、慣性權重值之更新方式為方法二：當群體最佳值更新時，慣性權重值下降0.05、一般PSO學習因子 $c_p: 1.4962$ 、 $c_g: 1.4962$ ，以及GLNPSO學習因子 $c_p: 2$ 、 $c_g: 2$ 、 $c_l: 1.5$ 、 $c_n: 1.5$ 。

4.3 範例測試

參數校估完成後，接下來將評估本研究之演算法績效，以及比較一般 PSO 與 GLNPSO 之求解績效。本小節主要分成兩個部分，首先使用套裝軟體求解小範例之最佳解，與本研究演算法之結果進行比較。第二部分本研究修改 Ropke and Pisinger[15]所提出之例題，以進行本研究演算法更大規模之範例測試，共計 36 題，例題修改之細節已於 4.1 節詳細說明。

由於套裝軟體無法在合理時間內求解規模較大之例題，因此本研究隨機產生

數題小範例進行測試，來測試粒子群最佳化應用於多場站收送貨問題之求解績效，測試範例之資料如表4-18所示。

表 4-18 用來評估演算法績效之測試範例資料

顧客訂單數	場站數	總車輛數	車容量限制
5	2	4	50
6	2	4	50
7	2	4	50

測試方式為針對每個範例以本研究之演算法運算十次，計算出該十次中之最佳值以及平均值，並將十次中之最佳值與套裝軟體所求出之最佳解進行比較，以誤差百分比表示，測試結果如表4-19與4-20所示。表4-21與4-22則為修改Ropke and Pisinger[15]MDPDPTW之例題而得之36題測試範例之測試結果。

表 4-19 一般 PSO 之演算結果與最佳解之誤差表

顧客訂 單數	PSO平均 值	PSO最佳 值	最佳值 與平均 值之差 距	PSO最佳 值之車 輛數	最佳解	最佳解 之車輛 數	誤差百 分比
5	3396.3	3345	51.3	1	3320	1	0.75%
6	4647.8	4549	98.8	1	4433	1	2.62%
7	4665	4043	622	1	3982	1	1.53%

表 4-20 GLNPSO 之演算結果與最佳解之誤差表

顧客訂 單數	PSO平均 值	PSO最佳 值	最佳值 與平均 值之差 距	PSO最佳 值之車 輛數	最佳解	最佳解 之車輛 數	誤差百 分比
5	3345	3345	0	1	3320	1	0.75%
6	4591	4549	42	1	4433	1	2.62%
7	4338.7	4043	295.7	1	3982	1	1.53%

從表4-19與表4-20可以發現，透過一般PSO與GLNPSO所求出之解與最佳解之間之誤差百分比一樣且很小，車輛數皆亦只有一輛，代表本研究方法對於小範例之求解績效不錯。另一方面，透過GLNPSO求解十次而得之平均值與最佳解之差距較一般PSO來的小，代表在求解小範例時，若選擇求解數次而取最佳值，建議採用GLNPSO來求解；若選擇求解的次數較多來取最佳值時，建議採用一般PSO，除了求解時間較GLNPSO短外，其求解品質會隨著求解之次數增加而與GLNPSO無異。

表 4-21 一般 PSO 求解測試範例之結果

測試範例編號	十次測試之平均值	十次測試之平均車輛數	十次測試中之最佳值	十次測試中最佳值之車輛數
50個顧客訂單				
1	28010.14	12.9	26087.35	10
2	28690.44	13.1	26430.24	11
3	26339.53	12.9	23532.81	12
4	27209.98	13.7	24550.58	13
5	21470.82	12.9	19721.38	12
6	20572.22	12	19639.86	12
7	21258.64	12.8	19362.27	14
8	20430.79	11.9	16716.02	10
9	23052.21	12.4	21847.95	11
10	25994.81	12.8	24554.97	12
11	24488.55	13.5	22434.7	13
12	27751.7	12.8	25359.87	14
100個顧客訂單				
13	62607.24	26	57897.31	25
14	58911.13	25	56452.73	25
15	61851.8	27.1	60930.45	26
16	61741.03	26.2	60795.39	24
17	48891.25	25.2	46198.43	23
18	50291.96	25.4	46627.99	25
19	51745.5	25.7	48073.03	23
20	46829.4	27.8	41919.21	27
21	58428.42	24.5	57134.9	23
22	56867.1	25.6	51972.62	26
23	56501.41	27.3	53119.79	24
24	58379.21	26.4	51418.25	26
250個顧客訂單				
25	172626.5	65	160945.6	60
26	170800.2	65.2	164095.8	62
27	165060.5	65.9	155783.8	61
28	167897.3	67.7	162788.7	64
29	125747.7	67.7	120003.6	62
30	139473.6	66.3	131232.5	53

31	143103.3	66.1	136563.4	64
32	150156.8	69.2	144335.3	72
33	156520.3	65.8	146951.6	69
34	153341.3	63.1	144610.4	43
35	161687.2	65.9	156508.4	59
36	154370.4	68.3	148637.8	66

註：每種顧客訂單數之各12題中，前4題之地理分佈方佈為均勻分佈，中間4題為半聚集分佈，後4題為聚集分佈。

表 4-22 GLNPSO 求解測試範例之結果

測試範例編號	十次測試之平均值	十次測試之平均車輛數	十次測試中之最佳值	十次測試中最佳值之車輛數
50個顧客訂單				
1	25584.06	12.1	23166.74	10
2	27808.11	12.5	25729.32	10
3	24263.17	12.2	21877.72	10
4	25016.96	13	22025.68	11
5	19611.26	12.5	16842.40	12
6	19162.45	11.5	17784.57	12
7	20591.97	12.4	17239.62	12
8	17394.84	11.5	14916.04	10
9	21501.78	12	19169.89	11
10	24774.78	12.2	21884.80	11
11	21446.23	13.1	19169.89	11
12	24949.44	12.4	22552.25	13
100個顧客訂單				
13	55206.03	24.3	51791.4	23
14	53794.07	23	47607.81	23
15	55529.08	25.2	50003.56	23
16	55340.8	24	49498.6	22
17	47607.81	22.2	44570.66	21
18	46959.35	23.9	41645.65	20
19	50361.90	24.1	45241.29	22
20	44234.56	22.3	40009.78	20
21	50722.54	22	40728.26	22
22	48800.3	22.8	42441.25	21
23	50186.02	23.4	47415.22	22

24	53455.45	23.7	47629.85	22
250個顧客訂單				
25	162007.86	60	150155.16	57
26	158644.48	61	145938.97	57
27	152155.16	60.2	140157.98	56
28	154270.67	64.6	145614.88	60
29	114866.51	63.1	104463.77	54
30	130550.49	60.3	123333.46	45
31	134887.21	64	121321.30	51
32	137827.24	65	124425.90	54
33	142114.68	68.1	132552.19	60
34	141885.81	61	130847.33	55
35	147478.21	62.4	141155.16	56
36	146583.38	65.3	138480.26	60

本研究將上述之測試結果進行分類討論，首先針對不同訂單數下，找出透過一般PSO與GLNPSO所求得之平均值與最佳值之誤差百分比，從表4-23可以發現，不管是哪一種演算法，所求得之最佳值與平均值之差距百分比會隨著顧客訂單數增加而減少，即在求解大規模之問題時，粒子群最佳化具有穩定性，可減少求解次數，來提昇作業時間上之效率。

另一方面，可以發現透過GLNPSO所求得之值皆較一般PSO優，因此在求解大規模問題時，GLNPSO的求解績效較好。

表 4-23 粒子群最佳化求解測試範例之顧客訂單數差距百分比

粒子群最佳化	顧客訂單數 (例題編號)	總平均值	最佳值之平 均值	最佳值與平均值 之差距百分比
一般PSO	50	24605.82	22519.83	9.26%
	100	56087.12	52711.68	6.40%
	250	155065.4	147704.7	4.98%
GLNPSO	50	22675.42	20196.58	12.27%
	100	51016.49	45715.28	11.60%
	250	143606	133203.9	7.81%

註：差距百分比=

$$\frac{(\text{總平均值}) - (\text{最佳值之平均值})}{(\text{最佳值之平均值})} \times 100\%$$

接下來探討粒子群最佳化是否會因為不同空間分布情況而求解績效產生差異，表4-24為針對不同地理分佈下，計算出透過一般PSO與GLNPSO所求得之平均值與最佳值之差距百分比，可以發現不管是哪種演算法，其誤差百分比不會隨著地理分布不同而有太大之差異。

表 4-24 粒子群最佳化求解測試範例之地理分佈誤差百分比

粒子群最佳化	地理分佈	各地理分佈之總平均值	各地理分佈下最佳值之平均值	最佳值與平均值之差距百分比
一般PSO	均勻分佈	85978.82	81690.9	5.25%
	半聚集分佈	69997.67	65866.08	6.27%
	聚集分佈	79781.88	75379.27	5.84%
GLNPSO	均勻分佈	79135.04	72797.32	8.71%
	半聚集分佈	65337.97	59316.2	10.15%
	聚集分佈	72824.89	67002.2	8.69%

註：誤差百分比為(各地理分佈之總平均值)減(各地理分佈下最佳值之平均值)再除以(各地理分佈下最佳值之平均值)而得。

最後本研究針對不同顧客訂單數與不同地理分佈，計算出十次測試之總平均車輛數與十次測試中最佳值之平均車輛數，如表4-25與4-26，發現最佳值之平均車輛數小於總平均車輛數，未來可在車輛數明顯足夠的情況下，將降低總使用車輛數之目標納入目標式，以更符合實際現況。

表 4-25 粒子群最佳化求解測試範例之平均車輛數

粒子群最佳化	顧客訂單數	總平均車輛數	最佳值之平均車輛數
一般PSO	50	12.80833	12
	100	26.01667	24.75
	250	66.35	61.25
GLNPSO	50	12.28333	11.08333
	100	23.40833	21.75
	250	62.91667	55.41667

表 4-26 粒子群最佳化求解測試範例在不同地理分佈下之平均車輛數

粒子群最佳化	地理分佈	各地理分佈下之平均車輛數	各地理分佈下最佳值之平均車輛數
一般PSO	均勻分佈	35.05833	32.75
	半聚集分佈	35.25	33.08333
	聚集分佈	34.86667	32.16667
GLNPSO	均勻分佈	32.675	30.16667
	半聚集分佈	32.73333	27.75
	聚集分佈	33.2	30.33333



五、結論與建議

5.1 結論

本研究首先結合多場站車輛途程問題與收送貨問題之特性，建構出MDPDP之數學模式，其限制包含優先限制、聯結限制與車容量限制、起迄點相同限制、場站車輛數相同限制。

由於此問題屬於NP-HARD，本研究以粒子群最佳化來求解，並針對慣性權重、學習因子、粒子數與迭代數之參數進行測試，找出適合於本研究之參數值分別為粒子數=40、迭代為=1000、慣性權重值之更新方式為當群體最佳值更新時，慣性權重值下降0.05、一般PSO學習因子 c_p :1.4962、 c_g :1.4962，以及GLNPSO學習因子 c_p :2、 c_g :2、 c_l :1.5、 c_n :1.5。

另外在參數測試中提出兩個新的慣性權重更新法則：方法一為慣性權重值從0.9開始遞減，遞減之原則為每100次迭代數就下降某固定數值，最低降至0.4，方法二為當群體最佳值每更新一次時，慣性權重值就下降0.05。

最後透過範例測試之方式，對於各範例皆能找出一組路徑成本最低之配送路徑，最後檢討本研究之演算法績效，並比較一般PSO與GLNPSO之求解績效，供未來研究之進一步探討。範例測試結果得知：

- (1) 粒子群最佳化對於小規模問題之求解績效不錯。
- (2) 在求解小規模問題時，若選擇較少求解次數來取最佳值，建議採用GLNPSO；若選擇較多求解次數來取最佳值，則建議採用一般PSO。
- (3) 粒子群最佳化求解大規模問題時具有穩定性，且GLNPSO求解績效較一般PSO來的優。
- (4) 粒子群最佳化不受到地理分佈不同而求解績效產生差異。

5.2 建議

(1) 本研究方法是採用隨機亂數之方式產生起始粒子，因此起始粒子之品質不一，若採用產生起始解之方式來產生，像是最遠插入法(以最遠之節點開始插入)、最近插入法(以最近之節點開始插入)或是根據顧客需求之多寡插入法等等，起始解之品質可能較隨機亂數所產生之起始解來的好，在相同的迭代數下，其較易能獲得品質不差的解。

(2) 本研究在測試參數時，對於不同之參數，皆事先提出欲測試之參數組，若能待測完某一參數後，再提出新的參數組，其在測試上亦較精確。舉例來說，本研究在測試完粒子數後決定用40作為本研究之粒子數，若能在測試迭代數與

學習因子時，粒子數皆設定為 40，對於本研究而言，將比粒子數設為 30 來的精確。

(3)本研究方法僅透過 2-OPT 進行交換法改善，改善之效率有限，若未來能配合更多交換法或是結合其他啟發式解法進行改善，求解品質將可能獲得提升。

(4)本研究在 MDPDP 的限制上包含了場站之車輛數相同之限制以及車輛需回到同個場站之限制，未來可將此一限制剔除，讓問題更接近實務。

(5)本研究之目標式為最小總旅行成本，未來可將總使用車輛數一併考慮進去，較符合現今之情況。



六、參考文獻

- [1] Bigang, H. and Changshi, L., “A Two-Stage Hybrid Heuristic for Vehicle Routing Problem with Pickups and Deliveries”, International Conference on Information Management, Innovation Management and Industrial Engineering, Vol. 1, pp. 92-97, Xi'an ,DEC26-27, 2009.
- [2] Catay, B., “A new saving-based ant algorithm for the Vehicle Routing Problem with Simultaneous Pickup and Delivery”, EXPERT SYSTEMS WITH APPLICATION, Vol.37, No. 10, pp. 6809-6817, OCT 2010.
- [3] Cortes, C.E., Saez, D., Nunez, A., et al., “Hybrid Adaptive Predictive Control for a Dynamic Pickup and Delivery Problem”, Transportation Science, Vol. 43, No. 1, pp. 27-42, 2009.
- [4] Goksal, F.P., Altiparmak, F., Karaoglan, I., “A hybrid particle swarm optimization for vehicle routing problem with simultaneous pickup and delivery”, 40th International Conference on Computers and Industrial Engineering(CIE), pp. 1-6, Awaji, JUL 25-28, 2010.
- [5] Irnich, S., “A multi-depot pickup and delivery problem with a single hub and heterogeneous vehicles”, European Journal of Operational Research, Vol. 122, No. 2, pp. 30-328, APR 16 2002.
- [6] Kennedy, J., Eberhart, R.C., “Particle Swarm Optimization”, 1995 IEEE International Conference on Neural Networks, pp. 1942-1948, Univ W Australia Perth Australia, 1995.
- [7] Kennedy, J., Eberhart, R.C., “Swarm intelligence”, Morgan Kaufmann division of Academic Press, 2001.
- [8] Laporte, G., Nobert, Y., Taillefer, S., “Solving a Family of Multi-depot Vehicle-Routing and Location-Routing Problems”, Transportation Science, Vol. 22, No. 3, pp. 161-172, AUG 1998.
- [9] Lee, C.G., Epelman, M.A., White, C.C., et al., “A shortest path approach to the multiple-vehicle routing problem with split pick-ups”, Transportation Research Part B-Methodological, Vol. 40, No. 4, pp. 265-284, MAY 2006.
- [10] Nagy, G., Salhi, S., “Heuristic Algorithms for Single and Multiple Depot Vehicle Routing Problems with Pickups and Deliveries”, European Journal of Operational Research, Vol. 162, No. 1, pp. 126-141, APR 1 2005.

- [11] Parragh, S.N., Doerner, K.F. and Hartl, R.F., “A survey on pickup and delivery problems Part I: Transportation between pickup and delivery locations”, *Journal für Betriebswirtschaft*, Vol 58, No.1, pp. 21-51, April 2008.
- [12] Parragh, S.N., Doerner, K.F. and Hartl, R.F., “A survey on pickup and delivery problems Part II: Transportation between pickup and delivery locations”, *Journal für Betriebswirtschaft*, Vol 58, No.2, pp. 81-117, June 2008.
- [13] Pepin, A.S., Desaulniers, G., Hertz, A., et al., “A comparison of five heuristics for the multiple depot vehicle scheduling problem”, *Journal of Scheduling*, Vol. 12, No. 1, pp. 17-30, 2009.
- [14] Renaud, J., Laporte, G., Boctor, F.F., “A tabu search for the multi-depot vehicle routing problem”, *COMPUTERS & OPERATIONS RESEARCH*, Vol. 23, No. 3, pp. 229-235, MAR 1996.
- [15] Ropke, S. and Pisinger, D., “An Adaptive Large Neighborhood Search Heuristic for the Pickup and Delivery Problem with Time Windows”, *Transportation Science*, Vol. 40, No. 4, pp. 455-472, NOV 2006.
- [16] Savelsbergh, M. W. P. and Solomon, M., “The General Pickup and Delivery Problem”, *Transportation Science*, Vol. 29, No. 1, pp. 17-29, 1995.
- [17] Shen, H., Zhu, Y.L., Liu, T., et al., “Particle Swarm Optimization in Solving Vehicle Routing Problem”, 2nd International Conference on Intelligent Computation Technology and Automation, pp. 287-291, Changsha Peoples R China, 2009.
- [18] Shi, Y. and Eberhart, R.C., “Empirical study of particle swarm optimization”, *Proceedings of the 1999 congress on Evolutionary Computation*, Vol. 3, Washington DC, 1999.
- [19] Sombuntham, P. and Kachitvichayanukul, V., “A Particle Swarm Optimization Algorithm Multi-depot Vehicle Routing Problem Pickup and Delivery Requests”, *International MultiConference of Engineering and Computer Scientists 2010*, pp. 1998-2003, Kowloon, Hong Kong China, 2010.
- [20] Song, M.P., Gu, G.C., “Research on Particle Swarm Optimization: A Review”, *International Conference on Machine Learning and Cybernetics*, pp. 2236-2241, Shanghai Peoples R China, 2004.
- [21] Thangiah, S.R., Salhi, S., “Genetic clustering: An adaptive heuristic for the multidepot vehicle routing problem”, *APPLIED ARTIFICIAL INTELLIGENCE*, Vol. 15, No. 4, pp. 361-383, APR 2001.

- [22] Toth, P. and Vigo, D., *The Vehicle Routing Problem*, SIAM, Philadelphia, 2002.
- [23] Trelea, IC., “The particle swarm optimization algorithm: convergence analysis and parameter selection”, *INFORMATION PROCESSING LETTERS*, Vol. 85, No. 6, pp. 317-325, MAR 2003.
- [24] Wang, S.X., Wang, L.Z., Wang, D.W., “Particle Swarm Optimization for Multi-Depots Single Vehicle Routing Problem”, 20th Chinese Control and Decision Conference, pp. 4659-4661, Yantai Peoples R China, 2008.
- [25] Wang, S.X., Wang, L.Z., Yuan, H.L., et al., “study on Multi-Depots Vehicle Scheduling Problem and Its Two-Phase Particle Swarm Optimization”, 5th International Conference on Intelligent Computing, pp. 748-756, Ulsan South Korea, 2009.
- [26] Wang, X.B., Sun, J.Y., and Ren, C.Y., “Study on hybrid heuristic algorithm for multi-depot vehicle routing problem with hybrid picking-delivery strategy”, 2009 Eighth International Conference on Machine Learning and Cybernetics (ICMLC), pp. 1451-1456, Baoding China, 2009.
- [27] Wen, L., Meng, F.H., “An Improved PSO Multi-Depot Vehicle Routing Problem with Time Windows”, Pacific-Asia Workshop on Computation Intelligence and Industrial Application, pp. 820-824, Wuhan Peoples R China, 2008.
- [28] Wu, B., Wang, W.L., Zhao, Y.W., et al., “A Novel Real Number Encoding Method of Particle Swarm Optimization for Vehicle Routing Problem”, 6th World Congress on Intelligent Control and Automation, pp. 3271-3275, Dalian Peoples R China, JUN 21-23 2006.
- [29] Z. Li-Ping, Y. Huan-jun, and H. Shang-Xu, “Optimal choice of parameters for the particle swarm optimization”, *Journal of Zhejiang University: science*, Vol. 6, pp. 528-534, 2005.
- [30] Zhang, R., Wu, C., “A Divide-and-Conquer Strategy with Particle Swarm Optimization for the Job Shop Scheduling Problem”, Vol. 42, No. 7, pp. 641-670, 2010.
- [31] 李旺蒼, 「以粒子群最佳化為基礎之混合式全域搜尋演算法求解含凹型節線成本最小成本轉運問題之研究」, 國立中央大學土木工程學系碩士論文, 民國九十五年六月。
- [32] 林至康, 「汽車客運多場站車輛排程問題之研究」, 國立交通大學運輸科技與管理學系博士論文, 民國九十八年三月。

- [33] 黃瓊萱, 「多場站具時窗限制並配合外包策略之車輛路線問題探討」, 國立東華大學全球運籌管理系碩士論文, 民國九十六年六月。
- [34] 許哲斌, 「在需求變動下具有軟時窗限制之同時收、送貨車輛途程問題」, 國立雲林科技大學工業工程與管理學系碩士論文, 民國九十七年六月。
- [35] 葉珮婷, 「應用變數產生法求解有時間窗限制的收送貨問題」, 國立交通大學運輸科技與管理學系碩士論文, 民國九十八年七月。
- [36] 業思緯, 「應用粒子群最佳化演算法於多目標存貨分類之研究」, 元智大學工業工程與管理系碩士論文, 民國九十三年六月。
- [37] 萬芸萍, 「國際企業供應鏈營運面匯率避險之研究」, 國立成功大學交通管理科學系碩士論文, 民國九十一年六月。



附錄一 各組參數測試之詳細結果

附表 1-1 測試粒子數之各組最佳目標值誤差百分比與求解時間

組合	組合 2		組合 3		組合 1		組合 4		組合 5	
顧客 訂單 與 地理 分佈	誤差 百分 比	求解 時間 (s)								
50- 均 勻	27.98 %	46.4 531	13.03 %	161. 906	8.06%	208. 125	6.95 %	220	9.24 %	269. 516
50- 聚 集	29.89 %	60.9 844	16.37 %	122. 953	27.30 %	234. 672	16.0 6%	349.1 41	11.27 %	463. 141
50- 半 聚 集	8.83%	74.2 656	9.93%	93.2 031	12.63 %	255. 453	9.12 %	202.6 88	11.59 %	276. 672
100 -均 勻	16.96 %	117. 141	10.86 %	574. 188	20.54 %	379. 453	21.3 3%	1126. 02	14.9 3%	901. 453
100 -聚 集	37.19 %	132. 391	17.91 %	548. 719	23.72 %	704. 313	7.74 %	980.2 19	13.0 2%	2248 .78
100 -半 聚 集	34.26 %	138. 219	29.31 %	378. 609	26.34 %	732. 734	11.3 9%	803.5 47	17.9 9%	1149 .14

平均	25.85 %	94.9 0885	16.23 %	313. 263	19.77 %	419. 125	12.1 0%	613.6 016	13.0 1%	884. 7839
----	------------	--------------	------------	-------------	------------	-------------	------------	--------------	------------	--------------

附表 1-2 測試粒子數之各組平均目標值誤差百分比與求解時間

組合	組合 2		組合 3		組合 1		組合 4		組合 5	
顧客 訂單 與 地理 分佈	誤差 百分 比	求解 時間 (s)								
50- 均 勻	33.89 %	59.2 4531	30.72 %	124. 6547	20.93 %	175. 7906	18.6 8%	260.6 344	23.94 %	286. 1234
50- 聚 集	43.46 %	54.5 4375	34.76 %	141. 1391	40.21 %	193. 6188	31.0 8%	252.4 531	34.09 %	322. 5594
50- 半 聚 集	35.01 %	59.2 3125	31.82 %	103. 6813	21.38 %	181. 6297	18.3 2%	253.0 125	18.19 %	293. 7266
100 -均 勻	30.13 %	144. 9281	29.57 %	344. 0031	27.98 %	432. 0969	28.7 4%	646.4 422	28.56 %	897. 3578
100 -聚 集	47.32 %	162. 1297	35.61 %	402. 4641	36.39 %	537. 8734	34.8 9%	941.6 516	29.10 %	1270 .223
100 -半 聚	39.73 %	164. 325	39.63 %	309. 8188	32.13 %	491. 3922	28.0 0%	704.3 438	27.80 %	822. 0188

集										
平均	38.26 %	107. 4005	33.68 %	237. 6268	29.84 %	335. 4003	509. 7563	-98.3 8%	26.95 %	648. 6682

附表 1-3 測試迭代數之各組最佳目標值誤差百分比與求解時間

組合	組合 6		組合 1		組合 7		組合 8	
顧客 訂單 與 地理 分佈	誤差百 分比	求解 時間 (s)	誤差百 分比	求解時 間(s)	誤差百 分比	求解時 間(s)	誤差百 分比	求解時 間(s)
50- 均 勻	40.78%	29.12 5	8.06%	208.12 5	0.88%	470.57 8	0.06%	609.406
50- 聚 集	58.50%	70.46 88	27.30%	234.67 2	10.29%	348.59 4	6.93%	577.719
50- 半 聚 集	34.87%	40.98 44	12.63%	255.45 3	0.00%	346.43 8	0.03%	569
100 -均 勻	34.58%	98.78 13	20.54%	379.45 3	8.63%	892.42 2	2.81%	3691.88
100 -聚 集	45.66%	122.9 22	23.72%	704.31 3	7.85%	3491.8 8	9.26%	4710.42

100 -半 聚 集	42.06%	123.4 22	26.34%	732.73 4	0.00%	2960.8 6	8.60%	2903.19
平 均	42.74%	80.95 052	19.77%	419.12 5	4.61%	1418.4 61	4.62%	2176.93 5

附表 1-4 測試迭代數之各組平均目標值誤差百分比與求解時間

組 合	組合 6		組合 1		組合 7		組合 8	
顧 客 訂 單 與 地 理 分 佈	誤差 百分 比	求解 時間 (s)	誤差百 分比	求解時 間(s)	誤差百 分比	求解時 間(s)	誤差百 分比	求解時 間(s)
50- 均 勻	45.81 %	33.4	20.93%	175.79 06	16.65%	387.43 44	11.13%	606.53 28
50- 聚 集	75.99 %	40.20 156	40.21%	193.61 88	19.63%	449.66 88	18.95%	617.05 31
50- 半 聚 集	45.84 %	33.64 531	21.38%	181.62 97	15.32%	356.72 97	10.61%	614.19 85
100 -均 勻	43.52 %	102.1 641	27.98%	432.09 69	16.88%	973.84 06	17.38%	2807.7 09
100	58.31	98.02	36.39%	537.87	21.28%	1484.1	22.74%	3277.4

-聚集	%	5		34		27		38
100-半聚集	51.01%	100.6156	32.13%	491.3922	17.52%	2118.256	19.83%	3138.628
平均	53.41%	68.00859	29.84%	335.4003	17.88%	961.676	16.77%	1843.593

附表 1-5 測試慣性權重值之各組最佳目標值誤差百分比與求解時間

組合	組合 1		組合 9		組合 10	
顧客訂單與地理分佈	誤差百分比	求解時間 (s)	誤差百分比	求解時間 (s)	誤差百分比	求解時間 (s)
50-均勻	8.06%	208.125	12.33%	271.734	0.00%	315.281
50-聚集	27.30%	234.672	5.97%	571.313	0.00%	252.969
50-半聚集	12.63%	255.453	11.17%	312.5	8.98%	223.813
100-均勻	20.54%	379.453	7.89%	2218.83	0.00%	1276.55
100-聚集	23.72%	704.313	4.02%	1745.36	0.00%	1584.14
100-半聚集	26.34%	732.734	9.88%	3202.39	5.34%	632.984
平均	19.77%	419.125	8.54%	1387.021	2.39%	714.2891

附表 1-6 測試慣性權重值之各組平均目標值誤差百分比與求解時間

組合	組合 1		組合 9		組合 10	
顧客訂單與地理分佈	誤差百分比	求解時間 (s)	誤差百分比	求解時間 (s)	誤差百分比	求解時間 (s)
50-均勻	20.93%	175.7906	19.39%	267.5281	12.05%	293.4172
50-聚集	40.21%	193.6188	29.35%	277.4625	19.37%	232.4328
50-半聚集	21.38%	181.6297	32.85%	276.0016	20.36%	306.3563

100-均勻	27.98%	432.0969	17.63%	1815.156	14.50%	1182.388
100-聚集	36.39%	537.8734	16.13%	2669.758	13.26%	1176.184
100-半聚集	32.13%	491.3922	22.71%	2018.17	17.25%	906.3828
平均	29.84%	335.4003	23.01%	1220.679	16.13%	682.8602

附表 1-7 一般 PSO 學習因子之各組最佳目標值誤差百分比與求解時間

組合	組合 1		組合 11		組合 12	
顧客訂單 與地理分 佈	誤差百 分比	求解時間 (s)	誤差百 分比	求解時間 (s)	誤差百 分比	求解時間 (s)
50-均勻	40.78%	29.125	26.00%	28.2656	44.38%	28.9219
50-聚集	58.50%	70.4688	31.67%	29.7344	51.50%	31.6406
50-半聚集	34.87%	40.9844	16.48%	28.125	44.39%	34.5156
100-均勻	34.58%	98.7813	27.64%	59.1094	40.62%	81.9688
100-聚集	45.66%	122.922	35.12%	62.875	49.48%	83.5313
100-半聚集	42.06%	123.422	29.70%	62.7813	42.20%	69.4844
平均	42.74%	80.95052	27.77%	45.14844	45.43%	55.01042

附表 1-8 一般 PSO 學習因子之各組平均目標值誤差百分比與求解時間

組合	組合 1		組合 11		組合 12	
顧客訂單 與地理分 佈	誤差百 分比	求解時間 (s)	誤差百 分比	求解時間 (s)	誤差百 分比	求解時間 (s)
50-均勻	45.81%	33.4	38.91%	30.57813	52.11%	29.92656
50-聚集	75.99%	40.20156	47.02%	30.3625	76.83%	29.4625
50-半聚集	45.84%	33.64531	38.44%	29.30156	53.77%	29.7
100-均勻	43.52%	102.1641	34.54%	62.40781	45.10%	77.57188
100-聚集	58.31%	98.025	48.64%	61.82813	63.62%	76.17031
100-半聚	51.01%	100.6156	41.21%	62.77969	53.72%	74.75

集						
平均	53.41%	68.00859	41.46%	46.20964	57.53%	52.93021

附表 1-9 GLNPSO 學習因子之各組最佳目標值誤差百分比與求解時間

組合	組合 13		組合 14		組合 15	
顧客訂單 與地理分 佈	誤差百 分比	求解時間 (s)	誤差百 分比	求解時間 (s)	誤差百 分比	求解時間 (s)
50-均勻	12.23%	364.641	24.07%	150.594	4.77%	185.422
50-聚集	33.65%	334.016	33.00%	147.063	34.28%	222.781
50-半聚集	20.82%	358.813	43.85%	135.688	35.06%	205.766
100-均勻	2.62%	892.422	7.61%	659.109	6.57%	581.96
100-聚集	35.12%	1068.36	43.26%	415.375	54.24%	608.281
100-半聚 集	19.56%	923.42	29.70%	662.781	29.97%	585.313
平均	20.67%	656.9453	30.25%	361.7682	27.48%	398.2552

附表 1-10 GLNPSO 學習因子各組平均目標值誤差百分比與求解時間

組合	組合 13		組合 14		組合 15	
顧客訂單 與地理分 佈	誤差百 分比	求解時間 (s)	誤差百 分比	求解時間 (s)	誤差百 分比	求解時間 (s)
50-均勻	28.89%	294.65	47.23%	149.609	38.55%	159.134
50-聚集	43.92%	297.503	58.52%	140.491	65.31%	256.553
50-半聚集	32.83%	299.15	50.03%	138.989	45.82%	157.411
100-均勻	7.86%	973.841	14.52%	572.408	17.67%	587.572
100-聚集	48.64%	927.873	54.54%	577.873	57.64%	717.438
100-半聚 集	27.39%	937.816	41.12%	537.78	34.82%	639.55
平均	31.59%	621.8055	44.33%	352.8583	43.30%	419.6097