行政院國家科學委員會專題研究計畫 成果報告

應用粒子族群最佳化演算法及相關巨集啟發式演算法於物流中心揀貨路徑整合分區儲存、訂單批量之研究(II)

研究成果報告(精簡版)

計 畫 類 別 ： 個別型
計 畫 編 號 ： NSC 96-2221-E-216-002-
執 行 期 間 ： 96 年 08 月 01 日至 97 年 07 月 31 日
執 行 單 位 ： 中華大學科技管理學系(所)

計 畫 主 持 人 ： 謝玲芬

計畫參與人員 ： 碩士班研究生-兼任助理人員：楊笠詩
碩士班研究生-兼任助理人員：王正安
碩士班研究生-兼任助理人員：黃羿蓁

報 告 附 件 ： 出席國際會議研究心得報告及發表論文

處 理 方 式 ： 本計畫涉及專利或其他智慧財產權，1 年後可公開查詢

中 華 民 國 97 年 10 月 09 日

行政院國家科學委員會補助專題研究計畫 ☑ 成 果 報 告
□期中進度報告

應用粒子群最佳化演算法及相關巨集啟發式演算法於物流中心揀貨路徑整合分區儲存、訂單批量之研究(II)

計畫類別：☑ 個別型計畫　　□ 整合型計畫
計畫編號：NSC 96－2221－ E －216－002－
執行期間： 96 年 8 月 1 日至 97 年 7 月 31 日

計畫主持人：謝玲芬
共同主持人：
計畫參與人員：楊笠詩、王正安、黃羿蓁

成果報告類型(依經費核定清單規定繳交)：☑精簡報告　□完整報告

本成果報告包括以下應繳交之附件：
□赴國外出差或研習心得報告一份
□赴大陸地區出差或研習心得報告一份
☑出席國際學術會議心得報告及發表之論文各乙份
□國際合作研究計畫國外研究報告書一份

處理方式：除產學合作研究計畫、提升產業技術及人才培育研究計畫、列管計畫及下列情形者外，得立即公開查詢
□涉及專利或其他智慧財產權，☑一年□二年後可公開查詢

執行單位：中華大學科技管理學系

中　華　民　國　97 年　10　月　10　日

中文摘要

為了因應經濟快速發展及消費型態的改變，消費者求新求快，且更多元化，需求呈現少量多樣化的特色，導致商品的生命週期大幅的縮短，而物流中心的通路型態也由傳統的多層式架構，轉變為供應鏈中直接連接製造商與面對消費者的重要角色。揀貨作業為物流中心內最主要的作業活動，也是物流中心成本的主要來源，因此有效的改善物流中心內部揀貨作業的效率，將有助於提升物流中心的服務品質及降低營運成本。

本研究將儲位指派、訂單批量及揀貨路徑規劃作最佳整合規劃，並且嘗試分別應用粒子群演算法、螞蟻理論、基因演算法、模擬退火及橫跨法改良策略於揀貨路徑規劃，透過模擬實驗，比較上述五種演算法與最大迴圈插入法之績效差異，發現螞蟻理論與最大迴圈插入法表現最佳；更進一步證實，以最大迴圈插入法所得之解當作四種巨集演算法的起始解，可以有效提升 4 種演算法之績效。另外發現當組合因子為較差之儲存與訂單批量策略時，粒子群演算法因為搜尋方式較發散，導致在較差決策時收斂情形略遜於其他演算法。而透過模擬實驗驗證並以總揀貨距離、總揀貨時間、揀貨車利用率三項績效指標評估各因子水準組合，再利用統計分析手法找出各因子之最佳組合，期望本研究結果能提供物流業者在決策時之參考。

關鍵詞： 揀貨系統、儲位指派、訂單批量、揀貨路徑規劃、巨集啟發式演算法

Abstract

In order to adapt to quickly economic development and the changes of expense state that reduced the life cycle of goods. The state of channel in distribution center form traditional multi-layer framework transform into the important role that connect with manufacturers directly and face to consumers in supply chain. Order picking operation is the principal activity in warehouses and causes the main cost. Thus, improving efficiency of the order picking operation will increase the quality of service and decrease the operation costs.

This project integrates class storage, order batching and routing to do the best planning, and try to compare the performance of routing policy of the Particle Swarm Optimization algorithm (PSO), Ant System (AS), Genetic Algorithm (GA), Simulation Analysis (SA), Cross^{+} strategy and Maximum Loop Insertion (MLI). Through the simulation experiment, it verified that the MLI and AS algorithm are the best algorithm of routing strategy. Furthermore, we confirm that using MLI solution to find an initial solution for all the four Meta-heuristics, it can improve the efficiency of the Meta-heuristics solution and has the better effect of average total order picking distance. Besides, to compare the result that PSO is not good when it combines the worst storage and order batching strategies. Because of the PSO's particle has larger search area that also induces the PSO has the worse convergence than other Meta-heuristics. According to the simulation result are analyzed by 3 indexes including total picking distance, total picking time and picking vehicle utilization to find the optimal combination for order picking system. Consequently, the result of this project will enhance the whole performance of order picking systems in distribution centers and provide the industry as a reference in the future.

Keywords: Order picking systems, Storage assignment, Order batching, Picking routing planning, Meta-heuristics

1. 前言

物流中心隨著物流服務競爭多樣化與經營戰略的要求，導致物流成本不斷向上攀升，學者[23]在研究中皆提到揀貨作業約佔倉儲營運成本 65%，而揀貨作業時間佔整個作業時間的 30%，因此，揀貨作業效率化將可有效降低營運成本。本研究將針對物流中心揀貨作業之訂單批次處理、儲位指派與揀貨路徑規劃的多重改善下，進而達到最小化揀貨行走距離以提高揀貨作業的效率的目標。因此本研究目的為：1.嘗試運用粒子群演算法、螞蟻理論、基因演算法、模擬退火法於揀貨路徑規劃，並比較四種演算法之績效差異。更進一步將粒子演算法運用於訂單批量，也與其他演算法作績效之比較。2.運用前期計畫所發展出之最大迴圈插入法，將其當作粒子族群最佳化演算法、螞蟻理論、基因演算法及模擬退火法之起始解，並比較其與未將最大迴圈插入法當起始解之結果有何不同。3.對於影響物流中心中之儲位指派、訂單批量與揀貨路徑策略之各種不同組合進行比較分析，以得到最佳的揀貨作業組合。

2. 文獻探討

本計畫將對於影響物流中心之揀貨距離、揀貨時間、空間利用率之「儲位指派」、「訂單揀取方法」、「揀貨路徑規劃」等三大方向來進行探討。此外，也針對「螞蟻理論」、「粒子族群最佳化演算法」、「基因演算法」與「模擬退火法」四種巨集啟發式演算法加以探討，考量上述四種演算法均以尋求全域最佳解為主要目標，且可應用於訂單批量與揀貨路徑規劃，如此更能進一步提昇揀貨作業效率。

2.1. 儲位指派

儲位指派是考慮品項儲存位置的問題，一般情形下，我們假設每個產品有其固定儲存區域，及管理人可以有能力的指派產品到任何位置並追蹤目前儲存產品，透過良好的儲位指派可以減少出入庫移動的距離、縮短作業時間，甚至能夠充分利用儲存空間。Petersen 與 Schmenner[5]提出依需求量分配高低的四種儲位指派法則，分別為斜角佈置(Diagonal)、通道間法(Within-Aisle)、跨走道佈置(Across-Aisle)與週邊佈置(Perimeter)搭配揀貨路徑策略做比較，該研究指出通道間法(Within-Aisle)儲位指派較其他的儲存法節省約 10~20%的揀貨距離。在學者[1]研究中，將 Petersen 與 Schmenner 提出的四種儲位指派法則結合產品關連法則，於產品儲位指派時能同時考量產品之間的關聯程度，並將產品間關聯程度高者擺放於同一走道，以節省揀貨距離。

關聯法則分析由 Agrawal 在 1993 年提出，主要目的是從龐大銷售交易記錄資料庫中，尋找銷售項目間令人感到有興趣的關聯或相互關係[3,8]，最典型的應用是市場購物籃分析[9, 22]。目前已經被發展出之關聯法則中，最具有效性首推 Agrawal et. al. [3]所提出之 Apriori 演算法，此法則先從資料庫中尋找項目集合所出現的頻繁度，若頻繁度大於或等於最小支持度 (Minimal Support) 門檻值，即稱這些項目集為頻繁項目集合(Frequent Itemsets)，隨後利用所高頻繁項目組，推導出所有的關聯法則。

2.2. 訂單批量

一般揀貨方式可分為四種：訂單別揀取、批量揀取、彙整訂單揀取、複合揀取。Lin 與 Lu[13]提出五種訂單分類及兩種策略之搭配，經由模擬驗證，發現多樣多量與少樣少量是適合單一訂單揀貨及批量區域揀貨策略。Ruben and Jacobs[18]比較五種訂單批次法，分別為隨機批量 (Random Batching; RAN)、最少批量(First-Fit- Decreasing; FF-D)、連續最小距離(Sequential Minimal Distance; SMD)、最適封包批量(First Fit-Envelope Based Batching; FF-EBB)、最適分級批量(First Fit-Class Based Batching; FF-CBB)，研究中指出最適封包批量在大多環境中有較佳的績效。

近幾年有學者以資料挖礦(Data Mining)以及關聯法則的觀念應用在求解訂單批次的問

題。Chen 等人[4]。以資料挖礦裡關聯法則中的 Apriori Algorithm 計算每兩張訂單間的關聯度，將關聯度高者優先加入批次裡，其驗證結果中皆能有不錯的績效表現。

2.3. 揀貨路徑規劃

Tompkins et al. [23]提及旅途時間(Travel Time)約佔揀貨活動 50%。因此透過良好的揀貨路徑規劃，將可縮短揀貨時間並提昇揀貨績效。Roodbergen 與 Koster[17]在交叉走道的倉儲環境系統下，針對各種不同的揀貨策略比較在不同走道數目、不同品項數以及不同走道寬度的揀貨路徑評估。求算最短路徑之啟發式演算法包括：S 型法、最大間隙法、走道接走道法、最佳法、結合啟發法(Combined Heuristic)以及結合法改良策略(Combine$^+$ Heuristic)，所提出的結合法改良策略有最佳績效表現，而學者[1]又將其加以改善，提出橫跨法策略(Cross Strategy)，並加入兩點改善法則，將其稱為橫跨法改良策略(Cross$^+$ Strategy)，該研究所提出的橫跨法改良策略，經實驗結果證明，較結合法改良策略更能有效的改善揀貨距離。

謝玲芬與黃建霖[2]提出一啟發式路徑規劃演算法－最大迴圈插入法(Maximum Loop Insertion；簡稱 MLI)，MLI 是以 I/O 為起始點，找出最左邊走道最遠的揀取點與最後一個走道中最遠的點，此三點先形成一個迴圈，再依次找尋加入後增加距離最小的點為下一個加入之揀取點，重覆此步驟直至全部揀取點規劃完成為止。研究發現 MLI 在揀貨距離上明顯優於（NC 最接近矩形中心啟發式(Nearest Center of Rectangular Insertion; NCRI)與最短旅行迴圈插入啟發式(Minimum Traveling Loop Insertion; MTLI)演算法。

2.4. 螞蟻理論

螞蟻系統(Ant System; AS)最早是由 Dorigo 於 1992 年[6]所提出來，其理論為觀察自然界螞蟻搜尋食物的過程所發展出來的。螞蟻在搜尋食物時，會在所走過的路上留下揮發性的化學物質，稱為費洛蒙(Pheromone)，而幾乎全盲的螞蟻便是藉由費洛蒙來傳遞訊息，以進行溝通。

Dorigo 最早提出來應用於銷售員旅行問題的 AS 公式如下：

$$p_{ij}^{k}(t)=\begin{cases}\dfrac{[\tau_{ij}(t)]^{\alpha}\times[\eta_{ij}]^{\beta}}{\sum_{k\in \mathrm{allowed}_k}[\tau_{ik}(t)]^{\alpha}\times[\eta_{ik}]^{\beta}} & \text{if} \quad j\in \mathrm{allowed}_k, \\ 0 & \text{otherwise}\end{cases} \tag{2.1}$$

其中 $P_{ij}^{k}(t)$表示第 k 隻螞蟻在第 t 次迭代中，從城市 i 選擇下一個城市 j 的機率；$\tau_{ij}(t)$ 表示迭代 t 時，由城市 i 到城市 j 之費落蒙素濃度；η_{ij} 表示迭代 t 時，由城市 i 到城市 j 的視覺能力，公式(2.1)主要依據螞蟻行走時所遺留的費洛蒙與其視覺能力，兩者所構建而成。

$$\tau_{ij}(t+n)=\rho\cdot\tau_{ij}(t)+\Delta\tau_{ij} \tag{2.2}$$

$$\Delta\tau_{ij}=\sum_{k=1}^{m}\Delta\tau_{ij}^{k} \tag{2.3}$$

公式(2.2)為費洛蒙更新機制，隨著時間的經過，費洛蒙將會逐漸的蒸發，所以藉由 ρ 表示此蒸發機制，而費洛蒙亦會因螞蟻行走於相同的路徑上而逐漸累積，由公式(2.3)計算所有 m 隻螞蟻，於城市 i 到城市 j 所遺留的費洛蒙總和。

針對費洛蒙累積($\Delta\tau_{ij}$)之更新方式著手，所以有許多學者針對此更新方式提出許多研究，而主要分為兩類：

一、局部更新法(Local Update)

每當螞蟻經過一路徑，不等到建構出完整路徑，即立刻更新路徑上的費洛蒙，主要是降低該隻螞蟻所建構路徑的費洛蒙濃度，避免吸引其他的螞蟻走上相同路徑，使其他螞蟻有機會收尋其他路徑。

二、全域更新法(Global Update)

於每迭代中螞蟻建構出完整路徑(tour)後，才進行費洛蒙更新，能促使最佳解之路徑有較高的費洛蒙濃度，引導其餘螞蟻往此路徑探索(Explore)或開發(Exploit)。

2.5. 模擬退火法

模擬退火演算法(Simulated Annealing；SA)最初是由 Metropolis 等人[15]於 1953 年所提出，直到 1983 年才由 Kirkpatrick 等人[12]開始應用於求解組合最佳化問題，至今亦廣泛應用於各領域之中。而 SA 演算法是利用材料學中的退火過程所產生的能量損耗現象，將此能量損耗轉換為問題之目標函數，並求得最佳解。此法則最大之優點是在除了接受較佳解之外，並沒有完全否定較差解，SA 法會隨機給定一機率值與波茲曼函數值來判斷是否接受此較差之可行解，如此便可幫助跳脫局部最佳解。至於 Metropolis 等人提出演算步驟如下：

步驟一：設定一初始狀態 I_i，並求其能量 $E(I_i)$，及控制參數 T

步驟二：經由擾動產生新的狀態 N_i，並求其能量 $E(N_i)$。

步驟三：計算 $E(N_i)$與 $E(I_i)$能量差 $\Delta E = E(N_i) - E(I_i)$

步驟四：若 $\Delta E < 0$，表示新能量解較佳，因此接受 Ni，並跳至步驟五。反之若 $\Delta E \geqq 0$ 則產生隨機變數 $X \sim U(0,1)$，並利用波茲曼機率分佈函數檢驗是否接受狀態 N_i，而其函數型態為：$P(\Delta E,T)=\exp(-\Delta E/T)$，T 為目前溫度。當 $P(\Delta E,T) > X$ 時，則接受狀態 N_i，並跳至步驟五，反之則拒絕，且跳至步驟 2。

步驟五：狀態 Ni 取代 I_i，並回至步驟 2，重複擾動並迭代之。

2.6. 基因演算法

基因演算法(Genetic Algorithm；GA)是最佳化問題中常用的一種最佳化搜尋方法，其基本精神在於模仿生物界物競天擇、適者生存的自然演化過程，以求得問題最終穩定且最佳的結果。其最早是由 Holland 於 1975 年所提出。基因演算法的運算過程中，先將問題中的參數編碼(Encode)為染色體(Chromosome)，染色體也稱作個體(Individual)，每個染色體皆視為搜尋空間中的解，經由比較適應值(fitness value)高低來評估染色體的好壞，接著再進行複製(Reproduction)、交配(Crossover)以及突變(Mutation)等動作。

基因演算法發展至今已應用於各領域之最佳化問題，如銷售員旅行問題(Traveling Salesman Problem)、車輛途程問題(Vehicle Routing Problem)等等的問題。也有學者於參數設定做深入研究，如母體數、交配率、突變率[14,7]，以防演算法搜尋時過早收斂。另外 Hsu 等人[10]應用 GA 發展出一啟發式演算法(GA Batching-Method；GABM)求解訂單批次問題，其實驗結果顯示出應用 GA 求解訂單批量問題能有相當程度的改善，且不論訂單規模大小，皆適合應用在現實世界裡。

2.7. 粒子群最佳化演算法

1995 由 Kennedy and Eberhart[11]共同提出粒子群最佳化演算法(Particle Swarm Optimization, PSO)，其主要概念源自於鳥類或魚群覓食的社會行為。PSO 為具有群體智能(Swarm Intelligence)的演算法，透過群體智能來求解問題，主要依據三個因子，分別為自己目前前進的方向、自己先前的最佳經驗與群體的最佳經驗，依此三者不斷修正自己的速度與位置，逐漸往食物的所在位置移動。Shi 和 Eberhart 於 1998 所提出慣性權重(Inertia Weight, w)[21]底下為慣性權重公式：

$$v_{id}^{new} = w * v_{id}^{old} + c_1 * rand() * (p_{id} - x_{id}^{old}) + c_2 * Rand() * (p_{gd} - x_{id}^{old}) \qquad (2.4)$$

$$x_{id}^{new} = x_{id}^{old} + v_{id}^{new} \qquad (2.5)$$

其中 x_{id} 表示第 i 個粒子於第 d 個空間維度的位置，v_{id} 表示第 i 個粒子於第 d 個空間維度的速度。 x_{id}^{old} 代表目前這個迭代所在位置，x_{id}^{new} 則代表下一個世代新的位置，v_{id}^{old} 代表目前這個世代移動速度，v_{id}^{new} 則代表下一個世代新的移動速度。公式中 rand( )、Rand( )為兩個介於[0,1]的隨機變數。c1 及 c2 為正數的學習係數。w 為慣性權重，可使求解的過程中更快找到全域的最佳解，後續兩位學者更針對慣性權重提出較好的經驗設定[19,20]。從公式(2.4)中可看出粒子主要依據三個因子不斷更新搜尋方向，有了新的搜尋方向，粒子便可透過公式(2.5)從目前位置移動到新的位置，如此經過數次迭代後，粒子即可搜尋到最佳解。

近年來 PSO 已被廣泛運用到組合優化、神經網路訓練及複雜的系統模擬中。學者[2]以 PSO 應用揀貨作業的路徑規劃，並發展一路徑規劃啟發式演算法代入 PSO 的初始解，結果顯示以一個不錯的初始解代入 PSO 中，能有效減低演算法收斂時間，且與其它路徑規劃方法相較之下有相當不錯的績效表現。

3. 模式構建

3.1. 儲位指派

在儲位指派方面則運用學者[1]所提出的兩階段指派：第一階段將品項存取頻率依 ABC 儲存方法將儲位分為三個儲區，隨後依產品存取頻率的高與低依次指派離 I/O 點最近和遠；第二階段是將 ABC 三類品項分別以 Apriori 演算法決定產品關聯性，隨之再加以調整儲位。

3.2. 訂單批量方法

前期計畫已考量單一訂單、最適封包批量及關聯訂單批量三種不同訂單批量方法。而此研究再延伸加入新的批量方法來加以比較，其為應用粒子群最佳化演算法(Particle Swarm Optimization；PSO)，所發展出的一啟發式演算法進行訂單批次的規劃(稱為 PSO Batching-Method；簡稱 PSOBM)，其演算流程如圖 3.3，而步驟如下：

步驟一：設定 PSO 各項參數。其粒子數、慣性權重、學習因子(c1、c2)、最大速度(V_{max})、終止條件等設定如下：粒子數：粒子數設定為 30。慣性權重：慣性權重值(w)設為 0.8。學習因子：學習因子 c1、c2 設為 2。最大速度：設定速度 V_{id} 介於(-3,3)之間，最大速度設為 6。終止條件：迭代次數到達 200 次、全部粒子收斂至同一點或群體最佳位置 30 世代未改變，滿足以上任一條件則演算法終止。其中最大速度設為 6，是由於粒子大幅度的移動容易造成批次問題中不易收斂的結果，因而調整速度限制以防止不易收斂的情況產生。

步驟二：隨機產生訂單順序並以揀貨車容積限制作為批次考量，建立一批次初始解及隨機產生各粒子初始速度，迭代數為 0。

步驟三：依揀貨路徑規劃計算各粒子適應函數，並求得各個體與群體最佳位置(P_{best}、G_{best})。

步驟四：判斷適應值是否大於原先適應值來決定是否更新各別粒子之 P_{best} 與 G_{best}。

步驟五：根據目前位置與慣性權重、各個 P_{best} 與 G_{best} 修正粒子速度及訂單批次順序。

步驟六：若滿足終止條件，則進行步驟七，否則回步驟三，且迭代數加 1。

步驟七：輸出訂單批次結果。

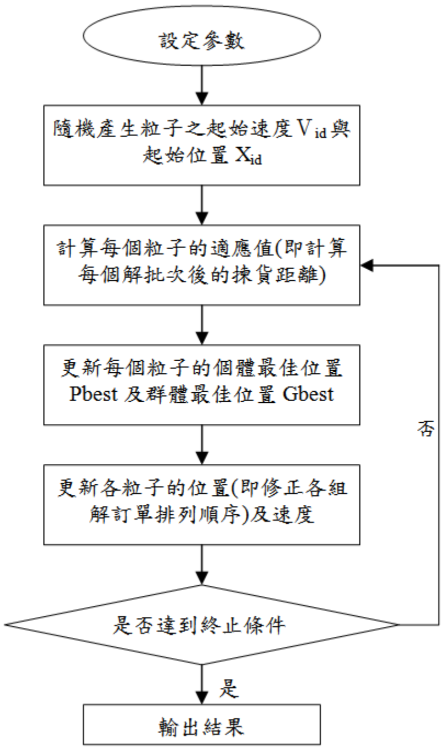


圖 3.3 PSOBM 演算流程圖

3.3. 揀貨路徑規劃

在路徑規劃上，本期較前期增加了學者[1]所提出的橫跨法改良策略，以及模擬退火法、基因演算法等兩種巨集式演算法。而以下將針對基因演算法與模擬退火法則運用於路徑規劃上作進一步介紹。

3.3.1. 模擬退火法

本研究是將演算法中的狀態視為所要撿取品項之揀取順序，而能量 E 則為在某揀貨順序下之揀貨距離。而其詳細路徑規劃流程如圖 3.6，至於步驟如下敘述：

步驟一： 隨機產生一揀貨順序 I_i，及計算其揀貨距離 $E(I_i)$

步驟二： 設定模擬退火參數。在初始溫度方面，本研究運用 Kirkpatrick 等人[12]所提出之建議，將 $Ts = \Delta E/\ln P$，P 為初始接受之機率，而太高之接受機率會使得執行時間加長，因此本研究設定 P 為 0.7，如此初始溫度則設定為 99，另外在其他參數方面，終止溫度(Te)為 10、迭代數為 100，冷卻率 α 則參考 Kirkpatrick 所提供介於 0.8~0.99 之間，此範圍可兼顧計算時間與求解品質，因此本實驗設定冷卻率為 0.8。

步驟三：根據揀貨順序 I_i 加以擾動，產生新的揀貨排序狀態 N_i，並計算其距離 $E(N_i)$，而本研究之擾動方法則運用任意兩工件互換法，此種擾動方式是將 I_i 的揀貨排序，隨機選取兩品項排序位置互換，如此便可產生新的揀貨順序 Ni。

步驟四：計算兩種揀貨順序之距離差距 $\Delta E = E(N_i) - E(I_i)$。

步驟五：若 $\Delta E < 0$，表示新的揀貨順序 N_i 距離較短，則取代順序 I_i，並跳至步驟 6。如果 $\Delta E \geqq 0$，則隨機產生一介於 0 到 1 的亂數值 X，若 $X < P(\Delta E,T)$，則揀貨順序 Ni 取代順序 I_i 為當前最佳解，並令 $I_i = N_i$；反之，則不接受 N_i 為新解。

步驟六：檢查是否達此溫度下搜尋次數 L(100)，若已搜尋 L 次則往步驟七；否則回到步驟三。

步驟七：執行退火程序 $Tn + 1 = \alpha * Tn$，並把迭代次數(L)歸零。而後檢查溫度是否小於終止溫度，如果等於或小於終止溫度則停止演算法；否則跳至步驟三繼續搜尋可行解。

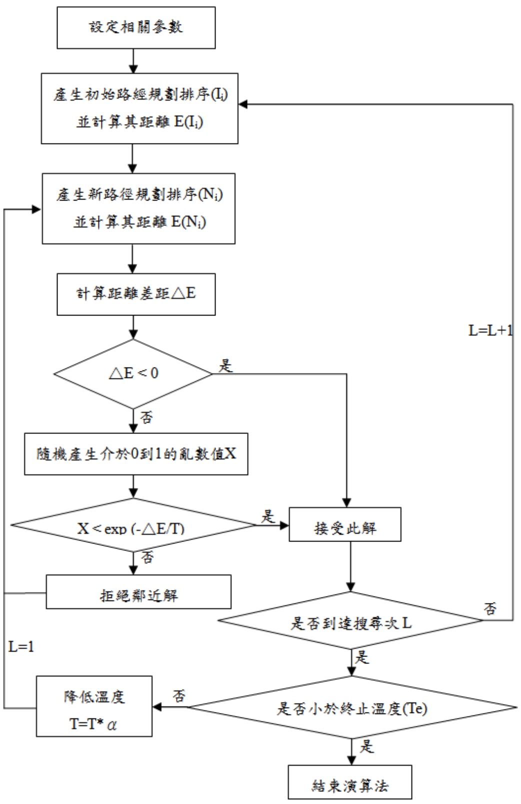


圖 3.6 模擬退火演算法於路徑規劃流程圖

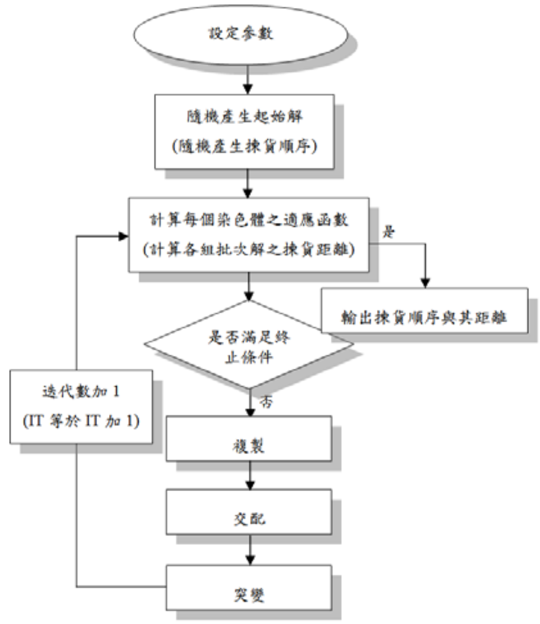


圖 3.7 應用基因演算法於路徑規劃流程圖

3.3.2. 基因演算法

本研究目的為降低揀貨作業時間及距離，因此適應函數則以路徑揀貨距離最小化為考量。而後便可經由適應值比較，將適應函數值較高的染色體複製到交配池中進行交配的作業，而適應值低者則遭淘汰，另外以競爭法決定被複製的染色體，假設複製機率為 0.5。然在交配方法上則是用兩點交配進行交配動作，產生兩個新的染色體。至於在突變方面，則設突變機率設為一極小值，且突變時為原染色體中之揀貨順序排序之品項兩點互換後，便可成為新的染色體。其步驟敘述如下，而圖 3.7 為應用基因演算法於路徑規劃流程圖：

步驟一：設定起始參數值，包括染色體數目、複製率、交配率、突變率與終止條件。

步驟二：隨機產生起始解，於揀貨路徑規劃問題中即為隨機產生揀貨順序。

步驟三：計算每個染色體的適應值，即計算各組揀貨順序之揀貨距離。

步驟四：檢查是否滿足終止條件。而本研究假設若滿足迭代次數為到達 200 次或其群體最佳位置 30 世代未改變，則至步驟六。否則跳至步驟五。

步驟五：依序執行複製、交配與突變的作業，結束後回步驟三。

步驟六：輸出揀貨路徑規劃順序與其距離。

4. 模擬驗證分析

4.1. 模擬環境與實驗因子組合

本實驗採一矩形物流中心，共有 10 條主走道，走道兩側各有 20 個儲位，共計 400 個儲位，儲位寬、深度皆為 1 公尺，倉儲前後方分別有前、後走道，中央有 1 條交叉走道，寬度皆為 2.5 公尺，領單點與集貨點(I/O) 皆於左下角，而揀貨車容積上限值為 50 單位。

本研究實驗考慮兩種儲位指派、一種產品關聯性、四種訂單批量及十種揀貨路徑規劃，共有 80 種排列組合。以 Visual Basic for Application 6.0 模擬軟體建構物流中心，由電腦產生

10,000 筆歷史訂單，每次實驗隨機從中抽取 100 張訂單，重複 30 次模擬實驗。以三項績效指標評估績效，分別為(1) 總揀貨距離、(2) 總揀貨時間與(3)揀貨車利用率，並將模擬結果利用 SPSS 12.0 統計軟體進行資料分析，期望能找出最佳實驗組合。

表 4.1 各實驗組合表

實驗編號	實驗因子組合	實驗編號	實驗因子組合
1	分區儲存*單一訂單* Cross$^+$ Strategy	41	分區儲存*關聯法批量* Cross$^+$ Strategy
2	不分區儲存*單一訂單*Cross$^+$Strategy	42	不分區儲存*關聯法批量*Cross$^+$ Strategy
3	分區儲存*單一訂單* MLI	43	分區儲存*關聯法批量* MLI
4	不分區儲存*單一訂單* MLI	44	不分區儲存*關聯法批量* MLI
5	分區儲存*單一訂單*SA	45	分區儲存*關聯法批量*SA
6	不分區儲存*單一訂單*SA	46	不分區儲存*關聯法批量*SA
7	分區儲存*單一訂單*PSO	47	分區儲存*關聯法批量*PSO
8	不分區儲存*單一訂單*PSO	48	不分區儲存*關聯法批量*PSO
9	分區儲存*單一訂單*GA	49	分區儲存*關聯法批量*GA
10	不分區儲存*單一訂單*GA	50	不分區儲存*關聯法批量*GA
11	分區儲存*單一訂單*AS	51	分區儲存*關聯法批量*AS
12	不分區儲存*單一訂單*AS	52	不分區儲存*關聯法批量*AS
13	分區儲存*單一訂單*MLI 結合 SA	53	分區儲存*關聯法批量*MLI 結合 SA
14	不分區儲存*單一訂單*MLI 結合 SA	54	不分區儲存*關聯法批量*MLI 結合 SA
15	分區儲存*單一訂單*MLI 結合 PSO	55	分區儲存*關聯法批量*MLI 結合 PSO
16	不分區儲存*單一訂單*MLI 結合 PSO	56	不分區儲存*關聯法批量*MLI 結合 PSO
17	分區儲存*單一訂單*MLI 結合 GA	57	分區儲存*關聯法批量*MLI 結合 GA
18	不分區儲存*單一訂單*MLI 結合 GA	58	不分區儲存*關聯法批量*MLI 結合 GA
19	分區儲存*單一訂單*MLI 結合 AS	59	分區儲存*關聯法批量*MLI 結合 AS
20	不分區儲存*單一訂單*MLI 結合 AS	60	不分區儲存*關聯法批量*MLI 結合 AS
21	分區儲存*最適封包批量* Cross$^+$ Strategy	61	分區儲存*PSOBM* Cross$^+$ Strategy
22	不分區儲存*最適封包批量*Cross$^+$Strategy	62	不分區儲存* PSOBM *Cross$^+$ Strategy
23	分區儲存*最適封包批量* MLI	63	分區儲存* PSOBM * MLI
24	不分區儲存*最適封包批量* MLI	64	不分區儲存* PSOBM * MLI
25	分區儲存*最適封包批量*SA	65	分區儲存* PSOBM *SA
26	不分區儲存*最適封包批量*SA	66	不分區儲存* PSOBM *SA
27	分區儲存*最適封包批量*PSO	67	分區儲存* PSOBM *PSO
28	不分區儲存*最適封包批量*PSO	68	不分區儲存* PSOBM *PSO
29	分區儲存*最適封包批量*GA	69	分區儲存* PSOBM *GA
30	不分區儲存*最適封包批量*GA	70	不分區儲存* PSOBM *GA
31	分區儲存*最適封包批量*AS	71	分區儲存* PSOBM *AS
32	不分區儲存*最適封包批量*AS	72	不分區儲存* PSOBM *AS
33	分區儲存*最適封包批量*MLI 結合 SA	73	分區儲存* PSOBM *MLI 結合 SA
34	不分區儲存*最適封包批量*MLI 結合 SA	74	不分區儲存* PSOBM *MLI 結合 SA
35	分區儲存*最適封包批量*MLI 結合 PSO	75	分區儲存* PSOBM *MLI 結合 PSO
36	不分區儲存*最適封包批量*MLI 結合 PSO	76	不分區儲存* PSOBM *MLI 結合 PSO
37	分區儲存*最適封包批量*MLI 結合 GA	77	分區儲存* PSOBM *MLI 結合 GA
38	不分區儲存*最適封包批量*MLI 結合 GA	78	不分區儲存* PSOBM *MLI 結合 GA
39	分區儲存*最適封包批量*MLI 結合 AS	79	分區儲存* PSOBM *MLI 結合 AS
40	不分區儲存*最適封包批量*MLI 結合 AS	80	不分區儲存* PSOBM *MLI 結合 AS

4.2. 實驗結果分析

本研究將探討各因子之不同水準於不同績效指標下是否具有顯著差異，並以 95%信賴水準進行變異數分析，分析是否達顯著差異，若有顯著差異，將再以 Duncan 分群作事後多重比較檢定，以瞭解各因子水準間差異型態為何。從表 4.2 中可得知不同儲位指派、訂單批量、揀貨路徑規劃對於揀貨距離與揀貨時間皆有顯著的影響，不過在揀貨車利用率方面只有訂單批量策略是有顯著差異。而從表 4.2 得知，在儲位指派方面，分區儲存明顯優於不分區儲存；訂單批量決策上，在揀貨距離與時間上，整體是以關聯訂單與最適封包批量法則表現最佳，而在揀貨車利用率上則是以關聯訂單批量表現最好，其次是 PSOBM。在揀貨路徑規劃上，發現其 4 種的巨集啟發式演算法中，AS 在路徑規劃上不論在揀貨距離或時間上，皆比起 GA、SA 與 PSO 還佳，至於表現最差為 PSO 演算法。如果當以 MLI 為初始解時，仍是 AS 表現最佳，且當最大迴圈插入法結合 4 種巨集式啟發式演算法時，可以有效提升 4 種啟發式演算法之績效。因此可證實適當的加入初始解將可有效提升演算法之求解品質與效率。

表4.2 各因子水準於三項績效指標之平均數

	來源	揀貨距離	揀貨時間	揀貨車利用率
儲存策略	分區儲存	5746.76	8763.61	—
	不分區儲存	12176.13	15192.98	—
訂單批量	關聯訂單批量	7602.1(1)	10619.7(1)	0.9219(1)
	最適封包批量	7829.26(1)(2)	10845.86(1)(2)	0.8471(3)
	PSOBM	8077.87(2)	11094.47(2)	0.8796(2)
	單一訂單	12336.54(3)	15353.14(3)	0.3017(4)
揀貨路徑規劃	MLI結合AS	6855.44(1)	9872.29(1)	—
	MLI	6885.95(1)	9902.8(1)	—
	AS	7747.4(2)	10764.25(2)	—
	MLI結合SA	8220.8(2)(3)	11237.65(2)(3)	—
	MLI結合GA	8741.32(3)(4)	11758.17(3)(4)	—
	MLI結合PSO	9206.64(4)(5)	12223.49(4)(5)	—
	Cross+ strategy	9729.97(5)(6)	12746.82(5)(6)	—
	GA	10020.22(6)	13037.076)	—
	SA	10770.46(7)	13787.31(7)	—
	PSO	11436.26(7)	14453.11(7)	—

註： — 符號表示無顯著差異

()內數字表示 Duncan 分群結果

網底表示為最佳績效

從表 4.3 中可知，在儲位指派、訂單批量與揀貨路徑規劃三因子交互作用下，於揀貨路徑與揀貨時間績效上，以實驗編號 39 和 23 表現最佳，即為分區儲存搭配最適封包批量與結合 MLI 或 MLI+AS。而根據表 4.3 發現關聯法則並非表 4.2 所呈現，在揀貨距離與時間上為最佳的批次演算法，其主要原因為在分區儲存的情況下，最適封包不論在何路徑組合下皆優於關聯批量，但在不分區儲存的情況下，由於最適封包是以品項所座落的走道來形成訂單批量，因此當在隨機儲存品項時，導致最適封包批量的方法劣於關聯法則批量，其詳細結果本研究以揀貨距離為例呈現於表 4.4，如此造成最適封包批量整體平均效率劣於關聯訂單批量。

另外在路徑規劃上也發生此情形，以揀貨距離之 PSO 演算法為例，從表 4.5 可發現當儲位指派為不分區儲存時，PSO 演算法在 4 種的批量手法上所花的旅行距離最多；不過在分區儲存時，除了單一批量外，其他三種之批次手法中表現最差皆為演算法 SA，綜合以上結果得知，較差之儲存與訂單批量策略使得 PSO 演算法相較於其他演算法求解品質大幅降低。本研究對於此結果作以下推論：由於此 4 種演算法有著不同的求解特性與模式，當環境或揀貨策略較差時，模擬退火法求解過程，是運用一組初始解不斷迭代更新產生新解。基因演算法則是利用挑選最優秀(適應值最佳)進行複製，而後利用輪盤法取兩染色體做隨機兩點交配，如此反覆迭代找尋最佳解。由上述可知 SA 與 GA 求解時之搜尋擾動方式較為簡單。而螞蟻理論與粒子群演算法，則是運用所設定之螞蟻與粒子數發散式搜尋而產生數個初始解，因此搜尋範圍之設定相對於此兩演算法相當重要，其中螞蟻理論是運用蒸發係數與公式(2.1)來控制螞蟻朝向整體性最佳路徑前進；至於粒子群演算法則是運用速度控制搜尋。在揀貨路徑規劃，不管何種演算法，都是在搜尋最佳的揀貨順序，本研究發現粒子群演算法在產生揀貨順序時，其擾動方式導致在揀貨品項排序時變化較其他演算法大，如此容易造成在限定的迭代次數裡，粒子群搜尋許多種組合，但尚未求得至最佳組合。因此，可以推論粒子群演算法在較差環境時，因為粒子演算法較大發散搜尋方式使得收斂速度較其他演算法緩慢。

有鑒於最差之策略會影響整體平均績效，本研究挑出對於各路徑規劃最有利之組合，進一步清楚比較 4 種巨集啟發式演算法、MLI 與 $Cross^{+}$ Strategy 等路徑規劃之優劣，至於在 4 種巨集啟發式演算法上，由於已證明加入 MLI 當起始解將有助於求解品質，因此將 MLI 結合每種巨集演算法進行比較，如此可得表 4.6。經由 4.6 得知，MLI 與 AS+MLI 之路徑規劃無庸置疑為最佳解，其次為 GA+MLI 與 PSO+MLI，而表現最差的為 SA+MLI。因此本研究所介紹之 4 種巨集啟發式演算法運用於路徑上，以螞蟻演法表現最佳，其次為粒子群演算法

與基因演算法，而表現最差為模擬退火法。

表4.3 各實驗組合於各績效指標之Duncan Test

總揀貨距離						揀貨時間						利用率					
實驗編號	平均數	集合	實驗編號	平均數	集合	實驗編號	平均數	集合	實驗編號	平均數	集合	實驗編號	平均數	集合	實驗編號	平均數	集合
39	2841	1	74	7867	20	39	5857	1	74	10884	19	1	0.301	1	61	0.879	3
23	2849	1	80	8229	21	23	5865	1	80	11245	20	2	0.301	1	62	0.879	3
59	3205	2	64	8263	21	59	6223	2	64	11280	20	3	0.301	1	63	0.879	3
43	3217	2	40	8366	21,22	43	6234	2	40	11383	20,21	4	0.301	1	64	0.879	3
79	3220	2	24	8399	21,22	79	6237	2	24	11415	20,21	5	0.301	1	65	0.879	3
63	3236	2	9	8418	21,22	63	6252	2	9	11435	20,21	6	0.301	1	66	0.879	3
31	3361	2	52	8494	22	31	6378	2	52	1512	21	7	0.301	1	67	0.879	3
51	3722	3	7	8807	23	51	6740	3	7	11823	22	8	0.301	1	68	0.879	3
71	3790	3	72	9084	24	71	6806	3	72	12101	23	9	0.301	1	69	0.879	3
37	4108	4	32	9288	25	37	7125	4	32	12304	24	10	0.301	1	70	0.879	3
35	4162	4	5	10340	26	35	7179	4	5	13356	25	11	0.301	1	71	0.879	3
57	4431	5	13	10375	26	57	7449	5	13	13391	25	12	0.301	1	72	0.879	3
21	4449	5	58	10508	26	21	7466	5,6	58	13526	25	13	0.301	1	73	0.879	3
41	4574	5,6	14	10957	27	41	7592	5,6,7	14	13973	26	14	0.301	1	74	0.879	3
55	4608	5,6,7	38	10984	27	55	7625	5,6,7	38	14001	26	15	0.301	1	75	0.879	3
77	4655	6,7,8	56	11248	28	77	7671	6,7,8	56	14265	27	16	0.301	1	76	0.879	3
61	4763	7,8	46	11269	28	61	7780	7,8	46	14287	27	17	0.301	1	77	0.879	3
75	4828	8	78	11285	28	75	7844	8	78	14302	27	18	0.301	1	78	0.879	3
29	5823	9	50	11299	28	29	8840	9	50	14317	27	19	0.301	1	79	0.879	3
49	6070	10	36	11754	29	49	9087	10	36	14770	28	20	0.301	1	80	0.879	3
19	6175	10,11	66	12056	30	19	9192	10,11	66	15073	29	21	0.847	2	41	0.922	4
3	6215	10,11	26	12077	30	3	9232	10,11	26	15094	29	22	0.847	2	42	0.922	4
1	6289	11	42	12108	30	1	9306	11	42	15126	29	23	0.847	2	43	0.922	4
69	6328	11	30	12114	30	69	9344	11	30	15130	29	24	0.847	2	44	0.922	4
27	6678	12	70	12131	30	27	9694	12	70	15148	29	25	0.847	2	45	0.922	4
47	6784	12,13	76	12152	30	47	9802	12	76	15169	29	26	0.847	2	46	0.922	4
15	6880	13,14	62	13112	31	15	9897	12,13,14	62	16129	30	27	0.847	2	47	0.922	4
17	6885	13,14	22	13167	31	17	9901	12,13,14	22	16184	30	28	0.847	2	48	0.922	4
11	7018	14,15	48	13440	32	11	10035	13,14	48	16457	31	29	0.847	2	49	0.922	4
33	7064	14,15	68	14367	33	33	10081	14,15	68	17384	32	30	0.847	2	50	0.922	4
53	7143	15,16	28	14371	33	53	10160	14,15	28	17388	32	31	0.847	2	51	0.922	4
67	7162	15,16	20	15117	34	67	10179	14,15	20	18133	33	32	0.847	2	52	0.922	4
25	7177	15,16	4	15187	34	25	10194	14,15	4	18204	33	33	0.847	2	53	0.922	4
45	7193	15,16	18	17074	35	45	10210	14,15	18	20090	34	34	0.847	2	54	0.922	4
54	7315	16	12	17221	35	54	10333	15,16	12	20238	34	35	0.847	2	55	0.922	4
73	7494	17	10	17979	36	73	10510	16,17	10	20996	35	36	0.847	2	56	0.922	4
65	7536	17,18	16	18022	36	65	10552	17,18	16	21039	35	37	0.847	2	57	0.922	4
34	7552	17,18,19	6	18516	37	34	10569	17,18	6	21533	36	38	0.847	2	58	0.922	4
60	7691	18,19,20	2	19377	38	60	10709	17,18,19	2	22393	37	39	0.847	2	59	0.922	4
44	7722	19,20	8	19881	39	44	10740	18,19	8	22897	38	40	0.847	2	60	0.922	4

表4.4 不同揀貨路徑與儲存政策於各訂單批量方法之揀貨距離表現

	Cross+ strategy		MLI		SA		PSO		GA	
	分區	不分區	分區	不分區	分區	不分區	分區	不分區	分區	不分區
訂單批量	集合	集合	集合	集合	集合	集合	集合	集合	集合	集合
單一訂單	11	38	10,11	34	26	37	23	39	21,22	36
最適封包	5	31	1	21,22	15,16	30	12	33	9	30
關聯法則	5,6	30	2	19,20	15,16	28	12,13	32	10	28
PSOBM	7,8	31	2	21	17,18	30	15,16	33	11	30
	AS		**SA+MLI**		**PSO+MLI**		**GA+MLI**		**AS+MLI**	
	分區	不分區	分區	不分區	分區	不分區	分區	不分區	分區	不分區
訂單批量	集合	集合	集合	集合	集合	集合	集合	集合	集合	集合
單一訂單	14,15	35	26	27	13,14	36	13,14	35	10,11	34
最適封包	2	25	14,15	17,18,19	4	29	4	27	1	21,22
關聯法則	3	22	15,16	16	5	28	5	26	2	18,19,20
PSOBM	3	24	17	20	6,7,8	30	6,7,8	28	2	21

註：灰色標記表示為排名最佳之訂單批量

表 4.5 不同訂單批量與儲存政策於各揀貨路徑方法之揀貨距離表現

最適封包批量				關聯法則批量				PSOBM				單一批量			
分區		不分區		分區		不分區		分區		不分區		分區		不分區	
路徑	排名	路徑	排名	路徑	排名	路徑	排名	路徑	排名	路徑	排名	路徑	排名	路徑	排名
AS+MLI	1	SA+MLI	17,18,19	AS+MLI	2	SA+MLI	22	AS+MLI	2	SA+MLI	20	AS+MLI	10,11	SA+MLI	25
MLI	1	AS+MLI	21,22	MLI	2	MLI	19,20	MLI	2	MLI	21	MLI	10,11	MLI	27
AS	2	MLI	21,22	AS	3	AS+MLI	18,19,20	AS	3	AS+MLI	21	Cross+ strategy	11	AS+MLI	34
GA+MLI	4	AS	25	GA+MLI	5	AS	22	GA+MLI	6,7,8	AS	24	PSO+MLI	13,14	AS	35
PSO+MLI	4	GA+MLI	27	Cross+ strategy	5,6	GA+MLI	26	Cross+ strategy	7,8	GA+MLI	28	GA+MLI	13,14	GA+MLI	35
Cross+ strategy	5	PSO+MLI	29	PSO+MLI	5,6,7	GA	28	PSO+ MLI	8	PSO+MLI	30	AS	14,15	GA	36
GA	9	SA	30	GA	10	PSO+MLI	28	GA	11	SA	30	GA	21,22	PSO+MLI	36
PSO	12	GA	30	PSO	12,13	SA	28	PSO	15,16	GA	30	SA	26	SA	37
SA+MLI	14,15	Cross+ strategy	31	SA+MLI	14,15	Cross+ strategy	30	SA+MLI	17	Cross+ strategy	31	SA+MLI	26	Cross+ strategy	38
SA	15,16	PSO	33	SA	15,16	PSO	32	SA	17,18	PSO	33	PSO	27	PSO	39

註：灰色標記表示為排名最佳之路徑規劃方法
框框為排名最差之路徑規劃法

表 4.6 各路徑演算法下之最佳揀貨距離表現組合

組合	平均數	集合
分區儲存*最適封包批量*AS+MLI	2841	1
分區儲存*最適封包批量*MLI	2849	1
分區儲存*最適封包批量*GA+MLI	4108	4
分區儲存*最適封包批量*PSO+MLI	4162	4
分區儲存*最適封包批量*Cross+ Strategy	4449	5
分區儲存*最適封包批量*SA+MLI	7064	14,15

註：灰色標記表示為排名最佳之訂單批量

5. 結論與建議

本研究之研究目的乃在探討如何提高物流中心檢貨系統之揀貨效率，因影響平均訂單揀貨績效之因子並非僅揀貨路徑方法，因此本研究與過去學者所提出最佳之揀貨路徑方法作比較，以及考量訂單批量法、揀貨型態，並驗證在各因子間之組合以找出最佳組合，因此根據本研究之模擬實驗所蒐集之數據，經過統計分析，可以得到下面的幾點結論：

一、本研究嘗試將過去學者常用之螞蟻理論、粒子群演算法、基因演算法、模擬退火法等巨集啟發式演算運用於路徑規劃上，從模擬之結果發現，螞蟻理論在路徑規劃上不論在揀貨距離或時間上，皆比起粒子演算法、基因演算、模擬退火法還佳。另外發現當組合因子為較差之儲存與訂單批量策略時，PSO 演算法因為搜尋方式較發散，導致在較差決策時收斂情形略遜於其他演算法。因此在設計相關倉儲環境與相關決策時，必須更加全面考量，否則將影響後續作業之規劃。

二、再次驗證了最大迴圈插入法(MLI)，確實能有相當不錯之績效，且當最大迴圈插入法結合螞蟻理論、粒子群演算法、基因演算法、模擬退火法等 4 種巨集啟發式演算法時，可以有效提升 4 種演算法之績效。這也更加說明初始解對於巨集啟發式演算法的重要性。

三、經由模擬結果顯示任一訂單批量法於各績效指標皆優於單一訂單，其中又以最適封包批量法為最佳，且與其他因子之組合皆有最佳之表現。但在揀貨車利用上關聯法為最佳的訂單批量方法，其次為 PSOBM。

四、從三因子組合分析下，更加證實分區儲位指派優於未經規劃之不分區儲存手法，且經

模擬得到表現最好之組合，其為分區儲存配合最適封包與路徑規劃 MLI，以及分區儲存搭配最適封包與螞蟻理論結合 MLI 之路徑規劃。

五、本研究為符合實務上物流中心實際情況，於倉儲環境佈置中加入交叉走道，考慮之設定囊括儲位指派、訂單揀取法與揀貨路徑規劃，使整體環境更符合實際狀況，並考慮三項績效指標，經模擬實驗驗證各實驗組合之績效，冀望本研究之結果可提供物流中心業者於倉儲規劃設計與績效改善時參考。

6. 參考文獻

1. 謝玲芬、林嘉慶，「倉儲系統撿貨策略與儲區交叉走道配置之研究」，2003 中國工業工程學會年會暨學術研討會，彰化建國技術學院。
2. 謝玲芬、黃建霖，「物流中心之粒子群最佳化演算法應用效率之提昇」，2006 中國工業工程學會年會暨學術研討會，台中東海大學。
3. Agrawal, R., T. Imielinski, and Swami, A. (1993), "Mining Association Rules between Sets of Items in Large Data Bases," In Proceedings of the ACM SIGMOD Conference in Management of Data, Washington DC, USA.
4. Chen, M. C., and Wu, H. P., (2005), "An association-based clustering approach to order batching considering customer demand patterns", Omega., Vol. 33, No. 4, pp. 333-343.
5. Coyle, J. J., E. J. Bardi, and Langley, C. J. (1996), "The Management of Business Logistics," St Paul, MN: West.
6. Dorigo, M. (1992), "Optimization, Learning and Natural Algorithms," Ph.D. Thesis. Politecnico di Milano, Italy, EU.
7. Grefenstette, J. J., "Optimization of Control Parameters for Genetic Algorithms," IEEE Transactions on Systems, man & cybernetics, pp.122-128, 1986.
8. Han, J., and Fu, Y. (1999), "Mining Multiple-Level Association Rules from Large Database," IEEE Transaction on Knowledge and Data Engineering, Vol. 11, No. 5, pp.1-8.
9. Han, J., and Kamber, M. (2000), "Data Mining: Concepts and Techniques," Morgan Kaufmann Publishers, San Franscisco.
10. Hsu, C. M., K. Y., Chen and Chen, M. C., (2005) "Batching orders in warehouses by minimizing travel distance with genetic algorithms", Computers in Industry, Vol. 56, No. 2, pp. 169-178.
11. Kennedy, J. and Eberhart, R.C. (1995), "Particle Swarm Optimization," Proceedings IEEE International Conference on Neural Networks, Vol. 4, pp.1942-1948.
12. Kirkpatrick, S., Gelatt, C.D.Jr., and Vecchi, M.P., (1983), "Optimization by simulated annealing," Science, Vol. 220, pp. 671-680.
13. Lin, C.H., and Lu, I.Y. (1999), "The Procedure of Determining the Order Picking Strategies in Distribution Center," International Journal Production Economics, Vol. 60, No. 61, pp. 301–307.
14. Malmborg, C. J. (1996), "A Genetic Algorithm for Service Level Based Vehicle Routing Scheduling", European Journal of Operation Research, Vol.93, pp.121–134.
15. Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., Teller, E., (1953), "Equation of state calculations by fast computing machines," Journal of Chemical Physics, Vol.21, pp.1087–1092.
16. Petersen, C.G. and Schmenner, R.W. (1999), "An evaluation of routing and volume-based storage policies in an order picking operation," Decision Science, Vol. 30, No.2, pp. 481–501.
17. Roodbergen, K. J., and Koster, R. D. (2001), "Routing Method for Warehouse with Multiple Aisles," International Journal of Production Research, Vol. 39, No. 9, pp.1865-1883.
18. Ruben, R. A. and Jacobs, I. R. (1999), "Batch Construction Heuristics and Storage Assignment Strategies for Walk/Ride and Pick Systems," Management Science, Vol. 45, No. 4, pp.575-594.
19. Shi, Y. and Eberhart, R., (1999), "Empirical Study of Particle Swarm Optimization," Proceedings of the 1999 Congress on Evolutionary Computation, Washington, DC, USA, Vol. 3, pp. 1945-1950.
20. Shi, Y. H. and Eberhart, R. C. (1998), "Parameter Selection in Particle Swarm Optimization," The 7th Annual Conference on Evolutionary Programming, San Diego, pp. 591-600.
21. Shi, Y. H. and Eberhart, R. C. (1998), "A Modified Particle Swarm Optimizer," Proceedings of IEEE International Conference on Evolutionary Computation, Anchorage, Alaska, pp. 69-73.
22. Simoudis, E. (1996), "Reality Check for Data Mining," Institute of Electrical and Electronic Engineers Expert, Vol. 11, No. 5, pp.26-33.
23. Tompkins, J. A., J. A. White, Y. A. Bozer, E. H. Fraz elle, J. M. A. Tanchoco, and Trevino, J.

(1996), “Facilities Planning,” Wiley New York.

計畫成果自評

本研究主要針對物流中心儲位規劃問題，延續上一期計畫之研究結果，考慮分區與不分區兩種方式，並藉由關聯法則進一步改善儲位指派，在訂單揀取法中比較單一訂單別、最適封包批量與關聯訂單批量，在揀貨路徑規劃部分，本研究將運用近幾年才提出之粒子群最佳化演算法求解最佳路徑，並建構一啟發式演算法作為粒子群最佳化演算法之起始解，稱為最大迴圈插入法(Maximum Loop Insertion, 簡稱MLI)以提昇求解效率，並與文獻中有較佳表現之巨集演算法(包括：蝎蟻理論、基因演算法、模擬退火法及粒子群最佳化演算法)與最短旅行迴圈啟發式演算法一同比較。以總揀貨成本最小化為目標，進而達到單位時間內揀貨績效提昇，透過模擬實驗驗證並以三項績效指標(最小化旅行距離、最小化旅行時間及最大化揀貨車之負載量)評估各因子水準組合，利用統計分析手法找出各因子之最佳水準組合，並將本研究之成果提供給物流業者用於物流規劃與績效提昇之參考。完全依照計畫書的想法進行，目前已經全部完成，研究結果與原預期之計畫目標相符合，而且整理完手稿，投稿國際期刊中。在計畫執行期間曾參與國際研討會(ICOTA 7)，國際會議心得報告及發表之論文詳見附錄。

附錄：出席 ICOTA 7 國際會議心得報告及發表之論文

The 7th International Conference on Optimization: Techniques and Applications (ICOTA 7)

Kobe International Conference Center, Kobe, Japan

December 12-15, 2007

謝玲芬

中華大學科技管理學系

一、會議經過

The 7th International Conference on Optimization: Techniques and Applications (ICOTA 7)是探討最佳化理論領域一個相當正式的研討會，每三年舉辦一次，其主要目的是提供所有在產、關、學及軟體開發者一個國際聚會的機會，大家齊聚一堂彼此交換構想、分享經驗及心得，並藉由溝通及討論，研擬未來之發展方向。2007 年是第七屆，由十二月十二日至十五日共四天，地點選在日本的神戶。每屆發表論文的人數都相當多，今年共有來自32個國家，254篇論文發表，大會並特別安排六場精彩的專題演講，參加本次研討討會真是受益良多。大會共分為六十餘個場次發表論文，每個場次約四至五篇論文，讓與會者能充份挑選自己有興趣的主題之場次參與討論。大會主辦單位在閉幕餐會上除了供應豐盛的餐點外，並安排傳統的日本歌舞表演，整個研討會到此在一片歡樂中圓滿閉幕。

由於十二月中旬參加研討會，恰逢聖誕前之前，神戶街頭相當熱鬧，趁著研討會的空檔，親身搭乘日本的電車，逛逛當地的商場，順道體驗當地的民俗風情。且神戶當地為紀念之前的阪神大地震，為祈求神戶居民之平安，當地每年十二月均舉辦燈祭，以祈求人民的平安及世界和平，場面相當壯觀！當地居民告訴我們今年是舉辦燈祭的最後一年！大家更是珍惜此一機會，留下深刻的記憶。另外我也品嚐的當地著名且道地的大阪燒，參觀當地的古蹟－姬路城，真是收穫良多。

神戶燈祭(1)

神戶燈祭(2)

神戶鶴橋風月的大阪燒

神戶古蹟－姬路城

神戶港

日本歌舞表演

論文發表會場

二、研討會發表之論文

Optimal Order Picking Planning for Distribution Center with Cross Aisle

Ling-Feng Hsieh

Professor, Department of Technology Management, Chung Hua University, Hsinchu 300, Taiwan, R.O.C.

lfhsieh@chu.edu.tw

Chien-Lin Huang

Graduate student, Department of Technology Management, Chung Hua University, Hsinchu 300, Taiwan, R.O.C.

m09403033@chu.edu.tw

Abstract

Order picking method is one of the most important operations in the distribution center. The route planning of order picking systems will allow for the possibilities of increasing in production efficiency, reducing the operation cost in distribution center, and improving the corporation competitiveness. In a distribution center with cross aisle, although the cross aisle layout may reduce the order picking distance, it also may raise the complexity for picking routing planning. Focus on this problem, a heuristic algorithm (called Maximum Loop Insertion) is proposed in this paper, as well as compare with other famous algorithms and Particle Swarm Optimization (PSO), in order to improve the order picking performance. According to the simulation experiment, it verified that the Maximum Loop Insertion algorithm actually achieves the better performance. Overall, the result of this research will enhance the best route planning of order picking systems in distribution center and provide the industry as a reference in the warehouse design in the future.

Keywords: Order picking system, Cross aisle, Maximum Loop Insertion, Particle Swarm Optimization, Performance

1 Introduction

As economy develops and changes in consuming habit, it makes the types and structures of marketing channel are transferred to the supplier transported the products to retailer by distribution center. Therefore, it is an important issue about how to improve the distribution center operation efficiency. Due to the fact that consumer's request has been changed from few items and small volume to many items and small volume, it is hoped that if the order picking operations can be finished in reasonable time or not will influence the operation cost and service level of distribution center. In internal operations of distribution center, the factors included warehouse layout, storage assignment strategies, order picking operations, which will influence order picking system. The order picking operations is an important complex work. Coyle et al. [5] and Tompkins et al. [3] proposed that the order picking operations of the system takes more than 65% of the whole warehouse operation cost. In addition, the traveling time takes about 50% of the order picking activities. Therefore, it expected reducing the traveling distance by planning the order picking routing, in order to improve the whole operation efficiency in distribution center.

In order to improve order picking routing planning, we proposed a new heuristic algorithm in this paper. We also expect it can improve the picking performance. In the same time, we apply Particle Swarm Optimization (PSO) in order picking routing, and compared with the best order picking method in literature review. In this paper, it used eM-plant to construct the warehouse system, and verify the order picking method which is proposed in this paper can has the better performance. The performance index to evaluate each order picking method is average total order picking distance and average total CPU run time. Finally, it is expected this study can consult to the industries.

2 Literature Review

2.1 Route Planning of Order Picking System

According to the well order picking routing planning, it can make the minimize order picking distance, reducing order picking time, and improve order picking performance. Hall [12] supposed that did not consider the width of the aisle, as well as evaluate and compare with different order picking strategies. The order picking strategies included Traversal, Midpoint Return, Largest Gap Return, and the best performance is using Largest Gap Return. Petersen and Schmenner [1] had two major policy decisions that determine the efficiency of order picking operations, which are storage policies and routing policies. It also considered Transversal Strategy, Return Strategy, Midpoint Strategy, Largest Gap Strategy, and Composite Strategy; five different kinds of order picking routing strategies, as well as compare with the optimal solution, the Composite Strategy have the best performance. Ho and Su [13] proposed two kind of heuristic algorithms, which are Nearest Center of Rectangular Insertion (NCRI) and Minimum Traveling Loop Insertion (MTLI). He also compared with previous scholar's method, for example, Largest Gap Strategy, Nearest Center of Geometry Insertion Heuristic, the results shown that the two heuristic algorithms which is proposed by author has the better performance.

Ratliff and Rosenthal [2] discuss the influence of add the cross aisle of order picking routing, the results shown that the order picking routing planning will become more complex in warehouse which has cross aisles. Roodbergen and Koster [7] focused on estimate variety kinds of order picking strategy to compare different number of aisles, different number of items and different width of aisles. The algorithms to evaluate the shortest path included S-shape Heuristic, Largest Gap Heuristic, Aisle-by-aisle Heuristic, Optimal Algorithm, Combined Heuristic, and Combine$^+$ Heuristic. The best performance is Combine$^+$ strategy. Hsieh et al. [8] applying PSO in order picking routing and storage assignment, and compare with previous scholar, it verified that applying PSO has the better performance.

2.2 Particle Swarm Optimization

Kennedy, J. and R. C. Eberhart [6] proposed Particle Swarm Optimization (PSO) in 1995. It is similar to John Holland [10] proposed Genetic Algorithms (GA) in 1960. They are all belonging to Evolutionary computation and within the evolutionary generation to optimal solutions. The main concept of GA is the survival of the fittest which is proposed by Charles Darwin. That is using three basic operations, which is Reproduction, Crossover, and Mutation to imitate natural evolutionary process, and according to the evolutionary generation to optimal solutions. Therefore, PSO has no crossover and mutation, it is easier than GA, but is has better global optimal solution ability.

PSO and Dorigo [10] proposed Ant Colony Optimization (ACO) in 1992 is all Swarm Intelligence algorithms that are according to swarm intelligence to solve problems. ACO is a cooperative heuristic searching algorithm inspired by the methodological study on the behavior of ants. The ants can find out the food is done by an indirect communication known as pheromone, left by the ants on the paths, and constructive to the shortest distance between the nest and food.

PSO is according to three factors to find out the optimal solution, that is (1) the current moving direction by itself, (2) the previous experiment by itself, (3) the swarm experiments, and compare with Fitness Value which is computed by Fitness Function to revise the velocity and position of itself.

The definitions of PSO related variances, X_{id}^{l} is the particle i, d dimension, l stage position. X_{id}^{l+1} is the particle i, d dimension, in $l+1$ stage position. P_{id} represents the optimum position recorded by the i^{th} particle in d dimension. P_{gd} is the optimum position resolved by a population of particles in d dimension. V_{id}^{l} is the velocity of the i^{th} particle, d dimension in l stage. V_{id}^{l+1} is the velocity of the ith particle, d dimension in $l+1$ stage. $rand()$ is random number between [0, 1]. c_1 and c_2 are learning factors which controls the acceleration of particle velocity. w is the inertial constant that allows user to control the parameters. A small w value will direct searches within current space, and a large w value will indicate searches in new space. Appropriate selection of w value, c_1 and c_2 learning factors can expand search space to achieve a balanced result. The velocity and position update formula is shown in Formula (1) and Formula (2).

$$V_{id}^{l+1} = w \cdot V_{id}^{l} + c_1 \cdot rand() \cdot (P_{id} - X_{id}^{l}) + c_2 \times rand() \cdot (P_{gd} - X_{id}^{l}) \qquad (1)$$

$$X_{id}^{l+1} = X_{id}^{l} + V_{id}^{l+1} \qquad (2)$$

The original two scholars which is proposed PSO is not using inertia weight w. The inertia weight w is proposed by Shiand and Eberhart [11] in 1998, illustrated inertia weight w using can make the solution process to find out the global best solution faster. The characteristic of inertia weight w is similar to cooling parameter of Simulated Annealing (SA) that can make the solution become convergence. Shi and Eberhart also illustrates w between 0.8 and 1.2, it has more chance to find out the global solution.

3 Model Construction

3.1 Model Constructing

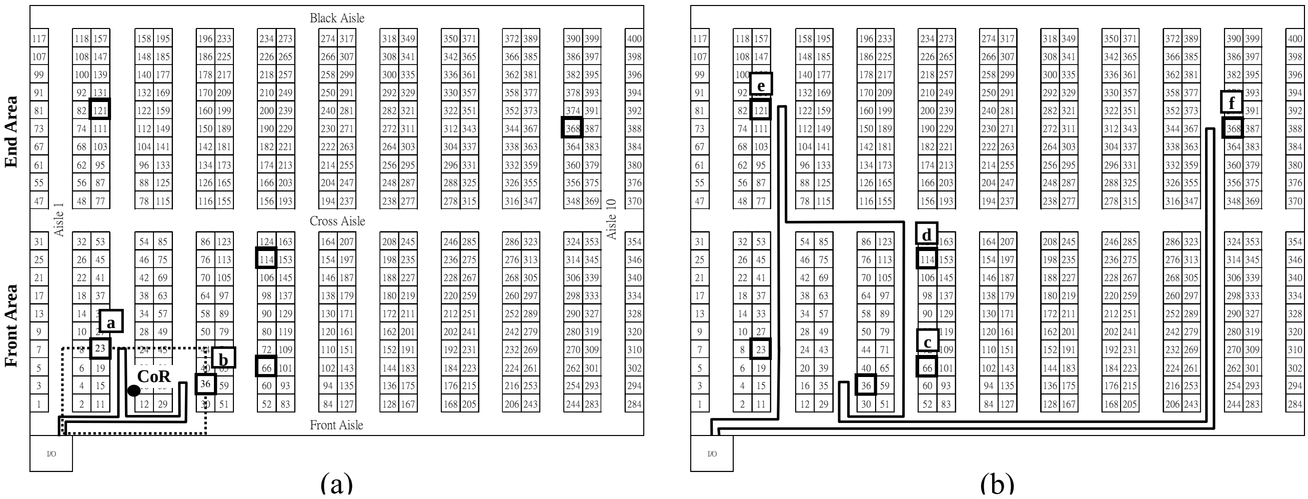
This research probes into the warehouse environment layout is shown in Fig. 1. There are 10 aisles, in each aisle of left hand side and right hand side all have 20 storage locations, so total locations are 400. Suppose the depth of storage is 1m, the width is 1m; the main aisle width is 2.5m, sub-aisle length is 10m, and width is 2.5m. There has front aisle, end aisle, and 1 cross aisle, the width all are 2.5m. The input depot and output depot is the same point in the front of left side. The experiment assumes that order pickers begin the tour at the input depot and end at the output depot. Thus, upon completing the retrieval of the order, the order picker immediately begins retrieval of the next order released. In order to show actual situation, the rectilinear distance is considered for calculation, and give the each locations a number according to the distance between the locations to I/O depot.

Because the well storage strategy can reduce the moving distance between in warehouse and out warehouse, reducing operation time, and full using storage space. Therefore, it adapted classification storage, put the high frequency produce near the I/O depot, and put the low frequency product far away I/O depot in this paper. It can reducing the order picking distance, and improves the order picking efficiency. The access frequency is adapted 80/20 method, the meaning is 20% products is own order picking activity 80%. For this reason, we put these 20% products near the I/O depot.

The part of order batching is adapted the single order method. This method is the general method in industry. The advantage of this method is can reduce the complex of order picking, this is different order batching which has more consuming time to separate the combine order products.

3.2 Route Planning of Order Picking System

The main discussion of this paper is focus on order picking routing planning. First, we introduced PSO, and the two order picking method in literatures (Nearest Center of Rectangular Insertion; NCRI and Minimum Traveling Loop Insertion; MTLI). Then we introduced Maximum Loop Insertion which is proposed in this paper. We try to use the planning result computed by the algorithm in this paper to PSO, and expect PSO can find the better planning solution. Take the one order for example, which is shown in Fig. 1, the order should pick of location 23, 121, 36, 66, 114, and 368. The order picking sequence is a, b, c, d, e, and f in Fig. 1.



(a) (b)

Figure 1: NCRI Order Picking Routing

3.2.1 Nearest Center of Rectangular Insertion (NCRI)

In this sub-section, we will introduce the Nearest Center of Rectangular Insertion (NCRI) which is proposed by Ho and Su [13]. It supposed the pickers walked in the middle of the aisle, the pickers can pick the two sides products in the same time. Consequently, the two sides locations can considered the same point, and supposed the I/O point is (0,0), and (x_j, y_j) is means mi products in the order of each item j's location, such as $\forall$ j =1,2,..., mi.

First, we are choosing the nearest two order picking points from I/O, such as the point a, b, in Fig. 1 (a). These two points and I/O depot of the practice traveling path is surrounded to rectangular circle. All have picked order picking points will form to the Loop Set (LS). According to the Formula (3), and compute the all picking points' location in LS, which form to Center of Rectangular (CoR).

$$\mathrm{CoR} = \left(\frac{\max\{x_n ; n \in \mathrm{LS}\} + \min\{x_n ; n \in \mathrm{LS}\}}{2}, \frac{\max\{y_n ; n \in \mathrm{LS}\} + \min\{y_n ; n \in \mathrm{LS}\}}{2} \right) \quad (3)$$

Calculation the distance between all other order picking points j to CoR, which is calculated by Formula (4). We find the nearest distance order picking point k, and insert it into the LS, shown in Formula (5). If there are more than two orders picking points are all the same nearest distance of CoR, then choosing whichever one to insert.

$$d_{\mathrm{CoR},j} = \left| x_j - x_{\mathrm{CoR}} \right| + \left| y_j - y_{\mathrm{CoR}} \right| ; \forall j \notin \mathrm{LS} \quad (4)$$

$$d_{\mathrm{CoR},k} = \min\{ d_{\mathrm{CoR},j} ; j \notin \mathrm{LS} \} \quad (5)$$

The pickers from order picking point u' to order picking u", the practical traveling distances is $TD_{u'u''}$. If the order picking point u' and u" are both in cross aisle's front area or end area, the computation formula is shown in Formula (6). The W is sub-aisle's length, if in different areas, then using the Formula (7).

$$TD_{u'u''} = \begin{cases} |x_{u'} - x_{u''}| + \min(y_{u'} + y_{u''}, |2W - y_{u'} - y_{u''}|); \text{if } x_{u'} \neq x_{u''} \\ |y_{u'} - y_{u''}| \qquad ; \text{if } x_{u'} = x_{u''} \end{cases} \tag{6}$$

$$TD_{u'u''} = \begin{cases} |x_{u'} - x_{u''}| + |y_{u'} - y_{u''}|; \text{if } x_{u'} \neq x_{u''} \\ |y_{u'} - y_{u''}| \qquad ; \text{if } x_{u'} = x_{u''} \end{cases} \tag{7}$$

Find out the edge $\overline{u'u''}$ of loop L, insert the point k, and the practical distance increasing at least ($TD_{u'k}$ + $TD_{ku''}$ − $TD_{u'u''}$), then repeat the previous step and compute. Insert the other order picking point to insert to the loop, until included all order picking point of order i, which is shown in Fig. 1 (b). Then, we will get the order picking sequence of I/O, 23, 121, 114, 66, 36, 368, and I/O. Finally, compute the order picking distance of order i. (Formula (8))

$$D_i = \sum_{u',u'' \in m_i} TD_{u'u''} \tag{8}$$

3.2.2 Minimum Traveling Loop Insertion (MTLI)

In this sub-section, we illustrate the Ho and Su [13] proposed the Minimum Traveling Loop Insertion (MTLI). First, find out the order picking point nearest the I/O point, shown in Fig. 2 (a), point a. The particle traveling path of this point and I/O depot form Traveling Loop. Then find out the other order picking points which point insert to loop L will add the shortest distance. By Formula (4) or Formula (5), find each order picking point j to insert into the edge of the loop L will add the shortest traveling distance ($TD_{u'j}$ + $TD_{ju''}$ − $TD_{u'u''}$). Hence, it can from all order picking points j, to find a point k to add the shortest distance. Then, using the same method, repeat, and find the next order picking point, until included all order picking points in order i, shown in Fig. 2 (b), and obtained the order sequence I/O, 121, 368, 114, 66, 36, 23, and I/O. Finally, compute the order picking distance of order i.

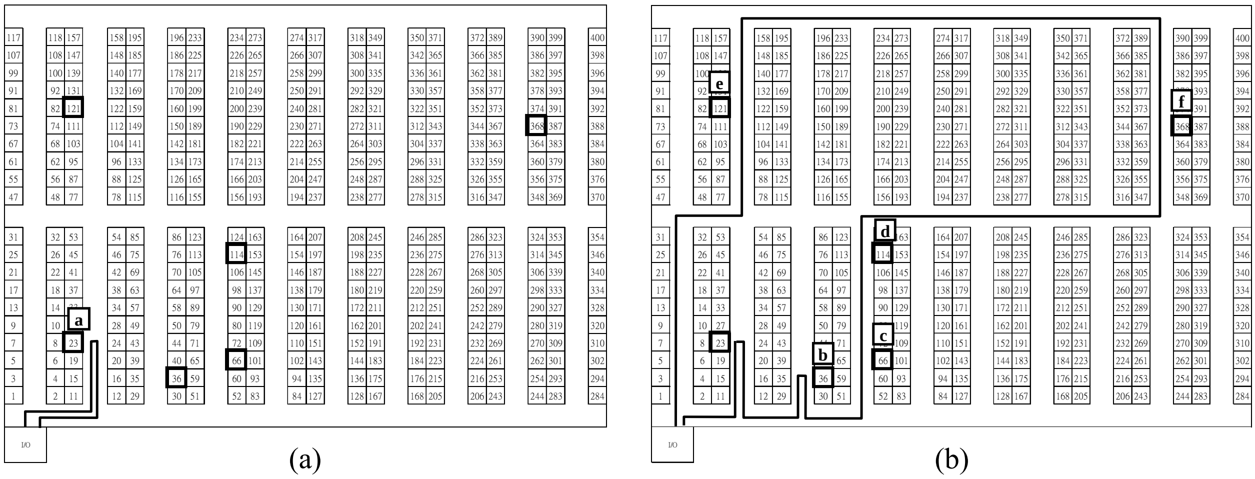


Figure 2: MTLI Order Picking Routing

3.2.3 Particle Swarm Optimization

Particle Swarm Optimization is an optimal tool of evolutionary generation, and is Swarm Intelligence algorithm. It found the each particle by self optimal memory solution, and swarm optimal solution. Then update the velocity and position until all particles find out the global optimal solution.

3.2.3.1 PSO Parameter Setting

According to result of Hsieh et al. [9], the parameters setting is as following:

Number of Particles: The Particle setting is 30. Maximum Velocity: Because the storage locations added to 400, so in this paper, we raises the velocity V_{id}^{1} is between (-80, 80), then the maximum velocity is at 160. Learning Factor: Learning factors of c_1 and c_2 usually have a value of 2. Inertia Weight: PSO with an inertia weight is set 0.8. Stop Condition: The maximum number of iterations is 200 or all particles converge in the same point.

3.2.3.2 PSO Fitting Function

The function of the PSO fitting equation is evaluate the particle obtain the optimal solution or not. Therefore, it set up different function based on different problem. In this paper, the main objective is minimizing total order picking distances.

3.2.3.3 PSO Algorithm Process

The PSO algorithm process is shown in Fig. 3

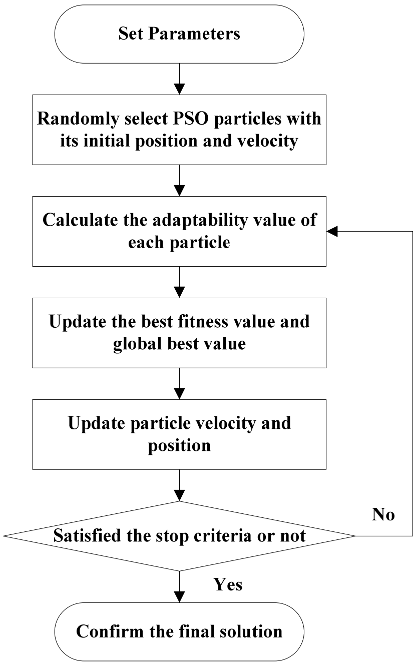


Figure 3: Flow Chart of Pso

3.2.4 Maximum Loop Insertion (MLI)

According to previous scholars, who proposed the order picking method, almost using the shortest distance to construct order picking routing. Hence, in this paper, we proposed the Maximum Loop Insertion (MLI), that makes solution has whole concept of all order picking points, and improve the solution performance of algorithms.

This algorithm first focus on the all order picking points, and find out the three order picking points to form a maximum loop. The first order picking point can search all aisles having order picking points. Then, find out the nearest aisle from I/O depot, and find the nearest order picking point from I/O in that aisle, in Fig. 4 (a), point a. The second order picking point, find the farthest order picking point from I/O point on Y axis. If there are order picking points with the same distance, then choosing the nearest order picking point from I/O depot, in Fig. 4 (a), point b. The third order picking point, finding all aisles which has order picking points, and find the farthest aisle from I/O depot. Then, find the nearest order picking point from I/O in that aisle, shown in Fig. 4 (a), point c. After got these three points, it need to check the points, if any one repeats, then delete the repeat point, then according to the remnant points and I/O point, using the shortest path to construct the maximum loop path.

Although, it called Maximum Loop path, but when finished order picking routing planning, the traveling distance of this path is the must traveling distance picking the outside boundary locations. So, it must has the whole conception of must order picking locations. Then, find the others of each order picking point j, insert into any edge of the maximum loop, that make additional distance ($TD_{u'j} + TD_{ju''} - TD_{u'u''}$) shortest. Then, using the same method to repeat, and find the next order picking point, until included all order picking points of order i, shown in Fig. 4 (b). Then, obtained the order picking sequence, I/O, 23, 121, 368, 114, 66, 36, and I/O. Finally, calculated the order picking distance of order i, it can avoid when construct the path, according to the shortest distance, then fall in local solution.

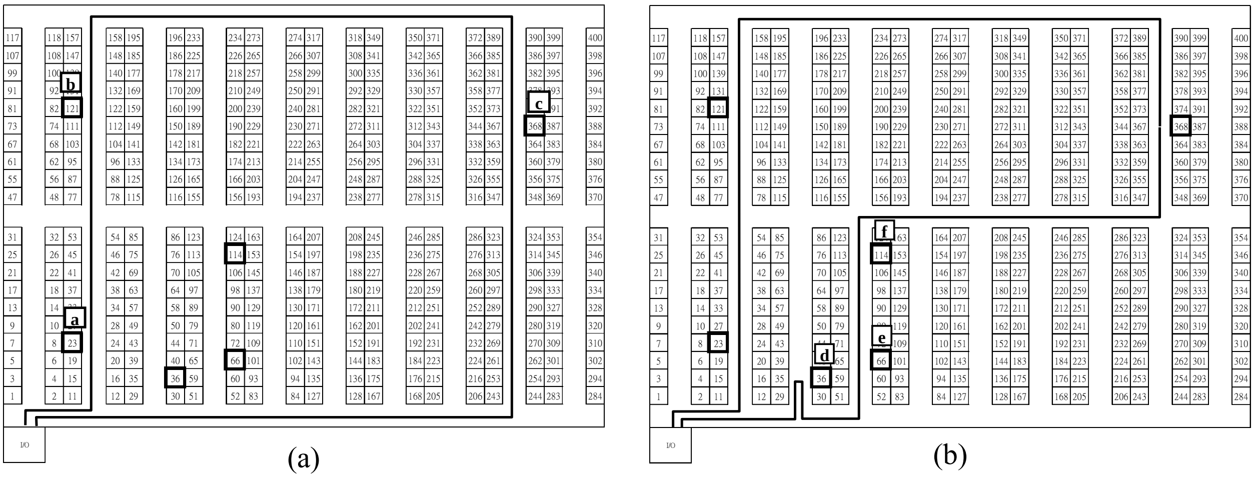


Figure 4: MLI Order Picking Routing

The MLI algorithm process is as following:

Step 1: First, find the nearest aisle which has order picking points from I/O depot. Then, find out the nearest order picking point from I/O in that aisle, shown in Fig. 4 (a), point a.

Step 2: Find the farthest depth order picking point from I/O depot. If there are the same depth of order picking points, then choosing the nearest order picking point from I/O, shown Fig. 4 (a), point b.

Step 3: Find the farthest aisle from I/O depot, and then find the nearest order picking point from I/O depot, shown in Fig. 4 (a), point c.

Step 4: Check the previous three steps, and find is it repeated or not. If there is repeat one, then delete the repeat point, then using remnant points and I/O depot, and construct the maximum loop traveling routing with the shortest traveling distance.

Step 5: According to the shortest insert distance, then from the non choosing points to find the insert order picking point k which add the shortest distance.

Step 6: Insert this order picking point k in the loop, it formed to a new loop. If there are the same shortest distance points, then choosing by random.

Step 7: Check the constructed loop included all order picking points or not. If already included all order picking points, then dropped to step 8, if not then go back to step 5.

Step 8: Compute the traveling distance of the constructed loop, then finished the order picking routing planning.

4 Simulation Analysis

According to the 3.1 sub-section warehouse layout, we use eM-plant 7.0 to construct the order picking environment system in distribution center in this paper. We repeat the simulations 30 times, and generate the 100 orders by computer randomly. It compared with NCRI, MTLI, MLI, PSO and MLI combined PSO, considered the difference between the performances of each order picking method. The MLI combine PSO, means using MLI solution to be the initial solution of PSO. The analysis by SPSS 10.0, and expect the shortest average total order picking distance (unit: m), and the shortest CPU run time (unit: s).

From the average total order picking distance, we use 95% confidence level to do the variances analysis, shown in Table 1. The P value is less than 0.05, so the different order picking method has significance difference in average order picking distance. Therefore, using Duncan Test, it will cluster each order picking method, which shown in Table 2. It clusters the four group of order picking method, it found MLI combine with PSO, and MLI all fall in the first group, has the best performance. It verified that we proposed MLI in this paper is better than the literature algorithms (MTLI, NCRI) and PSO, all that has significance difference. If using the MLI initial solution to PSO, can make PSO to find the better order picking routing, and improve the PSO solution performance.

Table 1: Variances Analysis of Average Total Order Picking Distance

Order Picking Method	Numbers	Average	Standard deviation	F Test	P Value
NCRI	30	14970.45	369.1524	135.1036	0.000
MTLI	30	14696.25	335.0943		
MLI	30	14335.65	310.091		
PSO	30	16239.18	518.9535		

Table 2: Duncan Test of Average Total Order Picking Distance

Order Picking Method	Numbers	Duncan Group			
		1	2	3	4
MLI Combine with PSO	30	14275.62			
MLI	30	14335.65			
MTLI	30		14696.25		
NCRI	30			14970.45	
PSO	30				16239.18
Significance		0.536	1.000	1.000	1.000

From the average total CPU run time, we use 95% confidence level to do variance analysis, shown in Table

3. We can see from P value less than 0.05, different order picking method has significance difference to the average total CPU run time. Therefore, using Duncan Test, it cluster all order picking methods, shown in Table 4, it clustered the order picking method by three groups, that NCRI, MTLI and MLI all has the best performance, and fall in the same group. It all in 1 second finished 100 orders of order picking routing. We can see form Table 4, it also can found that PSO is the worst performance of average total CPU run time, but if using MLI initial solution to PSO, it can reducing CPU run time efficiency, it can improve the solution efficiency.

Table 3: Variance Analysis of Average Total CPU Run Time

Order Picking Method	Numbers	Average	Standard deviation	F Test	P Value
NCRI	30	0.619	0.038	3544.853	0.000
MTLI	30	1.561	0.126		
MLI	30	1.954	0.170		
PSO	30	188.215	15.820		
MLI Combine with PSO	30	150.591	10.750		

Table 4: Duncan Test of Average Total CPU Run Time

Order Picking Method	Numbers	Duncan Group		
		1	2	3
NCRI	30	0.619		
MTLI	30	1.561		
MLI	30	1.954		
MLI Combine with PSO	30		150.591	
PSO	30			188.215
Significance		.573	1.000	1.000

5 Conclusions

The main purpose of this paper is discussing the improvement of order picking operations in distribution center. It expects according to order picking method to improve order picking routing planning, and improves the order picking performance in distribution center. Therefore, it discusses all kinds of order picking methods influence the order picking performance. And this research verified the performance of MLI. It is significance reducing order picking distance, and using the solution to be an initial solution of PSO, and find the better order picking routing. The contributions in this paper are generation as following:

The MLI which is proposed in this paper exactly has the significance improve of order picking performance. The average total order picking distance is significance better than the better performance algorithms in literatures (NCRI and MTLI). In the meantime, the algorithm MLI has the better performance in the average total CPU run time, and the same as NCRI, and MTLI.

In this study, we try to combine MLI and PSO by putting the solution of MLI as the initial solution of PSO, and verified if it is a better solution to do the initial solution of PSO, then can find the better solution by PSO. It also can improve the PSO solution efficiency, avoid the blindly searching solution, and has the better solution of average total order picking distance, reducing solution time, and CPU run time. It makes the PSO more suitable in practice.

In a practical sense, this paper adapted 80/20 method to deal with products locations, using classification storage, and adding cross aisles between storage spaces. The method which proposed in this paper, also verified can improve the overall efficiency of the distribution center and it can be a good reference for distribution industries to consult.

ACKNOWLEDGEMENT

The author would like to thank The National Science Council, Republic of China, for financially supporting this research under the grant number of **NSC-96-2221-E-216-002-**.

References

[1] C. G. Petersen II, R. W. Schmenner, “An Evaluation of Routing and Volume-based Storage Policies in an Order Picking Operation,” *Decision Sciences*, Vol. 30, No. 2, pp. 481-501, 1999.

[2] H. D. Ratliff, A. S. Rosenthal, “Order-picking in a rectangular warehouse: a solvable case of the traveling salesman problem,” *Operations Research*, Vol. 31, pp. 507-521, 1983.

[3] J. A. Tompkins, J. A. White, Y. A. Bozer, E. H. Frazelle, J. M. A. Tanchoco, J. Trevino, “Facilities Planning,” Wiley New York, 1996.

[4] J. Holland, “Adaptation in Natural and Artificial Systems,” University of Michigan Press Ann Arbor, 1975.

[5] J. J. Coyle, E. J. Bardi, C. J. Langley, “The Management of Business Logistics,” (St Paul, MN: West), 1996.

[6] J. Kennedy, R. C. Eberhart, “Particle swarm optimization,” *Proceedings IEEE International Conference on Neural Networks*, Vol. 4, 1942-1948, 1995.

[7] K. J. Roodbergen, R. D. Koster, “Routing method for warehouse with multiple aisles,” *International Journal of Production Research*, Vol. 39, No. 9, pp. 1865-1883, 2001.

[8] L. F. Hsieh, C. J. Huang, C. L. Huang, “Applying Particle Swarm Optimization to Plan Picking Routing

and Order Batching in Distribution Center," *Proceedings of International Conference on Pacific Rim Management 16th Annual Meeting*, Hawaiian pp. 520-525, 2006.

[9] L. F. Hsieh, C. J. Huang, C. L. Huang, "Applying Particle Swarm Optimization to Plan Picking Routing in Warehouse Systems," *Proceedings of International Conference on Intelligent System Application to Power Systems*, Taiwan, pp. 117-121, 2006.

[10] M. Dorigo, "Optimization, Learning and Natural Algorithms," Ph.D. Thesis. Politecnico di Milano, Italy, EU. (In Italian.), 1992.

[11] R. C. Eberhart, Y. Shi, "Particle swarm optimization: developments, applications and resources," *Proceedings Congress on Evolutionary Computation*, Vol. 1, pp. 27-30, 2001.

[12] R. W. Hall, "Distance approximations for routing manual pickers in a warehouse," *IIE Transactions*, Vol. 25, No. 4, pp. 76-87, 1993.

[13] Y. C. Ho, T. S. Su, "A Study on Optimizing Order-Picking Operations in Distribution Centers," *The Fourth Asia-Pacific Conference on Industrial Engineering and Management Systems*, Taipei Taiwan, pp. 18-20, 2002.

出席 ICOTA 7 國際會議心得報告及發表之論文

The 7th International Conference on Optimization: Techniques and Applications (ICOTA 7)

Kobe International Conference Center, Kobe, Japan

December 12-15, 2007

謝玲芬

中華大學科技管理學系

一、會議經過

The 7th International Conference on Optimization: Techniques and Applications (ICOTA 7)是探討最佳化理論領域一個相當正式的研討會，每三年舉辦一次，其主要目的是提供所有在產、關、學及軟體開發者一個國際聚會的機會，大家齊聚一堂彼此交換構想、分享經驗及心得，並藉由溝通及討論，研擬未來之發展方向。2007 年是第七屆，由十二月十二日至十五日共四天，地點選在日本的神戶。每屆發表論文的人數都相當多，今年共有來自32個國家，254篇論文發表，大會並特別安排六場精彩的專題演講，參加本次研討討會真是受益良多。大會共分為六十餘個場次發表論文，每個場次約四至五篇論文，讓與會者能充份挑選自己有興趣的主題之場次參與討論。大會主辦單位在閉幕餐會上除了供應豐盛的餐點外，並安排傳統的日本歌舞表演，整個研討會到此在一片歡樂中圓滿閉幕。

由於十二月中旬參加研討會，恰逢聖誕前之前，神戶街頭相當熱鬧，趁著研討會的空檔，親身搭乘日本的電車，逛逛當地的商場，順道體驗當地的民俗風情。且神戶當地為紀念之前的阪神大地震，為祈求神戶居民之平安，當地每年十二月均舉辦燈祭，以祈求人民的平安及世界和平，場面相當壯觀！當地居民告訴我門今年是舉辦燈祭的最後一年！大家更是珍惜此一機會，留下深刻的記憶。另外我也品嚐的當地著名且道地的大阪燒，參觀當地的古蹟－姬路城，真是收穫良多。

神戶燈祭(1)

神戶燈祭(2)

神戶鶴橋風月的大阪燒

神戶古蹟－姬路城

神戶港　　　　日本歌舞表演

論文發表會場

二、研討會發表之論文

Optimal Order Picking Planning for Distribution Center with Cross Aisle

<u>Ling-Feng Hsieh</u>

Professor, Department of Technology Management, Chung Hua University, Hsinchu 300, Taiwan, R.O.C.

lfhsieh@chu.edu.tw

Chien-Lin Huang

Graduate student, Department of Technology Management, Chung Hua University, Hsinchu 300, Taiwan, R.O.C.

m09403033@chu.edu.tw

Abstract

Order picking method is one of the most important operations in the distribution center. The route planning of order picking systems will allow for the possibilities of increasing in production efficiency, reducing the operation cost in distribution center, and improving the corporation competitiveness. In a distribution center with cross aisle, although the cross aisle layout may reduce the order picking distance, it also may raise the complexity for picking routing planning. Focus on this problem, a heuristic algorithm (called Maximum Loop Insertion) is proposed in this paper, as well as compare with other famous algorithms and Particle Swarm Optimization (PSO), in order to improve the order picking performance. According to the simulation experiment, it verified that the Maximum Loop Insertion algorithm actually achieves the better performance. Overall, the result of this research will enhance the best route planning of order picking systems in distribution center and provide the industry as a reference in the warehouse design in the future.

Keywords: Order picking system, Cross aisle, Maximum Loop Insertion, Particle Swarm Optimization, Performance

1 Introduction

As economy develops and changes in consuming habit, it makes the types and structures of marketing channel are transferred to the supplier transported the products to retailer by distribution center. Therefore, it is an important issue about how to improve the distribution center operation efficiency. Due to the fact that consumer’s request has been changed from few items and small volume to many items and small volume, it is hoped that if the order picking operations can be finished in reasonable time or not will influence the operation cost and service level of distribution center. In internal operations of distribution center, the factors included warehouse layout, storage assignment strategies, order picking operations, which will influence order picking system. The order picking operations is an important complex work. Coyle et al. [5] and Tompkins et al. [3] proposed that the order picking operations of the system takes more than 65% of the whole warehouse operation cost. In addition, the traveling time takes about 50% of the order picking activities. Therefore, it expected reducing the traveling distance by planning the order picking routing, in order to improve the whole operation efficiency in distribution center.

In order to improve order picking routing planning, we proposed a new heuristic algorithm in this paper. We also expect it can improve the picking performance. In the same time, we apply Particle Swarm Optimization (PSO) in order picking routing, and compared with the best order picking method in literature review. In this paper, it used eM-plant to construct the warehouse system, and verify the order picking method which is proposed in this paper can has the better performance. The performance index to evaluate each order picking method is average total order picking distance and average total CPU run time. Finally, it is expected this study can consult to the industries.

2 Literature Review

2.1 Route Planning of Order Picking System

According to the well order picking routing planning, it can make the minimize order picking distance, reducing order picking time, and improve order picking performance. Hall [12] supposed that did not consider the width of the aisle, as well as evaluate and compare with different order picking strategies. The order picking strategies included Traversal, Midpoint Return, Largest Gap Return, and the best performance is using Largest Gap Return. Petersen and Schmenner [1] had two major policy decisions that determine the efficiency of order picking operations, which are storage policies and routing policies. It also considered Transversal Strategy, Return Strategy, Midpoint Strategy, Largest Gap Strategy, and Composite Strategy; five different kinds of order picking routing strategies, as well as compare with the optimal solution, the Composite Strategy have the best performance. Ho and Su [13] proposed two kind of heuristic algorithms, which are Nearest Center of Rectangular Insertion (NCRI) and Minimum Traveling Loop Insertion (MTLI). He also compared with previous scholar's method, for example, Largest Gap Strategy, Nearest Center of Geometry Insertion Heuristic, the results shown that the two heuristic algorithms which is proposed by author has the better performance.

Ratliff and Rosenthal [2] discuss the influence of add the cross aisle of order picking routing, the results shown that the order picking routing planning will become more complex in warehouse which has cross aisles. Roodbergen and Koster [7] focused on estimate variety kinds of order picking strategy to compare different number of aisles, different number of items and different width of aisles. The algorithms to evaluate the shortest path included S-shape Heuristic, Largest Gap Heuristic, Aisle-by-aisle Heuristic, Optimal Algorithm, Combined Heuristic, and Combine$^+$ Heuristic. The best performance is Combine$^+$ strategy. Hsieh et al. [8] applying PSO in order picking routing and storage assignment, and compare with previous scholar, it verified that applying PSO has the better performance.

2.2 Particle Swarm Optimization

Kennedy, J. and R. C. Eberhart [6] proposed Particle Swarm Optimization (PSO) in 1995. It is similar to John Holland [10] proposed Genetic Algorithms (GA) in 1960. They are all belonging to Evolutionary computation and within the evolutionary generation to optimal solutions. The main concept of GA is the survival of the fittest which is proposed by Charles Darwin. That is using three basic operations, which is Reproduction, Crossover, and Mutation to imitate natural evolutionary process, and according to the evolutionary generation to optimal solutions. Therefore, PSO has no crossover and mutation, it is easier than GA, but is has better global optimal solution ability.

PSO and Dorigo [10] proposed Ant Colony Optimization (ACO) in 1992 is all Swarm Intelligence algorithms that are according to swarm intelligence to solve problems. ACO is a cooperative heuristic searching algorithm inspired by the methodological study on the behavior of ants. The ants can find out the food is done by an indirect communication known as pheromone, left by the ants on the paths, and constructive to the shortest distance between the nest and food.

PSO is according to three factors to find out the optimal solution, that is (1) the current moving direction by itself, (2) the previous experiment by itself, (3) the swarm experiments, and compare with Fitness Value which is computed by Fitness Function to revise the velocity and position of itself.

The definitions of PSO related variances, X_{id}^{l} is the particle i, d dimension, l stage position. X_{id}^{l+1} is the particle i, d dimension, in $l+1$ stage position. P_{id} represents the optimum position recorded by the i^th^ particle in d dimension. P_{gd} is the optimum position resolved by a population of particles in d dimension. V_{id}^{l} is the velocity of the i^{th} particle, d dimension in l stage. V_{id}^{l+1} is the velocity of the ith particle, d dimension in $l+1$ stage. $rand()$ is random number between [0, 1]. c_1 and c_2 are learning factors which controls the acceleration of particle velocity. w is the inertial constant that allows user to control the parameters. A small w value will direct searches within current space, and a large w value will indicate searches in new space. Appropriate selection of w value, c_1 and c_2 learning factors can expand search space to achieve a balanced result. The velocity and position update formula is shown in Formula (1) and Formula (2).

$$V_{id}^{l+1} = w \cdot V_{id}^{l} + c_1 \cdot rand() \cdot (P_{id} - X_{id}^{l}) + c_2 \times rand() \cdot (P_{gd} - X_{id}^{l}) \qquad (1)$$

$$X_{id}^{l+1} = X_{id}^{l} + V_{id}^{l+1} \qquad (2)$$

The original two scholars which is proposed PSO is not using inertia weight w. The inertia weight w is proposed by Shiand and Eberhart [11] in 1998, illustrated inertia weight w using can make the solution process to find out the global best solution faster. The characteristic of inertia weight w is similar to cooling parameter of Simulated Annealing (SA) that can make the solution become convergence. Shi and Eberhart also illustrates w between 0.8 and 1.2, it has more chance to find out the global solution.

3 Model Construction

3.1 Model Constructing

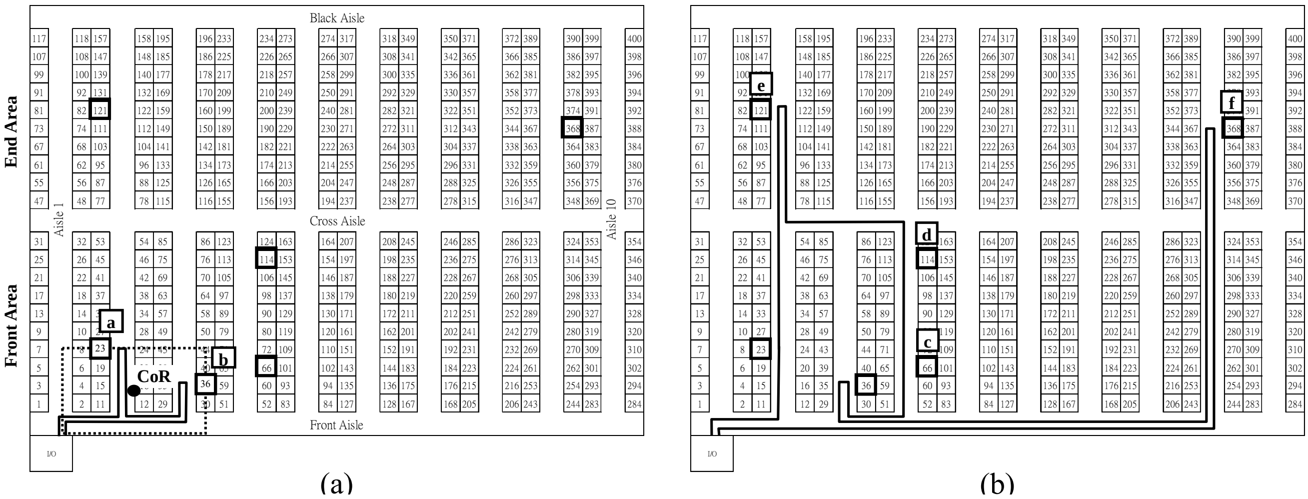
This research probes into the warehouse environment layout is shown in Fig. 1. There are 10 aisles, in each aisle of left hand side and right hand side all have 20 storage locations, so total locations are 400. Suppose the depth of storage is 1m, the width is 1m; the main aisle width is 2.5m, sub-aisle length is 10m, and width is 2.5m. There has front aisle, end aisle, and 1 cross aisle, the width all are 2.5m. The input depot and output depot is the same point in the front of left side. The experiment assumes that order pickers begin the tour at the input depot and end at the output depot. Thus, upon completing the retrieval of the order, the order picker immediately begins retrieval of the next order released. In order to show actual situation, the rectilinear distance is considered for calculation, and give the each locations a number according to the distance between the locations to I/O depot.

Because the well storage strategy can reduce the moving distance between in warehouse and out warehouse, reducing operation time, and full using storage space. Therefore, it adapted classification storage, put the high frequency produce near the I/O depot, and put the low frequency product far away I/O depot in this paper. It can reducing the order picking distance, and improves the order picking efficiency. The access frequency is adapted 80/20 method, the meaning is 20% products is own order picking activity 80%. For this reason, we put these 20% products near the I/O depot.

The part of order batching is adapted the single order method. This method is the general method in industry. The advantage of this method is can reduce the complex of order picking, this is different order batching which has more consuming time to separate the combine order products.

3.2 Route Planning of Order Picking System

The main discussion of this paper is focus on order picking routing planning. First, we introduced PSO, and the two order picking method in literatures (Nearest Center of Rectangular Insertion; NCRI and Minimum Traveling Loop Insertion; MTLI). Then we introduced Maximum Loop Insertion which is proposed in this paper. We try to use the planning result computed by the algorithm in this paper to PSO, and expect PSO can find the better planning solution. Take the one order for example, which is shown in Fig. 1, the order should pick of location 23, 121, 36, 66, 114, and 368. The order picking sequence is a, b, c, d, e, and f in Fig. 1.



(a) (b)

Figure 1: NCRI Order Picking Routing

3.2.1 Nearest Center of Rectangular Insertion (NCRI)

In this sub-section, we will introduce the Nearest Center of Rectangular Insertion (NCRI) which is proposed by Ho and Su [13]. It supposed the pickers walked in the middle of the aisle, the pickers can pick the two sides products in the same time. Consequently, the two sides locations can considered the same point, and supposed the I/O point is (0,0), and (x_j, y_j) is means mi products in the order of each item j's location, such as $\forall j = 1,2,..., mi$.

First, we are choosing the nearest two order picking points from I/O, such as the point a, b, in Fig. 1 (a). These two points and I/O depot of the practice traveling path is surrounded to rectangular circle. All have picked order picking points will form to the Loop Set (LS). According to the Formula (3), and compute the all picking points' location in LS, which form to Center of Rectangular (CoR).

$$CoR = \left(\frac{\max\{x_n; n \in LS\} + \min\{x_n; n \in LS\}}{2}, \frac{\max\{y_n; n \in LS\} + \min\{y_n; n \in LS\}}{2} \right) \tag{3}$$

Calculation the distance between all other order picking points j to CoR, which is calculated by Formula (4). We find the nearest distance order picking point k, and insert it into the LS, shown in Formula (5). If there are more than two orders picking points are all the same nearest distance of CoR, then choosing whichever one to insert.

$$d_{CoR,j} = |x_j - x_{CoR}| + |y_j - y_{CoR}|; \forall j \notin LS \tag{4}$$

$$d_{CoR,k} = \min\{d_{CoR,j}; j \notin LS\} \tag{5}$$

The pickers from order picking point u' to order picking u'', the practical traveling distances is $TD_{u'u''}$. If the order picking point u' and u" are both in cross aisle's front area or end area, the computation formula is shown in Formula (6). The W is sub-aisle's length, if in different areas, then using the Formula (7).

$$TD_{u'u''} = \begin{cases} |x_{u'} - x_{u''}| + \min\left(y_{u'} + y_{u''}, |2W - y_{u'} - y_{u''}|\right); \text{if } x_{u'} \neq x_{u''} \\ |y_{u'} - y_{u''}| \qquad ; \text{if } x_{u'} = x_{u''} \end{cases} \qquad (6)$$

$$TD_{u'u''} = \begin{cases} |x_{u'} - x_{u''}| + |y_{u'} - y_{u''}|; \text{if } x_{u'} \neq x_{u''} \\ |y_{u'} - y_{u''}| \qquad ; \text{if } x_{u'} = x_{u''} \end{cases} \qquad (7)$$

Find out the edge $\overline{u'u''}$ of loop L, insert the point k, and the practical distance increasing at least ($TD_{u'k}$ + $TD_{ku''}$ − $TD_{u'u''}$), then repeat the previous step and compute. Insert the other order picking point to insert to the loop, until included all order picking point of order i, which is shown in Fig. 1 (b). Then, we will get the order picking sequence of I/O, 23, 121, 114, 66, 36, 368, and I/O. Finally, compute the order picking distance of order i. (Formula (8))

$$D_i = \sum_{u',u'' \in m_i} TD_{u'u''} \qquad (8)$$

3.2.2 Minimum Traveling Loop Insertion (MTLI)

In this sub-section, we illustrate the Ho and Su [13] proposed the Minimum Traveling Loop Insertion (MTLI). First, find out the order picking point nearest the I/O point, shown in Fig. 2 (a), point a. The particle traveling path of this point and I/O depot form Traveling Loop. Then find out the other order picking points which point insert to loop L will add the shortest distance. By Formula (4) or Formula (5), find each order picking point j to insert into the edge of the loop L will add the shortest traveling distance ($TD_{u'j}$ + $TD_{ju''}$ − $TD_{u'u''}$). Hence, it can from all order picking points j, to find a point k to add the shortest distance. Then, using the same method, repeat, and find the next order picking point, until included all order picking points in order i, shown in Fig. 2 (b), and obtained the order sequence I/O, 121, 368, 114, 66, 36, 23, and I/O. Finally, compute the order picking distance of order i.

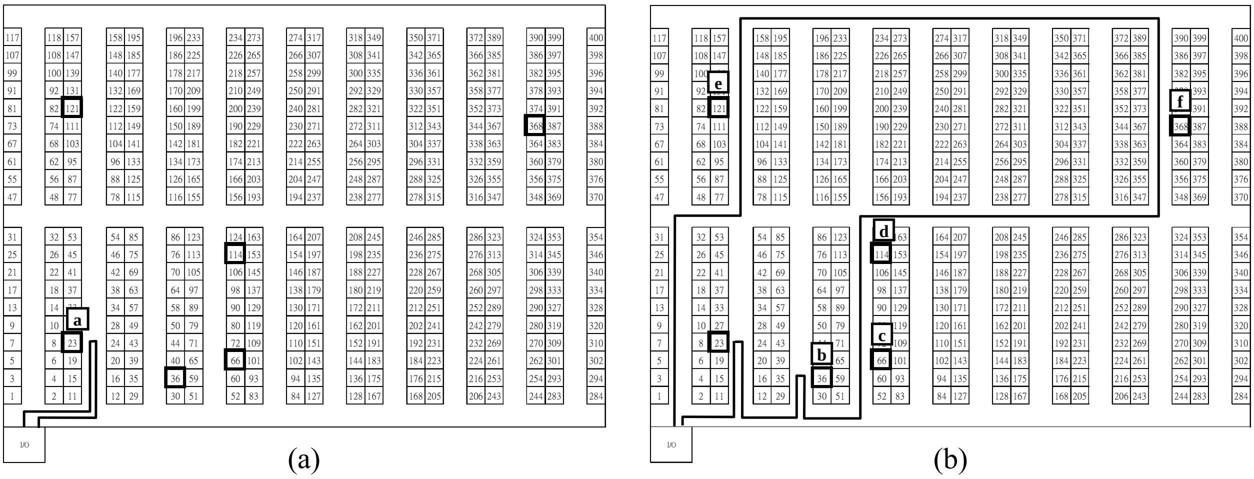


Figure 2: MTLI Order Picking Routing

3.2.3 Particle Swarm Optimization

Particle Swarm Optimization is an optimal tool of evolutionary generation, and is Swarm Intelligence algorithm. It found the each particle by self optimal memory solution, and swarm optimal solution. Then update the velocity and position until all particles find out the global optimal solution.

3.2.3.1 PSO Parameter Setting

According to result of Hsieh et al. [9], the parameters setting is as following:

Number of Particles: The Particle setting is 30. Maximum Velocity: Because the storage locations added to 400, so in this paper, we raises the velocity V_{id}^{1} is between (-80, 80), then the maximum velocity is at 160. Learning Factor: Learning factors of c_1 and c_2 usually have a value of 2. Inertia Weight: PSO with an inertia weight is set 0.8. Stop Condition: The maximum number of iterations is 200 or all particles converge in the same point.

3.2.3.2 PSO Fitting Function

The function of the PSO fitting equation is evaluate the particle obtain the optimal solution or not. Therefore, it set up different function based on different problem. In this paper, the main objective is minimizing total order picking distances.

3.2.3.3 PSO Algorithm Process

The PSO algorithm process is shown in Fig. 3

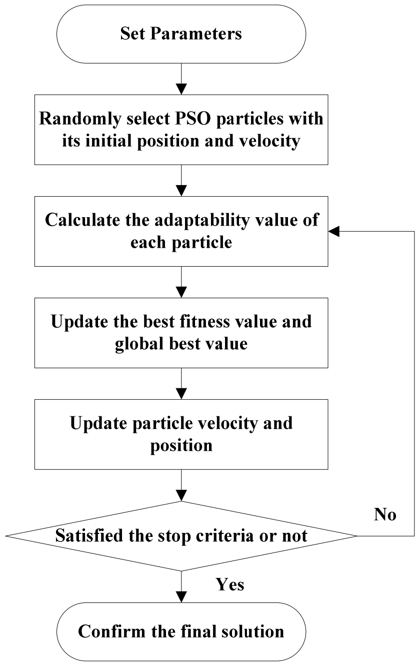


Figure 3: Flow Chart of Pso

3.2.4 Maximum Loop Insertion (MLI)

According to previous scholars, who proposed the order picking method, almost using the shortest distance to construct order picking routing. Hence, in this paper, we proposed the Maximum Loop Insertion (MLI), that makes solution has whole concept of all order picking points, and improve the solution performance of algorithms.

This algorithm first focus on the all order picking points, and find out the three order picking points to form a maximum loop. The first order picking point can search all aisles having order picking points. Then, find out the nearest aisle from I/O depot, and find the nearest order picking point from I/O in that aisle, in Fig. 4 (a), point a. The second order picking point, find the farthest order picking point from I/O point on Y axis. If there are order picking points with the same distance, then choosing the nearest order picking point from I/O depot, in Fig. 4 (a), point b. The third order picking point, finding all aisles which has order picking points, and find the farthest aisle from I/O depot. Then, find the nearest order picking point from I/O in that aisle, shown in Fig. 4 (a), point c. After got these three points, it need to check the points, if any one repeats, then delete the repeat point, then according to the remnant points and I/O point, using the shortest path to construct the maximum loop path.

Although, it called Maximum Loop path, but when finished order picking routing planning, the traveling distance of this path is the must traveling distance picking the outside boundary locations. So, it must has the whole conception of must order picking locations. Then, find the others of each order picking point j, insert into any edge of the maximum loop, that make additional distance ($TD_{u'j} + TD_{ju''} - TD_{u'u''}$) shortest. Then, using the same method to repeat, and find the next order picking point, until included all order picking points of order i, shown in Fig. 4 (b). Then, obtained the order picking sequence, I/O, 23, 121, 368, 114, 66, 36, and I/O. Finally, calculated the order picking distance of order i, it can avoid when construct the path, according to the shortest distance, then fall in local solution.

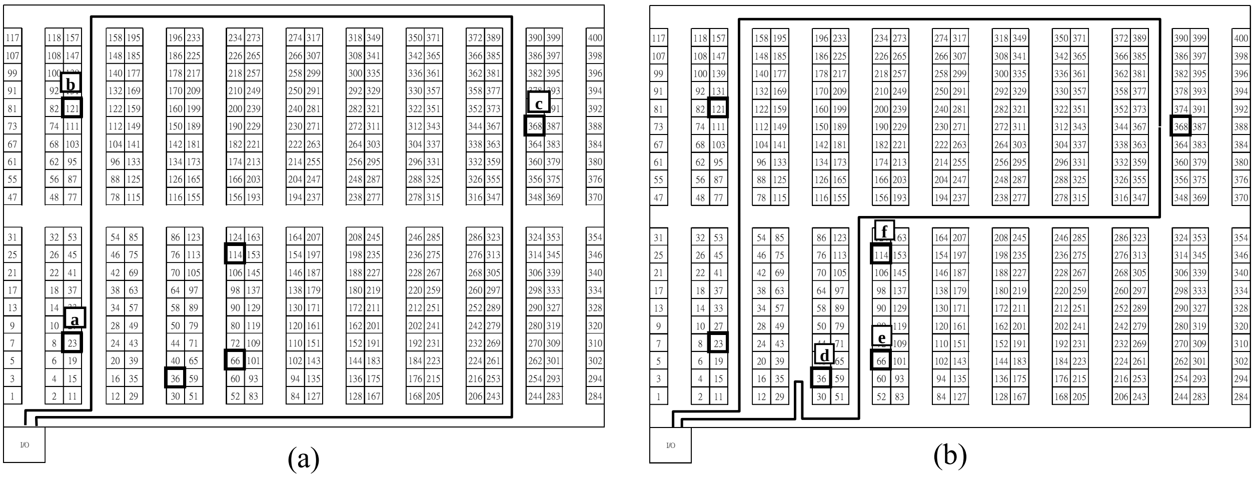


Figure 4: MLI Order Picking Routing

The MLI algorithm process is as following:

Step 1: First, find the nearest aisle which has order picking points from I/O depot. Then, find out the nearest order picking point from I/O in that aisle, shown in Fig. 4 (a), point a.

Step 2: Find the farthest depth order picking point from I/O depot. If there are the same depth of order picking points, then choosing the nearest order picking point from I/O, shown Fig. 4 (a), point b.

Step 3: Find the farthest aisle from I/O depot, and then find the nearest order picking point from I/O depot, shown in Fig. 4 (a), point c.

Step 4: Check the previous three steps, and find is it repeated or not. If there is repeat one, then delete the repeat point, then using remnant points and I/O depot, and construct the maximum loop traveling routing with the shortest traveling distance.

Step 5: According to the shortest insert distance, then from the non choosing points to find the insert order picking point k which add the shortest distance.

Step 6: Insert this order picking point k in the loop, it formed to a new loop. If there are the same shortest distance points, then choosing by random.

Step 7: Check the constructed loop included all order picking points or not. If already included all order picking points, then dropped to step 8, if not then go back to step 5.

Step 8: Compute the traveling distance of the constructed loop, then finished the order picking routing planning.

4 Simulation Analysis

According to the 3.1 sub-section warehouse layout, we use eM-plant 7.0 to construct the order picking environment system in distribution center in this paper. We repeat the simulations 30 times, and generate the 100 orders by computer randomly. It compared with NCRI, MTLI, MLI, PSO and MLI combined PSO, considered the difference between the performances of each order picking method. The MLI combine PSO, means using MLI solution to be the initial solution of PSO. The analysis by SPSS 10.0, and expect the shortest average total order picking distance (unit: m), and the shortest CPU run time (unit: s).

From the average total order picking distance, we use 95% confidence level to do the variances analysis, shown in Table 1. The P value is less than 0.05, so the different order picking method has significance difference in average order picking distance. Therefore, using Duncan Test, it will cluster each order picking method, which shown in Table 2. It clusters the four group of order picking method, it found MLI combine with PSO, and MLI all fall in the first group, has the best performance. It verified that we proposed MLI in this paper is better than the literature algorithms (MTLI, NCRI) and PSO, all that has significance difference. If using the MLI initial solution to PSO, can make PSO to find the better order picking routing, and improve the PSO solution performance.

Table 1: Variances Analysis of Average Total Order Picking Distance

Order Picking Method	Numbers	Average	Standard deviation	F Test	P Value
NCRI	30	14970.45	369.1524	135.1036	0.000
MTLI	30	14696.25	335.0943		
MLI	30	14335.65	310.091		
PSO	30	16239.18	518.9535		

Table 2: Duncan Test of Average Total Order Picking Distance

Order Picking Method	Numbers	Duncan Group			
		1	2	3	4
MLI Combine with PSO	30	14275.62			
MLI	30	14335.65			
MTLI	30		14696.25		
NCRI	30			14970.45	
PSO	30				16239.18
Significance		0.536	1.000	1.000	1.000

From the average total CPU run time, we use 95% confidence level to do variance analysis, shown in Table

3. We can see from P value less than 0.05, different order picking method has significance difference to the average total CPU run time. Therefore, using Duncan Test, it cluster all order picking methods, shown in Table 4, it clustered the order picking method by three groups, that NCRI, MTLI and MLI all has the best performance, and fall in the same group. It all in 1 second finished 100 orders of order picking routing. We can see form Table 4, it also can found that PSO is the worst performance of average total CPU run time, but if using MLI initial solution to PSO, it can reducing CPU run time efficiency, it can improve the solution efficiency.

Table 3: Variance Analysis of Average Total CPU Run Time

Order Picking Method	Numbers	Average	Standard deviation	F Test	P Value
NCRI	30	0.619	0.038	3544.853	0.000
MTLI	30	1.561	0.126		
MLI	30	1.954	0.170		
PSO	30	188.215	15.820		
MLI Combine with PSO	30	150.591	10.750		

Table 4: Duncan Test of Average Total CPU Run Time

Order Picking Method	Numbers	Duncan Group		
		1	2	3
NCRI	30	0.619		
MTLI	30	1.561		
MLI	30	1.954		
MLI Combine with PSO	30		150.591	
PSO	30			188.215
Significance		.573	1.000	1.000

5 Conclusions

The main purpose of this paper is discussing the improvement of order picking operations in distribution center. It expects according to order picking method to improve order picking routing planning, and improves the order picking performance in distribution center. Therefore, it discusses all kinds of order picking methods influence the order picking performance. And this research verified the performance of MLI. It is significance reducing order picking distance, and using the solution to be an initial solution of PSO, and find the better order picking routing. The contributions in this paper are generation as following:

The MLI which is proposed in this paper exactly has the significance improve of order picking performance. The average total order picking distance is significance better than the better performance algorithms in literatures (NCRI and MTLI). In the meantime, the algorithm MLI has the better performance in the average total CPU run time, and the same as NCRI, and MTLI.

In this study, we try to combine MLI and PSO by putting the solution of MLI as the initial solution of PSO, and verified if it is a better solution to do the initial solution of PSO, then can find the better solution by PSO. It also can improve the PSO solution efficiency, avoid the blindly searching solution, and has the better solution of average total order picking distance, reducing solution time, and CPU run time. It makes the PSO more suitable in practice.

In a practical sense, this paper adapted 80/20 method to deal with products locations, using classification storage, and adding cross aisles between storage spaces. The method which proposed in this paper, also verified can improve the overall efficiency of the distribution center and it can be a good reference for distribution industries to consult.

ACKNOWLEDGEMENT

The author would like to thank The National Science Council, Republic of China, for financially supporting this research under the grant number of **NSC-96-2221-E-216-002-**.

References

[1] C. G. Petersen II, R. W. Schmenner, "An Evaluation of Routing and Volume-based Storage Policies in an Order Picking Operation," *Decision Sciences*, Vol. 30, No. 2, pp. 481-501, 1999.

[2] H. D. Ratliff, A. S. Rosenthal, "Order-picking in a rectangular warehouse: a solvable case of the traveling salesman problem," *Operations Research*, Vol. 31, pp. 507-521, 1983.

[3] J. A. Tompkins, J. A. White, Y. A. Bozer, E. H. Frazelle, J. M. A. Tanchoco, J. Trevino, "Facilities Planning," Wiley New York, 1996.

[4] J. Holland, "Adaptation in Natural and Artificial Systems," University of Michigan Press Ann Arbor, 1975.

[5] J. J. Coyle, E. J. Bardi, C. J. Langley, "The Management of Business Logistics," (St Paul, MN: West), 1996.

[6] J. Kennedy, R. C. Eberhart, "Particle swarm optimization," *Proceedings IEEE International Conference on Neural Networks*, Vol. 4, 1942-1948, 1995.

[7] K. J. Roodbergen, R. D. Koster, "Routing method for warehouse with multiple aisles," *International Journal of Production Research*, Vol. 39, No. 9, pp. 1865-1883, 2001.

[8] L. F. Hsieh, C. J. Huang, C. L. Huang, "Applying Particle Swarm Optimization to Plan Picking Routing

and Order Batching in Distribution Center," *Proceedings of International Conference on Pacific Rim Management 16th Annual Meeting*, Hawaiian pp. 520-525, 2006.

[9] L. F. Hsieh, C. J. Huang, C. L. Huang, "Applying Particle Swarm Optimization to Plan Picking Routing in Warehouse Systems," *Proceedings of International Conference on Intelligent System Application to Power Systems*, Taiwan, pp. 117-121, 2006.

[10] M. Dorigo, "Optimization, Learning and Natural Algorithms," Ph.D. Thesis. Politecnico di Milano, Italy, EU. (In Italian.), 1992.

[11] R. C. Eberhart, Y. Shi, "Particle swarm optimization: developments, applications and resources," *Proceedings Congress on Evolutionary Computation*, Vol. 1, pp. 27-30, 2001.

[12] R. W. Hall, "Distance approximations for routing manual pickers in a warehouse," *IIE Transactions*, Vol. 25, No. 4, pp. 76-87, 1993.

[13] Y. C. Ho, T. S. Su, "A Study on Optimizing Order-Picking Operations in Distribution Centers," *The Fourth Asia-Pacific Conference on Industrial Engineering and Management Systems*, Taipei Taiwan, pp. 18-20, 2002.