

多目標銷售損失存貨模式之決策問題 與解法探討

許晉雄* 鄒慶士**

摘要

存貨管理對於企業來說是極為重要的工作，其目的是如何運用最少的成本維持高度的服務水準，降低缺貨的可能性以滿足顧客對產品的需求。如何在這些衝突目標間做出權衡取捨，便是多目標存貨控制所面臨的一大挑戰。本研究將Agrell (1995)提出的缺貨後補下三目標 (s, Q) 存貨控制模式延伸至銷售損失的情況下，運用加入區域搜尋與群集機制的混合式多目標微粒群最佳化來求解不同模式的存貨控制問題，並將結果與強健柏拉圖進化式演算法比較，發現混合式多目標微粒群最佳化的非凌越解在三項績效衡量指標上明顯的勝過強健柏拉圖進化式演算法。此外，為了避免多目標存貨控制模式出現多餘 (redundant) 的目標，本研究將三個目標之存貨控制模式轉換為兩個雙目標之存貨控制模式，分別命名為缺貨次數與缺貨數量存貨模式，並進行求解與比較不同模型之間的差異。

關鍵字：存貨管理、多目標最佳化、微粒群演算法、銷售損失

* 東吳大學財務工程與精算數學系副教授，臺北市 10048 貴陽街一段 56 號，
電話：(02)23111531-3626 傳真：(02)23812510，E-mail: hsiung@scu.edu.tw

** 國立台北商業技術學院企業管理系教授，臺北市 10051 濟南路一段 321 號，
電話：(02)23226459，傳真：(02)27942285，E-mail: cstsou@mail.ntcb.edu.tw

1. 緒論

存貨管理的目的便是如何運用最少的成本維持高度的服務水準，降低缺貨的可能性以滿足顧客對產品的需求。一般而言，企業為了維持較高的服務水準，勢必會保有較多的存貨，但往往會造成資金的積壓及管理上的困擾；相反的，企業若基於成本的考量保有較少的存貨，容易造成經常性的缺貨，使企業之商譽受損與顧客流失。如何在滿足顧客的需求以及降低企業的成本兩衝突之目標間做出權衡取捨，便是存貨控制所面臨的一大挑戰。另外一方面，隨著時代的變遷，環境變得相當複雜，在發生缺貨的狀況下，消費者的態度不盡相同，某些顧客願意等待廠商將貨品補足後再繼續購買；而某些顧客則不願意等待，轉而向其他廠商購買，因此廠商便應考量這兩種狀況，發展不同的存貨控制模式。存貨控制是工業工程與作業研究領域的議題，除了 Harris (1913)提出的經濟訂購量模式外，常見的存貨控制機制是 (s, Q) 系統，其相關的目標概分為極小化存貨攸關成本並極大化顧客服務水準，這些項目彼此之間往往是衝突的，因此在本質上是屬於多目標決策問題，可運用多目標最佳化 (multi-objective optimization, MOPs) 來規劃存貨控制的決策，以協助管理者在多個存貨控制目標下，決定其訂貨之批量大小及安全庫存，維持較佳的服務水準。

過去在處理多目標的問題時，普遍使用兩種方法。第一種在求解時，先由分析人員在目標函數前設定權重，並將加權後的各個目標予以加總，使多目標規劃問題轉化成單目標的規劃問題，再利用傳統單目標規劃方式求解，此法稱為權重法 (weighted method)；另一種作法是基於計算上的考量，只將其中某項目標最佳化，其餘目標皆轉為限制式，藉由限制其他目標在不同的目標值上進而求得一系列相對應的單一目標最佳解，此法稱

為 ε -限制法 (ε -constraint method) (Deb, 2004)。這些求解方法最後只能得到近似解，無法保證是非凌越解 (non-dominated solution)，而且此種轉換方式下，決策者需要具備某些問題領域知識，才能賦予各目標權重，但在衝突目標的前提下，決策者很難釐清個別目標的相對重要性為何？因此，如何找到禁得起各目標考驗的最佳決策，便是多目標存貨控制中的一項挑戰。再者，傳統方式通常一次只能求解一個柏拉圖最佳解 (Pareto-optimal solution)，為了取得柏拉圖最佳解前緣 (Pareto-optimal front)，必須執行多次最佳化運算，故可能承受相當重的計算負荷，而且某些傳統方法對柏拉圖最佳解前緣的形狀很敏感。因此我們發現，要將傳統方法設計成適合多目標最佳化的演算法並不容易，就算找到最佳化的演算法，也無法有效搜尋最佳解，而微粒群最佳化 (particle swarm optimization, PSO) 是基於隨機多點並行的演算方式，非傳統單點循序搜尋方式，因此具有強大快速的搜尋能力，可避陷入局部最佳化 (local optimum)，應可克服上述傳統求解方式的困難，也因為微粒群最佳化演算法必須調整的參數較少，所以近年來國內外有許多研究採用微粒群演算法來尋求多目標決策問題的最佳解。

但如何求取多目標存貨控制問題的非凌越解又是面臨的另一項難題，近年來利用仿生型的最佳化演算法求解多目標問題成為一種趨勢 (Veldhuizen and Lamont, 2000)，而微粒群最佳化便屬於仿生型的智慧型演算法，此法下一次可求取多個非凌越解，提供多種決策的選擇，因此本研究便運用微粒群演算法來求解 (s, Q) 的多目標存貨控問題。除此之外，多目標最佳化有個重要觀念，就是在各個目標必須是彼此衝突的，如此一來，才有目標間權衡取捨的問題存在，但 Agrell (1995) 的模式中，其三個與存貨有關的目標彼此之間並非完全衝突，似乎出現了多餘 (redundant)

的狀況。因此，為了避免存貨控制模式出現多餘的目標，應將相互衝突的目標予以保留，並把多餘的刪除，以發展適當的多目標存貨模式。

在眾多的演算法中，如何提升演算法的效能也是一個重要的研究方向，Sierra and Coello (2004)曾將基因演算法中的突變機制加入微粒群演算法中，並利用該演算法求解多目標問題，最後將求解的結果與其他不同的演算法比較，發現加入突變機制的演算法具有不錯的效能。因此，本文將仿照過去學者的做法，為提升演算法搜尋不同存貨模式之非凌越解的能力，使用加入區域搜尋 (local search) 與集群 (clustering) 機制的混合式 (hybrid) 之多目標微粒群最佳化 (multi-objective particle swarm optimization, MOPSO) 求解不同狀況的存貨模式，以克服傳統多目標決策方法的瓶頸與限制。

本研究以混合式多目標微粒群最佳化 (hybrid multi-objective particle swarm optimization, HMOPSO) 的啟發式演算法處理多目標存貨控制問題，可免除傳統求解方式所遭遇到的困難，並可協助管理者作出最佳化的存貨控制決策，提升企業競爭力。本研究目的具體描述如下：1. 以 HMOPSO 與 SPEA 求解「銷售損失」下之三目標的存貨控制模式，並以績效衡量指標比較兩種演算法非凌越解之優劣。2. 為避免多目標存貨控制問題出現多餘的 (redundant) 目標，因此將「銷售損失」下三個目標之存貨控制模式分別轉換成兩個雙目標之存貨控制模式 (命名為缺貨次數模式與缺貨數量模式)，比較兩個存貨控制模式下非凌越解之差異，供存貨管理者選擇較適合之缺貨衡量模式。3. 將不同存貨模式的求解結果提供給管理者參考，讓決策者能事先估計在維持一定程度之服務水準應付出多少成本，以最小的存貨總成本，提供顧客完善的服務，並使企業資源能夠作最佳配置，以獲取競爭優勢。

2. 多目標銷售損失存貨控制模式

Agrell 於 1995 年提出 (s, Q) 系統在缺貨後補狀況下的存貨控制模式，該研究的三個目標分別為極小化年度預期存貨攸關成本（含訂購成本與持有成本）、年度預期缺貨次數與缺貨數量。此外，不同的消費者面對廠商發生缺貨的狀況所產生的反應也不盡相，例如某些顧客則不願意等待而轉向其他廠商購買，在不願等待補的情況下便會造成廠商的商譽受損與顧客流失，亦即產生銷售損失情況，為了提升服務水準，在發展存貨控制模式時廠商便應將銷售損失的因素考量進去，因此我們將 Agrell 的存貨控制模式延伸至銷售損失的狀況，建立銷售損失狀況下的多目標存貨控制模式。

首先介紹 Agrell (1995) 所提出的三個目標的缺貨後補存貨控制模式，該研究的假設如下：

1. 模式僅探討單一品項。
2. 該品項在前置時間內之需求 (D_L) 為常態分配，其平均數為 μ_L ，變異數為 σ_L^2 。
3. 前置時間確定已知且為固定，當存貨水準等於或低於再訂購點 (s) 時，立即向供應商訂購固定的批量 (Q) 。
4. 相較於平均存貨，平均缺貨很小，因此可以忽略。

模式中包含三個成本與缺貨有關的目標分別如下：

目標 1：極小化預期總攸關成本，包含了整年的訂購及持有成本。

目標 2：極小化每年預期缺貨次數，為缺貨機率與年度採購次數 $(\frac{D}{Q})$ 之乘積。

目標 3：極小化每年預期缺貨數量，前置時間內的需求量大於 s 時，便產生缺貨的情況，缺貨的數量為 $(x-s)$ 。

模式的之決策變數為訂購批量(Q)及安全因子(k)，兩決策變數的上下限制條件如下：訂購批量必須大於等於零，小於等於預期年需求量；而安全因子必須大於等於零，小於等於預期年需求除以前置時間內需求標準差，數學模式如下：

$$\text{minimize } C^B(Q, k) = \frac{SD}{Q} + hc \left(\frac{Q}{2} + k\sigma_L \right) \quad (1)$$

$$\text{minimize } N(Q, k) = \frac{D}{Q} (1 - \Phi(k)) \quad (2)$$

$$\begin{aligned} \text{minimize } B(Q, k) &= \frac{D}{Q} \int_k^\infty (x-s) f_{D_L}(x) dx \\ &= \frac{D\sigma_L}{Q} (\varphi(k) - k(1 - \Phi(k))) \end{aligned} \quad (3)$$

$$\text{subject to } 0 \leq Q \leq D, \quad (4)$$

$$0 \leq k \leq \frac{D}{\sigma_L}. \quad (5)$$

其中符號代表意義如下：

c ：採購單位成本。

S ：訂購成本。

h ：年度存貨持有成本比率（佔採購單位成本之比率）。

D ：預期年需求量。

$\varphi(x)$ ：標準常態分配之機率密度函數。

$\Phi(x)$ ：標準常態分配之累積分佈函數。

由於本模式假設前置時間為標準常態分配，可透過安全因子推算前置時間的缺貨機率，進而透過缺貨機率計算缺貨次數與缺貨數量，此種存貨控制模式不必估計缺貨成本，能避免在資訊不完全的狀況下錯估缺貨成本的情形。

2.1 三目標之銷售損失存貨控制模式

但在銷售損失的情況下，廠商為了避免顧客不願意等待補貨所造成的銷售損失，會保有更多的存貨以預防因缺貨所造成的顧客流失與服務水準下降，然而廠商應增加多少的存貨才能預防銷售損失呢？根據 Tersine (1994)的看法，增加的庫存量即等於預期年度缺貨數量，一旦廠商的庫存量增加，持有成本便會提高，因此在建構此類存貨控制模式時，總攸關成本的目標函數在持有成本部份應加入年度預期缺貨的數量與單位持有成本的乘積。在這個模式中，第一個目標為極小化預期總攸關成本，與缺貨後補相較，持有成本增加的部分即為預期年度缺貨數量乘以單位持有成本，所以將第一個目標改寫為(6)式，即銷售損失狀況下的存貨攸關成本，第二第三個目標仍為極小化預期年度缺貨次數與數量，即為(2)與(3)式，決策變數上下界與缺貨後補狀況下相同，即為(4)與(5)式，數學模式如下：

$$\text{minimize } C^L(Q, k) = \frac{SD}{Q} + hc \left(\frac{Q}{2} + k\sigma_L + \frac{D\sigma_L}{Q} (\varphi(k) - k(1 - \Phi(k))) \right) \quad (6)$$

$$\text{minimize } N(Q, k) = \frac{D}{Q} (1 - \Phi(k))$$

$$\text{minimize } B(Q, k) = \frac{D\sigma_L}{Q} (\varphi(k) - k(1 - \Phi(k)))$$

$$\text{subject to } 0 \leq Q \leq D ,$$

$$0 \leq k \leq \frac{D}{\sigma_L}.$$

我們不難發現年度缺貨次數及缺貨數量這兩個目標並非完全衝突，似乎存在多餘的狀況，但多目標規劃的原則之一便是希望所有的目標應要相互衝突，因此在我們的研究中將此三個目標的存貨控制模式分為兩個雙目標之存貨控制模式，分別命名為缺貨次數與缺貨數量模式，並將不同模式求解的結果進行比較。

2.2 雙目標之存貨控制模式

2.2.1 銷售損失下缺貨次數模式 (N-model under lost sales)

此模式第一個目標為極小化攸關成本，即(6)式，第二個目標為極小化預期年度缺貨次數，即(2)式，決策變數上下界與缺貨後補的狀況相同，即為(4)與(5)式，數學模式如下：

$$\text{minimize } C^L(Q, k) = \frac{SD}{Q} + hc \left(\frac{Q}{2} + k\sigma_L + \frac{D\sigma_L}{Q} (\varphi(k) - k(1 - \Phi(k))) \right)$$

$$\text{minimize } N(Q, k) = \frac{D}{Q} (1 - \Phi(k))$$

$$\text{subject to } \sqrt{\frac{2SD}{hc}} \leq Q \leq D,$$

$$0 \leq k \leq \frac{D}{\sigma_L}.$$

2.2.2 銷售損失下缺貨數量模式 (B-model under lost sales)

在這個模式中，除了第二個目標為極小化預期年度缺貨數量外（即(3)式），決策變數上下界與缺貨次數模式相同，數學模式如下：

$$\text{minimize } C^L(Q, k) = \frac{SD}{Q} + hc \left(\frac{Q}{2} + k\sigma_L + \frac{D\sigma_L}{Q} (\varphi(k) - k(1 - \Phi(k))) \right)$$

$$\text{minimize } B(Q, k) = \frac{D\sigma_L}{Q} (\varphi(k) - k(1 - \Phi(k)))$$

$$\text{subject to } \sqrt{\frac{2SD}{hc}} \leq Q \leq D,$$

$$0 \leq k \leq \frac{D}{\sigma_L}.$$

3. 混合式多目標微粒群演算法

3.1 多目標最佳化

真實世界中的最佳化的問題大多具有多目標的概念，需以多個目標函數加以判斷解的優劣，並且這些目標函數之間往往是互相衝突的，這些問題我們稱為多目標最佳化問題。典型的多目標最佳化問題具有 K 個連續的實數值目標函數與 M 個限制式，其數學模式如下：

$$\text{Minimize } \bar{\mathbf{f}}(\bar{\mathbf{x}}) = [f_1(\bar{\mathbf{x}}), f_2(\bar{\mathbf{x}}), \dots, f_K(\bar{\mathbf{x}})]^T$$

$$\text{subject to } \bar{\mathbf{x}} \in \Omega \tag{7}$$

$$\Omega = \{ \bar{\mathbf{x}} \mid g_m(\bar{\mathbf{x}}) \leq 0, m = 1, 2, \dots, M \}$$

其中 $\bar{\mathbf{x}} = [x_1, x_2, \dots, x_D]^T$ 為 D 維度的決策向量，每一個 $x_d (d = 1, 2, \dots, D)$ 可以是實數或整數值， $f_i(\bar{\mathbf{x}}) (i = 1, 2, \dots, K)$ 及 $g_m(\bar{\mathbf{x}}) (m = 1, 2, \dots, M)$ 是線性或非線性函數。多目標最佳化問題是在求解滿足 M 個限制式的條件下極小化的目標向量 $\bar{\mathbf{f}}(\bar{\mathbf{x}}) = [f_1(\bar{\mathbf{x}}), f_2(\bar{\mathbf{x}}), \dots, f_K(\bar{\mathbf{x}})]^T$ 的決策向量 $\bar{\mathbf{x}} = [x_1, x_2, \dots, x_D]^T$ 。

對於多目標最佳化問題，很顯然地在所有子目標之間會相互有衝突，

因此幾乎不可能找到一組設計變數為所有子目標的最佳解，要找的解不是所有目標的最佳解，而是所謂的柏拉圖最佳解，但在真實的多目標問題中，往往無法得知真正的柏拉圖最佳解，所以一般在求解多目標最佳化問題時，主要是找尋目標空間中接近柏拉圖最佳解的非凌越解集合，稱為柏拉圖最佳解前緣，因此柏拉圖最佳解前緣中的每一個解又稱為非凌越解，凌越概念的數學表示方式如下：

假設有兩組決策向量 $\bar{\mathbf{u}} = (u_1, u_2, \dots, u_D)$ 及 $\bar{\mathbf{v}} = (v_1, v_2, \dots, v_D)$ ，

1. 若 $f_i(\bar{\mathbf{u}}) \leq f_i(\bar{\mathbf{v}}), \forall i \in \{1, 2, \dots, K\}$ 且 $f_i(\bar{\mathbf{u}}) < f_i(\bar{\mathbf{v}}), \exists i \in \{1, 2, \dots, K\}$ ，則我們稱 $\bar{\mathbf{u}}$ 強凌越 (strictly dominate) $\bar{\mathbf{v}}$ ，以符號 $\bar{\mathbf{v}} \prec \bar{\mathbf{u}}$ 表示。
2. 若 $f_i(\bar{\mathbf{u}}) \leq f_i(\bar{\mathbf{v}}), \forall i \in \{1, 2, \dots, K\}$ ，則我們稱 $\bar{\mathbf{u}}$ 弱凌越 (weakly dominate) $\bar{\mathbf{v}}$ ，以符號 $\bar{\mathbf{v}} \preceq \bar{\mathbf{u}}$ 表示。

所以，非凌越解集合是指在集合中，所有的決策向量均未被集合中其他決策向量所凌越。

3.2 微粒群最佳化

微粒群最佳化的原則為群體成員（在此稱為為粒子）漸漸地往問題空間中較佳的方向移動，以進行求解。在微粒群最佳化演算法中，每隻鳥即為演算法中的一個粒子 x_{nd} ，其中 n 代表第 n 個粒子， d 則代表粒子所搜尋的空間的維度 ($d = 1, 2, \dots, D$)，在求解空間中，進行食物（最佳解）搜尋，所有的鳥類透過距離食物最近的那隻鳥，可知目前距離食物有多遠。求解的過程中，所有的粒子都會有一個對於最佳化問題的適應值 (fitness value)，每個粒子都有一個速度值來決定它們飛翔的方向及距離。粒子不僅根據自己先前的最佳位置 ($Pbest$)，並且依據群體經歷的最佳位置 ($Gbest$) 做修正，每回合粒子可以透過兩種經驗的導引逐漸修正個體的速度與位置，

經過一段時間的學習後將會移到全域最佳解的附近。微粒群演算法的流程說明如下：

1. 初始化：以隨機的方式初始族群中每一個粒子在 d 維空間裡的位置與速度。
2. 計算適應函數：針對所設定的目標函數，評估每一個粒子的適應函數值。
3. 尋找個體最佳解 ($Pbest$)：適應函數值與粒子本身的個體最佳解 ($Pbest$) 記憶比較，更新為較佳函數值。
4. 尋找全域最佳解 ($Gbest$)：個體最佳解與全域最佳解 ($Gbest$) 作比較，如個體最佳解優於全域最佳解，則修正全域最佳解。
5. 更新微粒的速度和位置：利用下列公式改變粒子的速度與位置，即

$$v_{nd}^{new} = w \cdot v_{nd}^{old} + c_1 \cdot r_1 \cdot (p_{nd} - x_{nd}) + c_2 \cdot r_2 \cdot (g_d - x_{nd}) \quad (8)$$

$$x_{nd}^{new} = x_{nd}^{old} + v_{nd}^{new} \quad (9)$$

$$\text{If } v_{nd}^{new} > v_{\max}, v_{nd}^{new} = v_{\max}$$

$$\text{Else if } v_{nd}^{new} < -v_{\max}, v_{nd}^{new} = -v_{\max}$$

x_{nd} ：第 n 個粒子在 d 維度的位置。

x_{nd}^{old} ：第 n 個粒子在 d 維度原本的位置。

x_{nd}^{new} ：第 n 個粒子在 d 維度新的位置。

v_{nd}^{old} ：第 n 個粒子在 d 維度原本的速度。

v_{nd}^{new} ：粒子 n 在 d 維度新的速度。

v_{\max} ：最大的粒子速度，為固定常數。

w ：慣性權重，介於 0.8~1.2 之間的值。

c_1 、 c_2 ：學習因子，介於 1~4 之間的值。

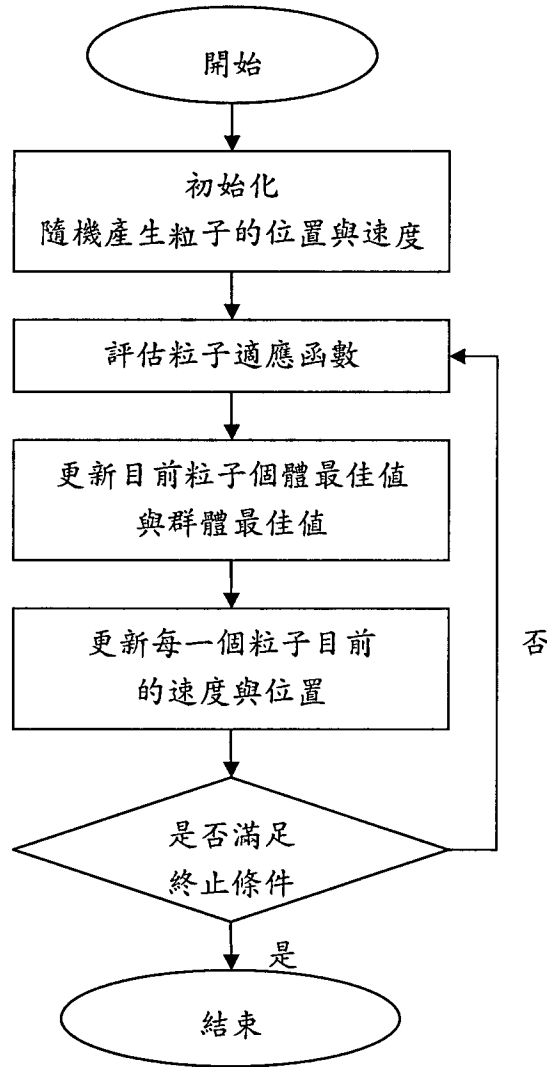
g_d ：第 d 個維度的全域最佳解。

r_1 、 r_2 ：隨機亂數，介於 0~1 之間的值。

6. 若滿足終止條件，則停止，否則回到步驟 2，繼續執行演算法，終止條件通常為達到最大迭代次數。

在(8)式及(9)式的粒子速度及位置更新方法稱為慣性權重法 (inertia weight method)，是 Shi 及 Eberhart (1998a)提出的更新法則，慣性權重可使粒子保持運動的慣性，使其有擴展搜索空間的趨勢，他們對慣性因子有兩種定義，一是將慣性因子定為常數，另一個是讓慣性因子依時間變異。Shi 及 Eberhart (1998b)同樣對慣性因子做討論，不過其基本理論並沒有改變，該研究中提出了慣性因子較好的經驗設定值，因此本研究採用的方法是 Shi 及 Eberhart (1998b)的慣性權重法，此法可清除基本 PSO 對 v_{\max} 的需要，因為 w 本身具有維護全域和局部搜索能力平衡的作用，當 v_{\max} 增加時，可通過減少 w 來達到平衡搜索， w 的減少可使得所需的迭代次數變小，本研究將 v_{\max} 固定為每維變量的變化範圍，只對 w 進行調節。在演算法展開初期，我希望具有較高的探索能力，因此可將慣性因子(w)的值設定得比較大，隨著迭代次數增加，在演算的後期我們希望粒子能作區域的局部搜尋，並增快收斂速度，可將慣性因子設定的值往下調整，此慣性權重法在操作上較為簡單，能避開失速的狀況發生，在過去研究中也有不錯的成果，本研究即採用該法來做為粒子更新速度與位置的方法。

我們可以將微粒群演算法的流程以下圖表示：



《圖1》微粒群最佳化演算法的演算流程

3.3 混合式多目標微粒群演算法

微粒群最佳化演算法具有較強的全域搜尋能力，在整個求解空間的搜尋能力相當不錯，但同時亦有陷入局部最佳解的缺點，因此若在微粒群最佳化演算法中加入區域搜尋的機制，必能強化演算法的效能，提升演算法找到最佳解的能力，加快其收斂速度。此外為了增加解的多樣性與減少電腦運算時間，亦使用加群集機制，以增進演算法效能。過去也有許多學者將微粒群演算法加入不同機制後與其他演算法相比較，發現加入不同機制

之微粒群演算法的效能不但優於傳統的微粒群演算法，也和其他進化式演算法不分上下，甚至有某些效能是超越他們的（Sierra 及 Coello, 2004），以下先介紹區域搜尋與群集的概念。

3.3.1 區域搜尋

本研究中的混合式多目標微粒群演算法運用區域搜尋對非凌越解集合中每一個解的鄰近區域做細微的搜尋以獲得更多及更佳的非凌越解。 $LSITER$ 是進行區域搜尋的次數， δ 是區域搜尋的步距，由於演算法開始時，非凌越解集合尚未穩定，可以在較大的空間搜尋，避免落入區域最佳解，隨這迭代次數增加非凌越解集合逐漸趨近於柏拉圖前緣，此時可縮小搜尋範圍，將步距 δ 減小，在現有之非凌越解集合附近搜尋，可增加非凌越解的多樣性，使許多凌越解更趨近於柏拉圖前緣。

3.3.2 群集

非凌越解數量越多，判斷某一目標向量是否要加入非凌越解的時間越長，會增加電腦運算時間而降低執行效率，所以本研究將利用群集的機制，能將每次迭代產生的非凌越解縮減至某個固定數量。在 MOPSO 演算法中，非凌越解集合是引導粒子前進方向的重要指標，所以當非凌越解在某一區段的目標空間密度過於集中時，微粒被導引到該區段的機率將會增加，所求得的非凌越解集合容易陷入局部最佳解，因此使用群集演算法來縮減非凌越解可將距離接近的非凌越解合併為同一族群，只保留位於族群中間的解，讓非凌越解的散佈密度更為平均，增加解的多樣性。另外，Zitzler 及 Thiele (1999) 過去採用平均聯結法（average linkage method）來縮減非凌越解集合有相當不錯的表現，因此本研究也採用該法。

結合區域搜尋以及群集機制之混合式微粒群演算法的執行步驟如下：

1. 首先設定迭代次數(T)、粒子數(N)、決策向量的維度(D) (在此為訂購批量及安全因子兩個維度)、儲存非凌越解的陣列 \tilde{A} 。
2. 隨機產生 N 個粒子的初始位置($\bar{\mathbf{x}}_n^D$)與初始速度($\bar{\mathbf{v}}_n^D$)，此時個體的最佳解 $\bar{\mathbf{P}}_n^D$ 亦為粒子的初始位置 $\bar{\mathbf{x}}_n^D$ ，即 $\bar{\mathbf{P}}_n^D = \bar{\mathbf{x}}_n^D$ ；而為所有粒子比較後所得之全域最佳解的目標向量為 $\bar{\mathbf{g}}_n^D = \bar{\mathbf{x}}_n^D$ 。
3. 設定區域搜尋次數 ($LSITER$) 與搜尋步距(δ)，在非凌越集合(\tilde{A})中抽取 n 個非凌越解進行區域搜尋，判斷微粒($\bar{\mathbf{x}}_n^D$)是否不被所有非凌越解所凌越，若 $\bar{\mathbf{x}}_n^D$ 不被所有非凌越解所凌越，則將 $\bar{\mathbf{x}}_n^D$ 加入 \tilde{A} 中，並移除 \tilde{A} 中被 $\bar{\mathbf{x}}_n^D$ 所凌越的解。
4. 在每一次迭代中，粒子先從 \tilde{A} 中隨機選出一組非凌越解作為全域最佳解($\bar{\mathbf{g}}_n^D$)，再依 (8)式、(9)式改變位置($\bar{\mathbf{x}}_n^D$)及速度($\bar{\mathbf{v}}_n^D$)，並依目標函式及限制式計算出存貨規劃的目標向量($\bar{\mathbf{y}}_n^K$)，再判斷微粒($\bar{\mathbf{x}}_n^D$)是否加入 \tilde{A} 。若 $\bar{\mathbf{x}}_n^D$ 弱凌越 $\bar{\mathbf{P}}_n^D$ ，或 $\bar{\mathbf{x}}_n^D$ 與 $\bar{\mathbf{P}}_n^D$ 相互不強凌越，則更新粒子本身的個體最佳解 $\bar{\mathbf{P}}_n^D$ 。
5. 最後，以 *ClusterNondominatedSol()* 函數縮減 \tilde{A} 至固定數量，以維持非凌越解的多樣性，演算法持續執行到最大迭代數(T)後結束， \tilde{A} 即為非凌越解集合。

《表1》混合式多目標微粒群演算法虛擬碼

```

01  $\tilde{A} = \emptyset$ 
02  $\{\bar{\mathbf{x}}_n^D, \bar{\mathbf{v}}_n^D, \bar{\mathbf{p}}_n^D, \bar{\mathbf{g}}_n^D\}_{n=1}^N = Initialize()$ 
03 for  $t = 1$  to  $T$ 
04  $\delta = ((\max\_ \delta - \min\_ \delta) * (T - t) / T) + \min\_ \delta$ 
05  $LocalSearch(LSITER, \delta, \tilde{A})$ 
06 for  $n = 1$  to  $N$ 
07  $\bar{\mathbf{g}}_n^D = Random(\tilde{A})$ 
08 for  $d = 1$  to  $D$ 
09  $v_{nd}^{new} = w \cdot v_{nd}^{old} + c_1 \cdot r_1 \cdot (p_{nd} - x_{nd}) + c_2 \cdot r_2 \cdot (g_{nd} - x_{nd})$ 
10  $x_{nd}^{new} = x_{nd}^{old} + v_{nd}^{new}$ 
11 end for
12  $\bar{\mathbf{y}}_n^K = \bar{\mathbf{f}}(\bar{\mathbf{x}}_n^D) = [f_1(\bar{\mathbf{x}}_n^D), f_2(\bar{\mathbf{x}}_n^D), \dots, f_K(\bar{\mathbf{x}}_n^D)]$ 
13  $NondominatedSet(\bar{\mathbf{x}}_n^D)$ 
14 if  $(\bar{\mathbf{x}}_n^D \preceq \bar{\mathbf{p}}_n^D)$  or  $(\bar{\mathbf{x}}_n^D \prec \bar{\mathbf{p}}_n^D)$ 
15  $\bar{\mathbf{p}}_n^D = \bar{\mathbf{x}}_n^D$ 
16 end if
17 end for
18  $ClusterNondominatedSol()$ 
19 end for

```

3.4 非凌越解績效評估

本研究主要探討銷售損失的狀況，在不同狀況下，比較不同存貨控制模式（N-model 與 B-model）非凌越解的差異，所採用的績效衡量指標為覆蓋率（set coverage metric）、非凌越解間距（spacing）與最大散佈距離（maximum spread），以下我們將介紹這三種績效衡量指標。

1. 覆蓋率：（Okabe、Jin 及 Sendhoff，2003）

$C(U, V)$ 稱之為凌越覆蓋率，該數值是計算非凌越集合 V 中的所有解被

U 中的解所弱凌越個數的比率，可用來判別兩個集合中解的正確性，以及接近柏拉圖前緣的程度，其計算方式如下：

$$C(U, V) = \frac{|\{b \in V \mid \exists a \in U, a \preceq b\}|}{|V|} \quad (10)$$

$|\cdot|$ 指的是在某個模式中非凌越解集合的數量，其中未被參考集凌越的比率高，表示非凌越解的品質愈佳，愈接近柏拉圖前緣。

由於缺貨次數模式的目標為極小化每年預期攸關成本以及每年預期缺貨次數；缺貨數量模式之目標為極小化每年預期攸關成本以及每年預期缺貨數量，兩模式之攸關成本是相同的目標向量，但缺貨次數與缺貨數量則因為單位不同，因此無法將缺貨次數與缺貨數量的非凌越解之優劣作公平的比較，因此本研究在利用混合式的多目標微粒群最佳化演算法求解出各模式的訂購批量與安全因子後，透過安全因子來計算服務水準，以攸關成本與服務水準來評估非凌越解之優劣，判斷那個模式的非凌越解最接近真實的柏拉圖前緣，其中服務水準的計算方式列於(11)式 (Silver et al., 1998)。

$$SL = 1 - P(D_L \geq \mu_L + k\sigma_L) \quad (11)$$

2. 非凌越解間距：(Deb, 2004)

用來評估非凌越解分布密集程度，在計算間距與最大散佈距離前，都必須將各目標函數值予以標準化 (normalization)，而後利用(12)式計算非凌越解的間距。

$$S = \sqrt{\frac{1}{|A|} \sum_{i=1}^{|A|} (d_i - \bar{d})^2} \quad (12)$$

其中 d_i 是與鄰近解最短距離，亦即 $d_i = \min_{j \in \tilde{A} \wedge j \neq i} \sum_{k=1}^K |f_k^i - f_k^j|$ ，其中 f_k^i 代表在 \tilde{A} 中第 i 個元素的第 k 目標，且 \tilde{A} 為柏拉圖最佳化集合中的某一個集合，因為 \bar{d} 是 d_i 的平均距離，因此 $\bar{d} = \sum_{i=1}^{|\tilde{A}|} \frac{d_i}{|\tilde{A}|}$ 。

3. 最大散佈距離

在計算非凌越解集合中相距最遠的兩個非凌越解之間的距離，將標準化後的目標函數利用(13)式計算最大散佈距離，此值愈大表示非凌越解的分佈愈為寬廣。

$$D = \sqrt{\sum_{k=1}^K \left(\max_{i=1}^{|\tilde{A}|} f_k^i - \min_{i=1}^{|\tilde{A}|} f_k^i \right)^2} \quad (13)$$

4. 銷售損失之實例驗證

表 2 為某財團法人醫院藥劑科之藥品入出庫記錄，本研究利用此資料庫進行存貨模式之求解，並比較不同存貨模式差異，進而探討多目標微粒群演算法的實用性。

《表2》藥品範例基本資料表

藥品代號	年需求量 (D)	前置時間內需求量標準差 (σ_L)	訂購成本 (S)	單位採購成本 (c)	持有成本比率 (h)
1	3412	53.354	80	27.5	0.26
2	490	5.027	80	241	0.30
3	4736	57.911	135	29.41	0.30
4	200	2.969	80	233	0.26
5	215	2.781	80	435	0.30
6	22774	245.333	135	12.6	0.26
7	10557	85.395	135	2.14	0.26

HMOPSO 的參數有粒子數、迭代次數、粒子之移動速率、區域搜尋等參數必須設定。在測試參數的過程中，將決策空間各維度之最大速率分別設定為決策變數上限的 1/100 與 1/200，整體來說最大速率在 1/100 時，決策變數的變異係數較小；此外，將迭代次數設定為分別設定為 100 與 50，發現在迭代 100 次時，決策變數的變異係數($CV_Q=4.940\sim 5.717$ ， $CV_k=3.481\sim 4.278$) 小於迭代次數 50 次的 ($CV_Q=7.407\sim 8.797$ ， $CV_k=4.376\sim 5.877$)；粒子數則各分別設定為 40 與 60，粒子數為 40 時其決策變數的變異係數 ($CV_Q=4.940$ ， $CV_k=3.481$) 小於 60 的變異係數 ($CV_Q=5.717$ ， $CV_k=3.590$)，且在粒子數為 40 時執行時間較短。因此本研究將最大速率設定為 1/100，迭代次數設為 100 次，粒子數設定為 40，並進行 1 次的區域搜尋，同時為了後續方便比較演算法的績效，我們將最大非凌越解組數設為 30 組。表 3 為 HMOPSO 參數校調的結果。

《表3》HMOPSO參數校調測試結果

最大 速率	迭代 次數	粒子數	平均值		標準差		變異係數		執行時間
			\bar{x}_Q	\bar{x}_k	s_Q	s_k	CV_Q	CV_k	
1/100	50	40	307.12	2.16	24.044	0.1132	7.828	5.240	21.075
1/100	50	60	308.37	2.13	22.842	0.1252	7.407	5.877	26.716
1/100	100	40	312.74	2.14	15.451	0.0745	4.940	3.481	45.123
1/100	100	60	309.91	2.15	17.718	0.0772	5.717	3.590	48.473
1/200	50	40	309.12	2.10	25.146	0.0919	8.134	4.376	18.717
1/200	50	60	306.72	2.18	26.984	0.1127	8.797	5.169	27.678
1/200	100	40	311.19	2.12	16.451	0.0907	5.286	4.278	47.213
1/200	100	60	310.64	2.17	17.718	0.0897	5.703	4.133	50.841

4.1. SPEA 求解

SPEA 是 Zitzler 於 1999 年所提出的，它是基因演算法的一種，同樣透過選擇、交配與突變等機制進行演化，特別之處在於其適應值的評估方

式，並在演算法中加入群集的機制。SPEA 在演化的過程中，會將非凌越解集合先儲存於外部的族群裡，外部族群中的非凌越解適應值大小是以它凌越基因族群數之多寡而決定的，外部族群的非凌越解凌越內部族群的數量越多，則其適應值越大，基因內部族群的適應值則是以外部族群中所有凌越內部族群的非凌越解之適應值加總，最後合併內部與外部族群進行下一次的演化，並以群集演算法來縮減外部族群中非凌越解的數量，以防止產生過多的非凌越解而降低演算法的執行效能，此演算法在多目標的背包問題與六個測試函數的比較中效能勝過 VEGA (vector evaluated genetic algorithm)、NPGA (niched Pareto genetic algorithm)、HLGA (Hajela's and Lin's genetic algorithm) 以及 NSGA (non-dominated sorting genetic algorithm) 等演算法。由於 SPEA 與 HMOPSO 都屬於仿生型的最佳化演算法，況且 SPEA 在過去研究成果都相當不錯，因此本研究將利用 SPEA 來求解，並與 HMOPSO 比較。

SPEA 必須設定交配率與突變率，基於 GA 的算法多將交配率設為 0.7 至 0.9 間，而突變率則介於 0.05 至 0.2 間。因此，本研究將交配率設定為 0.7、0.8 及 0.9，突變率則設定為 0.05、0.1 及 0.2，配對出 9 組參數組合，先行求解前述之多目標存貨問題，結果顯示突變率設定為 0.9，交配率設定為 0.2 時其非凌越解的覆蓋率表現最佳，因此後續 SPEA 的交配率與突變率分別設定為 0.9 與 0.2，最大非凌越解組數設定為 30 組，以進一步與 HMOPSO 比較。

4.2. 銷售損失下之三目標存貨控制之求解

本節將分別採用 HMOPSO 以及 SPEA 兩種進化式的演算法求解在銷售損失狀況下的多目標存貨控制問題，目標為極小化年度存貨攸關成本、年度缺貨次數、年度缺貨數量。首先，說明列示 HMOPSO 與 SPEA 的求

解結果；接下來將 HMOPSO 與 SPEA 求解結果進行比較。

4.2.1 以 HMOPSO 與 SPEA 求解

1. 以 HMOPSO 求解：

HMOPSO 粒子數設定為 40，而迭代次數則是設定在 100 次，移動速率設定為決策變數上限的 1/100，進行 1 次的區域搜尋，最大非凌越解組數設定為 30。表 4 為以 HMOPSO 求解藥品代號 1 在銷售損失狀況下的非凌越解與目標函數值，30 組非凌越解訂購批量由 214.4 分佈至 505.6 不等，而安全因子則是分佈在 1.48 至 4.68。

2. 以 SPEA 求解：

SPEA 中交配率與突變率分別設為 0.9 與 0.2，最大非凌越解組數設定為 30 組。表 5 為 SPEA 求解藥品代號 1 的非凌越解與目標函數值。

表 5 的 30 組非凌越解，訂購批量由 163.34 分佈至 434.4 不等，而安全因子則是分佈在 1.89 至 4.81 的範圍之中。

《表4》以HMOPSO求解之非凌越解及目標函數值

非凌越解代號	Q	k	C	N	B
1	359.54	3.79	3491	0.0005	0.07
2	338.76	2.67	3045	0.0364	1.34
3	316.88	1.8	2764	0.3873	11.57
4	505.6	4.61	4106	0.0000	0
5	258.15	2.33	2898	0.1289	4.01
6	299.64	1.86	2768	0.3582	10.66
7	274.66	2.43	2924	0.0917	2.96
8	308.61	1.92	2785	0.3031	9
9	314.15	1.63	2736	0.5610	17.04
10	292.47	2.53	2960	0.0645	2.18
11	314.15	1.69	2743	0.4952	14.94
12	290.7	2.21	2855	0.1576	4.78
13	307.53	1.76	2752	0.4356	13.05
14	294.46	1.85	2765	0.3728	11.1
15	298.39	2.1	2826	0.2033	6.07
16	302.87	1.97	2794	0.2747	8.16
17	298.74	1.48	2723	0.7948	24.66
18	294.24	1.58	2727	0.6630	20.27
19	306.3	1.54	2725	0.6897	21.2
20	292.11	1.7	2740	0.5215	15.71
21	324.67	1.99	2812	0.2443	7.26
22	305.95	1.58	2728	0.6376	19.49
23	214.4	4.68	3825	0.0000	0
24	305.79	1.51	2724	0.7327	22.63
25	307.63	1.61	2731	0.5968	18.17
26	439.38	3.03	3351	0.0085	0.43
27	296.25	3.19	3201	0.0070	0.44
28	310.36	1.79	2758	0.4042	12.08
29	301.88	1.6	2729	0.6207	18.92
30	287.04	1.76	2748	0.4667	13.98

附註：訂購批量與安全因子四捨五入至小數第二位，攸關成本四捨五入至整數，年度缺貨次數四捨五入至小數點第四位，缺貨成本均四捨五入至小數點第二位。

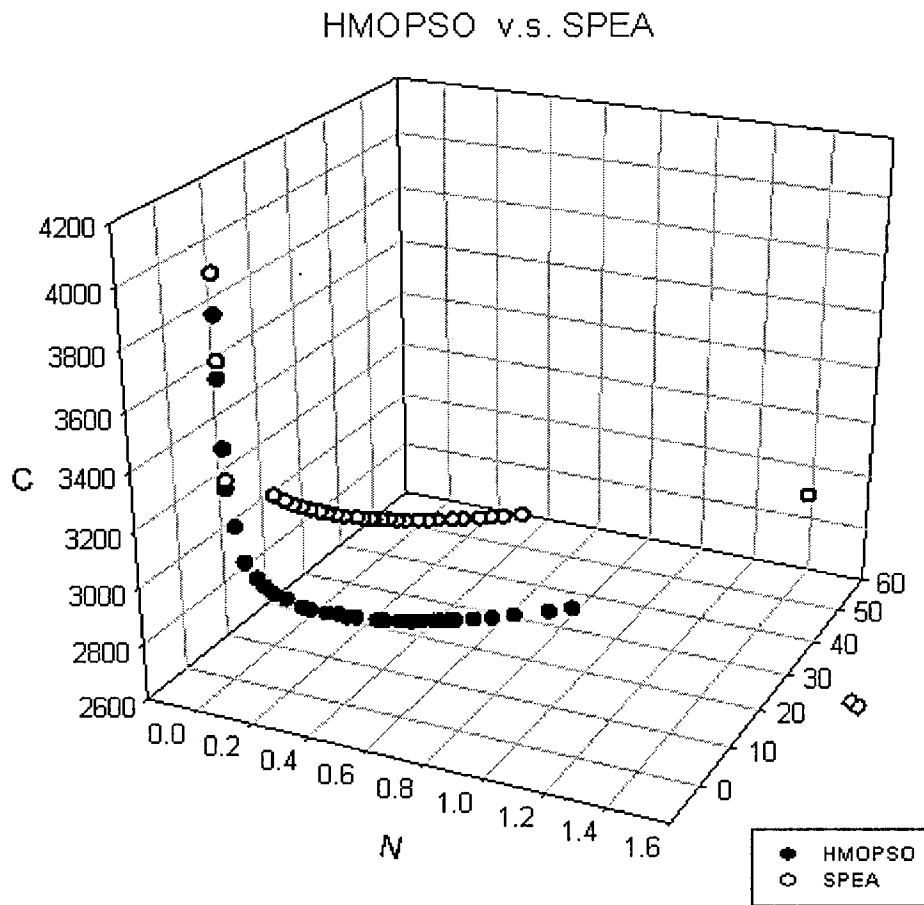
《表5》以SPEA求解之非凌越解及目標函數值

非凌越解代號	Q	k	C	N	B
1	434.4	3.44	3495	0.0018	0.16
2	163.82	3.18	3471	0.0131	0.81
3	163.51	3.98	3773	0.0004	0.09
4	163.5	2.96	3393	0.0291	1.38
5	163.79	4.81	4087	0.0000	0
6	163.5	2.81	3340	0.0483	1.98
7	163.82	2.7	3300	0.0686	2.57
8	163.5	2.64	3282	0.0828	2.98
9	163.49	2.58	3263	0.0994	3.45
10	163.67	2.53	3246	0.1152	3.89
11	163.46	2.49	3235	0.1297	4.29
12	163.51	2.46	3225	0.1414	4.61
13	163.46	2.42	3214	0.1585	5.09
14	163.34	2.38	3203	0.1774	5.62
15	163.66	2.35	3193	0.1924	6.03
16	163.83	2.32	3183	0.2087	6.48
17	163.5	2.29	3178	0.2268	6.99
18	163.83	2.27	3170	0.2388	7.32
19	163.81	2.25	3165	0.2518	7.69
20	163.82	2.22	3158	0.2725	8.27
21	163.8	2.19	3151	0.2946	8.9
22	163.51	2.16	3146	0.3188	9.59
23	163.83	2.14	3140	0.3348	10.04
24	163.81	2.13	3138	0.3434	10.29
25	163.83	2.12	3136	0.3521	10.54
26	163.83	2.1	3132	0.3702	11.06
27	244.71	1.89	2798	0.4096	12.18
28	163.81	2.09	3130	0.3796	11.33
29	163.51	2.07	3129	0.3996	11.91
30	163.51	2.06	3127	0.4096	12.2

附註：訂購批量與安全因子四捨五入至小數第二位，攸關成本四捨五入至整數，年度缺貨次數四捨五入至小數點第四位，缺貨數量四捨五入至小數點第二位。

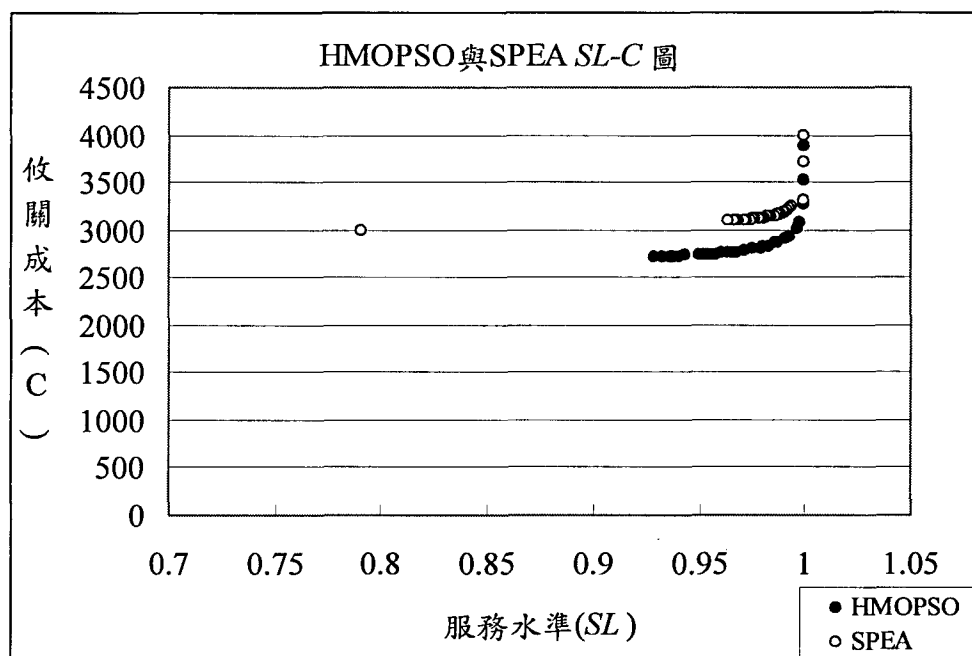
4.2.2 非凌越解之績效評估

圖 2 是以 HMOPSO 求解的 30 組非凌越解與 SPEA 求解的 30 組非凌越解在目標空間的的分佈情形。



《圖2》 HMOPSO與SPEA在目標空間的分佈

將 HMOPSO 與 SPEA 所求解的非凌越解透過安全因子(k)計算服務水準(SL)，以攸關成本與服務水準比較二種進化式演算法的求解結果，其服務水準與成本的散佈圖形呈現於圖 3。



《圖3》HMOPSO與SPEA $SL-C$ 圖

由圖 3 我們不難發現，HMOPSO 的散佈性較佳，且在相同的服水準下可以找到攸關成本較低的解。在銷售損失的狀況下，無論是 HMOPSO 或 SPEA，找到的解服務水準都比較高，在銷售損失的圖形中，服務水準則集中於 0.9 以上的區域。因為在銷售損失的狀況下，顧客不願等待補貨，廠商擔心因缺貨而造成顧客流失，所以會維持較的服務水準以因應前置時間的需求，預防缺貨所造成的損失。

接著同樣為了比較 HMOPSO 與 SPEA 求解的結果，分別將 HMOPSO 與 SPEA 試行 30 次，以進行比非凌越解之績效評估。

1. 覆蓋率

HMOPSO 與 SPEA 分別試行 30 次後，其覆蓋率如表 6，HMOPSO 的解凌越 SPEA 的比率平均為 89.11%，變異係數為 5.8%，而 SPEA 的解凌越 HMOPSO 的比率平均為 3.44%，變異係數為 89.82%。

《表6》HMOPSO與SPEA覆蓋率

試行 次數	$C(HMOPSO, SPEA)$	試行 次數	$C(HMOPSO, SPEA)$	試行 次數	$C(SPEA, HMOPSO)$	試行 次數	$C(SPEA, HMOPSO)$
第1次	0.9333	第16次	0.8333	第1次	0.0333	第16次	0.0667
第2次	0.8333	第17次	0.9000	第2次	0.0667	第17次	0.0000
第3次	0.9000	第18次	0.9333	第3次	0.0333	第18次	0.0000
第4次	0.9000	第19次	0.8667	第4次	0.0667	第19次	0.0333
第5次	0.9333	第20次	0.8667	第5次	0.0000	第20次	0.0000
第6次	0.8667	第21次	0.8667	第6次	0.1000	第21次	0.0333
第7次	0.9000	第22次	0.7667	第7次	0.0667	第22次	0.0333
第8次	0.8667	第23次	0.9333	第8次	0.0333	第23次	0.0333
第9次	0.8000	第24次	0.9667	第9次	0.0000	第24次	0.0000
第10次	0.9667	第25次	0.8333	第10次	0.0000	第25次	0.1000
第11次	0.9000	第26次	0.8667	第11次	0.0667	第26次	0.0000
第12次	0.8333	第27次	0.9333	第12次	0.0333	第27次	0.0667
第13次	0.9000	第28次	1.0000	第13次	0.0333	第28次	0.0000
第14次	0.9000	第29次	0.8667	第14次	0.0667	第29次	0.0000
第15次	0.9333	第30次	0.9333	第15次	0.0333	第30次	0.0333
平均數	0.8911***			0.0344			
中位數	0.9000			0.0333			
最大值	1.0000			0.1000			
最小值	0.7667			0.0000			
標準差	0.0517			0.0309			
變異 係數	5.8%			89.82%			

附註：***表 $p < 0.01$ ，**表 $p < 0.05$ ，*表 $p < 0.1$

將兩演算法試行 30 次的覆蓋率作成對樣本的 T 檢定，t 值為 69.692，自由度為 29，p 值為 0.000，小於 0.01，因此 HMOPSO 的非凌越解凌越 SPEA 的非凌越解比率顯著高於 SPEA 之非凌越解凌越 HMOPSO 的非凌越解比率，所以 HMOPSO 的非凌越解比 SPEA 的非凌越越解更接近柏拉圖前緣。

2. 非凌越解間距

表 7 為 HMOPSO 與 SPEA 分別試行 30 次其非凌越解的間距，HMOPSO 平均解的間距為 0.0237，變異係數為 22.36%，而 SPEA 的間距為 0.0334，變異係數為 123.65%。

《表 7》HMOPSO 與 SPEA 非凌越解間距

試行 次數	HMOPSO 間距	試行 次數	HMOPSO 間距	試行 次數	SPEA 間距	試行 次數	SPEA 間距
第 1 次	0.0115	第 16 次	0.0255	第 1 次	0.0020	第 16 次	0.0122
第 2 次	0.0224	第 17 次	0.0231	第 2 次	0.0138	第 17 次	0.0989
第 3 次	0.0194	第 18 次	0.0249	第 3 次	0.0124	第 18 次	0.1470
第 4 次	0.0210	第 19 次	0.0187	第 4 次	0.0125	第 19 次	0.0144
第 5 次	0.0198	第 20 次	0.0272	第 5 次	0.0199	第 20 次	0.0147
第 6 次	0.0235	第 21 次	0.0264	第 6 次	0.0150	第 21 次	0.0151
第 7 次	0.0262	第 22 次	0.0187	第 7 次	0.0192	第 22 次	0.0235
第 8 次	0.0326	第 23 次	0.0222	第 8 次	0.0225	第 23 次	0.0144
第 9 次	0.0272	第 24 次	0.0306	第 9 次	0.0158	第 24 次	0.0152
第 10 次	0.0174	第 25 次	0.0187	第 10 次	0.0933	第 25 次	0.0088
第 11 次	0.0283	第 26 次	0.0182	第 11 次	0.0124	第 26 次	0.0293
第 12 次	0.0233	第 27 次	0.0391	第 12 次	0.0114	第 27 次	0.0140
第 13 次	0.0231	第 28 次	0.0282	第 13 次	0.1420	第 28 次	0.0210
第 14 次	0.0250	第 29 次	0.0231	第 14 次	0.0245	第 29 次	0.0155
第 15 次	0.0250	第 30 次	0.0194	第 15 次	0.1247	第 30 次	0.0155
平均數	0.0237			0.0334			
中位數	0.0232			0.0155			
最大值	0.0391			0.1470			
最小值	0.0115			0.0020			
標準差	0.0053			0.0413			
變異 係數	22.36%			123.65%			

進一步檢驗其平均數差異是否顯著，因此我們分別做了 HMOPSO 與 SPEA 間距的 T 檢定，t 值為 -1.282，自由度為 58，p 值為 0.250，大於 0.05，

由於 30 次試行下，SPEA 間距的變異較大，所以 HMOPSO 非凌越解的間距與 SPEA 非凌越解的間距的差距並不顯著。

3. 最大散佈距離

表 8 為 HMOPSO 與 SPEA 試行 30 次之最大散佈距離，HMOPSO 試行 30 次之最大散佈距離平均為 0.5604，變異係數為 7.3%。HMOPSO 平均為 0.4630，變異係數為 8.53%。

《表8》HMOPSO之最大散佈距離

試行次數	HMOPSO 最大散佈 距離	試行次數	HMOPSO 最大散佈 距離	試行次數	SPEA最大 散佈距離	試行次數	SPEA最大 散佈距離
第1次	0.4962	第16次	0.6064	第1次	0.4326	第16次	0.4248
第2次	0.5499	第17次	0.5690	第2次	0.4562	第17次	0.4918
第3次	0.5134	第18次	0.5714	第3次	0.4369	第18次	0.4688
第4次	0.5516	第19次	0.5033	第4次	0.4172	第19次	0.4338
第5次	0.5222	第20次	0.5721	第5次	0.5002	第20次	0.4518
第6次	0.5621	第21次	0.5942	第6次	0.4651	第21次	0.4387
第7次	0.5477	第22次	0.5060	第7次	0.4470	第22次	0.5961
第8次	0.6133	第23次	0.5376	第8次	0.4799	第23次	0.4677
第9次	0.5822	第24次	0.6246	第9次	0.4375	第24次	0.4413
第10次	0.5496	第25次	0.5375	第10次	0.4793	第25次	0.4674
第11次	0.5911	第26次	0.5069	第11次	0.4162	第26次	0.5314
第12次	0.5633	第27次	0.6861	第12次	0.3834	第27次	0.4635
第13次	0.5365	第28次	0.5821	第13次	0.4782	第28次	0.4942
第14次	0.5782	第29次	0.5394	第14次	0.5028	第29次	0.4525
第15次	0.5782	第30次	0.5408	第15次	0.4822	第30次	0.4525
平均數	0.5604 ***			0.4630			
中位數	0.5568			0.4598			
最大值	0.6861			0.5961			
最小值	0.4962			0.3834			
標準差	0.0409			0.0395			
變異係數	7.3%			8.53%			

附註：***表 $p < 0.01$ ，**表 $p < 0.05$ ，*表 $p < 0.1$

同樣為檢定最大散佈距離之平均數是否有顯著差異，我們分別做了 HMOPSO 與 SPEA 最大散佈距離的 T 檢定，t 值為 9.374，自由度為 58，p 值為 0.000，小於 0.01，因此，HMOPSO 非凌越解的最大散佈距離顯著大於 SPEA 非凌越解的最大散佈距離。

綜合上述結果，在銷售損失的情況下，加入群集與區域搜尋機制之 HMOPSO 求解出的非凌越解在覆蓋率及最大散佈距離的表現都顯著的優於 SPEA，由於 SPEA 所產生的非凌越解因間距的變異較大，兩演算法在間距的差異並不顯著，HMOPSO 的非凌越解較 SPEA 的非凌越解接近柏拉圖前緣，解的分佈性也較為廣泛。

4.3 銷售損失下之雙目標存貨控制之求解

4.3.1 以 HMOPSO 求解 N 與 B 存貨控制模式

利用 HMOPSO 分別求解銷售損失下之缺貨次數與缺貨數量模式，結果列於表 9。

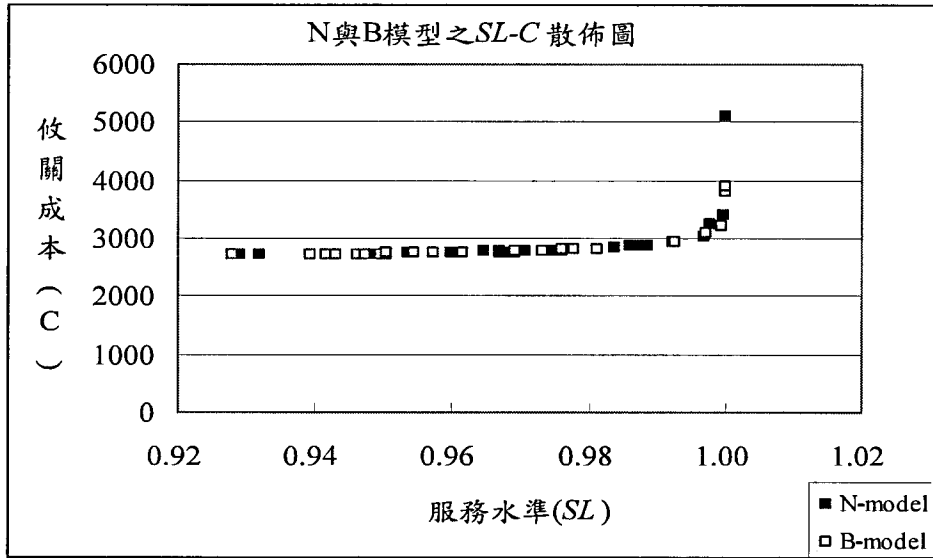
4.3.2 N 與 B 存貨控制模式比較

首先我們透過服務水準及攸關成本來繪製 N 與 B 模式之散佈圖形，圖形的分佈如圖 4 所示。由圖 4 我們發現，兩模式的服務水準愈接近 1，成本攀升的速度就愈快。N 模式與 B 模式在銷售損失的狀況下，服務水準也是集中在高服務水準的區間中，散佈於 0.92~0.99999 中。為了找到在銷售損失下，較具代表性的存貨模式，我們將兩模式用 HMOPSO 各別進行 30 次求解，並比較兩模式的覆蓋率、間距、最大散佈距，以找到較接近柏拉圖前緣，且散佈良好的模式。

《表9》N-model與B-model非凌越解與各目標函數值

非凌越解	N-model					B-model				
	<i>Q</i>	<i>k</i>	<i>C</i>	<i>N</i>	<i>SL</i>	<i>Q</i>	<i>k</i>	<i>C</i>	<i>B</i>	<i>SL</i>
1	294.05	1.55	2725	0.7044	0.9394	303.42	1.46	2722	25.3	0.9279
2	304.15	1.55	2726	0.6810	0.9394	300.36	1.55	2725	21.17	0.9394
3	300.25	1.98	2795	0.2706	0.9761	306.17	1.57	2727	19.9	0.9418
4	298.58	1.47	2722	0.8106	0.9292	299.52	1.57	2726	20.34	0.9418
5	300.25	1.98	2795	0.2706	0.9761	285.3	2.01	2800	7.88	0.9778
6	303.58	1.95	2789	0.2873	0.9744	340.06	2.78	3086	1.02	0.9973
7	291.75	1.69	2738	0.5332	0.9545	307.34	1.58	2728	19.41	0.9429
8	310.97	1.49	2724	0.7490	0.9319	309.38	1.61	2732	18.07	0.9463
9	323.67	2.43	2945	0.0778	0.9925	314.85	2.08	2829	6.04	0.9812
10	312.76	1.68	2741	0.5080	0.9535	320.4	2.44	2946	2.47	0.9927
11	303.58	2.01	2804	0.2491	0.9778	284.9	1.87	2768	10.96	0.9693
12	276.49	2.14	2835	0.1984	0.9838	311.51	1.65	2737	16.45	0.9505
13	862.19	4.44	5093	0.0000	1.0000	295.91	1.87	2769	10.55	0.9693
14	300.7	1.75	2748	0.4552	0.9599	340.06	2.78	3086	1.02	0.9973
15	300.25	1.98	2795	0.2706	0.9761	293.29	1.62	2731	18.65	0.9474
16	285.12	2.28	2876	0.1336	0.9887	296.15	1.64	2733	17.69	0.9495
17	419.32	2.85	3242	0.0165	0.9978	293.52	1.69	2738	15.99	0.9545
18	329.07	1.81	2774	0.3648	0.9649	310.09	1.72	2746	14.16	0.9573
19	321.22	2.25	2884	0.1284	0.9878	340.06	2.78	3086	1.02	0.9973
20	290.47	1.63	2732	0.6068	0.9484	340.06	2.78	3086	1.02	0.9973
21	284.06	2.25	2866	0.1452	0.9878	340.06	2.78	3086	1.02	0.9973
22	310.26	2.2	2861	0.1516	0.9861	297.3	1.65	2734	17.24	0.9505
23	313.06	1.89	2780	0.3202	0.9706	172.02	4.18	3797	0.04	1.0000
24	287.39	1.86	2766	0.3734	0.9686	366.74	4.85	3906	0	1.0000
25	300.25	1.98	2795	0.2706	0.9761	340.06	2.78	3086	1.02	0.9973
26	302.76	3.68	3389	0.0009	0.9999	341.91	1.98	2826	7.06	0.9761
27	295.07	1.84	2763	0.3805	0.9671	302.67	1.93	2784	8.97	0.9732
28	261.33	2.74	3034	0.0379	0.9969	293.18	1.77	2750	13.38	0.9616
29	300.25	1.98	2795	0.2706	0.9761	327.29	1.65	2745	15.66	0.9505
30	322.64	1.84	2776	0.3480	0.9671	313.12	3.23	3226	0.37	0.9994

附註：訂購批量四捨五入至小數點二位，安全因子四捨五入至小數第二位，攸關成本四捨五入至整數，缺貨次數四捨五入至小數第四位，缺貨數量四捨五入至小數第二位，服務水準四捨五入至小數點第四位。



《圖4》 N模式與B模式之SL-C圖

1. 覆蓋率

表 10 是 N 模式與 B 模式攸關成本及服務水準覆蓋率比較表，其中 N 模式的解凌越 B 模式的比率平均為 23.89%，變異係數為 41.77%；B 模式的解凌越 N 模式的比率平均則為 32.44%，變異係數為 24.88%。

《表10》N模式與B模式之覆蓋率

試行次數	C(N,B)	試行次數	C(N,B)	試行次數	C(B,N)	試行次數	C(B,N)
第1次	0.3667	第16次	0.5000	第1次	0.2667	第16次	0.3333
第2次	0.2333	第17次	0.1333	第2次	0.2000	第17次	0.4000
第3次	0.0667	第18次	0.2000	第3次	0.3333	第18次	0.4000
第4次	0.3000	第19次	0.2667	第4次	0.2000	第19次	0.4000
第5次	0.1667	第20次	0.1667	第5次	0.2667	第20次	0.2667
第6次	0.2667	第21次	0.2333	第6次	0.3000	第21次	0.2667
第7次	0.2000	第22次	0.3667	第7次	0.4333	第22次	0.3333
第8次	0.3667	第23次	0.1333	第8次	0.2667	第23次	0.5000
第9次	0.3000	第24次	0.3667	第9次	0.3333	第24次	0.2333
第10次	0.3000	第25次	0.0667	第10次	0.2333	第25次	0.4667
第11次	0.2333	第26次	0.2333	第11次	0.3333	第26次	0.4333
第12次	0.2000	第27次	0.2000	第12次	0.2333	第27次	0.3333
第13次	0.3667	第28次	0.2333	第13次	0.4333	第28次	0.3667
第14次	0.2667	第29次	0.1333	第14次	0.3333	第29次	0.2333
第15次	0.1667	第30次	0.1333	第15次	0.3000	第30次	0.3000
平均數	0.2389***			0.3244			
中位數	0.2333			0.3333			
最大值	0.5000			0.5000			
最小值	0.0667			0.2000			
標準差	0.0998			0.0807			
變異係數	41.77%			24.88%			

附註：***表 $p < 0.01$ ，**表 $p < 0.05$ ，*表 $p < 0.1$

進行兩模式之獨立 T 檢定，t 值為-3.296，自由度為 29，p 值為 0.003，小於 0.01，因此 N 模式解凌越 B 模式解之比率顯著的小於 B 模式解凌越 N 模式解之比率，由 B 模式所產生的解較接近柏拉圖前緣。

2. 非凌越解間距

表 11 為以 HMOPSO 求解分別求解 N 模式與 B 模式 30 次，其非凌越解間距的比較，N 模式在 30 次的演算中非凌越解的間距平均為 0.0083，變異係數 39.75%；B 模式平均為 0.0098，變異係數 61.22%。

《表11》N模式與B模式非凌越解間距

試行次數	N模式間距	試行次數	N模式間距	試行次數	B模式間距	試行次數	B模式間距
第1次	0.0194	第16次	0.0030	第1次	0.0104	第16次	0.0095
第2次	0.0073	第17次	0.0086	第2次	0.0045	第17次	0.0099
第3次	0.0117	第18次	0.0047	第3次	0.0111	第18次	0.0076
第4次	0.0077	第19次	0.0069	第4次	0.0061	第19次	0.0164
第5次	0.0120	第20次	0.0101	第5次	0.0094	第20次	0.0164
第6次	0.0067	第21次	0.0090	第6次	0.0074	第21次	0.0075
第7次	0.0083	第22次	0.0104	第7次	0.0057	第22次	0.0058
第8次	0.0090	第23次	0.0075	第8次	0.0060	第23次	0.0062
第9次	0.0066	第24次	0.0075	第9次	0.0074	第24次	0.0084
第10次	0.0106	第25次	0.0093	第10次	0.0106	第25次	0.0086
第11次	0.0071	第26次	0.0087	第11次	0.0061	第26次	0.0200
第12次	0.0040	第27次	0.0092	第12次	0.0039	第27次	0.0112
第13次	0.0034	第28次	0.0081	第13次	0.0049	第28次	0.0349
第14次	0.0018	第29次	0.0112	第14次	0.0075	第29次	0.0103
第15次	0.0080	第30次	0.0112	第15次	0.0105	第30次	0.0090
平均數	0.0083			0.0098			
中位數	0.0082			0.0085			
最大值	0.0194			0.0349			
最小值	0.0018			0.0039			
標準差	0.0033			0.0060			
變異係數	39.75%			61.22%			

檢驗兩模式非凌越解之間距是否有顯著差異，因此我們將兩模式之間距作獨立 T 檢定，t 值為-1.183，自由度為 58，p 值為 0.242，大於 0.05，因此雖然 N 模式非凌越解的間距比 B 模式小，但未達顯著水準。

3. 最大散佈距離

表 12 為計算 N 模式與 B 模式非凌越解最大散佈距離的結果，N 模式在 30 次的演算中非凌越解的最大散佈距離平均為 0.3898，變異係數 5.92%；B 模式平均為 0.3657，變異係數 8.19%。

《表12》N模式與B模式非凌越解最大散佈距離

試行次數	N模式 最大散佈 距離	試行次數	N模式 最大散佈 距離	試行次數	B模式 最大散佈 距離	試行次數	B模式 最大散佈 距離
第1次	0.4363	第16次	0.3971	第1次	0.3959	第16次	0.3686
第2次	0.4121	第17次	0.3935	第2次	0.4092	第17次	0.3694
第3次	0.3790	第18次	0.3781	第3次	0.3729	第18次	0.4229
第4次	0.4456	第19次	0.3553	第4次	0.3386	第19次	0.3132
第5次	0.4274	第20次	0.4053	第5次	0.3663	第20次	0.3712
第6次	0.4167	第21次	0.3804	第6次	0.4149	第21次	0.3729
第7次	0.4172	第22次	0.3999	第7次	0.3912	第22次	0.3262
第8次	0.3996	第23次	0.3723	第8次	0.3620	第23次	0.3911
第9次	0.4109	第24次	0.4126	第9次	0.3835	第24次	0.3233
第10次	0.3736	第25次	0.3994	第10次	0.3493	第25次	0.3599
第11次	0.3713	第26次	0.4234	第11次	0.3917	第26次	0.3830
第12次	0.3799	第27次	0.3891	第12次	0.3569	第27次	0.3704
第13次	0.4218	第28次	0.3847	第13次	0.3771	第28次	0.3704
第14次	0.3899	第29次	0.3982	第14次	0.2815	第29次	0.3465
第15次	0.3612	第30次	0.4363	第15次	0.3807	第30次	0.3567
平均數	0.3898***			0.3675			
中位數	0.3988			0.3704			
最大值	0.4456			0.4299			
最小值	0.3553			0.2815			
標準差	0.0231			0.0301			
變異係數	5.92%			8.19%			

附註：***表 $p < 0.01$ ，**表 $p < 0.05$ ，*表 $p < 0.1$

兩模式之最大散佈距離作 T 檢定的結果，t 值為 4.570，自由度為 58，p 等於 0.000，小於 0.01，因此 N 模式非凌越解的最大散佈距離顯著的大於 B 模式的最大散佈距離。

由上述幾種比較中可看出：利用 HMOPSO 分別求解 N 模式與 B 模式的非凌越解，B 模式所產生的非凌越解在覆蓋率上的表現顯著優於 N 模式；間距的表現上，N 模式非凌越解的間距雖小於 B 模式，但未達顯著水

準；而在最大散佈距離的表現，N 模式的散佈性顯著地較 B 模式寬廣。

5. 結論

本研究以進化式演算法來求解 (s, Q) 存貨控制模式下的非凌越控制參數，在銷售損失的狀況下，將混合式微粒群演算法之求解結果與 SPEA 比較，同時也檢視兩種存貨控制模式（N 模式與 B 模式）非凌越解優劣，最後將三目標、缺貨次數與缺貨數量之存貨控制模式加以比較。本文所用之 PSO 已經被證實為一收斂快速之全局搜尋算法，其探索（exploration）能力強，加入區域搜尋機制後，可強化其開發（exploit）優化解的能力。尤其對於多目標最佳化問題，其目標空間寬廣，非凌越解為數眾多，結合區域搜尋與群集機制的 HMOPSO，除了可有效平衡算法的探索與開發能力外，尚能透過對非凌越解的群集，獲得多樣性高且數量合理的解集合，以有效引領粒子們的下一次飛行。本研究除了驗證混合式微粒群演算法的有效性外，也提供管理者在選擇存貨控制模式的依據。綜合前述內容所得之結論如下：本研究採用 HMOPSO 進行求解，微粒群演算法在過去已應用在許多研究中，且都有相當不錯的成果，研究中除採用粒子群演算法的原則進行演算外，並將演算法加入了區域搜尋以及群集機制，更強化演算法的績效。在三目標下 HMOPSO 與其他求解方法比較結果，發現在銷售損失的狀況下，HMOPSO 的覆蓋率與最大散佈距離是顯著的優於 SPEA，間距則因為 SPEA 的變異性較大，所以在檢定時兩演算法並沒有顯著差異，因此我們認為 HMOPSO 在求解本研究的存貨控制問題時，其績效衡量均優於 SPEA。其次，利用 HMOPSO 分別求解 N 模式與 B 模式的非凌越解，在覆蓋率上的表現，B 模式都優於 N 模式，B 模式所求得的解較 N 模式接近柏拉圖前緣；而間距則無顯著差異；最大散佈距離則是以 N 模式

表現較佳。兩模型的非凌越解績效衡量指標上互有領先，因此模型的選擇端看決策者的偏好，若決策者希望找到的解可以有較小的攸關成本，較高的服務水準，應挑選缺貨數量模型（B 模式）。

參考文獻

- Agrell, P. J. (1995). A Multicriteria Framework for Inventory Control. *International Journal of Production Economics*, 41(1-3), 59-70.
- Deb, K. (2004). 'Multi-objective Optimization using Evolutionary Algorithms', New York: John Wiley & Sons.
- Harris, F. W. (1913). How Many Parts to Make at Once?. *The Magazine of Management*, 10(2), 135-136.
- Okabe, T., Jin, Y., and Sendhoff, B. (2003). A Critical Survey of Performance Indices for Multi-objective Optimization. *Proc. of 2003 Congress on Evolutionary Computation*, 878-885.
- Shi, Y. H. and Eberhart, R. C. (1998a). A Modified Particle Swarm Optimizer. *IEEE World Congress on Computational Intelligence*, IEEE Press.
- Shi, Y. H. and Eberhart, R. C. (1998b). Parameter Selection in Particle Swarm Optimization. *Evolutionary Programming VII* (Berlin), 591-600.
- Shi, Y. H. and Eberhart, R. C. (1999). Empirical Study of Particle Swarm Optimization. *Proceedings of the Evolutionary Computation 1999 Congress*, 3, 1945-1950.
- Sierra, M. R. and Coello, C. A. (2004). A New Multi-objective Particle Swarm Optimizer with Improved Selection and Diversity Mechanisms. *Proceeding of the 2004 Congress on Evolutionary Computation*, 1-39.

- Silver, E. A., Pyke, D. F., and Peterson, R. (1998). *Inventory Management and Production Planning and Scheduling*, New York: John Wiley & Sons.
- Tersine, R. J. (1994). *Principles of Inventory and Materials Management*, New Jersey: Wheetley Company.
- Veldhuizen, D. A. V. and Lamont, G. B. (2000). Multi-objective Evolutionary Algorithms: Analyzing the State-of-the-art. *Evolutionary Computation*, 8(2), 125-147.
- Zitzler, E. and Thiele, L. (1999). Multi-objective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. *IEEE Transactions on Evolutionary Computation*, 3, 257-271.

Discussion of Decision Problems and Algorithms in Multi-Objective Lost Sales Inventory Models

Chin-Hsiung Hsu^{*} *Ching-Shih Tsou*^{**}

Abstract

Inventory management is an important work to the enterprise. Inventory control involves tradeoff between conflicting objectives such as cost minimization and inventory availability in order to attain the goal of customer satisfaction. So inventory management could be regard as a multi-objective optimization problem (MOP). This work extends Agrell's inventory control problem from backorder to lost sales, and applies hybrid Multi-Objective Particle Swarm Optimization (HMOPSO), which incorporates a local search and clustering method, to an inventory planning problem. The way of multi-objective analysis can determine lot size and safety factor simultaneously under the objectives of minimizing the expected total relevant cost and some measurements about stockout. HMOPSO is compared with Strength Pareto Evolutionary Algorithm (SPEA). The comparative results show that the HMOPSO surpasses the SPEA on the three performance indexes. However, HMOPSO can find lots of non-dominated solution in a single run and traditional approaches just search for one in a single run. Otherwise, in order to avoid the redundance in objective functions, we reorient Agrell's model to two multi-objective inventory control models emerge redundant objective, base on Agrell's objective, we construct two bi-objective inventory models, named the stockout occasions model (N-model) and the number of items stocked out model (B-model). Finally, we find the non-dominated solutions of each model and make some comparisons among these two inventory models.

Keywords: Inventory Management, Multi-Objective Optimization, Particle Swarm Optimization, Lost Sales.

* Associate Professor, Department of Financial Engineering and Actuarial Mathematics, Soochow University, Tel: +886223111531 ext 3626, Fax: +886223812510, E-mail: hsiung@scu.edu.tw

** Professor, Department of Business Administration, National Taipei College of Business, Tel: +886223226459, Fax: +886227942285, E-mail: ctsou@mail.ntcb.edu.tw