利用細菌覓食演算法解決投資組合最佳化問題

高有成
大同大學資訊經營所
ykao@tt.edu.tw

鄭秀姿
大同大學資訊經營所
joyce.cheng69@gmail.com

摘要

在投資組合問題中，除了要考慮收益外，同時須考慮風險因素，所以投資組合問題是屬於一種多目標的問題，當可供選擇的股票標的數量愈多時，問題的複雜度就更高，其所需要的求解時間也相對的更多。因此，本研究希望藉由有效的啟發式演算法在合理的求解時間下，求得效益較佳的投資組合。近年來有學者分別提出以基因演算法、粒子群演算法來求解投資組合最佳化問題，而這類演算法都是在空間中隨機搜尋，其所需要的求解時間相對較長。本文提出的細菌覓食演算法當演算過程中的求解方向不為最佳解方向時，其能隨機轉向另一個方向再繼續搜尋以求得最佳解。本文經由實驗證明，細菌覓食演算法提供了一種較為省時，而且有效率的求解方法。

關鍵詞：細菌覓食演算法、投資組合最佳化、群體智慧演算法。

1. 簡介

近十年台灣實質薪資接近零成長，一般大眾對投資行為趨向主動，期望藉由有效率的投資，獲得薪資以外的收益。然而，因全球不景氣及金融風暴的影響，投資人多以「風險」為優先考量。因為薪資成長有限，導致在希望獲得報酬的同時，對投資的風險承受能力降低。投資人都希望降低風險增加收益。然而高收益必定伴隨高風險，而高風險卻並不一定能獲得高收益。因此，投資者希望藉著分散投資標的來分散其投資風險，而投資者所持有全部投資標的的集合就是投資組合。一般而言，一個優良投資組合的基本條件需具備收益最大化風險極小化這樣的特性，此外尚需具備高流動與低固定交易成本等特點。在利益最大化且風險極小化的前提下，如何配置出最有效率的投資組合，即是本研究著眼的主題。

投資組合理論最早是由美國諾貝爾獎經濟學家 Markkowitz 於 1952 年所提出[4]，在投資組合理論中以"均值"和"變異數"來表示。"均值"指的是單一證券預期收益率的加權平均，加權的權重為該證券的投資比例。"變異數"指的是投資組合收益率的變異。投資組合理論中另一重要觀點為效率前緣(Efficient Frontier)。效率前緣指的是所有最佳化組合的點所集合而成的一條曲線。在該曲線上的任何一個點都代表著一個最佳投資組合，也就是說該曲線上的任何一個點都代表著在各種不同風險承受力下所獲得最大收益的一種組合，或是在各種不同收益率下所承受最小風險的一種組合。

由於投資組合最佳化愈來愈熱門，近年有愈來愈多的學者相繼投入該領域。而啟發式演算法也成為大多數學者的首選。啟發式演算法(heuristic algorithms)目前被廣泛的應用在各種領域之問題。在啟發式演算法中較具代表性的有基因演算法(Genetic Algorithm, GA)、禁忌搜尋法(Tabu Search)、模擬退火法（Simulated Annealing，SA）和群體智慧 (Swarm Intelligence)。群體智慧主要的概念為群體中的個體會以其它個體過往的經驗來做為引導，藉由其它個體的經驗及智慧來發展出最佳的行為。群體智慧演算法中較具代

表性的有粒子群演算法(Particle Swarm Optimization, PSO)、螞蟻演算法(Ant Colony Optimization, ACO)、螢火蟲演算法(Glowworm Swarm Optimization, GSO)、細菌覓食演算法(Bacterial Foraging Optimization, BFO)…等等。

近年來分別有學者提出運用基因演算法及粒子群演算法於投資組合最佳化問題。在基因演算法方面，分別有 Xia 等學者在 2000 年提出運用基因演算法於預期收益下之投資組合最佳化 [10]及 Soleimani 等學者在 2009 提出運用基因演算法解決基數限制及最小交易量問題之投資組合最佳化[5]。在粒子群演算法方面，有 Cura 於 2009 年提出運用粒子群演算法於投資組合最佳化[9]。以下將就基因演算法(GA)及粒子群演算法(PSO)分別介紹如下。

基因演算法(Genetic Algorithm, GA)，是 J. H. Holland 於 1975 所提出的一種演化計算方式[6]。該演算法主要概念是源自達爾文的"物競天擇"原理。GA 的演化方法為將要求解的問題視為一個染色體，該染色體在每個世代中不斷重複的進行複製、交配及突變使其更加優良。複製指的是在一群染色體中挑選適應值較佳的 n 條染色體。交配指的是兩條染色體各自供獻一半的基因產生出一條新的染色體。而在整個演算過程中若只倚靠複製和交配很難產生出完全不同的染色體，演化過程也可能因此陷入區域最佳解。因此，在該演算過程中設計了突變這項操作，突變可以有機會帶領染色體跳脫區域最佳解。

粒子群演算法 (Particle Swarm Optimization, PSO)，是 J. Kennedy 和 R. Eberhart 於 1995 年發展出的一種演化計算方式[7]。PSO 是一種以群體為基礎的最佳化演化方法。其主要概念為，群體中的每一隻鳥以自己過往的經驗及鳥群中其它鳥的經驗為引導，使其飛往較佳的地方。在 PSO 中，每一隻鳥代表著一顆粒子，每顆粒子皆有自己的適應值及自己的飛行速度和位置，每顆粒子會參考自己過去的飛行經驗及粒子群中最佳的飛行經驗，做為其下次飛行的參考。粒子群演算法和基因演算法相類似，唯粒子群演算法所提供的記憶性是基因演算法所缺乏的。

2.數學模式

本研究使用的目標函式及限制式參考自[9]。目標函式為求得投資組合在各種不同風險厭惡因子下的總風險減去總收益，其值愈小愈好。

符號說明：

λ	：	風險厭惡因子
N	：	可供選擇的股票數量
Z_i	：	是否持有股票 i（決策變數）
X_i	：	股票 i 的投資比例（決策變數）
σ_{ij}	：	同時持有股票 i 及股票 j 的風險
μ_i	：	投資股票 i 的收益
κ	：	投資組合中的股票總數量
ε_i	：	股票 i 的投資比例下限
δ_i	：	股票 i 的投資比例上限

目標函式：

$$\min \lambda\left[\sum_{i=1}^{N}\sum_{j=1}^{N} Z_i X_i Z_j X_j \sigma_{ij}\right] - (1-\lambda)\left[\sum_{i=1}^{N} Z_i X_i \mu_i\right] \qquad (1)$$

限制式：

$$\sum_{i=1}^{N} X_i = 1 \qquad (2)$$

$$0 \le X_i \le 1,\ i = 1, \ldots, N \qquad (3)$$

$$\sum_{i=1}^{N} Z_i = K \qquad (4)$$

$$\varepsilon_i Z_i \le X_i \le \delta_i Z_i,\ i = 1, \ldots, N \qquad (5)$$

$$Z_i \in \{0, 1\}\ i = 1, \ldots, N \qquad (6)$$

限制式(2)為所持有股票的投資比例總和必須為 1。限制式(3)為所持有股票的投資比例必須介於 0 與 1 之間。限制式(4)為所持有股票總數量必須等於投資組合中的股票總數量。限制式(5)為所持有股票的投資比例必須介於該股票的投資比例上下限之間。限制式(6)為股票的持有狀態，其值必定為 0 或 1。若要持有該股票，其值為 1；否則，其值為 0。

3.細菌覓食演算法

細菌覓食演算法是 K. M. Passino 於 2002 所提出的一種新形態的演化計算方式[8]。細菌覓食演算法是模擬大腸桿菌在人體腸道覓食的一種演算法。細菌覓食演算法和基因演算法類似，但細菌覓食演算法在演算過程中的求解方向不是最佳解方向時，人工細菌會停止往該方向搜尋，進而隨機轉向其他方向再繼續搜尋。因此，可以大幅的降低求解所需的時間。

細菌覓食演算法透過全域隨機搜尋的過程，找出品質較佳的解。該演算法主要有三項操作：趨化、複製、淘汰。以下分別就此三項操作說明：

1. 趨化（chemotaxis）：

趨化是為了使細菌透過不斷的移動，逐步趨向較好的環境，以求得品質較佳的解。大腸桿菌主要透過兩種運動方式來移動它的位置：翻轉和游泳。大腸桿菌依賴其表面的鞭毛不斷的擺動來達成翻轉和游泳，當鞭毛進行順時針方向的擺動時，大腸桿菌會在原地做翻轉，當鞭毛進行逆時針方向擺動時，大腸桿菌會往前游動。大腸桿菌通常是進行一次隨機翻轉後，往有利的方向連續游動一段距離，再進行一次翻轉然後再連續游動。當大腸桿菌在比較差的環境裡它會比較頻繁的翻轉，試圖尋求較好的方向；反之，當大腸桿菌在比較好的環境裡它會比較頻繁的游動，減少翻轉的次數。在大腸桿菌的生命週期中，大腸桿菌會持續不斷的進行這兩項運動，而這種現象就稱為趨化。

2. 複製(reproduction)：

複製是為了使搜尋能力較強的細菌取代搜尋能力較弱的細菌。生物演化的過程中存在著"適者生存，不適者淘汰"這樣的現象。大腸桿菌在經過一段時間的搜尋食物後，會將搜尋食物能力較強的細菌進行複製以取代搜尋能力較弱的細菌。在該演算法中，將要複製及死亡的細菌數量各設為 $S_r = S/2$，操作的方式為，依照每隻細菌的適應值進行排序，令擁有較差適應值的 S_r 隻細菌死亡，接著再將擁有較好適應值的 S_r 隻細菌複製成和自己完全相同的細菌，以維持細菌的總個數不變。這種行為和基因演算法中的天擇相類似，它使得擁有較好適應值的細菌得以留存至下一世代。

3. 淘汰(elimination)：

在整個演算過程中單憑趨化及複製難以產生出全新的解，淘汰將有機會產生出完全不同的新解。在細菌覓食演算法中，這種淘汰行為只會發生在某些機率(P_{ed})下。當細菌符合淘汰的條件時，會令該隻細菌死亡，同時隨機產生出一隻新的細菌。這種行為很類似基因演算法中的突變，它有機會帶領細菌跳離區域最佳解。

3.1 解的表達

在本研究中每隻細菌代表著一個投資組合候選解。每隻細菌在搜尋空間的位置分為兩段，分別為股票的投資比例(X 值)及是否持有股票 (Z 值)。位置的維度以 N × 2 的方式顯示，N 代表可供選擇的股票數量，如表 1 所示。

表 1 解的表達

	股票 1	股票 2	…	股票 N
X	X_1	X_2		X_N
Z	Z_1	Z_2		Z_N

3.2 演算流程

細菌在演化的過程中必須不斷的移動，逐步趨向較好的環境，以求得品質較佳的解。細菌演化的公式介紹如下：

$$NX_i^{P+1} = X_i^P + \Delta D * C(i) \quad (7)$$

$$NZ_i^{P+1} = Z_i^P + \Delta D * C(i) \quad (8)$$

$$X_i^{P+1} = \begin{cases} NX_i^{P+1} & \text{如果適應值較佳,} \\ X_i^P & \text{否則,} \end{cases} \quad (9)$$

$$Z_i^{P+1} = \begin{cases} NZ_i^{P+1} & \text{如果適應值較佳,} \\ Z_i^P & \text{否則,} \end{cases} \quad (10)$$

符號說明：

NX_i^{P+1} ： 細菌在第 P+1 次趨化時，股票 i 的投資比例之新值

X_i^P ： 細菌在第 P 次趨化時，股票 i 的投資比例之值

NZ_i^{P+1}	:	細菌在第 P+1 次趨化時，持有股票 i 之新狀態
Z_i^P	:	細菌在第 P 次趨化時，持有股票 i 之狀態
P	:	第幾次趨化操作 P=1,…,($N_c \times N_s$)
ΔD	:	細菌翻轉的方向，ΔD ∈ [−1, 1]
C(i)	:	細菌每步游動的長度

公式(7)為細菌在第 P+1 次趨化時，股票 i 新的投資比例值(細菌的新位置)。公式(8)為細菌在第 P+1 次趨化時，持有股票 i 之新狀態。公式(9)為決定細菌在第 P+1 次趨化時，股票 i 的投資比例值是否更新。公式(10)為決定細菌在第 P+1 次趨化時，持有股票 i 的狀態是否改變。

演算流程如下：

Step1：決定參數(風險厭惡因子的變化量(Δλ)、細菌數量(S)、趨化次數(N_c)、游泳次數(N_s)、細菌每步游動的長度(C(i))、複製次數(N_{re})、淘汰次數(N_{ed})、淘汰率(P_{ed})、投資組合中的股票總數量(κ)、每支股票投資比例下限(ε_i)、每支股票投資比例上限(δ_i))。

Step2：初始化一群細菌，隨機產生其所在位置，並依據公式(1)計算每隻細菌的適應值。

Step3：趨化操作

每隻細菌依序對每支股票重複趨化操作達到趨化次數(N_c)為止。每次趨化操作包括隨機翻轉一次，接著游動數次，敘述如下。

（a）細菌隨機翻轉一次(決定 ΔD 的隨機值)來改變移動方向，並依據公式(7)及公式(8)求得細菌所在的新位置。

（b）判斷此新位置是否符合限制式(2)至限制式(6)，若不符合則以修補法進行修補，再依據公式(1)求得適應值。修補法將於 3.3 小節介紹。

（c）依據公式(9)及公式(10)決定細菌是否移動到新的位置。

（d）依據(b)所求得的適應值，判斷翻轉後位置是否較佳。當翻轉後位置較佳時，細菌會循新方向向前游動數次，每次游動的距離為 C(i)，直到適應值不再更好或是達到游泳次數(N_s)為止。否則，細菌會再次重新翻轉，尋找較佳新方向。

Step4：複製操作

（a）令擁有較差適應值的一半細菌死亡，同時令擁有較好適應值的另一半細菌複製一隻和自己完全相同的細菌。

Step5：重複趨化操作及複製操作，直到達到複製次數(N_{re})為止。

Step6：淘汰操作

（a）因為每隻細菌所在的位置以 N×2 個維度來表達，所以對每隻細菌產生 N 個隨機亂數，並判斷該 N 個隨機亂數是否小於淘汰率(P_{ed})。

（b）令隨機亂數小於淘汰率這個維度的 Z 值為 0。同時，為了滿足限制式(2)至限制式(6)，會依 3.3 小節方法進行修補，然後依據公式(1)重新求得每隻細菌的適應值。若淘汰操作後之適應值較佳，則以淘汰後的位置取代原本的位置。

Step7：重複趨化操作、複製操作及淘汰操作，直到達到淘汰次數(N_{ed})為止。

細菌覓食演算法的演算流程圖如圖 1 所示。

3.3 不合理解處理

由於細菌不斷的在演化，所以在演化過程中其產生出的解可能會違反限制式(2)至限制式(6)，此時，需要針對不合理解進行修補。修補的方式有三種：第一種是修補股票持有狀態(限制式(6))，為第二種為投資組合股票總數量固定的修補（限制式(4)），第三種為股票投資比例的修補（限

制式(2)、限制式(3)及限制式(5))。

第一種是修補股票持有狀態(Z 值)：

細菌經過趨化操作後，對每支股票的持有狀態(Z 值)不會剛好等於 0 或 1，因此需要對其做修補，以符合限制式(6)。若 Z 值 ≥ 1 則為要持有該股票，將 Z 值設為 1。若 Z 值 < 1 則為不持有該股票，將 Z 值設為 0。

第二種是修補股票持有數(K*)：

細菌經過趨化和淘汰操作後，對每支股票的持有狀態(Z 值)會改變，因此每隻細菌持有的股票數量也會跟著改變。為了維持目前持有股票數量和投資組合中的股票總數量相等（限制式(4)），對每隻細菌，令目前持有股票數量為 K*，與投資組合中的股票總數量 κ 相比較，有三種狀況。其方法如下：

1. K* < κ ：

   首先，取一亂數，並判斷是否小於 0.5。當亂數小於 0.5 時，隨機選擇一支尚未被選取的股票，同時令該股票的 Z 值等於 1；當亂數大於等於 0.5 時，從尚未被選取的股票中選擇一支 C 值最大的股票，同時令該股票的 Z 值等於 1。C 值是一種報酬對風險比，公式見[9]，其值介於 0 與 1 之間，值越大越好。然後，將目前持有股票數量加 1。重複以上步驟，直到目前持有股票數量和投資組合中的股票總數量相等為止。

2. K* > κ ：

   首先，取一亂數，並判斷是否小於 0.5。當亂數小於 0.5 時，隨機刪除一支已經被選取的股票，同時令該股票的 Z 值等於 0；當亂數大於等於 0.5 時，從已經被選取的股票中刪除一支 C 值最小的股票，同時令該股票的 Z 值等於 0。然後，將目前持有股票數量減 1。重複以上步驟，直到目前持有股票數量和投資組合中的股票總數量相等為止。

3. K* = κ ：符合限制式(4)，不用處理。

第三種是修補股票持有比例(X 值)：

細菌經過趨化操作、淘汰操作及第二種不合理解修補後，會造成股票持有狀態(Z 值)及投資比例(X 值)的改變，為了維持股票的投資比例介於上下限之間（限制式(3)及限制式(5)）及目前持有股票的投資比例總和為 1（限制式(2)），有兩種狀況，其方法如下：

1. 當要持有該股票(Z_i = 1)：

   為滿足限制式(5)，判斷該股票的投資比例(X 值)是否小於投資比例下限（ε），當小於投資比例下限時，將該股票投資比例設為投資比例下限。否則，判斷該股票的投資比例(X 值)是否大於投資比例上限（δ），當大於投資比例上限時，將該股票投資比例設為投資比例上限。最後，將所有要持有股票的投資比例值加總，然後針對這些股票將投資比例做正規化，以符合限制式(2)。

2. 當不持有該股票(Z_i = 0)：

   令該股票的投資比例(X 值)為 0。

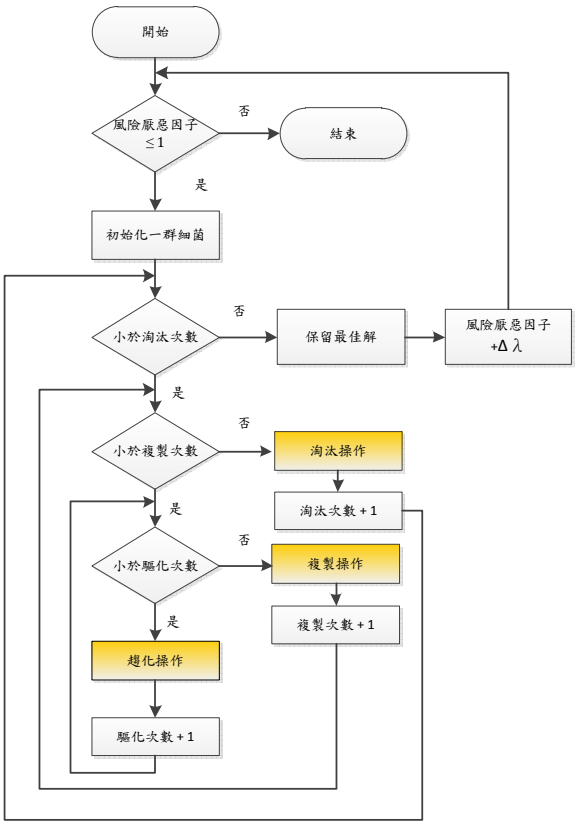

圖 1　細菌覓食演算法流程圖

4. 範例說明

在本範例中，假設我們欲從五支股票中挑選三支股票來做投資。在演算過程中假設共有兩隻細菌，每隻細菌每步游動的長度(C(i))為 0.1；淘汰率(P_{ed})為 0.25，風險厭惡因子(λ)為 0.02，每支股票的投資比例下限(ε_i)為 0.01，上限(δ_i)為 1。

表 2 為每支股票的 C 值及收益(μ)，表 3 為每支股票的持有風險，表 4 為第一次隨機產生的初始解。

4.1 細菌翻轉

假設細菌 a 在五支股票中的翻轉方向(ΔD)分別為 0.2、0.3、0.1、-0.1 及-0.5。細菌翻轉後其相對應的 X 值及 Z 值會跟著改變，例如：細菌 a 的股票 1，其 X 值原本為 0.2，因為其翻轉方向為 0.2，所以翻轉後以公式(7)求得其 X 值為 0.22，以公式(8)求得其 Z 值為 1.02。其它股票的 X 值及 Z 值以此類推。如表 5 所示。

1. 不合理解處理

由表 5 顯示，細菌 a 在翻轉後，其五支股票的 Z 值分別為 1.02、0.03、1.01、0.99 及-0.05。為滿足限制式(6)，將根據 3.3 小節的第一種修補方法，將 Z 值 ≥1 的股票，Z 值改為 1。Z 值 <1 的股票，Z 值改為 0。以細菌 a 來說，只有股票 1 及股票 3 的 Z 值 ≥1，所以將股票 1 及股票 3 的 Z 值改為 1，其餘 3 支股票的 Z 值改為 0。細菌 b 也以相同方式處理。

細菌 a 目前持有的股票數量為兩股(股票 1 及股票 3，K* = 2)，低於投資組合中的股票總數量(κ = 3)。由於細菌 a 違反限制式(4)($K^* \neq \kappa$)，所以根據 3.3 小節的第二種修補方法補選一股，假設採用 C 值最大方法來補選，選中股票 5。而細菌 b 目前持有的股票數量為兩股(股票 2 及股票 3)，由於細菌 b 違反限制式(4)，所以根據 3.3 小節的第二種修補方法補選一股，假設採用隨機方法來補選，選中股票 4。

細菌 a 目前持有的股票為股票 1、股票 3 及股票 5。因為股票 2 及股票 4 沒有選中，所以其投資比例更改為 0。而股票 5 的投資比例低於下限，所以將投資比例設為下限。然後針對選中的三支股票做投資比例總和等於 1 的處理(限制式(2))，處理後其投資比例分別為 0.41、0.57、0.02。最後，細菌 a 以公式(1)重新計算適應值為 -0.28489。細菌 b 也以相同方式處理，表 6 列出細菌翻轉後不合理解處理結果。

2. 判斷是否取代原解

根據公式(9)及公式(10)分別判斷細菌 a 及細菌 b 是否需要改變所在位置。由表 6 顯示，細菌 a 其翻轉後適應值比翻轉前的適應值好(適應值愈小愈好)，所以取代原解。細菌 b 翻轉後適應值比翻轉前的適應值差，所以不取代原解。細菌翻轉後判斷是否取代原解處理結果如表 7 所示。

表 2　每支股票的 C 值及收益

股票	1	2	3	4	5
C 值	0.1	0.2	0.3	0.4	0.5
收益	0.3	0.25	0.3	0.2	0.4

表 3　每支股票的持有風險

σ_{ij}		j				
		1	2	3	4	5
i	1	1	0.1	0.1	0.3	0.4
	2	0.1	1	0.1	0.2	0.5
	3	0.1	0.1	1	0.6	0.3
	4	0.3	0.2	0.6	1	0.5
	5	0.4	0.5	0.3	0.5	1

表 4　第一次隨機產生的初始解

細菌	項目	股票				
		1	2	3	4	5
a	X	0.2	0	0.3	0.5	0
	Z	1	0	1	1	0
	適應值	-0.23236				
b	X	0.2	0.6	0.2	0	0
	Z	1	1	1	0	0
	適應值	-0.25468				

表 5　細菌翻轉後結果

細菌	項目	股票				
		1	2	3	4	5
a	X	0.22	0.03	0.31	0.49	-0.05
	Z	1.02	0.03	1.01	0.99	-0.05
	ΔD	0.2	0.3	0.1	-0.1	-0.5
b	X	0.18	0.62	0.23	-0.01	0.03
	Z	0.98	1.02	1.03	-0.01	0.03
	ΔD	-0.2	0.2	0.3	-0.1	0.3

表 6　細菌翻轉後不合理解處理結果

細菌	項目	股票				
		1	2	3	4	5
a	X	0.41	0	0.57	0	0.02
	Z	1	0	1	0	1
	適應值	-0.28489				
b	X	0	0.72	0.27	0.01	0
	Z	0	1	1	1	0
	適應值	-0.24501				

表 7　細菌翻轉後判斷是否取代原解處理結果

細菌	項目	股票				
		1	2	3	4	5
a	X	0.41	0	0.57	0	0.02
	Z	1	0	1	0	1
	適應值	-0.28489				
b	X	0.2	0.6	0.2	0	0
	Z	1	1	1	0	0
	適應值	-0.25468				

4.2 細菌游泳

細菌 a 翻轉後適應值比翻轉前的適應值好，所以維持原方向向前游動(ΔD 不變)。細菌 b 翻轉後適應值比翻轉前的適應值差，所以將重新翻轉選擇新的方向後再向前游動(重新決定 ΔD)。細菌 b 重新選擇方向後為 0.5、0.3、0.6、0.1、0.2。細菌向前游動時，以公式(7)及公式(8)重新計算 X 值及 Z 值。細菌游泳一次後結果如表 8 所示。

1. 不合理解處理

由表 8 顯示，細菌 a 在游泳後，其五支股票的 Z 值分別為 1.02、0.03、1.01、-0.01 及 0.95。為滿足限制式(6)，將根據 3.3 小節的第一種修補方法，將股票 1 及股票 3 的 Z 值改為 1，其餘 3 支股票的 Z 值改為 0。細菌 b 也以相同方式處理。

細菌 a 目前持有的股票數量為兩股，由於細菌 a 違反限制式(4)，所以根據 3.3 小節的第二種修補方法補選一股，假設選中股票 2。而細菌 b 目前持有的股票數量為三股，所以不需要進行修補。

細菌 a 目前持有的三支股票的投資比例皆介於上下限之間，所以只需對其做投資比例總和等於 1 的處理(限制式(2))。最後，細菌 a 以公式(1)重新計算適應值為 -0.28184。細菌 b 也以相同方式處理。細菌游泳一次後不合理解處理結果如表 9 所示。

2. 判斷是否取代原解

根據公式(9)及公式(10)分別判斷細菌 a 及細菌 b 是否需要改變所在位置。細菌游泳一次後判斷是否取代原解處理結果如表 10 所示。

4.3 細菌複製

因為總共只有兩隻細菌，由表 10 顯示，細菌 b 的適應值較差，所以令細菌 b 死亡，將細菌 a 複製一隻和自己完全相同的細菌。細菌複製結果如表 11 所示。

4.4 細菌淘汰

細菌 a 與 b 依據 3.2 節的 step 6 進行淘汰操作，並進行以下處理。

1. 不合理解處理

假設細菌 a 在淘汰後，只選中股票 3 及股票 5。由於細菌 a 違反限制式(4)，所以根據 3.3 小節的第二種修補方法補選一股，假設選中股票 2。接著，針對選中的三支股票，根據 3.3 小節的第三種修補方法對投資比例進行修補。最後，細菌 a 以公式(1)重新計算適應值為-0.27745。假設細菌 b 沒有做任何淘汰的動作，所以細菌 b 的持股狀態及投資比例不需改變。細菌淘汰後不合理解處理結果如表 12 所示。

2. 判斷是否取代原解

根據公式(9)及公式(10)分別判斷細菌 a 及細菌 b 是否需要改變所在位置。細菌淘汰後判斷是否取代原解處理結果如表 13 所示。

表 8　細菌游泳一次結果

細菌	項目	股票				
		1	2	3	4	5
a	X	0.43	0.03	0.58	-0.01	-0.03
	Z	1.02	0.03	1.01	-0.01	0.95
	ΔD	0.2	0.3	0.1	-0.1	-0.5
b	X	0.25	0.63	0.26	0.01	0.02
	Z	1.05	1.03	1.06	0.01	0.02
	ΔD	0.5	0.3	0.6	0.1	0.2

表 9　細菌游泳一次後不合理解處理結果

細菌	項目	股票				
		1	2	3	4	5
a	X	0.41	0.03	0.56	0	0
	Z	1	1	1	0	0
	適應值	-0.28184				
b	X	0.22	0.55	0.23	0	0
	Z	1	1	1	0	0
	適應值	-0.25778				

表 10　細菌游泳一次後判斷是否取代原解

細菌	項目	股票				
		1	2	3	4	5
a	X	0.41	0	0.57	0	0.02
	Z	1	0	1	0	1
	適應值	-0.28489				
b	X	0.22	0.55	0.23	0	0
	Z	1	1	1	0	0
	適應值	-0.25778				

表 11　細菌複製後結果

細菌	項目	股票				
		1	2	3	4	5
a	X	0.41	0	0.57	0	0.02
	Z	1	0	1	0	1
	適應值	-0.28489				
b	X	0.41	0	0.57	0	0.02
	Z	1	0	1	0	1
	適應值	-0.28489				

表 12　細菌淘汰後不合理解處理結果

細菌	項目	股票				
		1	2	3	4	5
a	X	0	0.02	0.95	0	0.03
	Z	0	1	1	0	1
	適應值	-0.27745				
b	X	0.41	0	0.57	0	0.02
	Z	1	0	1	0	1
	適應值	-0.28489				

表 13　細菌淘汰後判斷是否取代原解處理結果

細菌	項目	股票				
		1	2	3	4	5
a	X	0.41	0	0.57	0	0.02
	Z	1	0	1	0	1
	適應值	-0.28489				
b	X	0.41	0	0.57	0	0.02
	Z	1	0	1	0	1
	適應值	-0.28489				

5. 實驗

本節將利用本研究所提出的細菌覓食演算法進行實驗。本演算法程式以 JAVA 撰寫，執行作業環境為 Intel(R) CPU 1.80 GHz 1.0 GB RAM，作業系統為 Windows XP SP3。本研究的實驗資料來自 http://people.brunel.ac.uk/~mastjjb/jeb/orlib/portinfo.html，其資料內容為：香港 Hang Seng、德國 DAX 100，英國 FTSE 100、美國 S&P100、日本 Nikkei，自 1992/03 ~ 1997/09 的每週價格。實驗共分為兩階段，第一階段為證明本實驗所求得的最佳解為有效合理解，會將本實驗所求得的最佳解和題庫標準解，分別繪畫成兩條效率前緣曲線做比較。第二階段將本實驗所求得的最佳解，分別以 5.2 小節的衡量指標和[9]的實驗數據進行比較。

5.1 參數設定

本實驗使用的參數資料，是經由廣泛的實驗測試所得到的結果。各參數值如下所述：

S	：	細菌數(20)
C(i)	：	每步游動的長度(0.1)
N_c	：	趨化次數(50)
N_s	：	游泳次數(4)
N_{re}	：	複製次數(4)
N_{ed}	：	淘汰次數(2)
P_{ed}	：	淘汰率(0.25)
N	：	可供選擇的股票數量(31)
λ	：	風險厭惡因子(0 ~ 1) 每次增加 0.02，共 51 種
κ	：	投資組合中的股票總數量(10)
ε_i	：	股票 i 的投資比例下限(0.01)
δ_i	：	股票 i 的投資比例上限(1)

5.2 衡量指標

本實驗所使用的衡量指標分別為 Mean Euclidian Distance（MED，平均歐幾里得距離）、Variance of Return Error（VRE，風險誤差）、Mean Return Error

（MRE，收益誤差）及 TIME(執行時間)。前三項衡量指標在[3]及[9]的實驗中也被採用。MED 為本演算法所求得的效率前緣曲線上的各點和題庫標準效率前綠曲線上的各點的距離總合之平均值，其值愈小愈好。VRE 為本演算法所求得的效率前緣曲線上的各點和題庫標準效率前綠曲線上的各點的風險誤差總合之平均值，其值愈小愈好。MRE 為本演算法所求得的效率前緣曲線上的各點和題庫標準效率前綠曲線上的各點的收益誤差總合之平均值，其值愈小愈好。各衡量指標的公式如下所示：

Mean Euclidian Distance (MED)

$$\left(\sum_{i=1}^{\xi}\sqrt{(v_{ij}^{s}-v_{j}^{h})^{2}-(r_{ij}^{s}-r_{j}^{h})^{2}}\right)/\xi \qquad (11)$$

Variance of Return Error(VRE)

$$\left(\sum_{i=1}^{\xi}100\left|v_{ij}^{s}-v_{j}^{h}\right|/v_{j}^{h}\right)/\frac{1}{\xi} \qquad (12)$$

Mean Return Error (MRE)

$$\left(\sum_{i=1}^{\xi}100\left|r_{ij}^{s}-r_{j}^{h}\right|/r_{j}^{h}\right)/\frac{1}{\xi} \qquad (13)$$

符號說明：

v_j^h ： 本演算法所求得的最佳解(風險值)

r_j^h ： 本演算法所求得的最佳解(收益值)

v_i^s ： 題庫提供的最佳解(風險值)

r_i^s ： 題庫提供的最佳解(收益值)

v_{ij}^s ： 題庫的解（v_i^s）之中，與 v_j^h 最接近的解

r_{ij}^s ： 題庫的解（r_i^s）之中，與 r_j^h 最接近的解

5.3 實驗結果

第一階段的實驗是以細菌覓食演算法針對香港 Hang Seng 這個例題，以 51 種不同的風險厭惡因子所執行的結果，再和題庫標準解分別繪畫成兩條效率前緣曲線做比較。由圖 2 可以看出，細菌覓食演算法所求得的最佳解和題庫標準解是相當的接近，整條效率前緣曲線幾乎是重疊的，由此可以證明本演算法所求得的最佳解確實是有效合理解。

第二階段的實驗是以細菌覓食演算法針對香港 Hang Seng 這個例題，運用 5.2 小節的衡量指標(公式(11)至公式(13))，以 51 種不同的風險厭惡因子所執行的結果和粒子群演算法實驗結果之比較，粒子群演算法之實驗數據來自[9]。由表 14 可以看出，細菌覓食演算法其求解的品質相較於粒子群演算法是較佳的，而在求解時間上也比粒子群演算法更加的快速，節省的時間接近於一倍。所以整體而言，細菌覓食演算法是比粒子群演算法來的有效率。

6. 結論

本研究所提出的運用細菌覓食演算法解決投資組合最佳化問題，經由上述實驗結果顯示，該演算法所求得的最佳解相較於粒子群演算法所求得的最佳解其品質更佳，而在求解時間上其表現也較粒子群演算法更快速，尤其當求解例題的可供選擇的股票數量愈大時，其時間的差距則愈加明顯。

未來研究方向，將針對細菌覓食演算法參數太過複雜及缺乏群體經驗的方向加以改善。

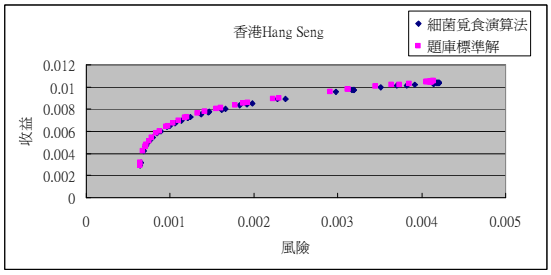

圖 2　香港 Hang Seng 執行結果

表 14　香港 Hang Seng 執行結果

	粒子演算法	細菌覓食演算法
VRE	2.2421	1.971195
MRE	0.7427	0.652264
MED	0.0049	0.000085
TIME	34	18

參考文獻

[1]林書羽，“應用變動鄰域搜尋法於投資組合最佳化問題之研究”，*元智大學研究所碩士論文*，2008。

[2]張瑜庭，“應用和弦演算法於投資組合最佳化問題之研究”，*元智大學研究所碩士論文*，2010。

[3]A. Fernandez, S. Gomez, "Portfolio selection using neural networks," *Computers & Operations Research*, Vol. 34, Issue 4, April 2007, pp. 1177-1191.

[4]H. Markowitz, "Portfolio selection," *Journal*, Vol. 7, No.1, Mar. 1952, pp. 77-91.

[5]H. Soleimani, H. R. Golmakani, M. H. Salimi, "Markowitz-based portfolio selection with minimum transaction lots, cardinality constraints and regarding sector capitalization using genetic algorithm," *Expert Systems with Applications*, Vol. 36, Issue 3, Part 1, April 2009, pp. 5058-5063.

[6]J. H. Holland, "Adaptation in natural and artificial systems," *University of Michigan Press*, 1975.

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

[8]K. M. Passino, "Biomimicry of bacterial foraging for distributed optimization and control," *IEEE Journals*, Vol. 22 , Issue 3, 2002, pp. 52-67.

[9]T. Cura, "Particles warm optimization approach to portfolio optimization," *Nonlinear Analysis: Real World Applications*, Vol. 10, Issue 4, Aug 2009, pp. 2396-2406.

[10] Y. Xia, B. Liu, S. Wang & K. K. Lai, "A model for portfolio selection with order of expected returns," *Computers & Operations Research*, Vol. 27, Issue 5, April 2000, pp. 409-422.