

李維平、李元傑、謝明勳（2014），『以群中心策略改良人工蜂群演算法』，
資訊管理學報，第二十一卷，第一期，25-44。

以群中心策略改良人工蜂群演算法

李維平*

中原大學資訊管理學系

李元傑

中原大學資訊管理學系

謝明勳

中原大學資訊管理學系

摘要

人工蜂群演算法（Artificial Bee Colony）是學者 Karaboga 於 2005 年所提出之最佳化演算法，具有良好的穩定性、優秀的求解能力、控制參數少、計算簡潔及易於實現等優點，但也存在後期過早收斂、開發精度不佳等問題。因此，本研究提出一種新式的群中心改良策略，以改善人工蜂群演算法之搜尋能力。

本研究以常見的六個測試函數進行實驗，從結果得知，本研究提出之群中心策略有效地加強人工蜂群演算法的搜尋能力，使其在演算法後期持續開發而不會過早收斂，在大部分測試函數上都有明顯的改善。

關鍵詞：人工蜂群演算法、最佳化演算法、演化式計算

* 本文通訊作者。電子郵件信箱：wplee@cycu.edu.tw

2012/12/19 投稿；2013/9/22 第一次修訂；2013/9/30 接受

Lee, W.P., Lee, Y.C. and Hsieh, M.H. (2014), 'Enhancing Artificial Bee Colony Algorithm with Centroid Strategy', *Journal of Information Management*, Vol. 21, No. 1, pp. 25-44

Enhancing Artificial Bee Colony Algorithm with Centroid Strategy

Wei-Ping Lee*

Department of Information Management, Chung Yuan Christian University

Yuan-Chieh Lee

Department of Information Management, Chung Yuan Christian University

Ming-Hsun Hsieh

Department of Information Management, Chung Yuan Christian University

Abstract

Artificial Bee Colony algorithm (ABC) is an optimization algorithm proposed by Karaboga in 2005. This method has a good investigation capability, and it is also simple and easy to implement. Though ABC has many advantages, there are still some drawbacks, such as premature convergence and falling into local optimal solutions. In this study, we utilize the centroid strategy to enhance ABC for improving these weak points.

In this research we use 6 benchmark functions to test our method and related researches. The results show that our algorithm can enhance the searching capability of ABC and it is better than the other researches in most of benchmark functions.

Keywords: Artificial Bee Colony Algorithm, Optimization Algorithm, Evolutionary Computation.

* Corresponding author. Email: wplee@cycu.edu.tw
2012/12/19 received; 2013/9/22 1st revised; 2013/9/30 accepted

壹、緒論

由於一些複雜的數學問題的出現，如「難以微分」或「具有多個局部最佳解」等性質的問題，使學者們開始嘗試以隨機性的計算方式來尋求解答，因而促進了隨機搜尋演算法的發展（鄭富升 2005）。這些演算法的目的在於以自我學習（self-learning）的方式，找出一個近似最佳解的可行解（feasible solution），而這些演算法的特性，皆是以「適者生存、不適者淘汰」的方式來逐步改善解的品質，因此又稱為演化式計算（Evolutionary Computation: EC）。

人工蜂群演算法是近年所提出的一種演化式計算，具有「控制參數少」、「易於實現」、「計算簡潔」、「收斂速度快」及「穩定性好」等優點（暴勵 & 曾建潮 2010; 羅鈞 & 樊鵬程 2009）；其優越的求解效能，也經由多個研究證實（Basturk & Karaboga 2006; Karaboga 2005; Karaboga & Akay 2008; Karaboga & Basturk 2007; Karaboga & Basturk 2008），並開始應用於各種工商業領域。

本研究提出一種新式的群中心策略（Centroid Strategy），並配合相對位置策略（Opposition Position Strategy），嘗試改良人工蜂群演算法的求解能力，藉此改善開發精度不高之問題，並增強人工蜂群演算法跳脫局部最佳解之能力。

貳、文獻探討

一、人工蜂群演算法簡介

在真實世界的蜂群中，某些任務交由特定的蜜蜂以有效率的分工執行，增加蜂巢裡的花蜜含量至最大值。人工蜂群演算法是參考 Seeley 的蜂群自我組織模型所提出的最佳化演算法（Karaboga 2005）。在人工蜂群演算法的架構中，包含了「工蜂（employed bee）」、「觀察蜂（onlooker bee）」與「偵察蜂（scout bee）」；其中，工蜂與觀察蜂的數量相同。工蜂負責探索花蜜來源，並提供食物源的品質訊息給於蜂巢等待的觀察蜂。觀察蜂依照工蜂所提供的訊息前往食物源搜索。偵察蜂會在食物源收益度無法改善時，於搜索空間隨機性地探索，藉以求得新的食物源。

原始的人工蜂群演算法（Karaboga 2005）流程如下：

Step1：初始化。

設定初始參數，包括工蜂數量（ SN ），最大迭代數（ MCN ），每一個食物源未改善次數的極限值（ $limit$ ）。

將每隻工蜂隨機放置於解空間中的一個位置（隨機放置的方法請見公式(1)），

此一位置即稱為蜜蜂搜索的一個食物源，食物源位置的適應值即稱為食物源的收益度。

$$x_{ij} = x_{\min}^j + rand[0, 1](x_{\max}^j - x_{\min}^j) \quad (1)$$

x_{ij} 是第 i 隻工蜂第 j 維度的初始值， $x_{\min}^j$ 為搜索空間第 j 維度的最小值， $x_{\max}^j$ 為搜索空間第 j 維度的最大值， $rand[0,1]$ 是 0~1 之間的隨機數。

Step2：每隻工蜂依照公式(2)移動到新食物源位置，並計算食物源收益度。

$$v_{ij} = x_{ij} + rand[-1, 1](x_{ij} - x_{kj}) \quad (2)$$

v_{ij} 是第 i 隻工蜂第 j 維度移動後的新位置， x_{ij} 是第 i 隻工蜂第 j 維度移動前的位置， x_{kj} 為隨機選擇另一工蜂 k 第 j 維度的位置， $rand[-1,1]$ 是 -1 至 1 之間的隨機數。以下示意圖表達二維解空間，工蜂的移動路線。在原始人工蜂群演算法，工蜂只隨機選擇一個維度作移動，但為了幫助讀者易於了解，我們在示意圖中，是假設工蜂在兩個維度同時移動。

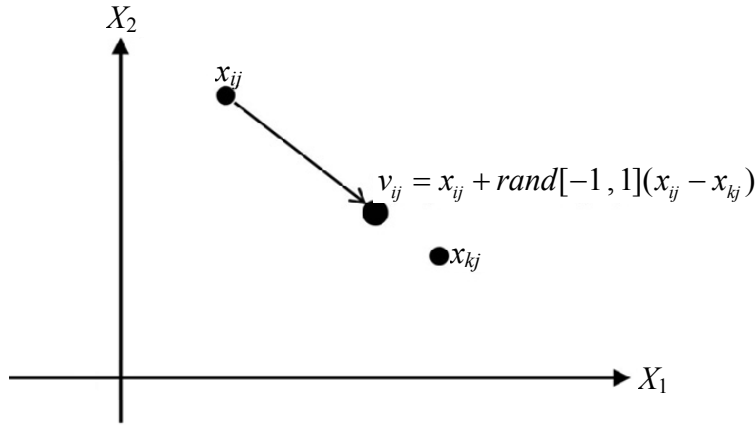


圖 1：工蜂移動方式二維示意圖

Step3：利用輪盤法（見公式(3)），決定每一隻觀察蜂應到那一個食物源去協助搜尋。觀察蜂將依照公式(2)移動到食物源鄰近位置搜尋，並計算食物源收益度。

$$p_i = \frac{fit_i}{\sum_{n=1}^{SN} fit_n} \quad (3)$$

p_i 是第 i 個食物源被選擇的機率， fit_i 為第 i 個食物源的收益度， SN 是食物源的總數。

Step4：當任一食物源的收益度經過 $limit$ 次搜尋後仍沒有改善，則放棄此食物源。

出動一隻偵察蜂，依公式(1)尋找一個新的食物源，來取代被放棄的食物源。

Step5：記憶到目前為止收益度最高的食物源。

記錄目前最佳的食物源收益度作為演算法的最佳解。

Step6：循環直到符合終止條件。

判斷是否符合終止條件（亦即目前迭代數是否為最大迭代數 MCN ）。若符合終止條件，則演算法停止，並輸出最佳解；若尚未符合，則回到 Step2。

人工蜂群演算法流程中，工蜂與觀察蜂負責區域開發，偵察蜂則負責全域探索。區域開發的方式為移動到該食物源的鄰近點（公式(2)），全域探索則是偵察蜂以新的位置取代舊食物源位置（公式(1)）。

二、人工蜂群演算法相關研究

人工蜂群演算法自從 2005 年由 Karaboga 提出後，引起許多學者重視，至今已提出多種改良的研究，以提升演算法的效能。

相關研究中，Li 等（2007）嘗試控制局部最佳解的判斷機制，以提高人工蜂群演算法在全域搜索的效能。Feng 與 Ding（2008）以群組對稱的方式產生初始位置，並以 Boltzmann 選擇機制取代輪盤法，改良人工蜂群演算法。Quan 與 Shi（2008）使用壓縮映射的不動點定理，於 Banach 空間產生新的迭代，藉此提高人工蜂群演算法的收斂速度。Tsai 等（2008）為了使觀察蜂獲得最好的開發能力，以牛頓的萬有引力定理，改良觀察蜂的選擇機制。羅鈞與樊鵬程（2009）以基因演算法的交配策略改良人工蜂群演算法，增加食物源的多樣性，增強解質量與減小落入局部最佳解的機率。暴勵與曾建潮（2010）以動態調整搜索空間改良人工蜂群演算法，逐步縮小搜索區域，並利用混沌變量跳脫局部最佳解。Alatas（2010）以混沌的概念改良人工蜂群演算法，隨機選擇不同的混沌地圖產生新的迭代，藉此跳脫局部最佳解。Narasimhan（2009）以平行處理的概念改良人工蜂群演算法，藉此增加運算速度與解的品質。Akay 與 Karaboga（2012）參考 DE 演算法，加入修改率與比例因子。

學者 Zhu 與 Kwong（2010）提出 Gbest-guided Artificial Bee Colony algorithm（GABC），其基本架構與原始的人工蜂群演算法（Karaboga 2005）非常相似，方法是以目前所找到的最佳解作為導引，來引導工蜂與觀察蜂往此方向進行搜索，此方法雖然簡單，然而效果卻非常顯著；因此，本研究將此文獻作為主要的比較對象。

GABC 演算法的搜索公式是以公式(2)為基礎架構，再輔以當前最佳食物源的引導，如公式(4)。

$$v_{ij} = x_{ij} + rand[-1, 1](x_{ij} - x_{kj}) + rand[0, 1.5](y_j - x_{ij}) \quad (4)$$

v_{ij} 是第 i 隻工蜂第 j 維度移動後的新位置， x_{ij} 是第 i 隻工蜂第 j 維度移動前的位置， x_{kj} 為隨機選擇另一工蜂 k 第 j 維度的位置， $rand[-1, 1]$ 是 -1 至 1 之間的隨機數，是當前最佳食物源的第 j 維度， $rand[0, 1.5]$ 是 0 至 1.5 之間的隨機數。

從上述文獻可知，已有許多學者傾力於改良人工蜂群演算法，其主要的改良方向，大致可分為「初始迭代的產生方式」、「工蜂與觀察蜂的演化機制」、「觀察蜂的選擇機制」等。本研究主要是針對「工蜂與觀察蜂的演化機制」與「偵察蜂的探索機制」進行改良，期能改善人工蜂群演算法後期開發精度不佳的問題。

三、群中心策略

群中心是一個具有發展潛力的位置 (Liu et al. 2007)。位於群中心的位置，普遍具有良好的適應值，有很大的機率是整個群體的全域最佳解。演算法迭代初期，群的中心與全域最佳解分屬於兩個不同的位置，隨著演化迭代數的增加，兩者間的距離會逐漸縮短，直至迭代結束，兩者將會收斂到相近的位置。因此，藉由群中心策略可以引導整個群體移動至具有發展潛力的區域進行搜索，並且加快收斂的速度。Xu 等 (1993) 指出遠離中心點的粒子，適應值從未贏過靠近中心點附近的粒子。Yi 等 (2010) 指出若未適當使用群中心策略於分群演算法，則會使分群演算法容易陷入區域最佳解。

四、相對位置策略

Tizhoosh 提出相對位置的概念 (Tizhoosh 2005)，其原理是在目前位置之相對位置，產生新的位置，在多維空間中，候選解 $(x_1, x_2, \dots, x_n)$ ， n 為解的維度， $(x_1, x_2, \dots, x_n) \in R$ 且 $x_i \in \{a_i, b_i\}$ ， a_i 為解空間中第 i 維的最小值， b_i 為解空間中第 i 維的最大值，相對位置 $(\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_n)$ 如公式(5)。

$$\tilde{x}_i = a_i + b_i - x_i \quad i = 1, 2, \dots, n \quad (5)$$

解空間的範圍裡，除了根據演算法所生成的隨機位置進行搜索外，也會考慮到其相對位置進行搜索 (Tizhoosh 2005)。相對位置與隨機位置兩者間相比較，相對位置有更大的機率接近全域最佳解 (Rahnamayan et al. 2008)。在相關研究上，

Zhang 等人提出對稱學習初始化，及產生一個模範粒子分享資訊給其他粒子改良粒子群演算法，使粒子群演算法加速收斂及跳脫區域最佳解 (Zhang et al. 2008)。Rahnamayan 等人提出對稱學習改善初始化及相對位置跳脫策略，增進收斂速度及避免陷入區域最佳解 (Rahnamayan et al. 2006)。

本研究嘗試用相對位置策略，藉由其避免陷入區域最佳解的特性，以增進人工蜂群演算法的效能。

參、演算法設計

一、群中心策略設計

由於人工蜂群演算法採分散式學習，具有隨機搜尋，因此具有大範圍的探索能力，但缺乏小範圍的開發能力，因此我們引入群中心策略，可加強小範圍的搜尋能力，以改進人工蜂群演算法的弱點。群中心位置計算如下：

$$C = \frac{\sum_{i=1}^{SN} X_i}{SN} \quad (6)$$

C 是群中心的位置， X_i 是第 i 隻工蜂的位置， SN 是食物源之數量。其示意圖如圖 2 所示，星號為搜索空間的中心點， $Old Food_i$ 為最後一隻觀察蜂所在的食物源， $New Food_i$ 為群中心點，最後一隻觀察蜂放棄 $Old Food_i$ 到群中心點 $New Food_i$ 探索。

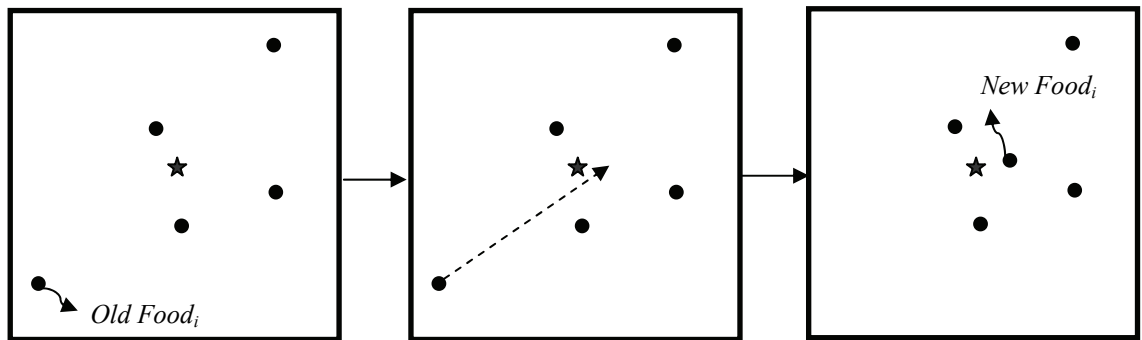


圖 2：群中心策略之示意圖

二、相對位置策略設計

原始人工蜂群演算法之偵察蜂，以隨機方式產生新的位置，雖然能夠有效地跳脫局部最佳解，但並不能有效地探索到好的食物源。為了改善此問題，本研究以適應值最好的食物源之相對位置策略（Opposition）取代人工蜂群演算法之偵察蜂機制，在大部分的函數能夠獲得較好的結果，其示意圖如圖 3 所示，星號為搜索空間的中心點， $Food_{old}$ 為工蜂所在欲放棄之食物源， $Food_{new}$ 為適應值最好的食物源之相對位置，偵查蜂放棄 $Food_{old}$ 到 $Food_{new}$ 探索。公式如(7)，

$$x_j = x_{\min}^j + x_{\max}^j - x_{best}^j, j = 1, 2, \dots, n \quad (7)$$

x_j 是 $Food_{new}$ 第 j 維度的值， $x_{\min}^j$ 為搜索空間第 j 維度的最小值， $x_{\max}^j$ 為搜索空間第 j 維度的最大值， x_{best}^j 為最好食物源第 j 維度的值。

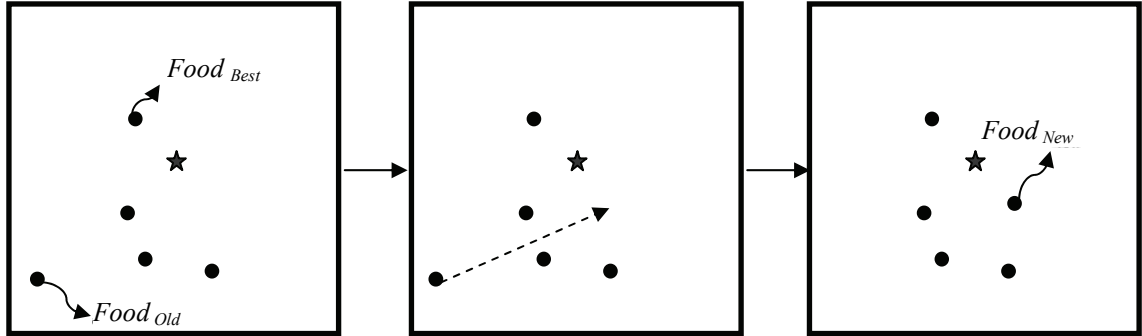


圖 3：相對位置策略之示意圖

三、運用群中心策略改良人工蜂群演算法

本研究將提出之演算法，命名為 CSABC（Centroid-Strategy-based Artificial Bee Colony algorithm），完整的演算法描述如下：

Algorithm CSABC

Step 1：設定初始參數。

Step 2：工蜂根據公式(2)進行鄰近食物源之搜尋。

Step 3：第 1 到 $SN-1$ 隻觀察蜂以輪盤法選擇食物源，並根據公式(2)進行鄰近食物源之搜尋。

Step 4：最後 1 隻觀察蜂到所有食物源的中心位置去探索，若得到更佳的食物源，則其會告知最差食物源的工蜂，放棄現有食物源。

Step 5：偵察蜂以目前最佳解的相對位置進行探索。

Step 6：如到達終止條件（迭代數 = MCN ），則輸出最佳解；如未到達終止條件，則回到 Step 2。

四、本研究結合其他蜂群演算法（CSGABC）

本方法易於與其他人工蜂群演算法作結合，以強化蜂群的搜索能力。目前文獻中 GABC 為一相當優良的改良方法，方法簡單但效果顯著；因此本研究將所提出之 CSABC 演算法與 GABC 演算法相互結合，稱為 CSGABC（Centroid-Strategy and Gbest-guided Artificial Bee Colony algorithm），藉此檢視兩種方法的融合是否具有互補的成效，也用以試驗 CSABC 與他種方法之間結合的可能性。

CSGABC 是以公式(4)作為工蜂與觀察蜂新的搜索方式，更動 CSABC 流程中的 Step 2 與 Step 3，使用全域最佳解導引搜索，提升區域開發能力，完整的 CSGABC 演算法描述如下：

Algorithm CSGABC

Step 1：設定初始參數。

Step 2：工蜂根據公式(4)進行鄰近食物源之搜尋。

Step 3：第 1 到 $SN-1$ 隻觀察蜂以輪盤法選擇食物源，並根據公式(4)進行鄰近食物源之搜尋。

Step 4：最後 1 隻觀察蜂到所有食物源的中心位置去探索，若得到更佳的食物源，則其會告知最差食物源的工蜂，放棄現有食物源。

Step 5：偵察蜂以目前最佳解的相對位置進行探索。

Step 6：如到達終止條件（迭代數 = MCN ），則輸出最佳解；如未到達終止條件，則回到 Step 2。

肆、實驗測試

一、實驗環境

本研究之實驗環境如表 1 所示，所使用的軟體為 MATLAB，因它具有優越的繪圖能力，可以明顯地展現出結果的差異性。

表 1：實驗環境

實驗環境	
作業系統	Microsoft Windows 7
中央處理器	Intel Core i7 920 2.67 GHz
記憶體	12 GB
軟體	MATLAB 7.10.0

二、參數設定

本研究選擇 Zhu 與 Kwong (2010) 的文獻作為主要的比較對象，此篇研究以原本人工蜂群演算法的移動公式混合粒子群演算法為其修改概念，提出一 Gbest-guided Artificial Bee Colony algorithm (簡稱 GABC)，此篇論文是目前我們所能找到的相關文獻中，改良結果最好的蜂群演算法研究。

為了在相同的基礎作公平的比較，我們參考此篇論文的參數設定，如表 2 所示，以相同的實驗環境來測試，以便有公平的比較基礎。

表 2：實驗環境參數設定

參數設定	
蜂群數量	80
Limit	100
評估次數	100
迭代數	5000

三、測試函數

本實驗使用的測試函數如表 3 及圖 4 到圖 9 所示，共有 6 個，這些測試函數為演化式計算領域所常使用的，其評估指標分別為平均值 (Mean) 與標準差 (Standard Deviation)。

每個測試函數自身皆有一個最佳解，依照測試函數的不同，此解可能是函數中的最大值，或是最小值，因此最佳解具有唯一性；如果演算法求得的解，數值越接近最佳解，亦或是能直接搜尋到最佳解，代表演算法在此測試函數的表現越好。而為了檢視演算法的穩健性，一般利用重複多次獨立實驗的方式，計算多次實驗所求得的解其平均值與標準差；透過平均值的檢視，以看出演算法的求解優

越性，透過標準差的檢視，可觀察演算法求解的穩定性，因此，此兩項指標，是相關學術領域常使用的評估方式。

在表 3 中，每個測試函數 $f(x)$ ，其中的 x 為 n 維的向量， $x = (x_1, x_2, x_3, \dots, x_n)$ 。

表 3：測試函數

代號	測試函數	最佳解	搜尋範圍	公式
$f1$	Sphere	0	$[-100, 100]^n$	$f(x) = \sum_{i=1}^n x_i^2$
$f2$	Rosenbrock	0	$[-50, 50]^n$	$f(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$
$f3$	Rastrigin	0	$[-5.12, 5.12]^n$	$f(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$
$f4$	Griewank	0	$[-600.600]^n$	$f(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$
$f5$	Ackley	0	$[-32.768, 32.768]^n$	$f(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e$
$f6$	Schwefel	隨維度變動	$[-500, 500]^n$	$f(x) = \sum_{i=1}^n -x_i \times \sin(\sqrt{ x_i })$

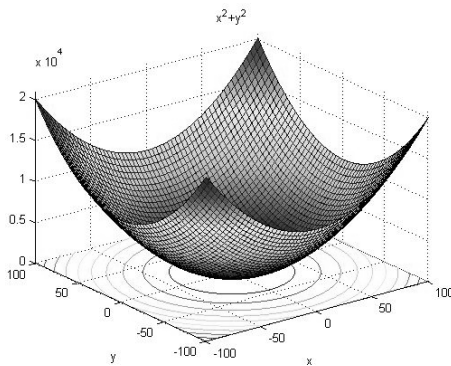


圖 4：Sphere

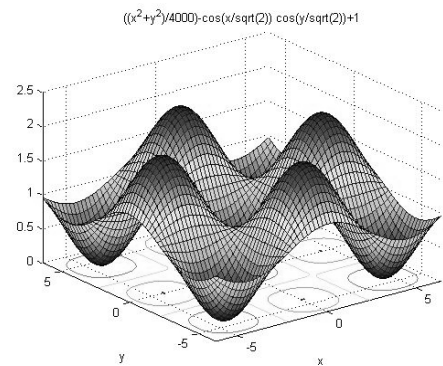


圖 5：Griewank

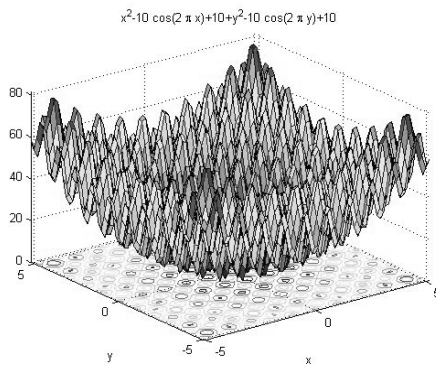


圖 6：Rastrigin

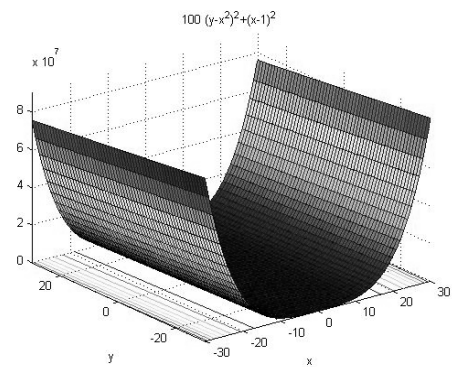


圖 7：Rosenbrock

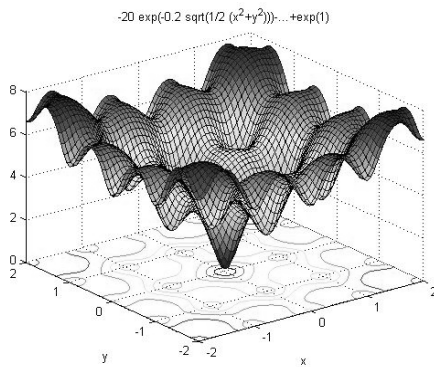


圖 8：Ackley

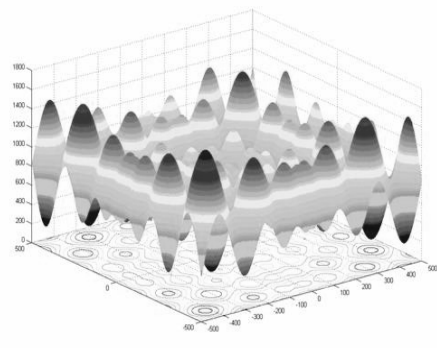


圖 9：Schwefel

四、實驗結果

本研究主要提出之演算法為 CSABC；此外，我們將 CSABC 演算法與 GABC 演算法 (Zhu & Kwong 2010) 作結合，稱為 CSGABC。我們將這兩個演算法與 ABC 和 GABC 進行比較測試。

測試 CSGABC 的目的，要說明我們的方法除了可單獨使用，也可結合其他蜂群演算法，以達到相輔相乘的綜效 (synergy)。從以下的實驗結果，可發現 CSGABC 亦有相當優異的搜尋能力，通常可勝過單獨使用 CSABC 和 GABC。

從表 4 (f1)，可看出 Sphere 函數測試結果，CSABC 與 CSGABC 在開發能力與穩定性上，都較 ABC 與 GABC 還要優越。從圖 10 可看出，在維度 60，ABC 也是在 2600 代左右就已經收斂，GABC 則在 1600 代左右也已停止開發，CSABC 與 CSGABC 皆保有 ABC 與 GABC 之收斂速度，有效地持續開發，而找到更好的解。

Rosenbrock 函數的測試結果 (見表 4f2)，CSABC 的表現並不理想，但是

CSGABC 卻仍然保有很好的優勢，其開發能力與穩定性也較 ABC 與 GABC 還要優越。從圖 11 可看出，在維度 3 時，CSABC 在收斂速度上也是較 ABC 稍快，但是在 300 代左右，開發速度便會開始下降，CSGABC 都能保持其優勢，除了收斂速度較快之外，還能夠穩定地持續開發。

Rastrigin 函數的測試結果（見表 4/f3），在維度 30 時，CSABC 與 CSGABC 在開發能力與穩定性上，都較 ABC 與 GABC 還要優越，且都能夠找到最佳解，而在維度 60 時，CSABC 則較 GABC 稍微弱勢，而 CSGABC 則保持其優勢，且依然能夠找到最佳解。從圖 12 可以看出，在維度 60 時，ABC 在 3500 代左右已呈現停滯狀態，GABC 也是約在 3500 代左右停滯，CSABC 在 3300 代左右也停滯，然而，CSGABC 卻可以有效地持續開發，進而找到最佳解。

Griewank 函數的測試結果（見表 4/f4），CSABC 與 CSGABC 在開發能力與穩定性上，都較 ABC 與 GABC 還要優越，且 CSABC 與 CSGABC 皆能夠找到最佳解。從圖 13 可以看出，在維度 60 時，ABC 在 2800 代左右，就幾乎呈現停滯狀態，而 GABC 在 3700 代左右，也已無法持續開發，CSABC 與 CSGABC 皆能夠保有穩定地開發能力，且都能找到最佳解。

Ackley 函數的測試結果（見表 4/f5），CSABC 與 CSGABC 在開發能力與穩定性上，都較 ABC 與 GABC 還要優越。從圖 14 可以看出，在維度 60 時，ABC 因收斂速度緩慢，在 4500 代左右才呈現停滯狀態，GABC 則在 2700 代左右也已無法繼續開發，CSABC 與 CSGABC 皆能夠保有 ABC 與 GABC 原有之收斂速度，且當 ABC 與 GABC 停止開發時，CSABC 與 CSGABC 皆能夠持續地開發，進而找到更好的結果。

Schwefel 函數的測試結果（見表 4/f6），CSABC 在此函數也並不理想，在維度 30 時，CSGABC 雖與 GABC 的結果很接近，但在維度 60 時，結果卻較 GABC 差，而 CSABC 與 CSGABC 在維度提升後，都較 ABC 較好。從圖 15 可以看出，CSABC 與 CSGABC 雖然能夠保有 ABC 與 GABC 原有之收斂速度，在後期開發能力卻都較 GABC 弱勢。

表 4：測試結果

測試 函數	演算法	ABC		GABC		CSABC		CSGABC	
	維度	Mean	Std	Mean	Std	Mean	Std	Mean	Std
<i>f1</i>	30	6.34859e -016	9.71074e -017	4.16236e -016	6.69959e -017	5.29657e -054	5.17198e -053	5.29420e -068	3.10670e -067
	60	2.21062e -015	3.63010e -016	1.34153e -015	1.45550e -016	3.39865e -020	2.21619e -019	2.81780e -031	1.45646e -030

測試 函數	演算法	ABC		GABC		CSABC		CSGABC	
	維度	Mean	Std	Mean	Std	Mean	Std	Mean	Std
f_2	2	5.02329e -003	6.90834e -003	5.06976e -005	7.94764e -005	2.21660e -002	1.80043e -002	3.05066e -009	4.66937e -009
	3	3.27513e -002	4.00285e -002	4.39925e -004	4.84642e -004	9.87035e -002	4.69132e -002	3.14892e -006	3.93273e -006
f_3	30	9.89075e -014	5.21229e -014	7.38964e -015	1.92129e -014	0.00000e +000	0.00000e +000	0.00000e +000	0.00000e +000
	60	1.23421e -008	1.11427e -007	3.95630e -013	1.65956e -013	4.06999e -013	2.13259e -013	0.00000e +000	0.00000e +000
f_4	30	5.07538e -014	4.94468e -013	5.44009e -017	8.42349e -017	0.00000e +000	0.00000e +000	0.00000e +000	0.00000e +000
	60	1.76553e -013	6.93804e -013	8.20455e -016	4.39267e -016	0.00000e +000	0.00000e +000	0.00000e +000	0.00000e +000
f_5	30	4.95959e -014	6.43295e -015	3.47455e -014	3.40446e -015	4.05009e -015	1.11721e -015	4.08562e -015	1.07118e -015
	60	1.58096e -013	2.51842e -014	1.01998e -013	7.95886e -015	9.24061e -014	1.66033e -014	4.68958e -015	1.15756e -015
f_6	30	-1.25695e +004	6.51531e -007	-1.25695e +004	1.13581e -011	-1.25694e +004	2.96182e -001	-1.25695e +004	1.11847e -011
	60	-2.50286e +004	7.34310e +001	-2.51390e +004	1.16255e -001	-2.50344e +004	7.28883e +001	-2.51354e +004	2.03058e +001

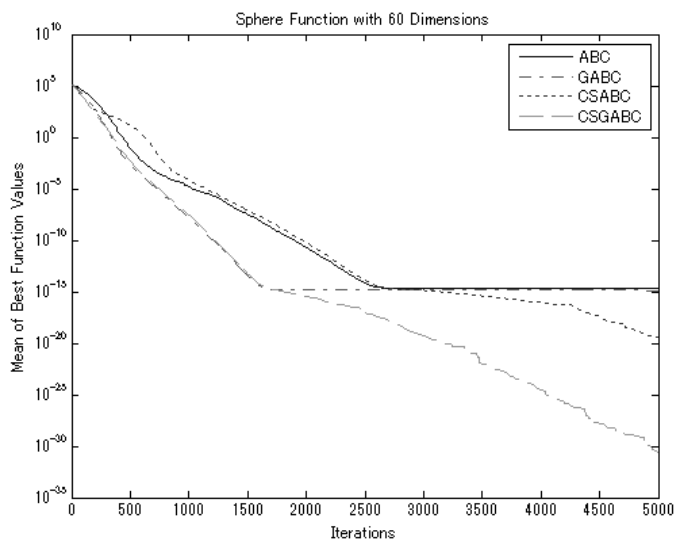


圖 10：Sphere Function 60D

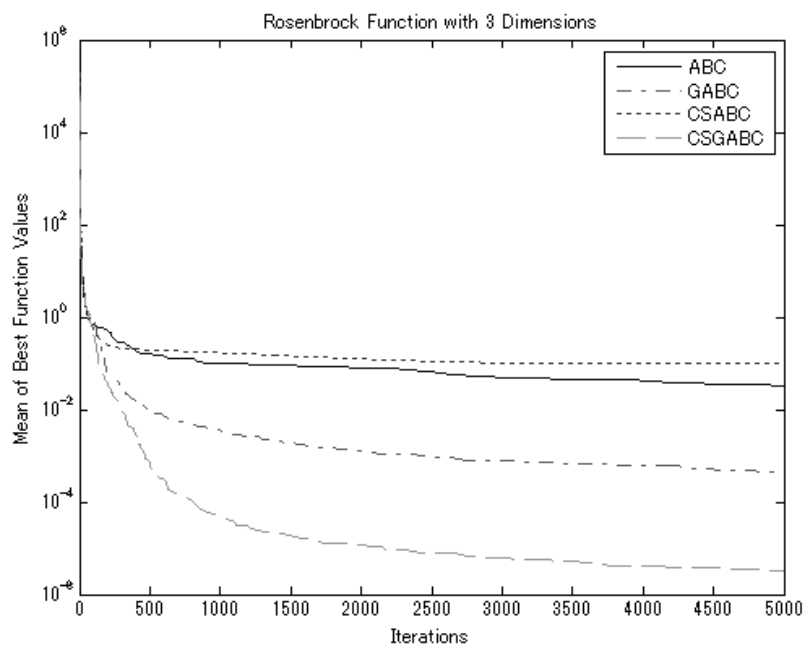


圖 11：Rosenbrock Function 3D

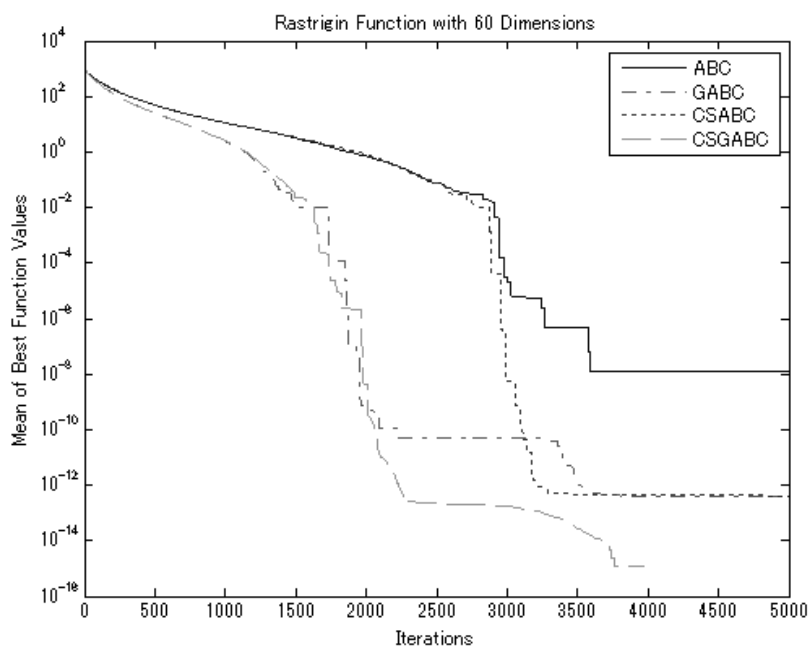


圖 12：Rastrigin Function 60D

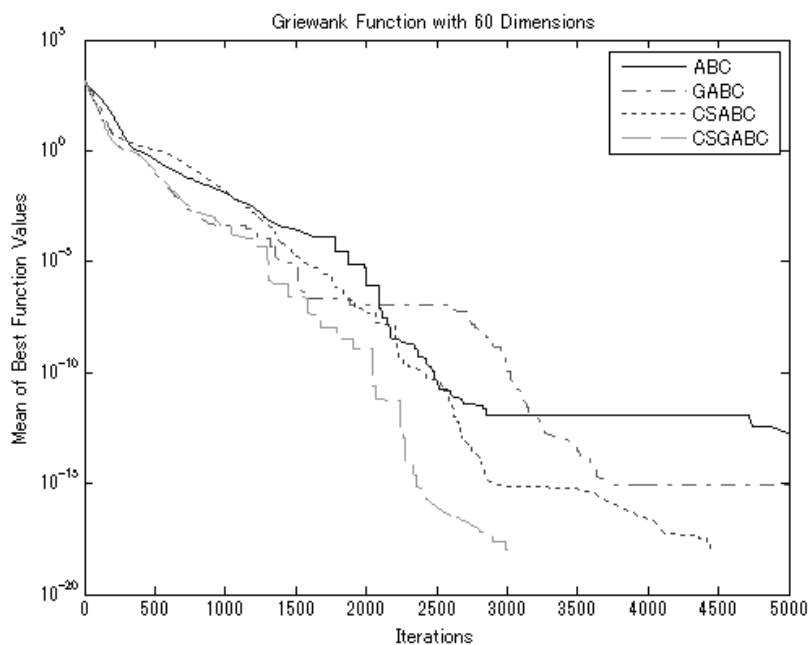


圖 13：Griewank Function 60D

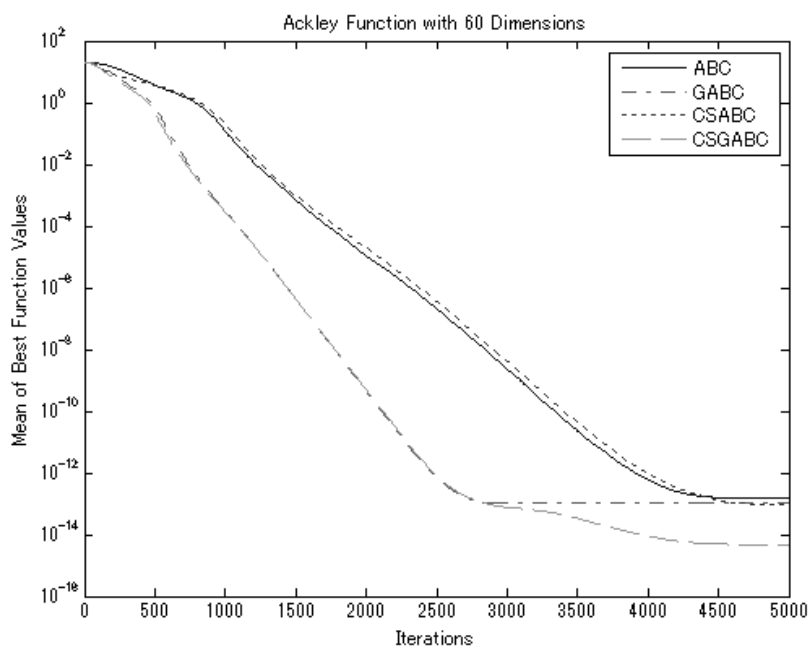


圖 14：Ackley Function 60D

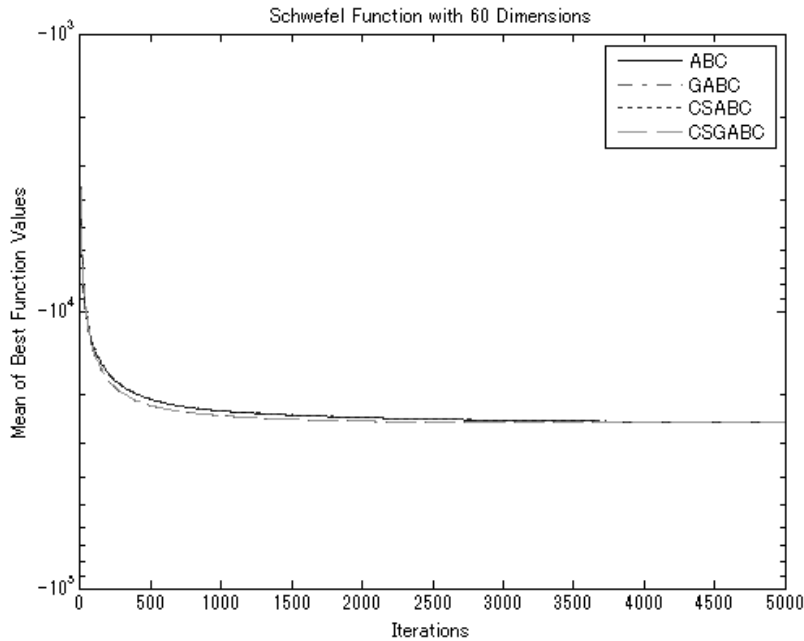


圖 15：Schwefel Function 60D

伍、結論與未來研究

本研究所提出之 CSABC，以群中心策略與相對位置策略改良人工蜂群演算法，在觀察蜂方面派遣最後一隻觀察蜂至所有食物源的中心位置進行搜索，以加強小範圍的搜尋能力；在偵察蜂方面利用相對位置策略，加強演算法跳脫區域最佳解的能力。除此之外，另將此改良方法與 GABC 演算法相結合，提出 CSGABC，用以試驗本研究與他種方法之間結合的可能性。

經過實驗，我們發現 CSABC 與 CSGABC 演算法在大部分的測試函數都有相當不錯的表現，顯示本研究所提出之群中心策略與相對位置策略，除了可單獨使用，也可結合其他蜂群演算法，以達到相輔相乘的綜效。

本研究的實驗，主要測試維度為 30 維與 60 維；後續研究，可針對不同的參數加以實驗，特別是高維度的測試（例如 300 維、500 維），以便更全面地檢視本方法之績效表現。

本論文運用群中心與相對位置策略，來加強演算法在解空間的搜尋能力。未來的學者也可嘗試運用不同演算法的概念（如：基因、粒子群及螞蟻演算法），以改良蜂群演算法，相信將有機會得到更多的突破。此外，本方法也可應用於工商業界的最佳化問題，對實務界應該會有相當的幫助。

參考文獻

- 鄭富升 (2005),『蟑螂演算法在含限制條件問題的應用』, 第十屆人工智慧與應用研討會論文集, 高雄, 台灣, 頁 89。
- 暴勵、曾建潮 (2010),『自我調整搜索空間的混沌蜂群演算法』, 計算機應用研究, 第二十七卷, 第四期, 頁 1330-1334。
- 羅鈞、樊鵬程 (2009),『基於遺傳交叉因子的改進蜂群優化算法』, 計算機應用研究, 第二十六卷, 第十期, 頁 3716-3717。
- Akay, B. and Karaboga, D. (2012), 'A modified artificial bee colony algorithm for real-parameter optimization', *Information Sciences*, Vol. 192, pp. 120-142.
- Alatas, B. (2010), 'Chaotic bee colony algorithms for global numerical optimization', *Expert Systems with Applications*, Vol. 37, No. 8, pp. 5682-5687.
- Basturk, B. and Karaboga, D. (2006), 'An Artificial Bee Colony (ABC) algorithm for numeric function optimization', *Proceedings of IEEE Swarm Intelligence Symposium (SIS 2006)*, Indiana, USA, May 12-14, pp. 12-14.
- Feng, Q.X. and Ding, H.J. (2008), 'Bee colony algorithm for the function optimization', *Science Paper Online*, available at http://www.paper.edu.cn/en/paper.php?serial_number=200808-448 (accessed 25 November 2013)
- Karaboga, D. (2005), 'An idea based on honey bee swarm for numerical optimization', Technical Report-TR06, Erciyes Univ. Press, Erciyes.
- Karaboga, D. and Akay, B. (2008), 'Solving large scale numerical problems using artificial bee colony algorithm', *Proceedings of the Sixth International Symposium on Intelligent and Manufacturing Systems Features, Strategies and Innovation*, Sakarya, Turkiye, October 14-17, pp. 14-17.
- Karaboga, D. and Basturk, B. (2007), 'A powerful and efficient algorithm for numerical function optimization: Artificial Bee Colony (ABC) algorithm', *Journal of Global Optimization*, Vol. 39, No. 3, pp. 459-471.
- Karaboga, D. and Basturk, B. (2008), 'On the performance of artificial bee colony (ABC) algorithm', *Applied Soft Computing*, Vol. 8, No. 1, pp. 687-697.
- Li, F.L., Ding, H.J. and Fang, X. (2007), 'The parameter improvement of bee colony algorithm in TSP problem', *Science Paper Online*, available at http://www.paper.edu.cn/en/paper.php?serial_number=200711-226 (accessed 25 November 2013)
- Liu, Y., Qin, Z., Shi, Z. and Lu, J. (2007), 'Center particle swarm optimization', *Neurocomputing*, Vol. 70, No. 4-6, pp. 672-679.
- Narasimhan, H. (2009), 'Parallel Artificial Bee Colony (PABC) algorithm', *World*

- Congress on Nature & Biologically Inspired Computing (NABIC 2009)*, Coimbatore, India, Dec 9-11, pp. 306-311.
- Quan, H. and Shi, X. (2008), 'On the analysis of performance of the improved artificial-bee-colony algorithm', *Proceedings of Fourth IEEE International Conference on Natural Computation*, Jinan, China, October 18-20, pp. 25-27.
- Rahnamayan, S., Tizhoosh, H.R. and Salama, M.M.A. (2006), 'Opposition-based differential evolution algorithms', *Proceedings of IEEE Congress on Evolutionary Computation (CEC 2006)*, Vancouver, Canada, July 16-21, pp. 2010-2017.
- Rahnamayan, S., Tizhoosh, H.R. and Salama, M.M.A. (2008), 'Opposition versus randomness in soft computing techniques', *Applied Soft Computing Journal*, Vol. 8, No. 2, pp. 906-918.
- Tizhoosh, H.R. (2005), 'Opposition-based learning: a new scheme for machine intelligence', *Proceedings of the Computational Intelligence for Modeling Control and Automation (CIMCA.2005)*, Vienna, Austria, Nov. 28-30, pp. 695-701.
- Tsai, P.W., Pan, J.S., Liao, B.Y., and Chu, S.C. (2009), 'Enhanced Artificial Bee Colony Optimization', *International Journal of Innovative Computing, Information and Control*, Vol. 5, No. 12, pp. 5081-5092.
- Xu, L., Krzyzak, A. and Oja, E. (1993). 'Rival penalized competitive learning for clustering analysis, RBF net, and curve detection', *IEEE Transactions on Neural Networks*, Vol. 4, No. 4, pp. 636-649.
- Yi, B., Qiao, H.Q., Yang, F. and Xu, C.W. (2010), 'An improved initialization center algorithm for K-means clustering', *Proceedings of 2010 International Conference on Computational Intelligence and Software Engineering (CISE 2010)*, Wuhan, China, Dec 10-12, pp. 1-4.
- Zhang, C., Ni, Z.W., Wu, Z.J. and Gu, L.C. (2009), 'A novel swarm model with quasi-oppositional particle', *Proceedings of International Forum on Information Technology and Applications (IFITA 2009)*, Chengdu, China, May 15-17, pp. 325-330.
- Zhu, G. and Kwong, S. (2010), 'Gbest-guided artificial bee colony algorithm for numerical function optimization', *Applied Mathematics and Computation*, Vol. 217, No. 7, pp. 3166-3173.