

模糊蜂群演算法

蔡賢亮¹ 卓子菱²

¹義守大學資訊管理學系

²義守大學資訊管理學系

摘要

本研究針對2005年學者 Karaboga 所提出之人工蜂群演算法(Artificial Bee Colony algorithm, ABC)進行探討。該演算具有收斂速度穩定、求解能力佳及少量參數控制之優點，但仍存在晚期收斂速度緩慢及陷入區域解等問題。本研究參考若干改良的人工蜂群演算法文獻，進而提出了模糊蜂群演算法。我們將模糊理論及差分演算法融合進人工蜂群演算法，幫助蜂群降低陷入區域解之情形且能有效地持續開發。另外在本研究中，我們還將模糊差分策略融入與目前效能較佳的人工蜂群演算法 GABC(Gbest-guided Artificial Bee Colony algorithm)，以更提升蜂群求解能力。

本研究利用測試函數進行實驗評估，從實驗結果得知，本研究提出之模糊蜂群演算法及與 GABC 相結合之模糊蜂群演算法，確實改善陷入區域解之情形，也使其在後期仍可以持續地有效開發最佳解。

關鍵詞：人工蜂群演算法、模糊理論、差分演算法。

¹ 義守大學資訊管理學系助理教授。

1. 前言

最佳化演算法簡單來說就是一個搜尋演算法，這類演算法針對一個給定的問題，此問題需定義合法的解集合（或稱解空間，solution space），還需定義適應函數(fitness function)用以判別一個解的好壞，然後最佳化演算法就會在解集合中找出符合要求的一個解。

最佳化問題廣見於工程、經濟等領域。很多最佳化問題本質上是屬於高困難度且帶有不確定性，很多問題都屬於 NP-Complete(NPC)問題。全世界有不少學者熱衷於使用隨機性方法，來求解這些高困難度之不確定性問題。目前最常見的最佳化演算法有人工蜂群演算法(Artificial Bee Colony algorithm, ABC)、粒子群演算法(Particle Swarm Optimization, PSO)、基因演算法(Genetic Algorithm, GA)、布穀鳥演算法(Cuckoo Search, CS)及模擬退火演算法 (Simulate Annealing Arithmetic, SA) 等等。

本研究主要針對人工蜂群演算法進行探討。人工蜂群演算法具較高執行效率及較少參數設定，已被廣泛地應用於很多領域，但它仍有求解能力不佳及過早收斂局域解的問題，因此本研究將針對此問題提出改善策略。在本論文中，我們稱之為模糊蜂群演算法。

模糊蜂群演算法係將模糊理論及差分演算法融合進人工蜂群演算法，幫助蜂群降低陷入區域解之情形且能有效地持續開發。在實驗中，我們利用公認的測試函數

將人工蜂群演算法與模糊蜂群演算法進行效能驗證分析。

2. 文獻探討

本章節中，首先介紹原始人工蜂群演算法、改良人工蜂群演算法。最後我們會探討模糊理論及差分演算法的基本概念。

2.1 人工蜂群演算法

人工蜂群演算法 (Artificial Bee Colony algorithm) 在2005被 Karaboga 所提出，用於解決最佳化問題。該演算法是模仿大自然中蜂群的覓食行為。作者經由探討蜜蜂的智慧覓食行為特性、蜜蜂的自我組織及分工觀念，了解到蜂群如何分工協助搜尋到最多的食物。該演算法之特性為運用少量的參數、流程架構簡單及運行速度較快，但它仍有陷入區域解及晚期收斂速度緩慢之問題。

2.1.1 人工蜂群演算法流程

Karaboga(2005)依據蜂群的覓食行為定義出人工蜂群演算法流程架構，定義操作流程如下：

- (1) 首先，初始化參數設定及雇用蜂的位置，將它的 limit 初始為0。其中每一隻雇用蜂的初始位置皆代表每一個食物源開採的位置。
- (2) 依據雇用蜂開採之食物源位置進行適應值(fitness，或稱食物收益度)評估。
- (3) 每一隻雇用蜂依據移動公式至鄰近食物源進行搜尋，並重新評估該食物源

適應值，如果新的食物源適應值較佳則前往開採，否則停留在原地且 limit 加1。

- (4) 雇用蜂開採過程會以舞蹈的方式將訊息傳遞給在蜂巢等待的觀察蜂，此時觀察蜂透過輪盤法挑選出某一個食物源前往搜尋，並評估該食物源適應值，如果新的食物源適應值較佳則前往開採，否則停留在原地且 limit 加1。
- (5) 當食物源收益度一直處於未改善之狀況(limit 值達到 LIMIT 門檻值)，此時蜂群將會啟動偵查蜂，隨機移動到新的食物源進行開採，並且將 limit 歸0。
- (6) 記錄當前最佳的食物源收益度(適應值)，並檢視是否符合終止條件。如果達到最大的迭代次數則結束，否則回到(3)重複執行，直到符合終止條件。

人工蜂群演算法有三個可調控的參數，蜂群數量(SN)、LIMIT 門檻值及最大迭代(MCN)。演算法開頭需先設定此三參數。

欲求解之問題空間維度為 D，每個維度的合適的解範圍[min, max]。依求解問題而有不同的設定。

首先隨機在解空間中放置雇用蜂，每隻雇用蜂即代表一個食物源，如公式(2-1)所示。每一個食物源 x_i ($i=1,2,...SN$)，都代表一個問題解向量。每一個食物源 x_i 的香味、花蜜量及位置等，不同的食物源特性可得到不同的適應值 fit_i ，如公式(2-2)所示。

$$x_{i,j} = min_j + rand(0,1) \times (max_j - min_j)$$

(2-1)

$x_{i,j}$ ：第 i 隻雇用蜂第 j 維度的初始值

$i=1, 2, 3, \dots SN, j=1, 2, 3, \dots D$

min_j ：解空間第 j 維度的最小範圍

max_j ：解空間第 j 維度的最大範圍

$$fit_i = F(x_i) \quad (2-2)$$

$F()$ 為解空間的適應函式

接下來，雇用蜂會移動至新的食物源 $v_{i,j}$ ，並根據公式(2-2)計算新食物源的適應值，然後以貪婪式選擇來決定是否前往新食物源進行採集。若得到比原先位置適應值較好，就前往該位置採集；若得到比原先適應值還差則停留在原食物源位置 x_i 繼續採集，並將 limit 加1。雇用蜂根據移動公式(2-3)做移動，且每次移動只會隨機挑選單一維度進行移動。

$$v_{i,j} = x_{i,j} + rand(-1,1) \times (x_{i,j} - x_{k,j}) \quad (2-3)$$

$x_{i,j}$ ：第 i 隻雇用蜂第 j 維度的位置

$v_{i,j}$ ：第 i 隻雇用蜂第 j 維度的移動後的新位置

$x_{k,j}$ ：隨機挑選第 k 隻雇用蜂第 j 維度的位置

雇用蜂每次採集完後會將食物源資訊傳遞給在觀察蜂，觀察蜂則會依據所接收到的適應值資訊，根據公式(2-4)計算每個食物源的輪盤機率。然後再以輪盤式選擇

法，決定觀察蜂欲前往協助開發之食物源。觀察蜂根據公式(2-2)移動至該食物源位置，根據公式(2-3)重新計算適應值，並決定是否要在原食物源位置採集食物。

$$P_i = \frac{fit_i}{\sum_{n=1}^{SN} fit_n} \quad (2-4)$$

當食物源收益度效果一直沒有得到改善時，偵查蜂會根據公式(2-1)在解空間中隨機尋找全新的食物源，並且將 limit 歸為0。偵查蜂每次移動時不一定會得到正向回饋，也有可能找到比原來的食物源更差的新食物源，但對於蜂群陷入局部區域解時，有極大的幫助使蜜蜂跳出區域解情形。

2.2 人工蜂群演算法相關文獻

人工蜂群演算法至今已有十餘年左右，而該演算之研究可分為兩種類型，分別為創新應用及演算法效能改良。

國內外有不少的學者針對原始人工蜂群演算法的搜尋能力進行改良研究。例如 Karaboga 及 Basturk(2006；2007)針對人工蜂群演算法的參數進行測試比較。並發現當蜂群數量(SN)、LIMIT 門檻值及最大迭代次數(MCN)均對人工蜂群之求解能力有重大影響。

有學者認為當 LIMIT 門檻值設定於族群數量×問題解維度($SN \times D$)或二分之一族群數量×問題解維度($0.5 \times SN \times D$)時，可以得到相對較好得最佳解(Karaboga and Basturk, 2008；2009)。

Zhu 與 Kwong 提出 Gbest-guided Artificial Bee Colony algorithm (GABC)，其基本架構與人工蜂群演算法相似，針對原始移動公式(2-2)進行修正，增加當前找到之最佳解作為引導，如公式(2-5)。藉由當前最佳解引導雇用蜂及觀察蜂往該方向移動搜尋，此方法簡單但效果顯著(Zhu and Kwong, 2010)。

$$v_{ij} = x_{ij} + rand(-1,1) \times (x_{ij} - x_{kj}) + rand(0,1.5) \times (y_j - x_{ij}) \quad (2-5)$$

x_{ij} ：第 i 隻雇用蜂第 j 維度的原始位置

v_{ij} ：第 i 隻雇用蜂第 j 維度的移動後的新位置

y_j ：第 j 維度的當前最佳解

2.3 模糊理論

2.3.1 模糊理論簡介

模糊理論 (Fuzzy Theory) 以 L.A. Zadeh 提出模糊集合為基礎，定義模糊集合用以強調連續性級別元素，以彌補傳統非 0 即 1 的二元邏輯缺點，改以 0 到 1 之間的數值表示(L.A. Zadeh, 1965)。1973 年 Zadeh 提出模糊邏輯(Fuzzy Logic)強調真假之間仍存在著模糊地帶，其可能為真亦可能為假，此邏輯推理有助於幫助獲得更好的結果 (Zadeh, 1975)。

2.3.2 模糊集合

傳統集合屬於明確的集合，強調非 0

即 1，缺少了中間模糊地帶，因此常常在判斷一些程度上的問題時，會有判斷錯誤的情況產生。面對這些邊界不明確的情形時，使用不明確的集合可以幫助誤判的發生情形減少，這些不明確的集合又稱作模糊集合。在模糊集合內的元素中，若某一元素歸屬於某集合的程度越大時，則歸屬程度將越接近於 1，反之當歸屬於某集合的程度越小時，則越接近於 0，如圖 1 所示。左圖為傳統集合以二分條件規定每一個元素只能屬於或不屬於某個集合；右圖為模糊集合，其擁有一個歸屬函數 (Membership Function)，經歸屬函數值可表示某一個元素對集合的歸屬程度 (Degree Of Membership)。

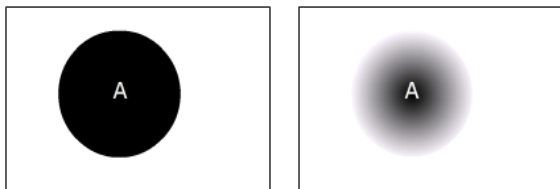


圖 1. 傳統集合與模糊集合

模糊集合包含許多層級元素，經由定義之歸屬函數將模糊集合內的元素進行量化，常見的歸屬函數可分為以下三角形、梯形、高斯歸屬函數，如下圖 2 所示。

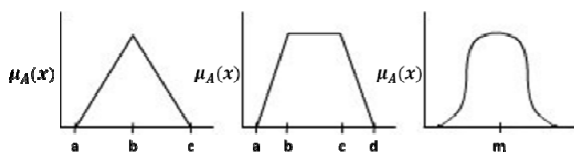


圖 2. 三種常見歸屬函數

2.4 差分演算法

2.4.1 差分演算法簡介

Storn 和 Price 於 1995 年提出差分演算法 (Differential Evolution, DE)，其概念以母體的差異性為基礎，透過個體差異向量在搜尋空間進行運算，加強區域探索能力和維持母體之多樣性。其運行流程與基因演算法十分相似，皆以隨機搜尋能力在空間進行探索，利用突變 (Mutation)、交配 (Crossover) 及選擇 (Selection) 等機制進行 (Storn et. Al, 1997)。

該演算法被證實為有效之優化方法，其概念簡單易於實作，運用少量的參數設定且能達到良好的收斂能力及高準確性 (Storn et. al. 1996)。目前已廣泛應用至許多組合最佳化問題，包含路徑規劃、網路應用及工程領域上。

2.4.2 差分演算法流程

差分演算法架構流程分別以初始化、突變、重組及選擇 (Mayer, 2005)，操作流程如下：

首先，初始化參數設定 (群體大小 SN、突變權重因子 F、交換率 CR)，並以隨機方式初始化目標向量，如公式 (2-5) 所示。

$$x_i(0) = [x_{i,1}, x_{i,2}, \dots, x_{i,d}], \quad (2-5)$$

$$i=1, 2, \dots, SN$$

SN：表示群體大小

d：表示維度

(1) 隨機從母體中挑選出三個目標向量，

分別為 $x_{r1}(t)$, $x_{r2}(t)$ 及 $x_{r3}(t)$ ，再透過突變權重因子 (F) 產生出新的向量 $v_i(t+1)$ ，如公式(2-6)所示。

$$v_i(t+1) = x_{r1}(t) + F(x_{r2}(t) - x_{r3}(t)) \quad (2-6)$$

$v_i(t+1)$ ：合成向量

F ：突變權重因子

t ：迭代數

(2) 經由突變後產生之合成向量 $v_i(t+1)$

與所選之目標向量 $x_i(t)$ 進行重組，重組後取得新的試驗向 $u_i(t+1)$ 量。當隨機亂數小於或等於交換率 (CR)，則得到新的試驗向量 $v_i(t+1)$ ；反之，隨機亂數大於交換率 (CR)，則 $x_i(t)$ 由取代之。如公式(2-7)所示。

$$u_i(t+1) = \begin{cases} v_i(t+1), & \text{if rand} \leq CR \\ x_i(t), & \text{if rand} > CR \end{cases} \quad (2-7)$$

$u_i(t+1)$ ：試驗向量

CR ：交換率

$rand$ ：亂數

經上述突變及重組後產生許多目標向量和試驗向量，透過適應值計算評估選出較好之目標向量和試驗向量，納入下一次迭代循環中的子代。如公式(2-8)所示。突變重組運行機制，如圖 3 所示。

$$x_i(t+1) = \begin{cases} u_i(t+1), & \text{if } F(u_i(t+1)) < F(x_i(t+1)) \\ x_i(t), & \text{else} \end{cases}$$

(2-8)

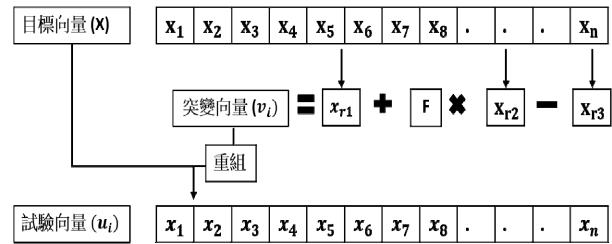


圖 3. 差分演算法之突變重組運行機制

3. 研究方法

在眾多最佳化演算法中，人工蜂群演算法具有不錯的求解能力，但從文獻探討得知它存在區域解及晚期開發緩慢之情況。因此本研究針對人工蜂群演算法進行改良。

3.1 研究流程

首先，我們針對人工蜂群演算法之特性進行初步實驗分析，探討此演算法的收斂效能及陷入區域解情況。接下來，我們針對它的特性及問題便提出若干改進策略，來改良人工蜂群演算法，以降低該演算法區域解之問題，並改善求解的效能。

在本研究中，我們會提出使用模糊理論及差分演算法來改良人工蜂群演算法，並會利用一些公認的測試函數比較人工蜂群演算法改良前後之性能並進行探討。

最後我們還會將改良策略融入目前搜尋能力較好的 GABC 人工蜂群演算法 (Gbest-guided Artificial Bee Colony algorithm)，並進行實驗探討。

3.2 區域最佳解定義

在本研究中，我們為了要對演算法評量陷入區域解的情況，所以我們必需定義何種形式的解屬區域解。區域解定義是用來判定演算法是否陷入區域解的情形。

我們對區域解的定義如下：演算法所求的解中，若有一個或一個以上的維度與真實最佳解差距大於解空間範圍的千分之一時，我們便可以認定此解為區域解。

3.3 初步實驗分析

初步實驗中，我們主要是針對人工蜂群演算法依據不同族群數量(SN)、LIMIT 門檻及問題空間維度(D)對人工蜂群演算法進行實驗探討。

實驗將族群數量(SN)設定 10 或 20、LIMIT 門檻設定為 100 或 $SN \times D$ 、問題維度(D)設定 30 或 60。每種設定組合各進行 100 次實驗，且每次實驗運行 5000 迭代。100 次的實驗平均結果如表 1、表 2、表 3 及表 4 所示。

綜合以上初步實驗探討，發現蜜蜂在數量少時，求解能力較差且有區域解之狀況發生。因此本研究主要以少量蜂群及高維度進行研究探討，希望可以幫助減少蜜蜂陷入區域解之情形產生。

表 1. D=60 及 LIMIT 門檻=100 實驗數據

Function	SN	Mean	Std.	Local	Min.	Max.
Sphere	10	3.52E-12	1.89E-11	0	5.01E-17	1.47E-10
	20	2.61E-13	6.52E-13	0	1.97E-15	4.24E-12
Rastrigin	10	1.38E-01	3.46E-01	17	1.68E-19	1.32E+00
	20	3.11E-09	3.07E-08	0	1.67E-19	3.07E-07
Girewank	10	1.03E-03	3.76E-03	8	2.13E-15	2.21E-02
	20	8.88E-07	6.00E-06	0	6.21E-20	4.84E-05
Ackley	10	5.23E-08	8.34E-08	0	8.37E-11	3.98E-07
	20	2.08E-08	2.44E-08	0	4.73E-10	1.16E-07

從實驗數據中，在相同 LIMIT 門檻及不同維度，族群數量(SN)多時，蜂群的求解能力比較好；族群數量(SN)少時，在部分測試函數中有陷入區域解之情形發生。

在相同維度及相同族群數量(SN)時，若將 LIMIT 門檻若設為 $SN \times D$ 時，確實明顯可以提升蜂群演算法的求解之能力，但陷入區域解得情況也普遍增加。

表 2. D=30 及 LIMIT 門檻=100 實驗數據

Function	SN	Mean	Std.	Local	Min.	Max.
Sphere	10	1.57E-16	6.25E-16	0	3.71E-24	5.22E-15
	20	5.35E-17	2.58E-16	0	1.47E-20	2.50E-15
Rastrigin	10	5.59E-10	5.51E-09	0	5.05E-21	5.51E-08
	20	8.27E-15	3.90E-14	0	2.71E-20	3.21E-13
Girewank	10	5.37E-04	2.81E-03	3	7.69E-19	2.01E-02
	20	1.31E-09	9.28E-09	0	6.53E-21	9.24E-08
Ackley	10	8.59E-09	2.37E-08	0	9.95E-12	1.64E-07
	20	3.10E-09	5.23E-09	0	5.22E-11	3.32E-08

3.4 研究方法介紹

人工蜂群演算法的概念是利用蜜蜂移動尋找食物源並採集花蜜。在人工蜂群演算法中，有兩種移動公式，雇用蜂及觀察蜂根據公式(2-2)移動，偵察蜂根據公式(2-1)移動。

表 3. D=60 及 LIMIT 門檻=SN×D 實驗數據

Function	SN.	Mean	Std.	Local.	Min.	Max.
Sphere	10	3.79E-20	1.44E-19	0	1.22E-22	9.35E-19
	20	2.04E-22	6.07E-22	0	5.75E-24	5.46E-21
Rastrigin	10	4.06E-02	1.96E-01	5	1.39E-22	9.95E-01
	20	3.93E-16	2.95E-15	0	1.87E-23	2.86E-14
Girewank	10	2.39E-03	7.04E-03	14	8.62E-24	4.19E-02
	20	5.93E-04	2.88E-03	5	1.51E-24	1.97E-02
Ackley	10	3.53E-11	6.38E-11	0	1.24E-12	4.30E-10
	20	3.20E-12	2.48E-12	0	5.55E-13	1.58E-11

表 4. D=30 及 LIMIT 門檻=SN×D 實驗數據

Function	SN.	Mean	Std.	Local.	Min.	Max.
Sphere	10	6.92E-40	6.42E-39	0	9.81E-46	6.42E-38
	20	1.46E-43	3.99E-43	0	1.17E-46	2.89E-42
Rastrigin	10	1.43E-14	7.27E-14	0	1.32E-36	6.09E-13
	20	1.30E-29	7.73E-29	0	3.26E-40	7.23E-28
Girewank	10	1.63E-03	4.94E-03	11	3.68E-26	2.21E-02
	20	1.27E-12	9.16E-12	0	1.54E-26	7.93E-11
Ackley	10	8.43E-14	1.29E-13	0	3.24E-14	1.14E-12
	20	4.08E-14	4.24E-15	0	3.24E-14	5.02E-14

觀察蜂要往那個食物源移動是由輪盤法決定的。較佳的食物源會得到較好的開發，期望能尋獲更好的食物源。但這種機制是否會造成過度開發當前較佳解的區域，放棄當前較差的食物源的區域，因此也很可能就陷入了區域解。

人工蜂群演算法為了改善蜂群陷入區域解的現象，在每次迭代中會適時啟動偵查蜂，以隨機方式產生新位置並移動過去，確實可以幫助蜜蜂有效地跳脫區域解。

針對人工蜂群演算法之問題，近年來紛紛有學者提出一些改善策略，例如演算法中加入群中心策略，讓觀察蜂每次迭代

都會挑選一隻最差的蜜蜂移動至群中心(李維平、李元傑、謝明勳，2014)；有學者運用多樣性策略和差分演算法中的DE/rand/1 突變機制改善求解過程陷入區域解之情況，同時並兼顧求解過程的開發和搜尋能力(李維平、簡璟蔚、蔡宛庭，2011)。

本研究提出的改良策略，主要參考李維平所提出的群中心策略及多樣性策略。在群中心策略中提及到「群體中心位置通常與最佳解最為接近」，因此本改良策略是透過群體中心位置做為每次迭代中評估食物源之優劣標準，然後透過模糊理論去表示優劣程度，最後結合差分演算法的突變機制改進人工蜂群演算法，希望能提升蜂群的突變能力，讓蜂群可以降低陷入區域解之情況發生，並提升求解的能力。

3.4.1 模糊蜂群演算法

在本節中，我們提出利用模糊理論與差分演算法改良人工蜂群演算法。在本研究中，我們稱之為模糊蜂群演算法(FABC)。

人工蜂群演算法在每次迭代運行時，不依據食物源的好壞，皆以相同的方式進行開發及探索。模糊蜂群演算法提出每次迭代蜜蜂移動之前，先對蜜蜂優劣層度區別，以群體間蜜蜂採集的食物源適應值的好壞進行階層劃分。對於群體間搜尋能力

較差的蜜蜂其突變機率增加，幫助其能跳脫區域解；群體間搜尋能力較好的蜜蜂保留原人工蜂群演算法的開發方式。我們將群體間搜尋能力較差的蜜蜂透過差分突變方式，降低陷入區域解之情形，並能有效持續開發。

首先，運用模糊理論的概念對食物源的優劣(或蜂群適應值的好壞)進行層級劃分。劃分標準依蜂群平均適應值(fit_{ave})加上集體蜂群平均適應值的 ± 0.5 倍(此範圍定義是我們利用嘗試錯誤法界定其範圍)。蜂層級劃分方式，如公式(3-1)所示。計算結果可得到食物源差的歸屬程度 μ_a 。

$$\mu_{a,i} = \begin{cases} 1, & fit_i \geq (fit_{ave} \times 1.5) \\ 0, & fit_i \leq (fit_{ave} \times 0.5) \\ \frac{fit_i - (fit_{ave} \times 0.5)}{fit_{ave}}, & (fit_{ave} \times 0.5) \leq fit_i \leq (fit_{ave} \times 1.5) \end{cases} \quad (3-1)$$

$$i=1, 2, 3, \dots, SN$$

經上述公式(3-1)計算取得每個食物源差的歸屬程度值 μ_a ，將它結合於移動公式中。食物源差的歸屬程度 μ_a 為 1，表示該蜜蜂所在食物源較差，也表示該隻蜜蜂可能陷入區域解中，我們需要讓該蜜蜂跳出區域解。

為了降低蜜蜂陷入區域解之情形，應用差分演算法的 DE/rand/1 突變公式對原始人工蜂群演算法之移動公式進行改良，

透過公式(3-2)所示。公式前半段保留原始人工蜂群演算法的移動公式，而後半段結合差分演算法移動公式，並結合食物源差的歸屬程度值 μ_a 進行相乘。我們希望讓搜尋能力較差的蜜蜂可藉由差分演法之突變公式，可以更隨機的跳動以降低落入區域解。

$$v_{i,j} = x_{i,j} + \text{rand}(-1,1) \times (x_{i,j} - x_{k,j}) + (\mu_{a,i} \times F(x_{r2,j} - x_{r3,j})) \quad (3-2)$$

$x_{i,j}$ ：第 i 隻雇用蜂第 j 維度的原始位置

$v_{i,j}$ ：第 i 隻雇用蜂第 j 維度的移動後的新位置

$\mu_{a,i}$ ：第 i 隻雇用蜂搜尋能力差的歸屬程度值

F ：突變權重因子

3.4.2 模糊蜂群演算法流程架構

以下為模糊蜂群演算法流程架構，定義操作流程如下：

- (1) 初始化參數設定，根據公式(2-1)初始雇用蜂的位置。每一隻雇用蜂的初始位置皆代表每一個食物源的位置。
- (2) 根據公式(2-2)計算各個食物源的食物收益度(適應值)。
- (3) 依每個食物源的適應值，根據公式(3-1)計算每個食物源差的歸屬程度值 μ_a 。
- (4) 每一隻雇用蜂依據新的移動公式(3-2)進行食物源開採，並評估該食物源適應值，若新的食物源適應值較佳則前

- 往開採，否則停留在原地且 limit 加 1。
- (5) 再依每個食物源的適應值，根據公式(3-1)計算每個食物源的歸屬程度值 μ_a 。
- (6) 觀察蜂依輪盤法運用新的移動公式(3-2)前往協尋，並評估該食物源收益度(適應值)，若新的食物源收益度(適應值)較佳則前往開採，否則停留在原地且 limit 加 1。
- (7) 當食物源收益度一直處於未改善之狀況，此時蜜蜂移動到新的食物源進行開採，且 limit 歸 0。
- (8) 最後記錄這個迭代的當前最佳食物源資訊，並檢視是否符合終止條件，如果達到迭代次數則結束，否則回步驟(3)重複執行，直到符合終止條件。

3.5 模糊 GABC 演算法

3.5.1 模糊演算法結合 GABC 演算法

本研究中，我們除了提出模糊策略改良原始人工蜂群演算法，另外我們也將模糊改良策略與 Zhu 及 Kwong 提出 Gbest-guided Artificial Bee Colony algorithm (GABC) 進行結合。在本研究中，我們稱之為 FGABC 蜂群演算法。該演算法仍保有模糊蜂群之移動公式(3-2)，並增加 GABC 將當前最佳解作為引導的機制，增強蜂群的開發能力。

依據模糊改良策略，對所有食物源的優劣進行層級劃分，劃分標準仍依每次迭

代中集體蜂群平均適應值(fit_{ave})加上集體蜂群平均適應值的 ± 0.5 倍。依據該範圍劃分蜜蜂層級以判定蜜蜂歸屬程度。依據公式(3-1) 計算可得食物源差的模糊歸屬程度 μ_a 。依據公式(3-3)計算可得食物源佳的模糊歸屬程度 μ_b 。

$$\mu_{b,i} = \begin{cases} 0, & fit_i \geq (fit_{ave} \times 1.5) \\ 1, & fit_i \leq (fit_{ave} \times 0.5) \\ 1 - \frac{fit_i - (fit_{ave} \times 0.5)}{fit_{ave}}, & (fit_{ave} \times 0.5) \leq fit_i \leq (fit_{ave} \times 1.5) \end{cases} \quad (3-3)$$

經上述公式(3-1)及公式(3-3)計算取得蜂群歸屬函數值 μ_a 及 μ_b ，將其應用在改良模糊蜂群之移動公式(3-4)中。公式(3-4)前半段保留模糊蜂群演算法的移動公式，而後半段結合 GABC 移動公式，並將先前求得的食物源模糊歸程度 μ_a 及 μ_b 進行相乘。這種結合有二種功能：1. 能讓較差的食物源可以藉由差分演法之突變公式，增加隨機的跳脫能力，以降低落入區域解。2. 能讓較佳的食物源可以透過當前最佳解 Y_j 作為引導，增強蜂群的開發能力。如公式(3-4)所示。

$$v_{i,j} = x_{i,j} + \text{rand}(-1,1) \times (x_{i,j} - x_{r1,j}) + (\mu_{a,i} \times F(x_{r2,j} - x_{r3,j})) + (\mu_{b,i} \times \text{rand}(0,1.5) \times (y_j - x_{ij})) \quad (3-4)$$

μ_a ：食物源差的歸屬程度值

μ_b ：食物源佳的歸屬程度值

y_j : 為當前最佳解

3.5.2 模糊理論結合 GABC 演算法流程

以下為模糊蜂群演算法流程架構，定義操作流程如下：

- (1) 初始化參數設定，根據公式(2-1)初始化雇用蜂的位置。每一隻雇用蜂的初始位置皆代表每一個食物源開採的位置。
- (2) 根據公式(2-2)計算食物源的適應值。
- (3) 依每個食物源的適應值，根據公式(3-1)及公式(3-3)計算每個食物源的歸屬程度值 μ_a 及 μ_b 值。
- (4) 每一隻雇用蜂依據新的移動公式(3-4)進行食物源開採，並評估該食物源適應值，若新的食物源適應值較佳則前往開採，否則停留在原地且 limit 加 1。
- (5) 再依每個食物源的適應值，根據公式(3-1)及公式(3-3)計算每個食物源的歸屬程度值 μ_a 及 μ_b 值。
- (6) 觀察蜂運用新的移動公式(3-4)前往協尋，並評估該食物源適應值，若新的食物源適應值較佳則前往開採，否則停留在原地且 limit 加 1。
- (7) 當食物源收益度一直處於未改善之狀況，此時蜜蜂將會轉換成偵查蜂移動到新的食物源進行開採，且 limit 歸 0。
- (8) 記錄當前最好的食物源適應值，並檢視是否符合終止條件，如果達到迭代次數則結束，否則回到步驟(3)重複執

行，直到符合終止條件。

4. 實驗分析

本章節將對人工蜂群演算法(Artificial Bee Colony algorithm)及本研究所提出之模糊蜂群演算法(Fuzzy Artificial Bee Colony algorithm)及 FGABC 蜂群演算法做效能實驗分析。在實驗中，我們利用公認的測試函數進行實驗測試，並針對測試結果進行比較求解效能陷入區域解問題探討。

4.1 實驗環境

本研究之實驗環境，如表 5 所示。

表 5. 實驗環境

作業系統	Windows 10
處理器	Intel(R) Core(TM) i5
記憶體	16.0 GB
執行程式語言	Java

4.2 實驗參數設定

本研究之實驗的設定參數，如表 6 所示。

表 6. 實驗參數

族群數目(SN)	10
維度(D)	60
迭代次數	5000
實驗評估次數	100
LIMIT門檻	100

4.3 測試函數

本實驗利用數學型函數進行實驗測試，如附錄所示。共有 4 個測試函數，分別有單模態與多模態，其中 Sphere 函數為單模態函數，為平滑曲線且只有一個全域最小值，用來測試求解優越性；Rastrigin、Girewank 及 Ackley 為非線性的多模態函數，具許多局部最小值，較不易搜尋到全域最小值，可用來測試搜尋狀態是否過早收斂到區域解。在實驗中，我們依照不同測試函數的範圍界定及最小值來進行實驗測試，但為了確保有效檢視該演算法之效能，將利用多次實驗所求得之平均值 (Mean)、標準差 (Standard Deviation)、區域解次數 (Local)、最小值 (Minimums) 與最大值 (Maximums) 進行檢驗。平均值用以檢視求解優越性；標準差用以觀察該演算法求解之穩定性；區域解次數可辨別實驗結果掉入區域解情況；最小值則可以呈現 100 次實驗的最佳結果；最大值則可以呈現 100 次實驗的最差結果。

4.4 實驗結果

4.4.1 模糊蜂群演算法 (FABC)

在本節中，我們將針對模糊蜂群演算法 (FABC) 與人工蜂群演算法 (ABC) 進行比較。根據表 6 設定之實驗參數針對四個測試函數進行 100 次實驗分析。

Sphere 測試函數結果如表 7 所示。表 7 呈現 FABC 在開發能力與穩定性上，都較 ABC 還要優越。從圖 4 中，100 次實驗平均比較呈現 ABC 在 500 迭代前開發速度較 FABC 優越，但 500 迭代以後開發能力逐漸緩慢，而 FABC 仍持續保有穩定開發能力。從圖 5 中，前 20 次最佳實驗平均比較呈現與圖 4-1 相似的結果。

表 7. Sphere 測試函數 100 次實驗平均結果以較

Algorithm	Mean	Std.	Local	Min.	Max.
ABC	1.92E-12	8.10E-12	0	1.14E-16	7.79E-11
FABC	5.22E-31	9.77E-31	0	1.02E-32	5.36E-30

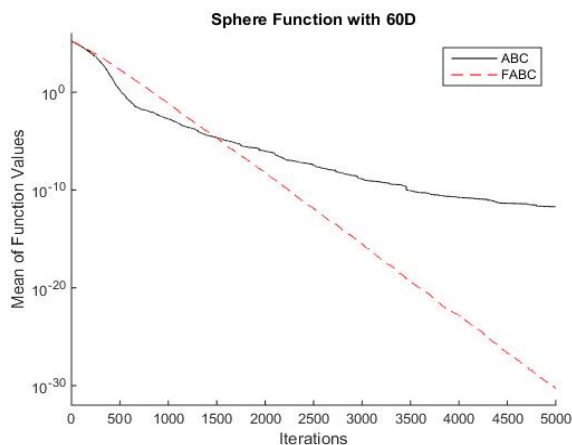


圖 4. Sphere 100 次實驗平均

Rastrigin 測試函數結果如表 8 所示。表 8 呈現 FABC 在開發能力略勝於 ABC，穩定性上也較 ABC 好一些。雖然在區域解的部分 FABC 確實明顯比 ABC 少許多，但仍有區域解情況存在。從最小值來看，

FABC 的求解能力較 ABC 好。

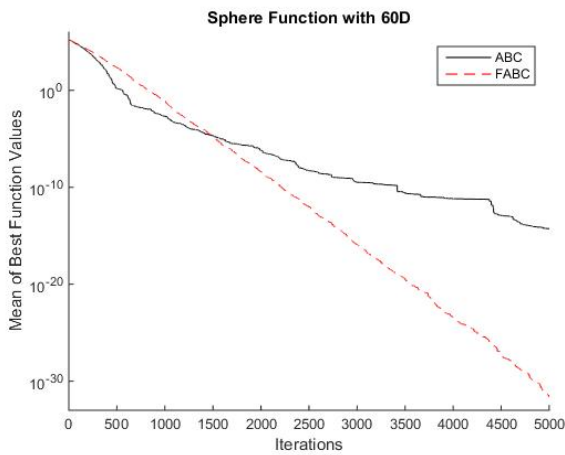


圖 5. Sphere 前 20 次最佳解平均

從圖 6 所示，100 次實驗平均比較呈現 ABC 及 FABC 在 3400 迭代前都保有穩定的開發能力，但 ABC 在 3500、4250 及 4750 迭代後有開發緩慢及停滯的情況產生；反觀 FABC，雖然 3500 迭代前收斂速度較 ABC 稍緩慢，但仍保有後續持續穩定開發之能力。從圖 7 所示，前 20 次最佳解平均呈現 FABC 在迭代前期收斂速度和 ABC 相差不大，但後期開發能力較 ABC 好。

表 8. Rastrigin 測試函數 100 次實驗平均結果以較

Algorithm	Mean	Std.	Local	Min.	Max.
ABC	1.45E-01	3.44E-01	19	3.58E-15	1.19E+00
FABC	1.12E-02	1.00E-01	2	1.87E-19	9.95E-01

Girewank 測試函數結果如表 9 所示。

表 9 呈現 FABC 在開發能力略勝於 ABC，穩定性上也較 ABC 好一些。雖然在區域解的部分 FABC 確實明顯比 ABC 少許多，但仍有區域解情況存在。從最小值來看 FABC 的求解能力較 ABC 好。

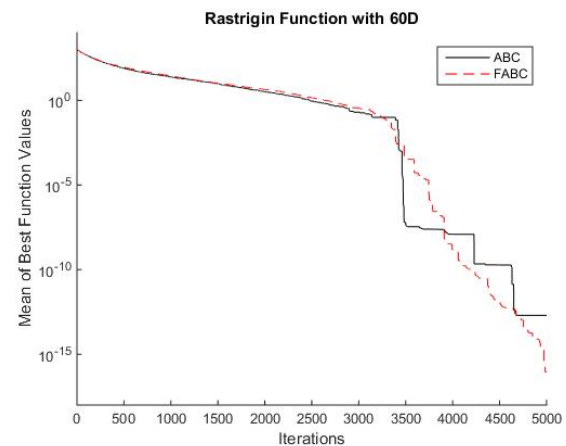


圖 6. Rastrigin 100 次實驗平均

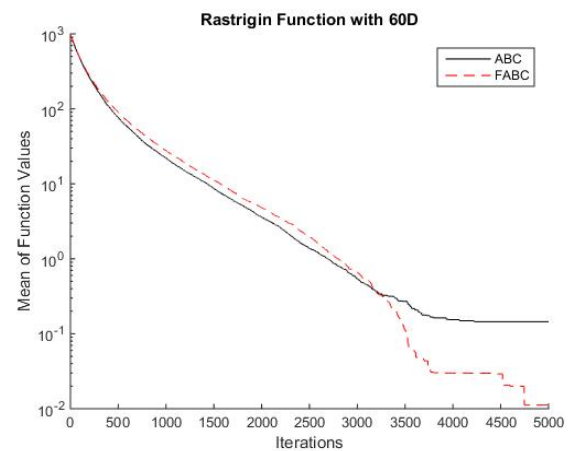


圖 7. Rastrigin 前 20 次最佳解平均

從圖 8 所示，100 次實驗平均比較呈現 ABC 在 1200 迭代前開發能力優於 FABC，但在 1200 迭代後 ABC 開發速度有逐漸緩慢且停滯的情況產生；反觀 FABC 在 1200 迭代後開發速度也有逐漸

緩慢的趨勢，但搜尋能力仍較 ABC 好。

從圖 9 所示，前 20 次最佳解平均呈現 FABC 在迭代前期收斂速度較 ABC 稍差，但後期開發能力較 ABC 好且有持續開發能力，且陷入去區域解的情況較 ABC 少。

表 9. Girewank 測試函數 100 次實驗平均結果以較

Algorithm	Mean	Std.	Local	Min.	Max.
ABC	1.74E-03	6.23E-03	9	1.11E-16	3.83E-02
FABC	2.99E-04	1.42E-03	2	1.24E-22	8.35E-03

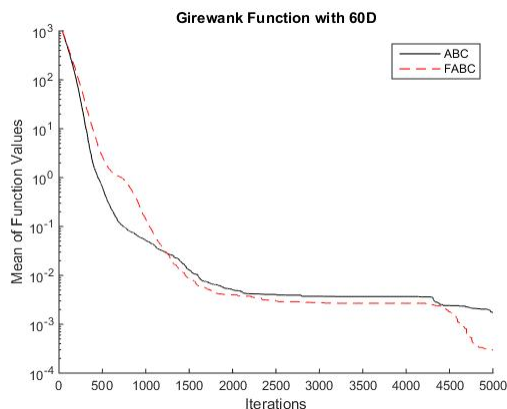


圖 8. Girewank 100 次實驗平均

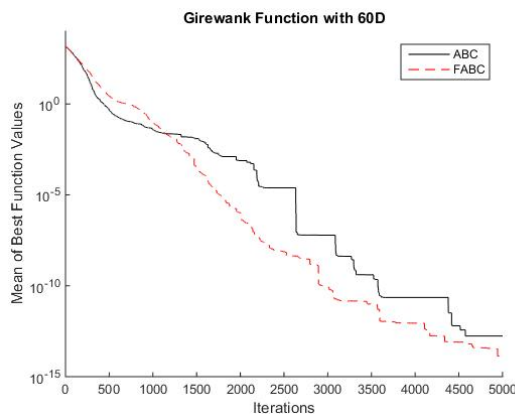


圖 9. Girewank 前 20 次最佳解平均

Ackley 測試函數結果如表 10 所示。

表 10 呈現 FABC 在開發能力優於 ABC，穩定性上也較 ABC 好一些，且 FABC 搜尋最小值及最大值之求解能力比 ABC 優。

從圖 10 所示，100 次實驗平均比較呈現 ABC 及 FABC 在 3500 迭代前開發能力優於 FABC，但在 3500 迭代後 ABC 開發速度有逐漸緩慢的情況產生；FABC 在 3500 迭代前收斂速度及開發情況確實比 ABC 來的緩慢，但在 3500 迭代後仍一直保持持續開發之情形。從圖 11 所示，前 20 次最佳解平均呈現 FABC 在迭代前期收斂速度確實較 ABC 差，但後期開發能力較 ABC 好且有持續開發之能力。

表 10. Ackley 測試函數 100 次實驗平均結果以較

Algorithm	Mean	Std.	Local	Min.	Max.
ABC	4.28E-08	5.83E-08	0	6.43E-10	3.30E-07
FABC	8.82E-11	1.10E-10	0	1.11E-13	6.79E-10

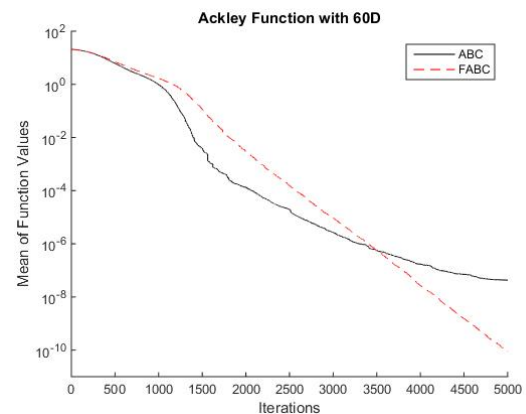


圖 10. Ackley 100 次實驗平均

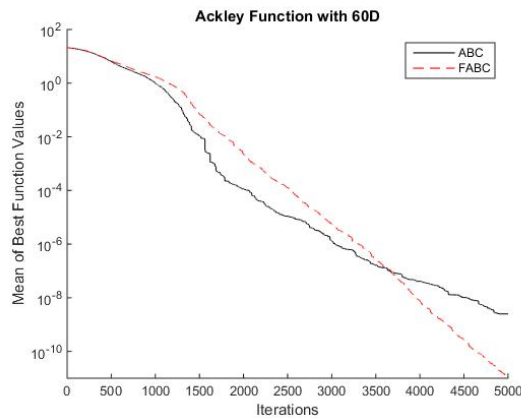


圖 11. Ackley 前 20 次最佳解平均

4.4.2 模糊 GABC 演算法(FGABC)

在本節中，我們將針對模糊 GABC 演算法與 GABC 蜂群演算法進行比較。根據表 6 設定之實驗參數針對四個測試函數進行 100 次實驗分析。

Sphere 測試函數結果如表 11 所示。表 11 呈現 FGABC 在開發能力、穩定性、最小值及最大值都較 GABC 還要優越。

從圖 12 所示，100 次實驗平均比較呈現 GABC 在 1500 迭代前開發速度較 FGABC 優越，但在 1500 迭代以後開發能力逐漸緩慢，但 FGABC 在 1500 迭代後仍持續保有穩定開發能力。圖 13 所示，前 20 次最佳解平均呈現 GABC 還是有晚期開發能力下降的情況產生；FGABC 則保有持穩定開發之能力。

表 11. Sphere 測試函數 100 次實驗平均結果以較

Algorithm	Mean	Std.	Local	Min.	Max.
GABC	4.94E-24	4.37E-23	0	1.60E-40	4.36E-22
FGABC	5.87E-47	1.28E-46	0	5.43E-49	1.10E-45

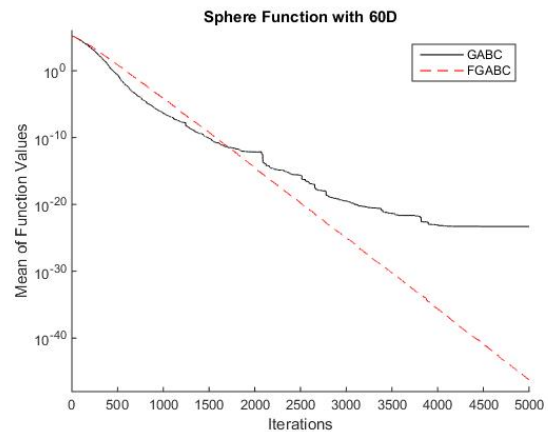


圖 12. Sphere100 次實驗平均

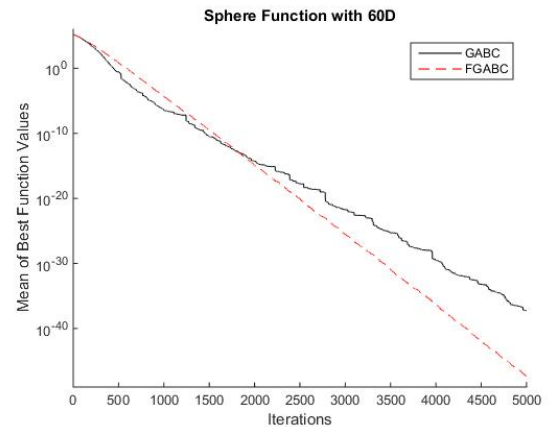


圖 13. Sphere 前 20 次最佳解平均

Rastrigin 測試函數結果如表 12 所示。表 12 呈現 FGABC 在開發能力及穩定性皆比 GABC 好一些，區域解的部分兩者之間差異不大，但 FGABC 在求解最小值的求解優越性比 GABC 好。

從圖 14 所示，100 次實驗平均比較呈現 GABC 在 2000 迭代後開發能力有逐漸減緩之趨勢且可能有陷入區域解的情況產生，但直到 3500 迭代開始其開發能力有提升之現象；FGABC 在 2500 迭代前都保有穩定的開發能力，但在 2500 迭代之後開發速度逐漸減緩，直到 4800 迭代開發能力才逐漸提升。從整體上來看兩者間都可能有區域解的情況產生，但 FGABC 的開發能力仍較 GABC 好。從圖 15 所示，前 20 次最佳解平均呈現 GABC 求解優越性及開發能力較 FGABC 差。

表 12. Rastrigin 測試函數 100 次實驗平均結果以較

Algorithm	Mean	Std.	Local	Min.	Max.
GABC	2.18E-02	1.48E-01	5	6.06E-31	1.10E+00
FGABC	3.44E-03	1.94E-02	4	1.21E-37	1.37E-01

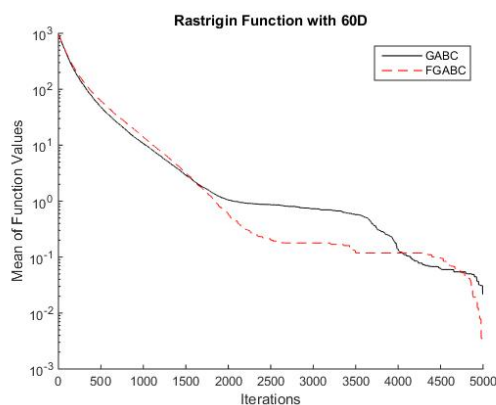


圖 14. Rastrigin 100 次實驗平均

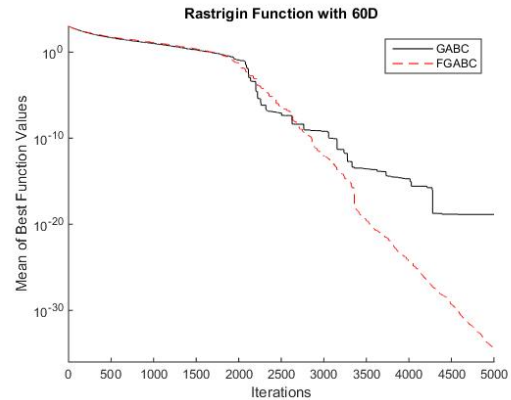


圖 15. Rastrigin 前 20 次最佳解平均

Girewank 測試函數結果如表 13 所示。

表 13 呈現 FGABC 在開發能力及穩定性上較 GABC 好一些。但陷入區域解的次數，FGABC 確實明顯比 ABC 少。從求解最小值來看 FGABC 的求解能力較 GABC 好。

從圖 16 所示，100 次實驗平均比較呈現 FGABC 在 1000 迭代前收斂速度較 GABC 差，但在 100 迭代後 GABC 開發速度有逐漸緩慢的情形，而 FGABC 開發速度雖然有逐漸緩慢的趨勢，但搜尋能力仍較 GABC 好，且在 3800 和 4800 迭代後收斂速度有提升之現象。從圖 17 所示，前 20 次最佳解平均呈現 FGABC 整體收斂狀況較 GABC 好，且其保有穩定之開發能力，而 GABC 在 2000 至 3000 迭代間則開發停滯之現象產生，隨然後其仍有持續開發的情形，但 FGABC 的開發能力仍比 GABC 好。

表 13. Girewank 測試函數 100 次實驗平均結果以較

Algorithm	Mean	Std.	Local	Min.	Max.
GABC	7.93E-04	2.83E-03	8	2.00E-22	1.72E-02
FGABC	1.00E-04	9.86E-04	1	1.91E-37	9.86E-03

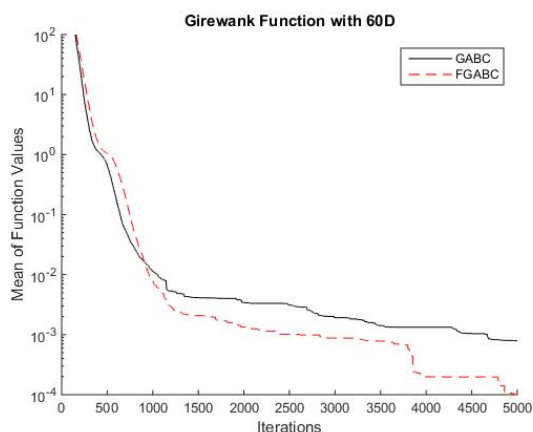


圖 16. Girewank 100 次實驗平均

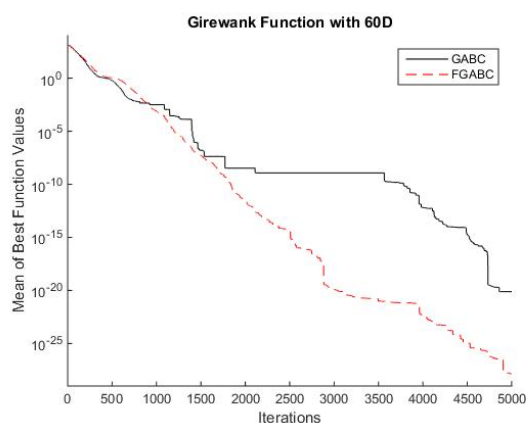


圖 17. Girewank 前 20 次最佳解平均

Ackley 測試函數結果如表 14 所示。

表 14 呈現 FGABC 在開發能力優於 GABC，穩定性上也較 GABC 好一些。

從圖 18 所示，100 次實驗平均比較呈現 GABC 在 1000 至 2200 迭代時開發

能力浮動較大，很有可能在前期有陷入區域解而後又跳脫區域解的情況產生；反觀 FGABC 則仍持續穩定開發的狀態，但是兩者在 3000 迭代後都有開發緩慢且停滯的情況產生。從圖 19 所示，前 20 次最佳解平均呈現 FGABC 在 3000 迭代前期收斂速度確實較 GABC 緩慢，但後期開發能力較 GABC 好且有持續開發之能力。

表 14. Ackley 測試函數 100 次實驗平均結果以較

Algorithm	Mean	Std.	Local	Min.	Max.
GABC	2.00E-11	8.73E-11	0	1.99E-13	7.61E-10
FGABC	1.25E-13	1.49E-14	0	9.64E-14	1.71E-13

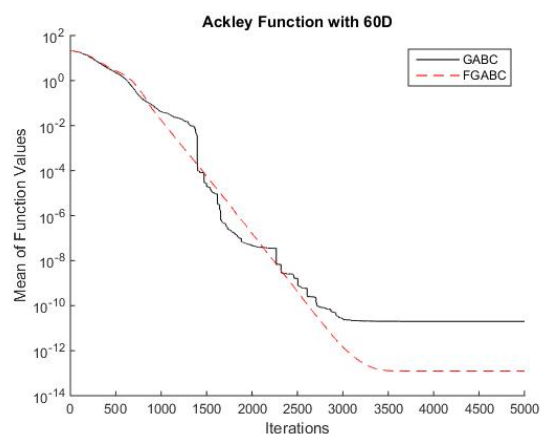


圖 18. Ackley 100 次實驗平均

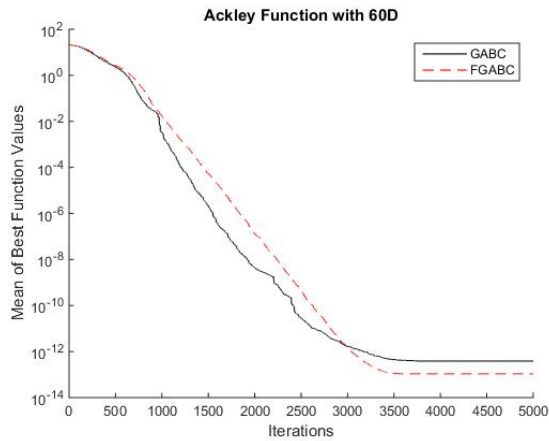


圖 19. Ackley 前 20 次最佳解平均

5. 結論與建議

5.1 結論

本研究所提出之 FABC 與 FGABC，在經過測試函數實驗測試後，從上述實驗分析結果都呈現確實使用模糊理論與差分演算法可以幫助蜂群在搜尋時降低陷入區域解的情況，且可以看出其開發能力確實有提升。從整體的收斂情況來看，FABC 及 FGABC 在大部分的測試函數中，開發前期的收斂狀況確實皆比 ABC 及 GABC 來的差，但是 FABC 及 FGABC 後期在開發穩定性、求解能力皆比 ABC 及 GABC 來的好。從此實驗結果，可顯示雖然改良後的 FABC 及 FGABC 對於區域解的減緩確實有不錯的幫助，此外也能保有持續開發的效果。由以上的分析證明本研究使用之模糊策略，的確可有效減少人工蜂群演算法陷入區域解之情形、提升開發能力及提升求解的品質。

5.2 建議

從眾多的人工蜂群演算法文獻中，我們發現有若干論文的實驗平均數據出現平均解為 0，標準差也為 0，這結果是很不可思議，因為這代表完全準確找到真實最佳解。其實並非如此，導致這種實驗結果主要原因在程式撰寫，程式設計者疏忽了浮點數有效位數的特性，所以建議讀者在撰寫程式時，要留意浮點數有效位數的問題。

參考文獻

- [1]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，21(1)，25-44。
- [2]李維平、簡璟蔚、蔡宛庭(2011)，改良突變權重的差分進化演算法，進工程學刊，vol.6。
- [3]Seeley, T.D. (1995) The Wisdom of the Hive. Harvard University Press, Cambridge.
- [4]Karaboga, D. (2005). "An idea based on honey bee swarm for numerical optimization", Technical Report TR06, Erciyes University, Engineering Faculty, Computer Engineering Department.
- [5]Karaboga, D. & Basturk, B. (2006). "An artificial bee colony (ABC) algorithm for numeric function optimization", In: Proceedings of the IEEE Swarm Intelligence Symposium 2006, Indianapolis, Indiana, USA, pp. 12-14.
- [6]Karaboga, D. & Basturk, B. (2007). "A powerful and Efficient Algorithm for Numerical Function Optimization: Artificial Bee Colony (ABC) Algorithm. Journal of Global Optimization", Vol. 39:1, pp. 459-171.

- [7]Karaboga, D. & Basturk, B. (2008). "On the performance of artificial bee colony (ABC) algorithm", Applied Soft Computing, Vol. 8, No. 1, pp. 687-697.
- [8]Karaboga, D. & Akay, B. (2009). "A Comparative Study of Artificial Bee Colony Algorithm", Applied Mathematics and Computation, Vol. 214, 108-132.
- [9]Storn, R. & Price, K. (1995). "Differential evolution—A simple efficient adaptive scheme for global optimization over continuous spaces", Technical report, California: International Computer Science Institute, Berkeley.
- [10]Storn R. and Price, K. (1996) "Minimizing the real functions of the ICEC'96 [11]contest by Differential Evolution", Int. Conf. on Evolutionary Computation, Nagoya, Japan, pp. 842-844."
- [12]Storn R, & Price, K. (1997). "Differential Evolution—A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces", Journal of Global Optimization, 11, 341-359.
- [13]Zadeh, L.A. (1965). "Fuzzy Sets", Information and Control vol. 8, pp. 338-353.
- [14]Zadeh, L.A. (1973). "Outline of a new approach to the analysis of complex systems and decision processes", IEEE Transactions on Systems, Man and Cybernetics, SMC-3(1), 28-44.
- [15]Zadeh, L.A. (1975). "The concept of a linguistic variable and its application to approximate reasoning", Information Sciences, Part I : 8, 199-249 ; Part II : 8, 301-357 ; Part III : 9,43-80
- [16]Mayer, D.G., Kinghorn, B.P., Archer, A.A. (2005). "Differential evolution - An easy and efficient evolutionary algorithm for model optimization", Agricultural System, 83(3), 315-3.

附錄

函數名稱	公式	函數範圍	最小值
Sphere	$f(x) = \sum_{i=1}^n x_i^2$	[-100, 100]	$f(0) = 0$
Rastrigin	$f(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	[-5.12, 5.12]	$f(0) = 0$
Girewank	$f(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^D \cos(\frac{x_i}{\sqrt{i}}) + 1$	[-600, 600]	$f(0) = 0$
Ackley	$f(x) = -20 \exp \left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2} \right) - \exp \left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i) \right) + 20 + \exp(1)$	[-32.768, 32.768]	$f(0) = 0$

Fuzzy Artificial Bee Colony Algorithm

Hsien-Leing Tsai¹ Xiao-Si Chen²

¹ Department of Information Management, I-Shou University, Taiwan

² Department of Information Management, I-Shou University, Taiwan

ABSTRACT

This study discussed Artificial Bee Colony Algorithm(ABC) was proposed by Karaboga in 2005. This optimization algorithm has a stable convergence rate, solving capabilities and an amount of control parameters. Though ABC has many advantages, there are still has some problems, such as the regional situation and the slow convergence. Therefore, this study refers to the research of Improved ABC, presented Fuzzy Artificial Bee Colony Algorithm(FABC). This research utilized fuzzy theory and Differential Evolution algorithm to improve the searching capability and falling into local optimal solution. In addition, the FABC is combined with GABC(FGABC), which is expected to improve the accuracy of the solution.

From the experimental results, it is show that FABC and FGABC have improved the situation of falling into the regional solution, so that it can be continuous and effective development.

Keywords: Artificial Bee Colony algorithm, Fuzzy Theory, Differential Evolution