

運用多樣性策略改良人工蜂群演算法

李維平

中原大學資管所

wplee@cycu.edu.tw

蔡宛庭

中原大學資管所

kiaiba@msn.com

摘要

人工蜂群演算法(Artificial Bee Colony Algorithm,ABC)擁有結構簡單、容易使用、及快速收斂之特性，能有效地運用於廣大的複雜問題之中，因此成為近年來新穎且正迅速發展的演化式計算。但在求解過程當中，人工蜂群演算法同樣存在演化式計算常見的問題，在區域開發能力和全域探索能力難以取得平衡，導致易過早收斂陷入區域最佳解的問題發生。本研究嘗試以多樣性(Diversity)概念改善人工蜂演算法，提出多樣性策略避免求解過程陷入區域最佳解，同時兼顧求解過程的開發和探索能力，以利提高求解的精準度

關鍵詞：人工蜂群演算法(Artificial Bee Colony Algorithm)、演化式計算(Evolutionary Computation)、多樣性(Diversity)、最佳化(Global Optimization)

1.前言

群體智慧定義為任何企圖設計演算法或解決分散式的問題，其靈感來自於昆蟲及其他動物的集體行為[15]。群體智慧和以前人工智慧相比，具有分散式控制、自我組織和適應的能力的特性[8]。在群體智慧研究著重設計出生物個體間互動規則及整體性的行為。

演化式演算法是藉以模仿生物社會行為，將社會行為轉為計算模型，利用計算模型解決複雜工程問題[3]。演化式計算包括有仿以生物進化論，適者生存不適者淘汰的基因演算法(Genetic Algorithm)；粒子群演算法(Particle Swarm Optimization)是模仿魚群及鳥群群聚行為，透過粒子本身過去飛行的經驗及個體最佳解和全域最佳解的交互影響，進而求出最佳解；螞蟻演算法(Ant Colony Optimization)是模仿螞蟻尋找食物行為，透過費洛蒙的計算求得最佳解；以及本文所研究的人工蜂群演算法(Artificial Bee Colony)[7]是模仿蜜蜂採集花蜜的行為。

於 1995 年，由 Seeley 提出模擬蜂群的自我組織模型[20]，在 2005 年，由 Karaboga 提出模擬蜜蜂採集花蜜行為的人工蜂群演算法[7]。人工蜂群演算法具有概念簡單、容易實作及較少的控制參數等優點[8,14]。其求解之優越性亦被證實[4-6,21]。雖然具有上述優點但仍存在演化式演算法常見問題，在區域開發及全域探索兩者能力難取得平衡的問題[11]。

基於以上所述，本研究提出了以多樣性策略改善人工蜂群演算法在求解過程中開發及探索的能力，藉此提高求解精準度。多樣性策略是以解向量在解空間分散情形作為指導演算法的開發和探索，有助於平衡演算法的開發及探索能力，使得在演化過程可以加速收斂但又能避免過早收斂的發生。

2. 文獻探討

2.1 人工蜂群演算法

Karaboga 於 2005 年提出人工蜂群演算法[7]。於 2006 年將蜜蜂採蜜行為應用在函數的數值優化問題上[2]。在多项實驗中驗證人工蜂群演算法求解的優越效能[5,6]，人工蜂群演算法成為近年來熱門的演算法之一。

人工蜂群演算法是由食物源及被雇用的工蜂(Employed Bees)及未被雇用的觀察蜂(Onlookers)和偵察蜂(Scouts)組成的自我組織模式，彼此分工合作實現群體智慧。食物源在人工蜂群演算法代表最佳化的可行問題解，食物源透過適應函數計算出花蜜量價值，花蜜量價值為適應值。演算法首先隨機初始化 SN 個食物源位置。每一個可行解 x_i ($i=1,2, \dots, SN$) 為解向量， D 代表維度即指解向量的大小。在初始化後，進行工蜂、觀察蜂及偵察蜂搜尋食物源，直到滿足最大迭代數。人工蜂群演算法整體流程圖，如圖 1 所示：

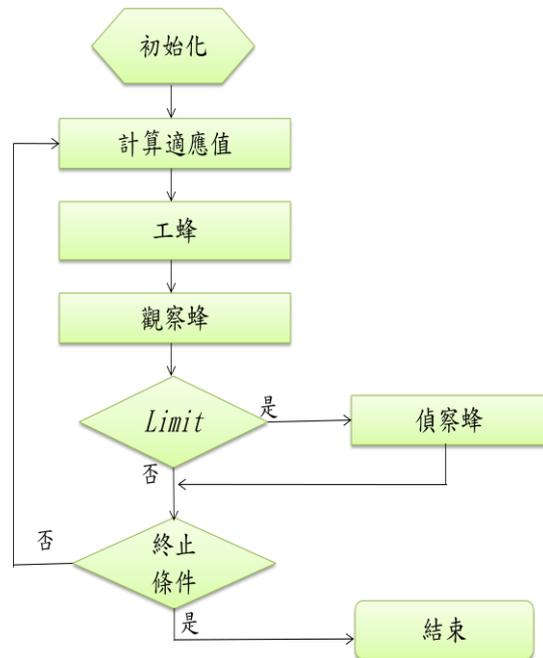


圖 1 人工蜂群演算法流程圖

人工蜂群演算法中有三個控制參數， SN 是設定食物源的數量； MCN 是設定演化的最大迭代數； $Limit$ 為食物源未改善次數的極限值，演化過程中食物源未能在附近找到更好的食物源的次數達到 $Limit$ 時，則工蜂會放棄食物源轉變為偵察蜂重新開始隨機尋找新的食物源。

觀察蜂選擇食物源是透過輪盤法方式計算機率，如公式(1)， fit_i 為全部食物源數量中第 i 個食物源的適應值， fit_n 為總適應值，機率值較高被選擇的機會亦較高。

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

工蜂尋找食物源和觀察蜂選擇食物源後，在食物源附近尋找新的食物源方式如公(2)， $k \in \{1,2, \dots, SN\}$ 和 $j \in \{1,2, \dots, D\}$ ， k 和 j 為隨機選擇， v_{ij} 為新的食物源； ϕ_{ij} 為在 $[-1,1]$ 的隨機亂數； x_{kj} 為隨機挑選食物源的位置。新的食物源 v_{ij} 透過選擇兩個食物源 x_{ij} 及 x_{kj} 的差值乘上亂數 ϕ_{ij} ，在 x_{ij} 產生擾動的方式，作為在該食物源 x_{ij} 於鄰近尋找新食物源的方法。並計算新的食物源適應值，若新的食物源適應值優於原有的，則選擇新的食物源，反之，則保留原有的食物源

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) \quad (2)$$

當食物源未能改善的情形達到 $Limit$ 時，工蜂會放棄該食物源，轉變為偵察蜂重新尋找食物源，計算新的食物源適應值，若新的食物源適應值優於原有的，則選擇新的食物源，反之，則保留原有的食物源，重新尋找食物源的方法如公式(3)， x_{min}^j 為解空間的下邊界(lower borders)， x_{max}^j 為解空間的上邊界(upper borders)。

$$x_i^j = x_{min}^j + rand[0,1](x_{max}^j - x_{min}^j) \quad (3)$$

2.2 多樣性

多樣性為解向量在解空間的散佈情形，較高的多樣性代表解向量散佈在解空間的範圍較大，因為彼此的差異性較大，較能增加解向量的變動情形。反之，較低的多樣性，代表解向量散佈在解空間範圍比較小，解向量彼此相似度高，較難增加解向量的變動情形，導致演算法容易發生過早收斂的情形。其方法有助於演算法加速收斂，但又可以避免過早收斂的情形，藉由多樣性的方法有助於平衡演算法開發及探索的能力。

Ursem 在 2002 年，提出了以多樣性方法指導演化式演算法，以多樣性方法作為指導演算法的開發及探索的能力[19]，衡量多樣性的方法是以距離至平均點的方式為基礎，計算解向量在每迭代下散佈在解空間的範圍，多樣性的計算如公式(4)所示。

$$diversity(P) = \frac{1}{|L| \cdot |P|} \cdot \sum_{i=1}^{|P|} \sqrt{\sum_{j=1}^N (s_{ij} - \bar{s}_j)^2} \quad (4)$$

$|L|$ 為解向量在搜尋範圍的對角線； P 為族群， $|P|$ 為族群規模； N 為解向量的維度； s_{ij} 為第 i 個解向量的 j 維度； \bar{s}_j 為所有解向量 j 維度的平均點，求得解向量到平均點的距離，再乘上 $|P|$ 的倒數計算平均距離，最後乘上 $|L|$ 解向量在解空間範圍的對角線，代表解向量在解空間散佈的範圍。多樣性 $diversity(P)$ 的值越大代表解向量在解空間散佈的範圍較廣；多樣性 $diversity(P)$ 的值越小代表解向量在解空間散佈的範圍較小。

首先設定 d_{low} 和 d_{high} 作為判斷多樣性 $diversity(P)$ 下限值和為上限值。當多樣性 $diversity(P)$ 比 d_{high} 大，代表解向量在該迭代下散佈範圍太廣，則演算法進入開發階段，加速收斂縮小解向量在解空間散佈範圍，促使多樣性的降低，開發的方法為演化式演算法的選擇及交換機制。反之，當多樣性 $diversity(P)$ 比 d_{low} 小，代表解向量在該迭代下散佈範圍太小，則演算法進入探索階段，將解向量在解空間的範圍擴大，以利提高多樣性，探索的方法為演化式演算法的突變機制。

Riget 及 Vesterstr，於 2002 年將多樣性方法應用於粒子群演算法，利用多樣性方法交替執行吸引及排斥階段[13]。當多樣性 $diversity(P)$ 比 d_{high} 大時，演算法進入吸引階段，使用經典粒子群演算法的速度及更新公式；當多樣性 $diversity(P)$ 比 d_{low} 小時，進入排斥階段，透過操作經典粒子群的速度公式的運算子，提高多樣性。

Pant, Radha 和 Singh 於 2008 年以多樣性方法應用於粒子群演算法利用多樣性方法執行三階段[16]，將 Riget 及 Vesterstr 的吸引及排斥二階段，變更為三階段，第三階段為介於吸引及排斥之間，強調個體最佳解經驗的影響。Jie 及 Zeng 於在 2007 年將多樣性應用於粒子群演算法，利用線性遞減概念修改多樣性判斷條件[12]。Niu, Chen 及 Zeng 在 2010 年，將多樣性應用於粒子群演算法，修改多樣性 $diversity(P)$ 的判斷條件 d_{low} 、 d_{high} ，並提出突變策略加快收斂速度[10]。

3.研究方法設計

本研究提出運用多樣性策略改良人工蜂群演算法(DABC)，多樣性策略有主要三

個部分，一為多樣性 $diversity(p)$ ，計算解向量的散佈情形的方法；其次判斷條件，用來衡量多樣性 $diversity(P)$ ；最後為開發及探索的子方法。演算法在初始化解向量後，依經典人工蜂群演算法的主要流程演化，演化後利用多樣性策略依解向量在解空間分散情形進行擾動。多樣性之人工蜂群演算法架構，如圖 2 所示。

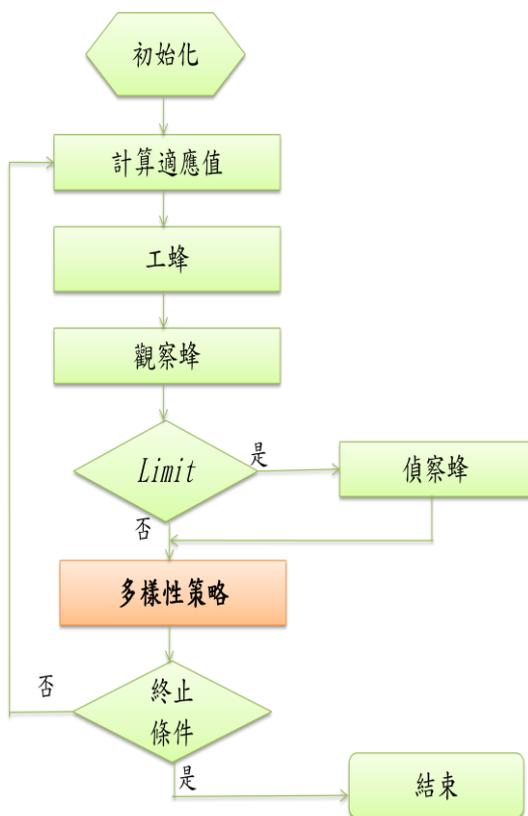


圖 2 DABC 架構圖

多樣性 $diversity(P)$ 的計算是以 Ursem 的方法為基礎，如公式(4)所示。在人工蜂群演算法中，族群規模不等於解向量數量，族群由工蜂及觀察蜂組成，食物源數量為族群的一半。因此調整參數設定，將公式(4)的族群規模 $|P|$ 變更為食物源個數 $|SN|$ ，修正後如公式(5)所示。

$$diversity(P) = \frac{1}{|L| \cdot |SN|} \cdot \sum_{i=1}^{|P|} \sqrt{\sum_{j=1}^N (s_{ij} - \bar{s}_j)^2} \quad (5)$$

判斷條件是作衡量多樣性 $diversity(P)$ 的依據，Ursem 學者採用固定值 d_{low} 、 d_{high} ，採用固定值的方法，在設定上較為困難，難以制定出一個適用於各種不同的環境，故本研究將以動態值為判斷條件。Jie 及 Zeng 提出的線性遞減的動態方法，隨著迭代增加在值域範圍內逐步遞減，此方法在面對不同環境下更為彈性靈活。但由於線性遞減的斜度設定較陡即值域範圍較大，易造成判斷上一分為二，初期為執行探索，後期為執行開發，導致失去線性遞減的優點。反之，其斜度較平坦即值域範圍較小，隨著迭代增加逐步遞減幅度較小，易造成開發及探索方法交互執行程度高，導致演算法太過隨機性，失去多樣性策略目的，故本研究提出以曲線動態為判斷條件。

曲線動態方式相較線性遞減更適用於不同問題上，使得在判斷上初期以探索為主，開發為輔；後期以開發為主，探索為輔，藉此促使演算法加快收斂的同時亦避免陷入區域最佳解的情形發生。曲線的計算法如公式(6)(7)所示。首先以公式(6)方式求出曲線 $Cur(t)$ ， $iter$ 為目前迭代數、 $MAXITER$ 為最大迭代數、 $w_{initial}$ 為 0.5、 w_{final} 為 0.05。接著透過公式(7)針對 $Cur(t)$ 隨機小幅度的變動，目的為提高曲線的隨機性， $r1$ 為 [0-0.05] 隨機數、 $r2$ 為 [0-1] 隨機數。 $r1$ 為變動程度，當 $r2$ 大於 0.5 則增加 $Cur(t)$ ；當 $r2$ 小於等於 0.5 則減少 $Cur(t)$ 。

$$Cur(t) = \exp\left(-3 \cdot \left(\frac{iter}{MAXITER}\right)^2\right) * (w_{initial} - w_{final}) + w_{initial} \quad (6)$$

$$RandCur(t) = \begin{cases} Cut(t) + r1, & \text{if}(r2 > 0.5) \\ Cut(t) - r1, & \text{if}(r2 \leq 0.5) \end{cases} \quad (7)$$

當多樣性 $diversity(P)$ 大過判斷條件 $RandCur(t)$ ，演算法進入開發階段，減少多樣性；反之，當多樣性 $diversity(P)$ 小過判斷條件 $RandCur(t)$ ，則演算法進入探索階段，增加多樣性。

經典人工蜂群演算法在全域探索能力較強，故多樣性策略將著重於提高演算法的開發能力。由於經典人工蜂群演算法工蜂和觀察蜂移動方式為每一個食物源只移動一個維度，導致演算法收斂較緩慢，故在開發子方法上將改採用移動全部維度，以利演算法加快收斂。在探索子方法維持只移動一個維度的方式。

開發子方法中，因為全域最佳解的附近極有可能有更好的食物源，所以先優化全域最佳解，在鄰近範圍內移動尋找新食物源。如公式(8)所示， $v_{gbest\ d}$ 為新食物源位置； $x_{gbest\ d}$ 為全域最佳解位置； φ 為 [-0.5- 0.5] 隨機數。透過此方式將全域最佳解位置在 φ 範圍內尋找新食物源，若優化後適應值優於原來的，則選擇新食物源；反之保留原來食物源。其次，將非全域最佳解，利用全域最佳解位置移動達到收斂目的，如公式(9)， v_{id} 為新食物源位置； x_{id} 為目前食物源位置； x_{kd} 為隨機挑選另一個食物源； ω 為 [0-1] 隨機數。若移動後適應值優於原來的，則選擇新食物源；反之保留原來食物源。

$$v_{gbest\ d} = x_{gbest\ d} + \varphi \cdot x_{gbest\ d} \quad (8)$$

$$v_{id} = x_{id} + \omega(x_{gbest\ d} - x_{kd}) \quad (9)$$

在探索子方法採用差分演算法[18]中的 DE/rand/1[1] 突變公式。DE/rand/1 比人

工蜂群演算法中的工蜂和觀察蜂移動方式更為隨機性，突變能力更強，利於全域探索。其如公式(10)所示，將食物源中的一個維度，透過隨機挑選三個食物源，並乘上權重因子(F)，的方法隨機突變產生新食物源。若突變後適應值優於原來的，則選擇新食物源；反之保留原來食物源。

$$v_{id} = x_{r1d} + F(x_{r2d} - x_{r3d}) \quad (10)$$

DABC 演算法流程如下

Step1：初始化(Initialize)，食物源 x_{ij} ， $i=1,2,\dots,SN$ 。

Step2：計算適應值。

Step3：工蜂透過公式(2)尋找新的食物源 v_{ij} 並計算食物源的適應值。

Step4：使用輪盤法如公式(1)所示，計算食物源的適應值，觀察蜂選擇較高適應值的食物源。

Step5：觀察蜂透過公式(2)尋找新的食物源 v_{ij} 並計算食物源的適應值。

Step6：若食物源未改善情況達到極限值 $Limit$ ，工蜂放棄此食物源，變成偵察蜂，透過公式(3)方法探索新的食物源。

Step7：計算 $diversity(P)$ 及 $RandCur(t)$ 。

Step8：當 $diversity(P) > RandCur(t)$ ，執行開發子方法，優化全域最佳解如公式(8)所示，再利用全域最佳解方向移動如公式(9)所示。若 $diversity(P) < RandCur(t)$ ，執行探索子方法，突變方法如公式(10)所示。

Step9：記憶最佳的食物源。

Step10：未能達到停止條件則回到 Step2，若達滿足條件則輸出最佳解。

4. 實驗

4.1 參數設定

本研究比較對象為經典人工蜂群演算法，實驗設計在迭代數為 5000 下，分別實驗在食物源為 20 個，且維度各別在 10、30 和 60 維的情形下測試，展現其演算法求解精準度。且為了進行之精確性，本研究所有的測試函數都將被重複執行 30 次並求取其平均值。

表 1 參數設定

參數名稱	設定值
SN(解向量數)	20
MCN(迭代數)	5000
Limit	100
Upper_bound	依測試函數設定
Low_bound	依測試函數設定
Dim(維度)	10、30、60
Times(執行次數)	30

4.2 測試函數

本研究為驗證此一架構下演算法之有效性，將使用學者公認之評估測試函數進行效能檢測，此四類函數為最經典也是最為基本之測試函數，其最佳解均為 0，在求解範圍分別設定如下，Sphere 及 Rosenbrock 求解範圍為 [-100,100]，Rastrigin 求解範圍[-5.12,5.12]，Griewank 求解範圍為[-600,600]。本研究將利用此四

個函數進行一個效能上的評估。

表 2 測試函數

函數名稱	公式
$f1$ -Sphere	$f_1(x) = \sum_{i=1}^n x_i^2$
$f2$ -Rosenbrock	$f_2(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$
$f3$ -Rastrigin	$f_3(x) = \sum_{i=1}^n [x_i^2 - 10\cos(2\pi x_i) + 10]$
$f4$ -Griewank	$f_4(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$

4.3 實驗結果

在不同維度下的實驗，將本研究所提出的以多樣性策略為主的改良式人工蜂群演算法與經典人工蜂群演算法進行比較。實驗結果如下：

表 3 $f1$ 函數實驗結果

$f1$	Dim.	Means	Std.	Best
ABC	10	6.9982E-17	1.90365E-17	2.64112E-17
	30	7.2318E-16	1.13221E-16	5.0539E-16
	60	7.12274E-15	8.07485E-15	2.09821E-15
DABC	10	2.99357E-17	9.91186E-18	1.46296E-17
	30	2.51057E-16	3.69282E-17	1.83223E-16
	60	7.38874E-16	1.7543E-16	3.25237E-16

表 4 f_2 函數實驗結果

f_2	Dim.	Means	Std.	Best
ABC	10	0.74616	0.81621	0.04363
	30	3.75199	5.47346	0.07172
	60	2.090696	68597	0.02393
DABC	10	1.46964 E-5	2.15032 E-5	2.93951E-9
	30	0.05833	0.08896	9.01898E-6
	60	0.25785	0.41376	8.58161E-4

表 5 f_3 函數實驗結果

f_3	Dim.	Means	Std.	Best
ABC	10	0.0E-307	0.0E-307	0.0E-307
	30	1.21976 E-14	3.46115E -14	0.0E-307
	60	1.70761 E-9	7.00306E -9	1.42108E -14
DABC	10	0.0E-307	0.0E-307	0.0E-307
	30	1.06581 E-15	1.34503E -15	0.0E-307
	60	8.76336 E-15	2.88839E -14	0.0E-307

表 6 f_4 函數實驗結果

f_4	Dim.	Means	Std.	Best
ABC	10	9.62193 E-17	3.77403E -17	0.0E-307
	30	8.40068 E-16	1.89462E -16	5.55111E -16
	60	6.72425 E-15	1.07994E -14	1.99841E -15
DABC	10	8.14163 E-17	4.90959E -17	0.0E-307
	30	3.55271 E-16	1.60631E -16	1.11022E -16
	60	7.17944 E-16	2.06181E -16	4.44089E -16

由本節實驗結果可以看出，其四個測試函數表現均略為突出。

實驗證明本研究提出的 DABC 演算法有助於改善求解精準度。經典人工蜂群在 Shpere、Griewank 和 Rastrigin 函數求解能力表現較好，在較複雜的 Rosenbrock 函數表現較差。透過本研究提出的多樣性策略改良後，既能優於在 Shpere、Griewank 和 Rastrigin 的求解能力，又能將經典人工蜂群在較複雜 Rosenbrock 函數求解能力較弱部分，透過多樣性策略改良後，提高求解精準度。

5. 結論

本研究提出多樣性概念應用於人工蜂群演算法中，其目的為了防止人工蜂群演算法過早收斂，導致落入區域最佳解，本研究利用多樣性策略讓問題解向量有更多機會可以搜尋新空間。以下將針對經由實驗結果所得到的結論：

1. 透過多樣性概念，可發現能有效的加強其搜尋的精準度。
2. 多樣性概念本身為以問題解向量在每一迭代的散佈情形，以此作為改良的基底有助於演算法兼顧開發及探索的能力。

本研究所提出之 DABC 演算法經由實驗證實能有效的解決複雜函數問題，其不管是在低維度或較高維度中均能有不錯的成效。目前在人工蜂群演算法改良中，也有許多相關參數的改良以及調整，本研究提出一個以多樣性概念為基礎的人工蜂群演算法，將適用於類似調整參數、架構式和策略改良之研究，未來或許可將此一演算法與其他相關改良演算法作結合，相信會有不錯的成效。

參考文獻

- [1] A.K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *Evolutionary Computation, IEEE Transactions on*, vol.13, no.2, pp. 398–417, 2009.
- [2] B. Basturk and D. Karaboga, "An artificial bee colony (ABC) algorithm for numeric function optimization," in *IEEE swarm intelligence symposium*, 2006, pp. 12–14.
- [3] D. B. Fogel, "What is evolutionary computation?," *Spectrum, IEEE*, vol.37, no.2, pp. 26–32, 2002.
- [4] D. J. Mala, V. Mohan, and M. Kamalpriya, "Automated software test optimisation framework-an artificial bee colony optimisation-based approach," *Software, IET*, vol.4, no.5, pp. 334–348, 2010.
- [5] D. Karaboga and B. Akay, "A comparative study of artificial bee colony algorithm," *Applied Mathematics and Computation*, vol.214, no.1, pp. 108–132, 2009.
- [6] D. Karaboga and B. Basturk, "On the performance of artificial bee colony (ABC) algorithm," *Applied Soft Computing*, vol.8, no.1, pp. 687–697, 2008.
- [7] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," *Techn. Rep. TR06*, Erciyes Univ. Press, Erciyes, 2005.
- [8] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm intelligence: from natural to artificial systems*. Oxford University Press, USA, 1999.
- [9] F. Kang, J. Li, H. Li, Z. Ma, and Q. Xu, "An Improved Artificial Bee Colony Algorithm," in *Intelligent Systems and Applications (ISA), 2010 2nd International Workshop on*, 2010, pp. 1–4.
- [10] G. Niu, B. Chen, and J. Zeng, "Repulsive Particle Swarm Optimization based on new diversity," in *Control and Decision Conference (CCDC), 2010 Chinese*, 2010, pp. 815–819.
- [11] G. Zhu and S. Kwong, "Gbest-Guided Artificial Bee Colony Algorithm for Numerical Function Optimization," *Applied Mathematics and Computation*, 2010.
- [12] J. Jie and J. Zeng, "Particle Swarm Optimization with Diversity-Controlled Acceleration Coefficients," in *Natural Computation, 2007. ICNC 2007. Third International Conference on*, 2007, vol.4, pp. 150–154.
- [13] J. Riget and J. S. Vesterström, "A diversity-guided particle swarm optimizer-the ARPSO," *Dept. Comput. Sci., Univ. of Aarhus, Aarhus, Denmark, Tech. Rep*, vol.2, pp. 2002, 2002.
- [14] L. Bao and J. Zeng, "Comparison and Analysis of the Selection Mechanism in the Artificial Bee Colony Algorithm," in *2009 Ninth International Conference on Hybrid Intelligent Systems*, 2009, pp. 411–416.
- [15] M. C. Schut, "Scientific Handbook for Simulation of Collective Intelligence,

version 2,” Available under Creative Commons License (2007), www.sci-sci.org.

- [16] M. Pant, T. Radha, and V. P. Singh, “A simple diversity guided particle swarm optimization,” in *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on, 2008*, pp. 3294–3299.
- [17] R. Storn and K. Price, “Minimizing the real functions of the ICEC’96 contest by differential evolution,” in *Evolutionary Computation, 1996., Proceedings of IEEE International Conference on, 2002*, pp. 842–844.
- [18] R. Storn and K. Price, “Minimizing the real functions of the ICEC’96 contest by differential evolution,” in *Evolutionary Computation, 1996., Proceedings of IEEE International Conference on, 2002*, pp. 842–844.
- [19] R. Ursem, “Diversity-guided evolutionary algorithms,” *Parallel Problem Solving from Nature—PPSN VII*, pp. 462–471, 2002.
- [20] T. D. Seeley, *The wisdom of the hive*. Harvard University Press Cambridge, MA, 1995.
- [21] Y. Marinakis, M. Marinaki, and N. Matsatsinis, “A hybrid discrete Artificial Bee Colony-GRASP algorithm for clustering,” in *Computers & Industrial Engineering, 2009. CIE 2009. International Conference on, 2009*, pp. 548–553.