

使用 Tsallis 隨機變數產生器於演化規劃的突變操作

賴宜和 鄧志堅*

大葉大學工業工程與科技管理系
51591 彰化縣大村鄉學府路 168 號

摘要

演化規劃通常使用高斯突變 (Gaussian mutation) 在非等向自我適應機制, 稱為傳統演化規劃 (classical evolutionary programming, CEP)。近年來已經有許多學者提出不同的突變操作用於演化規劃的非等向自我適應機制, 如 Yao 提出以柯西突變的自我適應機制, 稱為快速演化規劃 (fast evolutionary programming, FEP); Iwamatsu 利用 Tsallis 突變應用於演化規劃稱為廣域演化規劃 (generalized evolutionary programming, GEP)。Iwamatsu 所使用的 Tsallis 分佈的隨機變數變動的範圍比柯西分佈變動的範圍還大, 預期應比快速演化規劃效果來的更好, 然而 Iwamatsu 使用固定的尺度參數 (scale parameter) σ 為 1, 並且他使用的隨機變數產生器是 Tsallis 與 Stariolo 所提供的概估性的隨機變數產生器。因此廣域演化規劃的效果並沒有相當顯著並且結論有些誤導。針對這兩個缺點, 我們將使用共同的尺度參數 $\sigma = \sqrt{2}$ 並且用由 Deng 所提出精確的 Tsallis 隨機變數產生器產生 Tsallis 突變強度來改善。在處理 Iwamatsu 所解的 5 個問題中, 我們實驗結果指出當問題只有一個區域最佳解或許多區域最佳解時, $q=2.5$ 時的 GEP 比 CEP 優越, 但當問題具有少許最佳解時則反之。我們並且將上述的兩點修正應用於其他一個區域最佳解問題, 並且分析不同維度的時候, 目標值的收斂情形。

關鍵詞：廣域演化規劃, 尺度參數, Tsallis 隨機變數產生器

Using the Tsallis Random Number Generator as the Mutator in Evolutionary Programming

YI-HER LAI and JYHJENG DENG*

*Industrial Engineering and Technology Management Department, Da-Yeh University
168 University Rd., Dacun, Changhua 51591, Taiwan, R.O.C.*

ABSTRACT

Generalized evolutionary programming (GEP), commonly called classical evolutionary Programming (CEP), usually uses Gaussian mutation in non-isotropic self-adaptation. In recent years several researchers have used different mutation operators, such as the Cauchy mutator (by Yao) in fast evolutionary programming (FEP) and the Tsallis mutator (by Iwamatsu) in the generalized evolutionary strategy (GEP). Since the random walking distance in the Tsallis variate is much larger than that in Cauchy, Tsallis will thus render better performance in searching for a global optimum. However, Iwamatsu used a scale parameter σ of 1 and an approximate Tsallis variate generator

proposed by Tsallis and Stariolo, the application of which leads to a biased conclusion. To correct this bias, a common scale parameter σ of $\sqrt{2}$ was used in the present research, along with the exact random generator of the Tsallis distribution proposed by Deng, to investigate the performance of various adjusted q parameters. As a result of investigating five examples from Iwamatsu's report, it was concluded that, when $q=2.5$, the performance in reaching the global minimum is significantly superior to that obtained with the CEP for many multi-mode and single-mode problems but, nevertheless, inferior for a few multi-mode problems. This conclusion is slightly different from Iwamatsu's only for single-mode problems, where he claimed that CEP is better.

Key Words: generalized evolutionary programming, scale parameter, Tsallis variate generator

一、導論

演化計算 (evolutionary computation, EC) 具有簡單、容易操作及通用的特性，並且不受其他數學微分限制的條件。因此演化計算廣泛應用在工程、資訊、管理等各種領域。演化式計算，包括遺傳演算法 (genetic algorithm, GA)、演化策略 (evolution strategy, ES)、演化規劃 (evolutionary programming, EP) 與遺傳規劃 (genetic programming) [1]。以下僅就與本文有關的演化規劃和演化策略說明如后。

演化規劃由 Fogel 等人 [7] 於 1966 年提出，Fogel 以有限狀態機 (finite state machine, FSM) 的機制來解釋演化規劃的行為模式，有限狀態機是由一個起始狀態經由輸入和輸出至下一個狀態所組成，藉由改變符號的輸入輸出狀態使有限狀態機產生不一樣的行為模式，可視為演化規劃的突變機制。由於演化規劃只對於符號進行突變，因此限制了演化規劃的發展。Fogel [8] 於 1992 年使用實數參數的目標變數，其差別只在於原始的演化規劃只使用突變機制，而改進後的演化規劃除了使用突變機制還使用重組機制，這使得演化規劃在處理數值化問題已經跟演化策略很類似。

演化策略由 Rechenberg 於 1965 年所提出 [11]，演化策略的每一個體是由一組實數參數所組成，在進行重組時，對於實數參數思想到突變強度問題，突變強度需要考慮每一個體變異的分佈情形，通常使用標準常態分配來討論變異分佈的情形，而演化策略若無重組機制則可稱為演化規劃。

雖然本文的研究重點在討論突變強度的隨機變數 (特別是指 Tsallis 隨機變數) 對演化規劃績效的影響，但為了有一個更廣的全貌看法，以下我們針對演化策略作進一步的文獻探討。

二、文獻探討

關於演化策略我們分兩部分說明如下。第一是演化策略

的基本架構，第二是說明演化策略中的突變強度所使用新的隨機變數 Tsallis 分佈。

(一) 演化策略的基本架構

演化策略操作流程如下所示：

1. 初始設定

在操作演化策略前必需先設定初始群體、世代數等其它控制參數。

2. 重組

重組 (recombination) 與遺傳演算法的交配類似，都是個體間位元或實數上的交換，演化策略也具有遺傳演算法的單點交配、雙點交配、字罩交配，重組的遺傳操作發生在至少兩個或兩個以上的個體，Back 於 1996 年歸納出下列五種較常用的重組運算，如下所示 [5]：

(1) 無重組型：無重組型是直接由父代群體中直接挑選父代個體，作為子代的個體，假設選取

$A=[x_{a1}, x_{a2}, x_{a3}, x_{a4}, x_{a5}]$ 為父代個體為例，其中 A 可視為是 R^5 上的一個點，則子代個體為 $A'=[x_{a1}, x_{a2}, x_{a3}, x_{a4}, x_{a5}]=A$ 。

(2) 離散型：首先隨機選取二個父代個體如離散型所舉的 A, B 兩個例子，再隨機挑選這二個父代個體內的變數作為子代個體的變數，則子代個體的可能變數值為

$A'=[x_{b1}, x_{a2}, x_{a3}, x_{b4}, x_{b5}]$ 。

(3) 中點型：首先隨機選取二個父代個體，以 A, B 為隨機挑選的二個父代個體，再從二個父代個體中將父代個體內的變數值之平均，則新的子代個體如下所示，

$A'=[\frac{1}{2}(x_{a1} + x_{b1}), \frac{1}{2}(x_{a2} + x_{b2}), \dots, \frac{1}{2}(x_{a5} + x_{b5})]$ 。

(4) 全域離散型：與離散型類似，首先隨機選取 A, B 為隨機挑選的二個父代個體，將 A, B 個體的第 1 個變數的其中一個作為子代個體的第一個變數，再重新

挑選隨機的二個父代個體， $C=[x_{c1}, x_{c2}, x_{c3}, x_{c4}, x_{c5}]$ 、 $D=[x_{d1}, x_{d2}, x_{d3}, x_{d4}, x_{d5}]$ ，並隨機將 C 、 D 個體的第 2 個變數的其中的一個作為子代個體的第 2 個變數，則子代個體的前二個變數可能為 $B'=[x_{a1}, x_{d2}, \dots]$ ，其餘變數則依此類推。

- (5) 全域中點型：與中點型類似，首先隨機選取 A 、 B 為隨機挑選的二個父代個體，將 A 、 B 個體的第 1 個變數值的平均值作為子代個體的第 1 個變數，再重新挑選隨機的二個父代個體， $C=[x_{c1}, x_{c2}, x_{c3}, x_{c4}, x_{c5}]$ 、 $D=[x_{d1}, x_{d2}, x_{d3}, x_{d4}, x_{d5}]$ ，將 C 、 D 個體的第 2 個變數值的平均值作為子代個體的第 2 個變數，則子代個體的前二個變數可能為 $B'=[\frac{1}{2}(x_{a1} + x_{b1}), \frac{1}{2}(x_{c2} + x_{d2}), \dots]$ ，其餘變數則依此類推。

3. 自我適應機制

演化策略的目標變數是由實數參數所組成，可用來處理數值性的最佳化問題。由於目標變數是由實數所構成，因此在重組時會產生變數的突變強度問題，變數的突變強度可藉著自我適應機制調整突變強度的大小，較小的突變強度有利於細部搜尋，但不易跳脫區域最佳解，較大的突變強度容易跳脫區域最佳解，但不利於整體的搜尋 [3]。

早期的演化策略，使用固定的突變強度作為解的搜尋機制，Rechenberg [12] 於 1973 年提出的 1/5 成功規則，Rechenberg 認為在平均的情況下，每五次突變會有一次能改進目標值，可縮短接近最佳解的時間，1/5 成功規則是用來控制標準差，於演化過程中成功的突變次數與所有的突變次數的比率如果等於 1/5 則維持原來的標準差，如果大於 1/5 則增加標準差，小於 1/5 則減少標準差。1/5 成功規則隨著演化世代調整突變的強度，可視為簡單的自我適應機制。

然而，1/5 成功規則只對單一個體進行突變操作，因此 1/5 成功規則只適用於 (1+1)-ES。Schwefel [13] 於 1981 年提出 (μ, λ) -ES， μ 代表父代族群數， λ 代表子代族群數。Schwefel 依舊是以突變操作作為主要的搜尋方法，將突變強度設計成可以自我調整 (self-adaptive, SA) 的機制，突變強度仍受到機率分佈的影響，而分佈的機率由某些策略參數 (strategy parameter) 所控制。策略參數主要包含兩種： σ 與 α ， σ 為突變強度 (step size) 控制突變步寬， α 為旋轉

角度 (rotation angles) 控制突變的方向，而在非關聯性 (uncorrelated) 的自我適應機制， α 的功用不是很明顯，因此可忽略旋轉角度 [2]。在本文中，我們用 σ 代表兩種意義：第一是這裡的突變強度，第二是第肆節的尺度參數。雖然剛開始會覺得混淆，但隨著內文的敘述兩者之間的區別會非常清楚。

Schwefel [14] 於 1987 年提出關聯式 (correlated) 自我突變機制，Schwefel 希望個體的突變強度與變數座標系統相互獨立，因此考慮了旋轉角度的問題。然而關聯式的自我突變機制收斂效果較不明顯，因此關聯式的自我適應機制較不被廣泛的使用 [2]。而非關聯式的自我適應機制收斂較快、使用簡單，因此非關聯式的自我適應機制廣泛的被使用，非關聯式的自我適應機制可分為等向自我適應機制 (isotropic self-adaptation) 和非等向自我機制 (non-isotropic self-adaptation) 二種。

- (1) 等向自我適應機制：等向自我適應機制為在每一個變數使用單一的突變強度，隨著世代演化，突變強度會自我調整以增強演化效率。在公式 (1) 中，等向自我適應機制首先針對策略參數 $\sigma^{(t)}$ 進行突變，再將突變後的 $\sigma^{(t+1)}$ 針對個體的每一變數進行調整，其調整的學習參數為 τ_0 ， $i=1, 2, \dots, n$ 表示為第 i 個變數，調整後的子代個體如公式 (2) 所示。

$$\sigma_i^{(t+1)} = \sigma_i^{(t)} \exp(\tau_0 N(0,1)) \quad (1)$$

$$x_i^{(t+1)} = x_i^{(t)} + \sigma_i^{(t+1)} N_i(0,1) \quad (2)$$

策略參數 $\sigma_i^{(t)}$ 針對目標變數的每一個變數 $x_i^{(t)}$ 是固定的，因此 [4] 以一個父代為中心將每一次突變後的子代以一個圓形的區域範圍表示，如圖 1 所示。

- (2) 非等向自我適應機制：非等向自我適應機制為在每一個變數使用單一的突變強度，而每個單一突變強度對於每個變數突變進行調整，而調整方法有整體變數調整和個別變數調整，在公式 (3) 中，首先針對策略參數 $\sigma_i^{(t)}$ 進行突變，再將突變後的 $\sigma_i^{(t+1)}$ 針對個體的每一變數進行調整，對於每個變數突變的調整方法可分為調整整體變數突變的學習率 τ 和調整個別變數突變的學習率 τ' ，調整後的子代個體如公式 (4) 所示。在公式 (3) 與 (4) 中的變數 j 代表第 j 個粒子。

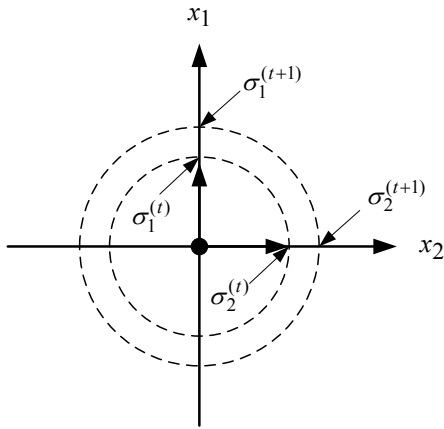


圖 1. 等向自我適應機制示意圖 [4]

$$\sigma_i^{(t+1)}(j) = \sigma_i^{(t)}(j) \exp(\tau N(0,1) + \tau' N_j(0,1)) \quad (3)$$

$$x_i^{(t+1)}(j) = x_i^{(t)}(j) + \sigma_i^{(t+1)}(j) N_j(0,1) \quad (4)$$

由於使用個別的突變強度進行調整 $\sigma_i^{(t)}$ ，而 $\sigma_i^{(t)}$ 用來調整其對應的目標變數 $x_i^{(t)}$ ，並且每一個目標變數的個別突變強度是不一致的，因此在搜尋空間上，突變後的子代將形成以父代為中心的橢圓形區域，如圖 2 所示。

早期的演化策略只使用等向自我適應機制，而等向自我適應機制使用常態機率分佈，只對於所有的變數進行常態性的調整，無法針對於個別變數進行調整，因此只有使用非等向自我適應機制才能針對於每個不同的變數使用不同的突變強度，而使用不同的機率分佈可產生不同的突變操作。在

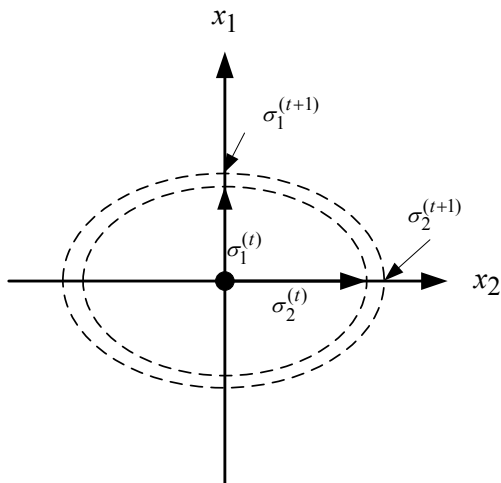


圖 2. 非等向自我適應機制示意圖 [4]

非等向自我適應機制上的突變操作可分為：

- 高斯突變操作：常態分配又稱高斯分配，在最初的演化策略都以常態分配作為個體間的突變強度，常態分配的機率密度函數如公式 (5) 所示：

$$f_G(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \quad (5)$$

使用常態分配之預計變異的範圍，稱為跳躍距離。以常態分配的累積機率密度函數計算，如公式 (6) 所示。

$$E_G(x) = 2 \int_0^{+\infty} x \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} dx = \frac{2}{\sqrt{2\pi}} = 0.8 \quad (6)$$

以高斯突變使用在演化策略的突變操作，稱為傳統演化規劃 (classical evolutionary programming, CEP)，廣泛的應用在各種的自我適應機制。傳統演化規劃子代的產生如下所示：

$$\sigma_i^{(t+1)}(j) = \sigma_i^{(t)}(j) \exp(\tau N(0,1) + \tau' N_j(0,1)) \quad (7)$$

$$x_i^{(t+1)}(j) = x_i^{(t)}(j) + \sigma_i^{(t+1)}(j) N_j(0,1) \quad (8)$$

公式 (7) 中的 $\sigma_i^{(t+1)}(j)$ 表示第 $(t+1)$ 個世代中第 i 個粒子中的第 j 個變數的突變強度。在公式 (8) 中， $N_j(0,1)$ 為在計算第 i 個粒子中的第 j 個變數時需要重新產生的常態隨機分配亂數，在公式 (7) 中， τ 為整體突變的學習率， $\tau = (2n)^{-1/2}$ ； τ' 為個體突變的學習率， $\tau' = (2n^{1/2})^{-1/2}$ 。公式 (7) 可產生高斯突變強度，再將公式 (7) 代入公式 (8) 中可產生新的子代。

- 柯西突變操作：Yao [16] 於 1999 年提出使用柯西分佈產生突變機制，稱為快速演化規劃 (fast evolutionary programming, FEP)。而柯西分配的機率密度函數如公式 (9) 所示。

$$f_C(x) = \frac{1}{\pi(1+x^2)} \quad (9)$$

以柯西分配的累積機率密度函數計算，其跳躍距離如公式 (10) 所示。

$$E_C(x) = 2 \int_0^{+\infty} x \frac{1}{\pi(1+x^2)} dx = +\infty \quad (10)$$

其第 $(t+1)$ 個子代 $x_i^{(t+1)}(j)$ 的產生如下：

$$\sigma_i^{(t+1)}(j) = \sigma_i^{(t)}(j) \exp(\tau N(0,1) + \tau' N_j(0,1)) \quad (11)$$

$$x_i^{(t+1)}(j) = x_i^{(t)}(j) + \sigma_i^{(t+1)}(j) \delta_j \quad (12)$$

在公式 (12) 中， δ_j 為柯西突變的隨機變數，在公式 (11) 中， τ 為整體突變的學習率， $\tau = (2n)^{-1/2}$ ； τ' 為個體突變的學習率， $\tau' = (2n^{1/2})^{-1/2}$ ，可產生常態性突變強度，再將公式 (11) 代入公式 (12) 中可產生新的子代。

依據 Yao 於實驗研究證明快速演化規劃比傳統演化規劃對於典型的多波谷型態函數較容易跳脫局部最佳解，但是對於單一波谷型態函數 (unimodal function) 於細部搜尋上，傳統演化策略較優於快速演化策略。在非等向自我適應機制中，目標變數 x 隨著函數的機率分佈調整突變強度，越大的突變強度則變數 x 越容易跳脫區域最佳解，因此多波谷分佈可使用能使目標變數 x 變動範圍較大的機率分佈函數，而 Tsallis 分佈的變數 x 之變動範圍又比柯西分佈來的更大，因此我們預期使用 Tsallis 分佈比柯西分佈在解多波谷分佈的問題上效果來的更好。

在有關 Tsallis 分佈應用在演化策略中，Iwamatsu [9] 利用 Tsallis 突變應用於演化策略提出廣域演化規劃 (generalized evolutionary strategy, GEP)，然而 Iwamatsu 使用固定的尺度參數為 1 (尺度參數會在第肆節介紹)，因此廣域演化規劃的效果並沒有很顯著。Tsallis [15] 則根據 Mantegna [10] 產生 Levy 穩定隨機過程 (Levy stable stochastic process) 演算法，用來模擬 Tsallis 分佈。然而當 $q_v \geq 2.4$ 時，Levy 隨機過程與理論上的 Tsallis 分佈並不一致，所以無法有效的整體模擬 Tsallis 分佈，因此 Deng [6] 於 2005 年提出的 Tsallis 隨機分佈產生器，可解決 $q_v \geq 2.4$ 時模擬 Tsallis 分佈所產生的問題，因此我們預期使用 Deng 提出的 Tsallis 隨機分佈產生器產生的突變機制應用於演化策略上，應可增加演化策略對各種不同問題的優越性。

4. 計算適應值

適應值是用來計算與評估父代及子代個體的適存程度，在最大化的問題中，可使適應值函數等於目標函數；而

在最小化的問題中，由於我們使用競賽法 (tournament method) 來選擇優質的子代所以在此我們也是以目標函數作為評估適應的指標，只是越小的目標函數值，即擁有越高的適應函數值。

5. 選擇

最初的演化策略的結構為 (1+1)-ES，(1+1)-ES 是由一個父代個體使用高斯突變操作產生一個子代個體，在父代個體與子代個體中選取較佳的適應值成為下一世代的父代個體。(1+1)-ES 存在很多弊端，比如有時無法收斂到全域最佳解以及效率較低，為了改進其效率，(1+1)-ES 增加族群大小，演變為 (μ, λ) -ES 與 $(\mu+\lambda)$ -ES 二種選擇形式。

(μ, λ) -ES 與 $(\mu+\lambda)$ -ES 是演化策略中基本的兩種選擇的形式， μ 代表父代的大小， λ 代表每一代的演化產生的子代數量，且 $\lambda > \mu$ 。 (μ, λ) -ES 與 $(\mu+\lambda)$ -ES 之間最大的差異點在於選取新的父代時， (μ, λ) -ES 是由 λ 個子代中選取最好的 μ 個作為新的父代； $(\mu+\lambda)$ -ES 是由 μ 個父代與 λ 個子代中選取最好的 μ 個作為新的父代。將這兩種天擇策略進行比較， $(\mu+\lambda)$ -ES 繼承父代的優良特性，收斂性較好，但容易陷入局部最小， (μ, λ) -ES 則較容易跳出局部最小，但由於放棄父代的結果，收斂較慢。

(二) Tsallis 分佈隨機變數產生器

Tsallis 分佈最初是由 Tsallis 與 Stariolo 於 1996 年發表 [15]，最初應用在模擬退火演算法。傳統的模擬退火演算法用波茲曼機器 (Boltzmann machine) 作為跳躍機制，Tsallis 用 Tsallis 分佈取代波茲曼機器而形成廣域退火模擬演算法 (generalized simulated annealing, GSA)。Tsallis 分佈公式如下所示：

$$g_{q_v}(\Delta x_t) = \left(\frac{q_v - 1}{\pi} \right)^{\frac{D}{2}} \times \frac{\Gamma\left(\frac{1}{q_v - 1} + \frac{D-1}{2}\right)}{\Gamma\left(\frac{1}{q_v - 1} - \frac{1}{2}\right)} \times \frac{\left[T_{q_v}^V(t) \right]^{-(D/(3-q_v))}}{\left\{ 1 + (q_v - 1) \frac{(\Delta x_t)^2}{\left[T_{q_v}^V(t) \right]^{2/(3-q_v)}} \right\}^{\frac{1}{(q_v - 1) + \frac{(D-1)}{2}}}} \quad (13)$$

在公式 (13) 中， Δx_t 為試驗的跳躍距離，從 x_t 到 x_{t+1} 的距離， $\Delta x_t = x_{t+1} - x_t$ ， D 為 Δx_t 的維度， q_v 為 [1,3) 之間的

非延伸自由度 (the degree of non-extensivity), $T_{q_v}^V(t)$ 為在時間 t 內的控制的溫度, $\Gamma(x)$ 為 Gamma 函數, 當 $D=1$, $q_v \rightarrow 1$, 公式 (13) 的機率分配函數變為常態分佈, 平均數為 0, 變異數為 $T_{q_v}^V(t)/2$, 當 $q_v=2$ 時, 機率分佈函數變為柯西分配。在 $q_v \rightarrow 1$ 的例子中, 可知標準常態分佈所相對的 $T_{q_v}^V(t)=2$ 。並且 Tsallis 分佈包含常態分佈和柯西分佈。

Tsallis 分佈的公式極為複雜, 在建構其隨機變數產生器的事上形成很大的障礙, 因此 Deng [6] 於 2005 年提出了化簡 Tsallis 分佈並使 Tsallis 分佈標準化, Tsallis 分佈化簡如公式 (14) 所示, D 為 1, Δx_t 為 x , $T_{q_v}^V(t)$ 為 T , Tsallis 機率分佈函數為:

$$g_{q_v}(x) = \left(\frac{q_v-1}{\pi}\right)^{\frac{1}{2}} \times \frac{\Gamma\left(\frac{1}{q_v-1}\right)}{\Gamma\left(\frac{1}{q_v-1} - \frac{1}{2}\right)} \times \frac{[T]^{(-1/(3-q_v))}}{\left\{1 + (q_v-1) \frac{(x)^2}{T^{2/(3-q_v)}}\right\}^{\frac{1}{(q_v-1)}}} \quad (14)$$

這裡只剩下 q_v 和 T 兩個變數, 我們可以更進一步簡化公式 (14) 如後。令 $y = \frac{x}{T^{3-q_v}} = g_1(x)$, 將變數 x 做線性轉

換, $g_1'(x) = \frac{1}{T^{3-q_v}}$, 因此, 標準化的 Tsallis 分佈如公式(15)

所示:

$$f_y(y) = \frac{f_x(x)}{|g_1'(x)|} = \frac{\left(\frac{q_v-1}{\pi}\right)^{\frac{1}{2}} \times \frac{\Gamma\left(\frac{1}{q_v-1}\right)}{\Gamma\left(\frac{1}{q_v-1} - \frac{1}{2}\right)} \times \frac{T^{(-1/(3-q_v))}}{\left\{1 + (q_v-1)y^2\right\}^{\frac{1}{(q_v-1)}}}}{\frac{1}{T^{1/(3-q_v)}}} = \left(\frac{q_v-1}{\pi}\right)^{\frac{1}{2}} \times \frac{\Gamma\left(\frac{1}{q_v-1}\right)}{\Gamma\left(\frac{1}{q_v-1} - \frac{1}{2}\right)} \times \frac{1}{\left\{1 + (q_v-1)y^2\right\}^{\frac{1}{(q_v-1)}}} \quad (15)$$

在公式 (15) 中, 溫度參數 T 值消失, 為標準 Tsallis 分佈, 任何的 Tsallis 分佈都可以藉著線性的轉換轉成標準的 Tsallis 分佈。而由於標準化的 Tsallis 分佈與 t 分佈極為相似, 因此 Deng 於 2005 年使用近似 t 分佈的手法模擬 Tsallis 分佈的隨機變數產生器。首先產生 2 個隨機變數 $Z \sim N(0,1)$ 及

$$V_1 \sim \text{Gamma}(\theta, k), \text{ 其中參數 } \theta=2, k = \nu - \frac{1}{2}。 \text{ 令 } \nu = \frac{1}{q_v-1},$$

因此參數 $k = \frac{1}{q_v-1} - \frac{1}{2}$, 隨機變數 V_1 的分佈變為

$$V_1 \sim \text{Gamma}\left(2, \frac{1}{q_v-1} - \frac{1}{2}\right)。 \text{ 經過 } T_1 = \frac{Z}{\sqrt{\frac{V_1}{\nu}}}$$

$T_1 = \frac{Z}{\sqrt{V_1(q_v-1)}}$, 為模擬標準 Tsallis 分佈的隨機變數產生

器。若再將 $T_1 \times T^{\frac{1}{3-q_v}}$, 則可模擬一般 Tsallis 分佈隨機變數產生器。

三、研究方法

本研究將使用未重組的非等向自我適應機制加入 Tsallis 分佈的突變操作, 稱為 Tsallis 演化規劃 (tsallis evolutionary programming, TEP), 若加入重組機制則稱為 Tsallis 演化策略 (tsallis evolutionary strategy, TES)。Tsallis 演化規劃透過 Tsallis 隨機變數產生器可以有效的模擬常態分佈、柯西分佈, TEP 操作流程如下所示:

1. 初始設定

設定 m 個初始母體族群數, 每個族群用 $j=1, 2, \dots, m$ 表示, 其族群數每一個體具有一對個實數向量 $x=[x_1, x_2, \dots, x_i, \dots, x_n]$ 和變異向量 $\sigma=[\sigma_1, \sigma_2, \dots, \sigma_i, \dots, \sigma_n]$, 每個維度的變數用 $i=1, 2, \dots, n$ 表示, 其初始 σ 設為 3.0。

2. 自我適應機制

將重組後的子代個體使用非等向自我適應機制並以 Tsallis 隨機變數產生器產生突變強度以作為調整後的子代個體, 調整後的子代個體如下式所示:

$$\sigma_i^{(t+1)}(j) = \sigma_i^{(t)}(j) \exp\left(\tau N(0,1) + \tau' N_j(0,1)\right) \quad (16)$$

$$x_i^{(t+1)}(j) = x_i^{(t)}(j) + \sigma_i^{(t+1)}(j) \eta_j \quad (17)$$

在式 (17) 中, η_j 為 Tsallis 隨機分佈產生器所產生的隨

機變數，在公式 (16) 中， τ 和 τ' 的意義如公式 (7) 和 (8) 所示。公式 (16) 可產生高斯突變強度，再將公式 (16) 代入公式 (17) 中可產生調整後的子代個體。

3. 計算適應值

我們用目標函數作為適應值函數，並以競賽法 (tournament method) 來選擇優質的子代。

4. 選擇

使用 $(\mu+\lambda)$ -ES 作為選擇母體族群數機制，從父代和子代個體選擇適應值較高的個體作為下一世代的父代個體。

5. 終止條件

若未達到演化世代數則重複 2 至 5 的步驟，當演化策略達到所設定的最大演化世代數，則終止演化循環。

整個 TEP 的操作流程可用圖 3 的流程圖表示。

本研究的最大貢獻在於提出一個尺度參數 σ 與 q 值所相對應的溫度參數 T 值之間的關係，我們說明如下。

Iwamatsu [9] 於 2002 年提出利用 Tsallis 突變應用於演化規劃，稱為廣域演化規劃，Iwamatsu 認為初始群體中的 $\sigma_i^{(1)}$ 固定為 3.0，並且認為 Tsallis 分佈中的尺度參數 $\sigma=1$ 。然而我們經過研究認為，為了比較 Tsallis 分佈和標準常態分佈以及相關的柯西分佈時，我們必須設定統一的尺度參

數。公式 (18) 描述尺度參數 σ 與參數 q 和 T 的函數關係，在此 q 為 q_v 的減縮。所謂尺度參數是指為了比較不同的分佈所設定的指標。不同分佈的績效必需在相同的指標 (即尺度參數) 下，做比較才有意義。由於 Tsallis 分佈包括高斯分佈和柯西分佈，並且一般的 Tsallis 分佈可以藉由 $y = \frac{x}{T^{1/(3-q)}}$ 的轉變為標準 Tsallis 分佈，因此 $\sigma = T^{1/(3-q)}$ 就成了一個很好的比較平台來比較高斯、柯西和 Tsallis 分佈。由於標準常態分佈所對應的尺度參數 $\sigma = \sqrt{2}$ ，因此我們可藉著公式 (18) 計算當不同的 q 值所相對應的溫度參數 T 值為何。

$$\sigma = T^{1/(3-q)} \quad (18)$$

我們解釋為何標準常態分佈所相對應的尺度參數 $\sigma = \sqrt{2}$ 。由前段可知，當 $q \rightarrow 1$ 時，Tsallis 分佈變為常態分佈，其變異數為 $T/2$ ，因此當 $q \rightarrow 1$ 、 $T=2$ 時，Tsallis 分佈變為標準常態分佈，將 $q \rightarrow 1$ 、 $T=2$ 帶入公式 (18)，我們可得 $\sigma = \sqrt{2}$ 。將尺度參數 $\sigma = \sqrt{2}$ 設定，就可以比較標準常態分佈與柯西、Tsallis 分佈在突變應用上的績效。

如果像原先 Iwamatsu 所設定將尺度參數設定為 1，則藉由公式 (18) 可知，不論 q 為何， T 都固定為 1。這樣一來，柯西分佈 ($q=2$ 、 $T=1$) 和 Tsallis 分佈 (例如 $q=2.5$ 、 $T=1$) 就僅與一般常態分佈 (即 $q=1$ 、 $T=1$ ，在此其標準差為 $\sqrt{1/2}$) 作比較，而不是與標準常態分佈作比較。由於傳統的演化規劃都是以標準常態分佈 (如公式 (8) 的 $N_j(0,1)$) 作為目標變數 $x_i^{(j)}$ 的突變，因此我們必須以尺度參數 $\sigma = \sqrt{2}$ 作為比較的基準。

四、實例驗證

我們將以 Iwamatsu 所使用 5 個測試函數來測試並以 Deng 所提出的 Tsallis 隨機變數產生器與 Iwamatsu 所提出的廣域演化規劃進行比較，族群數設為 50，以競賽法選擇我們新的父代數，進行競賽的數目為 10 個，我們將這 5 個測試函數各執行 50 次的實驗。

Iwamatsu 所使用 5 個測試函數，如表 1 所示， f_1 稱為 Sphere model，具有單一最小值的波谷型態函數， f_2 稱為 Griewank model、 f_3 為 Rastrigin model、 f_4 稱為 Ackley model

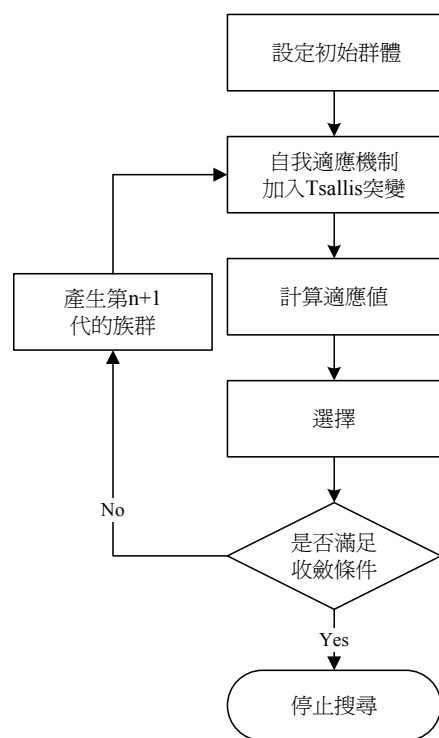


圖 3. TEP 操作流程圖

表 1. 測試函數表

Function	Minimum	Generation
$f_1(x) = \sum_{i=1}^n x_i^2$	0	1000
$f_2(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	0	5000
$f_3(x) = \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10)$	0	2000
$f_4(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e$	0	5000
$f_5(x) = -\sum_{i=1}^5 [(x_i - a_i)(x - a_i)^T + c_i]^{-1}$	-10.315	100

為具有較多的區域最小值的多波谷型態函數 (multimodal function), f_5 稱為 Schekel model 具有較少的區域最小值的多波谷型態函數, 除了在 f_5 的目標變數設為 4 個, 其他的函數的目標變數設為 10 個。

Iwamatsu 在實驗中以 Tsallis 分佈的 q 值作為測試, 參數 q 值分別設為 $q=1.0$ 、 $q=1.5$ 、 $q=2.0$ 、 $q=2.5$ 、 $q=2.9$, 尺度參數 σ 設為 1, 並未考慮 q 值與溫度參數 T 值的變動關係。

當我們使用公式 (18), 應用尺度參數 $\sigma = \sqrt{2} = T^{\frac{1}{3-q}}$ 時, q 值與溫度參數 T 值的變動關係如表 2 所示。

而 Iwamatsu 所建議的尺度參數為 1, 因此根據公式 (18) 不論 q 值為何, 溫度參數 T 值都設定為 1。在相同的尺度參數 $\sigma = \sqrt{2}$ 下, 依據表 2 所示, 不同的 q 值有不同形狀的 Tsallis 分佈, 如圖 4 所示。

在圖 4 中, 黑色的曲線代表 $q=2.9$ 、 $T=1.0353$ 的機率密度函數 (probability density function, PDF), 其中縱軸是以 \log 為尺度的, 藍色曲線代表 $q=1$, $T=2$ (標準常態分佈) 的機率密度函數, 該曲線集中在 $x=0$ 的區域由綠色曲線 ($q=1.5$, $T=1.6818$) 包住。當在同一尺度下, Tsallis 分佈隨機變數中 $q \geq 1.5$ 的機率密度函數, 其跳躍距離的幅度比標準常態分佈

表 2. 當 $\sigma = \sqrt{2}$ 時, q 值與溫度參數 T 值的變動關係

q 值	溫度參數 T 值
$q=1.0$	$T=2.0000$
$q=1.5$	$T=1.6818$
$q=2.0$	$T=1.4142$
$q=2.5$	$T=1.1892$
$q=2.9$	$T=1.0353$

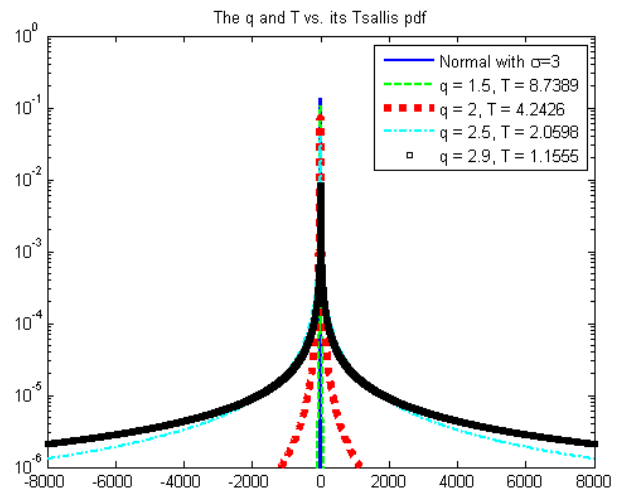


圖 4. Tsallis 隨機變數的跳躍距離

還大, 而且當 q 值越大時, Tsallis 分佈涵蓋範圍越廣, 同時隨機變數跳躍距離也越大。

Iwamatsu 經過實驗認為在具有多波谷的型態函數, 使用 $q=2.5$ 的移動函數效果比 $q=1.0$ 的傳統演化策略好。然而在 f_1 、 f_5 的單一波谷函數上, 使用 $q=1.0$ 的傳統演化策略效果較好。

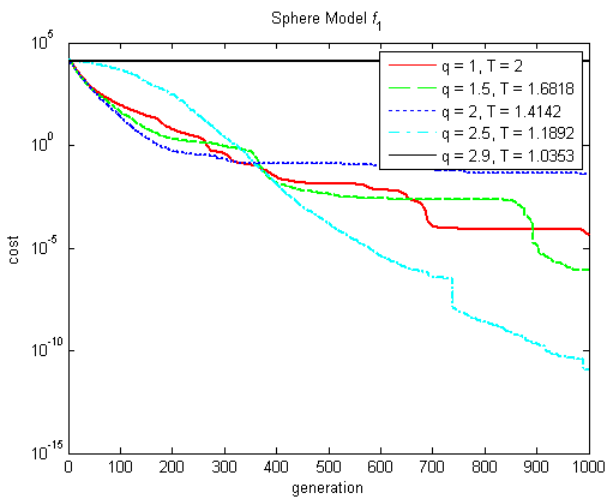
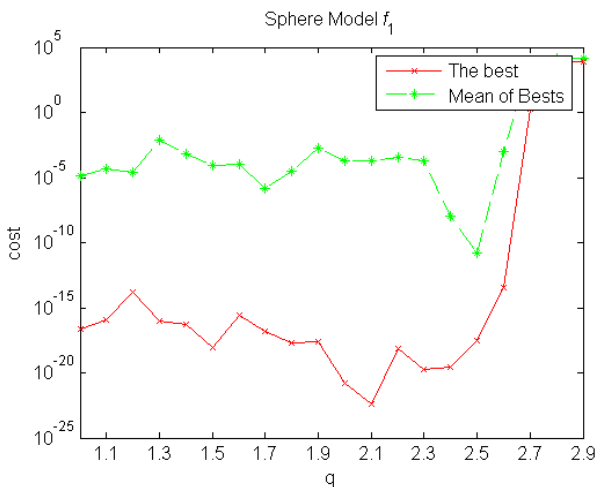
而我們使用 Iwamatsu 所提出的 5 個測試函數與我們所提出的方法進行比較, 如表 3 所示。在此我們把 Iwamatsu 所作的結果和我們所作的結果放在一起比較。

將 Iwamatsu 的結果放在左邊兩行由 CEP 和 GEP 表示, 而我們的結果放在右邊的兩行由 NCEP 和 NGEP 表示。其中 GEP 和 NGEP 所使用的 q 為 2.5。我們發現 $q=2.5$ 的 NGEP 在 f_1 ~ f_4 的函數問題上其實驗的效果 (平均值和標準差) 比 Iwamatsu 所提出的還佳, 但是在 f_5 問題上就略遜一籌。我們的結論是在對於簡單的單一波谷問題和複雜的多波谷問題, NGEP 在 $q=2.5$ 比 NCEP 來的優越, 但是在少量的多波谷問題, NCEP 的表現較好。然而 Iwamatsu 認為 CEP ($q=1.0$) 對於單一波谷型態問題有較優異的表現, 而我們所實驗的結果似乎與 Iwamatsu 所提出的觀點並不一致, NGEP 在 f_1 函數對不同 q 值執行結果如圖 5 和圖 6 所示。

圖 5 顯示出我們所改良的演算法在相同尺度 $\sigma = \sqrt{2}$ 下, 執行 5 個不同 q 值在處理 f_1 函數的收斂演化情形, 其中縱軸是目標函數值, 以 \log 尺度表示, 橫軸表示疊代的次數。由圖 5 可知當 $q=2.5$ 、 $T=1.1892$ 時, 執行 f_1 函數能最快達到收斂。圖 6 表示在相同尺度 $\sigma = \sqrt{2}$ 下, 20 個不同 q 值時,

表 3. 實驗結果比較表

	Generation	CEP Mean (STD)	GEP Mean (STD)	NCEP Mean (STD)	NGEP Mean (STD)
f_1	1000	8.19×10^{-9} (2.03×10^{-9})	2.14×10^{-7} (7.87×10^{-8})	1.268×10^{-5} (6.645×10^{-5})	1.451×10^{-11} (8.207×10^{-11})
f_2	5000	1.934 (3.526)	0.459 (1.062)	0.6206 (1.6806)	0.0787 (0.0425)
f_3	2000	3.542 (3.061)	1.782 (1.819)	28.078 (10.093)	0.8169 (0.9775)
f_4	5000	13.93 (5.582)	10.65 (5.388)	2.8521 (2.1023)	0.7604 (3.7634)
f_5	100	-6.868 (3.093)	-5.832 (3.002)	-6.8379 (3.3575)	-1.5666 (1.5807)

圖 5. NGEP 在 f_1 函數對不同尺度下 q 值的收斂效果圖 6. NGEP 在 f_1 函數的執行結果

執行 f_1 函數 50 次時，最後結果的小結，其縱軸是目標函數，以 \log 尺度表示，橫軸表示不同的 q 值，在不同的 q 值顯示 2 個結果，50 次執行的最佳值（由紅色線表示）和 50 次執行的平均值（由綠色線表示）。圖 6 的結果也顯示出當 $q=2.5$ 時，以我們提出的演算法執行 f_1 函數，其執行 50 次的平均最佳值也為最低，因此由圖 5 和圖 6 可以看出當 $q=2.5$ 時，其平均效能都要比 $q=1.0$ 的結果來的好，造成這個差異的原因可能是兩者在尺度參數 σ 的判斷上以及是否使用正確的 Tsallis 隨機變數產生器。

然而 Iwamatsu 所使用 5 個測試函數中只有 f_1 問題為單一波谷型態，上述 $q=2.5$ 的結論是否適合其他單一波谷問題並不明確。為了深入了解 $q=2.5$ 時，是否在其他單一波谷型態的問題也表現一樣優越，因此我們將測試 Yao 所使用的 7 個單一波谷型態的測試函數以彌補測試函數數目上的不足。Yao [16] 所使用的 7 個單一波谷型態的測試函數如表 4 所示。

Yao 所使用的 7 個單一波谷型態的測試函數中， $f_1 \sim f_5$ 是具有一個波谷的函數，稱為單一波谷函數（unimodal function）； f_6 是步階函數（step function），有一個最小值且不連續的函數， f_7 是噪音四次方函數（quartic function with noise），其中 $\text{random}[0,1]$ 為 0~1 之間的均一分配的隨機變數。然而在執行 f_1 函數時，因為 f_1 函數的數學方程式有 0~1 的隨機值產生，這使得每一次執行函數所產生的圖形都不相同，使 f_1 函數無法在同一標準進行 GEP 和 CEP 的比較，因此我們只使用函數 $f_1 \sim f_6$ 進行比較。

而 Iwamatsu 與 Yao 在之前的研究認為 $\text{CEP}(q=1.0)$ 在執行單一波谷型態的函數較為優越，為了驗證 Iwamatsu 與

表 4.7 個單一波谷型態測試函數

Test Function	n	S	Fmin	Generation
$f_1(x) = \sum_{i=1}^n x_i^2$	30	$[-100,100]^n$	0	1500
$f_2(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	30	$[-10,10]^n$	0	2000
$f_3(x) = \sum_{i=1}^n (\sum_{j=1}^i x_j)^2$	30	$[-100,100]^n$	0	5000
$f_4(x) = \text{Max} \{ x_i , 1 \leq i \leq n \}$	30	$[-100,100]^n$	0	5000
$f_5(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30	$[-30,30]^n$	0	20000
$f_6(x) = \sum_{i=1}^n (\lfloor x_i + 0.5 \rfloor)^2$	30	$[-100,100]^n$	0	1500
$f_7(x) = \sum_{i=1}^n ix_i^4 + \text{random}[0,1)$	30	$[-1.28,1.28]^n$	0	3000

Yao 的推論是否相符，我們使用與 Iwamatsu 相同的設定，族群數設為 50，以競賽法選擇我們新的父代數，進行競賽的數目為 10 個，尺度參數 σ 設為 $\sqrt{2}$ ，將 q 值分別設為 $q=1.0$ 、 $q=1.5$ 、 $q=2.0$ 、 $q=2.5$ 、 $q=2.9$ ，各執行 50 次，將平均期望值與平均標準差進行紀錄，並將變數分組分別設為 10 個和 30 個進行比較，以驗證其結果。我們將實驗結果以折線圖的方式顯示，如圖 7 至圖 18。

從圖 7 至圖 18 說明函數 f_1 到 f_6 在獨立變數的維度是 10 和 30，分別在 50 次的實驗當中的最低解以及平均的最低解。在 50 次的實驗當中，函數的最低解是由紅色的線來表示，而 50 次實驗中最佳解的平均值由綠色線來表示。每一個圖形的縱軸是由 log 尺度中基數為 10 的函數來表示。在執行不同的函數當中，最佳值的平均值（綠色線來表示）和最佳值的最佳值（紅色線表示）這些曲線會在 q 值不同時有不同的變化。例如當函數 f_1 在變數的維度是 10 所執行的結果（如圖 7 所示）指出當 $q=2.6$ 時，其最佳值的平均值會達

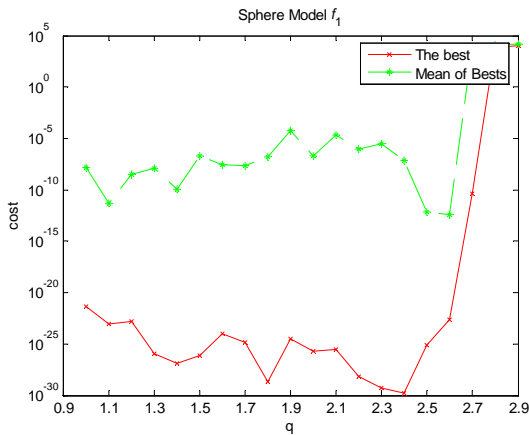


圖 7. 函數 f_1 的結果 (10 變數)

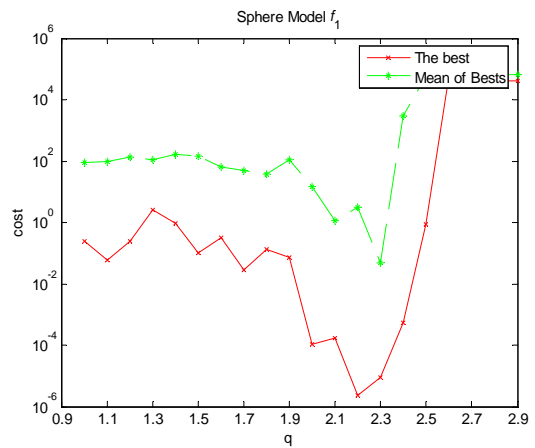


圖 8. 函數 f_1 的結果 (30 變數)

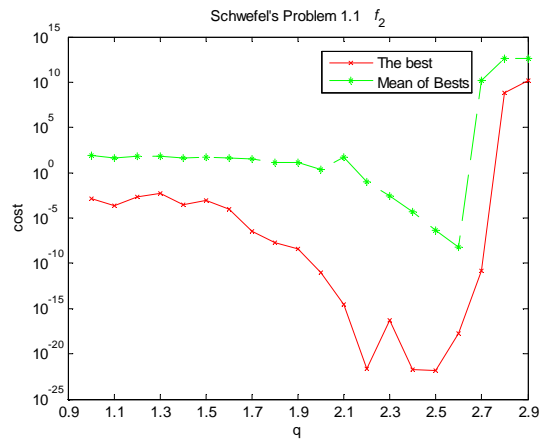
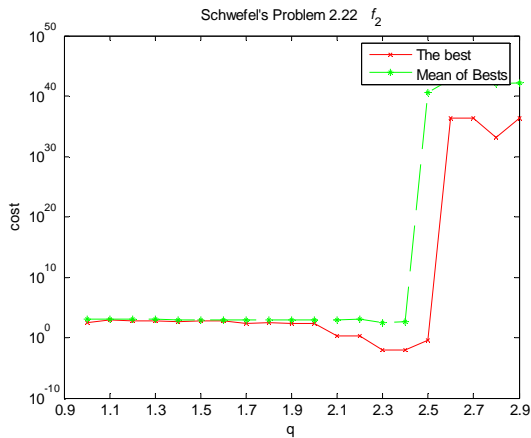
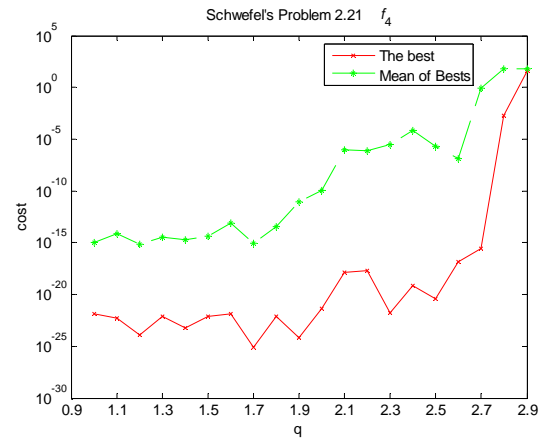
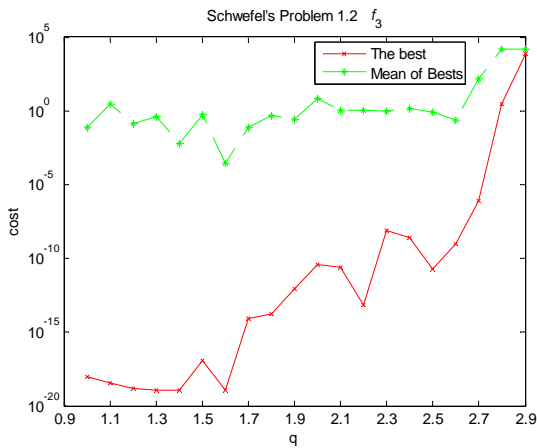
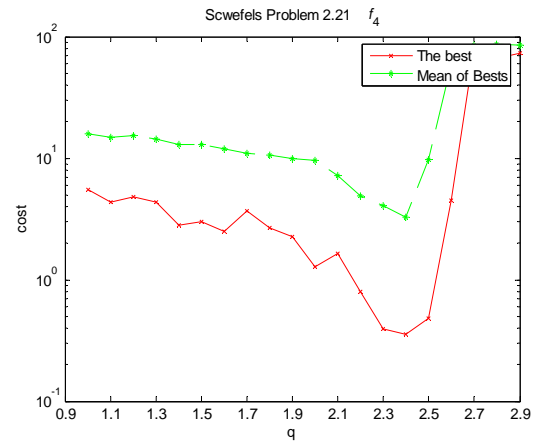
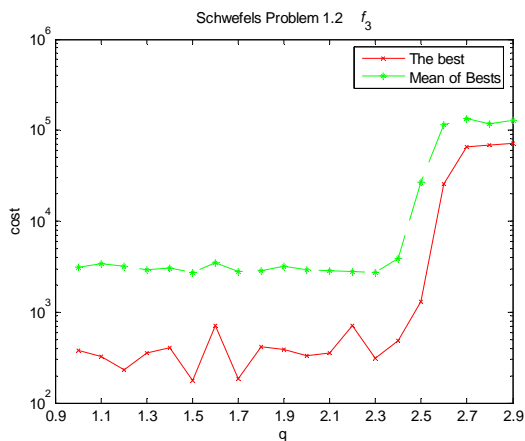
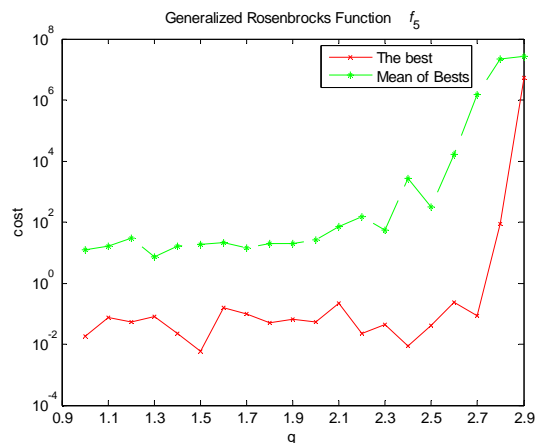


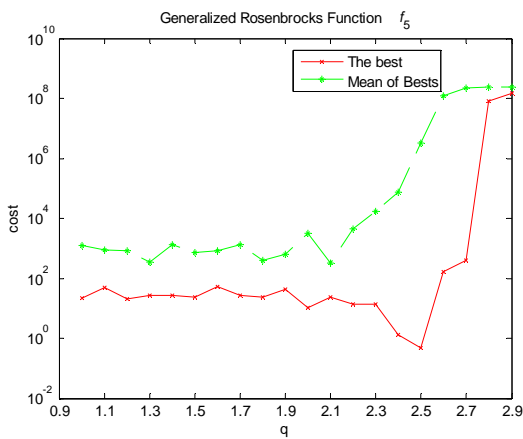
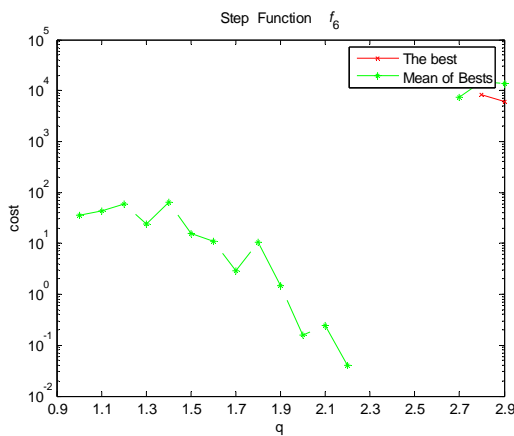
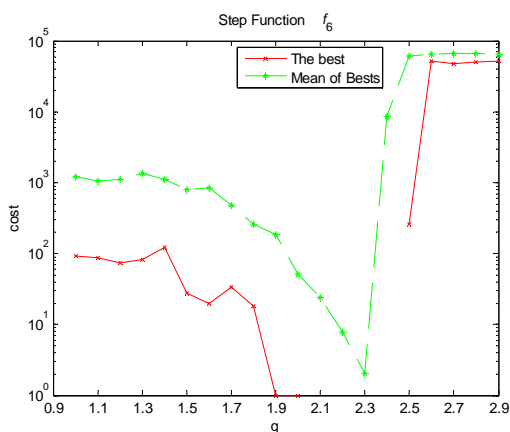
圖 9. 函數 f_2 的結果 (10 變數)

到最低並且它的績效遠超過 $q=1.0$ 的結果，它們值之間的相差比例約為 10^{-5} （這指出 $q=2.6$ 會比 $q=1.0$ 的在達到全域最佳解的平均績效優越十萬倍）。但是當函數是 f_4 時，它的獨立變數的維度是 10，最佳值的平均值與 q 值的函數關係又

圖 10. 函數 f_2 的結果 (30 變數)圖 13. 函數 f_4 的結果 (10 變數)圖 11. 函數 f_3 的結果 (10 變數)圖 14. 函數 f_4 的結果 (30 變數)圖 12. 函數 f_3 的結果 (30 變數)圖 15. 函數 f_5 的結果 (10 變數)

完全不一樣，如圖 13 所示。很明顯的， $q=1.0$ 遠優於 $q=2.5$ ，其優越的比例大約是 10^{-7} ，這說明只要函數的不一樣，雖然它們具有單一最佳值的特性，獨立變數的維度也相同，但是

所得的最佳值與 q 值之間的函數關係就會不一樣，這與 Wolpert [17] 學者提出沒有免費午餐 (no free lunch theorem) 的理論是一致的，Wolpert 的理論指出沒有一個演算法是可

圖 16. 函數 f_5 的結果 (30 變數)圖 17. 函數 f_6 的結果 (10 變數)圖 18. 函數 f_6 的結果 (30 變數)

以在所有不同問題中的績效都是最佳的，例如演算法 A 在問題 1 上表現的比演算法 B 還優越，若將問題 1 置換為問題 2 時，演算法 B 則可能表現的比演算法 A 還優越。

因此我們將參數設定值設為 $q=2.5$ ，它可以在一些函數表現優越（例如在函數 f_1 上），但是這樣的參數設定值卻會在其他的函數問題的最佳化上表現的較差（例如函數 f_4 ），因此不同的函數就需要不同的最佳化參數設定，才能使該函數搜尋的結果達到最好。在此我們使用變數函數維度為 10 來說明，但是當維度改變時（例如維度為 30），函數的最佳化結果和參數 q 值的關係也有類似的結果。不但如此，相同的函數在不同的維度上也有可能發生完全不一樣的結果，雖然在大部份的情況下，函數一樣維度不一樣的時候，函數的最佳化結果與參數 q 值的關係所形成的型態大約是相同的，例如函數 f_1 和 f_3 都是如此，但是在函數 f_3 和 f_4 就有很大的差異。

在函數 f_4 的例子中，當維度是 10 的時候，本來最佳值的平均值是隨著參數 q 的增加而增加，但是當維度是 30 的時候，整個曲線是往下走，直到 $q=2.4$ 時達到最低，之後再往上拉高。這樣的結果也可以用『沒有免費午餐』的理論來說明。最後在圖 7 和圖 18 當中，有斷線的現象發生，這是指明當參數 q 值等於某些值時，函數最佳解的平均值（綠色線表示）或是最佳解的最佳解（紅色線表示）已經達到理論的最佳解 0，由於 0 無法使用 \log 尺度來表示，因此發生斷線現象，所以在圖 17 中，函數 f_6 的執行結果顯示出當 $q=2.3\sim 2.6$ 時，在 50 次的實驗中每一次都達到最佳解 0，因此在這個區域當中綠色線是中斷的。另外不論何種函數，在哪一種獨立變數的維度上， $q=2.9$ 表現總是最差的。這是因為 $q=2.9$ 時，它的隨機變數幾乎不可能做小幅度的搜尋。

因此我們可以看出在 10 變數的單一波谷型態函數的問題上，其平均效果並非如之前的研究所認為參數 $q=1.0$ 為最佳，當 $q=2.5$ 時，在函數 f_1 、 f_2 和 f_6 的問題上仍有不錯的表現。而在 30 變數的單一波谷型態函數問題上，當參數 $q=2.3$ 時，其結果通常較佳。

五、結論

在許多演化計算中，演化策略適用於數值分析，而無重組性質的演化規劃也可使用在求實數解上，演化規劃的非等向自我適應機制依照機率分佈的變異強度進行自我調整，因此選擇使用何種機率分佈將對問題的收斂性造成重大的影響，我們透過 Tsallis 隨機變數產生器產生的突變強度應用在演化規劃上，相對於傳統演化規劃或快速演化規劃，使用 Tsallis 隨機變數產生器可藉著控制參數 q 值與溫度參數 T 值

針對問題的複雜度進行調整。

雖然 Iwamatsu 已提出利用 Tsallis 突變應用於演化規劃，稱為廣域演化規劃，然而 Iwamatsu 所使用的隨機變數產生器所產生的隨機變數為一個近似 Tsallis 分佈的隨機變數較不精準，而且並沒有考慮到參數 q 值與溫度參數 T 值之間的關係，而我們使用更精準的 Tsallis 隨機變數產生器應用於演化規劃上，並且考慮參數 q 值與溫度參數 T 值之間的關係，使 Tsallis 隨機變數產生器所產生的 Tsallis 分佈能夠符合與標準常態分佈相同的尺度規模，這使得我們所改良的演算法比 Iwamatsu 所提出的演算法在表現上更為優異。

公式 (18) 描述尺度參數 σ 與參數 q 和 T 的函數關係，所謂尺度參數是指為了比較不同的分佈所設定的指標。不同分佈的績效必需在相同的指標（即尺度參數）下，做比較才有意義。由於 Tsallis 分佈包括高斯分佈和柯西分佈，並且一般的 Tsallis 分佈可以藉由 $y = \frac{x}{T^{1/(3-q)}}$ 的轉變為標準 Tsallis 分佈，因此 $\sigma = T^{1/(3-q)}$ 就成了一個很好的比較平台來比較高斯、柯西和 Tsallis 分佈。我們解釋為何標準常態分佈所相對應的尺度參數 $\sigma = \sqrt{2}$ 。當 $q \rightarrow 1$ 時，Tsallis 分佈變為常態分佈，其變異數為 $T/2$ ，因此當 $q \rightarrow 1$ 、 $T=2$ 時，Tsallis 分佈變為標準常態分佈，將 $q \rightarrow 1$ 、 $T=2$ 帶入公式 (18)，我們可得 $\sigma = \sqrt{2}$ 。將尺度參數 $\sigma = \sqrt{2}$ 設定，就可以比較標準常態分佈與柯西、Tsallis 分佈在突變應用上的績效。

我們以 Iwamatsu 所使用的測試函數進行測試，發現在進行單一波谷型態的測試函數上所得出的結果並未如之前學者認為的傳統演化規 ($q=1.0$) 在單一波谷型態函數上表現較好，因此我們再更進一步的測試 Yao 所使用的單一波谷型態的 6 個測試函數，以不同維度進行測試，發現即使是相同的單一波谷函數問題，當變數的維度增加函數的複雜程度也隨之增加，使得合適的 q 值隨著函數問題變動。然而，當函數問題趨於複雜時（例如 30 變數的單一波谷測試函數問題），使用參數 $q=2.3$ 進行測試其結果通常較佳。

只是要如何設定參數 q 值和溫度參數 T 值才是函數最佳化的參數設定，亦可或自動隨著函數的複雜程度變更參數 q 值和溫度參數 T 值的設定，目前似乎還沒有有效的解決方法。函數的型態相當複雜，似乎無法歸類成適合哪一類型的演化計算，這也是日後研究上的重要課題。

參考文獻

1. 吳心怡 (民 95)，運用多目標模擬最佳化演算法求解生產控制與庫存管理問題，國立高雄第一科技大學運籌管理系研究所碩士論文。
2. 陳孟成 (民 93)，求解有限制條件的多目標最佳化問題的演化策略法，國立中興大學機械工程研究所碩士論文。
3. 郭文偉 (民 94)，類梯度搜尋演算策略方法，東海大學工業工程與經營資訊研究所碩士論文。
4. 謝長泰 (民 95)，發展粒子引導式演化策略演算法以處理實數參數之全域最佳化問題，國立交通大學資訊科學與工程研究所碩士論文。
5. Bäck, T. (1996) *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*, Oxford University Press, New York.
6. Deng, J., C. Chang and Z. Yang (2005) An exact random number generator for visiting distribution in GSA. *International Journal of Simulation: Systems, Science and Technology*, 6(12-13), 54-61.
7. Fogel, L. J., A. J. Owens and M. J. Walsh (1966) *Artificial Intelligence through Simulated Evolution*, Wiley, New York, NY.
8. Fogel, D. B. (1992) *Evolving artificial intelligence*, Ph.D. thesis, University of California, San Diego, CA.
9. Iwamatsu, M. (2002) Generalized evolutionary programming with Levy-type mutation. *Computer Physics Communications*, 147, 729-732.
10. Mantegna, R. N. (1994) Fast, accurate algorithm for numerical simulation of Levy stochastic process. *Physical Review E*, 49(5), 4677-4683.
11. Rechenberg, I. (1965) *Cybernetic solution path of an experimental problem*. In: *Royal Air-craft Establishment, Library Translation*, 1122, Farnborough, Hants., UK.
12. Rechenberg, I. (1973) *Evolutions strategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*, Frommann-Holzboog, Stuttgart, Germany.
13. Schwefel, H. P. (1981) *Numerical Optimization of Computer Models*, Wiley, Chichester, UK.
14. Schwefel, H. P. (1987) Collective intelligence in evolving systems. In W. Wolff, C. J. Soeder and F. Drepper (eds), *Ecodynamics-Contributions to Theoretical Ecology*,

-
- 95-100, Springer, Berlin.
15. Tsallis, C. and D. A. Stariolo (1996) Generalizes simulate annealing. *Physical A*, 233, 395-406.
16. Yao, X., Y., Y. Liu and G. M. Lin (1999) Evolutionary programming made faster. *IEEE Transactions on Evolutionary Computation*, 3(2), 82-102.
17. Wolpert, D. H. and W. G. Macready (1997) No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1), 67-82.

收件：98.07.03 修正：99.03.22 接受：99.04.14