

# 使用 Tsallis 隨機變數產生器於演化規劃的突變操作

賴宜和

大葉大學工業工程與科技  
管理學系，學生

e-mail :

R9515032@mail.dyu.edu.tw

鄧志堅

大葉大學工業工程與科  
技管理學系，教授

e-mail :

jdeng@mail.dyu.edu.tw

## 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於演化規劃的非等向自我適應機制，如 Yao[18]提出以柯西突變的自我適應機制，稱為快速演化規劃(Fast Evolutionary Strategy, FEP)、Iwamatsu[11]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[11]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[11]使用固定的尺度參數(scale parameter)為 1，並且他使用的隨機變數產生器是 Tsallis and 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  $\sqrt{2}$  並且用由 Deng[7]所提出精確的 Tsallis 隨機變數產生器產生 Tsallis 突變強度來改善。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，當  $q=2.5$  時的 GEP 比 CEP 優越。但當問題具有少許最佳解時則反之。

**關鍵詞：**廣域演化規劃、尺度參數、精準的 Tsallis 隨機變數產生器。

## Abstract

Evolutionary programming usually uses Gaussian mutation in the non-isotropic self-adaption and is commonly named as Classical Evolutionary Programming (CEP). In recent years many researchers use different mutation operators, such as Cauchy mutator in Yao in the fast evolutionary strategy (FEP) and Tsallis mutator in Iwamatsu in the generalized evolutionary strategy (GEP). The random walk distance of Tsallis variate is much larger than the

one in Cauchy, thus will render better performance in searching the global optimum. However, Iwamatsu used the scale parameter  $\sigma$  of 1 and used an approximate Tsallis variate generator suggested by Tsallis and Stariolo. The application of these leads to a biased conclusion. To correct that we use a common scale parameter  $\sigma$  of  $\sqrt{2}$  and take the exact random generator of Tsallis distribution proposed by Deng to investigate the performance of various adjusting parameter  $q$ . Five examples from Iwamatsu's paper are investigated and the conclusion is that when  $q=2.5$ , the performance of reaching the global minimum is significantly better than the CEP for the many multi-mode problems and a single mode problem, but are inferior for the few multi-mode problem. This conclusion is slightly different from the one of Iwamatsu only in the single mode problem where he claimed that CEP is better.

**Keywords:** Generalized Evolutionary Strategy, scale parameter, exact Tsallis random variate.

## 1. 前言

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。演化計算主要包括以下三種模式：演化規劃、演化策略以及基因演算法[1]。

演化規劃(Evolutionary Programming, EP)是由 L. J. Fogel 等人[8]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。D. B.

Fogel[9]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。

基因演算法(Genetic Algorithm, GA)是由 J. H. Holland[10]於1975年提出，基因演算法是以達爾文的進化論為基礎，模擬生物界「適者生存，不適者淘汰」的演化規則，每一物種會彼此不斷競爭，只有適應性強的物種才得以生存，而每一物種演化的特徵是改變物種內基因的排列順序，來產生更優良的子代。基因演算法在處理最佳化問題時，必須先將個體內的目標變數進行編碼，然後在經由複製(reproduction)、交配(crossover)、突變(mutation)的三個主要的遺傳機制進行操作，並計算適應值以淘汰不良的個體。基因演算法在求解最佳化問題上，使用適應值函數評估解的優劣，因此不需要推導複雜的數學式，可用於不同型態的問題，然而基因演算法在處理數值化問題必須先將目標變數進行編碼，容易造成字串過多且不易跳脫局部最佳解，因此在數值分析的最佳化問題，使用演化策略比基因演算法更有效率。

演化策略(Evolutionary Strategy, ES)是由 I. Rechenberg[13]於1965年所提出，演化策略是由每一個體是由一組實數參數所組成，因此在進行重組時，對於實數參數產生突變強度問題，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。

## 2. 文獻探討

### 2.1 演化策略的基本架構

演化策略與基因演算法操作流程類似，其操作流程如下所示：

#### 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、染色體數等其他控制參數。

#### 2. 重組：

重組(recombination)與基因演算法的交配類似，都是個體間位元或實數上的交換，演化策略也具有基因演算法的單點交配、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[6]歸納出下列五種較常使用的重組運算，如下所示：

#### (a)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，則子代個體為  $A'=[x_{a1}, x_{a2}, x_{a3}, x_{a4}, x_{a5}] = A$ 。

#### (b)離散型：

首先隨機選取二個父代個體，以  $A=[x_{a1}, x_{a2}, x_{a3}, x_{a4}, x_{a5}]$ 、 $B=[x_{b1}, x_{b2}, x_{b3}, x_{b4}, x_{b5}]$  為隨機挑選的二個父代個體，再隨機挑選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  $A'=[x_{b1}, x_{a2}, x_{a3}, x_{b4}, x_{b5}]$ 。

#### (c)中點型：

首先隨機選取二個父代個體，以  $A=[x_{a1}, x_{a2}, x_{a3}, x_{a4}, x_{a5}]$ 、 $B=[x_{b1}, x_{b2}, x_{b3}, x_{b4}, x_{b5}]$  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相加再平均，則新的子代個體如下所示， $A'=[\frac{1}{2}(x_{a1}+x_{b1}), \frac{1}{2}(x_{a2}+x_{b2}), \dots, \frac{1}{2}(x_{a5}+x_{b5})]$ 。

#### (d)全域離散型：

與離散型類似，首先隨機選取  $A=[x_{a1}, x_{a2}, x_{a3}, x_{a4}, x_{a5}]$ 、 $B=[x_{b1}, x_{b2}, x_{b3}, x_{b4}, x_{b5}]$  為隨機挑選的二個父代個體，將  $A$ 、 $B$  個體的第1個變數的其中的1個作為子代個體的第1個變數，再重新挑選隨機的二個父代個體， $C=[x_{c1}, x_{c2}, x_{c3}, x_{c4}, x_{c5}]$ 、 $D=[x_{d1}, x_{d2}, x_{d3}, x_{d4}, x_{d5}]$ ，並隨機將  $C$ 、 $D$  個體的第2個變數的其中的1個作為子代個體的第2個變數，為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。

#### e.全域中點型：

與中點型類似，首先隨機選取  $A=[x_{a1}, x_{a2}, x_{a3}, x_{a4}, x_{a5}]$ 、 $B=[x_{b1}, x_{b2}, x_{b3}, x_{b4}, x_{b5}]$  為隨機挑選的二個父代個體，將  $A$ 、 $B$  個體的第1個變數值相加再平均作為子代個體的第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個變數值相加再平均作為子代個體的第2個變數，為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(Evolutionary Strategy, ES)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[4]。

早期的演化策略，使用固定的突變強度作為解的搜尋機制，I. Rechenberg[14] 於 1973 年提出的 1/5 成功規則，Rechenberg[14]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[15] 於 1981 年提出  $(\mu, \lambda)$ -ES，Schwefel[15]以  $(\mu, \lambda)$  來代表父代與子代的族群數目， $\mu$  代表父代族群數， $\lambda$  代表子代族群數。Schwefel[15]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： $\sigma$  與  $\alpha$ ， $\sigma$  為突變強度(Step Size)控制突變步寬， $\alpha$  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， $\alpha$  的功用不是很明顯，因此可忽略旋轉角度[3]。

Schwefel[16] 於 1987 年提出關聯式(Correlated)自我突變機制，Schwefel[16]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(Correlated)的自我突變機制收斂效果較為緩慢，因此關聯式的自我適應機制較不被廣泛的使用[5]。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(Uncorrelated)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，增強演化效率。在式(2.1)中，等向自我適應機制首先針對策略參數  $\sigma^{(t)}$  進行突變，在將突變後的  $\sigma^{(t+1)}$  針對個體的每一變數進行調整，其調整的學習參數為  $\tau_0$ ， $i=1,2,\dots,n$  個變數，調整後的子代個體如式(2.2)所示。

$$\sigma_i^{(t+1)} = \sigma_i^{(t)} \exp(\tau_0 N(0,1)) \quad (2.1)$$

$$x_i^{(t+1)} = x_i^{(t)} + \sigma_i^{(t+1)} N_i(0,1) \quad (2.2)$$

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，因此突變後的子代在解空間上的分布範圍，將形成一個以父代為中心的圓形區域如圖 2.1 所示。

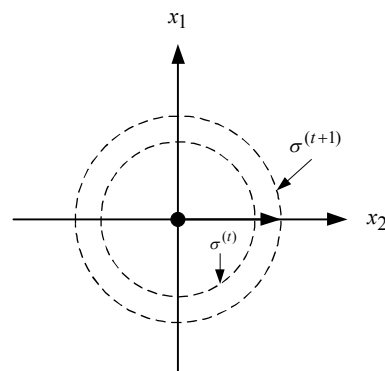


圖 2.1 等向自我適應機制示意圖[5]

#### 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，再式(2.3)中，首先針對策略參數  $\sigma_i^{(t)}$  進行突變，在將突變後的  $\sigma_i^{(t+1)}$  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  $\tau$  和調整個別變數突變的學習率  $\tau'$ ，調整後的子代個體如式(2.4)所示。

$$\sigma_i^{(t+1)} = \sigma_i^{(t)} \exp(\tau N(0,1) + \tau' N_i(0,1)) \quad (2.3)$$

$$x_i^{(t+1)} = x_i^{(t)} + \sigma_i^{(t+1)} N_i(0,1) \quad (2.4)$$

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  $x_i^{(t)}$ ，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 2.2 所示。

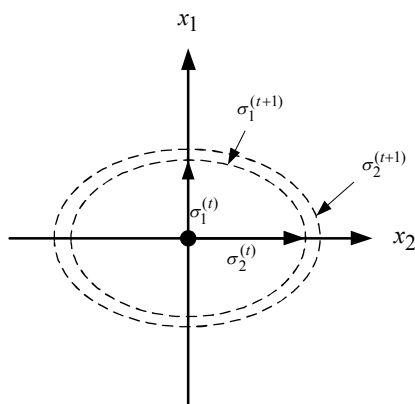


圖 2.2 非等向自我適應機制示意圖 [5]

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，在非等向自我適應機制上的突變操作可分為：

(a) 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式(2.5)所示：

$$f_G(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \quad (2.5)$$

而使用常態分配為預計變異的範圍，又稱為跳躍距離。以常態分配的累積機率密度函數計算，如式(2.6)所示。

$$E_G(x) = 2 \int_0^{+\infty} x \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} dx = \frac{2}{\sqrt{2\pi}} = 0.8 \quad (2.6)$$

以高斯突變使用在演化策略的突變操作以稱為傳統演化規劃(Classical Evolutionary Programming, CEP)廣泛的應用在各種的自我適應機制。產生傳統演化規劃子代如下所示：

$$\sigma_i^{(t+1)}(j) = \sigma_i^t(j) \exp(\tau N(0,1) + \tau' N_j(0,1)) \quad (2.7)$$

$$x_i^{(t+1)}(j) = x_i^t(j) + \sigma_i^{(t+1)}(j) N_j(0,1) \quad (2.8)$$

在式(2.8)中， $N_j(0,1)$  為在計算第  $i$  個子代中的第  $j$  個變數時需要重新產生的常態隨機分配亂數，在式(2.7)中， $\tau$  為整體突變的學習率， $\tau = (2n)^{-1/2}$ ； $\tau'$  為個體突變的學習率， $\tau' = (2n^{1/2})^{-1/2}$ 。式(2.7)可產生高斯突變強度，再將式(2.7)代入式(2.8)中可產生新的子代。

(b) 柯西突變操作

Yao[18]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式(2.9)所示。

$$f_C(x) = \frac{1}{\pi(1+x^2)} \quad (2.9)$$

而使用柯西分配為預計變異的範圍，又稱為跳躍距離。以柯西分配的累積機率密度函數計算，如式(2.10)所示。

$$E_C(x) = 2 \int_0^{+\infty} x \frac{1}{\pi(1+x^2)} dx = +\infty \quad (2.10)$$

在式(2.12)中， $\delta_j$  為柯西突變的隨機變數，在式(2.11)中， $\tau$  為整體突變的學習率， $\tau = (2n)^{-1/2}$ ； $\tau'$  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，再將式(2.11)代入式(2.12)中可產生新的子代。

$$\sigma_i^{(t+1)}(j) = \sigma_i^t(j) \exp(\tau N(0,1) + \tau' N_j(0,1)) \quad (2.11)$$

$$x_i^{(t+1)}(j) = x_i^t(j) + \sigma_i^{(t+1)}(j) \delta_j \quad (2.12)$$

依據 Yao[18]於實驗研究證明快速演化規劃比經典演化規劃對於典型的多峰型態函數較容易跳脫局部最佳解，但是對於單一波峰型態函數(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，因此多峰分佈則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峰分佈的問題上效果來的更好。

而在有關 Tsallis 分佈應用在演化策略中，

Iwamatsu[23]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[11]使用固定的突變強度為 1，因此廣域演化規劃的效果並沒有很顯著。Tsallis 則根據 R.N.Mantega[12]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[7]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[20]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適應性。

4. 計算適應值：

適應值是用來計算與評估父代個體、調整後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  $(\mu, \lambda)$ -ES 與  $(\mu + \lambda)$ -ES 二種選擇形式。

$(\mu, \lambda)$ -ES 與  $(\mu + \lambda)$ -ES[17]是演化策略中基本的兩種選擇的形式， $\mu$  代表父代的大小， $\lambda$  代表每一代的演化產生的子代數量，且  $\lambda > \mu$ 。 $(\mu, \lambda)$ -ES 與  $(\mu + \lambda)$ -ES 之間最大的差異點在於選取新的父代時， $(\mu, \lambda)$ -ES 是由  $\lambda$  個子代中選取最好的  $\mu$  個作為新的父代； $(\mu + \lambda)$ -ES 是由  $\mu$  個父代與  $\lambda$  個子代中選取最好的  $\mu$  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， $(\mu, \lambda)$ -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[2]。

2.2 Tsallis 分佈隨機變數產生器

Tsallis 分佈最初是由 Tsallis and Stariolo[17]於 1996 年發表，最初應用在模擬退火演算法以代替使用波茲曼機器(Boltzmann machine)的傳統模擬退火演算法(Classical simulated annealing, CSA)，稱為廣域退火模擬演算法(Generalized Simulated annealing, GSA)，Tsallis 分佈公式如下所示：

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

(2.13)

在式(2.13)中， $\Delta x_t$  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  $\Delta x_t$  的維度自由度， $q_v$  為 [1,3) 之間的分配參數， $T_{q_v}^V(t)$  為在時間  $t$  內確定控制的溫度， $\Gamma(x)$  為 Gamma 函數，當  $D=1$ ， $q_v \rightarrow 1$ ，(2.13) 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。

然而，Tsallis 分佈的公式極為複雜，因此 Deng[7]於 2005 年提出了化簡 Tsallis 分佈並使 Tsallis 分佈標準化，Tsallis 分佈化簡如式(2.14)所示， $D$  為 1， $\Delta x_t$  為  $x$ ， $T_{q_v}^V(t)$  為  $T$ ，Tsallis 機率分佈函數為：

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

(2.14)

由於只剩下  $q_v$  和  $T$  兩個變數，令  $y = \frac{x}{T^{\frac{1}{3-q_v}}} = g_1(x)$ ，將變數  $x$  做線性轉換， $g_1'(x) = \frac{1}{T^{\frac{1}{3-q_v}}}$ ，因此，標準化的 Tsallis 分佈如式(2.15)所示：

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

$$= \left(\frac{q_v-1}{\pi}\right)^{\frac{1}{2}} \times \frac{\Gamma\left(\frac{1}{q_v-1}\right)}{\Gamma\left(\frac{1}{q_v-1}-\frac{1}{2}\right)} \times \frac{1}{\left\{1+(q_v-1)y^2\right\}^{\frac{1}{q_v-1}}}$$

(2.15)

在式(2.15)中，當  $T=1$  時，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[7] 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  $Z \sim N(0,1)$  及  $V_1 \sim \text{Gamma}(\theta, k)$ ， $\nu$  為自由度，設  $\theta=2, k=\nu-\frac{1}{2}$  的隨機分佈，令  $\nu = \frac{1}{q_v-1}$ ，則  $k = \frac{1}{q_v-1} - \frac{1}{2}$ ，即  $V_1 \sim \text{Gamma}\left(2, \frac{1}{q_v-1} - \frac{1}{2}\right)$ ，經過  $T_1 = \frac{Z}{\sqrt{\frac{V_1}{\nu}}}$  轉

換可得  $T_1 = \frac{Z}{\sqrt{V_1(q_v-1)}}$ ，為模擬標準 Tsallis 分佈的隨機變數產生器。若再將  $T_1 \times T^{3-q_v}$ ，則可模擬一般 Tsallis 分佈隨機變數產生器。

### 3. 研究方法

本研究將使用非等向自我適應機制的演化策略加入 Tsallis 分佈的突變操作，稱為 Tsallis 演化策略 (Tsallis Evolutionary Strategy, Tsallis-ES)，並透過 Tsallis 隨機變數產生器可以有效的模擬常態分佈、柯西分佈，Tsallis-ES 操作流程如下所示：

#### 1. 初始設定：

設定  $j=1,2,\dots,m$  個初始母體族群數，其族群數每一個體具有一對  $i=1,2,\dots,n$  個實數向量  $x_i$  和變異  $\sigma_i$ ，其初始尺度參數  $\sigma$  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：

$$\sigma_i^{(t+1)}(j) = \sigma_i^t(j) \exp(\tau N(0,1) + \tau' N_j(0,1)) \quad (2.16)$$

$$x_i^{(t+1)}(j) = x_i^t(j) + \sigma_i^{(t+1)}(j) \delta_j \quad (2.17)$$

在式(2.17)中， $\delta_j$  為標準 Tsallis 隨機分佈產生器所產生的隨機變數  $T_1$ ，在式(2.16)中， $\tau$  為整體突變的學習率， $\tau = (2n)^{-1/2}$ ； $\tau'$  為個體突變的學習率， $\tau' = (2n^{1/2})^{-1/2}$ 。式(2.16)可產生高斯突變強度，再將式(2.16)代入式(2.17)中可產生調整後的子代個體。

#### 3. 計算適應值：

適應值是用來計算與評估父代個體、調整後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。

#### 4. 選擇：

使用  $(\mu + \lambda)$ -ES 作為選擇母體族群數機制，從父代個體和調整後的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。

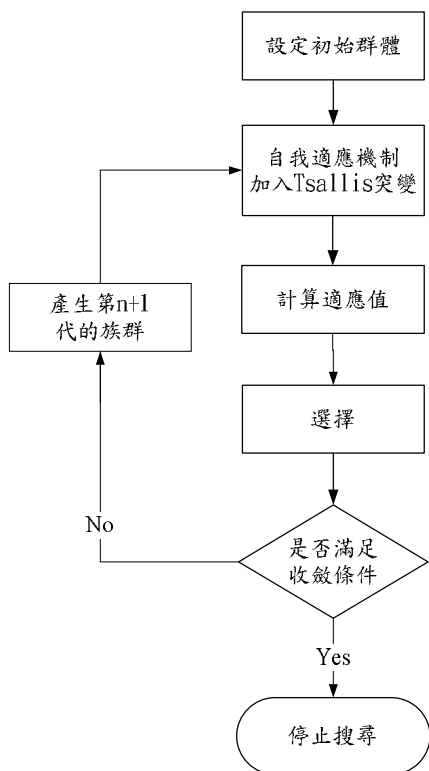


圖 2.3 Tsallis-ES 操作流程圖

#### 4. 實例驗證

Iwamatsu[11]於2002年提出利用 Tsallis 突變應用於演化策略，稱為廣域演化規劃 (Generalized Evolutionary Strategy, GEP)，Iwamatsu 認為初始群體中的  $\sigma_0^1$  固定為 3.0，並且認為 Tsallis 分佈中的  $\sigma = 1$ 。然而我們經過研究認為初始群體中的  $\sigma$  為尺度參數 (scale parameter)，並且為了比較它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。由於標準常態分佈所對應的尺度參數  $\sigma = \sqrt{2}$ ，因此我們可藉著式(2.18)計算當不同的  $q$  所相對應的溫度  $T$  為何。

$$\sigma = T^{\frac{1}{3-q}} \quad (2.18)$$

我們將以 Iwamatsu [11]所使用 5 個測試函數來測試並以 Deng[7]所提出的 Tsallis 隨機變數產生器與 Iwamatsu [11]所提出的廣域演化規劃 (Generalized Evolutionary Strategy, GEP) 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 [11]所使用 5 個測試函數，如表

4.1 所示， $f_1$  稱為 Sphere model，具有單一最小值的波峰函數 (Unimodal Function)， $f_2$  稱為 Griewank model、 $f_3$  為 Rastrigin model、 $f_4$  稱為 Ackley model 具有較多的區域最小值的多峰型態函數 (Multimodal Function)， $f_5$  稱為 Schekel model 具有較少的區域最小值的多峰型態函數，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。

表 4.1 測試函數表

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

Iwamatsu [11]在實驗中以 Tsallis 分佈的移動參數  $q$  作為測試，移動參數  $q$  分別設為  $q = 1.0$ 、 $q = 1.5$ 、 $q = 2.0$ 、 $q = 2.5$ 、 $q = 2.9$ ，尺度參數  $\sigma$  設為 1，並未考慮移動參數  $q$  與溫度參數  $T$  的變動關係。令尺度參數  $\sigma = T^{\frac{1}{3-q}}$  時，則可設定一般 Tsallis 分佈的尺度並且令  $\sigma_1$  為標準常態分佈的標準差，若尺度使用標準常態分佈時，尺度參數  $\sigma^2 = 2\sigma_1^2$  則  $\sigma = \sqrt{2}\sigma_1$ 。當  $\sigma_1$  固定為 1.0 時， $\sigma = \sqrt{2}$ ，移動參數  $q$  與溫度參數  $T$  的變動關係如表 4.2 所示：

表 4.2 當  $\sigma_1 = 1.0$  時， $q$  與  $T$  的變動關係

移動參數 $q$	溫度參數 $T$
$q = 1.0$	$T = 2$
$q = 1.5$	$T = 1.6818$
$q = 2.0$	$T = 1.4142$
$q = 2.5$	$T = 1.1892$
$q = 2.9$	$T = 1.0353$

而 Iwamatsu [11]所建議的尺度參數為 1，因此根據公式(2.18)不論移動參數  $q$  值為何，溫度參數  $T$  都設定為 1。

若  $\sigma_1 = 1.0$  時，Tsallis 分佈使用與標準常態分佈同樣的尺度，則依據表 4.2 所示，不同的移動參數  $q$  有不同形狀的 Tsallis 分佈，如圖 4.1 所示。

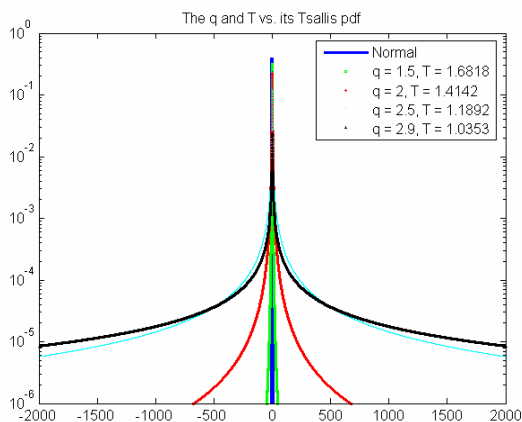


圖 4.1 Tsallis 隨機變數的跳躍距離

由圖 4.1 可看出當在同一尺度下，Tsallis 分佈隨機變數跳躍距離的幅度比標準常態分佈還大，而且當  $q$  越大時，則 Tsallis 分佈涵蓋範圍越廣則隨機變數跳躍距離也越大。

Iwamatsu [11] 經過實驗認為在具有多個波峰的型態函數，使用  $q > 2.0$  的移動函數效果比  $q = 1.0$  的傳統演化策略好。然而在  $f_1$ 、 $f_5$  的單一波峰函數上，使用  $q = 1.0$  的傳統演化策略效果較好。

而我們使用 Iwamatsu [11] 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 4.3 所示，我們把 Iwamatsu [11] 所作的結果和我們所作的結果放在一起比較如表 4.3。將 Iwamatsu [11] 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 GEP 在  $f_1 \sim f_4$  的函數問題上其實驗的效果(平均值和標準差)比 Iwamatsu [11] 所提出的還佳，但是在  $f_5$  的問題上就略遜一籌。我們的結論是在對於簡單的單峰問題和複雜的多峰問題，GEP 在  $q = 2.5$  的效果會比 CEP 來的優越，但是在少量的多峰問題，傳統的 CEP 較好。這個結論與 Iwamatsu [11] 的差異在於對單峰問題的認知，Iwamatsu [11] 認為 CEP 在單峰問題上表現的較好，但我們實驗的結論是在單峰的問題上 GEP 較好，如圖 4.2 和圖 4.3 所示。

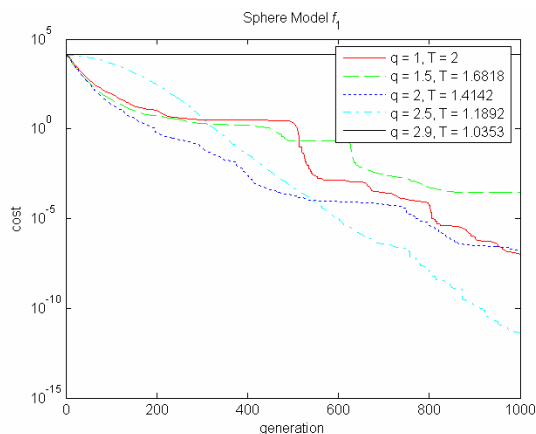


圖 4.2 NGEp 在  $f_1$  函數對不同  $q$  執行結果

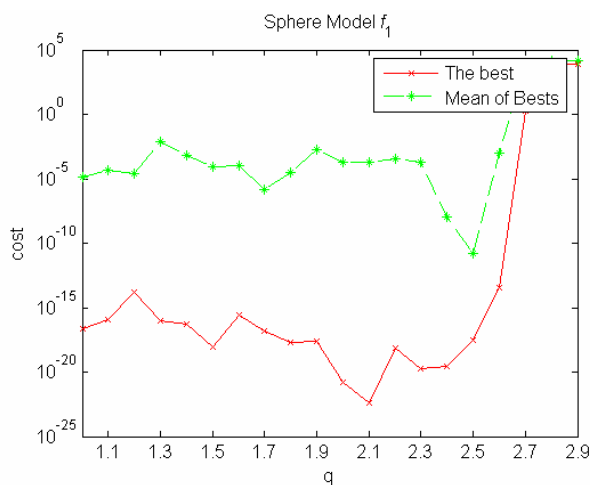


圖 4.3 NGEp 在  $f_1$  函數執行的結果

由圖 4.2 和圖 4.3 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  $\sigma$  的判斷上以及是否使用正確的 Tsallis 隨機變數產生器。

表 4.3 實驗結果比較表

	Gene - ration	CEP Mean (STD)	GEP Mean (STD)	NCEP Mean (STD)	NGEP Mean (STD)
$f_1$	1000	$8.19 \times 10^{-9}$ ( $2.03 \times 10^{-9}$ )	$2.14 \times 10^{-7}$ ( $7.87 \times 10^{-8}$ )	$1.268 \times 10^{-5}$ ( $6.645 \times 10^{-5}$ )	$1.451 \times 10^{-11}$ ( $8.207 \times 10^{-11}$ )
$f_2$	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)



## 參考文獻

- [1] 林豐澤, "演化式計算上篇: 演化式演算法的三種理論模式", *智慧科技與應用統計學報*, 第3卷第1期, 第1-28頁, 2005。
- [2] 陳信昌, "可處理離散和混和變數的演化策略法", 國立中興大學機械工程學系研究所, 2005。
- [3] 陳孟成, "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, 2004。
- [4] 郭文偉, 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, 2005。
- [5] 謝長泰, "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, 2006。
- [6] Bäck.T, "Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms". New York: Oxford University Press, 1996.
- [7] Deng, J., Chang, C. & Yang, Z, "An Exact Random Number Generator for Visiting Distribution in GSA". *I. J. of the Simulation* Vol.6 No 12-13, 2005.
- [8] Fogel, L. J., Owens, A. J., and Walsh, M. J., "Artificial Intelligence through Simulated Evolution", New York: Wiley, 1966.
- [9] Fogel, D. B., "Evolving artificial intelligence", Ph.D. thesis, University of California, San Diego, CA, 1992.
- [10] Holland, J. H., "Adaptation in natural and artificial systems", Ann Arbor, MI: The University of Michigan Press, 1975.
- [11] Masao Iwamatsu, "Generalized evolutionary programming with Levy-type mutation", *Computer Physics Communications*, 2002.
- [12] Manttega, R.N., "Fast, accurate algorithm for numerical simulation of Levy stochastic process", *Physical Review E*, 49, pp.4677-4683.
- [13] Rechenberg, I., "Cybernetic solution path of an experimental problem", Aircr. Establ., libr. Transl. 1122. Farnborough, Hants., UK, 1965.
- [14] Rechenberg . I, "Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipiender biologischen Evolution". Stuttgart, Germany: Frommann-Holzboog, 1973.
- [15] Schwefel, H.-P, Numerical Optimization of Computer Models. Chichester: Wiley, 1981.
- [16] Schwefel, H.-P. "Collective intelligence in evolving systems," in W. Wolff, C. J. Soederand F. Drepper (Eds), *Ecodynamics –Contributions to Theoretical Ecology*, pp. 95-100. Berlin: Springer, 1987a .
- [17] Tsallis, C. and Stariolo, D. A., "Generalizes simulate dannealing", *Physical A*, 1996.
- [18] Xin Yao, Yong Liu, and Guang ming Lin. "Evolutionary programming made faster". *IEEE Trans. Evolutionary Computation*, 1999.