

改良式粒子群演算法最佳化設計

邱進東

德霖技術學院 創意產品設計系 副教授

摘要

粒子群演算法(Particle Swarm Optimization, PSO)為啟發式演算法，是仿自鳥類群體覓食行為，已被廣泛運用於各個領域之最佳化問題，本文以粒子群演算法為基礎，藉由格點法空間分割的概念對搜尋空間進行分割與擇優，利用格點法一維搜尋的特性檢視各個維度的關聯度，經由關聯度的大小來選擇最佳化的機制，並基於粒子群演算法的架構，提出改良式粒子群演算法，以求解全域最佳解。

最後以五種標竿函數為例，測試粒子群演算法與改良式粒子群演算法收斂比較，結果顯示改良式粒子群演算法有很好的收斂效果。

關鍵字：粒子群演算法、標竿函數、最適化

Optimization Design of Modified Particle Swarm Method

Jinn-Tong Chiu

De Lin Institute of Technology, Department of Creative Product Design Associate Professor

Abstract

A new variant of PSO is developed to improve the performance of PSO, which has been widely used in various fields for optimization. The proposed PSO incorporates a space partitioning technique in grid method with PSO. In the searching process of the new algorithm, three position vectors are introduced to enhance the exploration of the particles in the population of PSO and hence helpful for the search of the global solution of a problem of interest.

First, the proposed variant of PSO is verified by applying it on the seven benchmark men-made functions and thereafter proved from the results as a robust one. Next, we applied the algorithm on the optimization design of 5 benchmark functions. The convergence test is compared between Particle Swarm Method and Modified Particle Swarm Method. The results shows the Modified Particle Swarm Method have good convergence.

Keywords: particle swarm optimization, benchmark function, optimization

壹、緒論

粒子群演算法啟發於鳥群遷移的生態行為。文獻[1]提出 PSO 對於低維度問題與單極值問題，大多能搜尋到全域最佳解，而處理高維度問題與多極值問題，往往搜尋不到全域最佳解。1999 年，Suganthan[2]就線性遞減慣性權重(L.PSO)提出動態性增加鄰近粒子數的方法；將線性遞減慣性權重函數的觀念，用於認知學習因子 C_1 與社群學習因子 C_2 參數上，對族群於全域最佳值的搜尋能力與精確度有一定的改進。2000 年，Kennedy 與 Clerc[3]由基本的數學與解析觀

點，提出壓縮因子式粒子群最佳化演算法(Particles Swarm Optimization With Constriction Factor，PSO-CF)，有效控制各個粒子的搜尋軌跡，且不設限粒子的最大速度 v_{\max} 。2001 年，Shi 與 Eberhart[4]以 PSO-CF 參數及其特性，修正 PSO-IW 中的慣性權重函數，並應用於動態性系統(dynamic systems) 問題的最佳化。2004 年，Chatterjee 與 Siarry[5]提出非線性動態慣性權重，依照不同的前一代與後一代的慣性權重關係選擇相對應的慣性權重，以增加粒子群的活性(離散度)。2005 年，Ling 等人[6]提出 PSO 結合混沌系統(chaotic system)，首先改良慣性權重，依照每個粒子與平均函數之間的關係選擇所要的慣性權重，接著加入混沌區域搜尋法(chaotic local search,CLS)，其增加 PSO 的活性及精確度。2006 年，Shu-Kai 與 Erwie[7]提出 PSO 結合單純形演算法概念，使單純形法的粒子移動模式取代 PSO 原有的粒子移動模式，利用單純形法粒子移動的擴散度來避免 PSO 落於區域最佳解，2010 年，Peng-Yeng Yin 與 Fred Glover[8]提出一數碼群演算法(Cyber Swarm algorithms)，其納入自適應記憶學習策略，使 pbest 分散搜索和路徑重新鏈接，以增加粒子群活性，增加搜尋精度，2011 年，Mohammed El-Abd[9]結合兩種群智能演算法，提出混合式人工蜂群(Artificial Bee Colony)粒子群演算法(PSO)，透過人工蜂群在多極值函數的優異表現，來改善粒子群演算法於多極值問題上搜尋的缺點。

綜合上述文獻可以發現，改良粒子群演算法的策略有三大類：(1)改變 PSO 的數學模型運算方式、(2)改良數學模型的參數如慣性權重 ω 與學習因子 C_1 及 C_2 、(3)於 PSO 搜尋期間加入強勢的區域搜尋機制；前兩項改善策略皆為增加 PSO 族群的活性，藉由不同的機制使族群的離散程度增加，以防止 PSO 過早收斂的缺點；而第三項改善策略則是希望透過較佳個體的進一步搜尋，避免可能落於全域最佳解附近的粒子被其他粒子吸引走，導致搜尋失敗。

貳、改良式粒子群演算法

一、格點法(Grid Method)

格點法(Grid Method)又稱爲一維全域搜尋法[10]，在已確定的搜尋區間 $[a, b]$ 內，選擇幾個內等分點 $\alpha_1, \alpha_2, \dots, \alpha_{nd}$ ，分成 (n_d+1) 個等分的子區間；接著，計算這些點的函數值，並找出最小函數值的點，將該點設爲 f_m ，即 $f_m = \min\{f(a), f(\alpha_1), f(\alpha_2), \dots, f(\alpha_{nd}), f(b)\}$ ；若 $f(\alpha)$ 在 $[a, b]$ 區間爲單峰函數，所以與 f_m 相對應的點 α_m 之左右鄰點 α_{m-1} 和 α_{m+1} 之間，應包 $f(\alpha)$ 的極小點 α^* ，因此取 $[\alpha_{m-1}, \alpha_{m+1}]$ 做爲新搜尋區間；重複上敘步驟，持續縮短搜尋區間，直到滿足預先設定之精度要求爲止。本文爲避免 $f(\alpha)$ 在區間爲非單峰函數或程序判斷失真，本文採用縮短比率 (ρ) 的

方式來進行新搜尋區間之給定。

本文將格點法延伸處理 n 維變數問題，處理方式為：首先於搜尋區間隨機選取一個體 X_{ref} ，逐序依每一維進行上述格點法之全域搜尋程序，而其他 $n-1$ 維之變數值將之固定，所得結果取代 X_{ref} 中相關維度之值，以得到 n 維變數問題之最佳解。

以逐次每一變數進行格點法搜尋又稱為迭代式格點法，它的計算程序非常簡單，對變數之間沒有交互作用存在的最佳化問題，很容易求得接近全域最佳解。

二、 粒子群演算法(Particle Swarm Optimization)速度更新模式

PSO 的初始世代以隨機方式產生粒子群係於每一世代中，粒子藉由兩種搜尋記憶以改變搜尋方向：第一種為粒子個體經驗最佳位置，稱為 S_i^{pbest} ，另一種為群體經驗最佳位置，稱為 S^{gbest} ；其搜尋示意圖如圖 1 所示。

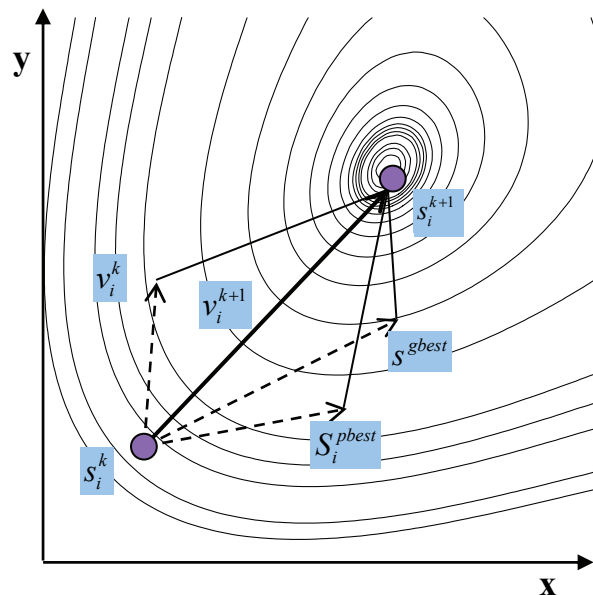


圖 1 粒子速度與位置更新示意圖

PSO 的演算程序說明如下：

- 初始粒子族群中，每一個粒子在 n 維空間裡的位置與速度係以隨機的方式產生之。
- 針對所設定的目標函數，評估每一個粒子的適應函數值。
- 適應函數值與粒子本身的經驗最佳函數值比較，而粒子依照個體目前最佳位置以修正下一次搜尋的粒子速度。
- 個體最佳函數值與群體最佳函數值作比較，當個體最佳值優於群體經驗最佳值，則群體最佳函數值將為個體最佳值所取代；同時每個粒子依照目前群體最佳位置以修正下一次搜尋的粒子速度。

e. 每一粒子的速度與位置進行更新

$$v_{ij}^{k+1} = wv_{ij}^k + c_1 \times rand() \times (s_{ij}^{pbest} - s_{ij}^k) + c_2 \times rand() \times (s^{gbest} - s_{ij}^k) \quad (1)$$

$$s_{ij}^{k+1} = s_{ij}^k + v_{ij}^{k+1} ; j = 1, \dots, n \quad (2)$$

其中 v_i^k 與 v_i^{k+1} 為粒子 i 分別於第 k 與 $k + 1$ 世代的的速度， w 為慣性權重(Inertia weight)， c_1 與 c_2 為學習因子(Acceleration constants)， s_i^{pbest} 為粒子 i 經驗最佳位置， s^{gbest} 為群體經驗最佳位置， s_i^k 與 s_i^{k+1} 為粒子 i 於第 k 與 $k + 1$ 第世代位置， $rand()$ 為介於 0~1 之間的隨機亂數， $rand()$ 之值於每維都不相同。

f. 若滿足設定之收斂終止條件則停止，否則跳回步驟 b，終止條件通常為達到全域最佳解或超過所設定之最大世代數。

在(1)式中，等號右邊第二項為粒子認知模式(Cognition-only Model)，而第三項為粒子族群社會模式(Social-only Model)；每一粒子的速度與移動位置必須限制在所設定之最大速度 v_{max} 與最大位置 s_{max} 之範圍，若粒子速度與位置超過該限制值時，則其速度與位置以 v_{max} 與 s_{max} 表示。

三、改良式粒子群演算法(Modified Particle Swarm Optimization, MPSO)

本文所提出之改良式粒子群演算法，結合上述迭代式格點法於多維度搜尋之概念，以得到四個個體，利用這四個個體間互相牽動的關係來改善粒子群演算法於高維度與多極值無法收斂等問題，個體 1、個體 2 分別為粒子群演算法經過世代演化而產生之群體最佳 s^{gbest} 及所有粒子最佳解之平均 X_{avg}^{pbest} ，個體 3 為經迭代式格點法而產生之最佳個體 CGM_{best} ，個體 4 為利用迭代式格點法之模式，產生的擾動個體 $X_{Disturb}$ ，將此個體引入族群中取代目前表現最差的粒子最佳位置，其中個 s^{gbest} 與 CGM_{best} 擇較優者引入族群。

本文所提出之改良式粒子群演算法流程如圖 2 所示，步驟如下：

- a. 設定粒子群演算法與迭代式格點法各項參數
 - (a) 搜尋空間
 - (b) 問題維度(n)
 - (c) 族群個體數(NP)
 - (d) 慣性權重(w)

- (e) 學習因子(c_1 、 c_2)
 - (f) 最大世代數(Itermax)
 - (g) 收斂條件
 - (h) 切割點數(N_d)
 - (i) 迭代式格點法範圍收縮率(ρ)
 - (j) 迭代式格點法最大迭代次數($CGM_{iter\ max}$)
 - (k) 隔 N_E 世代數執行一次迭代式格點法
- b. 於 n 維度的搜尋空間內，以隨機方式產生初始族群的位置 s_{ij}^k 與速度 v_{ij}^k ，
 $i=1,2,\dots, NP$ ， $j=1,2,\dots,n$ ， k 為世代數。
- c. 當 $\text{mod}(k, N_E) = 0$ ，執行迭代式格點法，求得最佳個體 CGM_{best} ，否則跳至步驟 d
- d
- (a) 在族群中，以個體最佳 s_i^{pbest} 取其平均產生一顆新的個體 X_{avg}^{pbest} ，在 k 世代中以族群 s_{ij}^k 取其平均產生第二顆新的個體 X_{avg}^k ，其表示如下：

$$X_{avg}^{pbest} = \frac{1}{NP} \sum_{i=1}^{NP} s_i^{pbest} \quad (3)$$

$$X_{avg}^k = \frac{1}{NP} \sum_{i=1}^{NP} s_i^k \quad (4)$$

- (b) 將個體 X_{avg}^{pbest} 帶入迭代式格點法，求得最佳個體 CGM_{best}

$$\text{萃取出的第 } j \text{ 維之上界限值 } \overline{X}_j^k = \max \{s_{ij}^k; i=1, \dots, NP\}$$

$$\text{萃取出的第 } j \text{ 維之下界限值 } \underline{X}_j^k = \min \{s_{ij}^k; i=1, \dots, NP\}$$

於 $\overline{X}_j^k \sim \underline{X}_j^k$ 間進行迭代式格點法搜尋， $G=1,2,\dots, CGM_{iter\ max}$

$$CGM_{best}^G = \left\{ \min \{f(X_{j_1}^G), f(X_{j_2}^G), \dots, f(X_{j_{N_d}}^G)\} \right\} \quad (5)$$

(c) 將個體 X_{avg}^k 、 s^{gbest} 與 CGM_{best} 以個體 X_{avg}^k 為基準，利用格點法之模式，找出各個維度最佳的組配，產生一個新的擾動個體 $X_{Disturb}$

(d) 將擾動個體 $X_{Disturb}$ 、個體最佳平均 X_{avg}^{pbest} 與

$X^{gbest} = \min\{f(CG M_{best}), f(s^{gbest})\}$ 引入族群中，取代最差的三個個體最佳

位置， $k = k + 1$ ，跳至步驟 d

d. 執行粒子群演算法進行族群演化

e. 當演化世代滿足收斂條件後停止，否則返回步驟 c

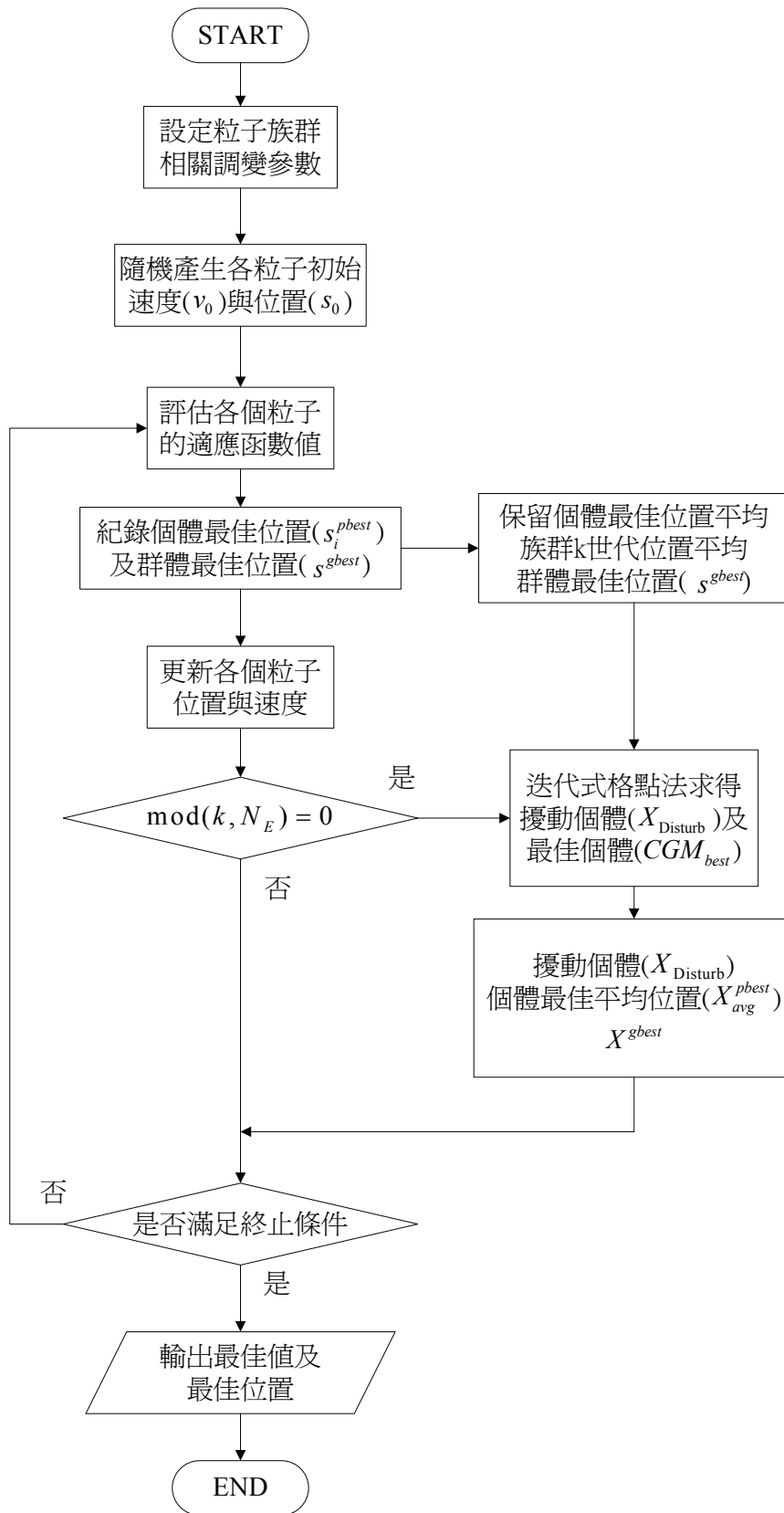


圖 2 改良式粒子群演算法流程圖

參、標竿函數最適化

本章第一節為介紹 5 個標竿問題，第二節將介紹標竿問題中可分離與不可分離之特性並提出判別準則，進行效能測試與比對。

一、標竿問題敘述

為了瞭解本文所提出的優化工具於搜尋全域最佳解之能力，本文選取常被文獻引用之七個古典標竿測試函數[10,11]和四個 CEC'2010 亂數平移式測試函數[12]，進行測試；F₁F₂ 測試函數為單極值問題，F₃~F₅ 測試函數為多極值問題。

此 5 種標竿測試函數可概分為可分離之測試函數與不可分離之測試函數，其中，可分離之測試函數較為單純易解，不可分離之測試函數在搜尋最佳解的過程中，各維度的解會有互相干擾的現象，導致搜尋時間變長以及不易收斂，上述之問題將在本章第二節進行深入探討。

七個古典標竿測試函數如表 1，F₁~F₅ 之二維函數圖形及等高線圖，如圖 3~22 所示。

表 1 七個古典測試標竿函數[10,11]，*n*=維數

代號	名稱	標竿函數	搜尋空間	解析解
F1	Sphere	$F_1(x) = \sum_{i=1}^n x_i^2$	$-100 \leq x_i \leq 100$ $i = 1, 2, \dots, n$	$f_1(\vec{0}) = 0$
F2	Quadric	$F_2(x) = \sum_{i=1}^n (\sum_{j=1}^n x_j)^2$	$-100 \leq x_i \leq 100$ $i = 1, 2, \dots, n$	$f_2(\vec{0}) = 0$
F3	Rosenbrock	$F_3(x) = \sum_{i=1}^{n-1} [100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2]$	$-30 \leq x_i \leq 30$ $i = 1, 2, \dots, n$	$f_3(\vec{1}) = 0$
F4	Ackley	$F_4(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e$	$-32 \leq x_i \leq 32$ $i = 1, 2, \dots, n$	$f_4(\vec{0}) = 0$
F5	Griewank	$F_5(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$-600 \leq x_i \leq 600$ $i = 1, 2, \dots, n$	$f_5(\vec{0}) = 0$

以下針對各個函數做說明：

1. Sphere

$$F_1(x) = \sum_{i=1}^n x_i^2$$

此函數為單極值函數，搜尋空間為 $x_i \in R, i = 1, 2, \dots, n$ ，全域最佳解極值為

$x_i = 0, i = 1, 2, \dots, n$ ，函數值 $F_1(x) = 0$ 。圖3至圖6分別為一維 $F_1(x)$ 之等高線圖、二維 $F_1(x)$

之等高線圖以及側視圖。

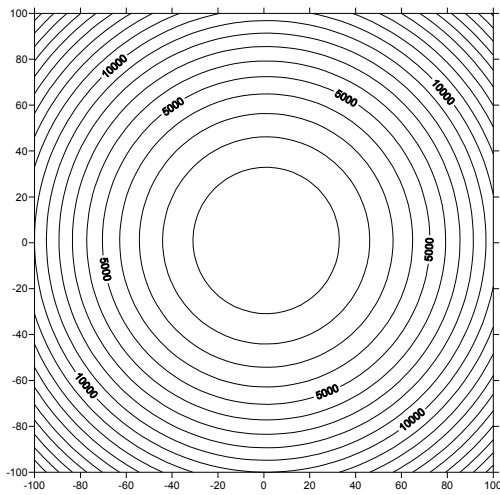


圖 3 二維 $F_1(x)$ 之等高線圖

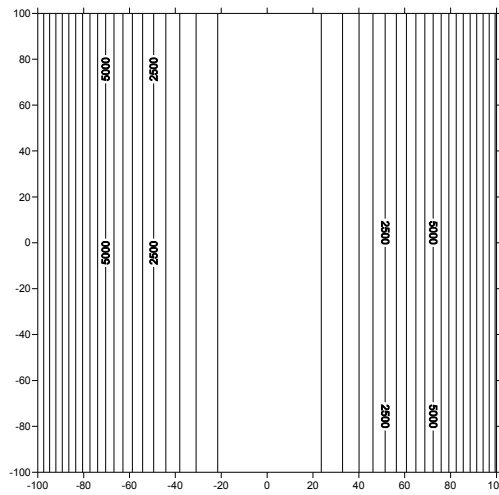


圖 4 一維 $F_1(x)$ 之等高線圖

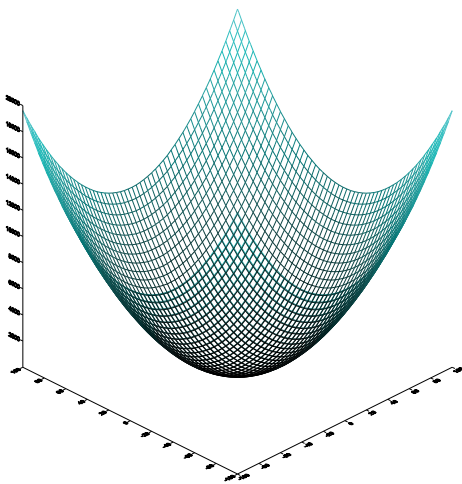


圖 5 二維 $F_1(x)$ 之側視圖

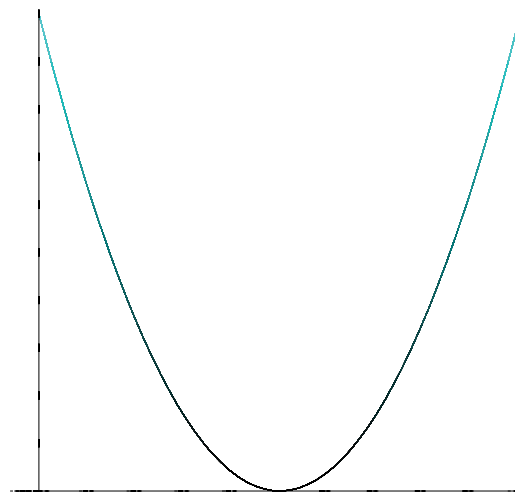


圖 6 一維 $F_1(x)$ 之正視圖

2. Quadric

$$F_2(x) = \sum_{i=1}^n \left(\sum_{j=1}^i x_j \right)^2$$

此函數為單極值函數，搜尋空間為 $x_i \in R, i = 1, 2, \dots, n$ ，全域最佳解極值為

$x_i = 0, i = 1, 2, \dots, n$ ，函數值 $F_2(x) = 0$ 。圖 7 至圖 10 分別為一維 $F_2(x)$ 之等高線圖、二維

$F_2(x)$ 之等高線圖以及側視圖。

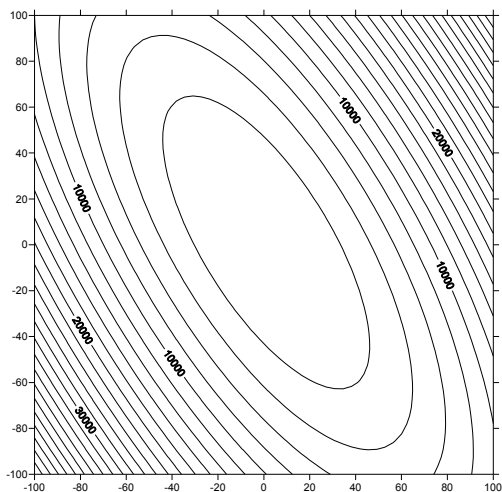


圖 7 二維 $F_2(x)$ 之等高線圖

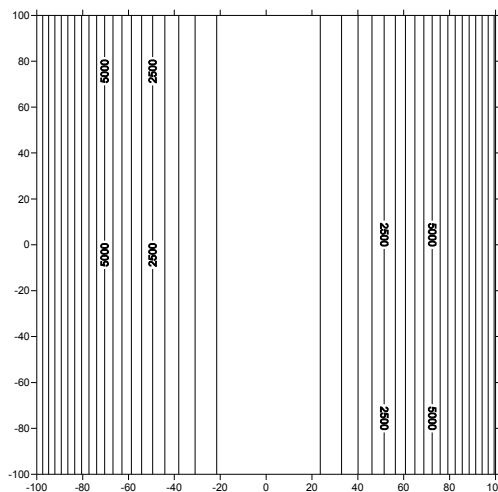


圖 8 一維 $F_2(x)$ 之等高線圖

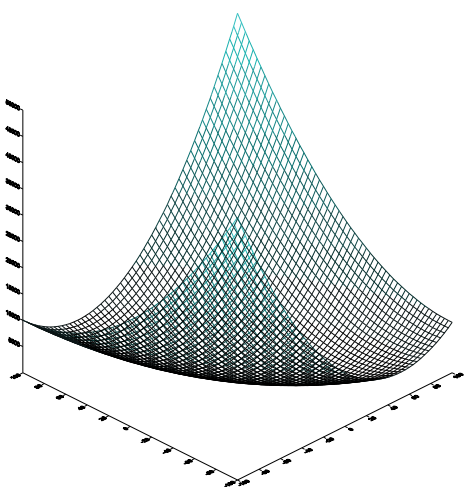


圖 9 二維 $F_2(x)$ 之側視圖

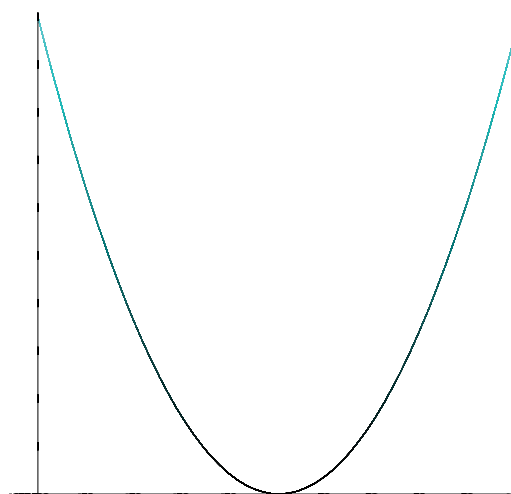


圖 10 一維 $F_2(x)$ 之正視圖

3. Rosenbrock

$$F_3(x) = \sum_{i=1}^{n-1} [100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2]$$

此函數為多極值函數(三維以上)，搜尋空間為 $x_i \in R, i = 1, 2, \dots, n$ ，圖形的範圍如香蕉一樣

形狀往極點收縮，全域最佳解極值為 $x_i = 1, i = 1, 2, \dots, n$ ，函數值 $F_3(x) = 0$ 。圖 11 至圖 14 分

別為二維 $F_3(x)$ 之等高線圖以及側視圖。由其等高線可以發現在極值附近為一狹長型的隙縫，搜尋時很難進入於隙縫中。

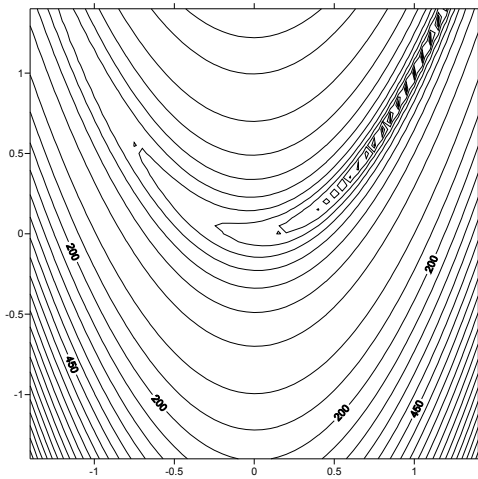


圖 11 二維 $F_3(x)$ 之等高線圖(a)

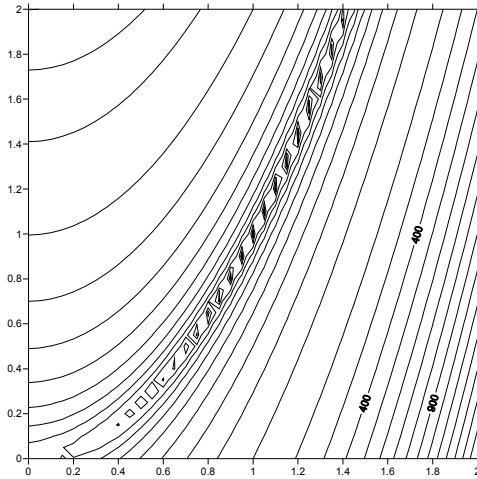


圖 12 二維 $F_3(x)$ 之等高線圖(b)

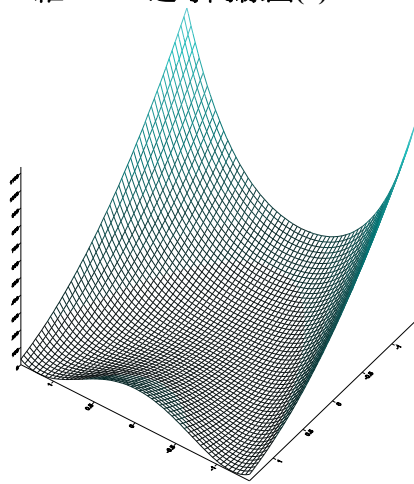


圖 13 二維 $F_3(x)$ 之側視圖(a)

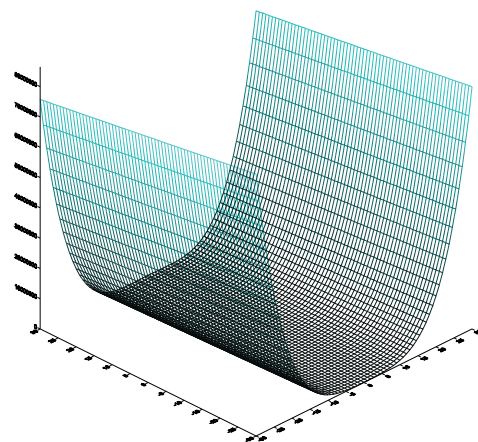


圖 14 二維 $F_3(x)$ 之側視圖(b)

4. Ackley

$$F_4(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e$$

此函數為多極值函數，搜尋空間為 $x_i \in R, i = 1, 2, \dots, n$ ，全域最佳解為 $x_i = 0, i = 1, 2, \dots, n$ ，函數值 $F_4(x) = 0$ 。如圖 15 至圖 18 分別為一維 $F_4(x)$ 之等高線圖、二維 $F_4(x)$ 之等高線圖以及側視圖。二維等高線圖得知全域最佳解附近之形狀為圓。

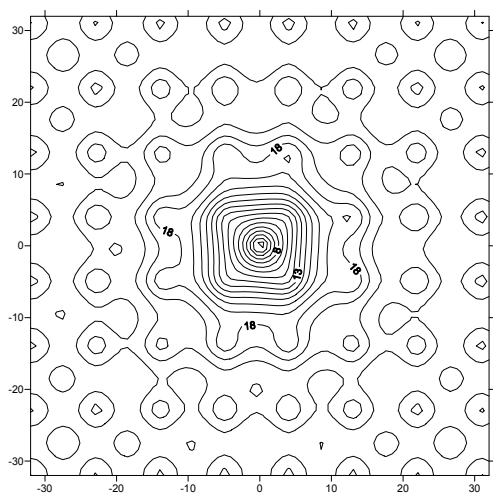


圖 15 二維 $F_4(x)$ 之等高線圖

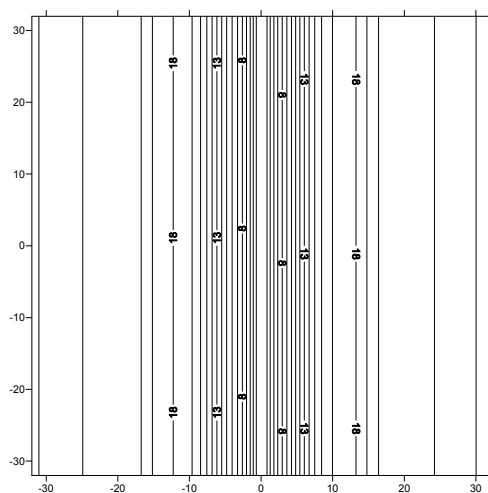


圖 16 一維 $F_4(x)$ 之等高線圖

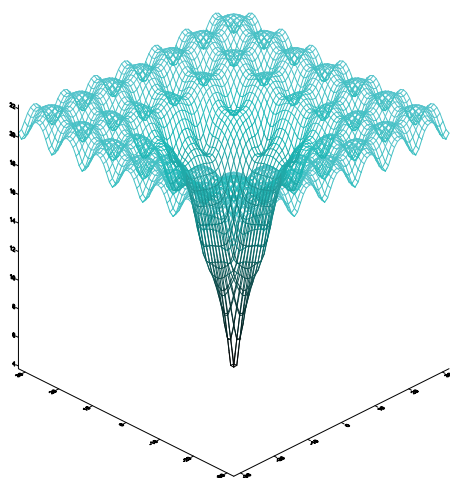


圖 17 二維 $F_4(x)$ 之側視圖

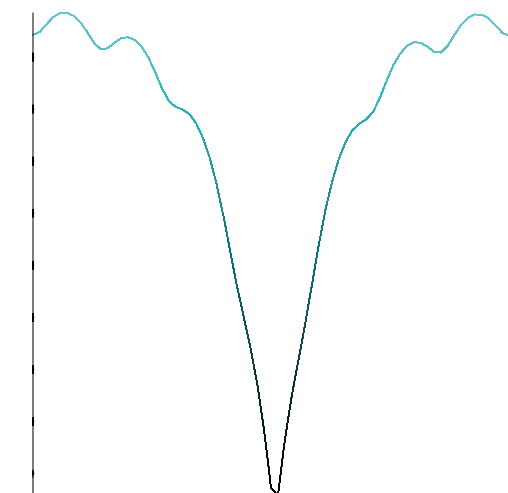


圖 18 一維 $F_4(x)$ 之正視圖

5. Griewank

$$F_5(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$$

此函數為多極值函數，搜尋空間為 $x_i \in R, i = 1, 2, \dots, n$ ，全域最佳解極值為 $x_i = 0, i = 1, 2, \dots, n$ ，函數值 $F_5(x) = 0$ 。圖 19 至圖 22 分別為二維 $F_5(x)$ 之等高線圖以及側視圖。

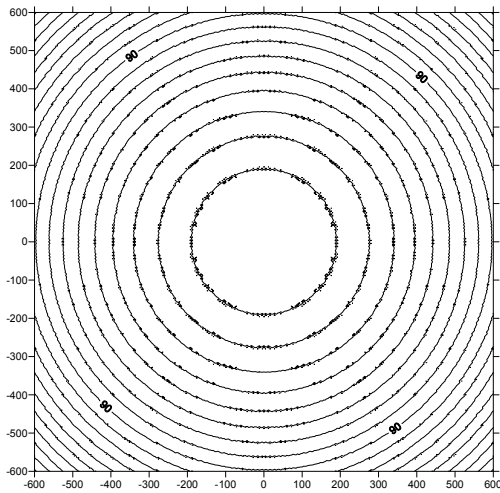


圖 19 二維 $F_5(x)$ 之等高線圖

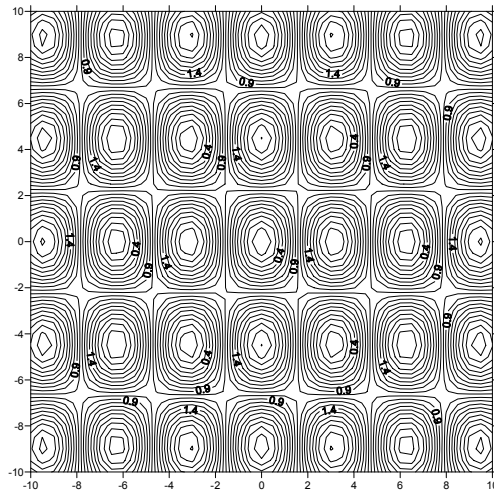


圖 20 二維 $F_5(x)$ 之等高線圖

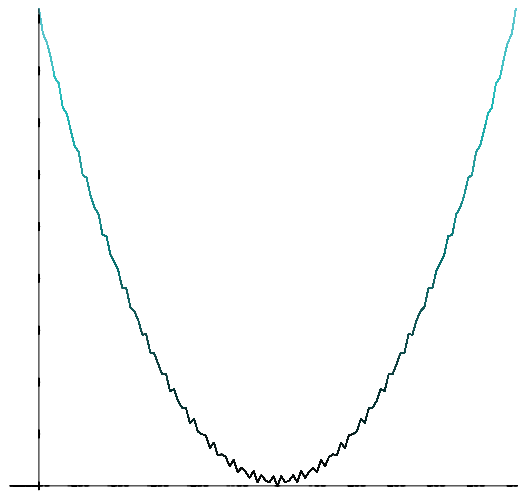
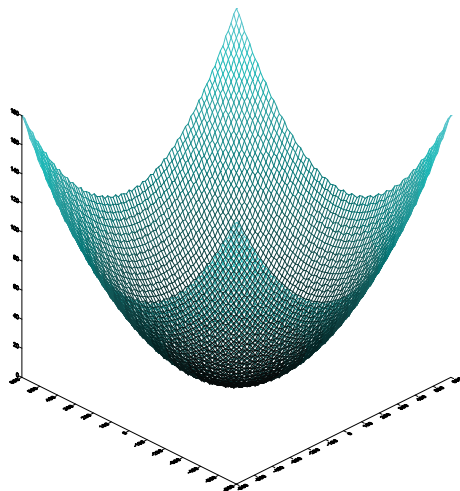


圖 21 二維 $F_5(x)$ 之側視圖

圖 22 一維 $F_5(x)$ 之正視圖

二、改良式粒子群演算法與粒子群演算法性能測試與比較

本文所提出之改良式粒子群演算法，目的為改善標準型粒子群演算法於高維度及多極值不易收斂等缺點，本章節將對改良前後之粒子群演算法以標竿函數進行測試與性能比較，粒子群演算法參數設定如下：

- (a) 搜尋空間如表 1 所示
- (b) 族群數(NP)=80
- (c) 慣性權重(W)=0.6
- (d) 學習因子 $c_1 = 1.5$ 、 $c_2 = 2$
- (e) 切割點數(N_d)=9
- (f) 迭代式格點法範圍收縮率(ρ)=0.98
- (g) 迭代式格點法最大迭代次數($CGM_{iter\ max}$)=10
- (h) 執行迭代式格點法世代數(N_E)=150
- (i) 最大世代數(iternax)=50000
- (j) 收斂條件為達最大世代數或收斂至 $1E - 06$
- (k) 成功率為 30 次測試中滿足收斂條件 $1E - 06$ 之比例

本節將以標竿函數 $F_1 \sim F_5$ 於 30 維、50 維與 100 維，進行 30 次測試，結果數據如表 2~4 所示， f_{av} 為函數平均值、 f_b 為函數最佳值，圖 23~27 為 100 維於 10000 世代前之收斂圖，圖中改良式粒子群演算法(MPSO)為實線，標準型粒子群演算法(CPSO)為虛線。經測試，改良式粒子群演算法於 $F_1 \sim F_5$ 之標竿測試成功率為 100%均可以搜尋到最佳解；改良式粒子群演算法收斂速度很明顯優於粒子群演算法。

表 2.30 維標竿測試函數測試結果

標竿代號	粒子群演算法			改良式粒子群演算法		
	f_{av}	f_b	成功率	f_{av}	f_b	成功率
F1	8.37E-07	5.89E-07	100%	8.76E-07	6.81E-07	100%
F2	3.46E+03	9.26E-07	40%	9.15E-07	0.00E+00	100%
F3	1.82E+04	1.00E-06	3.3%	1.20E-06	9.88E-07	100%
F4	1.30E+01	8.46E-07	23.3%	6.48E-07	8.88E-16	100%
F5	9.03E-03	7.72E-07	43.3%	6.94E-07	0.00E+00	100%

表 3.50 維標竿測試函數測試結果

標竿代號	粒子群演算法			改良式粒子群演算法		
	f_{av}	f_b	成功率	f_{av}	f_b	成功率
F1	9.32E-07	7.46E-07	100%	7.22E-07	0.00E+00	100%
F2	1.91E+04	9.99E-07	3.33%	9.62E-07	0.00E+00	100%
F3	2.18E+04	9.98E-07	10%	9.99E-07	9.95E-07	100%
F4	1.89E+01	9.63E-07	3.33%	8.88E-16	8.88E-16	100%
F5	1.44E-02	8.90E-07	43.33%	8.79E-07	0.00E+00	100%

表 4.100 維標竿測試函數測試結果

標竿代號	粒子群演算法			改良式粒子群演算法		
	f_{av}	f_b	成功率	f_{av}	f_b	成功率
F1	1.71E+04	9.97E-07	16.66%	0.00E+00	0.00E+00	100%
F2	6.70E+04	2.50E+04	-	9.98E-08	0.00E+00	100%
F3	5.15E+07	9.99E-07	6.66%	1.00E-06	9.97E-07	100%
F4	1.99E+01	1.99E+01	-	8.88E-16	8.88E-16	100%
F5	5.82E-02	7.08E-07	40%	5.22E-07	0.00E+00	100%

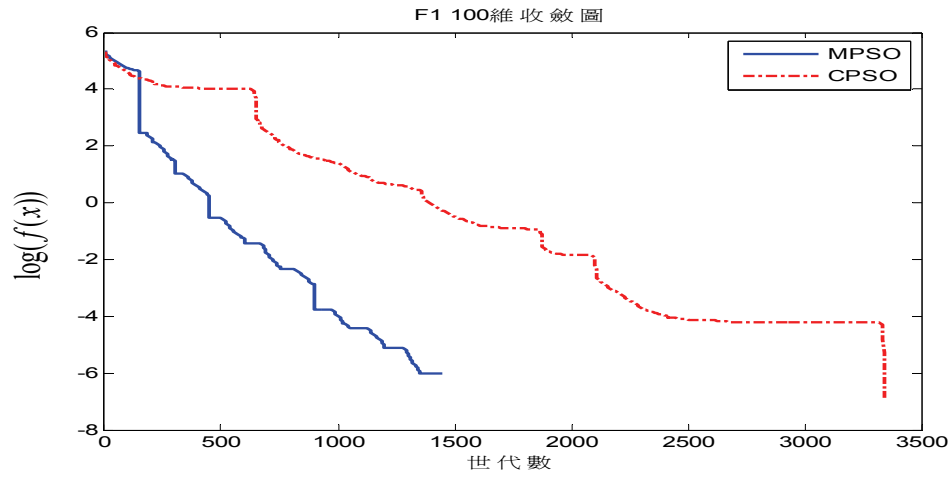


圖 23 MPSO 與 CPSO 於 100 維之 F1 標竿測試函數收斂圖

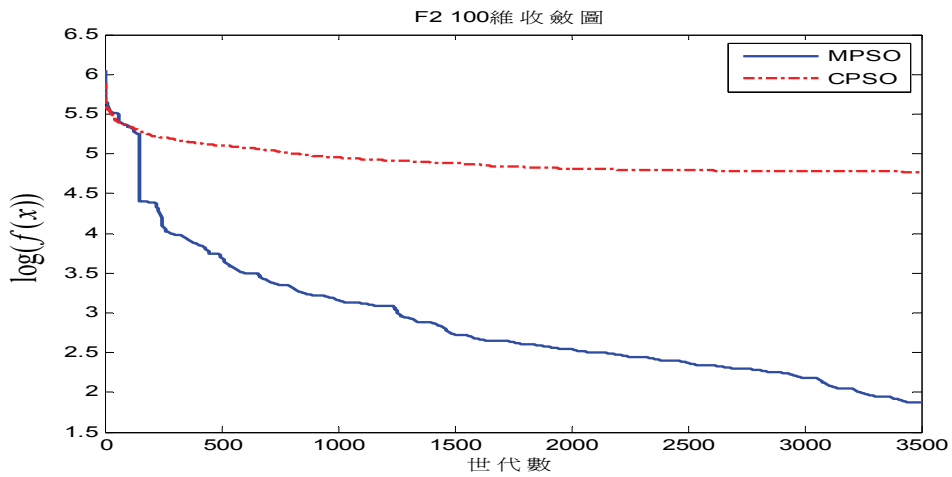


圖 24 MPSO 與 CPSO 於 100 維之 F2 標竿測試函數收斂圖

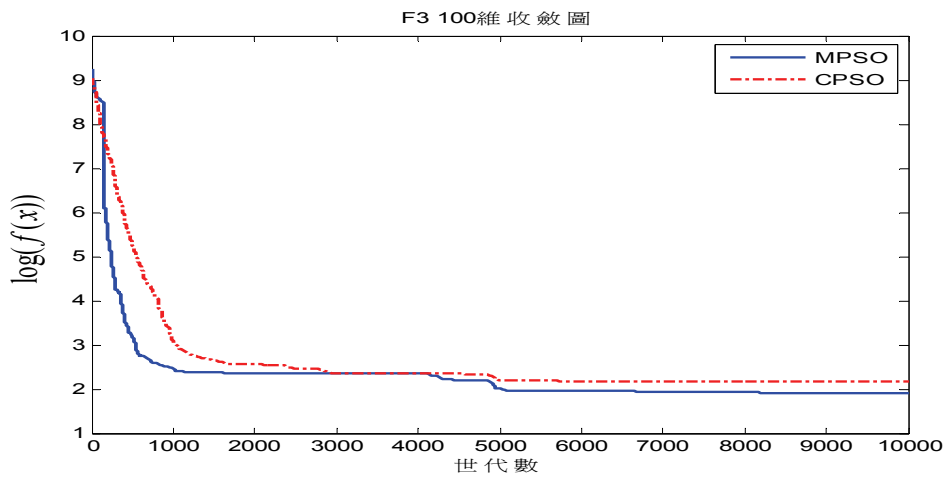


圖 25 MPSO 與 CPSO 於 100 維之 F3 標竿測試函數收斂圖

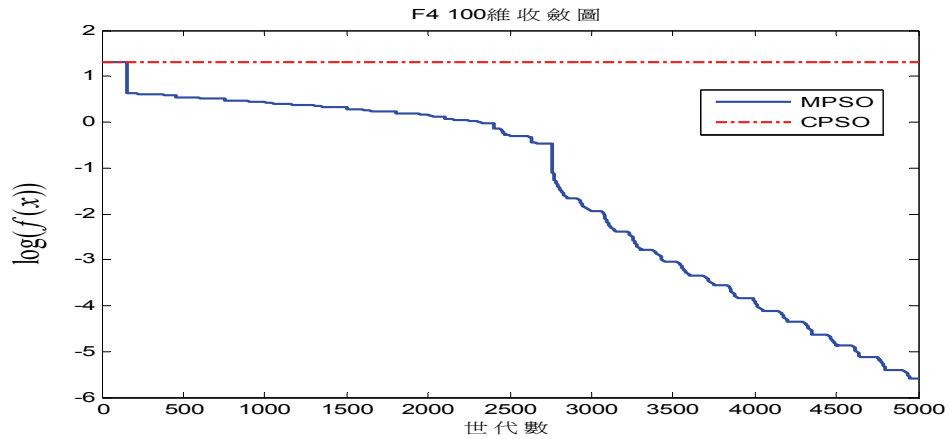


圖 26 MPSO 與 CPSO 於 100 維之 F4 標竿測試函數收斂圖

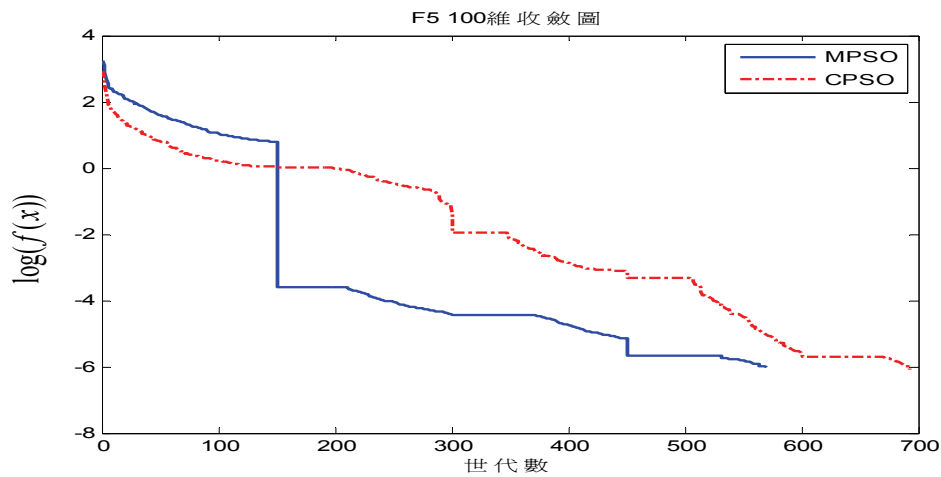


圖 27 MPSO 與 CPSO 於 100 維之 F5 標竿測試函數收斂圖

伍、結論

本章首先以圖形觀測與灰關聯度來判別是否為可分離問題，再將粒子群演算法與改良式粒子群演算法，經過 5 個標竿函數測試後，整理以下結論：

1. 以圖形觀測可分離與不可分離之問題，關鍵在參考點第 i 維 ($i=1\sim n$) 的最佳位置是否隨著其他維度位置的改變而改變，若不變則為可分離問題，同時藉由圖形是否為單調函數判定單或多極值問題。
2. 藉由灰色理論之灰關聯度可以明確判別單極值函數可分離與不可分離之特性，當各維度之灰關聯度絕對差值最大值小於 $1E-14$ 則判定為可分離問題，反之則為不可分離問題。
3. 本文對可分離標竿函數 F1、F4、F5 的收斂效果，相較於標準型粒子群演算法，改良式粒子群演算法有突破性的改善，而不可分離標竿函數 F2、F3，在收斂速度及準確度上也有明顯的

提升。

4. 標竿函數 F4、F5 為多極值問題，在文獻[2,3]中標準型粒子群演算法常常會陷入局部最佳解，本文提出改良式粒子群演算法，將可解決此問題。
5. 而改良式粒子群演算法則可以穩定的收斂，對於不可分離標竿函數 F2、F3，改良式粒子群演算法需要較長的收斂世代數才能達到較優的收斂效果。

陸、參考文獻

- [1]Acan, A. and A. Gunay,“Enhanced Particle Swarm Optimization through External Memory Support,”IEEE,pp.235-242(2005).
- [2]Suganthan, P.N., “Particle Swarm Optimiser with Neighbourhood Operator.” Proceedings of the 1999 Congress on Evolutionary Computation, Washington, DC, IEEE, Piscataway, NJ,USA, Vol. 3, pp.1958-1962,(1999).
- [3]Clerc, M. and J. Kennedy, “The Particle Swarm:Explosion, Stabiliry, and Convergence in a Multimodal Complex Space.” Proceedings of the Congress of Evolutionary Computation, Washington, DC, IEEE, Piscataway, NJ,USA, Vol. 6, pp.58-73,(2000).
- [4]Eberhart, R.C. and Shi, Y. “Tracking and Optimizing Dynamic Systems with Particle Swarms.” Proceedings of the Congress on Evolutionary Computation, Vol 1, 94–100, Soul, IEEE Press(2001).
- [5]Chatterjee, A. and P. Siarry, “Nonlinear Inertia Weight Variation for Dynamic Adaptation in Particle Swarm Optimization.” Computers and Operations Research, Vol. 33,pp.859-871(2004).
- [6]Liu Bo, Ling Wang, Yi-Hui Jin, Fang Tang and De-Xian Huang, “Improved Particle Swarm Optimization Combined with Chaos.” SCI Chaos, Solitons and Fractals, Vol.25,pp.1261-1271(2005).
- [7]S. Fan, Shu-Kai and Erwie Zahara, “A Hybrid Simplex Search and Particle Swarm Optimization for Unconstrained Optimization.” European Journal of Operational Research, Vol. 181, pp.527-548(2006).
- [8]Yin, P.Y., Glover, F., Laguna M., and Zhu J-X., “Cyber Swarm Algorithms–Improving Particle Swarm Optimization Using Adaptive Memory Strategies”, European Journal of Operational Research 201 (2), 377-389(2010).
- [9]Mohammed EI-Abd, “A Hybrid ABC-SPSO Algorithm for Continuous Function Optimization.” Swarm Intelligence (SIS), 2011 IEEE Symposium on 978-1-61284-053-6, IEEE Press(2011).
- [10] 羅俊彥，粒子群演算法於高維數全域最佳化問題，海洋大學造船工程研究所，碩士論文(2006)。
- [11] Kennedy, J. and Eberhart, R.C., “Particle Swarm Optimization.” Proc. IEEE International Conf.

Neural Networks, Vol.4, pp. 1942-1948(1995).

[12] Ke Tang , Xiaodong Li , P. N. Suganthan , Zhenyu Yang , and Thomas Weise , “Benchmark Functions for the CEC’2010 Special Session and Competition on Large-Scale Global Optimization.” Nanyang Technological University (NTU), Singapore, Tech. Rep. January.8.2010.