基因演算法解決Ackley function最小值問題

1102913王柏勝

問題陳述

Ackley function具有多個局部最小值，能成為測試優化算法的工具。嘗試是否可以使用基因演算法，來完成Ackley function的最小值問題。

目標

了解並分析使用Genetic Algorithms (GA) 演化計算算法在解決 Ackley 函數上的性能。

實作環境

編譯工具(軟體) : PyCharm

使用語言: Python

使用函式庫: numpy 、math 、geneticalgorithm、matplotlib…等

先備知識(程式內容)

Ackley Function:  
常用於優化問題的基準函數，其數學表達式如下：

一張含有 文字, 字型, 筆跡, 白色 的圖片

自動產生的描述

Roulette wheel(輪盤法):

此次以輪盤法作為Selection的選擇，所謂輪盤法是假設一個可轉動的輪盤，在輪盤上劃分許多扇形區塊 (這部分本來有實作，但如果每個generation都要顯示，且population的數目過大，將造成實作上的麻煩及檢視的不易，因此此次實作內容最多就顯示個體區塊面積的機率) ，區塊的面積大小正比於個體被複製的機率。因此，個體的適應值越高，適應值佔有族群適應值總和的比例也越高，在輪盤上所佔的面積也越大，而被選上的機率也越高。

菁英主義（Elitism）:

一種演化算法中的策略，其基本思想是在每一代中保留上一代中表現最好（最優秀）的個體，將其直接複製到下一代，以確保已經找到的最佳解不會被遺失。

問題表示法

使用實數陣列（real-valued array）表示解的向量。

演化計算架構(使用GA)

GA flow chart

一張含有 文字, 螢幕擷取畫面, 字型, 圖表 的圖片

自動產生的描述

Genetic Algorithms (GA):

**參數設定**：

#這裡不實作crossover，避免output難以收斂

群體大小：50

演化世代數目: 10

突變率：0.1

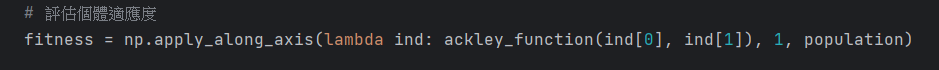
突變範圍：0.2

**演化流程**：

1. 初始化個體



1. 評估適應度



1. Selection

一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述

1. Crossover(略過不使用)
2. Mutation

一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述

1. 菁英主義(Elitism)

一張含有 文字, 字型, 螢幕擷取畫面, 設計 的圖片

自動產生的描述

1. 族群更新

一張含有 文字, 字型, 螢幕擷取畫面, 設計 的圖片

自動產生的描述

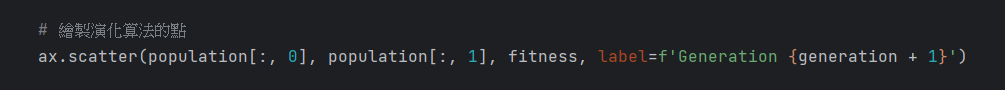
1. 繪製並輸出演化過程二、三維圖表

2D

一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述

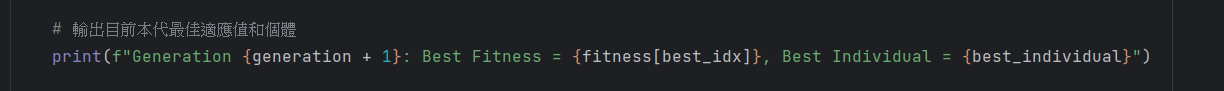
3D



一張含有 文字, 螢幕擷取畫面, 字型 的圖片

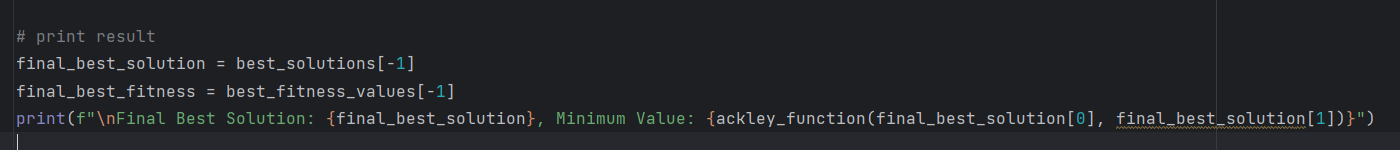
自動產生的描述

1. 輸出最佳解和對應Ackley函數值(也可以輸出每個世代個體輪盤法機率)



一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述



**終止條件**：

最大迭代次數 : 10

實驗觀察方向

在每個演化代數中記錄最佳個體及其適應值，觀察GA在解Ackley function上的收斂速度，並研究參數及GA演算法裡內部所使用的演化方法是否會對其函數找到最小值造成影響。

實作輸出(圖表及數據)

3D

一張含有 文字, 螢幕擷取畫面, 圖表, 地圖 的圖片

自動產生的描述

2D

一張含有 文字, 螢幕擷取畫面, 繪圖, 行 的圖片

自動產生的描述

Output

一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述

問題與討論

Q1:在參數上，若population size改大，會造成影響嗎?

通常會，因為個體增多，個體的多樣性也增多，這能夠防止演算法過早地收斂到局部最佳解，有助於找到全局最佳解。但缺點可能是要負擔更大的計算成本

Q2:在參數上，若增加演化generation的數量，會造成影響嗎?

可能會對結果產生影響，像是能夠更為全面地搜索整個空間，造成的原因是因為代數變多，有更多時間去進行演算法，但也有可能在某代就局部收斂，導致在某代後就進入穩定態。

Q3: 有沒有可能在某幾代趨於穩定，但在後面適應值又飆高，如圖所示

一張含有 文字, 行, 繪圖, 圖表 的圖片

自動產生的描述

可能會，原因有可能是在進行演算法時，後續的generation在算法上跳出了當前探索空間的局部最小值，發現了有可能是更佳的解，重新繼續迭代。

Q4: 使用輪盤法作為selection的選擇會是好的嗎?

可能不太合適，因為輪盤法主要在於選擇個體，而非最小化像是Ackley function這類的問題，容易陷入局部最佳解，無法有效探索整個空間。可以嘗試使用PSO算法

Q5: 通常使用菁英主義後，不會進行crossover和mutation，那在這個實作裡，是否不恰當?

不會(其實我不太確定)，這是可行的，mutation有助於維持族群的diversity，防止快速收斂到局部最小值。菁英主義則保留每代最好的個體，確保演算過程裡不會丟失已經獲得的優良數據。因此對於這兩個過程，彼此是可以並行的，但也可能造成些影響，像是演算法過度收斂，影響計算成本和速度。

Reference

<https://www.sinotech.org.tw/journal/pdfview.aspx?n=80&s=87>

https://zh.wikipedia.org/zh-tw/%E9%81%97%E4%BC%A0%E7%AE%97%E6%B3%95

<https://medium.com/hunter-cheng/python-%E5%9F%BA%E5%9B%A0%E6%BC%94%E7%AE%97%E6%B3%95-genetic-algorithm-ga-%E6%B1%82%E8%A7%A3%E6%9C%80%E4%BD%B3%E5%8C%96%E5%95%8F%E9%A1%8C-b7e6d635922>

https://ir.nctu.edu.tw/bitstream/11536/48945/11/654911.pdf

<https://blog.csdn.net/m0_72053284/article/details/129645207>

https://hackmd.io/@Tsen/ryz47B7wI?utm\_source=preview-mode&utm\_medium=rec