基因演算法

613k0007c 余品誼

報告使用遺傳演算法優化問題，需包含實驗結果、分析及討論，需包含所使用的參數、準確性、實驗結果的一致性、可獲致滿意結果的維度（D=10、D=30）等。

目標：

1. 優化單峰問題(Shere function)
2. 優化雙峰問題(Schwefel function)

名詞解釋：

1. Shere function

   是一個經典的單峰優化問題，用於測試優化算法的性能。

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

   x_i是變量，n是問題的維度。該函數的特性是：

   - 它的最小值位於 $x = 0$處，且全域最小值為 0。

   -因為它是一個單峰函數（只有一個全域最小值且沒有局部極小值），所以比較適合用來測試算法的全局收斂性。

2. Schwefel function

   是一個常用於測試優化算法的多峰函數

$$f(x) = 418.9829 \times n - \sum_{i=1}^{n} x_i \sin(\sqrt{|x_i|})$$

   x_i是變量，n是問題的維度。該函數的特性如下：

   -定義域通常為$x_i \in$ [-500,500]。

   -全域最小值位於$x = 420.9687 \times n$ 的位置，此時 $f(x) = 0$

   -函數具有許多局部極小值，這使得它是一個多峰函數，對優化算法的探索能力提出了挑戰，因為很容易陷入局部最優解。

3. Genetic Algorithm

是一種基於自然選擇和遺傳學的全局優化算法，模仿生物進化過程來解決複雜的問題。它通過以下步驟來進行優化：初始化種群、適應度評估、選擇（Selection）、交叉（Crossover）、變異（Mutation）、生成新種群、重複演化。

實驗方法(參數選擇)：

實驗一：

1. Shere function

使用 DEAP 庫做族群定義

$$n \in [-100,100]$$

```
toolbox.register("attr_float", random.uniform, -100, 100)  # 初始化範圍
toolbox.register("individual", tools.initRepeat, creator.Individual, toolbox.attr_float, 30)
toolbox.register("population", tools.initRepeat, list, toolbox.individual)
```

分別對基因演算法的步驟做定義

```
toolbox.register("mate", tools.cxBlend, alpha=0.5)  # 混合交叉
toolbox.register("mutate", tools.mutGaussian, mu=0, sigma=1, indpb=0.2)  # 高斯變異
toolbox.register("select", tools.selTournament, tournsize=3)  # Tournament Selection
```

- 混合交叉（Blend Crossover，簡稱 cxBlend）alpha=0.5：這是混合交叉的參數，表示兩個父母的基因對生成的子代影響的比例。
- 高斯變異（Gaussian Mutation）是一種基於高斯分佈來隨機改變個體的基因。
    - mu=0 這是高斯分佈的均值，表示變異後的基因值將圍繞這個均值進行變化。均值設定為 0，意味著變異後的基因值有可能偏向於 0。
    - sigma=1：這是高斯分佈的標準差，表示變異的範圍。標準差越大，變異的幅度越大，個體的新基因值可能與原來的基因值相差更遠。
    - indpb=0.2：這是變異的機率，表示每個基因被變異的概率。在這裡，20% 的概率會導致個體的某一基因被高斯變異。若這個值設置得較低，則變異的效果會減弱。

- 錦標賽選擇（Tournament Selection）是一種流行的選擇策略，用於從種群中選擇個體進行繁殖。tournsize=3：這是錦標賽的大小，表示在每次選擇中將隨機選擇 3 個個體進行比較。這些個體將根據它們的適應度進行競爭，適應度較高的個體將被選中。

2. Schwefel function

$$n \in [-500,500]$$

```
elf.toolbox = base.Toolbox()
elf.toolbox.register("attr_float", random.uniform, -500, 500)
elf.toolbox.register("individual", tools.initRepeat, creator.Individual, self.toolbox.attr_float, 10)
elf.toolbox.register("population", tools.initRepeat, list, self.toolbox.individual)
elf.toolbox.register("evaluate", self.eval_schwefel)
elf.toolbox.register("mate", tools.cxBlend, alpha=0.5)
elf.toolbox.register("mutate", tools.mutGaussian, mu=0, sigma=1, indpb=0.2)
elf.toolbox.register("select", tools.selTournament, tournsize=3)
```

其他基因演算法方法同 Shere function

實驗二：

1. Shere function

   混和交叉→單點交叉

```
toolbox.register("evaluate", evalSphere)
toolbox.register("mate", tools.cxTwoPoint)  # 單點交叉
toolbox.register("mutate", tools.mutGaussian, mu=0, sigma=1, indpb=0.2)  # 高斯變異
toolbox.register("select", tools.selTournament, tournsize=3)
```

2. Schwefel function

   混和交叉→單點交叉

   高斯變異→單點變異

   Tournament Selection →Roulette Wheel Selection

```
self.toolbox.register("mate", tools.cxOnePoint)  # 单点交叉
self.toolbox.register("mutate", tools.mutUniformInt, low=-500, up=500, indpb=0.2)  # Uniform mutation
self.toolbox.register("select", tools.selRoulette)  # Roulette Wheel Selection
```

實驗結果

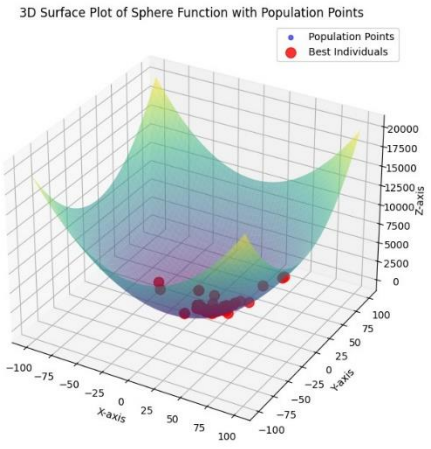
實驗一：

1. Shere function

   改變 D

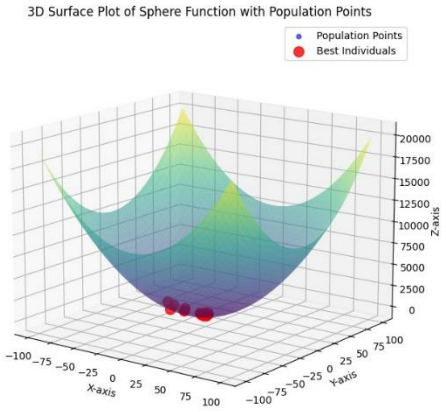
   D = 30

最優適應度: 0.0003773214828516388411917825 70



D = 10

最優適應度: 0.0000019221695004740583990110 37



紅點：圖中的紅點代表當前種群（population）的所有個體（individuals）。每個紅點的座標由其基因組成，並通過適應度函數（Schwefel 函數）計算其對應的 Z 軸值（高度）

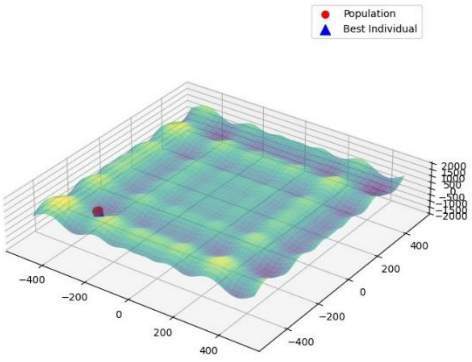
小節 1：最小化問題中，適應度越小越好，故 D = 10 時，成果會比 D = 30 還好。

2. Schwefel function

D = 30

全局最佳適應度: -533521.3674468126

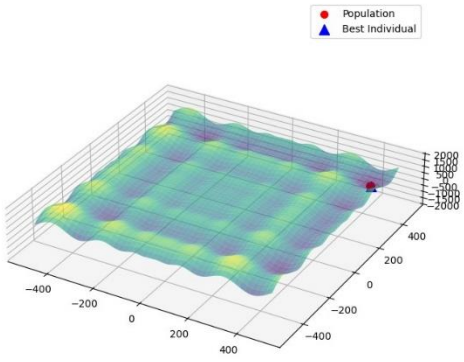
最終最佳個體: [-330.15455163742706, -319.55799716144486, -132.31336466522745, -284.8141948129106, 72.83634533394735, 178.94784120710563, 403.14401467944657, 66.7794595675896, 54.74365523670151, 402.6870558390617, -158.01474060573807, -305.33121421072735, 445.8750535615037, 412.4781764413085, -17.13124016881938, -2383.01553847167, 3.4463233244554705, 395.8081723010098, 204.57028857155527, -450.81503761442343, 83.47041439353517, -526.3485606660615, 59.14445357926086, 402.8999310656067, 214.3259812921213, -538130.673177331, -133.8232587907047, 218.35155779947794, 445.03220053553406, 712.6134287650359]



D = 10

全局最佳適應度: -289.8664246959852

最終最佳個體: [421.7456571897567, 420.3646930931186, 151.17899960137902, 717.7195852418236, 420.7051734025121, -302.4668483244109, 715.8364617716129, 422.2492677238063, -558.0406032383014, -559.8293210195674]



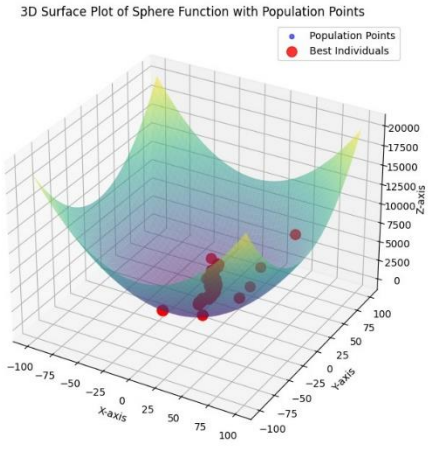
小節 2：D 越大找越小能力越好

實驗二：

1. Shere function

最優適應度: 0.30466858177110744954O636480378

最優個體: [0.11267499733629527, 0.08534927681985671, 0.12073862867399493, -0.04414423300463022, -0.10024743539190323, -0.10845840910151647, -0.10679314259092951, 0.25917695003303964, 0.20009087498707656, -0.07873801392672554, 0.0893071099018907, -0.0051097163395120715, 0.06562336301063565, 0.1735152848777599, -0.08591670552215536, -0.07488340522241407, 0.119909962955869, -0.07946627789297284, 0.04450626304781842, -0.08163217688488977, 0.11846919531363753, 0.0220385635267535, -0.02646303926916005, 0.009381137976143483, -0.010926997412156858, -0.04441832426471948, -0.0969022086305048, 0.023829957694844195, 0.05049425239356564, -0.082968388182321]

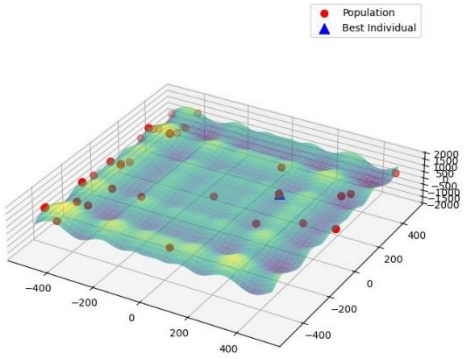


小節 3：fitness 有比混和交叉還差一點

2. Schwefel function

全局最佳適應度: 9381.307660002296

最終最佳個體: [188, 120, -41, 24, -478, -331, -373.55300451124, -189, -78, -372, -4, -28, 168, -500, 432, -94, 49, -442, 41, 39, -14, -232, -4, -299, -356, -117, -71, -499, -124, 413]



小節 4：利用不同基因演算法的方法，fitness 震動幅度比較沒這麼大，但是 fitness 的數據沒有使用 Gaussian 和 Tournament 來的好

結論

1. 結合小節 1 和 2 可以發現基因數越高，就越容易找到最小值。
2. 在 Shere function 裡，單點交叉的優化效果比混和交叉差
3. 在 Schwefel function 裡，利用 cxBlend、mutGaussian 和 selTournament，fitness 震動幅度比利用 cxOnePoint、mutUniformInt、selRoulette 來的大，但是利用 cxBlend、mutGaussian 和 selTournament，fitness 數據有機會比較好。

實驗結果討論與心得

1. 推測在 Schwefel function 裡，利用 cxBlend、mutGaussian 和 selTournament，fitness 震動幅度大，是因為未把數值轉為整數，這要等待後續實驗驗證。
2. 基因演算法是一個很棒的優化問題方法，之後論文可以使用看看。