

實驗流程

利用 GA 實作 Ackley's function，做 30 次，將每次的最佳值取出做平均值。

四種變異算法(Representation + Crossover)：

1. BIT String + 2 Points
2. BIT String + Uniform
3. Real Value + Whole arithmetic
4. Real Value + Uniform

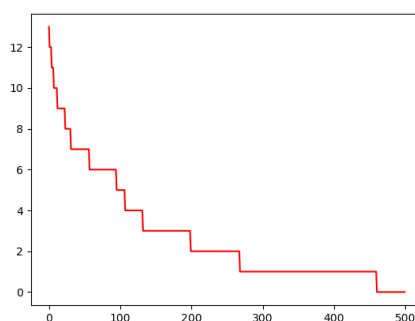
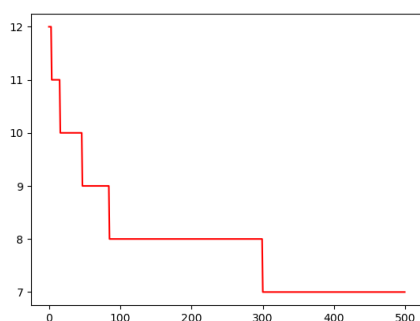
預設參數

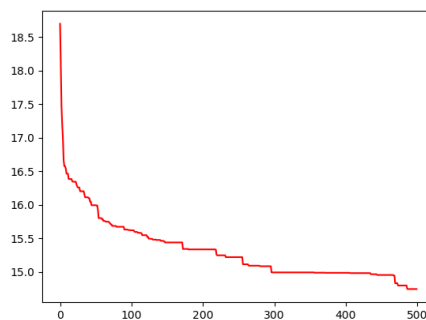
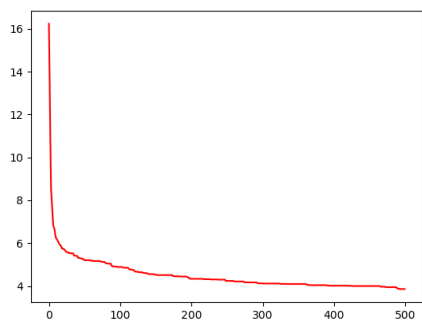
1. Parent Selection: k-tournament
2. Survivor Selection: $\mu+\lambda$
3. Crossover Rate (P_c): 0.9
4. Mutation Rate (P_m): 0.1
5. 世代數(G): 500
6. Tournament Size (k): 2
7. Population Size (μ): 100

使用預設參數，對四種變異算法結果比較

Representation 為 BIT String 的收斂都是一段一段的，Real Value 的收斂是連續。

Real Value + Uniform 效果最不好，最後收斂在 15 左右。

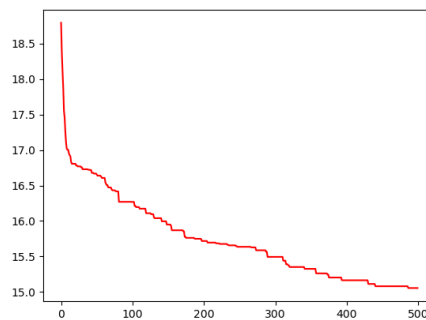
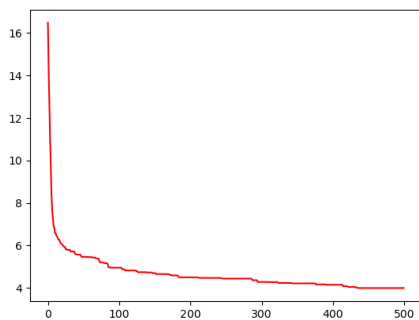
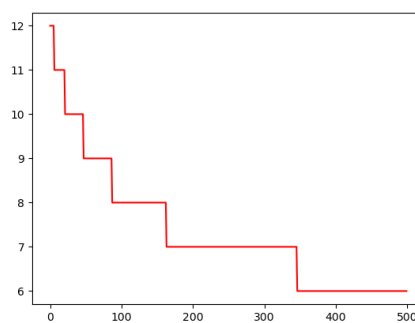
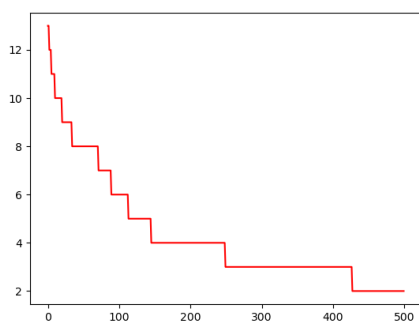




(依序：BIT String + 2 Points, BIT String + Uniform, Real Value + Whole arithmetic, Real Value + UniForm)

調整 P_c 值

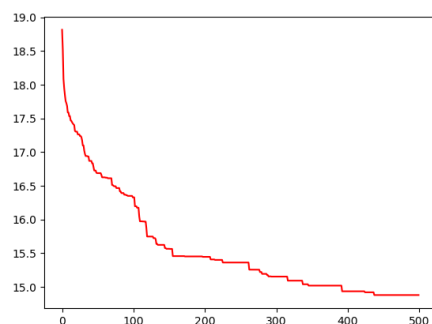
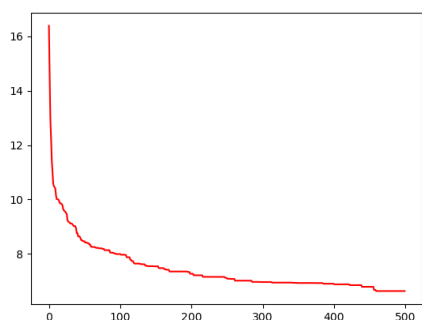
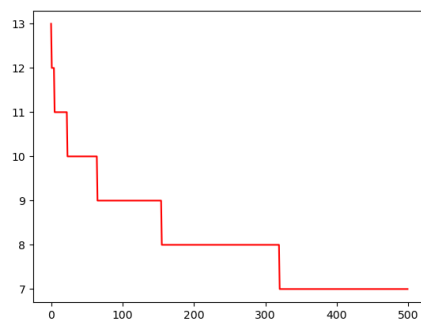
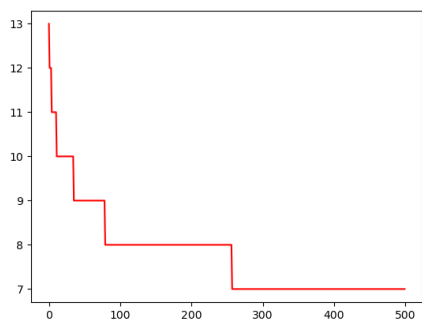
$P_c=0.6$ ，各變異算法沒有太大的變化



(依序：BIT String + 2 Points, BIT String + Uniform, Real Value + Whole arithmetic, Real Value + UniForm)

調整 P_m 值

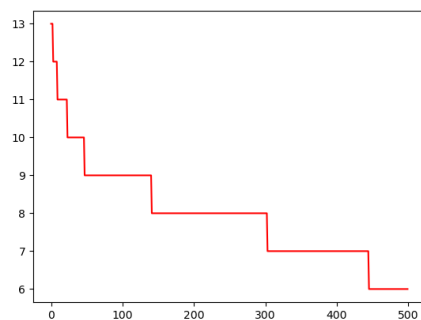
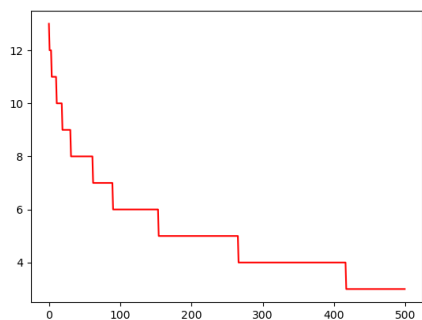
調整 $P_m=0.2$ ，Representation 為 BIT String 會較快收斂。

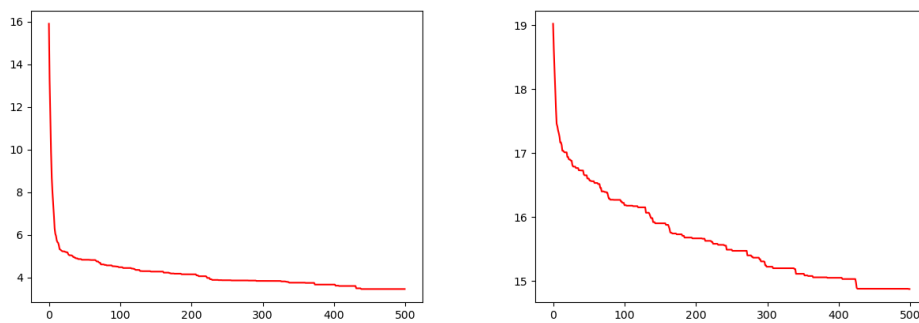


(依序：BIT String + 2 Points, BIT String + Uniform, Real Value + Whole arithmetic, Real Value + UniForm)

調整 K 值

原本想要把四種變異算法都調整 $K = 5$ ，但發現 Representation 為 RealValue 時無法停止，於是將，Representation 為 RealValue 的算法改為 $K = 1$ 。但無論是調整為 $K = 5$ 或 $K = 1$ 的算法結果都沒有明顯差異。

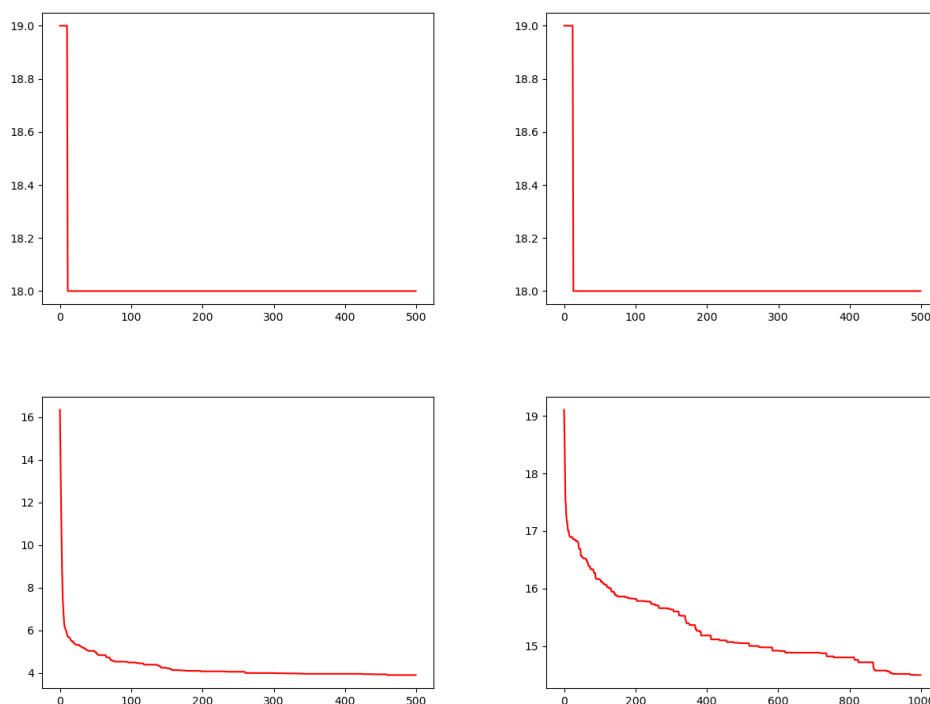




(依序：BIT String + 2 Points, BIT String + Uniform, Real Value + Whole arithmetic, Real Value + UniForm)

調整 Mu 值

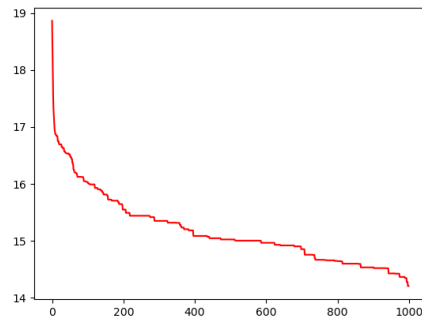
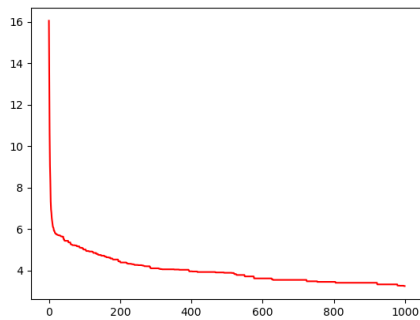
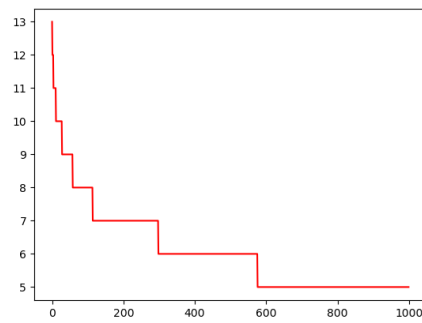
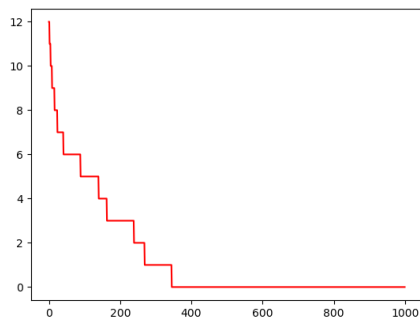
將 Mu 改成 150，Representation 為 BIT String 效果變得極差，幾乎沒有進步。



(依序：BIT String + 2 Points, BIT String + Uniform, Real Value + Whole arithmetic, Real Value + UniForm)

調整 G 值

調整 $G = 1000$ ，理論上進行比較多代演化，最終結果會更好，實際上初始化的個體，加上 Crossover 及 Mutation 都是隨機，因此最終結果和 $G = 500$ 時差不多。



(依序：BIT String + 2 Points, BIT String + Uniform, Real Value + Whole arithmetic, Real Value + Uniform)

總結

四種算法，Real Value + Whole arithmetic 平均表現最優，其次是 BIT String + 2 Points 和 BIT String + Uniform，最後是 Real Value + Uniform。

有些變數調整前後效果不大，比較顯著的調整在於 P_m 和 μ 值，Representation 為 BIT String 的算法，我推測原因是 BIT String 的組合相較於 Real Value 的表示法較少，參數改變就可以讓結果不同，此外 P_m 較大讓整個群組的多元性上升，競爭力加強，使得結果更快找到好的解。Population 增多使得生存壓力變小，因此不容易產出好的後代。