Evolutionary computation HW1

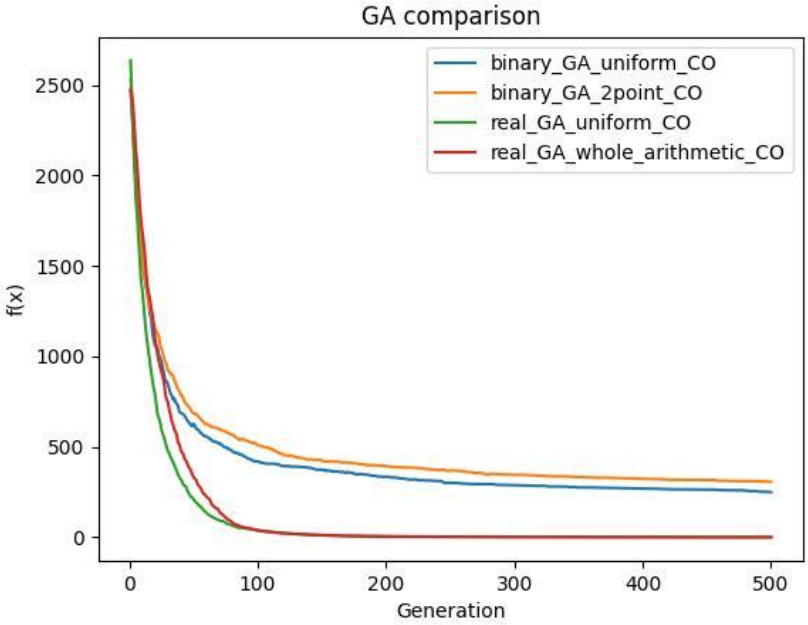
108062226 蘇志翔

1.

如.py 檔所示

執行方式：python3 108062226_HW1.py

2.



其中右上角的圖例名稱代表著下述的意思：

binary_GA_uniform_CO = Binary GA with uniform-crossover strategy.

binary_GA_2point_CO = Binary GA with 2-point-crossover strategy.

real_GA_uniform_CO = Real-valued GA with uniform-crossover strategy.

real_GA_whole_arithmetic_CO = Binary GA with whole_arithmetic -crossover strategy.

3.

Convergence speed: 從

Real-valued GA with uniform-crossover

> Real-valued GA with whole-arithmetic-crossover

≒ Binary GA with uniform crossover

≒ Binary GA with 2-point crossover

Solution quality(僅看最佳 fitness，而不看在演化結束後的最終 fitness):

Real-valued GA with whole-arithmetic-crossover

> Real-valued GA with uniform-crossover

> Binary GA with 2-point crossover

≒ Binary GA with uniform crossover

Note: 我比較了幾次，儘管 Real-valued GA 都是由 whole-arithmetic crossover 得到較低的 fitness，Binary GA 卻各有千秋，所以在上述比較中，把兩者劃上等號。

	Real-valued GA			Binary GA		
	uniform-crossover	whole-arithmetic-crossover	which is better?	uniform crossover	2-point crossover	which is better?
演化結束後的最終 fitness (30 次的平均)	0.603343	0.222663	**whole-arithmetic crossover**	249.5029	307.4297	**uniform crossover**
最佳 fitness	0.098765	0.032147	**whole-arithmetic crossover**	32.5696	18.8803	**2-point crossover**

原因討論：

Convergence speed:

我認為 Convergence speed 會由該 GA 的 Exploration 來決定，也就是會**根據 crossover 的策略而有所差異**。

在 real-valued GA 之中，假設有下面兩個 parent：

parent x = [x1, x2…x10]

parent y = [y1, y2…y10]

如果使用 uniform crossover 的話，第一個 element 的產生**不是 x1 就是 y1**；而如果使用 whole-arithmetic crossover[$(\mathbf{pc})\mathbf{x1} + (\mathbf{1} - \mathbf{pc})\mathbf{y1}$]，則**同時使用到了 x1 以及 y1 的值**，所以**變化的幅度沒有 uniform crossover 還要來的大**，所以收斂的速度也比較慢一些。

至於在 binary GA 之中，我認為 uniform crossover 以及 2-point crossover **在每次做 crossover 所交換的 alleles 的期望值都是 5 個 bit**，所以對值的影響沒有差異太大，導致不會相差太大。

Solution quality:

我認為最主要產生差異的原因是來自於 **Representation 的不同**。

由於 Schwefel function 本來就是在實數的值域中尋找最佳解，如果以 Binary representation 的形式呈現，會**產生整數與小數之間的誤差**。

而經過了 500 個回合，誤差會積累到很誇張的地步，最後導致了 binary GA 以及 real-valued GA 之間 **Solution quality 的差距**。

而在各個 representation 當中為何會產生 solution quality 的差距的主因我認為是因為 **crossover 的策略不同**。

Real-valued GA 中，若採用 whole-arithmetic crossover，則可以同時混合 parent 的部分基因，相較於 uniform crossover 會有**更好的 exploitation**。從這裡和 convergence speed 的比較也可以看的出來，儘管 whole-arithmetic 擁有較差的 exploration，但相對的也有較好的 exploitation，這便是 **exploration 和 exploitation 之間的權衡**。

Binary GA 中，一樣每次 **crossover 交換的 alleles 的期望值都是 5 個 bit**，所以對值的影響沒有差異太大，導致不會相差太大。

4.

n 的影響：

比較方式：在不更改其他參數的情況下，把 n 分別設成 2、3、4、6、8，並各自取 30 次的平均做出圖表。

結果觀察：

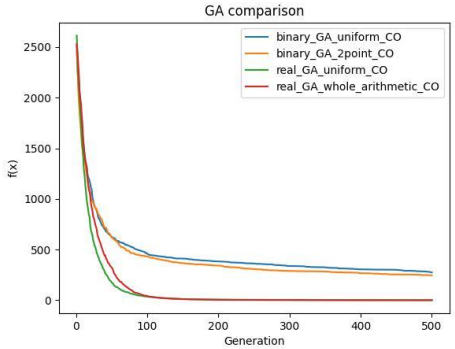
1. 隨著 n 增加，**耗費時間也跟著增加。**

原因：因為 n 增加代表一次 tournament selection 需要比較的 vector 數量增加了，所以耗費的時間也隨之增加。

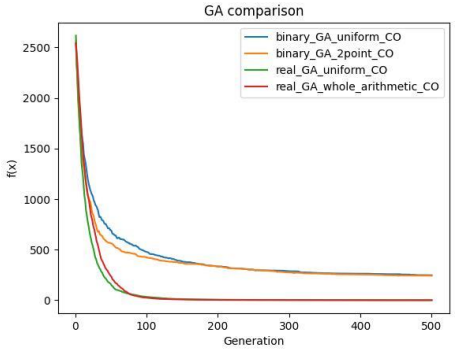
2. 在 n＝1 的情況下，**tournament selection 就變成了 random selection**，所以暫時不討論。

3. 隨著 n 的上升，**convergence speed 也逐漸加快，而 solution quality 也有些許提升**。

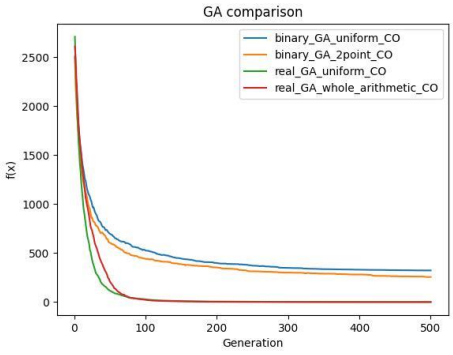
原因：增加了 n 代表在每次的 Tournament 中需要挑選的 individual 個數增加，這也代表選到 fitness 更加的 individual 的機會增加了許多，類似 elitism，所以收斂的速度以及最佳解的值也隨之進步許多。



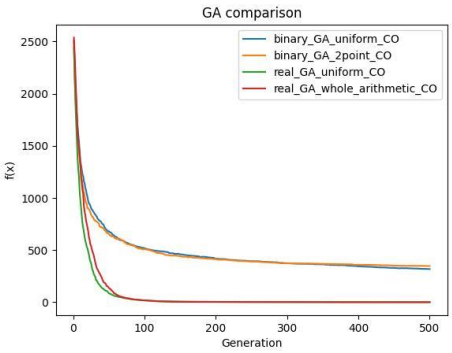
▲n = 2



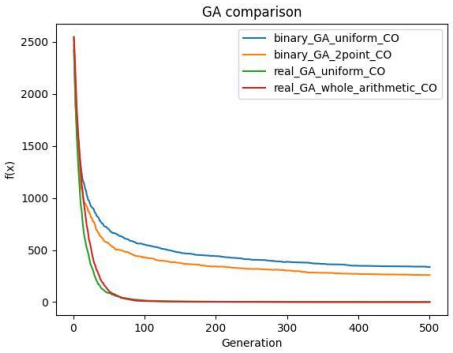
▲n = 3



▲n = 4



▲n = 6



▲n = 8

crossover rate 的影響：

比較方式：在不更改其他參數的情況下，把 crossover rate 分別設成 0、0.2、0.5、0.8、1，並各自取 30 次的平均做出圖表。

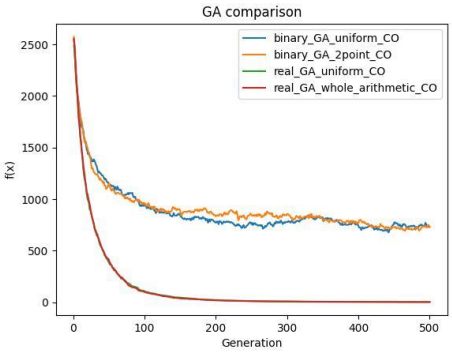
結果觀察：

1. Crossover rate = 0 時，上下波動很大，且 **Solution quality 以及 Convergence speed 都較低。**

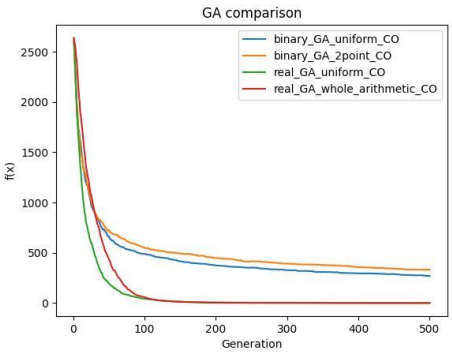
原因：因為在不進行 crossover 的前提下，能夠進行 evolution 的就只剩下 mutation 以及 selection operator。

2. 隨著 crossover rate 的增加，**convergence speed 也逐漸提升；solution quality 則沒有明顯的變化。**

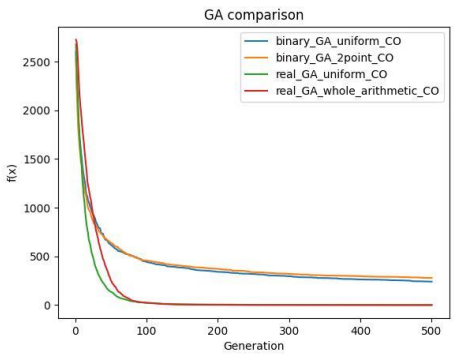
原因：增加了 crossover 的比例，也就**增加了 exploration，讓值變化的幅度增加許多**；再加上本身的 generational population model 以及 tournament selection，導致 fitness 時增時減。



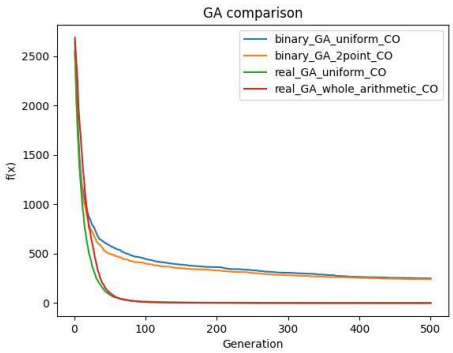
▲crossover rate = 0



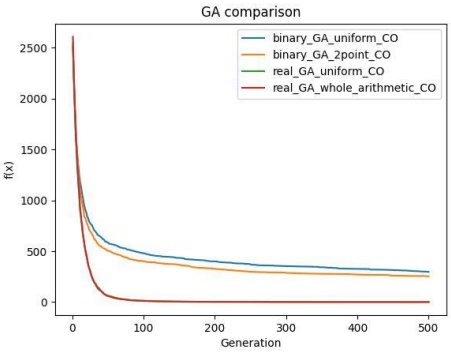
▲crossover rate = 0.2



▲crossover rate = 0.5



▲crossover rate = 0.8



▲crossover rate = 1

mutation rate 的影響：

比較方式：在不更改其他參數的情況下，把 crossover rate 分別設成 0、0.2、0.5、0.8、1，並各自取 10 次的平均做出圖表。

結果觀察：

1. 在完全不 Mutation(mutation rate = 0)的情況下，Solution quality 大幅減少，且會過早收斂。

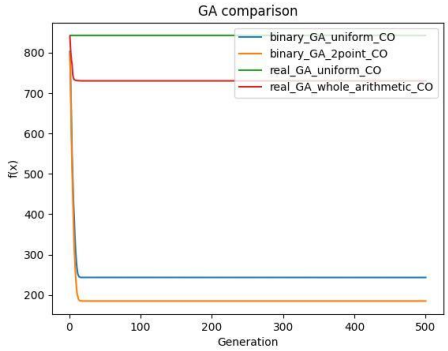
原因：推測是因為在沒有 mutation 的情況下，chromosomes 只剩下 crossover 來進行 variation。而 Crossover 儘管可以增加 exploration，但卻**沒有 operator 能增加 exploitation**，導致 solution quality 大幅下降；此外，只有 Crossover 很容易導致沒有 diversity 因而**被侷限在 local optima**。

2. 過高的 mutation rate 會導致 Convergence speed 以及 solution quality 嚴重下降。

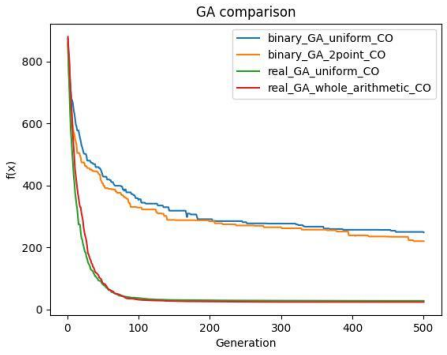
原因：過高的 mutation rate 反而會導致 mutation 發生太多次，進而讓**原本父母親遺留下來的基因沒辦法被繼承下去**，導致沒辦法找到最佳解。

3. 偏低的 mutation rate 會得到最好的結果。

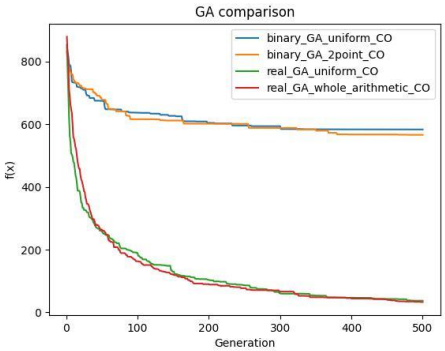
原因：適度的 mutation rate 能同時避免問題 1.以及問題 2.引發出來的結果，進而得到最佳解。



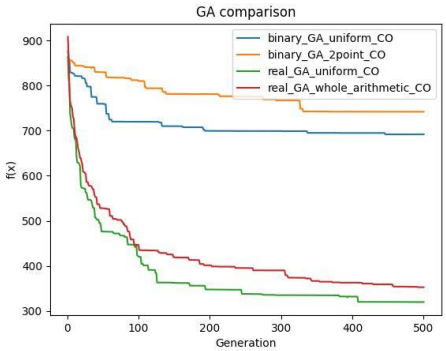
▲mutation rate = 0



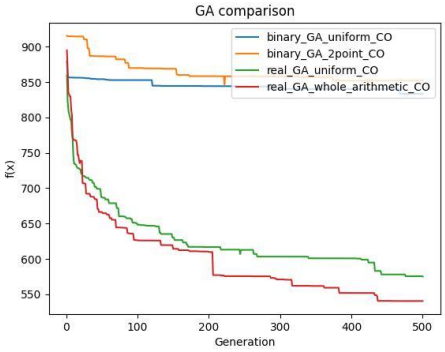
▲mutation rate = 0.2



▲mutation rate = 0.5



▲mutation rate = 0.8



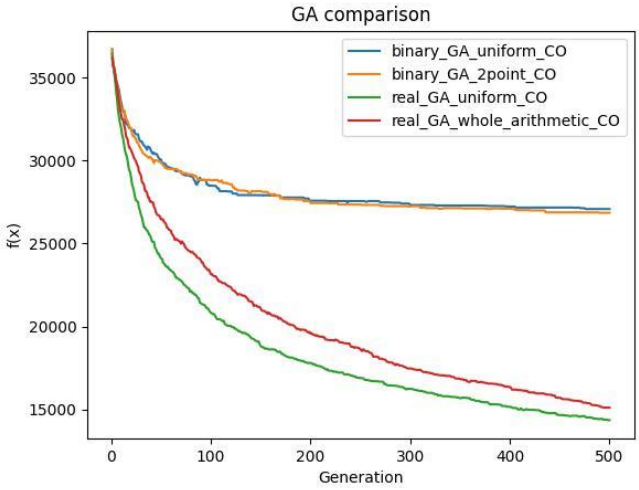
▲mutation rate = 1

5.

因為 100 維的計算很耗時間，所以我取 10 次作為平均。

很明顯的增加了許多執行的時間。

兩次的執行結果以及走向如下：



可以看到整體的 solution quality，**real-valued GA 依舊比 binary GA 的表現好很多**。

而 real-valued GA 中又以 **uniform crossover 最優**，再來才是 whole-arithmetic crossover。

而 binary GA 的話，則是 **2-point crossover 表現較良好**。

此外，從這張圖來看，很明顯的 binary GA 在大約 27000 附近時就逐漸收斂至 local optima 了；而 real-valued GA 則還在收斂，但因為限制只會演化 500 個 generation 而中斷。