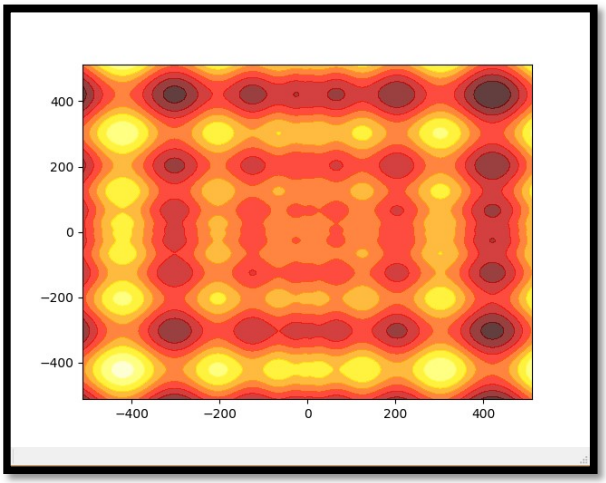
Evolutionary Computation

Program – GA in Numerical Optimization

Mar 26, 2019 107033530 陶郡賢

此次要作最佳化的 cost function 為 Schwefel function(SCH):

$f_{SCH}(\mathbf{x}) = 418.98291N - \sum_{i=1}^{N} x_i \sin(\sqrt{|x_i|})$，其中 $-512 \le x_i \le 511$，N 為維度



首先令 $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$，即 $N = 2$，來對 2 維平面不同點的 cost function 做驗證，可得此圖為 python 的輪廓圖，可以看到右上小處有最小值，透過 GA 計算其值約為 $\mathbf{x} = \begin{bmatrix} 420 \\ 420 \end{bmatrix}$附近，驗證了講義上 Fig1 的結果

接著我們**針對 $N = 10$ 的情況來分別做 Binary GA 與 Real-valued GA**，方式如下:

Representation: 因為每一維度的變數範圍在[-512~511]，總共有 1024 個數字，等同於 2 的 10 次方，故在 Binary GA 中，每個變數會使用 10 個 binary bit 來表示，而此處 $N = 10$，故每個體包含 100 個 binary bit。而在 Real-valued GA 則每一維度的變數直接就是一個基因，每個體有 10 個基因。

Parent Selection: 此處使用 Tournament Selection，且 n=2。每次從所有人口當中隨機挑選 2 人做競爭，擁有較佳(小)適應值者，可以被選中並被丟入交配池中。如此競爭直到選出與人口數等量的人數，接著便可進行交配。

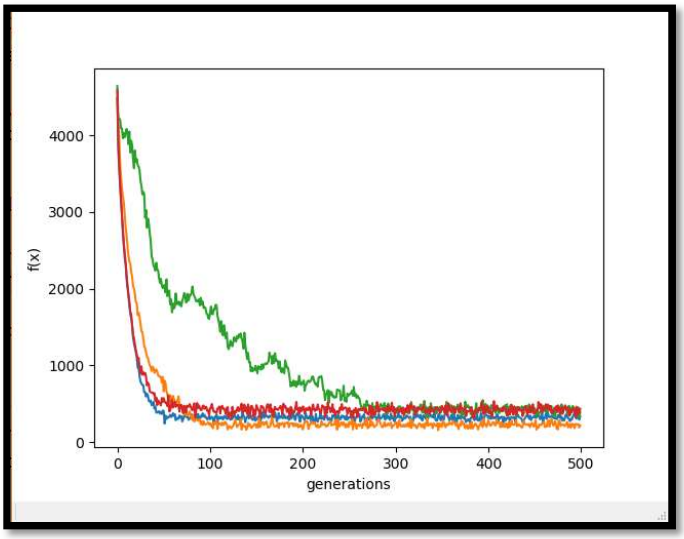
Mutation: 在 Binary GA 中使用 Bit-flip，也就是每個基因以 $1/\ell$ 的機率會做 flip。而在 Real-valued GA 中使用 Uniform，也是每個基因以 $1/\ell$ 的機率會隨機在[-512,511]當中產生實數。

Crossover: 每對親代有 $P_c = 0.9$ 的機率會做交配。在 Binary GA 有 Uniform 與 2-point 的方式，其中 Uniform 的每個子代的每個基因都有 50%的機會來自父親或母親，2-point 則是隨機產生 2 個切點，切點的前段、中段與後段可以來自父親、母親與父親或反之。Real-valued GA 有 Uniform 與 Whole Arithmetic 的方式，Uniform 與上相同，Whole Arithmetic 則是利用父親與母親基因的線型組合來產生子代。

Survivor Selection: $\mu + \lambda$ 親代與子代會共同競爭，每次產生完子代後，人口數會變為原本的 2 倍，這時要從全部人口中，挑出適應值排名在前段的人口才能存活下來，其餘則被淘汰掉。

以下會**針對 Binary GA 和 Real-valued GA 在不同 Crossover 方式下的結果**:

Binary GA		Real-valued GA	
Uniform Crossover	2-point Crossover	Uniform Crossover	Whole Arithmetic
橘色	藍色	紅色	綠色

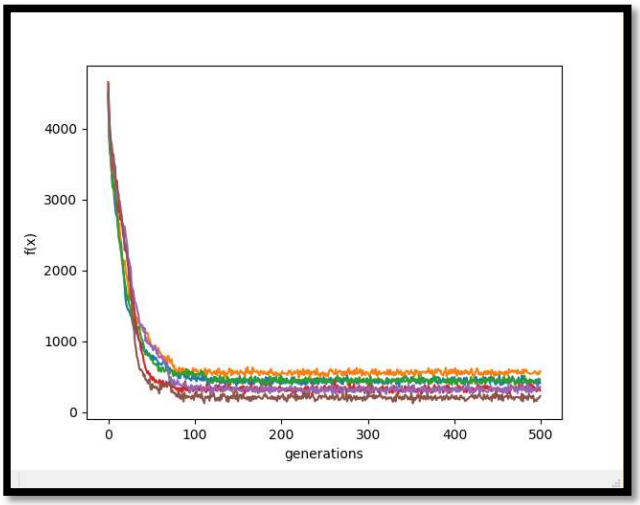


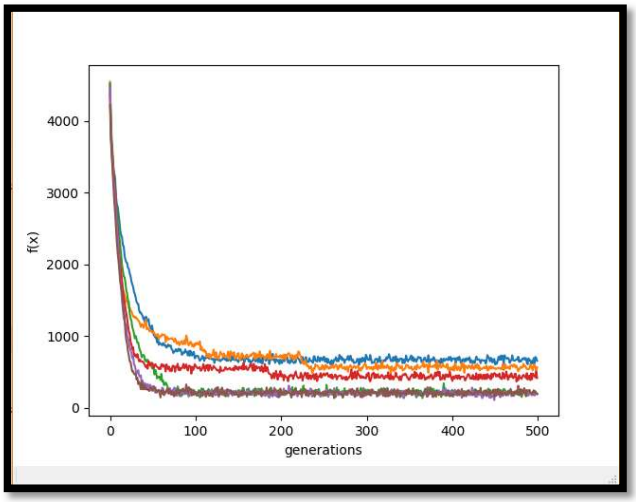
由結果可以發現，在前期收斂速度最快的是 Binary GA 的 2-point Crossover，最慢的是 Real-valued GA 的 Whole Arithmetic。而最終收斂後可以得到最佳(小)平均適應值的為 Real-valued GA 的 Uniform Crossover，得到最差(大)平均適應值的為 Real-valued GA 的 Whole Arithmetic。我認為 Binary GA 普遍有較佳收斂結果的原因可能是以 bit 為單位來做交配與突變，會有很大的機會跳脫原本親代的搜索區域，而找到其他的可能解，也就是對於 exploration 有較大的助益。反之 Real-valued GA 的 Whole Arithmetic 都是親代的線性組合，產生的子代也介於親代之間，故難以完整開發完整個搜索空間，容易過早的收斂到親代已找到的可能解附近。

以下**針對 Binary GA，使用不同的交配率做測試**

交配率	0.1	0.3	0.5	0.7	0.9	1.0
顏色	藍色	橘色	綠色	紅色	紫色	咖啡色

左圖為 Binary GA 的 Uniform Crossover，右圖為 Binary GA 的 2-point Crossover



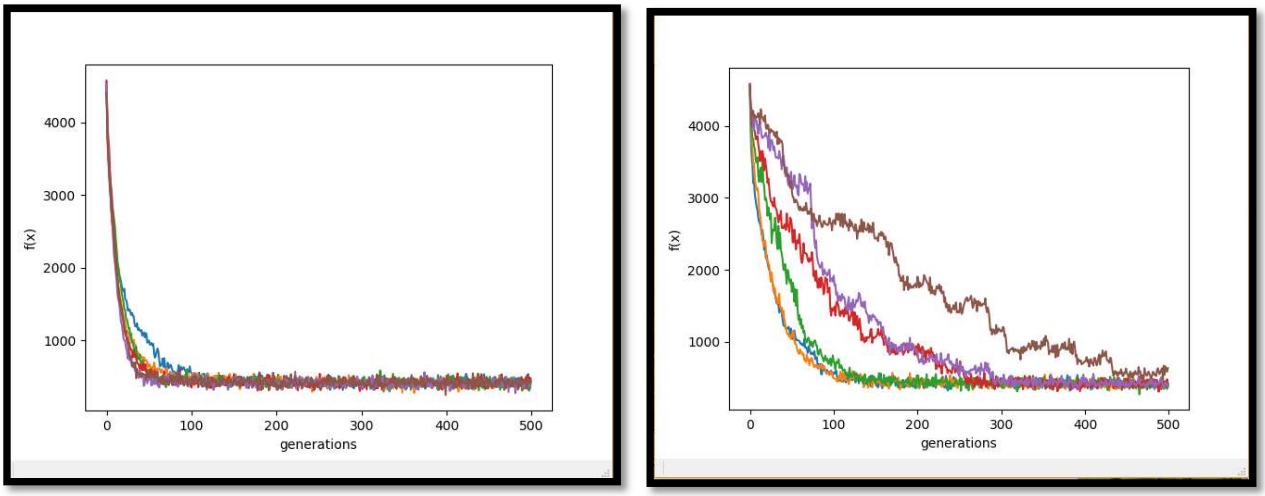


以大致的結果來看，交配率越高，最終收斂結果越好，不過以前期收斂速度來看，交配率越低前期收斂速度越快。這很有可能是因為大家很少繁衍，持續把目前找到的較佳基因遺傳下去，某些基因持續保留而壟斷了繁衍，最終收斂到的值只在 local optima 而非 global optima。

以下**針對 Real-valued GA，使用不同的交配率做測試**

交配率	0.1	0.3	0.5	0.7	0.9	1.0
顏色	藍色	橘色	綠色	紅色	紫色	咖啡色

左圖為 Real-valued GA 的 Uniform Crossover，右圖為 Real-valued GA 的 Whole Arithmetic Crossover

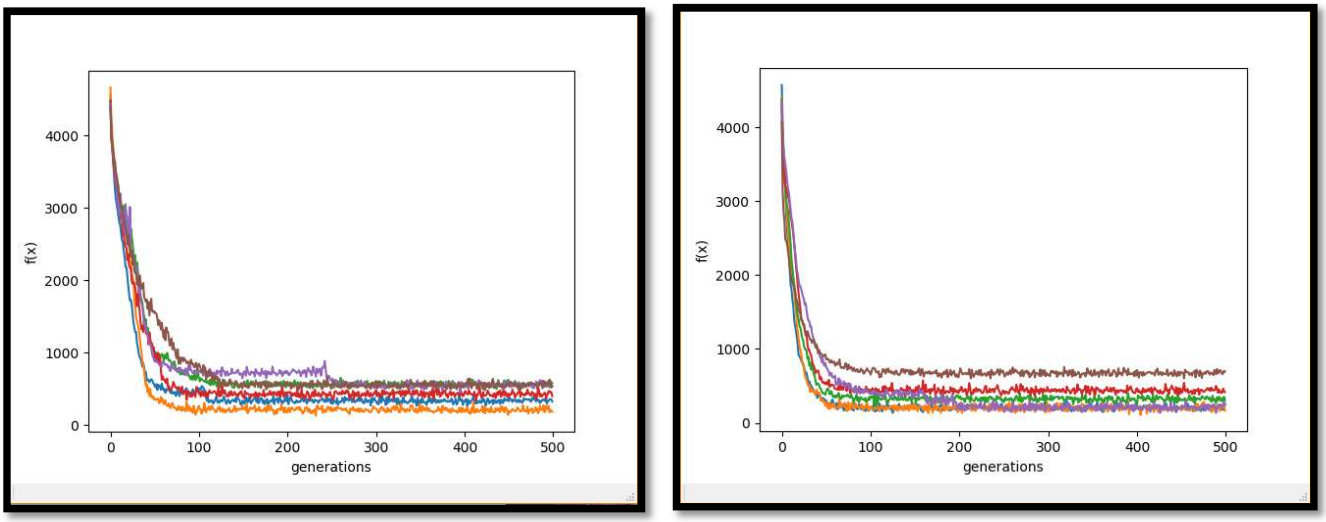


結果發現 Uniform Crossover 的收斂結果對於交配率的影響不大，但仍能有效的增加收斂速度，可能是因為數值與數值之間相關性較小。而 Whole Arithmetic Crossover 在交配率提升下，收斂結果變得更差了，可能是原本適應值偏高的個體受到適應值偏低個體影響，導致線性組合後產生的個體未必較佳，使得曲線收斂速度變得越來越慢。

以下**針對 Binary GA，使用不同 Tournament Selection 數量**:

Select n	1	2	5	10	20	30
顏色	藍色	橘色	綠色	紅色	紫色	咖啡色

左圖為 Binary GA 的 Uniform Crossover，右圖為 Binary GA 的 2-point Crossover

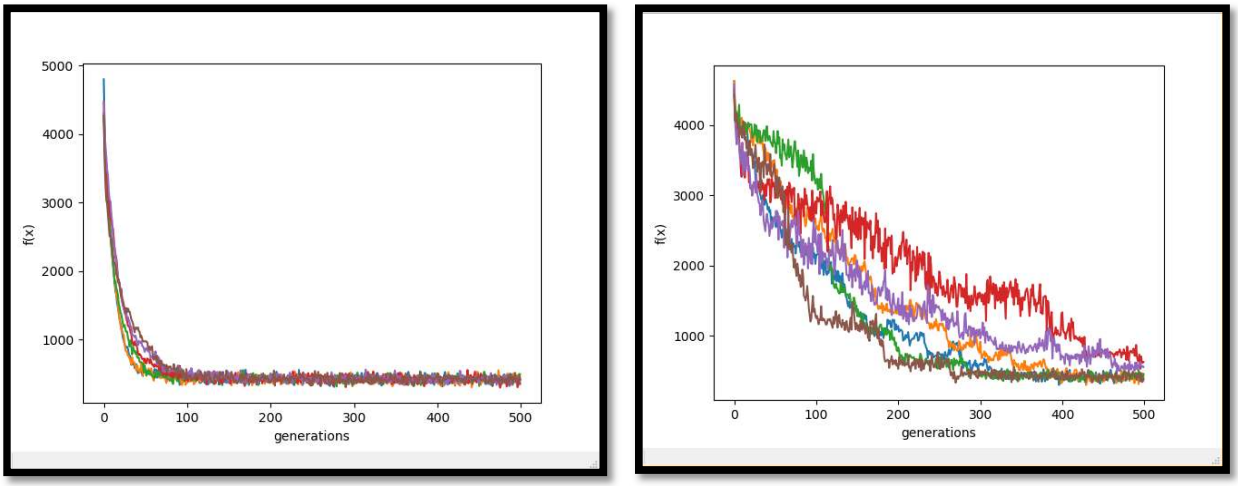


當 n=1 時，即是隨機選擇。結果發現約在 n=2 時最終收斂結果最好，前期收斂速度 n=1,2 差不多。然而當我把 n 逐漸調大時，其收斂結果都不盡理想，我認為可能是當 n 很大時，某些 local optima 就有較大的機率被選到，並且競爭下又具有較大的優勢，所以這些 local optima 就不斷被選中而壟斷市場，最後過早收斂，使得收斂結果反而較差。

以下**針對 Real-valued GA，使用不同 Tournament Selection 數量**:

Select n	1	2	5	10	20	30
顏色	藍色	橘色	綠色	紅色	紫色	咖啡色

左圖為 Real-valued GA 的 Uniform Crossover，右圖為 Real-valued GA 的 Whole Arithmetic Crossover

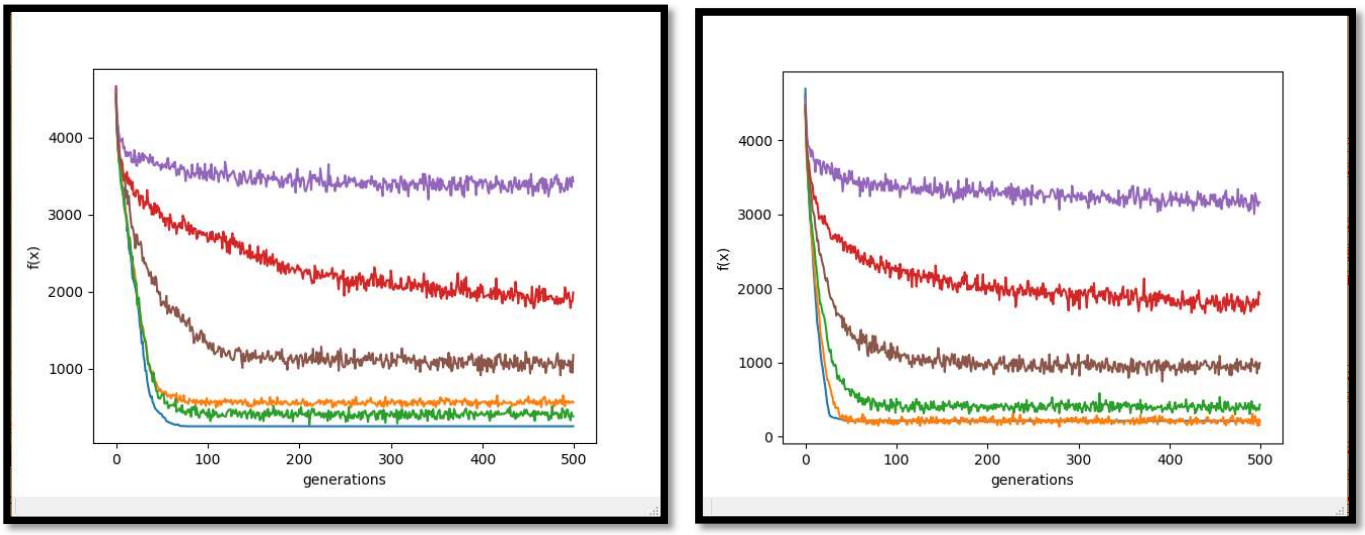


Uniform Crossover 受到 Tournament Selection 數量的影響也不大，表示 Uniform Crossover 不論交配池中的基因情況優劣，結果都相差無幾。Whole Arithmetic Crossover 在 Tournament Selection 數量很大時，也有不錯的表現，可能是因為當配給交配池的基因很優良時，透過線性組合，也可以使子代在優良基因得附近，使得交配完的子代結果不至於偏差太大。

以下**針對 Binary GA，使用不同 Mutation rate**:

突變率	0	$1/\ell$	$2/\ell$	$5/\ell$	$10/\ell$	$20/\ell$
顏色	藍色	橘色	綠色	咖啡色	紅色	紫色

左圖為 Binary GA 的 Uniform Crossover，右圖為 Binary GA 的 2-point Crossover

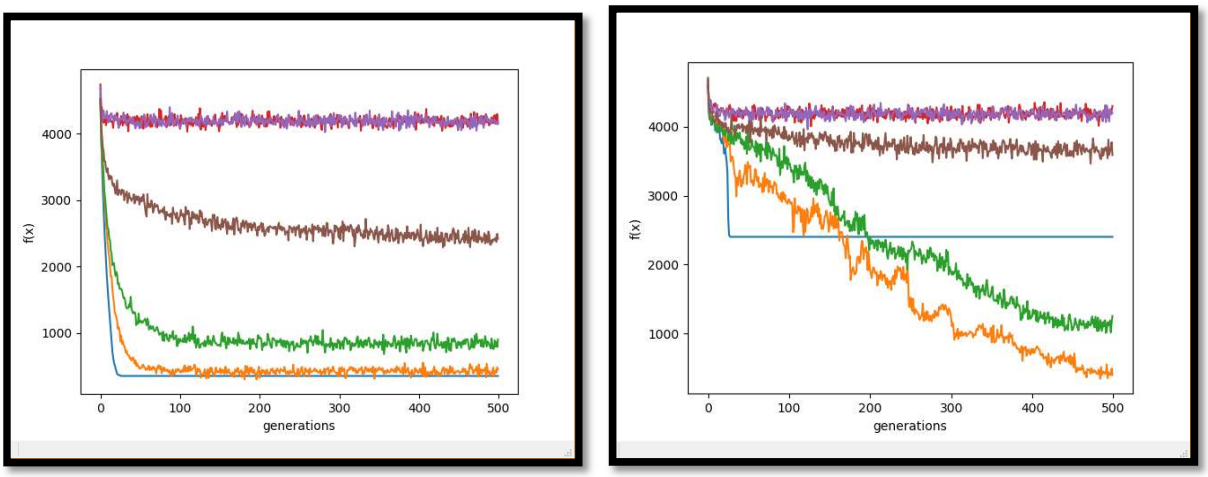


突變的結果會使得平均的適應值起伏而產生圖中曲線的震盪，在沒有突變時，其收斂速度最快，且最終收斂結果最佳，並且少有震盪，在此問題中，也許沒有突變效果會較佳，但也可能因為沒有搜索到 global optima 而收斂到錯誤的地方。以這個情況下來看，突變率越高，其最終收斂結果越差。

以下**針對 Real-valued GA，使用不同 Mutation rate**:

突變率	0	$1/\ell$	$2/\ell$	$5/\ell$	$10/\ell$	$20/\ell$
顏色	藍色	橘色	綠色	咖啡色	紅色	紫色

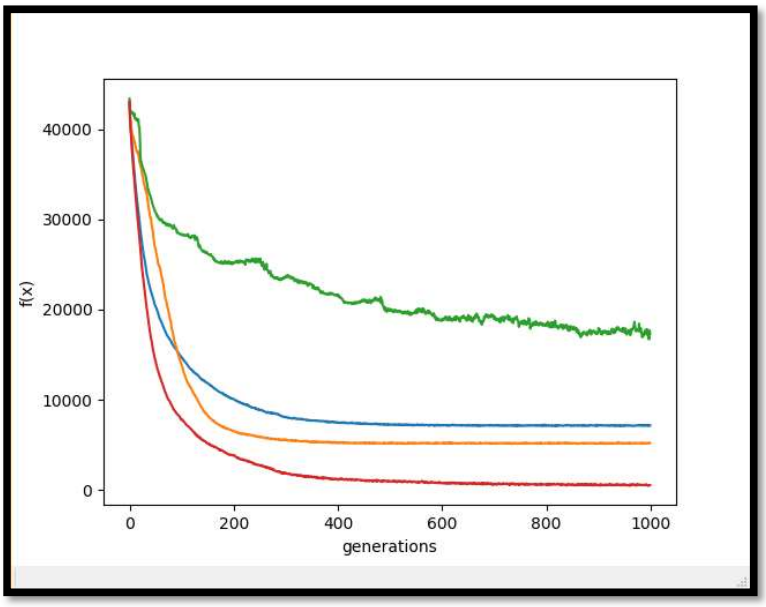
左圖為 Real-valued GA 的 Uniform Crossover，右圖為 Real-valued GA 的 Whole Arithmetic Crossover



Uniform Crossover 在突變率增加時，最終收斂結果明顯越變越差，可見此突變方法並不適合 Real-valued GA，可以考慮改為 Non-uniform mutation。Whole Arithmetic Crossover 在沒有突變的狀況下，找到了錯誤的 local optima 並且收斂，而得到錯誤的結果，而小的突變，雖然會使演進的收斂速度下降，最卻在最終收斂時，可以找到比沒有突變更好的收斂值，由這張圖，便可以清楚的看到我們之所以要做突變的原因。

以下為在 **N=100 的高維度，針對 Binary GA 和 Real-valued GA 在不同 Crossover 方式下的結果**:

Binary GA		Real-valued GA	
Uniform Crossover	2-point Crossover	Uniform Crossover	Whole Arithmetic
橘色	藍色	紅色	綠色



在解高維度的問題時，就可以明顯感受到使用 Real-valued GA 在執行上會快了許多，畢竟少了分段做解碼的步驟，與之前 N=10 情況類似，Binary GA 的 Uniform Crossover 前期收斂速度較慢，但後期收斂值比 2-point Crossover 更好。表現特別突出的是 Real-valued 的 Uniform Crossover，不但有快的收斂速度和執行效率，最終收斂值也較佳，我認為在高維度的實數空間當中，使用 Real-valued 的 Uniform Crossover 是有效率的。