演化計算 → 由基因演算法而來

evolutionary algorithms, genetic algorithms, evolution strategy, evolutionary programing, artificial life

Optimization: 最佳化 question

1. specific optimization (differential method
   gradient method)
   ↳ 用在特定problem

2. general .... (random search, GA)
   ↳ 用在通用problem
   基因演算法

P.S.O

Ant C.

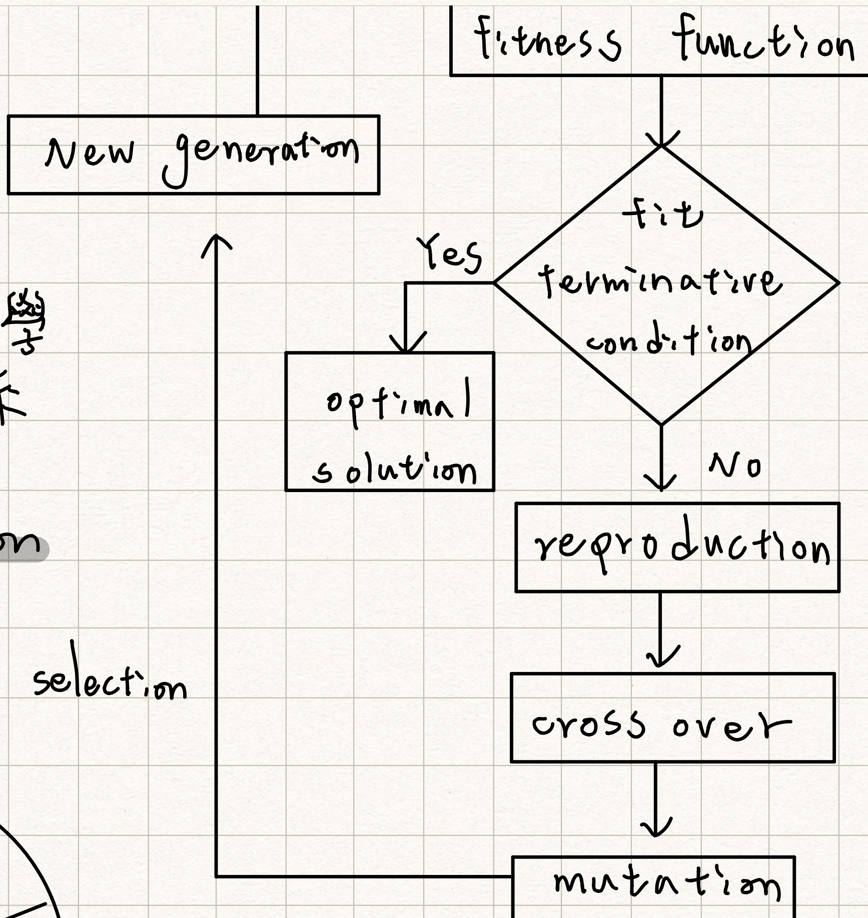
Δ. G. A (1975, Holland)

operators: 1. reproduction
   複製
2. crossover
   交配
3. mutation
   突變

generate initial generation randomly

↓

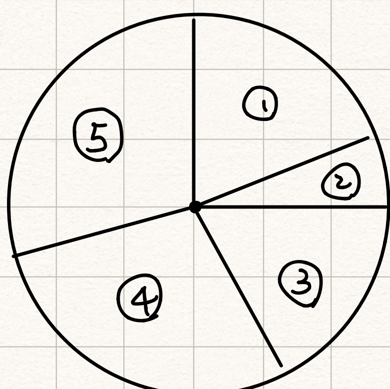
decode (binary) string and calculate



由遺傳基因學等發展而來

1. reproduction

① roulette wheel selection



間隔(面積)大小由 fitness function 來決定(適應值)

ex: 5、3、4、1、5

② tournament selection

Randomly select two or more chromosomes from parent generation and put the chromosome with highest fitness value in the mating pool.

Repeat N times, where N is the population size.

似人口，要維持不變

撈K個，選一個最大的，撈至N個數為止

可重複。

ex: 5、1、4、3、4

2. crossover

① one-point crossover

A = 0 0 0 0 0 ⟶ A' = 0 0 1 1 1

B = 1 1 1 1 1 ⟶ B' = 1 1 0 0 0

※2

① 五個位置，在#2後相互交換

② or #2開始交換

固定做法即可，but 做法不限定

② two-point crossover

0	0	0	1
0	1	1	0
1	0	1	0
1	1	0	1

※2 ※4

$$A = 0\ 0\ 0\ 0\ 0 \longrightarrow A' = 0\ 1\ 1\ 1\ 0$$
$$B = 1\ 1\ 1\ 1\ 1 \qquad\quad B' = 1\ 0\ 0\ 0\ 1$$

介於2~4之間的點進行交換

③ uniform crossover (mask crossover)

$$A = 0\ 0\ 0\ 0\ 0 \xrightarrow[\quad]{\text{mask } 00101} A' = 0\ 0\ 1\ 0\ 1$$
$$B = 1\ 1\ 1\ 1\ 1 \qquad\quad B' = 1\ 1\ 0\ 1\ 0$$

mask由亂數產生，看其為1的地方
交換數字

以上的※2，※4等等皆由亂數產生

3. mutation

① bit mutation

$$A = 0\ 1\ 0\ 1\ 0 \xrightarrow{\text{bit mutation } ※4} A' = 0\ 1\ 0\ 0\ 0$$

② string mutation

A= 0 1 0 1 0 $\longrightarrow$ A' = 1 0 1 0 1

全部倒反

③ uniform mutation

A= 0 1 0 1 0 $\xrightarrow[\text{※3 ※5}]{\substack{\text{mask} \\ 00101}}$ A' = 0 1 1 1 1

example: $f(x) = x^2$, $0 \le x \le 31$ (32個可能)

find x such that f(x) is maximum

(求極值, 一次微分等於 0)

基因演算法解之, 先進行編碼

0 0 0 0 0 $\longrightarrow$ 0

0 0 0 0 1 $\longrightarrow$ 1

⋮

1 1 1 1 1 $\longrightarrow$ 31

let population size = 4

step 1 : initial generation

\#1 : 0 1 1 1 0 ⟵ 14
\#2 : 1 1 0 0 0 ⟵ 24
\#3 : 1 0 0 0 1 ⟵ 17
\#4 : 0 0 1 1 1 ⟵ 7

encode

fitness function
(fitness value)

$$f_{fit}(x) = f(x) = x^2$$

$$= 2f(x)$$ 可有多種function

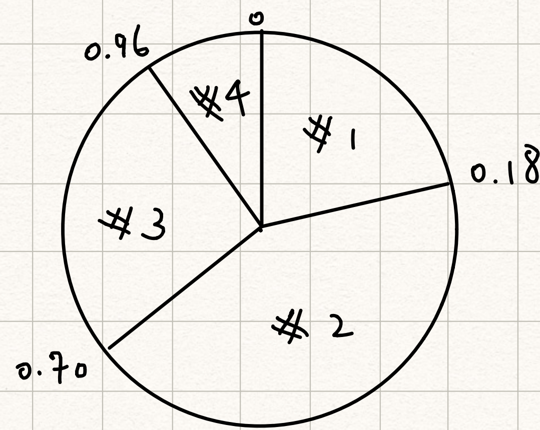
越符合答案的值，它的fitness value便越大

找出fitness value

14 : 196
24 : 576
17 : 289
7 : 49

統計%，將各數所佔比例得出

step 2 : reproduction (RWS)

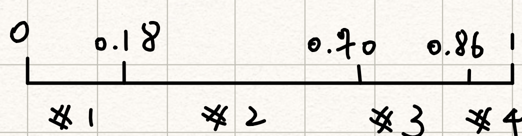


按百分比將※N畫入一圓餅圖中

照百分比
取和population一樣的數量

① 0 1 1 1 0 (※1)
② 1 1 0 0 0 (※2)
③ 1 1 0 0 0 (※2)
④ 1 0 0 0 1 (※3)

程式的寫法



0~1 隨意取值，取到何

Step 3 : crossover

let
crossover rate = 1 (100%)

0 < c.r. ≤ 1，表 population 中有多少比例能進行交配。有些人會和下一世代交配

For large population(100)
c.r. = 0.6
60 + 40，60 去入交配池進行計算

For small population(30)
c.r. = 0.9

一定都是兩兩交配

		bit※2					
②	1	1	0	0	0		
①	0	1	1	1	0		
②'	1	1	1	1	0	30	900
①'	0	1	0	0	0	8	64

值便是選到它惹

取到 0.18 為 ※1　依此
　　 0.70 為 ※2　類推

step 4 : mutation

let
mutation rate = 0.01

5 × 4 = 20 bit/gen
每一 generation 有 20 個 bit

20 × 0.01 = 0.2 bit/gen

不足 1，怎辦？，使之等於 1 時再突變即可
(5th generation)

隨機得到 0 1 0 0 0（※3 標於第三位）

↓

0 1 1 0 0

以單點突變之

突變後的人要取代原

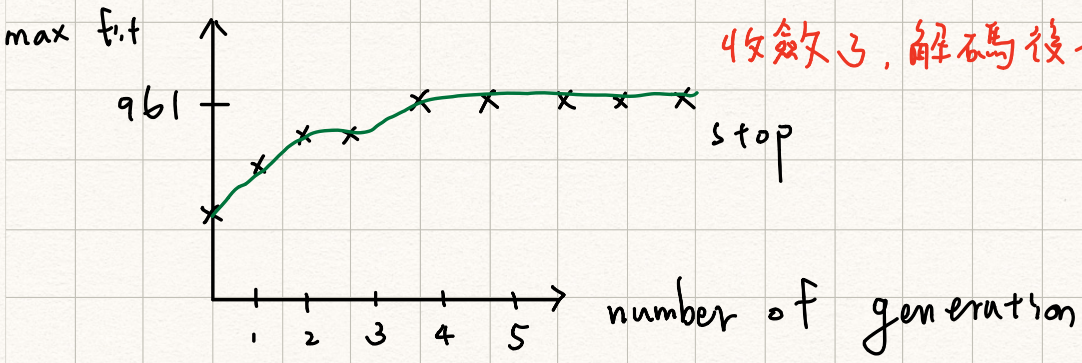
	bit #4					
④ 1	0	0	0	1		
③ 1	1	0	0	0		
④' 1	0	0	0	0	16	256
③' 1	1	0	0	1	25	625

本人的位置

突變用意在於突破區域最佳解的狀況
→基因演算化具備全域最佳解能力

回去reproduction

做題目時要畫出fitness value的趨勢圖
譬如找出每個generation中的Max值 or 平均
隨世代變換的結果



收斂了，解碼後得解☺

不一定要找出全域最佳解，若能允許誤差，則不必尋找最佳解

以實際測試來判斷此解是否OK
(若客戶、老闆皆可，那此解也不錯)

Δ. Characteristic of GAs :

1. search encoded parameters, not the parameters.

2. avoid local optimum.

3. use fitness function to determine the goodness.

4. use probability method instead of crisp rules.

Δ. Affecting factors of GAs:

1. parameter setting

① coding range
② string length
③ population size
④ crossover rate
⑤ mutation rate
⑥ fitness function

2. encoding & decoding

x_1 : 0 0 0 1 1 $\frac{3}{31}$

x_2 : 0 1 1 0 0 $\frac{12}{31}$

x_3 : 0 0 0 0 1 $\frac{1}{31}$

更多位元
→ 解析度增加
→ 答案越精確

$$f(x_1, x_2, x_3) = (x_1 - x_2)^2 + (x_3 + x_2)^2$$

find x_1, x_2, x_3 such that f is maximum, $0 \le x \le 1$

3. string length

① large length has higher accuracy, but takes more computation time

② length depends on the required accuracy

4. crossover rate

① higher rate may cause better

chromosomes to be changed via crossover process (好 × 壞 → 變壞了)

② lower rate may show down search process

5. mutation rate

mutation rate must be small, otherwise it will become random search

1	1	1	0	1
1	1	1	0	1
1	1	1	0	1
1	1	1	0	1

→ 交配後恆為定值

6. avoidance of local optimum

① annealing crossover rate and mutation rate (即為 crossover rate 和 mutation rate 不維持一致)

② increasing c.r. and m.r. while evolution process stop

7. fitness function

① problem dependent (=題目敘述)

② increasing the difference betwee best chromsome and better chromosome

31 → 961　　　　30 → 900

$[f(x)]^2$, 加大 range 使之明顯

停止條件：population 長得一樣
or 誤差在範圍內 (ex: 0.05%)

精度要高，編碼數要變長 → 用實數來做？

Δ. Real-Valued GAs

chromosome type: $\underline{X} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_p \end{bmatrix}$

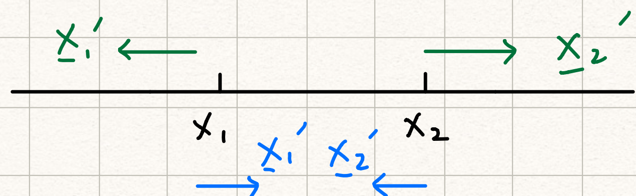
$x_1 \sim x_p$ 是不同的變數

Reproduction (same as binary GAs)

Crossover: for $x_1 < x_2$, $0 < \sigma < 1$

$$\begin{cases} \underline{x}_1' = \underline{x}_1 + \sigma(\underline{x}_1 - \underline{x}_2) \\ \underline{x}_2' = \underline{x}_2 - \sigma(\underline{x}_1 - \underline{x}_2) \end{cases}$$ [pull away]

σ 亂數抓，每次不一樣



轉成 $-1<\sigma<1$ 便可用一公式做即可

$$\begin{cases} \underline{x}_1' = \underline{x}_1 + \sigma(\underline{x}_2 - \underline{x}_1) \\ \underline{x}_2' = \underline{x}_2 - \sigma(\underline{x}_2 - \underline{x}_1) \end{cases}$$ [get closer]

A1: 1 0 1 0 1 0 1 0 (170) → A1': 1 0 0 0 0 0 1 0 (130)

A2: 0 0 0 0 0 0 0 0 (0) → A2': 0 0 1 0 1 0 0 0 (40)

A3: 1 0 1 1 1 1 1 0 (190) → A3': 1 0 1 0 1 0 1 0 (170)

A4: 1 1 1 0 1 0 1 1 (235) → A4': 1 1 1 1 1 1 1 1 (225)

觀察binary的crossover，出來的結果不是靠近就是拉遠，因此實數公式也是如此做的

Mutation:

A_1: 10101010 (170) → A_1': 10111010 (186)

A_2: 10101010 (170) → A_2': 01010101 (85)

突變後，不是變大就是變小

$\underline{x}' = \underline{x} + s \times$ random_noise (a vector)

noise $\in [-1 \quad +1]$, s is positive value

隨x值大小選擇

problem dependent

Δ. Evolutionary Algorithm (EA) $\quad \underline{X} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_p \end{bmatrix}$

Reproduction (same as binary GAs)

crossover: $\underline{x}_j$ and $\underline{x}_k$ are parents

① average crossover

$$\underline{x} = \frac{1}{2}\{\underline{x}_j + \underline{x}_k\}$$

② convex combination

$$\underline{x} = r\underline{x}_j + (1-r)\underline{x}_k$$

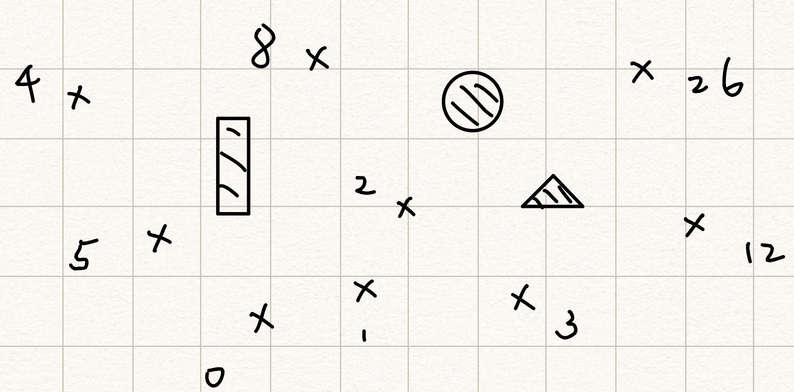
$r \in (0, 1)$ random

mutation:

$$\underline{x}' = \underline{x} + r\underline{d} \quad , \; r \in (0,1)$$

random

<EX> **Shortest Path**



從0走到26，中間有許多障礙物
每個節點為一種走法

兩者需有重複 (15)

crossover

parent 1 : 0 - 1 - 2 - 11 - 15 - 16 - 21 - 24 - 26

parent 2 : 0 - 4 - 8 - 10 - 15 - 20 - 23 - 26

⇓

.... 1′ : 0 - 1 - 2 - 11 - 15 - 20 - 23 - 26

.... 2′ : 0 - 4 - 8 - 10 - 15 - 16 - 21 - 24 - 26

如發生 0 - 1 - 3 - 9 - 11 - 15 - 10 - 7 - 11 - 26

11 ~ 11 間的 15.10.7
要刪掉，11、11 合併

mutation

↓
0 - 1 - 2 - 3 - 9 - 11 - 15 - 20 - 23 - 26

0 - 1 - 2 - 3 - 9 - 11 - 14 - 22 - 16

隨機抓取一個，其後 random

finess function 設定：D = d + d + ... (路徑和)

fitness = D^{-1} (越好的須越大)

〈Ex〉最佳控制序列

$$x(k+1) = Ax(k) + B\underline{u}(k)$$
$$Y = C\underline{X}(k) + Du(k)$$

$$u(0),\ u(1),\ u(2),\ \cdots\ u(k)\ ?$$

$$e = Y_d - Y$$

螞蟻演算法、PSO