1. GAs: What are They ?

Hillclimbing, Simulated Annealing, and Genetic Algorithms

◎ 문제 : Max f(V) = | 11 one(v) - 150 |

V : binary strings of length 30.

one(V) : number of 1s in the string V

(ex) V1 = (110110101110101111111011011),

V2 = (111000100100110111001010100011),

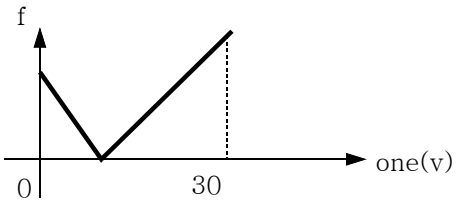
V3 = (000010000011001000000010001000),

f(V1) = |11·22 -150| = 92

f(V2) = |11·15 -150| = 15

f(V3) = |11·6 -150| = 84

(graph)



* Global maximum

Vg = (111111111111111111111111111111)

f(Vg) = |11·30 -150| = 180

* Loval maximum

Vl = (000000000000000000000000000000)

f(Vl) = |11·0 - 150| = 150

◎ Hillclimbing Algorithm

```
begin
   t <- 0
   repeat
      local <- FALSE
      select a current string Vc at random
      evaluate Vc
      repeat
            select 30 new strings in the neighborhood of Vc
               by flipping single bits of VC
            select the string Vn from the set of new strings
               with the largest value of objective function f        (Vn 생성)
               if  f(Vc) < f(Vn)
                  then Vc <- Vn                          (Vn 이 크면 계속 진행)
                  else local <- TRUE                     (Vn 이 작으면 Stop)
      until local
      t <- t+1
   until t = MAX
end
```

○ Neighborhood

	1	0	·····	0	1
	↓	↓		↓	↓
Vc	0	1		1	0

○ 초기의 Vc 에 따라 local (one(Vc) <= 13 경우) 또는 global maximum (one(Vc)>13 경우) 에 접근

◎ Simulated Annealing Algorithm

```
begin
    t <- 0
    initialize temperature T
    select a current string Vc at random
    evaluate Vc
    repeat
        repeat
            select a new string Vn
                in the neighborhood of Vc
                by flipping a single bit of Vc                    (Vn 생성)
                if  f(Vc) < f(Vn)
                  then Vc <- Vn                          (Vn 이 크면 계속 진행)
                  else if random[0,1) < exp{ f(Vn) - f(Vc)) / T }
                          then Vc <- Vn                                (1)
        until (termination condition)
        T <- g(T,t)                                                    (2)
        t <- t+1
    until (stop-criterion)
end
```

(1) Vn 이 Vc 보다 작은 경우, 0 -1 사이의 난수 r 을 발생시킨다. 이 때 r < p
(단, p = exp{ f(Vn) - f(Vc)) / T }) 이면 Vc<-Vn 으로 하여 계속 진행.
=> random walk
=> local maxima 를 빠져나올 수 있다.

* p 값이 크면 즉,
  - f(Vn) 과 f(Vc) 의 차이가 작은 경우,
  - T 가 큰 경우 random walk 의 확률이 높다.

(2) g(T,t) < T for all t.
=> T 값이 점차 작아진다.
=> random walk 의 확률이 작아진다.

+ Annealing (담금질) : 서서히 온도를 낮추며 안정상태 진입

(ex) Vc = (11100000010011011100101010000)

one(Vc) =12

f(Vc) = |11·12-150| = 18

이고, 새로이 선택된 Vn 이 다음과 같을 때

one(Vn) = 13

f(Vn) = 7

(1) Hillclimbing

- Vn 이 선택될 수 없다.
- local maxima Vl 방향으로 진행 (∵ one(Vc) ≤ 13)

(2) Simulated annealing

- p = exp((f(Vn) - f(Vc)) / T)

  = exp (-11/20) (if T=20)

  = 0.57695

- Vn 이 선택될 확률이 57.7%

* Genetic Algorithm (Idea)

Population of string :

v5 = (11111000000|01101110011101000000) : f(v5) = 16

v6 = (00000000000|11011100101011111111) : f(v6) = 16

=> crossover

v5' = (11111000000|11011100101011111111) : f(v5') = 59

★ Kangaroo 의 히말라야 최고봉 정복

- Hill-Climbing

처음 출발한 지점에서 가장 가까운 산 꼭대기를 최선을 다하여 오른다. 그러나 이 산이 최고봉인지는 알 수 없다.

- Simulated Annealing

캉가루가 술에 취하여 히말라야의 이 산 저 산을 배회하며 산을 오른다. 점차 술기운이 떨어지면서 정상에 오른다.

- Genetic Algorithm

여러마리의 캉가루를 히말라야 주변에 낙하산으로 투하시킨다.
캉가루들은 그들이 왜 히말라야에 투하되었는지 모른다.
그러나 낮은 지점에 있는 캉가루에 대한 사냥을 시작하면, 높이 오를 수 있는 캉가루들만이 생존하며 이들의 후손에 의하여 정상이 정복된다.

GA : What are they ?

○ Optimization of a simple function

Maximize f(x) = x sin(10π x) + 1.0

where x ∈ [-1, 2]

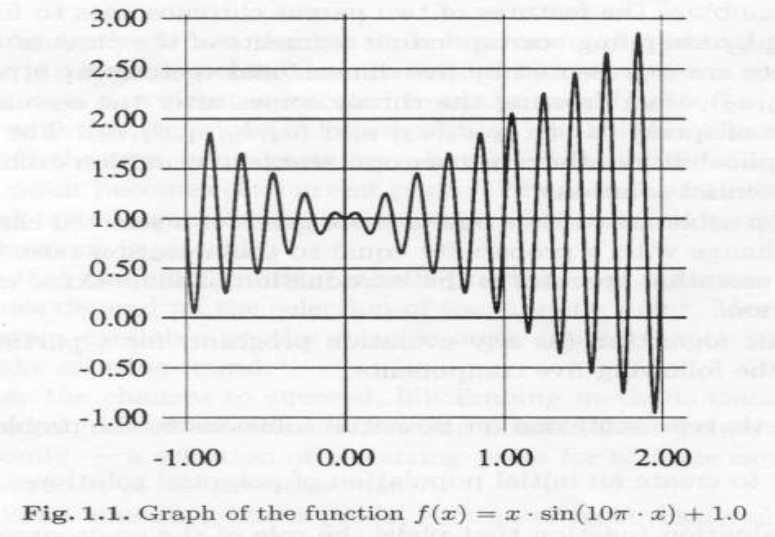


Fig. 1.1. Graph of the function $f(x) = x \cdot \sin(10\pi \cdot x) + 1.0$

■ Representation : **chromosome (염색체), gene (유전자)**

- x 를 binary vector (= string = chromosome = 염색체) 으로 표현
- vector length : Precision 에 따라 결정
- x 를 소숫점이하 6 자리까지 표시할 때,

size length (총 string 의 수) = (2-(-1)) * 1,000,000 = 3,000,000

$2097152 = 2^{21} < 3000000 \leq 2^{22} = 4194304$

∴ 22 bit 필요

* binary vector -> 실수 변환

(i) $(<b_{21}\ b_{20}\ \ldots\ b_0>)_2 = \left(\sum_{i=0}^{21} b_i \cdot 2^i\right)_{10} = x'$

(ii) $x = -1.0 + x' \cdot 3/(2^{22}-1)$

(ex) (1000101110110101000111)

x' = 2288967

x = -1.0 + 2288967 * 3 / 4194303 = 0.637197

* **gene** (유전자) : binary vector (염색체) 의 각 bit

■ Population

Chromosome (binary vector) 들의 집단.

■ Evaluation fuction : Fitness (적응도)

eval (v) = f(x)

(ex)

v1 = (1000101110110101000111)

v2 = (0000001110000000010000)

v3 = (1110000000111111000101)

-> x1 = 0.637197, x2 = -0.958973, x3=1.627888

-> eval(v1) = f(x1) =1.586345,

eval(v2) = f(x2) =0.078878

eval(v3) = f(x3) =2.250650

■ Genetic Operators

- Reproduction : 복제
- Crossover : 교배
- Mutation : 돌연변이

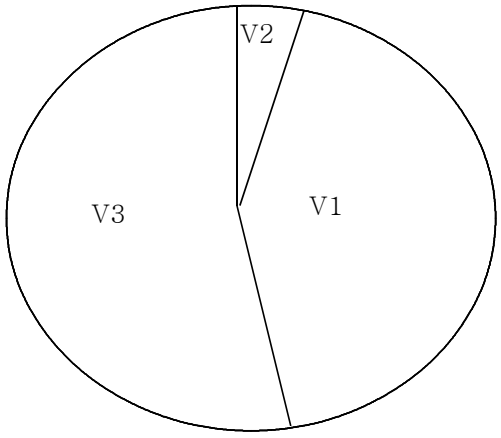
(1) Reproduction

: 어느 세대의 Population 으로부터 다음 세대의 Population 생성.

: 적응도가 우수한 Chromosome 복제 - 적자생존 원리

(ex)

No	Chromosome	Fitness	% of Total
1	V1	1.586345	40.5%
2	V2	0.078878	2.0%
3	V3	2.250650	57.5%
Total		3.915873	100%



* Roulette 게임을 population size 만큼 실시하여, 선택된 chromosome 으로 다음세대의 population 구성.

(예) 3 회 실시하여 v3, v1, v3 선택 경우

1 세대 (v1, v2, v3) -> 2 세대 (v3, v1, v3) : v2 도태

(2) Crossover

: 2 개의 Chromosome 을 혼합하여 새로운 Chromosome 생성.

(ex)

v2 = (00000|01110000000010000)

v3 = (11100|00000111111000101)

(5번 유전자 이후 교배)

v2' = (00000|00000111111000101) -> f(v2') = 0.940865

v3' = (11100|01110000000010000) -> f(v3') = 2.459245

(3) Mutation

: Chromosome 에서 한 개의 gene 을 반전

(ex)

v3 = (11100000000111111000101)

(10번째 유전자 돌연변이)

v3' = (11100000001111111000101)

-> x3' = 1.630818,

f(x3) = 2,342555 (increase fitness)

■ Parameters

- Population Size (pop_size) : binary vector 들의 수 (캥거루의 수)
- Probability of crossover (p_c) : Crossover 확률 (지정치)
- Probability of mutation (p_m) : Mutation 확률 (지정치)

◎ Genetic Algorithm (or Evolution Program)

```
begin
    t <- 0
    initialize P(t)
    evaluate P(t)
    while (not termination-condition) do
    begin
        t <- t+1
        select P(t) from P(t-1)              // Reproduction
        alter P(t)                           // Crossover, Mutation
        evaluate P(t)
     end
end
```

P(t) : t 번째 generation 의 Population

Generation Number	Evaluation Function
1	1.441942
6	2.250003
8	2.250283
9	2.250284
10	2.250363
12	2.328077
39	2.344251
40	2.345087
51	2.738930
99	2.849246
137	2.850217
145	2.850227