

~~QUESTION~~
Suppose whenever we want to solve any problem,
we write program.

Ques: The logic used in the program is called algorithm.

Two types of algorithm:

Deterministic

Output known to programmer.

Randomised

Output can be diff for
diff inputs.

Eg: If we have a list with some nos:

$$l \rightarrow \{2, 1, 3, 4, 6, 5\}$$

We want to sort it.

On applying any algorithm,

output $\rightarrow \{1, 2, 3, 4, 5, 6\}$ (Deterministic)

Time complexity = $O(n \log n)$ (not too high)

Eg: We want to find

max value of $f(x) = \sin x$ $0 \leq x \leq 3$

If we are not aware about the nature of the function how to write an algorithm for it?

break the interval into 100 (say) points

$$0 \leq x \leq 3 \quad (\text{equispaced})$$

$$\frac{3-0}{100} = 0.03$$

$$100$$

$$x_0, x_1, \dots$$

$$0 \approx 0.03$$

$$x_{99}$$

$$3$$

(to find max)

$$\text{No. of evaluations} = 100 + 100 + 100$$

(points x_i) (cal. of $f_i(x_i)$)

This will give a local value. As we increase the no. of intervals, we approach towards answer.

If we want approximate value, we will have to take 2^{100} points.

No. of evaluations reqd: 3×2^{100}

This is deterministic approach.

More Optimistic Approach: Using AI

e.g.: We want max CPI in a class of 21 students
So 21 evaluations needed.

But if we impose a condition that we will evaluate only those students whose CPI > 8 so no. of evaluations reduced.

We impose conditions such that we obtain those cases which respond that will decrease the computation.

The cases obtained will randomly lie in the search space. So these are called randomised algorithm.

* First randomised algo was written by John Holland in 1977.

* David Goldberg - student of John Holland
He created Binary GA.

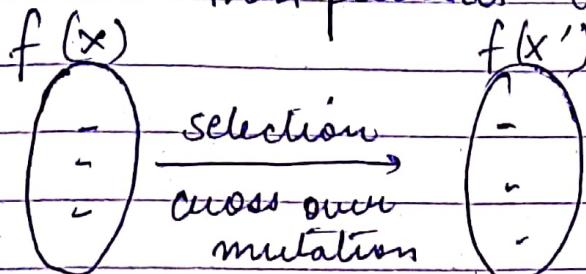
Genetic Algo based on Darwin Theory.

Natural Selection of Evolution

Darwin Theory :-

Generation is getting more intelligent.

If you have certain species at time t , there will be selection and crossover mutation to generate next generation species. The next gen. is better than previous one.



(t)

$$f(x) < f(x')$$

fitness of $x <$ fitness of x'

This theory is applied to binary GA.

$$\max f(x) = x^2 \quad 0 \leq x \leq 4$$

$$x = 0 \rightarrow 0$$

$\hookrightarrow x + h$

we have to write a fn to get this value as well.
(step size)

Binary GA

$$\max f(x) = \frac{x^2}{1+x^3} \quad 1 \leq x \leq 3$$

Let say we take 2^{10} points in the search space.

We know the binary coding of those 2^{10} solutions.

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

⋮
⋮

1 1 1 1 1 1 1 1 1

} Binary coding

Initially, we create initial population randomly.

00000100010
0010110011
1000100000
110000011

These random values we need to locate in the interval $1 \leq x \leq 84$.

$$X_{\text{actual}} = X_L + \frac{X_H - X_L}{2^{10} - 1} \times (\text{Decimal (bs) Value})$$

For: 000.0 0 1000.00

$$X_{\text{actual}} = 1 + \frac{4-1}{1023} \times (32)$$

e.g.: - $0 \leq x \leq 3$

4 part

$$00 \rightarrow 0 + \frac{3-0}{4-1} \times (0) = 0$$

$$01 \rightarrow = 1$$

$$10 \rightarrow = 2$$

$$11 \rightarrow = 3$$

Q) A search space is divided into 2^n parts.
Now if we increase the no. of points to 2^{n+1}
Then what will be the new values. Will the
old points lie in the new points?

4/9/19

binary GA \rightarrow Real parameter coding.

How can we use encoding?

There are 3 operators in Genetic Algo:-

1. Selection

2. Cross Over

3. Mutation

best sol -

exploitation

They are used for exploration.

We want to find

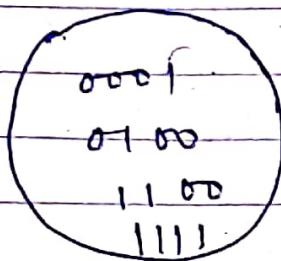
$$\max f(x) = \frac{x}{1+x^2} \quad 0 \leq x \leq 3$$

Step 1: Define size of the chosen search space

$$0 \leq x \leq 3$$

We'll define divide the search space into
 2^4 parts

How initial population will be created?
 In questions we take initial pop as 4
 but for simplicity in this eg: we'll take it as 4.



initial pop.
 (selected randomly)

Step 2: we'll calculate actual value of these solutions:

		fitness
0001	→	0.2
0100	→	0.8
1100	→	2.4
1111	→	3

$$\left(0 + 3 \times \frac{1}{15} \right)$$

We'll calculate the fitness of these sol's.

fitness of 0.2

$$f(0.2) = \frac{0.2}{1 + 0.04} = \frac{0.2}{1.04}$$

$$f(0.8) = \frac{0.8}{1 + 0.04} = \frac{0.8}{1.04}$$

We need to identify which region is representing good region.

Here, we need to maximise it. So, second solⁿ is the best solⁿ.

So, this "may" be the avg.

The size of initial pop. would never change.

All need to replace bad sol by good solⁿ.

Operator: binary tournament selection operator

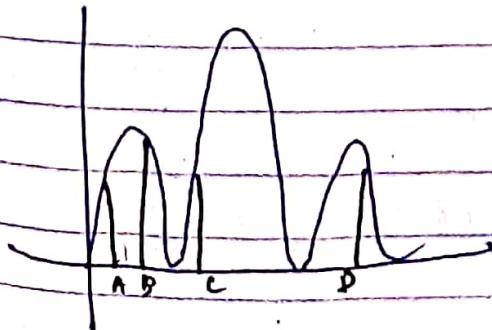
We'll choose 2 random solⁿ from the pool
say

0100 1111

We compare the fitness of them & solⁿs.
And best one will be selected.

Fitness of 0100 is more
4 times random values will be chosen.

0100
0100
1111



Why we are using randomizing?
We need to select good
solutions from the entire
space. If we would have
made avg of all as

threshold and then some regions (like) would be ignored as we would never be able to reach global max (in this case).

Its convergence rate is low because there is no guarantee of whether a point is useful or useless.

Explottation :-

If 0001 is a good solⁿ.
0010 may be the next good solⁿ.
We increment the solⁿ and check.

* We uses crossover and mutation both for exploitation and exploration.

$$\begin{array}{r} 0100 \\ \rightarrow 0101 \\ \rightarrow 0110 \end{array}$$

$$\begin{array}{r} 0100 \rightarrow 1011 \\ (\text{completely diff sol}^n) \end{array}$$

$$\begin{array}{c|ccccc} 0 & 1 & 0 & 0 & \rightarrow & 0 & 1 & 0 & 1 \\ \hline 1 & 1 & 1 & 1 & \rightarrow & 1 & 1 & 1 & 0 \end{array} \quad \left. \begin{array}{l} \\ \\ \end{array} \right\} \text{exploitation}$$

$$\begin{array}{c|ccccc} 0 & 1 & 0 & 0 & \rightarrow & 0 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & \rightarrow & 1 & 1 & 0 & 0 \end{array} \quad \left. \begin{array}{l} \\ \\ \end{array} \right\} \text{exploration}$$

Exploitation / exploration will depend on the cross side selected.

Prob of cross over. All parents will not produce new children.

$$\text{Prob} = 0.7$$

$$4 \times 0.7 = 2.8 \quad (\text{3 times})$$

01|00
11|11

\rightarrow 0111 } we want to include
 \rightarrow 1100 } these in the pool or
not. We will check
fitness:

Do cross over 3 times.

After cross over, you need to use mutation.

1100
1111
0100
1101

mutation



1011

1011

mutation

cond. after
cross over

$$\text{Prob. of mutation} = 0.6$$

$$4 \times 0.6 \approx 2.4 \quad (\text{so 3 times apply})$$

After selection / crossover / mutation, a new generation is made.

Crossover - exploration ? generally
mutation - exploration

The entire process G/C/M is repeated 24-25 times and various generations are obtained.

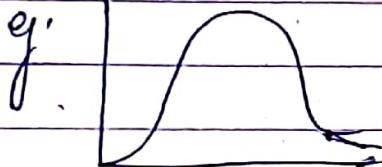
9/9/19

Problems:-

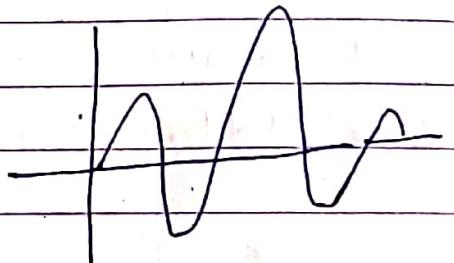
2 types of optimization problems:-

i. Unimodal

only one global optimal.



ii. multimodal

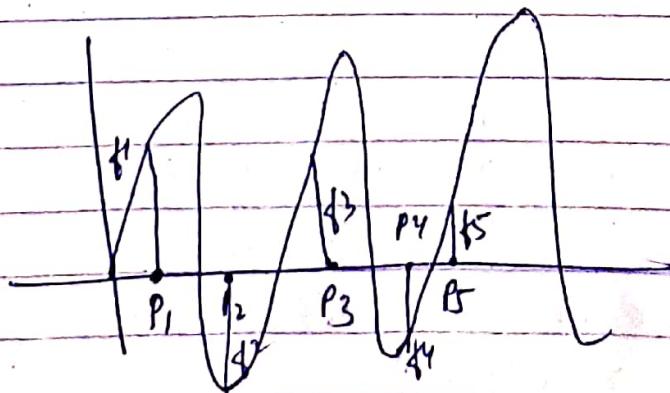


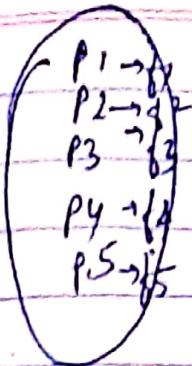
only exploitation reqd to reach global solⁿ.

both exploitation & exploration required.

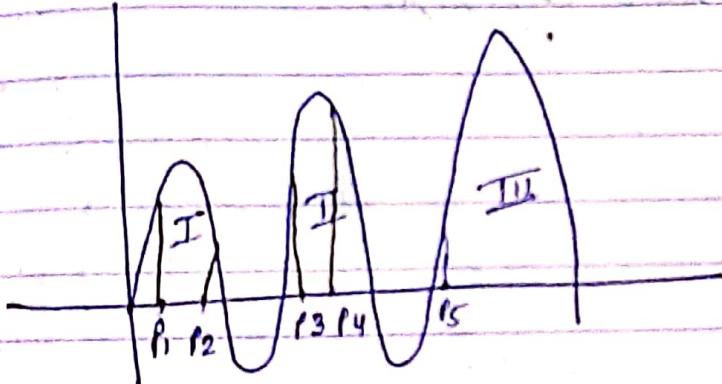
$$f(x) = \frac{x^2}{1+x^2} \quad 0 \leq x \leq 3$$

max





P_5, f_5 should be there in the sol^D even if f_5 is less than others.



If normal selection applied, P_5 will never be there. It is a multimodal problem, we want to cover entire region. So we want to choose one representative from every region.

Sharing fitness Concept

We will find avg fitness of P_1, P_2 for region I.

Avg fitness of P_3, P_4 for region II.

Euclidian dist. used to find if 2 pts. are close or distant.

$$\sigma = 0.5 \text{ (If dist} < \sigma \Rightarrow \text{close)} \\ (\text{say}) \quad \text{dist} > \sigma \Rightarrow \text{far}$$

(3,4)

P₁

(3,5)

P₂

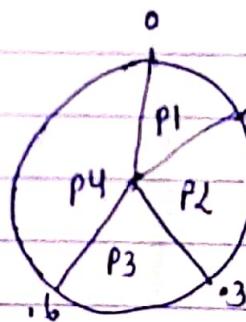
$$r = 1.5$$

$$d_{ij} = \sqrt{(0)^2 + (1)^2} = 1 < 1.5$$

∴ They share same region.

Doulette wheel selection operator

fitness	
P ₁ → 1	→ 0.1
P ₂ → 2	→ 0.2
P ₃ → 3	→ 0.3
P ₄ → 4	→ 0.4



Roulette wheel representing total sum of fitness value.

$$\text{max} f(x) = \frac{x^2}{T+x}$$

Normalise all fitnesses

$$\sum_{i=1}^4 f_i = 1+2+3+4 = 10$$

$$\frac{1}{10} = 0.1$$

$$\frac{2}{10} = 0.2$$

(Normalised fitness)

Generate a random value and check where it will lie.

$$r_1 = 0 \text{ to } 1 = .5 \rightarrow P_3$$

$$r_2 = .7 \rightarrow P_4$$

$$r_3 = .3 \rightarrow P_2$$

$$r_4 = .41 \rightarrow P_3$$

4 random values

Random no. of generated uniformly. So most values will lie in P_4 as P_4 covers max. part of the wheel.

Q: $f(x, y) = xy^2$ $0 \leq x \leq 1$ $1 \leq y \leq 5$

4 solution $\rightarrow 10$ bit

bit

Sol: Decide kitni ~~value~~ x k liye hoga, kitni y k liye hoga.

$$\begin{array}{c} x \\ 4 \\ 9 \\ 6 \end{array}$$

$$\begin{array}{c} 4 \\ 8 \\ 16 \\ 28 \\ -32 \end{array}$$

$$\begin{array}{r} 111100 \\ \times 111111 \\ \hline 10011111 \end{array}$$

4

fitness

$$11111111 \rightarrow (3, 6)$$

108

$$0101000000 \rightarrow (1, 1)$$

1

$$1010111100 \rightarrow (2, 5.76)$$

64.8

$$0000101010 \rightarrow (0, 3-0.4 \cdot 3)$$

0

$$0 \leq x \leq 3$$

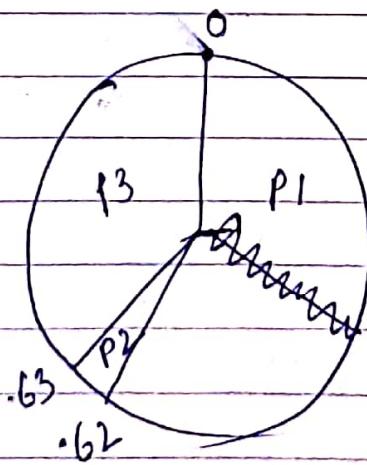
$$= 16$$

$$\begin{array}{ccc} 0, & 2 \\ \downarrow & \downarrow \\ 0000 & 0001 & 0010 \end{array}$$

Normalised fitness

$$\begin{array}{lll} P_1 \rightarrow f_1 = 108 & = \frac{108}{173.8} = .62 \\ P_2 \rightarrow f_2 = 1 & = \frac{1}{173.8} = .01 \\ P_3 \rightarrow f_3 = 64.8 & = \frac{64.8}{173.8} = .37 \\ P_4 \rightarrow f_4 = 0 & = 0 = 0 \end{array}$$

$$\sum f_i = \frac{1}{173.8} = 1 \text{ (sum)} \quad \checkmark$$



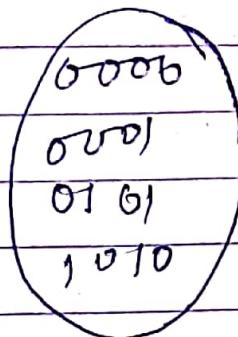
Now generate 4 random nos.

.2	-	P1
.8	-	P3
.1	-	P3
.6	-	P1

It is selecting good sol's only bcoz it selects only those which occupy good space in the wheel.

$$f(x) = \frac{x^2}{1+x^2} \quad 0 \leq x \leq 3$$

max



There are 4 bits. For more precision, we want to increase no. of bits. So can we use the same points in big domain?

0000

0

1

5

10

$$\left(\frac{x_H - x_L}{2^n - 1} \right) a = \left(\frac{x_H - x_L}{2^{2n} - 1} \right) x$$

$$x = a \cdot \frac{\left(\frac{x_H - x_L}{2^{2n} - 1} \right)}{\frac{a}{2^n - 1}} = (2^n + 1)a$$

$$\therefore \left(\frac{x}{a} \right) = 2^n + 1$$

~~11/9/19~~

Sharing Fitness Model

Sharing distance b/w 2 solutions

$$sh(d_{ij}) = \begin{cases} 1 - \left(\frac{d_{ij}}{\sigma_{\text{share}}}\right)^{\alpha} & \text{if } d_{ij} < \sigma_{\text{share}} \\ 0 & \text{if } d_{ij} \geq \sigma_{\text{share}} \end{cases}$$

Sharing dist. b/w
dist. i & j

we'll compare d_{ij} with σ_{share}
if $d_{ij} > \sigma_{\text{share}} \rightarrow$ (sharing effect)

most cases $\alpha = 1$

To calculate sharing fitness :

1	:
2	:
3	.
4	.

$$\sum_{j=1}^4 sh(d_{1j}) = n_1 \quad (\text{least count})$$

$$\sum_{j=1}^4 sh(d_{2j}) = n_2$$

$$\text{Sharing fitness of 1} = \frac{f(1)}{n_1}$$

$$\max f(x) = |\sin(\pi x)| \quad 0 \leq x \leq 2$$

binary G.A.

	String	decimal value	x_i	f_i	nG_i	f'_i
1	110100	52	1.651	0.890	2.856	0.312
2	101100	44	1.397	0.948	3.160	0.300
3	011101	29	0.921	0.246	1.048	0.235
4	001011	11	0.349	0.890	1.000	0.890
5	110000	18	1.524	0.997	3.364	0.296
6	101110	46	1.460	0.992	3.364	0.295

for nG_i

$$nG_i = sh(d_{i1}) + sh(d_{i2}) + sh(d_{i3}) + sh(d_{i4}) + sh(d_{i5}) + sh(d_{i6})$$

$$\begin{array}{l} \text{--- share} = 0.5 \\ q = 1 \end{array} \quad \text{say} \quad nG_i = 1 + 0.492 + 0 + 0 + 0.746 + 0.618 = 2.856$$

$$sh(d_{i1}) = 1 - \frac{d_{i1}}{\text{--- share}} = 1$$

$$sh(d_{i2}) = 1 - \frac{d_{i2}}{\text{--- share}} = 1 - \frac{1.397 - 1.651}{0.5} = 0.482$$

$$sh(d_{i3}) = 1 - \left(\frac{0.921 - 1.651}{0.5} \right) = 0$$

$$sh(f'_i) = \frac{0.890}{2.856}$$

sol 1 has effect of

1, 2, 5, 6

Genetic Algo → Evolutionary Algo

Particle Swarm Optimization :-
based on intelligence of birds/insects