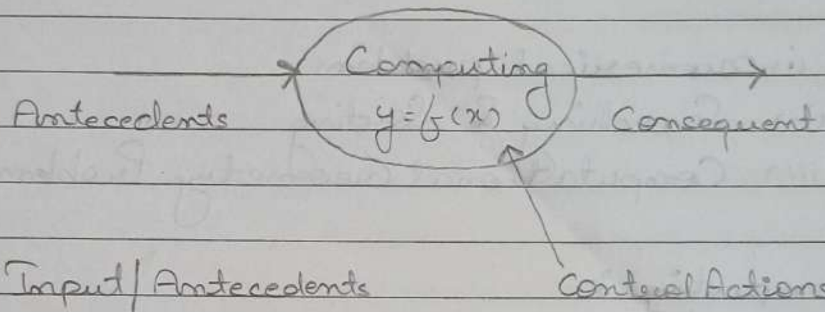
Soft Computing :-

→ what is Computing?
Computing is the process of using computer technology to complete a given goal-oriented task. Computing System contains ~~to~~ certain inputs to generate a Specific Output.



- x : Input / Antecedents
- y : Output / Consequent
- f : Mapping function / Control Action

→ Features :-
- Should ~~ac~~ accept user information in terms of feature sets and produce a desired output using a suitable ~~action~~ control action.
- Control action should be unambiguous and accurate.
- Control actions should be easy to conceptualise, Construct and operate.
- Should provide precise and optimized Solution.

→ Hard Computing :-
- Precise Result
- Control Actions
  - ↳ Unambiguous
  - ↳ Formally defined
- It is ~~primar~~ primal procedure for Solving engineering problems that can be formulated mathematically.

→ Features :-

- Input-output relationship are expressed in terms of mathematical expression.
- Control actions must be suitable to the problem defined.
- Control actions should be stable, highly predictable and accurate.

→ Solve – (i) numerical problem
  (ii) Searching & Sorting
  (iii) Computational Geometry Problem

→ Soft Computing :-

- It resembles the reasoning of human mind being a collection of biologically inspired methodologies.

→ Features :-

- Imprecision
- Dynamic
- Uncertainty
- Low solution cost
- Do not require mathematical model.

Doctor & Patient	Fuzzy logic
A, a, A, a → SC → A	Neural Network
CSK, KKR, RCB → SC → who wins 2019 IPL?	Evolutionary / Genetic Computing

→ Difference Soft Computing and Hard Computing

Hard Computing	Soft Computing
1. Precisely stated analytical model is required.	1. Imprecision is tolerable
2. More computational time is required.	2. Computational time required is less as it involves intelligent computational steps.
3. Precision is observed within the computation.	3. Approximation is observed in the computation.
4. Imprecision and uncertainity are undesirable properties	4. Tolerance for imprecision and uncertainty is exploited to achieve tractability.
5. Programs are written which follows standard rules.	5. Programs are evolved with new laws and theories.
6. Requires exact input data	6. Deals with ambiguous and noisy input data.
7. Strictly follows sequential computations	7. Allows parallel computation.
8. Produces precise result.	8. Produces approx. results

→ Hybrid Computing :-
Combination of conventional hard computing and emerging soft computing.

→ Fuzzy logic :-

- Logic represents and processes knowledge.
- ~~Crips~~ Crisp / binary / boolean and fuzzy logic are both useful for ~~ton~~ knowledge based systems.
- Fuzzy logic is a mathematical tool for dealing with uncertainity and hence a cognitive ~~p~~ process.
- Crisp logic deals with crisp sets and crisp / boolean algebra.

- Fuzzy logic deals with uncertainty using fuzzy sets and fuzzy algebra.

Fuzzy logic
↓
mathematical language

Relational logic + Boolean logic + Predicate logic

→ Features :-

- This Concept is flexible that we can easily understand and ~~im~~ implement.
- It is used for helping the minimization of logics created by the human.
- It is the best method for finding the solution of those problems which are suitable for approx or certain reasoning.
- In the fuzzy logic, any System which is logical can be easily fuzzified.
- It is based on natural language processing and allows users to integrate with the programming.

→ Difference Crisp logic and fuzzy logic

Crisp set	Fuzzy Set
1) It has sharp boundaries restricted by 0/F and 1/T.	1) It has fuzzy boundaries with a degree of membership
2) Some Crisp set can be fuzzy.	2) No fuzzy set can be Crisp.

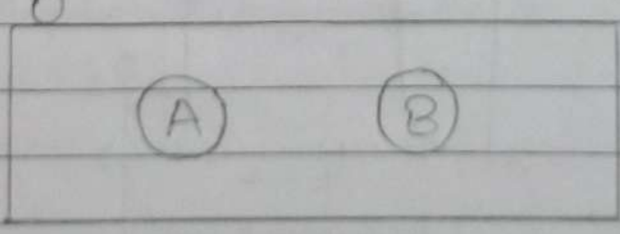
3) Defined by precise and certain characteristics	3) Prescribed by vague or ambiguous properties
4) Elements are single and either a member of a set or not.	4) Elements are ordered pairs and are allowed to be a partial member of set.
5) logic is bi-valued	5) logic is infinite val
6) Used in digital system design.	6) Used only in fuzzy controllers.
Crisp o/p ↳ Yes or No ↳ True or False	Fuzzy o/p ↳ May be ↳ May not be ↳ Absolutely ↳ Partially

→ Crisp set :-

For a Crisp set $A \subset U$ and any arbitrary element x

$$\text{either } x \in A \text{ or } x \notin A$$

U : All Student
A : In 10^{th}
B : In 12^{th}



∴ Disjoint set
1 element 1 hi set belong kar sak at a time.

→ Fuzzy Sets :-

Fuzzy sets are generalization of Crisp sets and it allows partial membership.

Fuzzy set A is defined as :

$$A = \{(x, \mu_A(x)) \mid x \in U\}$$

U = All students

G = Good Students

S = Bad Students

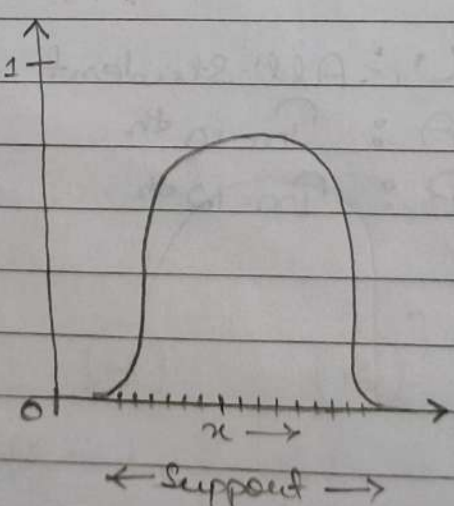
$G = \{G, \mu(G)\}$ $\mu()$ degree of goodness

$G = \{(A, 0.9), (B, 0.7), (C, 0.1), (D, 0.3)\}$

$S = \{(A, 0.1), (B, 0.3), (C, 0.9), (D, 0.7)\}$

→ Membership function :-

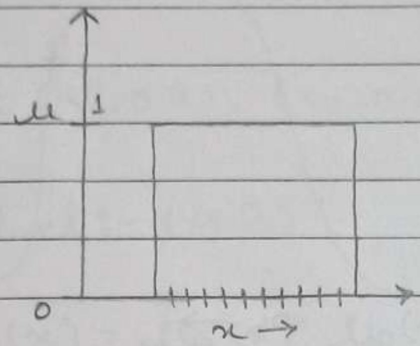
(I) Support : Set of all points such that $\{\mu_A(x) > 0\}$



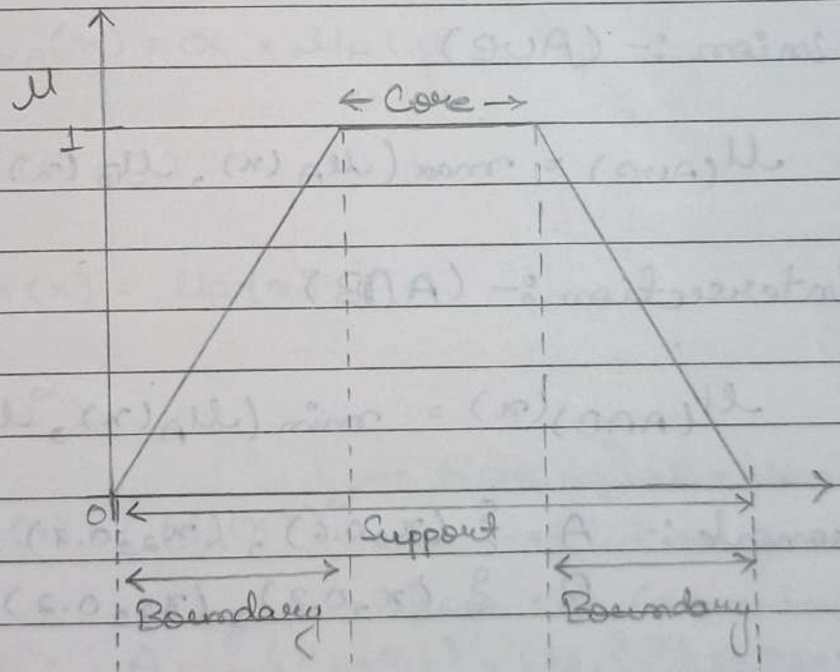
8 Support (A) = $\{x \mid \mu_A(x) > 0\}$

(II) Core : $\mu_A(x) = 1$

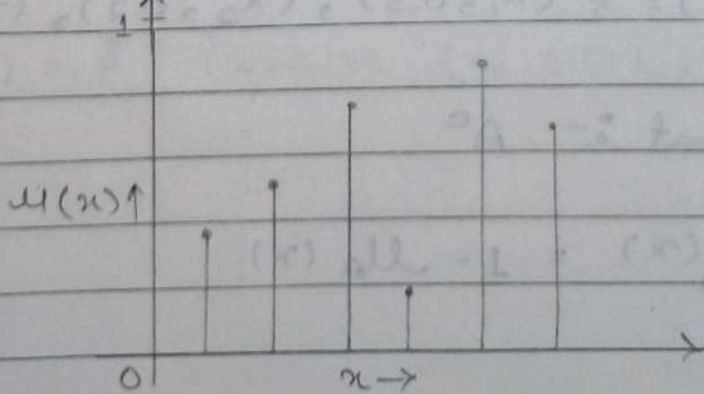
$$Core(A) = \{ x \mid \mu_A(x) = 1 \}$$



(III) Boundary : $(1 > \mu_A(x) > 0)$

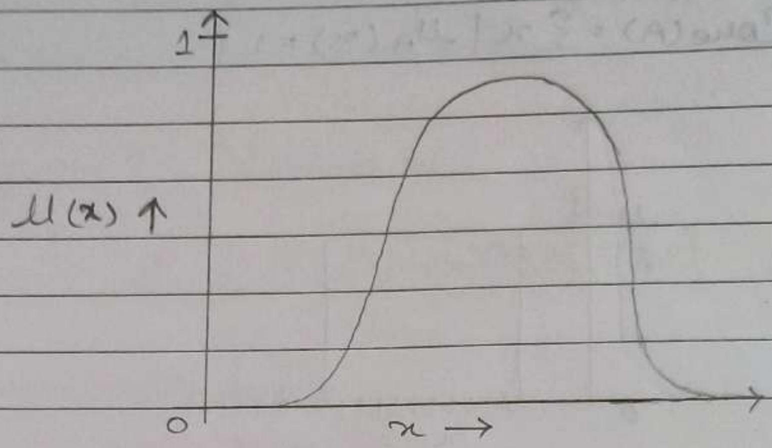


→ Discrete Membership function :-



Ye discrete MF tab ho
jab MF or element
MF and element m
se koi ek discrete
hoga.

→ Continuous Membership function :-



→ Fuzzy set operations :-

1) Union :- $(A \cup B)$

$$\mu_{(A \cup B)} = \max(\mu_A(x), \mu_B(x))$$

2) Intersection :- $(A \cap B)$

$$\mu_{(A \cap B)}(x) = \min(\mu_A(x), \mu_B(x))$$

Example :- $A = \{(x, 0.6), (x_2, 0.7), (x_3, 0.4)\}$
$B = \{(x, 0.3), (x_2, 0.2), (x_3, 0.5)\}$

$$(A \cup B) = \{(x_1, 0.6), (x_2, 0.7), (x_3, 0.5)\}$$

$$(A \cap B) = \{(x_1, 0.3), (x_2, 0.2), (x_3, 0.4)\}$$

3) Complement :- A^c

$$\mu_{(A^c)}(x) = 1 - \mu_A(x)$$

Date / /

Example :- $A = \{(x_1, 0.6), (x_2, 0.7), (x_3, 0.4)\}$

$B = \{(x_1, 0.3), (x_2, 0.2), (x_3, 0.5)\}$

$A^c = \{(x_1, 0.4), (x_2, 0.3), (x_3, 0.6)\}$

$B^c = \{(x_1, 0.7), (x_2, 0.8), (x_3, 0.5)\}$

4) Vector Product :- (A.B)

$$\mu_{AB}(x) = \mu_A(x) \cdot \mu_B(x)$$

5) Scalar Product :- ($\alpha \times A$)

$$\mu_{\alpha A}(x) = \alpha \times \mu_A(x)$$

6) Equality :- (A = B)

$$\mu_A(x) = \mu_B(x)$$

7) Power :- (A^α)

$$\mu_{A^\alpha}(x) = (\mu_A(x))^\alpha$$

→ Alpha ka value less than 1 hoga means dilution greater than 1 hoga toh Concentration.

Example :- $A = \{(x_1, 0.6), (x_2, 0.7), (x_3, 0.4)\}$

$B = \{(x_1, 0.3), (x_2, 0.2), (x_3, 0.5)\}$

$(A.B) = \{(x, 0.18), (x_2, 0.14), (x_3, 0.20)\}$

8) Sum (A+B) :-

$$\mu_{A+B}(x) = \mu_A(x) + \mu_B(x) - \mu_A(x) \cdot \mu_B(x)$$

9) Difference (A-B) :-

$$\mu_{A-B}(x) = \mu_{A \cap B^c}(x)$$

10) Disjunctive Sum ($A \oplus B$) :-

$$\mu_{(A \oplus B)}(x) = (A^c \cap B) \cup (A \cap B^c)$$

11) Cartesian Product ($A \times B$) :-

$$\mu_{A \times B}(x, y) = \min(\mu_A(x), \mu_B(y))$$

Example :-

$A(x) = \{(x_1, 0.2), (x_2, 0.3), (x_3, 0.5)\}$

$B(x) = \{(y_1, 0.8), (y_2, 0.6), (y_3, 0.3)\}$

$$A \times B = \begin{array}{c} \\ x_1 \\ x_2 \\ x_3 \end{array} \begin{array}{c} \begin{array}{ccc} y_1 & y_2 & y_3 \end{array} \\ \begin{bmatrix} 0.2 & 0.2 & 0.2 \\ 0.3 & 0.3 & 0.3 \\ 0.5 & 0.5 & 0.3 \end{bmatrix} \end{array}$$

→ Fuzzy set properties :-

1) Commutative :- $A \cap B = B \cap A$
$A \cup B = B \cup A$

Date / /

2) Associative :- $A \cup (B \cup C) = (A \cup B) \cup C$
$A \cap (B \cap C) = (A \cap B) \cap C$

3) Distributive :- $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$
$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$

4) Idempotence :- $A \cup A = A$; $A \cap A = \phi$
$A \cup \phi = A$; $A \cap \phi = \phi$

5) Transitive :- If $A \subseteq B$; $B \subset C$ then $A \subseteq C$

6) Demorgan's law :- $(A \cap B)^c = A^c \cup B^c$
$(A \cup B)^c = A^c \cap B^c$

→ Crisp Relation :-

$A = \{1, 2, 3\}$

$B = \{4, 7, 8\}$

$A \times B = \{(1,4), (1,7), (1,8), (2,4), (2,7), (2,8), (3,4), (3,7), (3,8)\}$

$R_1 = \{(a,b) \mid a > b, (a,b) \in A \times B\}$ If condition will not satisfy them $R = \{\phi\}$

$R_1 = \{(1,4), (1,7), (1,8), (2,4), (2,7), (2,8), (3,4), (3,7), (3,8)\}$

$$R = \begin{array}{c|c} & \begin{matrix} 4 & 7 & 8 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \end{matrix} & \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{array}$$

$R = \{(1,4), (2,7), (3,8)\}$

2) Associative :- $A \cup (B \cup C) = (A \cup B) \cup C$
$A \cap (B \cap C) = (A \cap B) \cap C$

3) Distributive :- $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$
$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$

4) Idempotence :- $A \cup A = A$; $A \cap A = \phi$
$A \cup \phi = A$; $A \cap \phi = \phi$

5) Transitive :- If $A \subseteq B$; $B \subset C$ then $A \subseteq C$

6) Demorgan's law :- $(A \cap B)^c = A^c \cup B^c$
$(A \cup B)^c = A^c \cap B^c$

→ Crisp Relation :-

$A = \{1, 2, 3\}$

$B = \{4, 7, 8\}$

$A \times B = \{(1,4), (1,7), (1,8), (2,4), (2,7), (2,8), (3,4), (3,$
$(3,8)$

$R_1 = \{(a,b) \mid a > b, (a,b) \in A \times B\}$ If condition w
not satisfy then
$R = \{\phi\}$

$R_1 = \{(1,4), (1,7), (1,8), (2,4), (2,7), (2,8), (3,4), (3,$
$(3,8)\}$

$$R = \begin{array}{c c} & \begin{array}{ccc} 4 & 7 & 8 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \end{array} & \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{array}$$

$R = \{(1,4), (2,7), (3,8$

→ Operation On Crisp Relation :-

→ Operation on fuzzy Relation :-

1) Union :($A \cup B$) we choose max bet. both the set

$A(x, y)$

$B(x, y)$

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \qquad B = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A \cup B = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

2) Intersection :($A \cap B$)

We choose min. bet. both the set.

$$A \cap B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

3) Complement : A^c Reverse of index.

$$A^c = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix}$$

∴ In Fuzzy relation we have to use $1 - \mu(x)$ formula for Complement

Camlin Page
Date / /

→ Fuzzy Relation :-

$$A = \{(x_1, 0.6), (x_2, 0.2), (x_3, 0.3)\}$$

$$B = \{(y_1, 0.7), (y_2, 0.3), (y_3, 0.4)\}$$

$$\mu_R(x,y) = \mu_{A\times B}(x,y) = \min\{\mu_A(x), \mu_B(y)\}$$

$$R = A \times B = \begin{array}{c c} & \begin{array}{ccc} y_1 & y_2 & y_3 \end{array} \\ \begin{array}{c} x_1 \\ x_2 \\ x_3 \end{array} & \begin{bmatrix} 0.6 & 0.3 & 0.4 \\ 0.2 & 0.2 & 0.2 \\ 0.3 & 0.3 & 0.3 \end{bmatrix} \end{array}$$

→ If then Rule in Fuzzy :-

- Fuzzy Implication
- Fuzzy Rule
- Fuzzy Conditional Statement

} Called as

If x is A then y is B

A and B is a set and x and y is a element that belong to set A and set B.

- x is 'A' : antecedents
  or
  Premise

- y is 'B' : Consequence
  or
  Conclusion

Fuzzy Rule : 'R' is denoted as

$$R : A \to B$$

Example : If Temp is high then pressure is low.

$T_{High} = \{(25, 0.1), (30, 0.2), (35, 0.5), (40, 0.6)\}$

$P_{Low} = \{(2, 0.3), (5, 0.5), (6, 0.4)\}$

If ~~tem~~ temp is high then pressure is low.

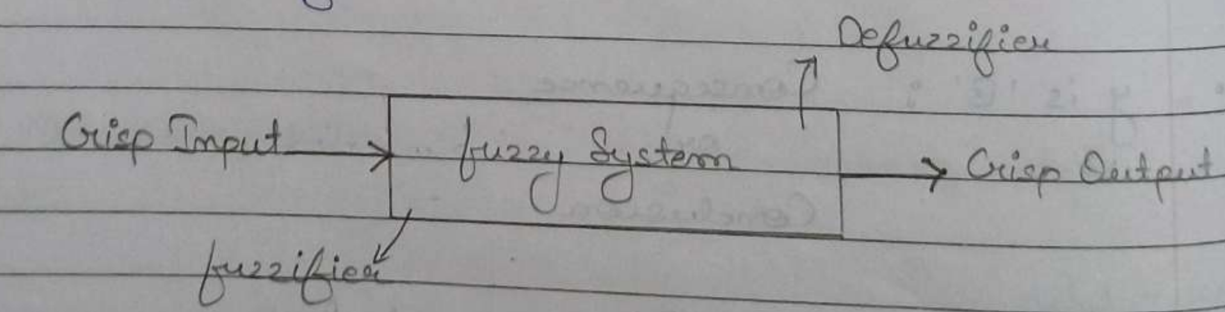
$R : T_{High} \rightarrow P_{Low}$

$R : A \rightarrow B$

$$R = A \times B = \begin{array}{c|ccc} & 2 & 5 & 6 \\ \hline 25 & 0.1 & 0.1 & 0.1 \\ 30 & 0.2 & 0.2 & 0.2 \\ 35 & 0.3 & 0.5 & 0.4 \\ 40 & 0.3 & 0.5 & 0.4 \end{array}$$

→ Defuzzification :-
Fuzzy to Crisp Conversion

Fuzzification :-
Crisp to fuzzy conversion



1) Lambda - cut method
2) Maxima methods
3) Weighted Sum method
4) Centroid methods.

Camlin Page
Date / /

1) Lambda cut-method :- fuzzy sets

- Fuzzy set $A \xrightarrow{To}$ Crisp set A_λ $(0 \leq \lambda \leq 1)$
- $A_\lambda = \{ x \mid \mu_A(x) \geq \lambda \}$

Example :-

$$A = \{ (x_1, 0.2), (x_2, 0.4), (x_3, 0.6) \}$$
$$\lambda = 0.3$$

$$A_{0.3} = \{ (x_1, 0), (x_2, 1), (x_3, 1) \}$$

$$= \{ x_2, x_3 \}$$

$$B = \{ (y_1, 0.5), (y_2, 0.4), (y_3, 0.7) \}$$
$$\lambda = 0.7$$

$$B_{0.7} = \{ (y_1, 0), (y_2, 0), (y_3, 1) \}$$

$$= \{ y_3 \}$$

→ Lambda cut-method :- fuzzy relation

$$R = \begin{bmatrix} 0.2 & 1 \\ 0.3 & 0.6 \end{bmatrix} \rightarrow \text{fuzzy relation} \qquad \lambda = 1, 0.5, 0.1, 0$$

$$R_\lambda = R_1 = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \rightarrow \text{Crisp relation}$$

∴ It is not necessary ki fuzzy relation ka crisp relation ek hi ho because it depend on the value of λ.

$R_{0.5} = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}$ $R_{0.1} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$

$R_0 = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$

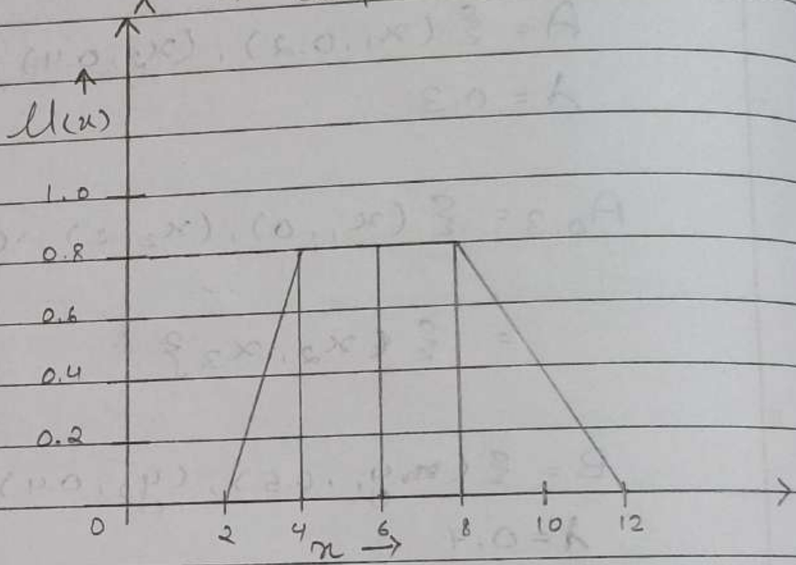
→ Maxima methods :- maxima means → height of fuzzy set

X^* → Crisp Value

1) First of Maxima (FOM)
$= x^* = 4$

2) Last of Maxima (LOM)
$= x^* = 8$

3) Mean of Maxima (MOM)



$$X^* = \frac{\sum x_i \in M \; x_i}{|M|}$$

$M = \{x \mid \mu_A(x) = \text{height of fuzzy set}\}$

$|M|$ = Cardinality of set M

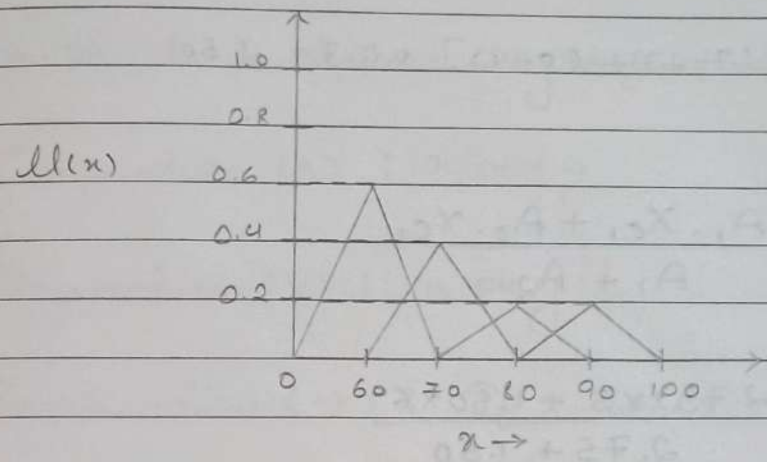
For mean of Maxima
$x = 4, 6, 8$

$$x^* = \frac{4+6+8}{3}$$

$$x^* = 6$$

→ Weighted Average Method :-

$$x^* = \frac{\sum \mu(x) \cdot x}{\sum \mu(x)}$$

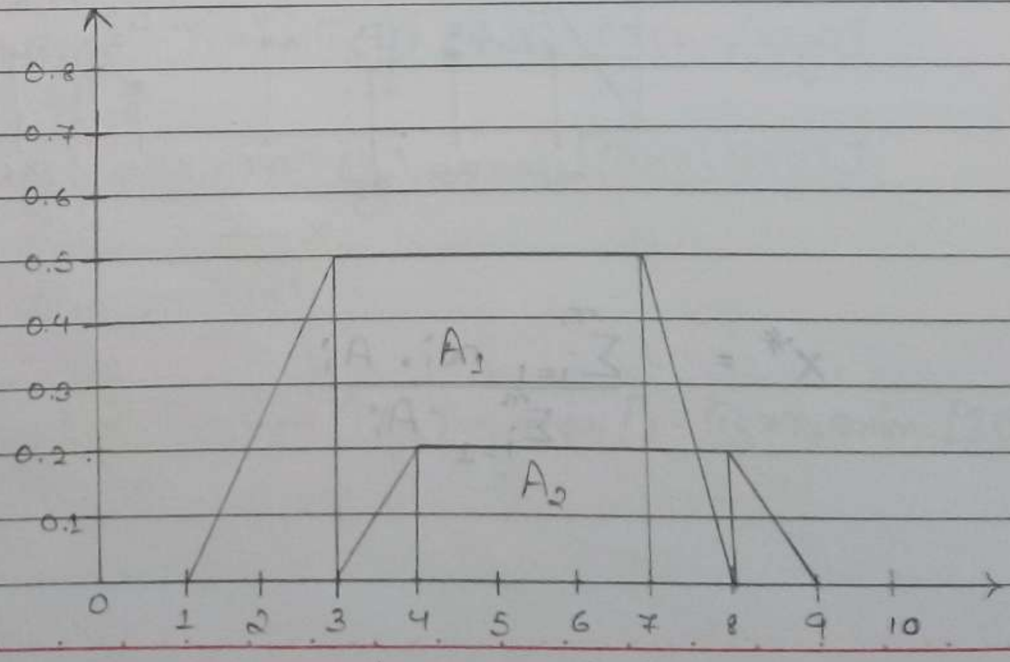


$$x^* = \frac{(60 \times 0.6 + 70 \times 0.4 + 80 \times 0.2 + 90 \times 0.2)}{0.6 + 0.4 + 0.2 + 0.2}$$

$$x^* = 70$$

→ Centroid Method :-

1) Centre of Sum (COS) :-



fuzzy sets C_1 & C_2

$$X^* = \frac{\Sigma A \, x_c}{\Sigma A}$$

$$A_1 = \frac{1}{2} [(8-1)+(7-3)] \times 0.5 = 2.75$$

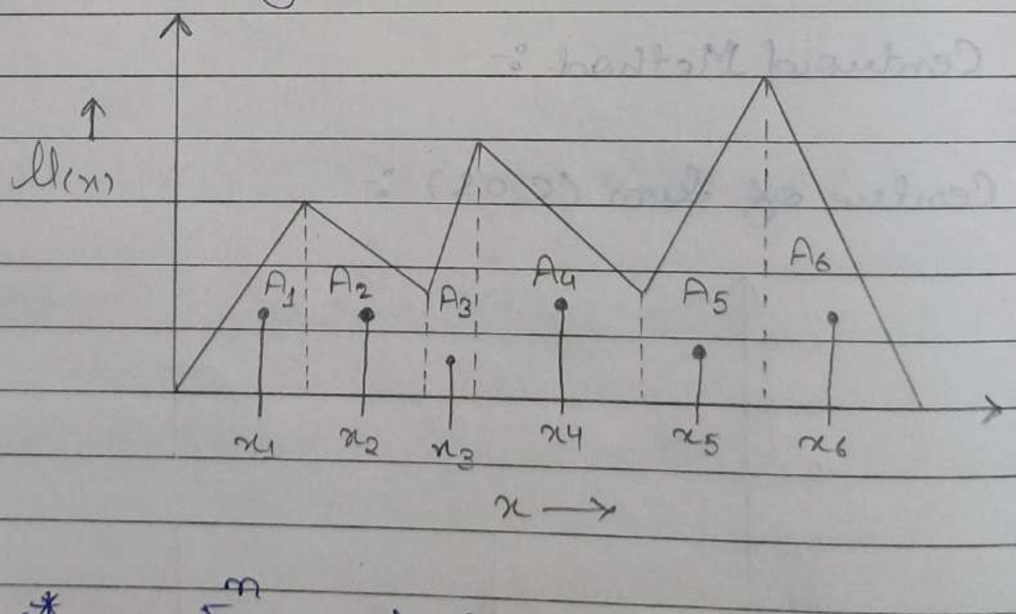
$$A_2 = \frac{1}{2} [(9-2)+(8-4)] \times 0.3 = 1.50$$

$$x^* = \frac{A_1 . X_{c_1} + A_2 . X_{c_2}}{A_1 + A_2}$$

$$= \frac{2.75 \times 5 + 1.50 \times 6}{2.75 + 1.50}$$

$$X^* = 5.35$$

2) Centre of Gravity (COG) :-



$$x^* = \frac{\sum_{i=1}^{n} x_i . A_i}{\sum_{i=1}^{n} A_i}$$

→ Truth Values and Tables in Fuzzy logic :-

- x is A

⇒ "London is in UK"
  ↓ Subject   ↓ predicate

- Truth table defines logic functions of 2 propositions

i) Conjuction ($\wedge$) : x and y

ii) Disjunction ($\vee$) : x or y

iii) Implication ($\rightarrow$) : If X Then Y

iv) Bidirectional ($\leftrightarrow$) : X IFF Y

- Truth value of propositions in fuzzy logic are in logic are in range [0, 1]

eg : P : Ram is Boy

$T(P) = 0.8$

- $T(x \text{ and } y) = T(x) \wedge T(y) = \min [T(x), T(y)]$

- $T(x \text{ or } y) = T(x) \vee T(y) = \max [T(x), T(y)]$

- $T(\text{Not } x) = 1 - T(x)$

- $T(x \rightarrow y) = T(x) \rightarrow T(y) = \max [1 - T(x), \min [T(x), T(y)]]$

→ Fuzzy Proposition & Crisp Proposition :-

P : Ram is a Boy

$T(P) = 0.0$
$T(P) = 0.2$
$T(P) = 0.8$
$T(P) = 1.0$

Q : Ram is intelligent

$T(Q) = 0.6$

=> Ram is not intelligent

$T(\bar{Q}) = 1 - T(Q) = 0.4$

=> Ram is Boy and so is intelligent

$T(P \wedge Q) = \min(T(P), T(Q))$

$= \min(0.8, 0.6)$

$= 0.6$

- Fuzzy predicates : " Shridhar is tall "
  (tall, short, quick)

- Fuzzy predicate modifiers :
  ("very, fairly, moderately, rather, slightly)

- Fuzzy quantifiers :
  (Most, Several, Many)

- Fuzzy quantifiers :
- Based on Truth (x is t)
- Based on Probability (x is λ)
- Based on Possibility (x is π)

→ Decomposition of Rules :-

1) Multiple Conjunctive antecedents :

If x is $A_1, A_2, A_3 \ldots A_m$ Then Y is B_m.

$$A_m = A_1 \cap A_2 \cap A_3 \cap \ldots \cap A_m$$

$$\mu A_m(x) = \min[\mu_{A_1}(x), \mu_{A_2}(x) \ldots \mu_{A_m}(x)]$$

⇒ If A_m Then B_m

2) Multiple disjunctive antecedents :-

3) Conditional Statements :-

If A_1 then B_1 Else B_2
⇒ If A_1 Then B_1
OR
If NOT A_1 Then B_2

4) Nested If-then rules
If A_1 Then [If A_2 then B_1]
⇒ If A_1 AND A_2 Then B_1

→ Aggregation of fuzzy rules :-

1) Conjunctive System of rules :-
- Rules to be jointly Satisfied
- Using AND
- Using intersection

$$y = y_1 \text{ AND } y_2 \text{ AND } y_3 \ldots \text{ AND } y_n$$

$$y = y_1 \cap y_2 \cap y_3 \ldots \cap y_n.$$

$$\mu_y(Y) = \min[\mu_{y_1}(Y), \mu_{y_2}(Y), \mu_{y_3}(Y) \ldots \mu_{y_n}(y)]$$

2) ~~So~~ Disjunctive System of rules :-
- The satisfaction of at least on rule.
- OR is used

$$y = y_1 \text{ or } y_2 \text{ or} \ldots \text{ or } y_n$$

$$y = y_1 \cup y_2 \cup y_3 \ldots \cup y_n$$

$$\mu_y(Y) = \max[\mu_{y_1}(y), \mu_{y_2}(y), \mu_{y_3}(y) \ldots \mu_{y_n}(y)]$$

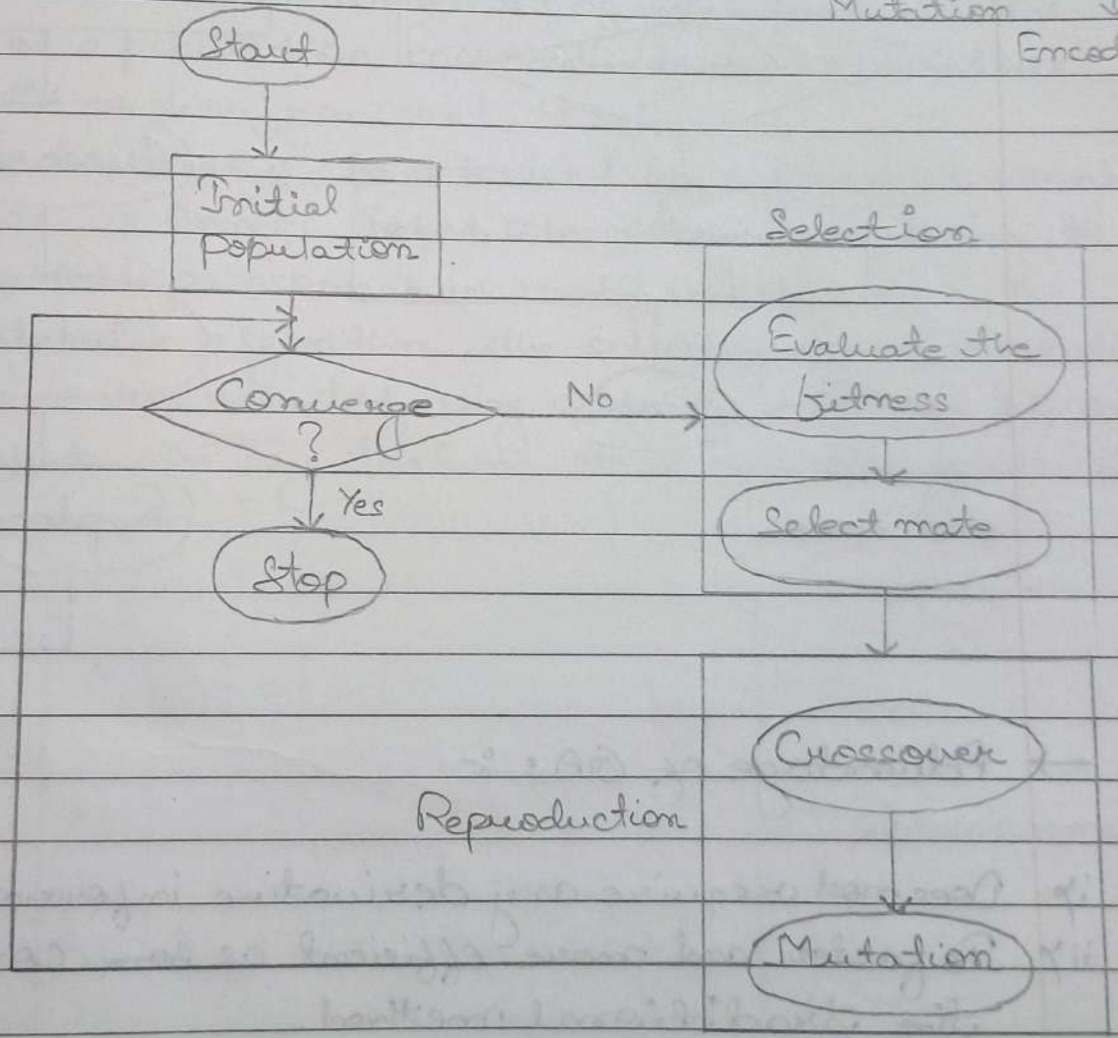
→ Genetic Algorithm :-
are adaptive heuristic search algo. that belongs to the larger of evolutionary algorithm. Genetic algo. are based on the ideas of ~~natural~~ natural selection and genetics. These are intelligent exploitation of random search provided ~~to~~ with historical data to direct the search into the region of better performance in solution ~~spa~~ Space. They are Commonly used to generate high quality Solution for optimization problems

and ~~prob~~ Search problems.

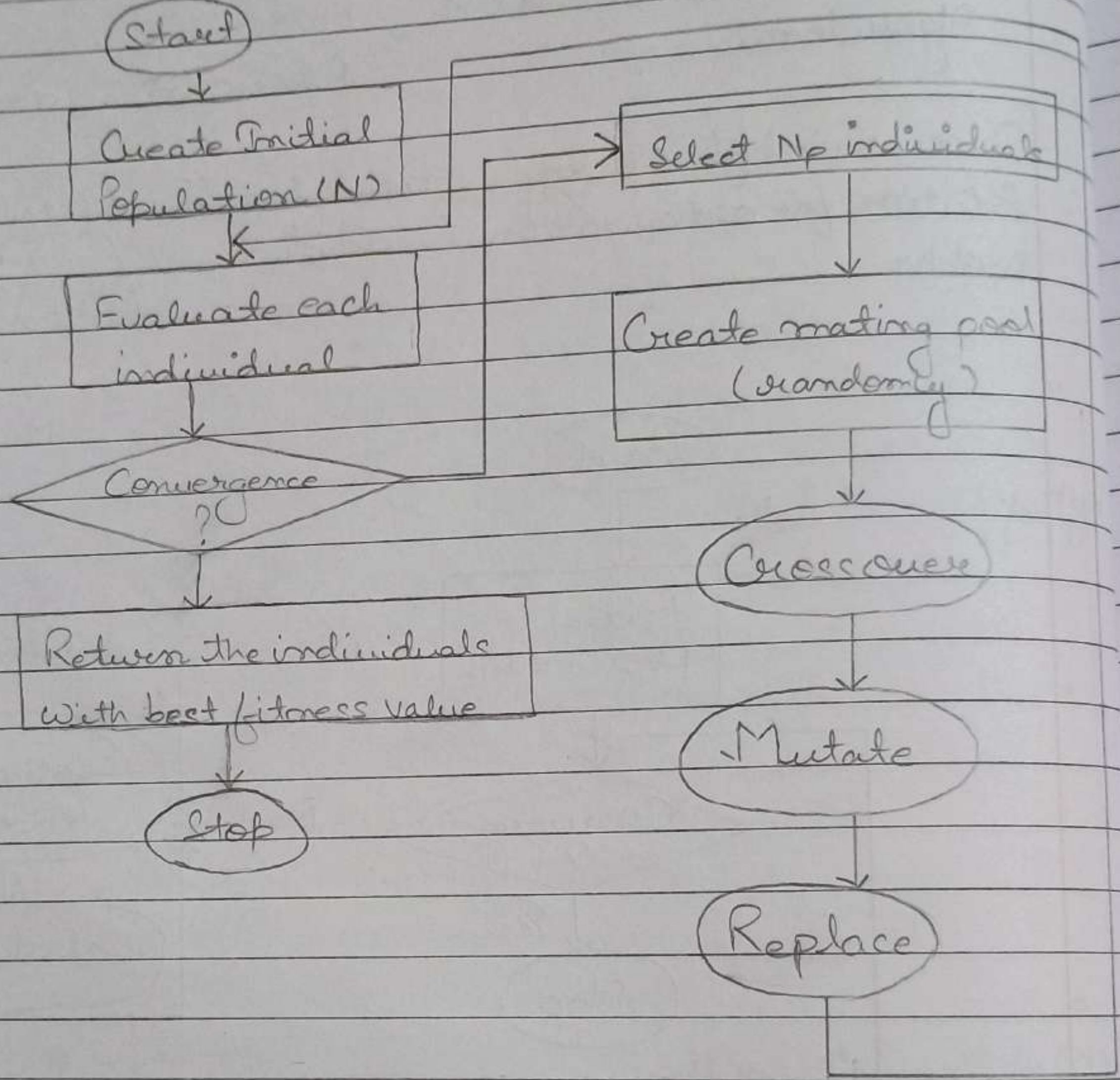
Adaptive heuristic Search, Evolutionary, Genetic &
Algorithm. Algorithm natural Selection.

To generate high-Quality Solution for optimization problem., Population & Individual, Operator of GA

Selection
Crossover
Mutation
Encoding



→ Simple Genetic Algorithm :-



→ Advantage of GAs :-

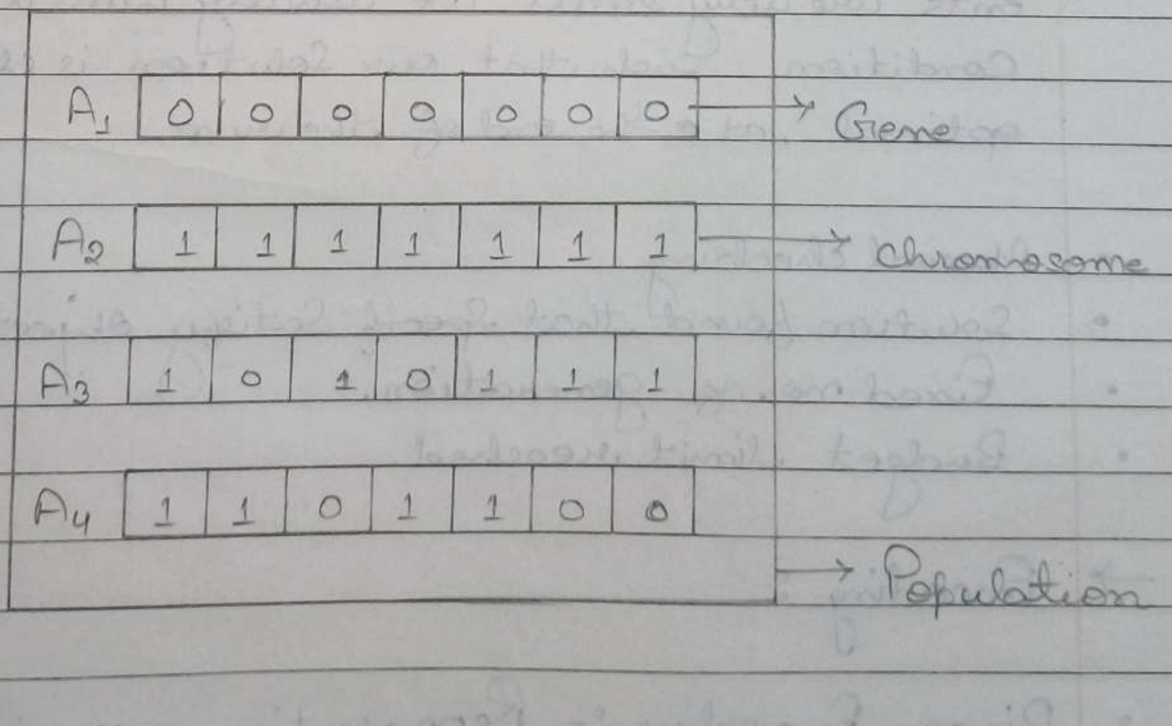
i) Does not require any derivative information
ii) Is faster and more efficient as ~~com~~ compared to the traditional method.
iii) Has very good parallel capabilities.
iv) Provides a list of "good" solutions and not just a single solution.
v) Always gets an answer to the problem, which gets better over the time.

→ GA phases :-

i) Initial population
ii) Fitness function
iii) Selection
iv) Crossover
v) Mutation.

i) Initial Population :-

- The process begins with a set of individuals which is called a population. Each individual is a Solution to the problem you want to solve.
- An individual is characterized by a set of parameter known as Genes. Genes are joined into string to form a chromosome.
- In a genetic algorithm, the set of genes of an individual is represented using a string, in terms of an alphabet. we say that we can encode the genes in chromosome.



→ Genetic Algorithm	Traditional Algorithm
i) Genetics & Natural Selection to solve optimization problem.	i) Step by step procedure to solve a given problem
ii) More advanced	ii) Not as advanced
iii) Used in field such as ML, AI.	iii) Used in fields such as programming, mathematics
iv) Probabilistic Rules	iv) Fully deterministic rules
v) Search on a population of points	v) Search on a single point

→ **Convergence Test / Termination Condition :-**

The termination condition of a genetic algorithm is important in determining when a GA run will end. It has been observed that initially, the GA progresses very fast with better solutions coming in every few iterations, but this tends to saturate in the later stages where the improvements are very small. We usually want a termination condition such that our solution is close to optimal, at the end of the run.

- Manual checking
- Solution found that ~~Specif~~ Satisfy objective criteria
- Fixed no. of generation.
- Budget limit reached.

→ **Encoding :-**

- Binary Encoding :- Representing a gene in terms of bits. In this encoding scheme, a gene or chromo-

Date / /

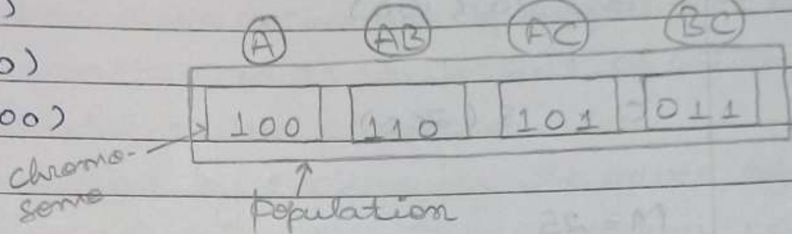
Gene is represented by a String of binary bits.

A $\circ$ (A) = 5 (100)

(B) = 10 (150)

(C) = 15 (200)

M = 25



Example : $f(x) = x^3$, $0 \leq x \leq 255$

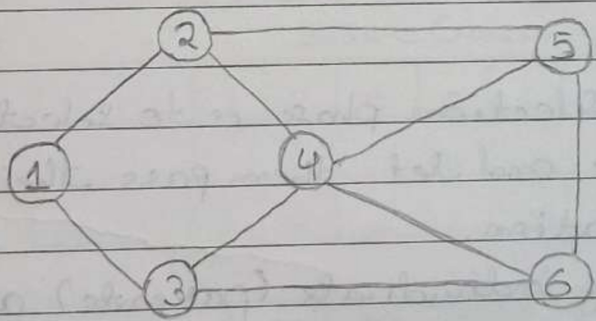
$\rightarrow$ 8 bit digit

10011011
155

11111111
255

ii) Fitness function :-

- Fitness function determines how fit an individual is
- It gives a fitness score to each individual
- The probability that an individual will be selected for reproduction is based on its fitness score



Path 1 : 1 2 5 6 4 3 1 - 18

Path 2 : 1 2 5 4 6 3 1 - 20

Path 3 : 1 2 4 5 6 3 1 - 12

Path 4 : 1 2 5 6 3 4 1 - 17

A = 5 (100)

B = 10 (150)

C = 15 (200)

M = 25

		w	P	
A	- 100	5	100	
AB	- 110	15	250	
AC	- 101	20	300	
BC	- 011	25	350	✓
ABC	- 111	30	450	x.

iii) Selection :-

- The idea of selection phase is to select the fittest individuals and let them pass their genes to the next generation.
- Two pairs of individuals (parents) are selected based on their fitness scores.
- Individuals with high fitness have more ~~chance~~ chance to be selected for reproduction.

iv) Crossover :-

- Crossover is the most significant phase in a genetic algorithm. For each pair of parents to be mated, a crossover point is chosen at random from within the genes.

- Once a pool of ~~making~~ mating pair are selected, they undergo through crossover operations.
- In crossover, there is an exchange of properties between ~~the~~ two parents and as a result of which two offspring solutions are produced.
- Offspring are created by exchanging the genes of parents among themselves until the crossover point is reached.

a) Single point crossover :-
- Here, we select the k point lying between 1 and L. let it be K.
- A single crossover point at k on both parent's string is selected.
- All data beyond that point in either string is swapped between the two parents.
- The resulting strings are the chromosomes of the offsprings produced.

Before Crossover

P_1 :	0	1	1	0	0	0	1	0
P_2 :	1	0	1	0	1	1	0	0

Two diploid from a mating pair

Crossover Point - K

O_1 :	0	1	1	0	1	1	0	0
O_2 :	1	0	1	0	0	0	1	0

After Crossover

b) Two point Crossover :-

- In this Scheme, we select two different crossover points k_1 and k_2 lying between 1 and L at random such that $k_1 \neq k_2$.
- The middle parts are swapped between the two strings.
- Alternatively, left and right parts also can be swapped.

Before Crossover

P_1 :	0	1	1	0	0	0	1	0
P_2 :	1	0	1	0	1	1	0	0

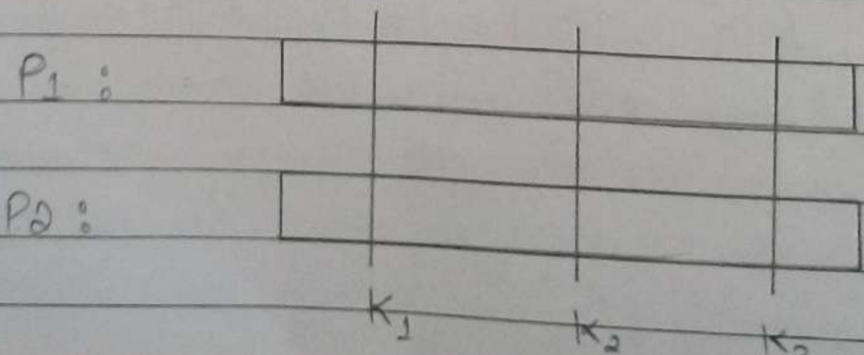
k_1 (after the 2nd bit), k_2 (after the 6th bit)

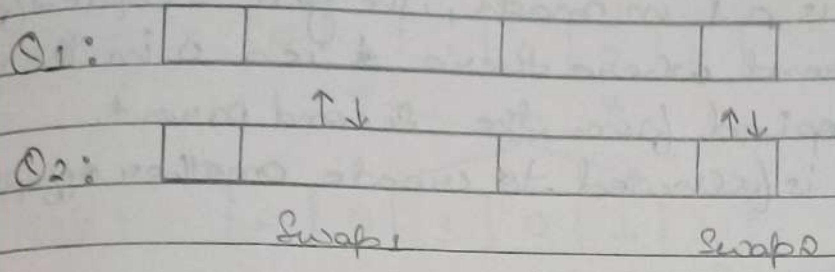
O_1 :	0	1	1	0	1	1	1	0
O_2 :	1	0	1	0	0	0	0	0

After Crossover

c) Multipoint Crossover :-

- In Case of multipoint Crossover, a number of crossover points are selected along the length of the string at random.
- The bits lying between alternate pairs of sites are then swapped.





d) Uniform Crossover :-

- Uniform crossover is a more general vision of multi-point crossover.
- In this scheme, at each bit ~~po~~ position of parent string, we toss a coin to determine whether there will be swap of the bits or not.
- The two bits are then swapped or remain unaltered, accordingly.

P_1:	1	1	0	0	0	1	0	1	1	0	0	1
P_2:	0	1	1	0	0	1	1	1	0	1	0	1

Coin tossing

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

O_1:	1	1	1	0	0	1	1	1	1	1	0	1
O_2:	0	1	0	0	0	1	0	1	0	0	0	1

e) Uniform Crossover with crossover mask :-

- Here, each gene is created in the offspring by copying the corresponding gene from one or the other parent chosen according to a random generated binary crossover mask of the same length as the chromosome.

- where there is a 1 in mask, the gene is copied from the first parent where there is a 0 in the mask the gene is copied from the second parent.
- The reverse is followed to create another offspring

Before Crossover

P1:

1	1	0	0	0

P2:

0	1	1	0	0

when there is a 1 in the mask the gene is copied from P1 else from P2.

Mask:

1	0	0	1	1

when there is a 1 in the mask the gene is copied from P2 else from P1.

After Crossover

O1:

1	1	1	0	0

O2:

0	1	0	0	0

→ Mutation :-

- In certain new offspring formed, some of their genes can be subjected to a mutation with a low random probability. This implies that some of the bits in the bit string can be flipped.
- Mutation occurs to maintain diversity within the population and prevent ~~pre~~ premature Convergence

Offspring:

0	1	1	0	0	1

MP(Mp):

0	1	0	0	0	1

MO:

0	0	1	0	0	0

offspring :

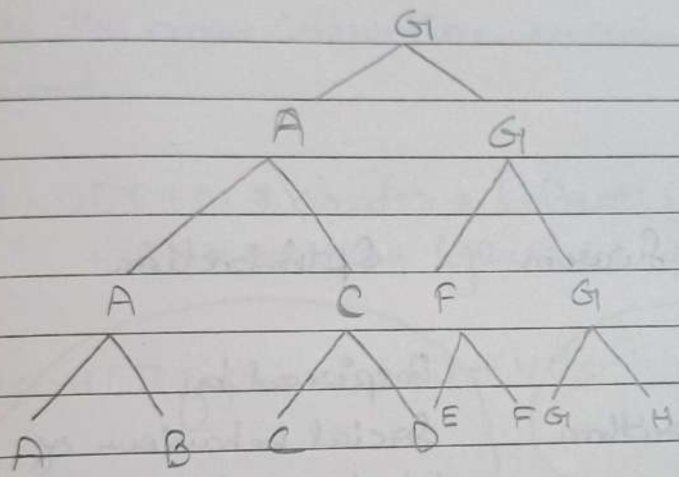
0	1	1	0	0	1

* *

MO :

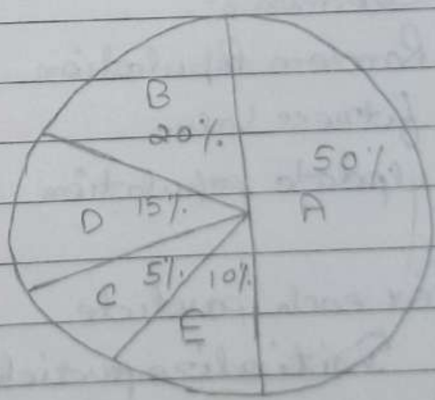
0	0	1	0	1	1

→ Tournament Selection :-



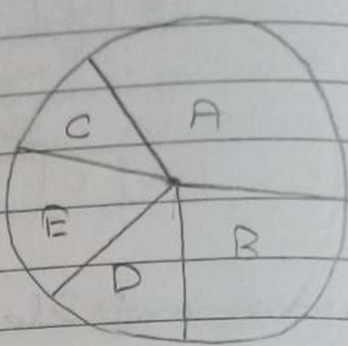
→ Roulette wheel Selection :-

C	F	
A	5	50%
B	2	20%
C	0.5	5%
D	1.5	15%
E	1	10%



→ Rank Based Selection :-

C	F	Rank	
A	5	5	= 33.33%
B	2	4	= 26.67%
C	0.5	1	= 6.67%
D	1.5	3	= 20%
E	1	2	= 13.33%
		15	



→ PSO :-

Particle Swarm Optimization

Population based Stochastic Algorithm

inspired by Social behaviour of ~~bir~~ bird flocking or fish schooling.

Common :-

Random Population

fitness Value

Update population

For each particle
  Initialize particle
END
DO
  For each particle
    Calculate fitness Value
    If the fitness Value is better than pBest in history
      Set Current Value as the new pBest.
END

Choose the particle with the best fitness value of all particles as the gBest.

For each particle
  Calculate particle velocity (1)
  Update particle Position (2)

END

While the max iteration or min error criteria is not attained.

1. v[] = v[] + c1 * rand() * (pBest[] - present[])
      + c2 * rand() * (gBest[] - present[])

2. present[] = present[] + v[]