Date: __/__/____

19/03/2025 **Genetic-Algorithm :-**

↳ 1965, Prof. J. Holland.

↳ Evolutionary-Algorithm.

→ Discrete-Optimization-Problems.

→ Evolution by mimicking natural living organisms - Evolution.

→ Guided Random-search Techniques.

→ Implicitly perform parallel-search simultaneously

→ Less-Probable to trap into the local-optima.

Main-Task of GA :-

① Initialization of Population.

* Let N = 10 (Population)

Chromosome (vector with multiple-elements).

'10' - solutions or - chromosomes.

| 0 | 1 | 0 | 0 | 0 | 1 | 0 | | | | | | | |

∴ We've 'N'-no. of solutions or chromosomes

- NOTE :- ① Content of chromosomes = '0' or '1'.

② All elements are diff. attributes/features, which are marked '0' or '1'.

Eg :- Travelling-salesman-Problem :-

we'll take only 4-cities

(8-cities) will be time consuming

* Reward-Table from travelling b/w 2-cities.

8-cities

		a	b	c	d	e	f	g	h
000	a	0							
001	b	10	0						
010	c	20	10	0					
011	d	5	20	5	0				
100	e	10	5	20	15	0			
101	f	20	5	30	40	20	0		
110	g	10	5	40	15	20	10	0	
111	h	10	30	20	25	15	5	10	0

* Length of each-chromosome :-

| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | | | | | |

→ 8 × 3 = 24.

#learnthesmarterway

Date: __/__/____

① let no. of Population => N = 4.

	a		b		c		d	
chromosome-1	0	0	0	1	1	0	1	1

	b		d		a		c	
chromosome-2	0	1	1	1	0	0	1	0

	c		d		a		b	
chromosome-3	1	0	1	1	0	0	0	1

	a		b		d		c	
chromosome-4	0	0	0	1	1	1	1	0

→ Order of cities taken is random here.

② calculate Objective-function :-

∵ start = End in TSP.

f(ch1) = Reward-point = $(a \to b) + (b \to c) + (c \to d) + (d \to a)$
$= 10 + 10 + 5 + 5 = 30$.

f(ch2) = $(b \to d) + (d \to a) + (a \to c) + (c \to b)$
$= 20 + 5 + 20 + 10 = 55$.

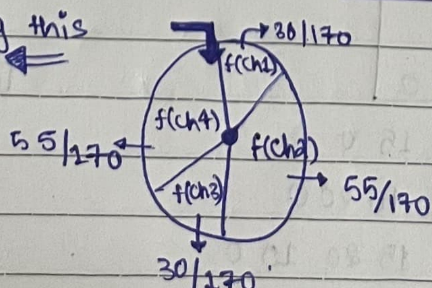
f(ch3) = $(c \to d) + (d \to a) + (a \to b) + (b \to c)$
$= 5 + 5 + 10 + 10 = 30$

f(ch4) = $(a \to b) + (b \to d) + (d \to c) + (c \to a)$
$= 10 + 20 + 5 + 20 = 55$.

③ selection for mating-pool :-

Roulette-wheel
- → Proportional-based-selection.
- → Process based on Darwin's-Principles.

⊙ let's say, after rotating this wheel, we get (randomly)



(Roulette-Wheel).

[ch2, ch4] [ch3, ch4] Mating Pairs.

#learnthesmarterway

21/03/2025. (Previous Problem continued)...

③ let's say $\begin{bmatrix} ch2 \\ ch4 \end{bmatrix}$ are selected.

$\begin{bmatrix} ch3 \\ ch4 \end{bmatrix}$

④ **crossover-operation :-** → chosen-randomly.

(crossovers-chosen)

Bisexual one-Point crossover →

ch2	0	1	1	1	0	0	1	0
ch4	0	0	0	1	1	1	1	0

⇒ Offspring-1 = | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |

Offspring-2 = | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |

ch3	1	0	1	1	0	0	0	1
ch4	0	0	0	1	1	1	1	0

⇒ Offspring-3 = | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 |

Offspring-4 = | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |

* **NOTE** :- How do you choose crossover-point?
  - ↳ choose random-no. between 0 & 1. (for e.g. '0.5')
  - ↳ Multiply it with size of chromosome (i.e., '8')

$0.5 \times 8 = ④ \rightarrow$ it's the crossover.

⑤ **Mutation-Rate = 0.05.**

* For each attribute of chromosome, we'll generate a random-no. b/w 0 & 1:
  - if random-no. > 0.05 (MR) - No change.
  - if random-no. < MR = Mutation.

mutation

Offspr. 1 =	0	1	1	1	1	1	0	0
Offspr. 2 =	0	0	0	1	0	0	1	0
Offspr. 3 =	1	0	0	1	1	1	1	0
Offspr. 4 =	0	0	0	1	0	0	0	1

Mutation

NOTE :- $0.05 \times (4 \times 8) = 1.6 \approx ② \rightarrow \therefore$ Max. '2' mutations can occur.

$\therefore f(\text{Offspr.1}) = f(b, d, \cancel{a}^{c}, a)$ ***(Imp)

↳ No-repetitions-allowed.

⇓

* replace with the left-out ones.

$f(\text{Offspr.2}) = f(a, b, \cancel{a}^{d}, c)$

$f(\text{Offspr.3}) = f(c, b, d, \cancel{c}^{a})$

$f(\text{Offspr.4}) = f(a, b, \cancel{a}^{c}, \cancel{b}^{d})$

Date: __/__/____

(6) Replace all in the actual chromosomes & calculate Objective-functions of new-offsprings :-

Offspr. 1 =	0	1	1	1	1	0	0	0
Offspr. 2 =	0	0	0	1	1	1	1	0
Offspr. 3 =	1	0	0	1	1	1	0	0
Offspr. 4 =	0	0	0	1	1	0	1	1

$\therefore f(\text{Offspr.}1) = f(b,d,c,a) = 20 + 5 + 20 + 10 = 55.$

$f(\text{Offspr.}2) = f(a,b,d,c) = 10 + 20 + 5 + 20 = 55.$

$f(\text{Offspr.}3) = f(c,b,d,a) = 10 + 20 + 5 + 20 = 55.$

$f(\text{Offspr.}4) = f(a,b,c,d) = 10 + 10 + 5 + 5 = 30.$

(7) (Optional) select chromosomes for next-iteration.

* Elitist-Solution ⇒

⇓

Replace "worst-fit" -solution with best-soln. from prev.-iteration.

Offspr. 1 = b d c a

Offspr. 2 = a b d c

Offspr. 3 = c b d a

Ch-4 = a b d c

↓

chosen from Roulette-wheel-score from prev. iteration.

NOTE → These two are same, so choose the second best solution from prev. iteration

⇓

∴ replace ch4 by ch-2.

(8) Stopping-criteria :-

yes ⇒ stop ⇒ return best-solution

No ⇒ Return to step-03.

#learnthesmarterway

Date: __/__/____

26/03/2025 Genetic - Algorithm

* out of 200 - features of a dataset, you have to select 6 features.

D = 200, d = 6

I. Initiation :-

- Binary
- integer → ch1. | 5 | 38 | 49 | 109 | 179 | 16 | ; Objective - f^n calculated.
- character

For e.g. TSP : 5 city | a | b | c | d | e |

II. Selection of Mating-Pool :

1. Random
2. Proportional-Based (Roulette-Wheel).
   ↳ Disadvantage :- Dominance - of certain-solution.

ch1	96
ch2	2
ch3	0.5
ch4	0.5

3. Tournament-Based → A small group of chromosomes is selected randomly & choose the best-solution.
   ↓ Disadvantage :-
   * Probability of best-fit & worst-fit sol^n being in the small group are same.

4. Rank-based-solution →

	Obj. f^ns.	Rank	Weight (whichever way you like)
ch1	96	1	10
ch2	2	2	8
ch3	0.75	3	6
ch4	0.25	4	4

III. Reproduction / Crossover - Operation :-

No. of Parents

ⓐ Asexual ⟶ P1 :

r: 0.3 0.7 0.4 0.3 0.2 0.8

| 1 | 0 | 0 | 1 | 1 | 1 |

⇓

| 1 | 1 | 0 | 1 | 1 | 0 |

** $\boxed{\underline{CR = 0.6}}$ → Mutat^n Rate

ⓑ Bisexual : Discussed

ⓒ Multisexual →

| 1 | 0 | 0 | 1 | 1 | | 0 | 1 | 0 | 1 | 0 | | 1 | 1 | 1 | 0 | 0 |

Offsprings ⇒
1 | 1 | 1 | 0 | 0 | 0 |
2 | 1 | 0 | 0 | 1 | 0 |
3 | 0 | 1 | 1 | 1 | 1 |

Point crossover
1. One-Point
2. Two-Point

1	0	0	1	1
0	0	1	1	0

=>

1	0	1	1	1
0	0	0	1	0

3. **Uniform-Crossover:**

r_i 0.1 0.7 0.8 0.6 0.3 0.2

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

=>

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

* if MR < r_i = same
else = choose from other parent.

IV. Mutation :-

1. Uniform
2. Inorder → will take 2-pos's. randomly. & interchange the values.

0	~~1~~ 0	0	1	~~0~~ 1	1

V. Selection of next-generation-chromosomes :-

1. All offsprings.
2. Some percentage of best-parent-chromosomes, should be kept.
3. Elitist-Method → Best-parent should replace the worst-offspring.
4. Hall of fame.

VI. Stopping-criteria:-

1. After a certain-no. of steps.
2. When set of chromosomes (population-set) isn't going to change.
3. When best obj. f^n. stops changing.

28/03/2025 **Differential-Evolution :-**

1. **Initialization** ⇒ No. of population = N.

$N = 200$.
$d = 5$.

** Here, these are "population-vectors", "chromosomes"-term is used in genetic-Algorithm.

35	40	125	139	140	$x_{i,G}$
29	27	99	148	96	$x_{r_1,G}$
17	08	22	33	199	$x_{r_2,G}$
192	179	187	181	97	$x_{r_3,G}$

$x_{i,G}$ = i^{th}-population-vector of generation-G.

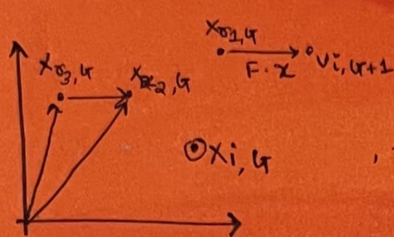
2. **Mutation** ⇒ for each population-vector $x_{i,G}$, calculate mutation-vector

$$\boxed{V_{i,G+1} = x_{r_1,G} + F(x_{r_2,G} - x_{r_3,G})}$$

, where r_1, r_2, r_3 are random-no in b/w 1 & N & not equal to 'i'.

, F = Amplification-parameter.

$0.1 < F < 2$.



, $x = (x_{r_2,G} - x_{r_3,G})$;

$\therefore V_{i,G+1} = [29 \;\; 27 \;\; 99 \;\; 148 \;\; 96] + \frac{1}{2}[-175 \;\; -171 \;\; -165 \;\; -148 \;\; 102]$

$= [\cancel{-58}^{85} \;\; \cancel{-58}^{72} \;\; 17 \;\; 74 \;\; 147]$.

↳ Rectifying ⇒ ① remove & replace all negatives with random-no.s.

② If multiple-same no.s are there, just keep one & change others with some random-no.s.

3. **crossover** ⇒

$x_{i,G}$ = | 35 | 40 | 125 | 139 | 140 |

$V_{i,G+1}$ = | 85 | 72 | 17 | 74 | 147 |

crossOver Rate (CR) = 0.6.

Trial vector/ crossOver-Vector ⇒ $U_{ij,G+1} = \begin{cases} V_{ij,G+1}, & \text{if } r_i > CR \\ x_{ij,G+1}, & \text{if } r_i \le CR \end{cases}$

→ r_i	0.2	0.4	0.7	0.9	0.3
$\therefore U_{ij,G+1}$	35	40	17	74	140

4. **selection** ⇒ $x_{i,G+1} = \begin{cases} x_{i,G}, & \text{if } f(x_{i,G}) > f(U_{i,G+1}) \\ U_{i,G+1}, & \text{else.} \end{cases}$

Next-genern i^{th}-populatn vector.

↳ Objective-fn's will be predefined by you or in question.

(contd.) → (On next to next-section)

Date: __/__/____

28/03/25 → S.C.

* changes w.r.t Genetic-Algorithms :-
  - → Mutations performed before crossover.
  - → only 1-type of crossover-possible. (Asexual).
  - → search-space is found by difference in vector.
  - → selection-for-mating pool not required.
  - → only 1-type of 'selection'- possible.

variations of DE :-

- DE/rand/1 → one-diff. operation.

Addition with one random vector.

$$V_{i,G+1} = X_{r_1,G} + F(X_{r_2,G} - X_{r_3,G})$$

- DE/rand/2

$$V_{i,G+1} = X_{r_1,G} + F_1(X_{r_2,G} - X_{r_3,G}) + F_2(X_{r_4,G} - X_{r_5,G}).$$

- DE/Best/1

$$V_{i,G+1} = X_{best,G} + F.(X_{r_2,G} - X_{r_3,G}).$$

⇒ Amplification - Factor :-

$0.1 < F < 2$

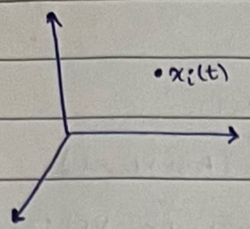
reducing effect of Difference Vector ← → maximizing effect of Difference vector.

02/04/2025 Particle - swarm - Optimization :-

↳ Population-based Evolutionary-Algorithm. mimicing the concept of flown of a swarm of birds.

↳ A particle improves its position by its own-experience and experience of swarm.

Let $x_i(t)$ ⇒ denotes position of particle 'i' at time 't'.



$\therefore \boxed{x_i(t+1) \Rightarrow x_i(t) + v_i(t+1)}$ — (1).

$i = 1$ to N, where N is the no. of particles in the swarm.

$x_i \in \mathbb{R}^D$.

Velocity :-

↳ derives the optimization process.

↳ reflects the experiential knowledge of the particle.

↳ socially exchange information from particle-neighbor.

◉ $v_i(t+1)$ = velocity of particle 'i' at time $t+1$.

(self-Improvement)

$$\therefore \boxed{v_{ij}(t+1) = v_{ij}(t) + c_1 \cdot r_{1j}(t) \cdot \underbrace{[y_{ij}(t) - x_{ij}(t)]}_{\text{cognitive component}} + c_2 \cdot r_{2j}(t) \cdot \underbrace{[\hat{y}_j(t) - x_{ij}(t)]}_{\text{social component}}}$$ — (2)

, $v_{ij}(t+1)$ = velocity of particle 'i' w.r.t dimension 'j' at time '$t+1$'.

y_i = the personal-best-position of particle 'i' since the first time step.

x_i = current-position of particle 'i'.

Minimization Problem :-

$$y_i(t+1) = \begin{cases} y_i(t) & , \text{ if } f(x_i(t+1)) > f(y_i(t)) \\ x_i(t+1) & , \text{ if } f(x_i(t+1)) < f(y_i(t)). \end{cases}$$

↳

$\hat{y}$ = Global-best-position.
= The best-position discovered by any of the particle so far.

, $f(\cdot) \Rightarrow$ Objective-funct^n.
(Minimization-f^n)

$$\hat{y}(t) \in \{y_1(t), \dots, y_N(t)\} \mid f(\hat{y}(t)) = \text{Min}\{f(y_1(t)), \dots, f(y_N(t))\}$$

r_{1j}, r_{2j} => Random-No. b/w 0 to 1.

c_1, c_2 => Positive - Acceleration - constant used to scale the contribution of the cognitive & social components respectively.

Algorithm :-

- create & initialize a D-Dimensional swarm of N-particles, & initialize parameter c_1 & c_2.

- Repeat

```
    for each particle i=01 to N
    {   if f(xi) < f(yi) then
            { yi = xi }
        if f(yi) < f(ŷ) then
            { ŷ = yi }
    }
    for each particle i=1 to N do
      {
         → update velocity using Eq.(2).
         → update position using Eq.(1).
      }
→   until (stopping-criterion).
```

◉ All particles will move / incline towards global-best-position (will have one-component that).

◉ It's a problem as solⁿ. 'll get trapped in local-optima so, it's called

PSO with Gr. Best :-

→ converge fast

→ lack of Diversity

→ susceptible to get trapped into local-optima.

Date: __/__/____

04/04/2025. * To tackle the problem of gbest PSO, we now study :-

① lbest - PSO :-

(Here 'j' denotes the component)

$$v_j(t+1) = \ldots + \ldots + c_2 r_{2j}\left[\underbrace{\tilde{y}_{ij}(t)} - x_{ij}(t)\right]$$

$\tilde{y}_{ij}(t) \Rightarrow$ best-posⁿ. within a nbrhd. of x_i till time 't'.

$N_i \Rightarrow$ Nbrhd. of particle 'i'.
$\Rightarrow$ it contains 10-different -nbrs.

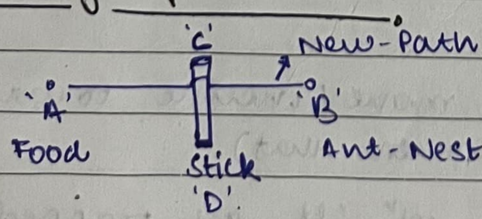
$$\tilde{y}_i \Rightarrow \tilde{y}_i \in \{y_i \mid y_i = \min f(x \mid x \in N_i)\}$$

so, the Advantage of lbest :-

1. Diversity ↑.
2. less susceptible to be trapped in local-optima.
3. convergence lately (i.e, will take more time or more iterations to converge).

* lbest => local-best
gbest => global-best

Ant-colony-Optimization :-



stigmergy : (How ants interact with each other).
- stigma → sign
- ergon → work

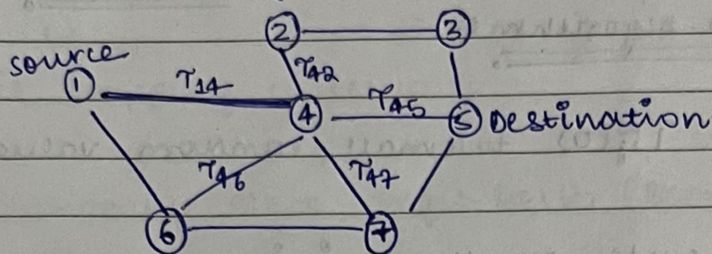
① The ants communicate via :- pheromone-trail (pheromones-deposit on the path from ants' body).

Date: __/__/____

* **Stigmergy** = it's the mechanism that mediate animal-to-animal interaction.

→ **ACO** :- proposed by Marco-Dorigo.

suppose there's graph : G(V,E) (we're finding the shortest path here).



$\tau_{ij}(t)$ = pheromone-concentration is associated with edge (i,j).

* From the possible set of paths take whose 'τ' is large. $\tau_{ij}(0)$ = A very small random-no. (close-to-'0').

(i.) Pheromone-Deposition
(ii.) Pheromone-Evaporation
} phe. conc. depends on these '2'.

$$\tau_{ij}(t+1) = \tau_{ij}(t)\cdot(1-\rho) \quad ; \quad '\rho' = \text{Evaporation-Rate.} \quad \text{— (2)}$$

* $\Delta \tau_{ij}^{k}(t)$ = Pheromone deposition by an ant 'k' at edge 'ij' of time 't'.

$\propto \frac{1}{L^k(t)}$, L^k = length of path constructed by ant 'k'. (path-src. to Dest.)

Modification of ph. concⁿ. of an edge.

* $$\boxed{\tau_{ij}(t+1) = \tau_{ij}(t) + \sum_{k=1}^{n_k} \Delta \tau_{ij}^{k}(t)} \quad \text{— (3).}$$

* consider - short path. 1
long path. 2
] consider both ant have same pheromone extract & cover path in same time.

⊙ Longer path will have lesser-deposⁿ.

Date: __/__/____

* $p_{ij}^{k}(t) = \begin{cases} \tau_{ij}^{\alpha}(t), & \text{if } j \in N_i^k \\ 0, & \text{if } j \notin N_i^k \end{cases}$; α = Amplific^n. factor. —(1).

N_i^k → Nbrs. (possible-paths).

transition probability (for taking dec^n.)

Determines which path an ant chooses to move to the nxt. node.

thus, ACO-Algorithm :-

[Initialize $\tau_{ij}(0)$ to small random-value $(t=0)$

[Place n_k ants on the original node.

Repeat

for each ant $k = 1$ to n_k do

$x^k(t) = 0$; # sol^n. for ant 'k'.

repeat

→ Select next-node based on probability calculated by eq^n. (1).

→ Add edge (i,j) to $x^k(t)$.

→ Until destination-node has been reached

* some sol^{n}s. may contain loops :-

so :

Remove all loops from $x^k(t)$.

calculate path length $f(x^k(t))$;

End;

for each edge (i,j) of the graph do.

Reduce the pheromone $\tau_{ij}(t)$ using eq^n. (2)

End;

#learnthesmarterway

Date: __/__/____

for each ant $k = 1, \ldots, n_k$ do.

for each edge (i, j) of $x^k(t)$ do.

$$\Delta \tau_{ij}^{k} = \frac{1}{f(x^k(t))}.$$

update τ_{ij} using eqn. (1).

End;

$t = t + 1$;

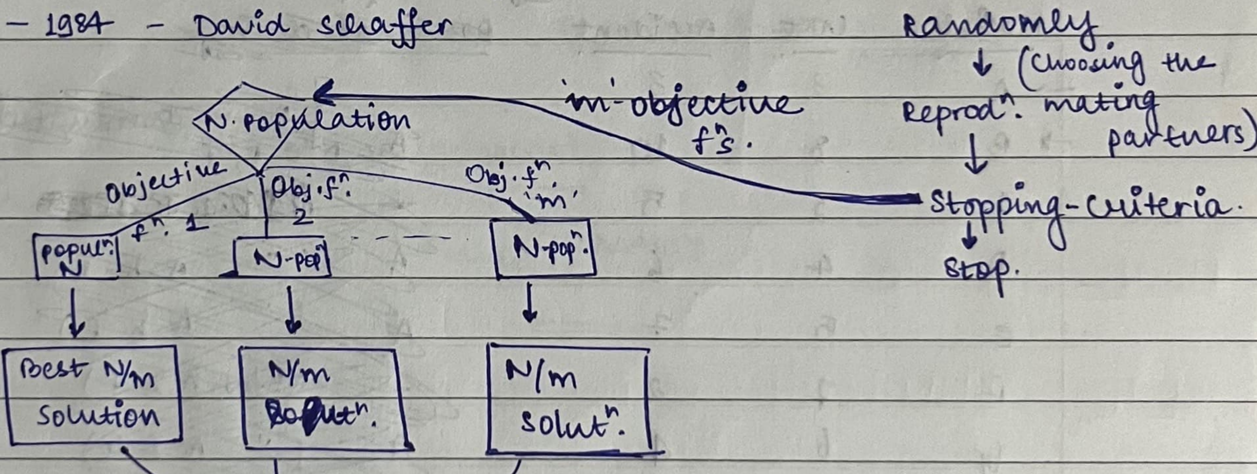
until stopping criteria;

NOTE :- * Evolutionary - Algos. Engel-Berth (BOOK). {for all Algos.}.

07/04/2025 **Multi-Objective-Function :-**

* Vector - Evaluated - Genetic - Algorithm (VEGA)

— 1984 — David schaffer



— 1896 — Vilfredo Pareto ⇒ for multi-objective-fns., there is a set of solutions, not a single - solution.

⇒ For multi-obj. fn., the optimal set contains a set of solutions, where each element is not dominated by any other soln. on the search space w.r.t. any of the obj. funcn.

The Pareto-optimal front is the set of all non-dominated solutions — the "best trade-offs" among the objectives.

A solution is non-dominated (or Pareto-optimal) if no other solution is better in all objectives.

#learnthesmarterway

Date: __/__/____

* Dominated and Non-Dominated-Solution :-

Let there be m - obj. f^n. (Maximization),

$f_i(\cdot)$, i = 1 to m.

NOTE :- In minimiz^n problem, this'll be vice-versa.

Let there be two-solutions u^1 & u^2. u^1 will dominate u^2 if following conditions satisfy.

1. $f_i(u^1) \nless f_i(u^2)$, i.e., $f_i(u^1) \geq f_i(u^2)$, i = 1 to m.

2. Atleast one-objective-f^n. $f(u^1) > f(u^2)$.

For e.g. car-Data-set :-

(Q.) Find pareto-optimal-set for this problem?

~~Dominated by~~

~~'B' is dominated by 'A'.~~

Model	Cost	Accident
→ A	3	3
→ B	8	10
C	2	5
D	4	6
E	5	7
F	7	2
G	6	4
H	9	4

	Dominated-by
(A)	
(B)	A, C, D, E, F, G
(C)	
(D)	A, C
(E)	A, C, D
(F)	
(G)	A
(H)	A, F, G.

∴ Pareto optimal set → {A, C, F}

because these are not dominated by anyone.

#learnthesmarterway

→ Performing Genetic-Algorithm over Multi-obj. f^n problem

- **Weighted-sum-Approach :-** (Ishibuchi & Murata, 1996).

m-obj. f^n. $f_i(\cdot)$; $i = 1$ to m

Population = N.

Pareto-optimal set contains 'n'-solutions.

keep-n-solution seperately and (N-n)-chromosomes.

$\underline{x} \Rightarrow$ chromosome.

$$f(\underline{x}) = w_1 f_1(\underline{x}) + w_2 f_2(\underline{x}) + \dots + w_m f_m(\underline{x})$$
$$= \sum_{i=1}^{m} w_i \cdot f_i(\underline{x}).$$

$$w_i = \frac{r_i}{\sum_{i=1}^{m} r_i}$$; $r_1, r_2, \dots, r_m \Rightarrow$ Random-Number.

$$\underbrace{p(\underline{x})}_{} = \frac{f(\underline{x}) - f_{min}(\text{current})}{\sum_{x \in \text{current}} \{f(x) - f_{min}(\text{current})\}}$$

relative weighted fitness value of the chromosome.

After this, we go for "Selection for Mating-Pool".

↓

Crossover

↓

Mutation

↓

→ (N-n) offsprings

+ n pareto-optimal-sol^n.

⇓

N-solution.

#learnthesmarterway

09/04/2025

Date: __/__/____

In ACO (studied prev.) :-

Src. Dest^n.

$$1 \xrightarrow{\tau_{14}} 4 \xrightarrow{\tau_{45}} 5$$

$$\Delta \tau_{ij}^{k}(t) = \frac{1}{f(x^k(t))} = \frac{1}{2}$$

* Here, it's a unit-edge.

$$f(x^k(t)) = \text{Path-length} = 1+1 = 2$$

$$\tau_{14}(t+1) = \tau_{14}(t) + \underbrace{\Delta \tau_{14}^{k}(t)}$$

for k^{th}-ant, pheromone density on this path.

Src. Dest^n.

$$1 \xrightarrow[5]{\tau_{14}} 4 \xrightarrow[10]{\tau_{45}} 5$$

* In weighted-edge-graph ; $f(x^k(t)) = 5 + 10 = 15$.

$$\boxed{\Delta \tau_{ij}^{k}(t) = \frac{w_{ij}}{f(x^k(t))}}$$

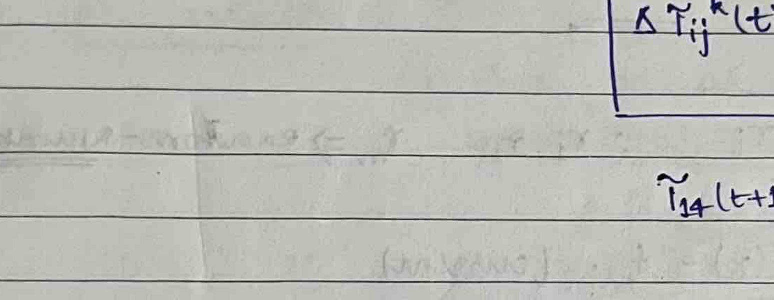
$$\tau_{14}(t+1) = \tau_{14}(t) + \frac{5}{15}$$

$$\tau_{45}(t+1) = \tau_{45}(t) + \frac{10}{15}.$$

11/04/25. ⊙ Taught through slides ... (Neural-Networks)

16/04/25. **ANN (Artificial-Neural-Network) :-**

- Perception :-



* δ = [redacted] Actual - Predicted $= y_i - \hat{y}_i$

⊙ $w_i(t+1) = w_i(t) + \underbrace{\Delta w_i}$

$$\{ \Delta w_i = \eta \, \delta \, x_i \}$$

, η = Learning-Rate (constant-Parameter)

δ = error $= y_i - \hat{y}_i$

#learnthesmarterway

Date: __/__/____

Eg:- AND

x_1	x_2	y
0	0	0
0	1	0
1	0	0
1	1	1

Assume: $\theta = 1.5$, $\eta = 0.5$.

Iteration-01 :- (1) $w_1 = 0$, $w_2 = 1$ | $x_1 = 0 = x_2$

$\Sigma w_i \cdot x_i = 0 + 0 = 0 < 1.5 \Rightarrow y = 0$.

$\delta = 0 - 0 = 0$

$\therefore$ no changes in the output.

(2) $x_1 = 0$, $x_2 = 1$ | $w_1 = 0$, $w_2 = 1$

$\Sigma w_i \cdot x_i = 1 < 1.5 \Rightarrow y = 0$.

$\therefore \delta = 0 - 0$, No change.

(3) $x_2 = 0$, $x_1 = 1$; $w_1 = 0$, $w_2 = 1$

$\Sigma w_i x_i = \underset{1.5}{0} \Rightarrow y = 0$ $\therefore \delta = 0 - 0 = 0$

$\therefore$ NO change

(4) $x_1 = 1$, $x_2 = 1$ | $w_1 = 0$, $w_2 = 1$

$\Sigma w_i \cdot x_i = 1 < 1.5 \Rightarrow y = 0$.

$\therefore \delta = 1 - 0 = \underline{1}$,

$\therefore w_1 = 0 + 0.5(1)(1) = \underline{0.5}$.

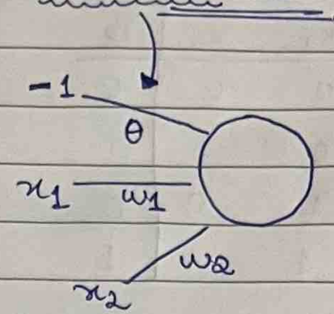
$w_2 = 1 + 0.5(1)(1) = \underline{1.5}$.

Iteration-02 :- $w_1 = 0.5$, $w_2 = 1.5$.

- If η is very high, 'w_i'- value will fluctuate b/w values.
- If η is very small, it'll take very long time to converge to actual-value.

⇒ Except the i/p-node, there's a bias node as well :-

i/p - value = -1
weight = θ

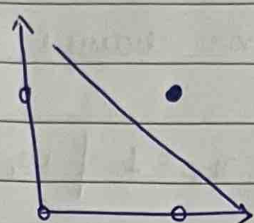


$$\sum_{i=1}^{n} w_i \cdot x_i < \theta$$

$$\downarrow$$

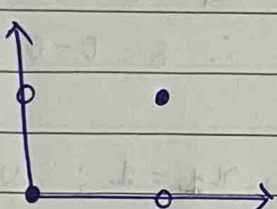
$$\sum_{i=1}^{n} w_i \cdot x_i - \theta < 0$$

→ AND-Gate

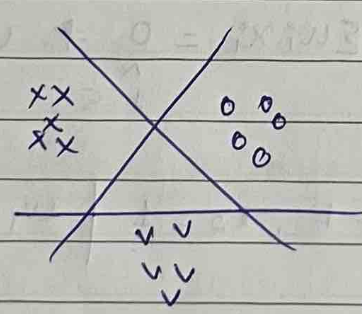


→ can be seperated using single-perceptron.

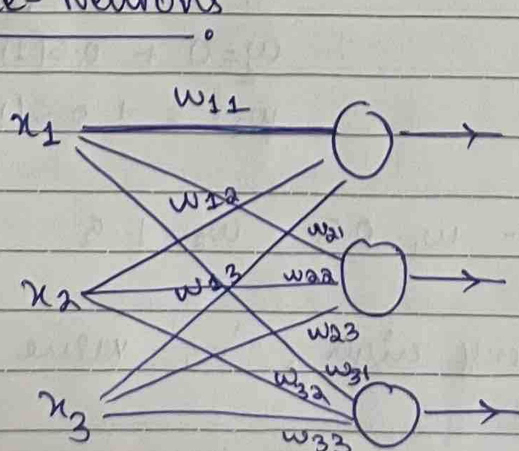
→ XOR



→ can't be seperated using a single-perceptron.



⇒ Multiple-Neurons



- Multi-layer gives convex-reg^n. as o/p.
- 2- convex-reg^n. give concave reg^n. as o/p.