1. Introduction of Soft Computing

* **Soft Computing** is a branch of Computer Science that tries to make Computers think like humans.

- It deals with approximate reasoning, learning and uncertainty Instead of using Strict rules.

[Soft Computing Computer ला manasa सारखी विचार करण्यासाठी उपयोगी असणारी branch aye]

e.g. "weather is warm" this Statement Can be understand by Soft Computing by fuzzy logic instead of exact temp. like $30^\circ C$.

* Soft Computing	* Hard Computing
1] Based on ~~exact~~ flexible Human like reasoning	Based on exact logic and rules.
2] Approximations are okay	Approximations are not allow
3] Output of this is probabilistic (i.e. best possible answer)	Output of this is deterministic (i.e. one correct answer)
4] eg. AI, fuzzy logic, Neural Network	eg. Traditional programming, arithmetic calculations
5] e.g. weather predictor ("It might be rain today")	e.g. A Calculator (2+2=4)

→ FL, NN, EC, Proboblistic reasoning

* Types of Soft Computing Techniques

1] Fuzzy logic (FL) :-

- It works on degrees of truth Not just true or False

e.g "The water is Somewhat Hot"

2] Neural Network (NN) :-

- Inspired by human brain, used to recognize patterns, images or voices

e.g face recognization, Detecting handwriting

3] Evolutionary Computing (EC) :-

- Based on Natural evolution
- uses process like Selection, mutation, and crossover to find the best Solution

e.g Genetic algorithms used in optimization problems

4] Probabilistic ~~Revenge~~ Reasoning :-

- Deals with uncertainty using probabilities

e.g. ~~Deals~~ Predicting if a patient has a disease based on symptoms.

* Basic tools of Soft Computing

1] fuzzy logic :- Handles imprecise information

2] Neural Network :- learn from data to make predictions

3] Evolutionary algorithms :- find best sol^n through trial and improvement
[just like Survival of fittest]

PAGE NO.
DATE:

In short
([illegible])

[illegible] System [illegible]
[illegible] decision [illegible]

* Intelligent agents :-

- An intelligent agent is a system that observe it's environment and take action to acheive it's goal

* Agent :- The decision maker (e.g. robot or software program)

* Environment :- The world where the agent works (e.g. room, internet, game)

* Structure of an agent.

an agent usually has :

1] Sensors :- to observe the environment
2] Actuators :- to take actions
3] Brain / programm :- to decide what to do.

e.g. A Vaccum cleaning robot = agent
Room :- environment
Sensors :- dust detector
Actuator :- wheels, cleaner
Program :- logic to clean efficiently

In short
(dokyani khelnura).

PAGE NO.
DATE:

म्हणजे असा system जो obser krun कार्य करायचं याचा decision gheto

* **Intelligent agents :-**

- An intelligent agent is a system that observe it's environment and take action to acheive it's goal

* **Agent :-** The decision maker (e.g. robot or software programm.

* **Environment :-** The world where the agent works (e.g. room, internet, game)

* **Structure of an agent.**

an agent usually has:

1] Sensors :- to observe the environment
2] Actuators :- to take actions
3] Brain / programm :- to decide what to do.

e.g. A Vaccum cleaning robot = agent
Room :- environment
Sensors :- dust detector
Actuator :- wheels, cleaner
Program :- logic to clean efficiently

PAGE NO.
DATE:

* Types of agent :-

(if ~~dirty~~ → clean)

1] Simple Reflex Agent :- act only on current input

2] model based agent :- Rember Past info
[model chta history bayat thevtay]

3] Goal based agent :- act to reach a goal
[goal sathi kaam krtaye]

4] utility based agent :- choose the most beneficial actio
[best action choose krtaye]

5] learning agent :- improve over time.
[tin Experience matter]

Applications of soft Computing :-

1] medical diagnosis [Predicting disease]
2] weather forecasting.
3] stock market prediction
4] Robotics (Path finding)
5] Image recognization (face or plant disease detection)
6] Industrial process control.

Fuzzy logic and fuzzy Systems

Unit-2

PAGE NO.
DATE:

* Introduction to fuzzy Set :-

In real life, many things are not Completely true or False they are "partly true" This "partly true" Concept is handled by fuzzy Set theory

e.g Temprature :- "Hot" or "Cold" ?.
If it's 29°C, it's not fully Hot, but not Cold either - it's partly hot.

* fuzzy Set Theory

- A fuzzy Set allows partial membership of elements in a Set.

- each element has a membership value (μ) between 0 and 1

Temprature (°C)	Membership in "HOT" set
20°C	0 (not hot)
25°C	0.4 (Slightly hot)
30°C	0.8 (Quite hot)
35°C	1 (complete hot)

↳ so fuzzy set describe how "hot" something is ⟶ not just Yes / No.

It Handles partial truth

PAGE NO.
DATE:

Fuzzy Set	Crisp Set
1] Membership value is between 0 & 1	1] It is ~~between~~ either 0 or 1
2] logic of this Set is exactly True or false	2] It's logic is ~~partly~~ partially True
3] Boundary of this Set is Blurred (vague)	3] Boundary of this set is sharp (clear)
4] E.g. He is tall with 0.8 degree truth	4] E.g. He is tall (Yes or No)

* Crisp Relations and fuzzy Relations :-

1] Crisp relation :- Either the elements are related (1) or not (0)

E.g. $(A, B) = 1$ if $A > B$

2] Fuzzy relation :- The relationship has degree of truth.

E.g. If $A = 30°C$ and $B = 35°C$

→ "A is almost equal to B" $= 0.7$

→ In short fuzzy relations describe how strongly ~~to~~ two things are related.

PAGE NO.
DATE:

* Membership functions

- A membership function (μ) defines how each element belongs to a fuzzy set.

- It gives a number between 0 and 1.

Common Shapes :-

1] Triangular △
2] Trapezoidal ⏢
3] Gaussian (bell-shaped)

* Example (triangular) :-

For "warm temprature" :

- $\mu = 0$ when temp < 20
- μ increases from 20 to 30
- $\mu = 1$ at 30
- μ decreases after 30.

Decision making with uncertainty

PAGE NO.
DATE:

* Fuzzy logic :-

Fuzzy logic extends classical logic by allowing degrees of truth instead of just True / False.

E.g. If temp = "warm" (0.7 truth), fan speed = "med"
If temp = "hot" (1.0 truth), fan speed = "High".

IF then logic for fuzzy reasoning

* Fuzzy rules and fuzzy reasoning :-

- Fuzzy logic uses IF-THEN rules to make decisions.

E.g.

- The System then uses fuzzy reasoning to decide output based on these rules.

Example Rules :-

1] If temp is HOT THEN fan Speed is High.
2] ______ warm THEN ______ medium.
3] ______ Cold THEN ______ low

* Fuzzy Inference System (FIS)

- A fuzzy Inference System is the main part of fuzzy logic control.

- It performs fuzzy reasoning using following steps

1] **Fuzzification** :- Convert Crisp Inputs (numeric values) into fuzzy values using membership function.
. E.g. Temp : 28°C → "warm = 0.6", "Hot = 0.4"

2] **Rule Evaluation (Inference)** :- Apply fuzzy If → Then rules using fuzzified ~~rules~~ values
E.g. "If warm Then fan = medium (0.6 truth)

3] **Aggregation** :- Combine all the rules output into one fuzzy set

4] **Deffuzzification** :- Convert the fuzzy output back into a crisp (numerical) output
E.g. fan Speed = 65%

PAGE NO.
DATE:

* Fuzzy Controllers

A fuzzy Controller is a System that uses fuzzy logic to control machines or processes

E.g. Automatic fan Controller.

- Input :- Room temprature
- Output :- Fan Speed.

Steps :-

1] Fuzzify temprature (HOT, warm, cold)
2] Apply fuzzy rules
3] Defuzzify to get exact fan speed

↳ Result :- fan automatically adjust smoothly
→ no sudden ON / OFF.

PAGE NO.
DATE:

Unit - III :- Hybrid Systems and Evolutionary Approaches

* **Hybrid Systems :-** each Soft computing technique (like NN, fuzzy logic, Genetic algorithm) has it's strengths and weaknesses. By Combine them all, we get a Hybrid System - which uses the best feature of each.

Technique	strength	weakness
1] NN	Learns from data automatically ↳ patterns	lacks - human like reasoning
2] fuzzy logic	Handles vague, Human like reasoning	Needs expert rules
3] Genetic algorithm	Find best sol^n by trial and error	Slow to converge

def :- Combination of 2 or more Soft computing techniques to create a Smarter and more efficient System Is known as Hybrid System

e.g

Neuro - fuzzy System = NN + fuzzy logic

Genetic - fuzzy System = Genetic algorithm + fuzzy logic

e.g. a washing machine that adjust wash time and detergent level
fuzzy [illegible] + learn from previous (NN)

PAGE NO.
DATE:

* Neuro-fuzzy System

:- A Neuro fuzzy System Combines
NN + fuzzy logic
learns from data — reasons like human using fuzzy rules

together it becomes a system that learns fuzzy rules automatically from data

* How it works ?
1] Input data (e.g. temp, humidity)
2] fuzzification
3] Neural network adjust the fuzzy rules and membership functions based on training data
4] Output (e.g "Fan Speed = Medium")

Eg. Automatic AC System
1] Input :- Room temp, humidity
2] Output :- Cooling level

System learns from user preferences and adjusts fuzzy rules automatically

Adaptive Neuro fuzzy Inference system

PAGE NO.
DATE:

* Architecture of ANFIS

* ANFIS :- Adaptive Neuro-fuzzy Inference System
→ A special model that combines
1] ~~Input to~~ Neural Networks and fuzzy logic

Layers in ANFIS :-

1] Input layer :- takes input variable
2] fuzzification layer :- Convert input to fuzzy set
3] Rule layer :- apply fuzzy If → Then rules
4] Normalization layer :- calculates strength of each rule
5] Defuzzification layer :- combines result into single crisp output

→ It is like fuzzy system whose parameters are trained automatically like a Neural Network.

* Genetic-fuzzy System :- (GFS)

* A Genetic-fuzzy system combines :
  - Fuzzy logic :- for decision making using rules
  - Genetic algorithm :- for optimizing ~~rules~~ fuzzy rules and membership functions

PAGE NO.
DATE:

* How It Works :-

1. Start with a set of fuzzy rules and membership functions.
2. use a Genetic Algorithm (GA) operations.:
   - Selection: choose best-performing rules.
   - Crossover: mix two good rules to create new one
   - Mutation: make Small random changes

3 Over generations, GA improves fuzzy rules to make the System more accurate

E.g. Crop disease detection System.
→ fuzzy logic predicts disease level
→ Genetic algorithm tunes fuzzy rules automatically for best accuracy

* Applications of Hybrid Systems.

Area	E.g.
1] Agriculture	Disease detection in plants
2] Medicine	Diagnosis Systems using neuro-fuzzy models.
3] Engineering	Automatic Control Systems (AC, washing machin)
4] Finance	Stock-market prediction using neuro-fuzzy
5] Robotics	Path planning & obstacle avoidance

PAGE NO.
DATE:

* Advantages of Genetic- fuzzy Systems

1] Can learn and adapt :- no need to define perfect rules Manually

2] Give more accurate and optimized ~~rules~~ results

3] works well in Complex and uncertain environment

4] Combines human-like reasoning (fuzzy logic) with evolutionary Search (GA).

5] Can Solve problems where mathematical models are hard to define.

* Case studies (Real world example)

1] Medical diagnosis.

- Neuro- fuzzy System analyze patients Symptoms and medical data
- learn from Historical cases to improve diagnosis accuracy.

2] Self- driving Car

- fuzzy logic decides "How fast to go"
- neural Network learn from road data
- GA optimizes fuzzy rules for best driving control.

* Rough set theory is help us make smart decisions even when data is incomplete or uncertain, It cleans the data, finds key attributes, & discover meaningful decision rules

PAGE NO.

DATE:

Unit V - Rough Set.

- A Rough Set is a mathematical tool used to deal with uncertainity & incomplete information

- It help us classify and analyze data when we are not 100 % sure about something

e.g. Conditions like knowing marks of only one subject of students & we have to find out who is good student

Def :- It is a Set that can not be clearly described using the available information.

It is defined by two sets :-

1] lower approximation :- elements that definitely (certain part) belong to the set.

2] Upper approximation :- elements that possibly belong (possible part) to the set

↳ Rough Set is belong between these two approximations.

e.g.

Student	Mark in one subject	category
A	90	Good
B	80	Good
C	50	Avg.
D	40	poor
E	70	Good

Lower app. :- {A, B} ⟶ definitely good

upper app :- {A, B, E} ⟶ possibly good

PAGE NO.
DATE:

* **Boundary Region :-**

The part where we are not Sure whether elements belong to the Set or Not

The Boundary region = (upper Approximation − lower Approximation)

In previous example :-
Boundary Region = {E}
↳ because we are not fully sure if E is "Good" or not

* If the boundary Region is empty, that means the Set is crisp [No uncertainty]
* If it's Not empty, the Set is Rough (has uncertainty)

* **Definability :-**

A Set is definable if it can be described exactly using given data If not it's roughly definable

e.g. If all students are either Good or not good then the Set is definable degree of uncertainty

PAGE NO.
DATE:

* **Roughness :-** Roughness tells us How uncertain or vague a Set is

↳ It measures how "rough" the set is between its upper and lower approximations.

Formula (simple idea) :-

$$\text{Roughness} = 1 - \frac{|\text{lower appr.}|}{|\text{upper appr.}|}$$

e.g. In previous example,

$$1 - \frac{2}{3} = \frac{1}{3} = 0.33$$

↳ Here, if

Roughness has High value → more uncertainty
lower value → more certainty

↳ जेव्हा आपल्याकडे data madhe khup attributes असतात तेव्हा काही attribute He reduntant mhnject important nstat Teva Rough set theory data Simplify krun important attribute शोधण्यासाठी मदत करतात.

* **Reduct :-** a reduct is a minimum set of attributes that can still classify data as accurately as the full set.

— It help us to reduce complexity while keeping the same decision power.

PAGE NO.
DATE:

E.g If we have attributes → mark, attendance, Discip-lane

and we find that marks & Disciplane alone can classify students well, then,
{marks, Disciplane} is Reduct

* Core :-

The Core is set of most important attributes —
the ones that appear in all reducts.

↳ core attribute વ્યા શિવાય correct classification Possiblech nhiye

$\boxed{\text{Core} \subseteq \text{Reduct}}$

* Decision Matrices :-

In Rough Set theory, data is usually Represented in Decision Table (or matrix)

— A decision matrix contains :-

1] Condition attributes :- input features (mark, attend)

2] decision attribute :- output or class (like pass / fail)

PAGE NO.
DATE:

Example Decision Matrix :-

Student	Mark	attendance	Decision
A	High	Good	Pass
B	medium	Good	Pass
C	low	Poor	fail
D	High	Poor	Pass

We use this table to :-

1] Find patterns or Rules (If Marks = High and attendance = Good then Pass)

2] Reduce unnecessary ~~of~~ attributes.
3] Handles uncertain or missing data.

* Applications of Rough Set theory :-

Area	Ex.
1] Data mining	Remove redundant features, find hidden patterns.
2] machine learning	feature Selection before training a model.
3] Medical diagnosis	classify patients based on symptoms even if some info is missing.
4] finance	loan approval based on partial Info
5] Engineering	Fault detection systems where censor data is uncertain

UNIT IV :- Introduction to optimization ~~learning~~ algorithms.

PAGE NO.
DATE:

* This chapter is all about finding best possible solⁿ (called optimum) for a given problem - Whether that's minimum cost, maximum efficiency or best performance

* **Optimization** :- It means finding the best solⁿ from all possible solutions.

It is used in -
1] Engineering design :- minimize material Cost
2] Machine learning :- minimize error
3] Production :- maximize Profit
4] Route planning :- shortest path

e.g. You want to design a Car engine that gives maximum mileage while keeping Cost low
- That's an optimization problem

* **Basic terms in Optimization** :-

1] **Design variable** :- The parameters we can change to optimize the output
e.g. Speed, temp, pressure, etc

2] **Objective function** :- The function to be optimized (minimized or maximized)
e.g. minimize cost = $f(x, y)$

3] **Constraints** :- Conditions or limits within which the Solⁿ must lie
e.g. $x+y \le 10$, $x \ge 0$

6] **Feasible Region** :- The Region where all constraint are satisfied.

⟶ Area under the constraint lines

5] **Optimum point** :- The point that gives the best (min / max) value of the objective function

↳ minimum cost point

7 **Problem formulation** :-

to solve a problem, you must formulate it properly.

Steps :-

1] Identify design variables ⟶ e.g. x_1, x_2

2] Define objective function ⟶ e.g. minimize $f(x_1, x_2) = x_1^2 + 3x_2$

3] write constraints ⟶ $g_1(x_1, x_2) \leq 0$, $h_1(x_1, x_2) = 0$

4] Specify variable limits ⟶ e.g. $x_1 \geq 0$, $x_2 \leq 5$

e.g. minimize cost :

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

Subject to :

$x_1 + x_2 \geq 1$ and $x_1, x_2 \geq 0$.

* Types of Optimization Problem :-

1] Single variable Optimization → only one variable (x)
2] multivariable optimization → more than one $(x_1, x_2 \dots)$
3] Constrained optimization → has conditions
4] unconstrained optimization → no restrictions.

* Single variable Optimization :-

e.g. find, min of $f(x) = x^2 - 4x + 5$

Steps :-

1] Differentiate → $f'(x) = 2x - 4$

2] Set, $f'(x) = 0$

$2x - 4 = 0$

$x = 2$

3] Check second derivative

$f''(x) = 2$

answer is +ve (minimum)

↳ So, minimum occurs at $x = 2$

* Multivariable Optimization :-

e.g. $f(x_1, x_2) = x_1^2 + x_2^2 - 4x_1 + 6x_2 + 13$

Partial derivatives :

$\frac{\partial f}{\partial x_1} = 2x_1 - 4$ $\qquad$ $\frac{\partial f}{\partial x_2} = 2x_2 + 6$

$2x_1 - 4 = 0$, $2x_2 + 6 = 0$

$x_1 = 2$ $\qquad$ $x_2 = -3$

↳ minimum point $(2, 3)$

PAGE NO.
DATE:

* Constrained Optimization :-

when there are constraints, we can not choose any value of variable freely.

for such cases, special techniques are used :

1] lagrange multiplier method
2] Penalty function method
3] Kuhn - Tucker (KKT) conditions.

need to expand ~~this~~ concept.

E.g. minimize $f(x,y) = x^2+y^2$
Subject to : $x+y=1$

use lagrange method :

$$L = x^2+y^2+\lambda(x+y-1)$$

↳ take partial derivatives and solve simultaneously

e.g. minimize $f(x,y) = x^2+y^2$
Subject to : $x+y=10$
form lagrange functions :

$$L = x^2+y^2+\lambda(x+y-10)$$

Now,

$$\frac{\partial L}{\partial x} = 2x+\lambda = 0 \quad \text{--- (1)}$$

$$\frac{\partial L}{\partial y} = 2y+\lambda = 0 \quad \text{--- (2)}$$

$$\frac{\partial L}{\partial \lambda} = x+y-10 = 0 \quad \text{--- (3)}$$

Now, $f(5,5) = 50$ which is +ve

$\therefore$ minimum = 50 at (5,5)

from eq^n (1) & (2)

$2x+\lambda = 2y+\lambda = 0$

$\boxed{x=y}$

from (3)

$x+y=10$

$2x=10$

$x=5$
$y=5$

It is a mathematical test for best point

* Optimality Criteria :-

to check if a point is optimal, we use certain rules :

1] First Derivative Test : $f'(x) = 0$

2] Second Derivative Test : $f''(x) > 0 \rightarrow$ minimize,
$f''(x) < 0 \longrightarrow$ maximize um.

3] for multi-variable : check positive definiteness of Hessian matrix → need to expand this concept

- If all positive → minimum
- If all negative → maximum

e.g. for, $f(x) = x^2 - 4x + 5$

$f'(x) = 2x - 4 = 0$

$f''(x) = 2$

Here,

$f''(x) > 0 \longrightarrow \therefore$ minimize value

$\therefore \boxed{x = 2 \text{ is optimal}}$

It is an algorithmic not numeric method.

* Direct Search methods :-

These methods do not need derivatives
↳ (useful when $f(x)$ is complex)

e.g. 1] Fibonacci Search
2] Golden Section Search
3] Pattern Search
4] Simplex Search (Nelder - mead method)

They directly test points and move toward the region where the function is better.

1] Simple Idea :-

1] Try few points
2] keep the best one
3] Search around it again

e.g. find minimum of $f(x) = (x-3)^2+1$

Try $x = 1, 2, 3, 4$

x	$f(x)$
1	5
2	2
3	1
4	2

minimum $= 1$ at $x = 3$

PAGE NO.
DATE:

* Simplex Search method :-

Simplex Search method is direct Search technique used for multivariable, unconstrained optimization

It forms a geometric shape (Δ, □, etc) and moves it around the function space to find the best point.

Steps :-

1] Start with n+1 points for n variable
2] Evaluate function values.
3] Reflect, expand or contract the Simplex based on Results.
4] Repeat untill the Convergence

↳ works even when derivatives are not available.

~~Here~~ only ~~x not y~~ is compared to non zero value also

* Linear Programming Problem (LPP)

Linear programming is an optimization technique where both :-

1] objective function
2] The constraints

are linear

e.g. maximize $Z = 3x_1 + 2x_2$

Subject to :

$x_1 + x_2 \le 4$

$x_1 \le 2, \quad x_2 \le 3, \quad x_1, x_2 \ge 0$

Steps :-

1] plot constraints
2] Identify feasible region
3] Evaluate Z at corner points
4] The corner point with best Z is the optimum Soln

This method is ~~Both LPP~~ by graphically

Instead of it

Next method is

Numerically.

* no need of graph
* x & y both are separately computed to non-zero values.

used when objectives & constraints are linear

* 2 or more variable

PAGE NO.
DATE:

* Simplex method (for LPP)

[simple case]

It is an iterative algorithm used to solve linear programming problems efficiently.

Steps:

1] Convert inequalities into equalities
↳ (add slack variable)

2] Set up the initial simplex table.

3] Identify entering and leaving variables.

4] Pivot ~~the~~ to update the table.

5] Continue untill the all coefficient in the objective row are positive (for max problem)

↳ This gives the optimal sol^n for linear problems.

e.g.

maximize, $Z = 3x + 2y$

Subject to:

$x + y \le 4$, $x \le 2$, $x, y \ge 0$, $y \le 3$

Here,

~~feasible points are~~

~~(0,4), (1,3), (2,2), (0,0), (2,0)~~

Convert inequalities to equalities.

$x + y = 4$, $y = 3$, $y = 0$

$x = 2$, $x = 0$.

Now, find all intersections by solving all pairs of equations.

Pair	Solve	Result
1] $x+y=4, x=0$	$y=4$	(0,4)
2] $x+y=4, y=0$	$x=4$	(4,0)
3] $x+y=4, y=3$	$x=1$	(1,3)
4] $x+y=4, x=2$	$y=2$	(2,2)
5] $x=2, y=3$	—	(2,3)
6] $x=2, y=0$	—	(2,0)
7] $x=0, y=0$	—	(0,0)
8] $x=0, y=3$	—	(0,3)

check feasibility

Now, check which points satisfy all inequalities.

Point	$x+y \le 4$	$x \le 2$	$y \le 3$	feasible
(0,4)	✓	✓	✗	✗
(4,0)	✓	✗	✓	✗
(1,3)	✓	✓	✓	✓
(2,2)	✓	✓	✓	✓
(2,3)	✗	✓	✓	✗
(2,0)	✓	✓	✓	✓
(0,0)	✓	✓	✓	✓
(0,3)	✓	✓	✓	✓

∴ feasible points are

(1,3), (2,2), (2,0), (0,0), (0,3)

PAGE NO.
DATE:

Compute, $z = 3x + 2y$ at each point

Point	$z = 3x + 2y$
(0,0)	0
(2,0)	6
(0,3)	6
(1,3)	7
(2,2)	10

so, $Z_{max} = 10$ at $(x,y) = (2,2)$

Problem formulation

Q. A farmer has 100m of fencing to make a rectangular field. maximize the area

→

let, length $= x$, breadth $= y$

Constraints :-

$$2x + 2y = 100$$

objective :-

$$A = xy$$

we have to maximize A.

from constraint

$$y = 50 - x$$

so,

$$A = 50x - x^2$$

$$\frac{\partial A}{\partial x} = 50 - 2x$$

$$2x - 50 = 0$$

$$x = 25$$

$$y = 25$$

∴ maximum area

$= 25 \times 25$

$= 625\ m^2$