Define, explain, discuss with suitable example

1. **Soft Computing**

- Soft computing is an emerging approach to computing which parallels the remarkable ability of the human mind to reason and learn in an environment of and imprecision
- Guiding principles of soft computing: exploit the tolerance for imprecision, uncertainty and partial truth to achieve tractability, robustness and low solution cost.
- Soft computing is a consortium of methodologies, which either singly or in combination, serve to provide effective tools
- Soft computing is better used in solving real-world problems as it is a randomly defined process that can be analyzed statistically but not with precision.
- Soft computing is based on the model of the human mind where it has probabilistic reasoning, fuzzy logic, and uses multivalued logic.
- Soft computing can handle an abundance of data and handles multiple computations which might not be exact in a parallel way
- soft computing tolerance of uncertainty and imprecision is estimated to achieve Machine Intelligence Quotient (MIQ) and lower cost. It also provides better communication.
- Soft computing resolves the nonlinear issues that involve uncertainty and impreciseness as it has human-like intelligence that can resolve the real-life issue.

2. **Hard Computing**

- Hard computing is best for solving the mathematical problems which dont solve the problems of the real world.
- Hard computing relies on binary logic and predefined instructions like a numerical analysis and brisk software and uses two-valued logic
- Hard computing needs exact input of the data and is sequential;
- Hard computing takes a lot of time to complete tasks and is costly
- Hard computing is best suited for solving mathematical problems which give some precise answers.
- Hard computing takes a lot of time in computing as it requires the stated analytical model and the model soft computing is based on is that of human intelligence.

3. **Artificial Intelligence (AI)**

• AI aims at emulating human intelligence so as to enable them to act and think like human beings • AI is a vast discipline of knowledge that includes logic, deductive reasoning, expert systems, case-based reasoning, machine learning, planning, intelligent search and perception building • AI is a combination of several research disciplines, such as computer science, physiology, philosophy, sociology and biology • Conventional AI mostly involves methods now classified as machine learning, characterized by formalism and statistical analysis • This is also known as symbolic AI, logical AI, neat AI and GoodOld-Fashioned AI

4. **Computational Intelligence (CI)**

• CI can be defined as the computational models and tools of intelligence capable of inputting raw numerical sensory data, processing them by exploiting representational parallelism, and pipelining of the problem, generating reliable and timely responses and withstanding high fault tolerance • CI is the study of adaptive mechanisms to enable or facilitate intelligent behavior in complex and changing environments • These mechanisms include paradigms that exhibit an ability to learn or adapt to new situations, to generalize, abstract, discover and associate • ANNs, fuzzy systems, EC, SI and AISs are the dominating paradigms of CI.

5. **Swarm intelligence**

• SI originated from the study of colonies (of ants, bees and termites, etc.) or swarms of social organisms flock of birds, school of fish • Studies of the social behavior of organisms (individuals) in swarms prompted the design of very efficient optimization and clustering algorithms • SI is an innovative distributed intelligent paradigm optimization problems • Applications include combinatorial optimization, function approximation, clustering, optimization of mechanical structures, and solving systems of equations

6. **Evolutionary Computing**

• EC models natural evolution, where the main concept is the survival of the fittest • In natural evolution, survival is achieved through reproduction. Offspring, reproduced from two parents, contain genetic material of both parents, hopefully the best characteristics of each parent • Those individuals that inherit the bad characteristics are weak and lose the battle to survive • In some bird species, a hatchling manages to get more food, gets stronger and kicks out all its siblings from the nest to die

Applications of EC • Optimization: Gas pipeline transmission, multiple-fault diagnosis, robot track determination, schedule optimization, load distribution by an electric utility • Classification: Evolution of neural networks, rule-based machine learning systems for pipeline operations and classifier systems for high-level semantic networks

7. **Fuzzy Logic**

• Set theory requires elements to be either part of a set or not. Binary-valued logic requires the values of parameters to be either 0 or 1, with similar constraints on the outcome of an inferencing process • Fuzzy sets and FL allow what is referred to as approximate reasoning • With fuzzy sets, an element belongs to a set to a certain degree of certainty • FL allows reasoning with these uncertain facts to infer new facts, with a degree of certainty associated with each fact • In a sense, FL allows the modeling of commonsense Faculty of Engineering and Technology

Applications of FL • Control Systems: subway systems, cement kilns, traffic signal systems, home appliances, video cameras, and various subsystems of automobiles including the transmission and brake systems • A familiar application is the circuitry inside a video camera that stabilizes the image in spite of the unsteady holding of the camera • Expert systems: medical diagnostics, foreign exchange trading, robot navigation, scheduling, automobile diagnostics, and the selection of business strategies

8. **Artificial Immune System (AIS)**

• AISs are biologically inspired models for immunization of engineering systems • The pioneering task of AIS is to detect and eliminate non-self-materials, called antigens, such as bacteria or cancer cells • The AIS also plays a great role to maintain its own system against dynamically changing environment • The immune systems thus aim at providing a new methodology suitable for dynamic problems dealing with unknown or hostile environments • Areas of applications: Classification, clustering, data mining and anomaly detection

9. **Fitness function**

• A mathematical function is used to quantify how good the solution represented by a chromosome is

• The fitness function f maps a chromosome representation into a scalar value, f : Γ nx → R, where Γ represents the data type of the elements of an nx -dimensional chromosome

• The fitness function represents the objective function, Φ, which describes the optimization problem

• The chromosome representation may not correspond to the representation expected by the objective function. In such cases the fitness function representation is f : SC Φ−→ SX Ψ −→ R Υ −→ R+

• SC : Search space Φ: Chromosome decoding function Ψ: Objective function Υ: Scaling function

10. Cross-over, mutation, chromosomes, genes, particles

Crossover is a genetic operator used to vary the programming of a chromosome or chromosomes from one generation to the next. Crossover is sexual reproduction. Two strings are picked from the mating pool at random to crossover in order to produce superior offspring. The method chosen depends on the Encoding Method.

Different types of crossover :

Single Point Crossover : A crossover point on the parent organism string is selected. All data beyond that point in the organism string is swapped between the two parent organisms. Strings are characterized by Positional Bias.

Two-Point Crossover : This is a specific case of a N-point Crossover technique. Two random points are chosen on the individual chromosomes (strings) and the genetic material is exchanged at these points.

Uniform Crossover : Each gene (bit) is selected randomly from one of the corresponding genes of the parent chromosomes.
Use tossing of a coin as an example technique.

Mutation

In simple terms, mutation may be defined as a small random tweak in the chromosome, to get a new solution. It is used to maintain and introduce diversity in the genetic population and is usually applied with a low probability – pm. If the probability is very high, the GA gets reduced to a random search.

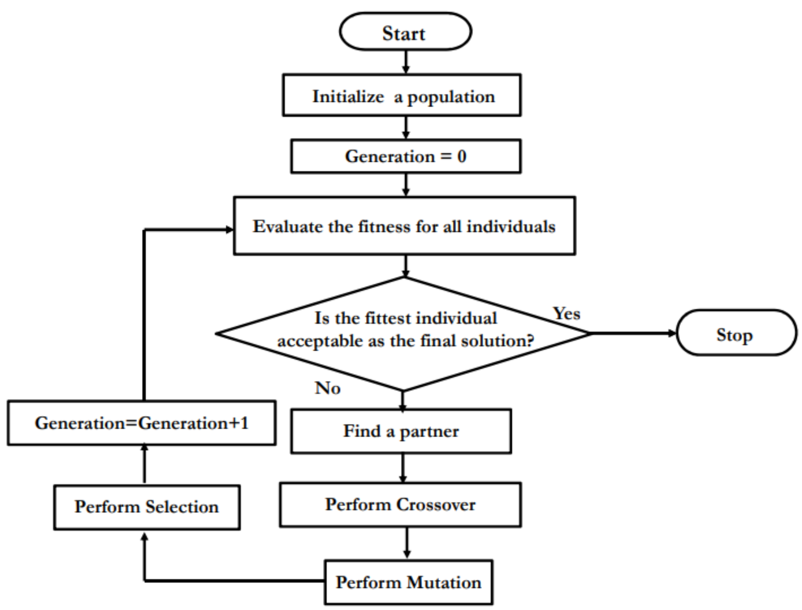
Mutation is the part of the GA which is related to the "exploration" of the search space. It has been observed that mutation is essential to the convergence of the GA while crossover is not.

Chromosome: A chromosome is one such solution to the given problem

Gene: A **gene** is one element position of a chromosome.

Explain pseudocode/flowchart, steps and details of the following algorithm

1. **Genetic Algorithm (GA)**



2. Particle swarm optimization (PSO)

STEP 1. Initialization

Initialize Parameters

Initialize Population

- Initialize Position ($\boldsymbol{x_i}$) Randomly for each Particle
- Initialize Velocity ($\boldsymbol{v_i}$) Randomly for each Particle

STEP 2. Evaluate Fitness f(x_i^t)

Calculate Fitness Value for Each Particle

If **Fitness Value is better than Best Fitness value (** ***gBest*** **)**

Than

Set New value as new (*gBest*)

Choose Particle with Best Fitness Value as *gBest*

STEP 3. For Each Particle calculate Velocity and Position.

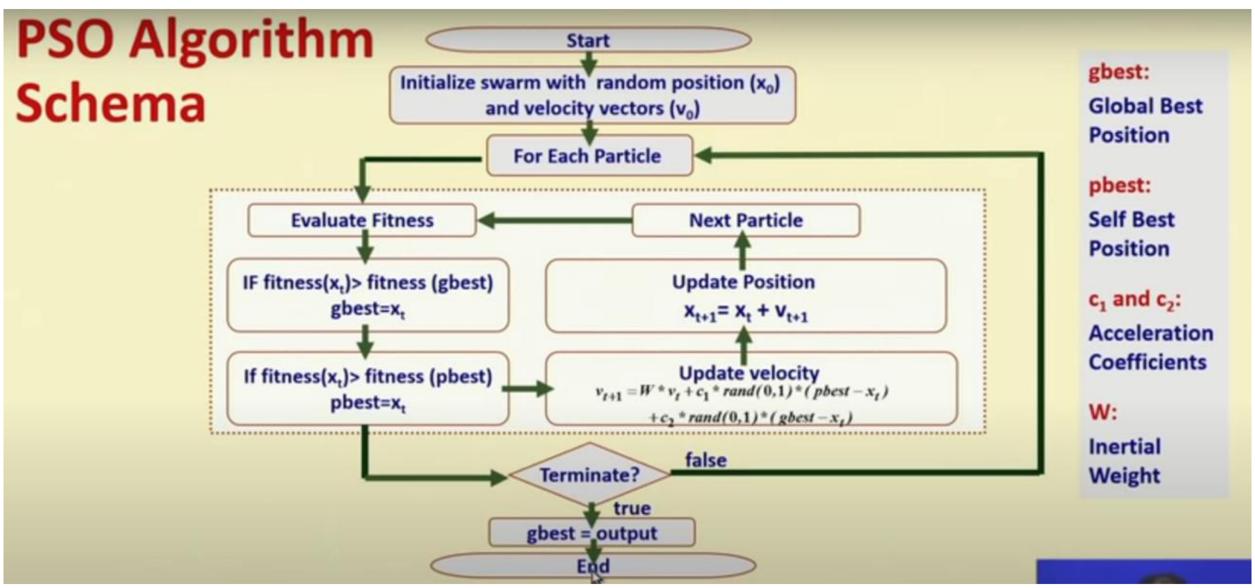
- **Calculate Particles Position by :** $x_i^{t+1} = x_i^t + v_i^t * t$
- **Calculate Velocity by:** $v_{k+1}^i = wv_k^i + c_1r_1(xBest_i^t - x_i^t) + c_2r_2(gBest_i^t - x_i^t)$

STEP 4. Evaluate Fitness f(x_i^t)

Find Current Best [*gBest*]

STEP 5. Update t=t+1

STEP 6. Output *gBest* & x_i^t



3. Artificial Bee Colony (ABC)

Algorithm 7B.1. ABC Algorithm

Initialize Population
while stopping criteria is met **do**
    Place employed bees to their food positions
    Place the onlooker bees on the food sources depending on their nectar amounts
    Send the scouts to the search area for discovering new food sources
    Memorize the best food source found so far
end while

4. Ant Colony Optimization (ACO)

Algorithm 17.2 Simple ACO Algorithm

Initialize $\tau_{ij}(0)$ to small random values;
Let $t = 0$;
Place n_k ants on the origin node;
repeat
    for *each ant* $k = 1, \ldots, n_k$ **do**
        //Construct a path $x^k(t)$;
        $x^k(t) = \emptyset$;
        repeat
            Select next node based on the probability defined in equation (17.2);
            Add link (i, j) to path $x^k(t)$;
        until *destination node has been reached*;
        Remove all loops from $x^k(t)$;
        Calculate the path length $f(x^k(t))$;
    end
    for *each link* (i, j) *of the graph* **do**
        //pheromone evaporation;
        Reduce the pheromone, $\tau_{ij}(t)$, using equation (17.5);
    end
    for *each ant* $k = 1, \ldots, n_k$ **do**
        for *each link* (i, j) *of* $x^k(t)$ **do**
            $\Delta\tau^k = \frac{1}{f(x^k(t))}$;
            Update τ_{ij} using equation (17.4);
        end
    end
    $t = t + 1$;
until *stopping condition is true*;
Return the path $x^k(t)$ with smallest $f(x^k(t))$ as the solution;

5. Bacterial Foraging Optimization (BFO)

Bacteria Foraging Optimization Algorithm Pesudocode

STEP 01: Parameter Initialization.

n: dimension of the search space,

S: the number of bacterium,

N_c: chemotactic steps,

N_s: swim steps,

N_{re}: reproductive steps,

N_{ed}: elimination and dispersal steps,

P_{ed}: probability of elimination,

$C(i)$: step size

Step 2: Elimination-dispersal loop: l = l+1

Step 3: Reproduction loop: k = k+1

Step 4: Chemotaxis loop: j = j+1

- Take a chemotactic step (i = 1,2,3...S)
- Compute Fitness Function and save this value until we may found better via run.
- Tumble Movement: in random direction [-1, 1].
- Swimming Process.
- Go to next bacterium (i+1): if **i != S** process the next bacteria.

Step 5. Ceeck If j < Chemotactic Steps, continue chemotaxis since the life of the bacteria is not over.

Step 6. Reproduction Process.

Step 7: Elimination and Dispersal.

Step 8: Check If Elimination-dispersal loop < elimination and dispersal steps, then go to Step 2, otherwise, STOP and Display Result.

6. Genetic Programming (GP)

* GENETIC PROGRAMMING:

- extension of genetic algorithms

main idea- represent a computer program as tree

—used when exact solution is not known in advance

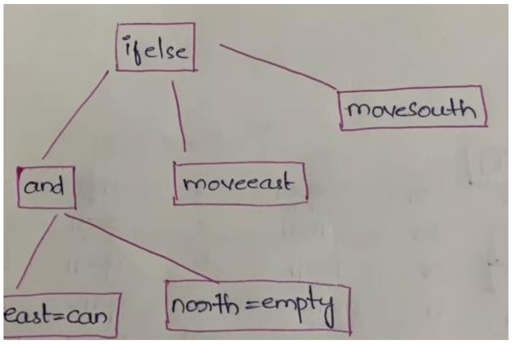
(all operations are same as genetic alg)

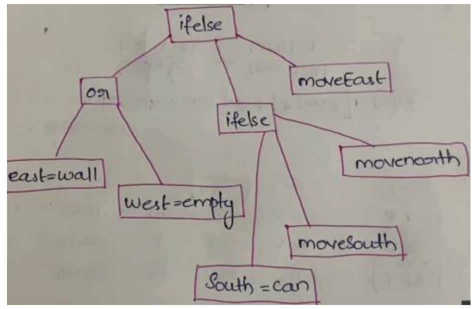
Example: plastic can collecting robot navigation

if (east = can & north = empty)

then moveeast

else movesouth

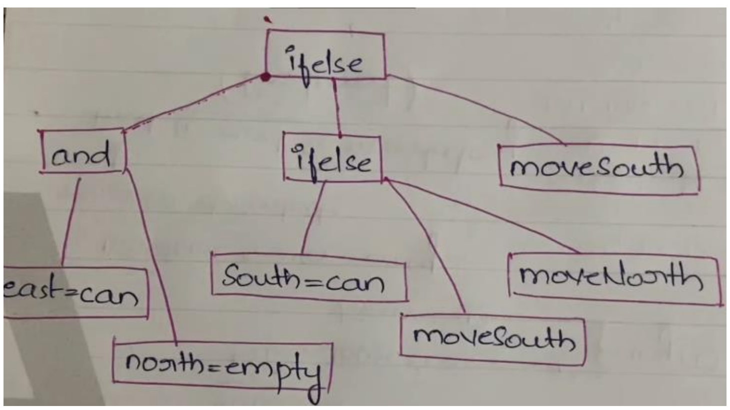




TREE 1

TREE 2

AFTER CROSSOVER WE GET:



7. Differential Equation (DE)

8. Clonal Selection Algorithm

The clonal selection principle describes the basic features of an immune response to an antigenic stimulus. It establishes the idea that only those cells that recognize the antigen proliferate, thus being selected against those that do not. The main features of the clonal selection theory are that:

➢ The new cells are copies of their parents (clone) subjected to a mutation mechanism with high rates (somatic hypermutation);

➢ Elimination of newly differentiated lymphocytes carrying self-reactive receptors;

➢ Proliferation and differentiation on contact of mature cells with antigens.

When an antibody strongly matches an antigen the corresponding B-cell is stimulated to produce clones of itself that then produce more antibodies.

This (hyper) mutation, is quite rapid, often as much as “one mutation per cell division”. This allows a very quick response to the antigens. It should be noted here that in the Artificial Immune Systems literature, often no distinction is made between B-cells and the antibodies they produce. Both are subsumed under the word ‘antibody’ and statements such as mutation of antibodies (rather than mutation of B-cells) are common. There are many more features of the immune system, including adaptation, immunological memory and protection against auto-immune attacks.

9. Danger Theory

Over the last decade, a new theory has become popular amongst immunologists. It is called the Danger Theory, and its chief advocate is Matzinger.

The theory is not simply a question of matching in the humoral immune system. It is fundamental that only the ‘correct’ cells are matched as otherwise this could lead to a self-destructive autoimmune reaction. Classical immunology stipulates that an immune response is triggered when the body encounters something non-self or foreign. In particular it is thought that the maturation process plays an important role to achieve self-tolerance by eliminating those T and B-cells that react to self. In addition, a ‘confirmation’ signal is required; that is, for either B-cell or T (killer) cell activation, a T (helper) lymphocyte must also be activated. This dual activation is further protection against the chance of accidentally reacting to self.

Matzinger’s Danger Theory points out that there must be discrimination happening that goes beyond the self-non-self-distinction described above. For instance:

➢ There is no immune reaction to foreign bacteria in the gut or to the food we eat although both are foreign entities.

➢ Conversely, some auto-reactive processes are useful, for example against selfmolecules expressed by stressed cells.

➢ The definition of self is problematic – realistically, self is confined to the subset actually seen by the lymphocytes during maturation.

➢ The human body changes over its lifetime and thus self-changes as well. Therefore, the question arises whether defenses against non-self-learned early in life might be autoreactive later.

Matzinger concludes that the immune system actually discriminates “some self from some non-self”. Danger Theory introduces not just new labels, but a way of escaping the semantic difficulties with self and non-self, and thus provides grounding for the immune response. The central idea in the Danger Theory is that the immune system does not respond to non-self but to danger. Thus, just like the self-non-self-theories, it fundamentally supports the need for discrimination. Instead of responding to foreignness, the immune system reacts to danger. This theory is borne out of the observation that there is no need to attack everything that is foreign, something that seems to be supported by the counter examples above. In this theory, danger is measured by damage to cells indicated by distress signals that are sent out when cells die an unnatural death (cell stress or lytic cell death, as opposed to programmed cell death, or apoptosis).

Figure 2 depicts how we might picture an immune response according to the Danger Theory. A cell that is in distress sends out an alarm signal, where upon antigens in the neighbourhood are captured by antigen-presenting cells such as macrophages, which then travel to the local lymph node and present the antigens to lymphocytes. Essentially, the danger signal establishes a danger zone around itself. Thus B-cells producing antibodies that match antigens within the danger zone get stimulated and undergo the clonal expansion process. Those that do not match or are too far away do not get stimulated.

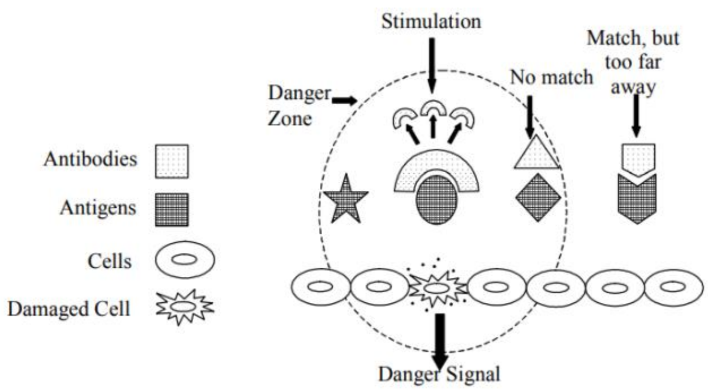
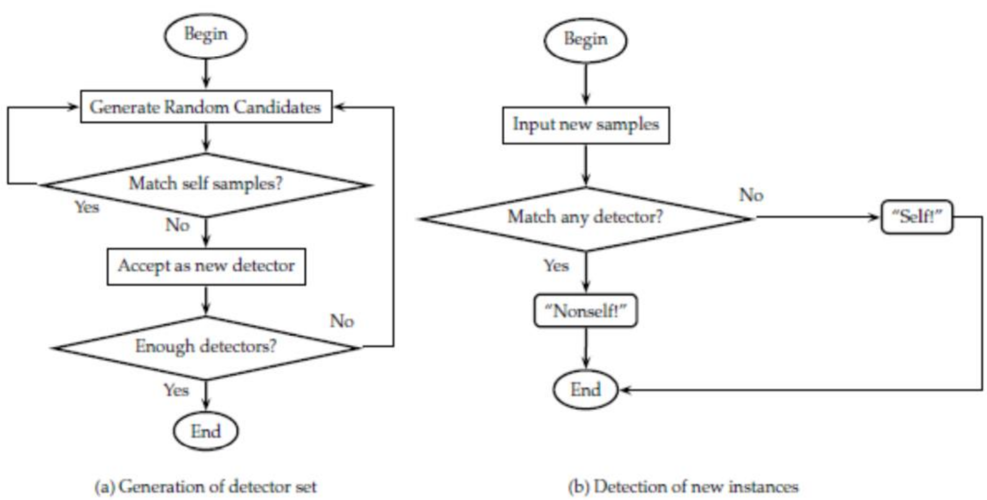


Figure 2: Danger Theory Illustration

10. Negative Selection Algorithm



11. **Pseudocode in AIS**

Algorithm 10.1. Pseudocode of AIS algorithm

Initialize population C;
Determine antigen pattern as a training set D_T ;
while some stopping condition not true **do**
  for each antigen pattern $z_p \in D_T$ **do**
    Select a subset of ALCs for exposure to z_p
    for each ALC, $x_i \in S$ **do**
      Calculate antigen affinity between z_p and x_i;
    end for
    Select a subset of ALCs with the highest calculated antigen affinity as population $H \subseteq S$;
    Adapt the ALCs in H with some selection method, based on the calculated antigen affinity and or the network affinity among ALCs in H;
    Update the stimulation level of each ALC in H;
  end for
end while

Define following terms with mathematical formulas

1. **Particle position in PSO**

Particle Swarm Optimization (PSO)

- Initial position and velocity of particles are generated randomly within the search space.
- Particle velocity (v) is determined as

$$v_i = wv_i + c_1 r_1 \left(p_{best,i} - X_i\right) + c_2 r_2 \left(g_{best} - X_i\right)$$

- Position of a particle is modified as

$$X_i = X_i + v_i$$

v_i	Velocity of the i^{th} particle
w	Inertia of the particles
c_1 and c_2	Acceleration coefficients
r_1 and r_2	Random numbers $\in [0,1]$ of size $(1xD)$
$p_{best,i}$	Personal best of i^{th} particle
g_{best}	Global best of i^{th} particle
X_i	Position of i^{th} particle

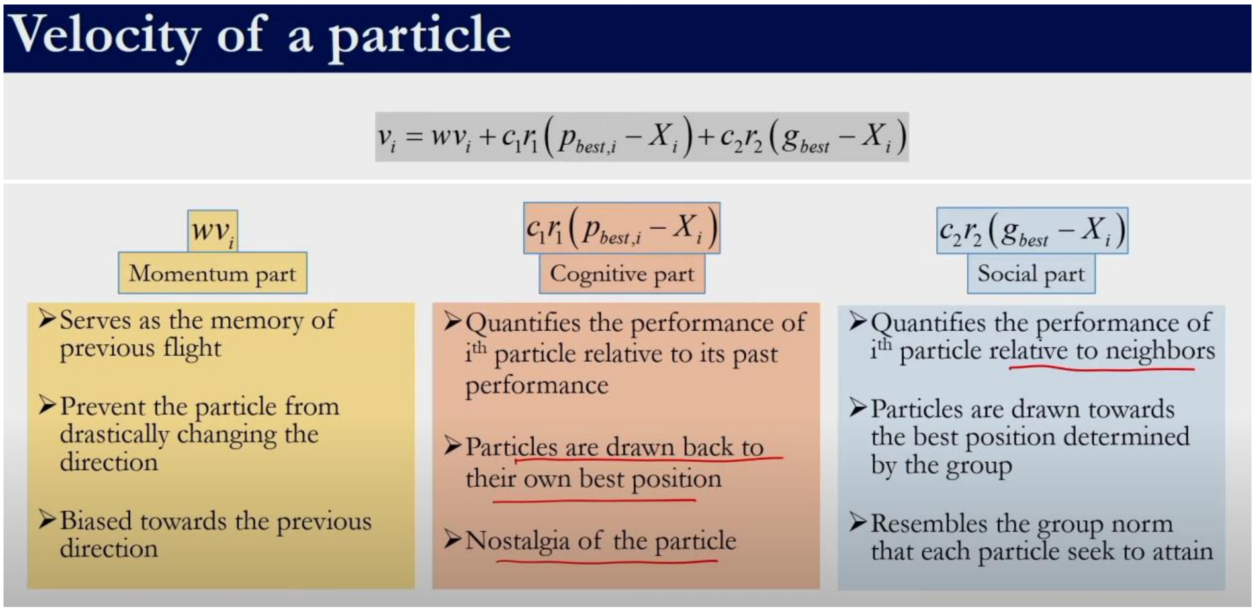
- Evaluate the objective function f_i and update the population, irrespective of the fitness

- Update p_{best} and g_{best} if

$$\left.\begin{aligned} p_{best,i} &= X_i \\ f_{p_{best,i}} &= f_i \end{aligned}\right\} if\ f_i < f_{p_{best,i}} \qquad \left.\begin{aligned} g_{best} &= p_{best,i} \\ f_{g_{best}} &= f_{p_{best,i}} \end{aligned}\right\} if\ f_{p_{best,i}} < f_{g_{best}}$$

.

2. **Velocity equations in PSO**



3. **Fitness calculation based on given problem**

4. **Employed bee, onlookers and scout bees in ABC**

Initialization Step in ABC

1. The first step is initialization where values as maximum population S, dimension D, maximum cycles k_{max} are initialized. Food source is a D dimensional vector represented as $x_{1D}, x_{2D} \ldots x_{SD}$. B is a constant value representing a maximum limit to search for food positions
2. Bees evaluate the given objective function f with initial random food positions to determine the fitness f for each x_{iD} where $i = 1, 2, 3 \ldots, S$
3. Each employed bee explores its neighboring food sources and apply a greedy selection strategy between its food source and food sources of its neighbors. If the f of the new position is higher then employed bee updates its x_{iD} otherwise it remains unchanged

Second Step: Onlooker Phase

1. A probability value is p_i is associated with each onlooker bee
2. Onlooker bee chooses a food source with the probability which is proportional to its quality
3. Different schemes can be used to calculate probability values as roulette wheel selection method or the expression given as $p_i = \frac{f_i}{\sum_{i=1}^{S} fi}$ where f_i is a fitness value for population S
4. A new candidate position from the existing memory is generated using $v_{ij} = x_{ij} + \phi_{ij} \cdot (x_{ij} - x_{oj})$, where $i, o \in 1, 2, \ldots, N$ and $j = 1, 2, \ldots, D$. o is randomly chosen such that $o \neq i$ and $-1 \leq \phi_i \leq 1$.

Third Step: Scout Phase

1. A control parameter B is used to abandon the food source
2. The position of food sources cannot be improved when a count of predetermined trials (T) exceeds B. This is where scout bees are generated
3. Scout bees discover new food position and randomly replace existing food position as given by: $x_{ij} = x^j_{\min} + q(x^j_{\max} - x^j_{\min})$, where q is a random number in $[0, 1]$

5. Chemotaxis and Reproduction in BFOA

Chemotaxis and Reproduction in BFOA

- The bacterium i at position P_i takes a chemotactic step j with the step size $C(i)$ and evaluates itself for objective function $J(P_i)$ at each step
- If at position $P_i(j+1)$, the value J is better than at position $P_i(j)$, then another step of same size $C(i)$ in the same direction will be taken again, if that step resulted in a position with a better value than at the previous step
- This swimming is continued until a minimum fitness value is reached, but only for a maximum number of steps N_s
- After N_c chemotactic steps, a reproduction step is taken, in which the population is sorted in ascending order of the objective function value J and the least healthy bacteria are replaced by the copies of the healthier bacteria

6. **Terms and equations in each algorithm**

Explain in brief

1. Global Optimization

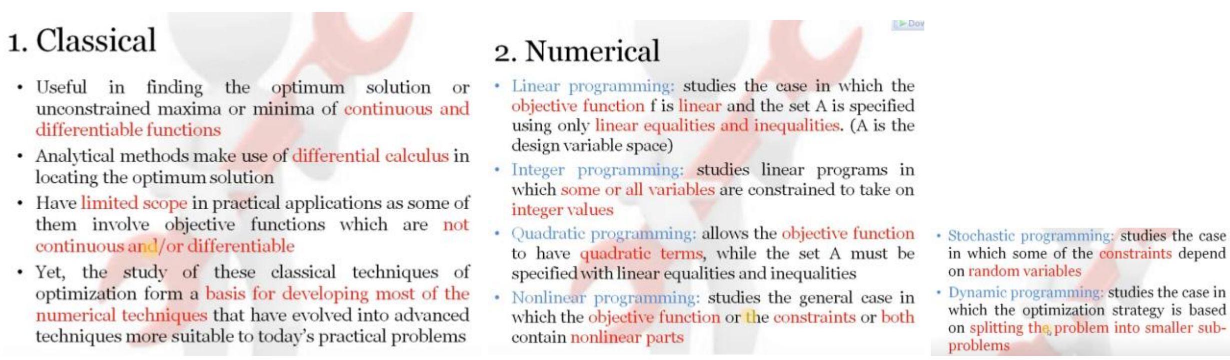
Global search algorithms may involve managing a single or a population of candidate solutions from which new candidate solutions are iteratively generated and evaluated to see if they result in an improvement and taken as the new working state.

There may be debate over what exactly constitutes a global search algorithm; nevertheless, three examples of global search algorithms using our definitions include:

- Genetic Algorithm
- Simulated Annealing
- Particle Swarm Optimization

A global optimization algorithm should be used when you know very little about the structure of the objective function response surface, or when you know that the function contains local optima

2. Types of optimization



Swarm Intelligence Based Algorithms

- Accelerated PSO
- Ant colony optimization
- Artificial Bee Colony
- Bacterial Foraging
- Bacteria-GA Foraging
- Bat Algorithm
- Bee Colony Optimization
- Bee System
- Bee Hive
- Wolf Search
- Bees Algorithm
- Bees Swarm Optimization
- Bumblebees
- Cat Swarm
- Consultant Guided search
- Cuckoo Search
- Eagle Strategy
- Fast Bacterial Swarming Algorithm
- Firefly Algorithm
- Fish Swarm
- Good Lattice Swarm Optimization
- Glowworm Swarm Optimization
- Hierarchical Swarm Algorithm
- Monkey Search
- PSO
- Virtual Ant Algorithm
- Virtual Bees
- Weightless Swarm Optimization

3 Advanced

- Swarm Intelligence Based Algorithms
- Bio-inspired (not SI-based) algorithms
- Physics and Chemistry based algorithms
- Others

Bio-inspired (not SI-based) algorithms

- Atmosphere cloud model
- Biogeography based optimization
- Brain Storm Optimization
- Differential Evolution
- Dolphin Echolocation
- Japanese Tree Frogs Calling
- Eco-inspired Evolutionary Algorithm
- Egyptian Vulture
- Flower Pollination Algorithm
- Gene Expression
- Group Search Optimizer
- Human Inspired Algorithm
- Invasive Weed Optimization
- Marriage in Honey Bees
- Paddy Field Algorithm
- Roach infestation algorithm
- Queen Bee evolution
- Shuffled frog leaping algorithm
- Termite colony optimization

Physics and Chemistry based algorithms

- Big Bang- Big Crunch
- Black Hole
- Central Force Optimization
- Charged System Search
- Electromagnetism Optimization
- Galaxy Based Search Algorithm
- Gravitational Search
- Harmony Search
- Intelligent water Drop
- River Formation Dynamics
- Self Propelled Particles
- Simulated Annealing
- Stochastic Diffusion Search
- Spiral Optimization
- Water Cycle Algorithm

3. Example of optimization

Optimization: A Day-to-Day Example 1

1. Ingredients: Water, Tea-powder, Sugar and Milk
2. Assumption: Taste of the tea depends only on ingredients
3. **How much of each ingredient do we use per cup of tea?**
4. Desired Outcome: Excellent taste!

Optimization: Day-to-Day Example 2

- Two knobs: H and C
- Knob settings θ_h and θ_c
- **What should be the right settings of θ_h and θ_c?**
- Desired Outcome: Very comfortable shower!

Example 3: Design Optimization

- An engineer wants to design a cylindrical can to hold 200 ml of a soft drink
- The can will be made from aluminum sheet
- Radius r and height h are the optimization variables
- What are the values of r^* and h^* that result in the minimum surface area?
- This is a two-dimensional problem
- The objective function to minimize is $A = 2\pi rh + 2(\pi r^2)$
- This is a relatively easy problem

Example 4: Time Management

- A student has registered for N courses C_i, $i = 1, 2, \cdots, N$
- The number of hours she spends studying course i is H_i
- She expects to get M_i marks in course i
- The objective is to earn as many total marks as possible, $\sum_{i=1}^{N} M_i$
- What are the best values of H_i^*, $i = 1, 2, \cdots, N$?
- Discrete optimization!

Example 5: HR Optimization

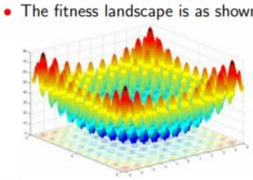
- A department seeks to appoint N faculty members from four cadres:

L	Lecturer
AP	Assistant Professor
ASP	Associate Professor
P	Professor

- The number of faculty members in these cadres is N_L, N_{AP}, N_{ASP} and N_P, respectively
- The objective is to pay as **low salary** as possible, but to get the **best possible quality**
- What are the best numbers N_L^*, N_{AP}^*, N_{ASP}^* and N_P^*?
- Multi-objective integer optimization!

Example 6: Rastrigin Function

- Let $\mathbf{x} = \{x_1, x_2, \ldots, x_N\}, x_i \in [-5.12, 5.12]$
- $f(\mathbf{x}) = 10N + \sum_{i=1}^{N} [x_i^2 - 10\cos(2\pi x_i)]$
- The objective is to find the point $\mathbf{x}^*$ such that $f(\mathbf{x}^*)$ is minimum
- The fitness landscape is as shown:



- Solution is $x_i^* = 0, i = 1, 2, \cdots, N$
- Multimodal landscape. Global optimization!

4. Discuss pros and cons of deterministic approaches to optimization

Advantages

- ✓ They are the first option to solve optimization problems
- ✓ If there is a match between the features of the problem and the conditions required by the method, the results are very competitive
- ✓ Under certain conditions, the computational cost can be low
- ✓ They produce the same results always

Disadvantages

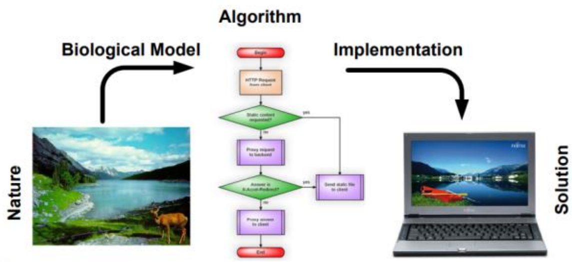
- ✗ In some problems, these techniques are either difficult to apply or they may take considerable time to reach to an acceptable solution
- ✗ These techniques suffer from the **curse-of-dimensionality**
- ✗ The application of these methods may require a transformation of the original model of the problem
- ✗ Some methods are difficult to use

5. Heuristic approaches to optimization

6. Relevance of biology to CI-based algorithms

Bio-Inspired Optimization

Bio-inspired optimization is the group of optimization algorithms whose behaviors are based on **biological phenomena**



Why Biology?

- Biology is a source of **adaptive mechanisms** where intelligent behavior emerges in changing and complex environments
- These mechanisms are capable of:
  1. Learning
  2. Generalizing
  3. Abstracting
  4. Discovering
  5. Associating
- These are studied under the title CI
- The five dominant paradigms of CI are:
  1. Artificial Immune Systems
  2. Artificial Neural Networks
  3. Evolutionary Computing
  4. Fuzzy Logic
  5. Swarm Intelligence

6. Stigmergy and Artificial Pheromone

Stigmergy and Artificial Pheromone

- Stigmergy refers to a form of indirect communication mediated by modifications of the environment. Sign-based stigmergy facilitates communication via a signaling mechanism via chemical compounds deposited by ants
- Each ant drops an amount of pheromone as it moves from food source to the nest
- Future ants choose their paths based on the amount of pheromone. Higher pheromone concentration means higher chance that the path is chosen
- Over time, shorter paths have stronger pheromone concentrations since they are chosen by ants most often
- Pheromone evaporates over time. The pheromone concentrations on the longer paths decrease more quickly than on the shorter paths

7. Fuzzy Logic: Introduction and example

Introduction

- The word "fuzzy" means "vaguness (ambiguity)".
- Fuzziness occurs when the boundary of a piece of information is not clear-cut.
- Fuzzy sets - 1965 Lotfi Zadeh as an extension of classical notation set.
- Classical set theory allows the membership of the elements in the set in **binary terms**.
- Fuzzy set theory permits membership function valued in the interval [0,1].

- In real world, there exist much fuzzy knowledge (i.e. vague, uncertain inexact etc).
- Human **thinking** and **reasoning** (analysis, logic, interpretation) frequently involved **fuzzy** information.
- Human can give satisfactory answers, which are probably true.
- Our systems are unable to answer many question because the systems are designed based upon classical set theory (Unreliable and incomplete).
- We want, our system should be able to cope with unreliable and incomplete information.
- Fuzzy system have been provide solution.

Example:

Words like young, tall, good or high are fuzzy.

- There is no single quantitative value which defines the term young.
- For some people, age 25 is young, and for others, age 35 is young.
- The concept young has no clean boundary.
- Age 35 has some possibility of being young and usually depends on the context in which it is being considered.

Fuzzy set theory is an extension of classical set theory where elements have degree of membership.

8. Membership Function: Triangular, Trapezoidal, Gaussian, Tall etc.

Fuzzy Sets (Continue)

Membership Function

- The membership function fully defines the fuzzy set
- A membership function provides a measure of *the degree of similarity* of an element to a fuzzy set

Membership functions can

- – either be chosen by the user arbitrarily, based on the user's experience (MF chosen by two users could be different depending upon their experiences, perspectives, etc.)
- – Or be designed using machine learning methods (e.g., artificial neural networks, genetic algorithms, etc.)

Fuzzy Sets (Continue)

There are different shapes of membership functions;

- **Triangular,**
- **Trapezoidal,**
- **Gaussian, etc**

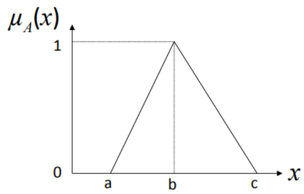
Fuzzy Sets (Continue)

- **Triangular membership function**

A *triangular* membership function is specified by three parameters {a, b, c}

a, b and c represent the *x* coordinates of the three vertices of $\mu_A(x)$ in a fuzzy set A (a: lower boundary and c: upper boundary where membership degree is zero, b: the centre where membership degree is 1)

$$\mu_A(x) = \begin{cases} 0 & if\ x \le a \\ \dfrac{x-a}{b-a} & if\ a \le x \le b \\ \dfrac{c-x}{c-b} & if\ b \le x \le c \\ 0 & if\ x \ge c \end{cases}$$



Fuzzy Sets (Continue)

- **Trapezoid membership function**
- A *trapezoidal* membership function is specified by four parameters {a, b, c, d} as follows:

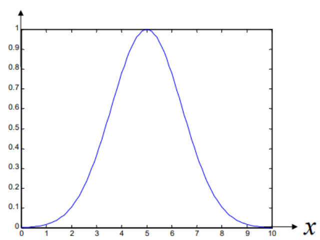
$$\mu_A(x) = \begin{cases} 0 & if\ x \le a \\ \dfrac{x-a}{b-a} & if\ a \le x \le b \\ 1 & if\ b \le x \le c \\ \dfrac{d-x}{d-c} & if\ c \le x \le d \\ 0 & if\ d \le x \end{cases}$$

- **Gaussian membership function**

$$\mu_A(x,c,s,m) = \exp\left[-\frac{1}{2}\left|\frac{x-c}{s}\right|^m\right]$$

- *c:* centre
- *s:* width
- *m:* fuzzification factor (e.g., *m*=2)

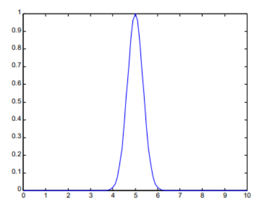
$\mu_A(x)$



c=5

s=2

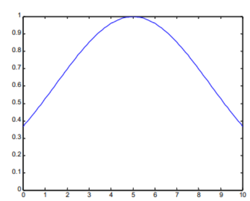
m=2



c=5

s=0.5

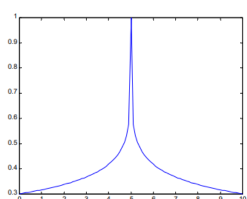
m=2



c=5

s=5

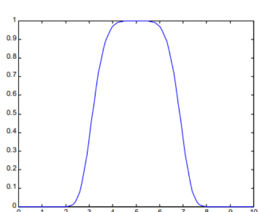
m=2



c=5

s=2

m=0.2



c=5

s=5

m=5

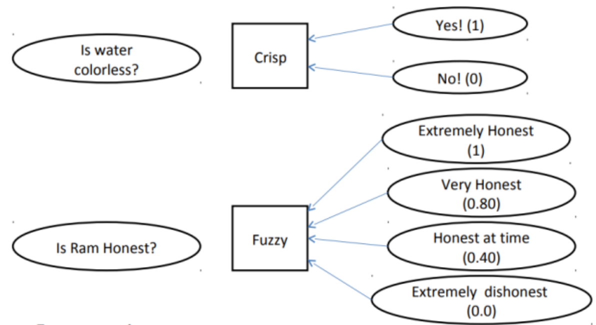
9. Differentiate between crisp logic and fuzzy logic

Introduction

Classical set theory	Fuzzy set theory
• Classes of objects with sharp boundaries.	• Classes of objects with un-sharp boundaries.
• A classical set is defined by crisp(exact) boundaries, i.e., there is no uncertainty about the location of the set boundaries.	• A fuzzy set is defined by its ambiguous boundaries, i.e., there exists uncertainty about the location of the set boundaries.
• Widely used in digital system design	• Used in fuzzy controllers.

Introduction (Continue)

Example



Fuzzy vs crips

10. Elements of fuzzy systems
11. Comparison of crisp and fuzzy logic

12. Fuzzy set operators

Fuzzy Set Operation

Given X to be the universe of discourse and Ã and Ḃ to be fuzzy sets with $\mu_A(x)$ and $\mu_B(x)$ are their respective membership function, the fuzzy set operations are as follows:

Union:

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

Intersection:

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

Complement:

$$\mu_A(x) = 1 - \mu_A(x)$$

Numerical Questions and scenario-based questions on:

1. Fuzzy sets

2. Cross-over

3. Mutation

4. PSO-position and velocity value