IIT KHARAGPUR

NPTEL ONLINE CERTIFICATION COURSES

NPTEL

Soft Computing

Fuzzy Logic Controller-I

Prof. Debasis Samanta

Department of Computer Science & Engineering

IIT Kharagpur

Fuzzy Logic Controller

- Applications of Fuzzy logic
- Fuzzy logic controller
- Modules of Fuzzy logic controller
- Approaches to Fuzzy logic controller design
    - Mamdani approach
    - Takagi and Sugeno’s approach

Debasis Samanta
CSE
IIT Kharagpur

Applications of Fuzzy Logic

Fuzzy Systems : Fuzzy Logic Controller

- Concept of fuzzy theory can be applied in many applications, such as fuzzy reasoning, fuzzy clustering, fuzzy programming, etc.

- Out of all these applications, fuzzy reasoning, also called "fuzzy logic controller (FLC)" is an important application.

- Fuzzy logic controllers are special expert systems. In general, a FLC employs a knowledge base expressed in terms of a fuzzy inference rules and a fuzzy inference engine to solve a problem.

Fuzzy Systems : Fuzzy Logic Controller

- We use FLC where an exact mathematical formulation of the problem is not possible or very difficult.

- These difficulties are due to non-linearities, time-varying nature of the process, large unpredictable environment disturbances, etc.

Fuzzy Systems : Fuzzy Logic Controller

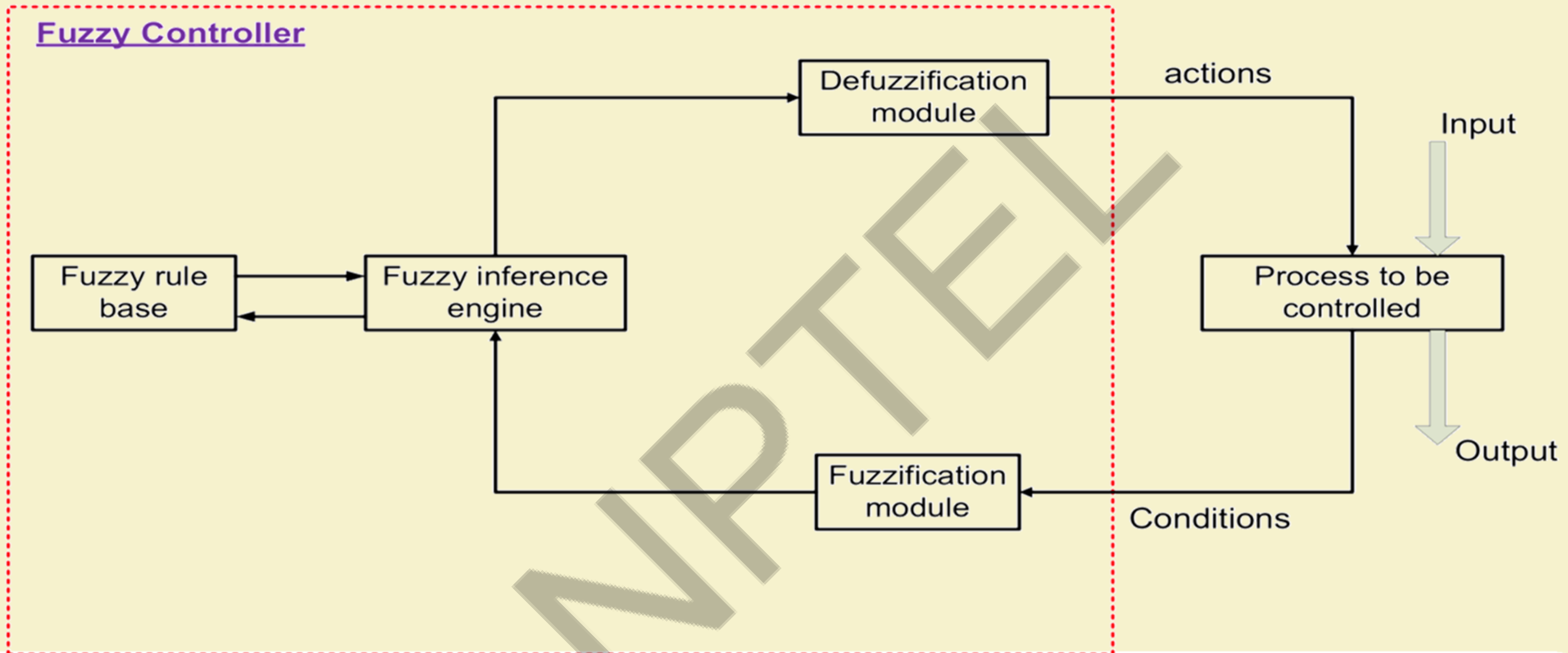


Figure 1: A general scheme of a fuzzy controller

Fuzzy Systems : Fuzzy Logic Controller

A general fuzzy controller consists of four modules:

- a fuzzy rule base,
- a fuzzy inference engine,
- a fuzzification module, and
- a defuzzification module.

Fuzzy Systems : Fuzzy Logic Controller

As shown in Figure 1, a fuzzy controller operates by repeating a cycle of the following four steps :

- **Step 1:** Measurements (inputs) are taken of all variables that represent relevant condition of controller process.
- **Step 2:** These measurements are converted into appropriate fuzzy sets to express measurements uncertainties. This step is called fuzzification.

Fuzzy Systems : Fuzzy Logic Controller

- **Step 3:** The fuzzified measurements are then used by the inference engine to evaluate the control rules stored in the fuzzy rule base. The result of this evaluation is a fuzzy set (or several fuzzy sets) defined on the universe of possible actions.

- **Step 4:** This output fuzzy set is then converted into a single (crisp) value (or a vector of values). This is the final step called defuzzification. The defuzzified values represent actions to be taken by the fuzzy controller.

Fuzzy Systems : Fuzzy Logic Controller

There are manly two approaches of FLC.

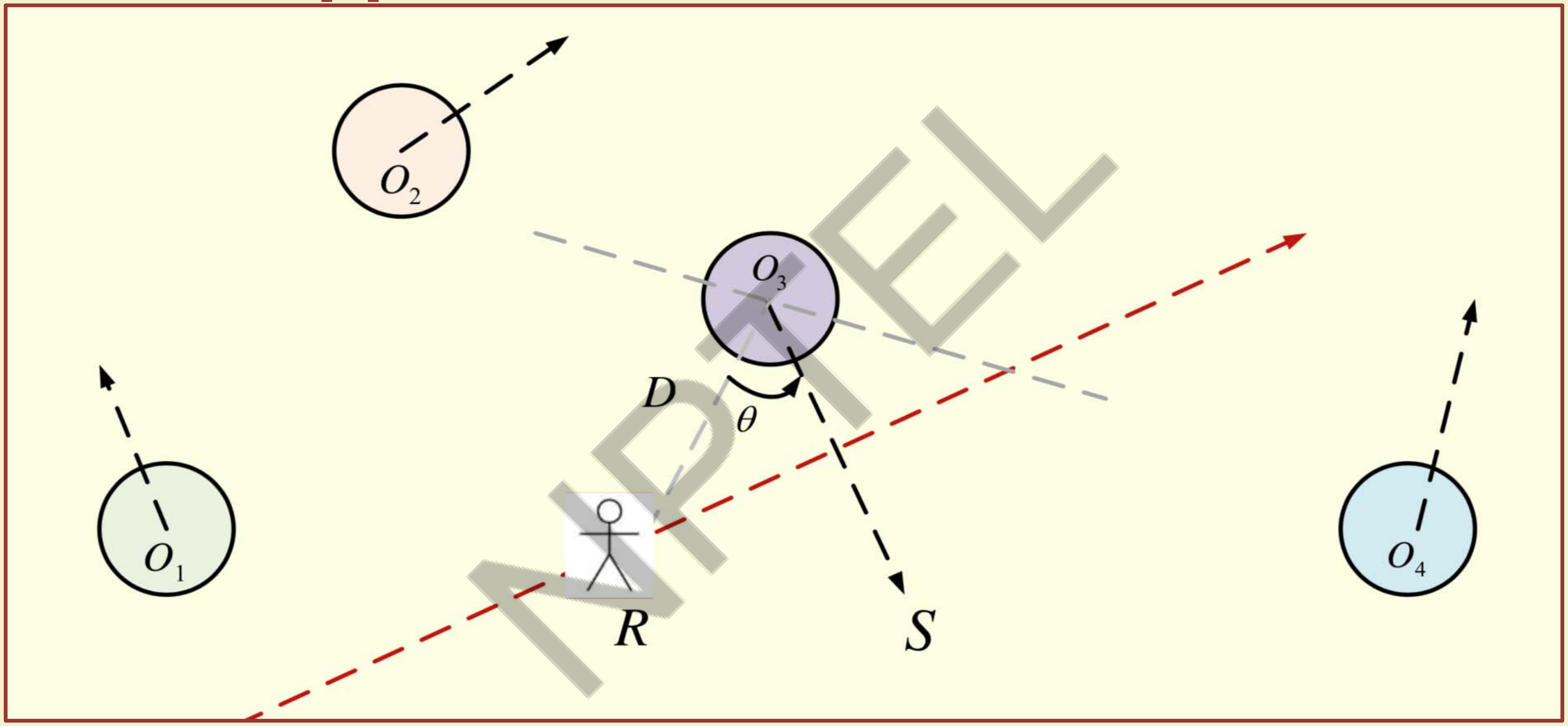
- Mamdani approach
- Takagi and sugeno's approach
  - Mamdani approach follows linguistic fuzzy modelling and characterized by its high interpretability and low accuracy.
  - On the other hand, Takagi and Sugeno's approach follows precise fuzzy modelling and obtains high accuracy but at the cost of low interpretability.

We illustrate the above two approaches with examples.

Mamdani approach : Mobile Robot

- Consider the control of navigation of a mobile robot in the presence of a number of moving objects.
- To make the problem simple, consider only four moving objects, each of equal size and moving with the same speed.
- A typical scenario is shown in Figure 2.

Mamdani approach : Mobile Robot

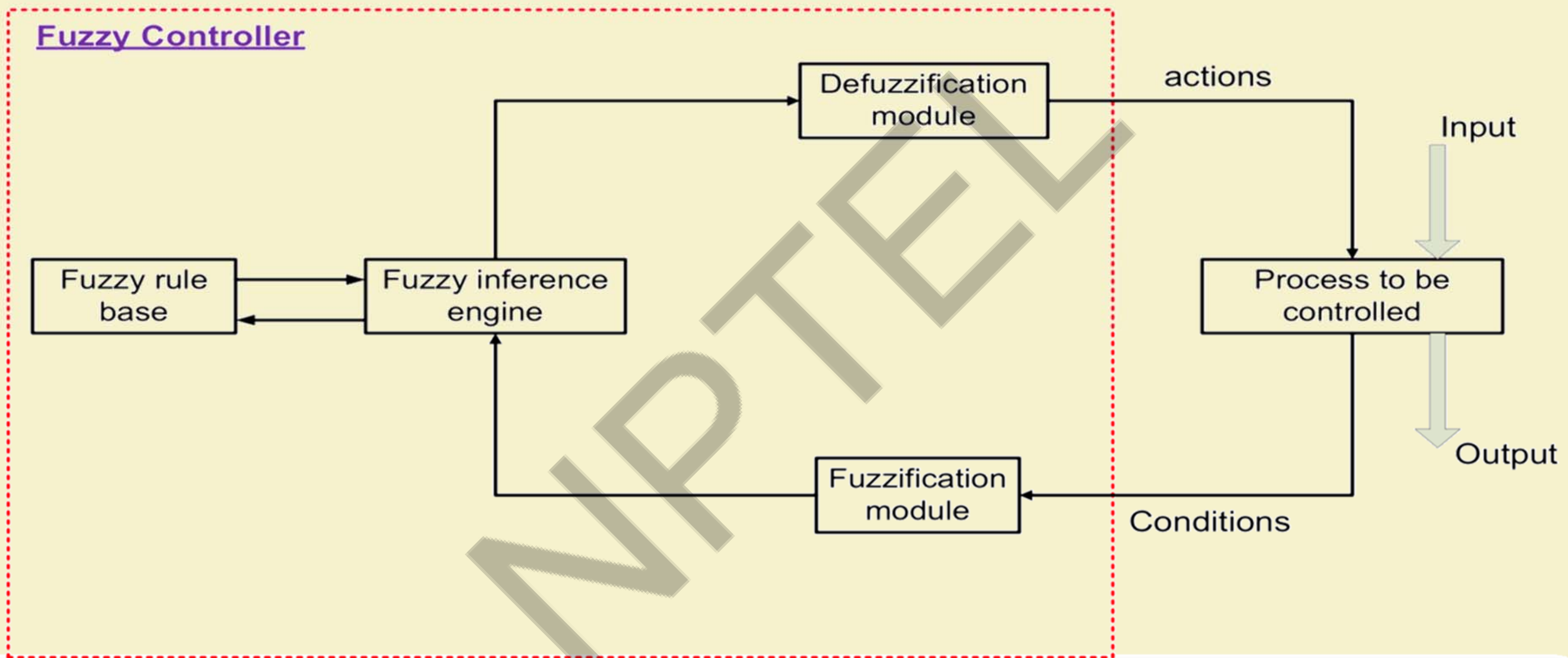


Mamdani approach : Mobile Robot

- We consider two parameters : D, the distance from the robot to an object and θ the angle of motion of an object with respect to the robot.
- The value of these parameters with respect to the most critical object will decide an output called deviation (δ).
- We assume the range of values of D is $[0.1, \dots . 2.2]$ in meter and θ is $[-90, \dots , 0, \dots \ 90]$ in degree.
- After identifying the relevant input and output variables of the controller and their range of values, the Mamdani approach is to select some meaningful states called "linguistic states" for each variable and express them by appropriate fuzzy sets.

Debasis Samanta
CSE
IIT Kharagpur

Fuzzy Logic Controller



Linguistic States

For the current example, we consider the following linguistic states for the three parameters.

Distance is represented using four linguistic states:

- VN : Very Near
- NR : Near
- VF : Very Far
- FR : Far

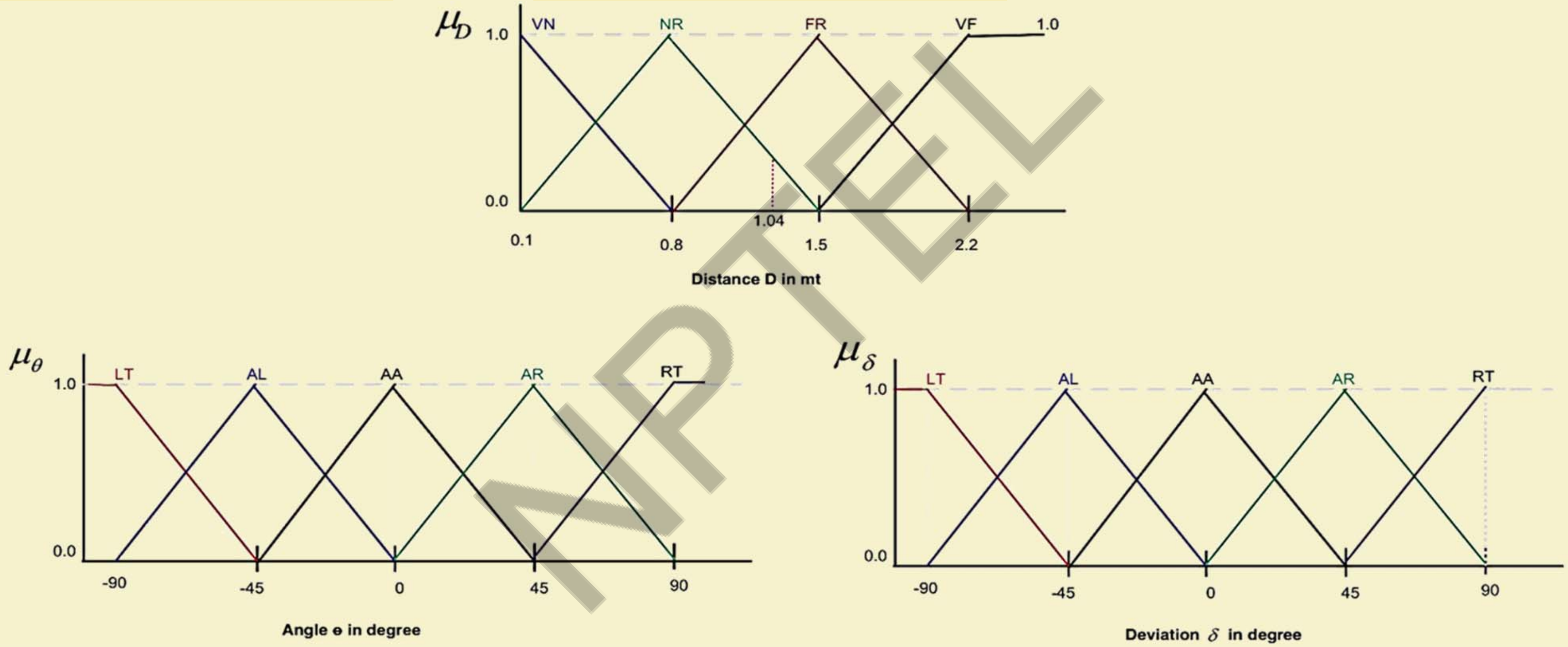
Linguistic States

Angle (for both angular direction (θ) and deviation (δ)) are represented using five linguistic states:

- LT : Left
- AL : Ahead Left
- AA : Ahead
- AR : Ahead Right
- RT : Right

Linguistic States

Three different fuzzy sets for the three different parameters are given below (Figure 3).



Fuzzy rule base

- Once the fuzzy sets of all parameters are worked out, our next step in FLC design is to decide fuzzy rule base of the FLC.

- The rule base for the FLC of mobile robot is shown in the form of a table below.

$$
\begin{array}{cc}
 & \begin{array}{ccccc} LT & AL & AA & AR & RT \end{array} \\
\begin{array}{c} VN \\ NR \\ FR \\ VF \end{array} &
\begin{bmatrix}
AA & AR & AL & AL & AA \\
AA & AA & RT & AA & AA \\
AA & AA & AR & AA & AA \\
AA & AA & AA & AA & AA
\end{bmatrix}
\end{array}
$$

Fuzzy rule base for the mobile robot

Note that this rule base defines 20 rules for all possible instances. These rules are simple rules and take in the following forms.

- **Rule 1**: If (distance is VN) and (angle is LT) Then (deviation is AA)

  .

  .

- **Rule 13**: If (distance is FR) and (angle is AA) Then (deviation is AR)

  .

  .

- **Rule 20**: If (distance is VF) and (angle is RT) Then (deviation is AA)

Fuzzy Systems : Fuzzy Logic Controller

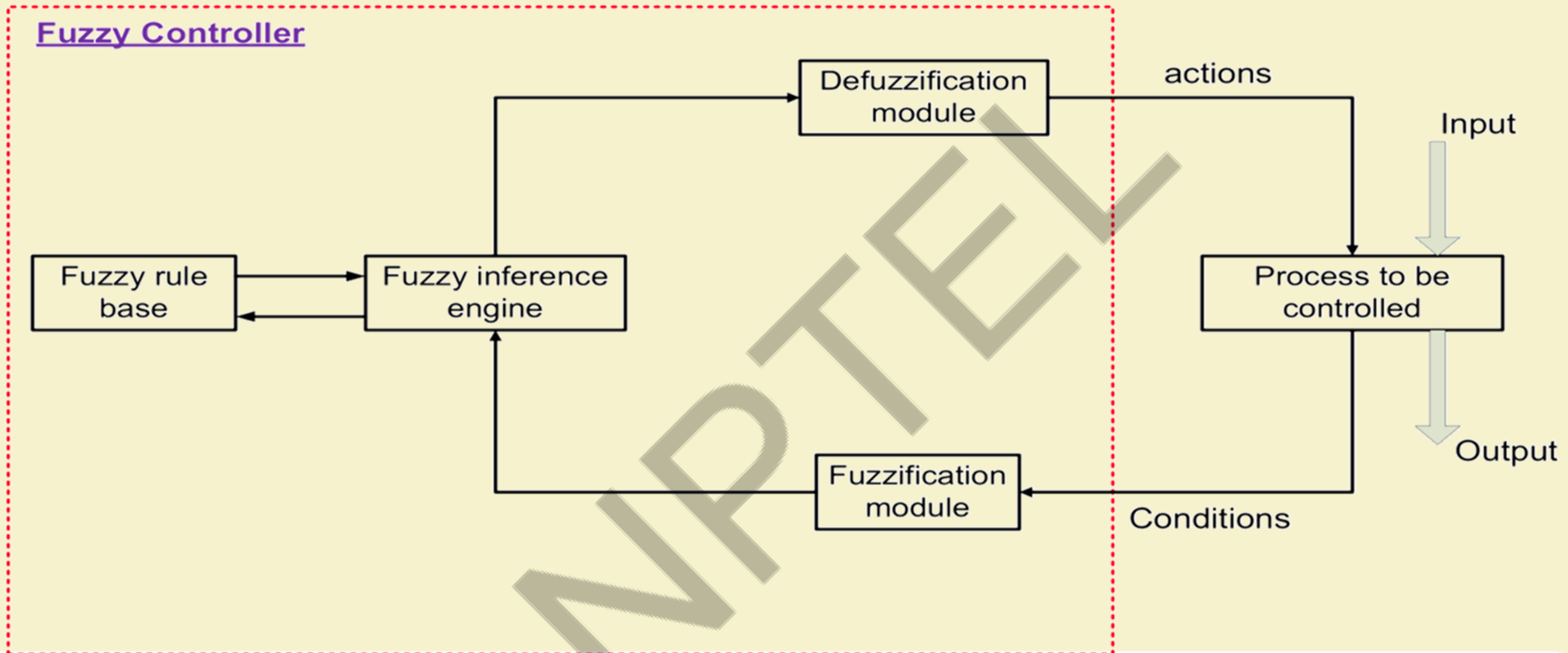
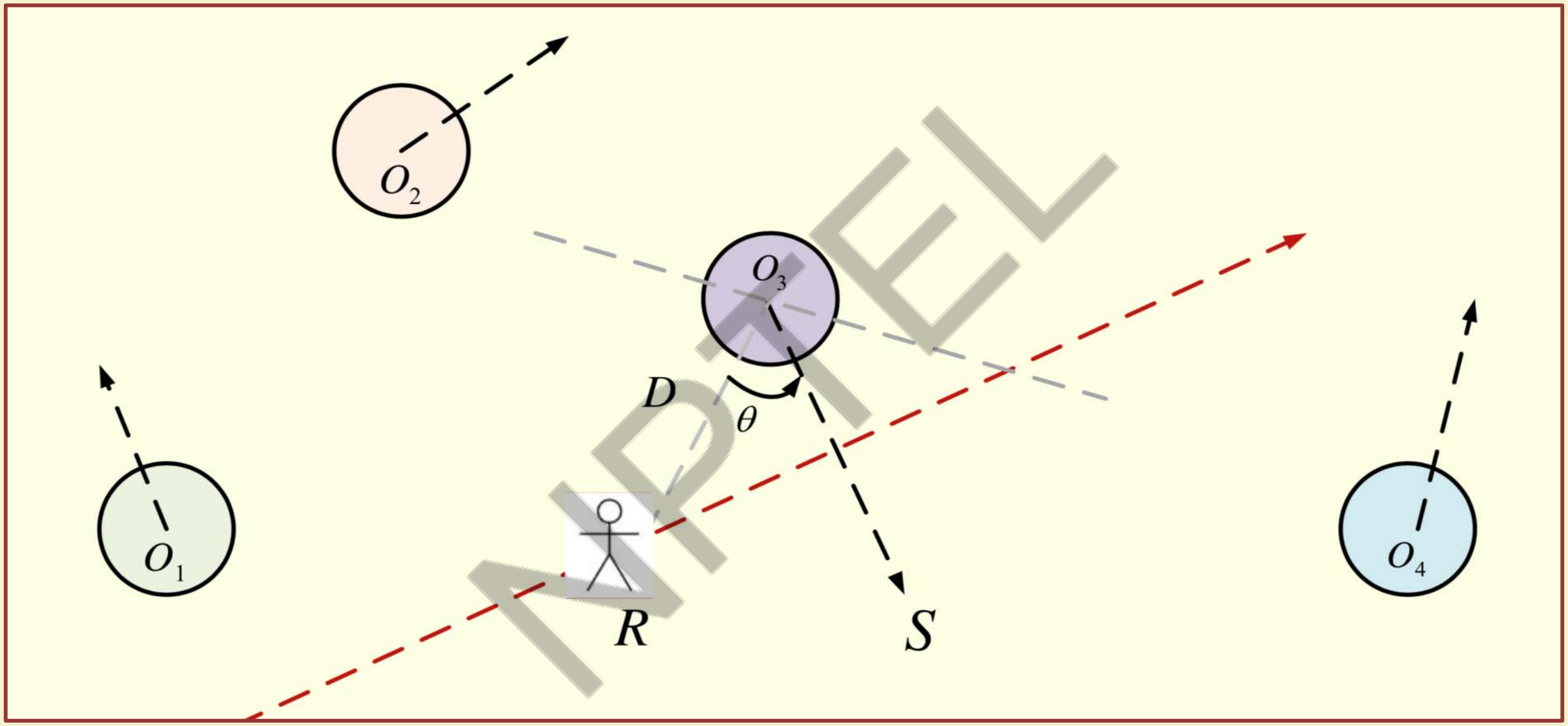


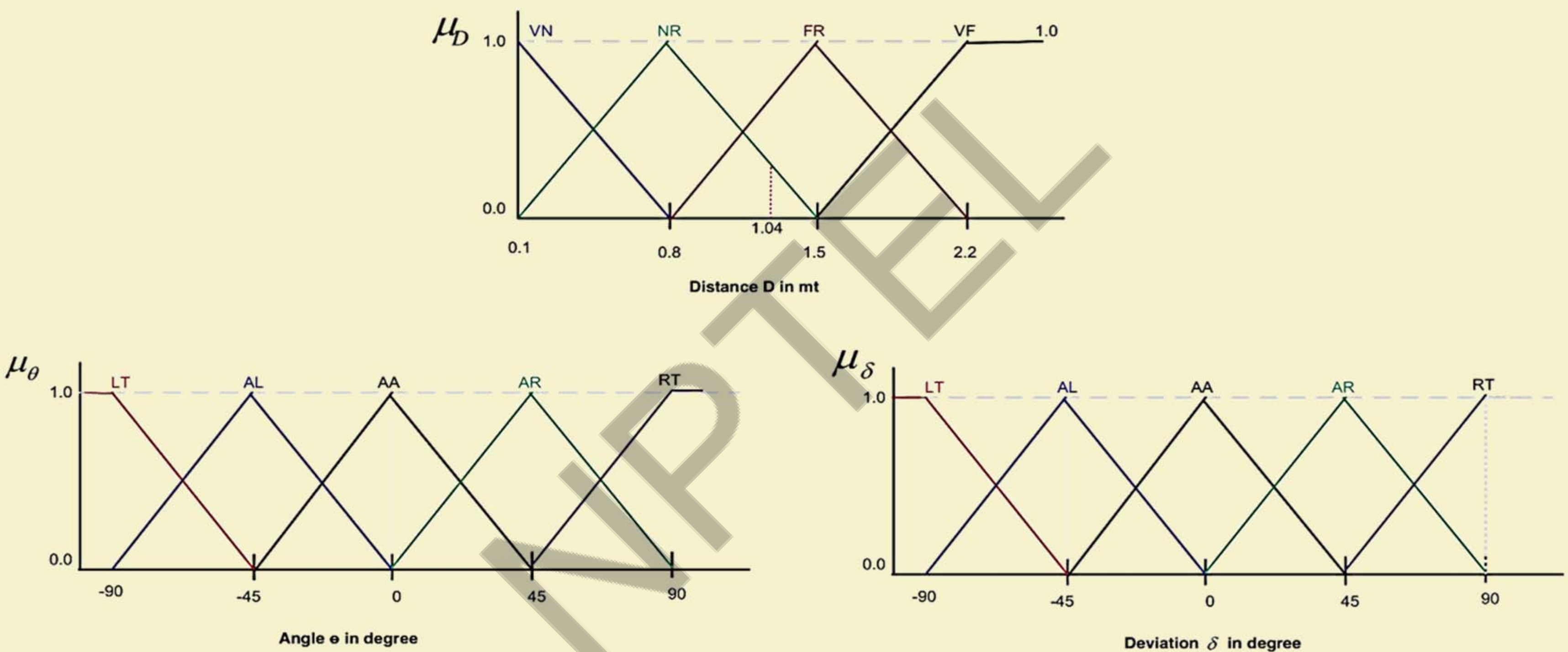
Figure 1: A general scheme of a fuzzy controller

Fuzzification of inputs

- The next step is the fuzzification of inputs. Let us consider, at any instant, the object O_3 is critical to the Mobile Robot and distance $D = 1.04\ m$ and angle $\theta = 30°$.
- For this input, we are to decide the deviation δ of the robot as output.

Mobile Robot



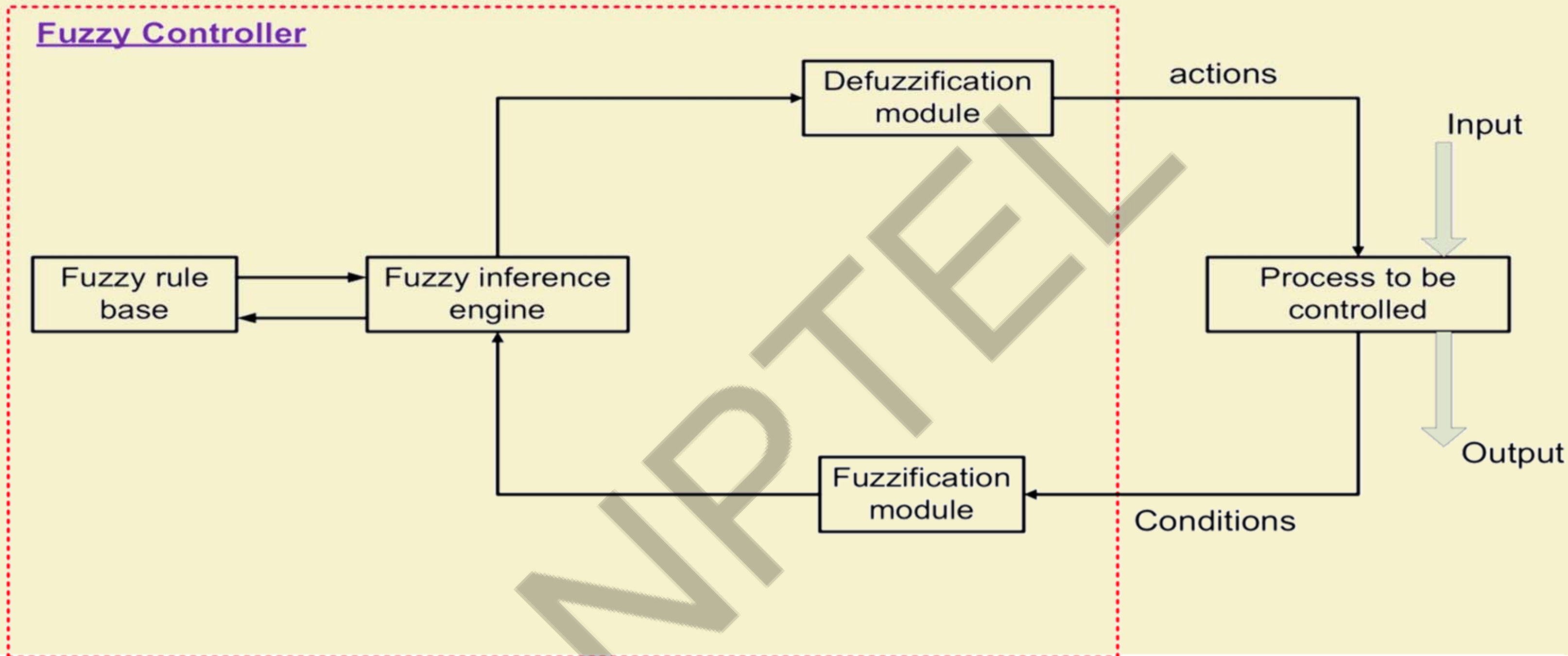


μ_D
1.0
0.0
VN
NR
FR
VF
1.0
0.1
0.8
1.04
1.5
2.2
Distance D in mt
μ_θ
1.0
0.0
LT
AL
AA
AR
RT
-90
-45
0
45
90
Angle θ in degree
μ_δ
1.0
0.0
LT
AL
AA
AR
RT
-90
-45
0
45
90
Deviation δ in degree

Fuzzification of inputs

- From the given fuzzy sets and input parameters' values, we say that the distance $D = 1.04\, m$ may be called as either NR (near) or FR (far).

- Similarly, the input angle $\theta = 30°$ can be declared as either AA (ahead) or AR (ahead right).


Fuzzy Controller
Defuzzification module
actions
Input
Fuzzy rule base
Fuzzy inference engine
Process to be controlled
Output
Fuzzification module
Conditions

Thank You!!

IIT KHARAGPUR

NPTEL ONLINE CERTIFICATION COURSES

NPTEL

Soft Computing

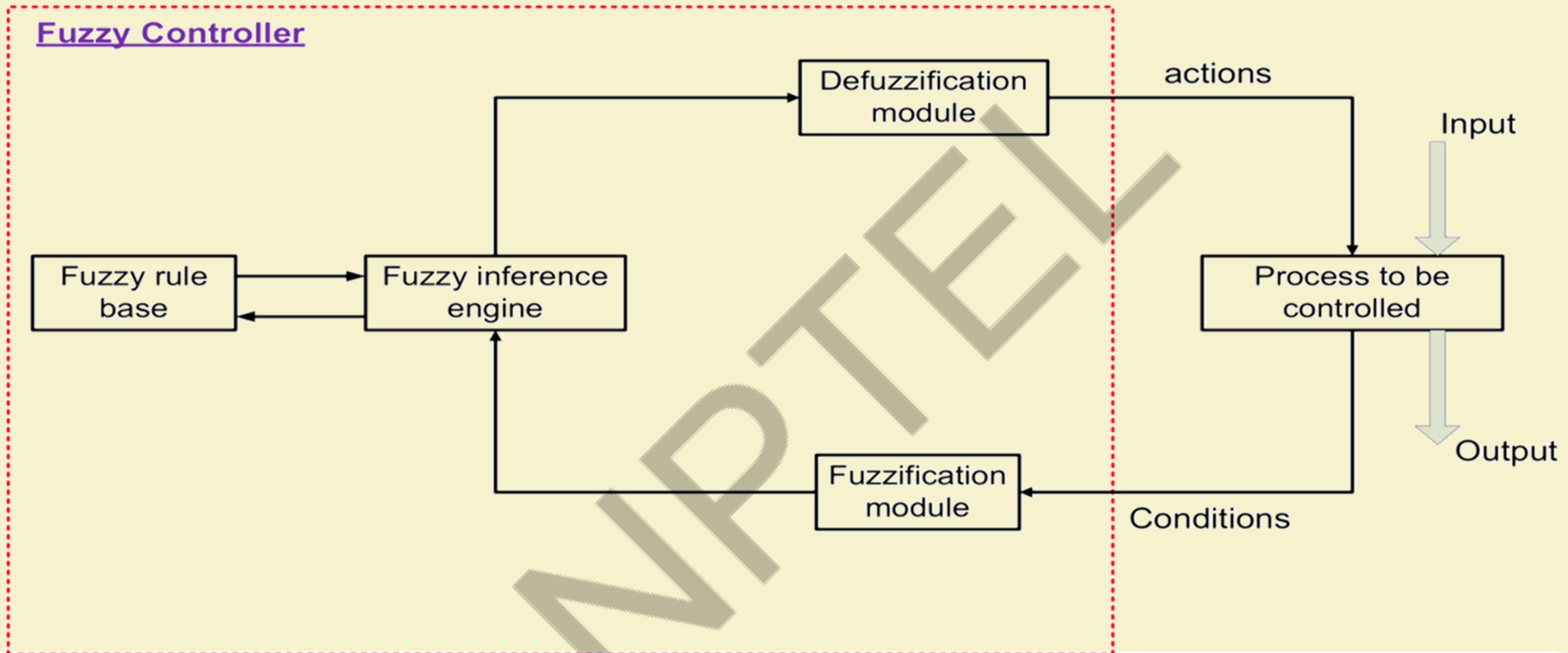
Fuzzy Logic Controller-II

Prof. Debasis Samanta

Department of Computer Science & Engineering

IIT Kharagpur

Fuzzy Systems : Fuzzy Logic Controller



A general scheme of a fuzzy controller

Debasis Samanta
CSE
IIT Kharagpur

Fuzzy Logic Controller

A FLC consists of four modules:

- a fuzzy rule base,
- a fuzzy inference engine,
- a fuzzification module, and
- a defuzzification module.

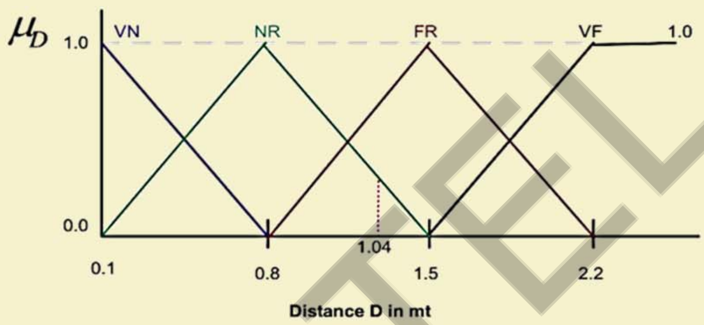
Fuzzification of inputs

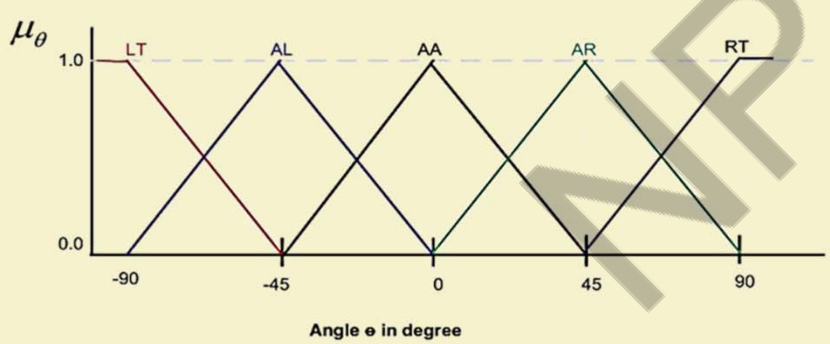
- Input

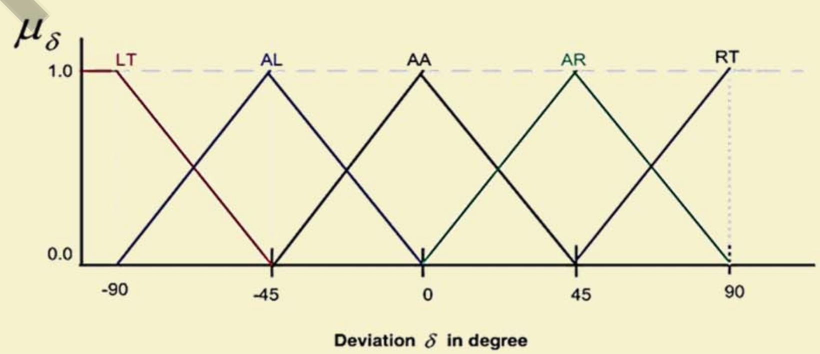
$D = 1.04\, m$

$\theta = 30°$

Linguistic States







Fuzzification of inputs

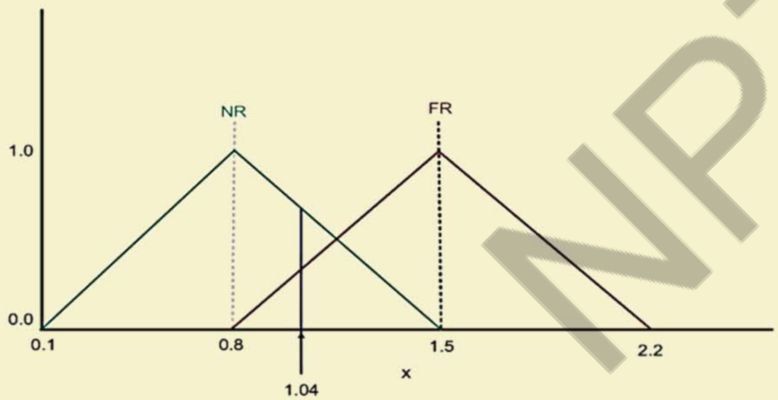
- We say that the distance $D = 1.04\ m$ may be called as either NR (near) or FR (far).
- Similarly, the input angle $\theta = 30°$ can be called as either AA (ahead) or AR (ahead right).

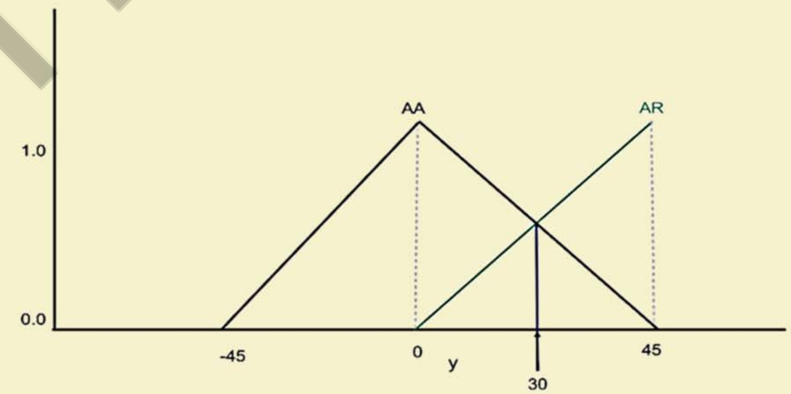
Fuzzification of inputs

Hence, we are to determine the membership values corresponding to these values, which is as follows.

$$x = 1.04\,m$$
$$\mu_{NR}(x) = 0.6571$$
$$\mu_{FR}(x) = 0.3429$$

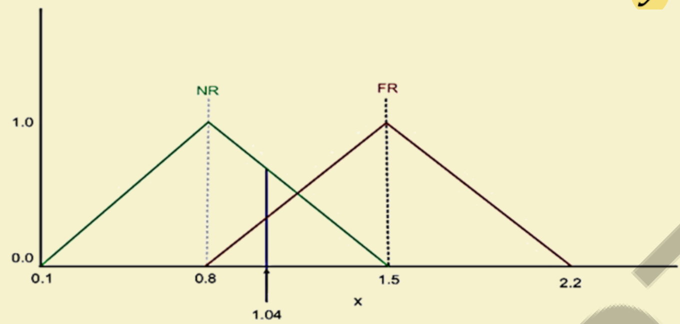
$$y = 30°$$
$$\mu_{AA}(y) = 0.3333$$
$$\mu_{AR}(y) = 0.6667$$

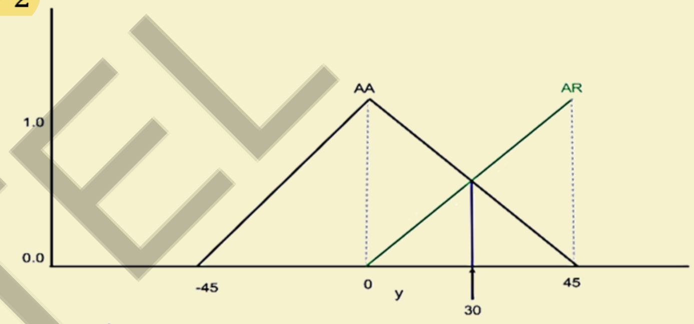




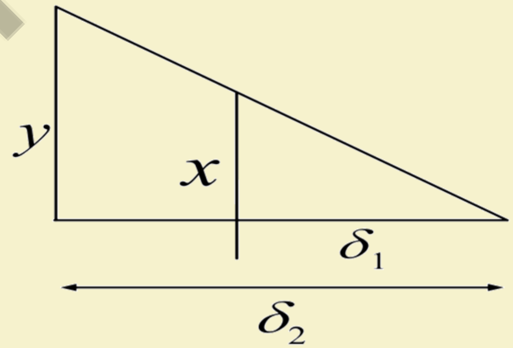
Fuzzification of inputs

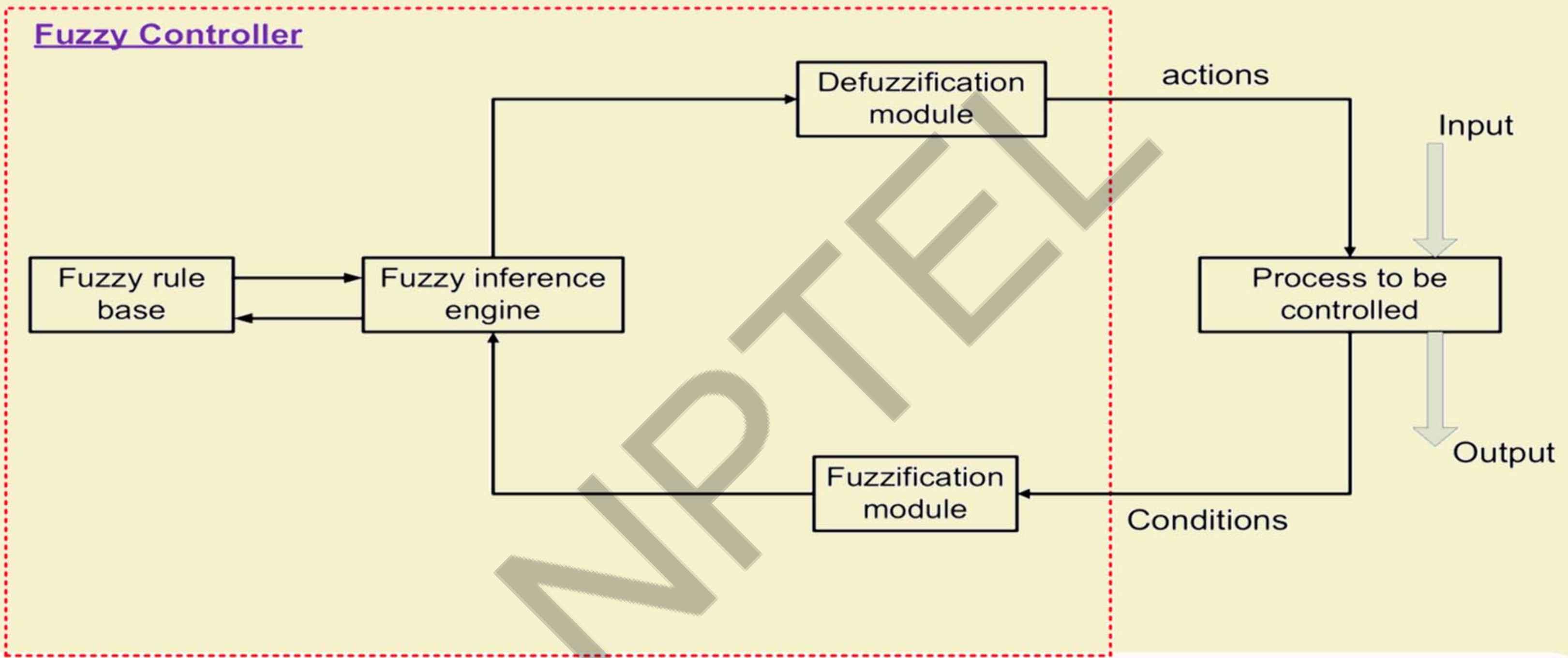
Hint : Use the principle of similarity. $\frac{x}{y} = \frac{\delta_1}{\delta_2}$





Thus, $\frac{x}{1} = \frac{1.5-1.04}{1.5-0.8}$, that is, $x = 0.6571$




Fuzzy Controller
Defuzzification module
actions
Input
Fuzzy rule base
Fuzzy inference engine
Process to be controlled
Output
Fuzzification module
Conditions

Fuzzy rule base

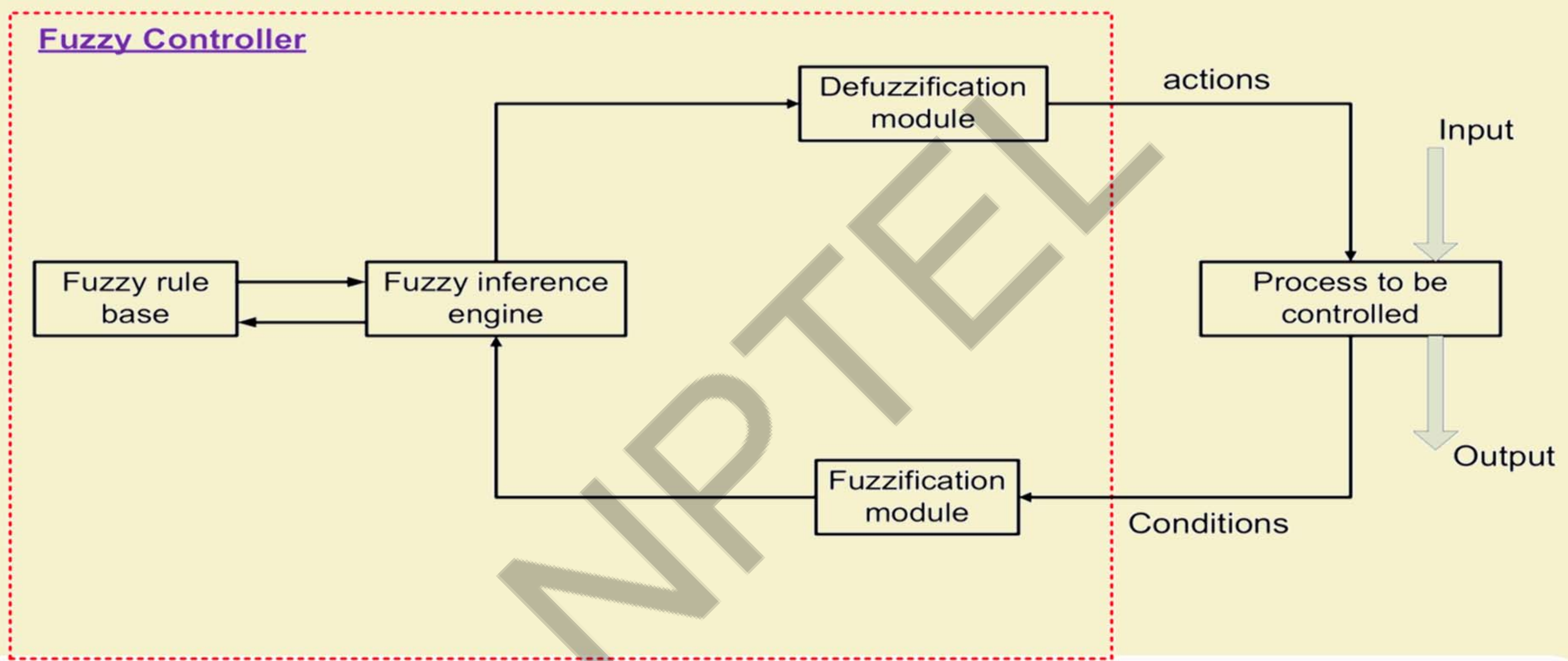
$$
\begin{array}{c c}
 & \begin{array}{ccccc} LT & AL & AA & AR & RT \end{array} \\
\begin{array}{c} VN \\ NR \\ FR \\ VF \end{array} & \begin{bmatrix} AA & AR & AL & AL & AA \\ AA & AA & RT & AA & AA \\ AA & AA & AR & AA & AA \\ AA & AA & AA & AA & AA \end{bmatrix}
\end{array}
$$

Rule strength computation

There are many rules in the rule base and all rules may not be applicable.

For the given $x = 1.04$ and $\theta = 30°$, only following four rules out of 20 rules are fireable.

- **R1**: If (distance is NR) and (angle is AA) Then (deviation is RT)
- **R2**: If (distance is NR) and (angle is AR) Then (deviation is AA)
- **R3**: If (distance is FR) and (angle is AA) Then (deviation is AR)
- **R4**: If (distance is FR) and (angle is AR) Then (deviation is AA)


Fuzzy Controller
Defuzzification module
actions
Input
Fuzzy rule base
Fuzzy inference engine
Process to be controlled
Output
Fuzzification module
Conditions

Rule strength computation

The strength (also called α values) of the firable rules are calculated as follows.

- $\alpha(R1) = \min(\mu_{NR}(x), \mu_{AA}(y)) = \min(0.6571, 0.3333) = 0.3333$
- $\alpha(R2) = \min(\mu_{NR}(x), \mu_{AR}(y)) = \min(0.6571, 0.6667) = 0.6571$
- $\alpha(R3) = \min(\mu_{FR}(x), \mu_{AA}(y)) = \min(0.3429, 0.3333) = 0.3333$
- $\alpha(R4) = \min(\mu_{FR}(x), \mu_{AR}(y)) = \min(0.3429, 0.6667) = 0.3429$

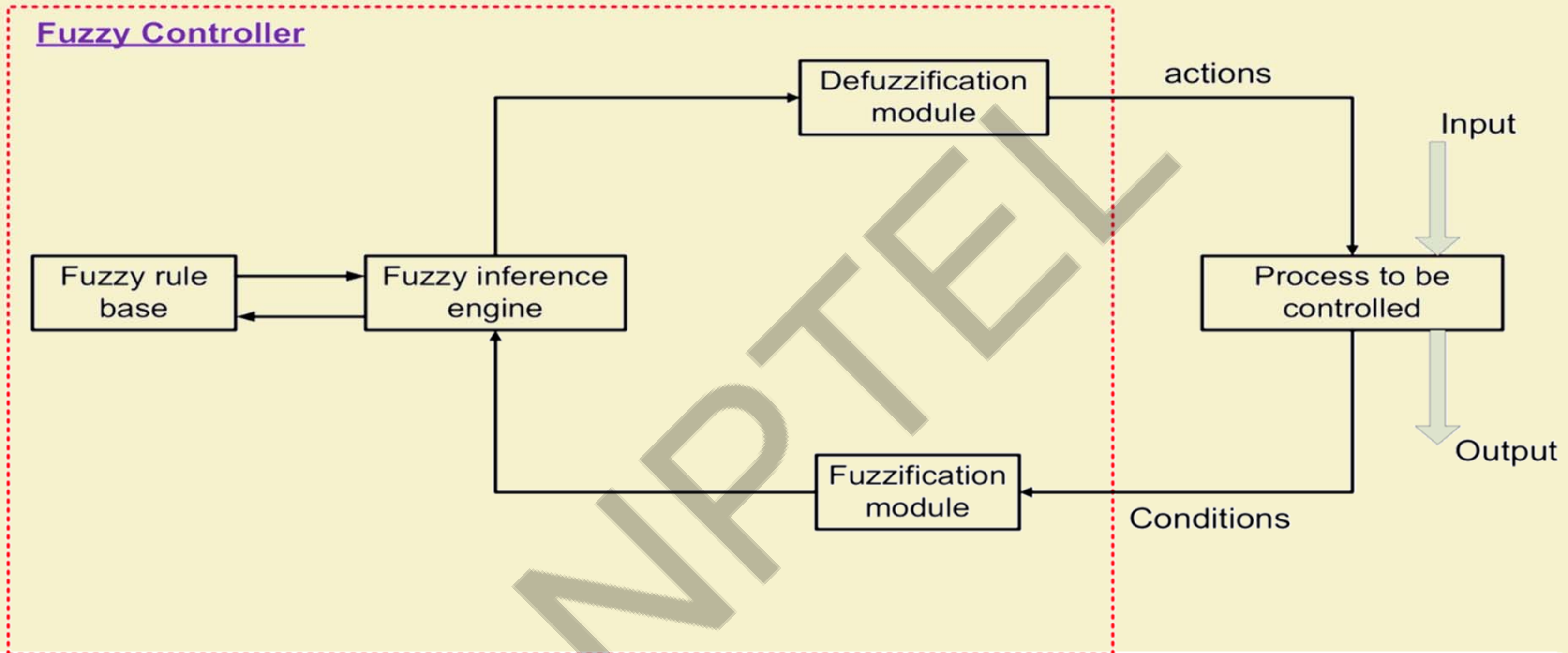
In practice, all rules which are above certain threshold value of the rule strength are selected for the output computation.

Rule strength computation

Let the threshold of α values) be 0.3400.

Then the selected rules are

- ~~$\alpha(R1) = \min(\mu_{NR}(x), \mu_{AA}(y)) = \min(0.6571, 0.3333) = 0.3333$~~
- $\alpha(R2) = \min(\mu_{NR}(x), \mu_{AR}(y)) = \min(0.6571, 0.6667) = 0.6571$
- ~~$\alpha(R3) = \min(\mu_{FR}(x), \mu_{AA}(y)) = \min(0.3429, 0.3333) = 0.3333$~~
- $\alpha(R4) = \min(\mu_{FR}(x), \mu_{AR}(y)) = \min(0.3429, 0.6667) = 0.3429$


Fuzzy Controller
Defuzzification module
actions
Input
Fuzzy rule base
Fuzzy inference engine
Process to be controlled
Output
Fuzzification module
Conditions

Fuzzy output

The next step is to determine the fuzzified outputs corresponding to each fired rules. The working principle of doing this is first discussed and then we illustrate with the running example.

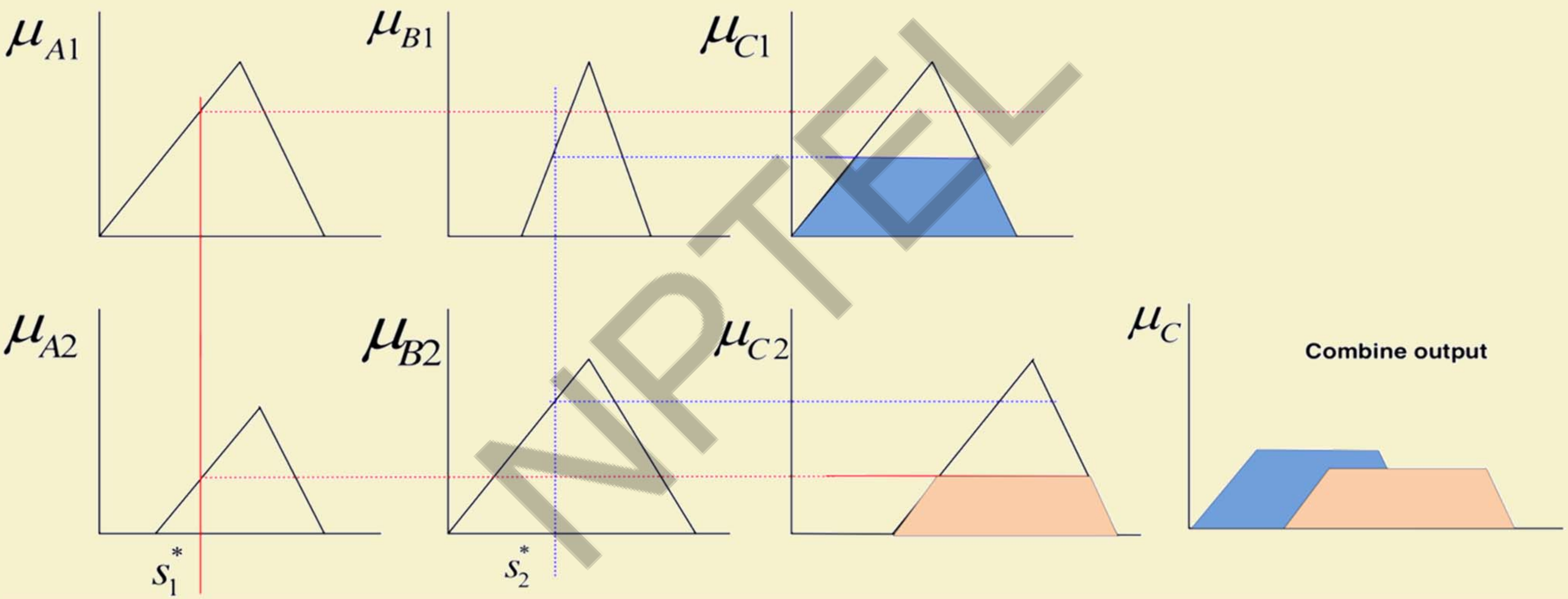
Suppose, only two fuzzy rules, $R1$ and $R2$, for which we are to decide fuzzy output.

- $R1$: $IF\ (s_1\ is\ A_1)\ AND\ (s_2\ is\ B_1)\ THEN\ (f\ is\ C_1)$
- $R2$: $IF\ (s_1\ is\ A_2)\ AND\ (s_2\ is\ B_2)\ THEN\ (f\ is\ C_2)$

Suppose, ${s_1}^*$ and ${s_2}^*$ are the inputs for fuzzy variables s_1 and s_2. μ_{A1}, μ_{A2}, μ_{B1}, μ_{B2}, μ_{C1} and μ_{C2} are the membership values for different fuzzy sets.

Fuzzy output

The fuzzy output computation is graphically shows in the following figure.



Fuzzy output

Note:

- We take min of membership function values for each rule.
- Output membership function is obtained by aggregating the membership function of result of each rule.
- Fuzzy output is nothing but fuzzy OR of all output of rules.

Illustration : Mobile Robot

For four rules, we find the following results.

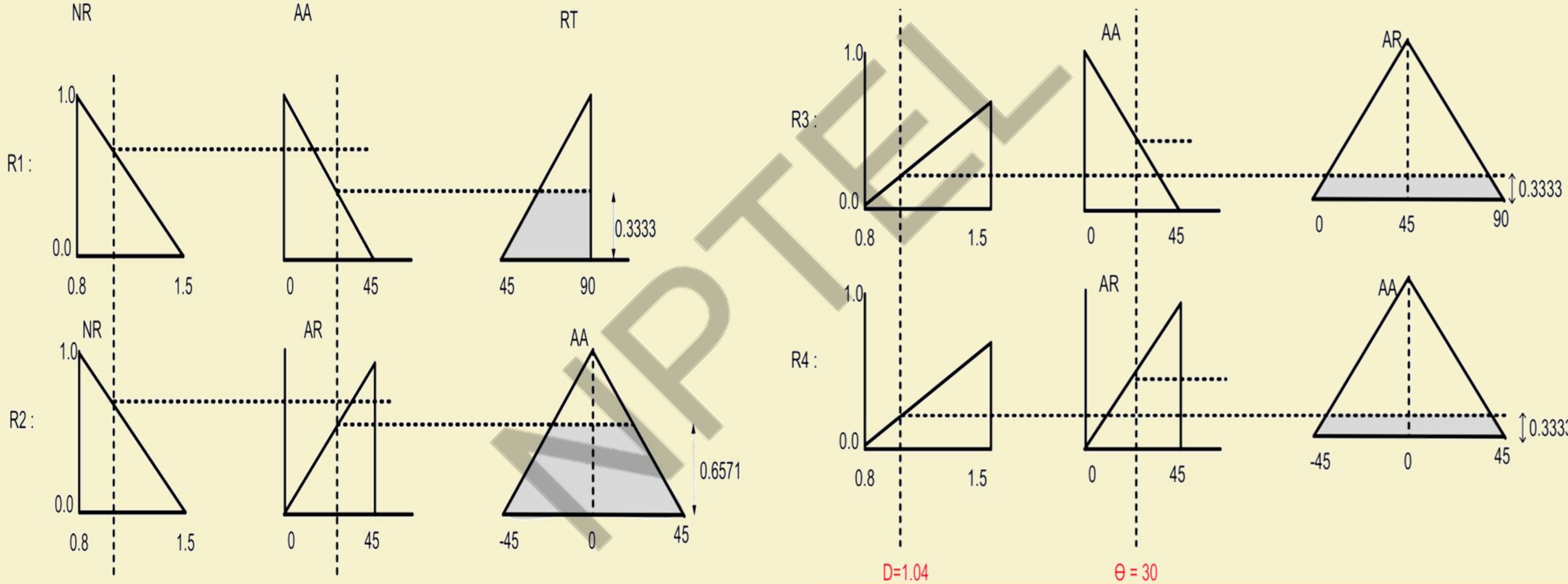
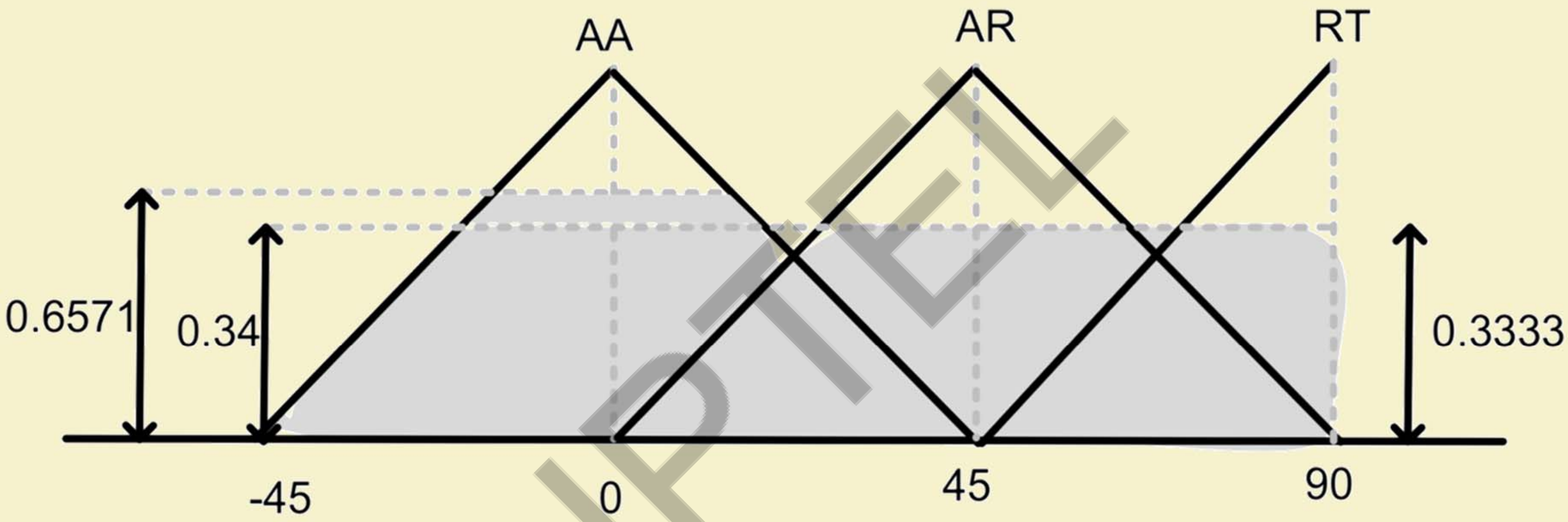


Illustration : Mobile Robot



Aggregation of all results

Defuzzification

The fuzzy output needs to be defuzzified and its crisp value has to be determined for the output to take decision.

Mobile Robot

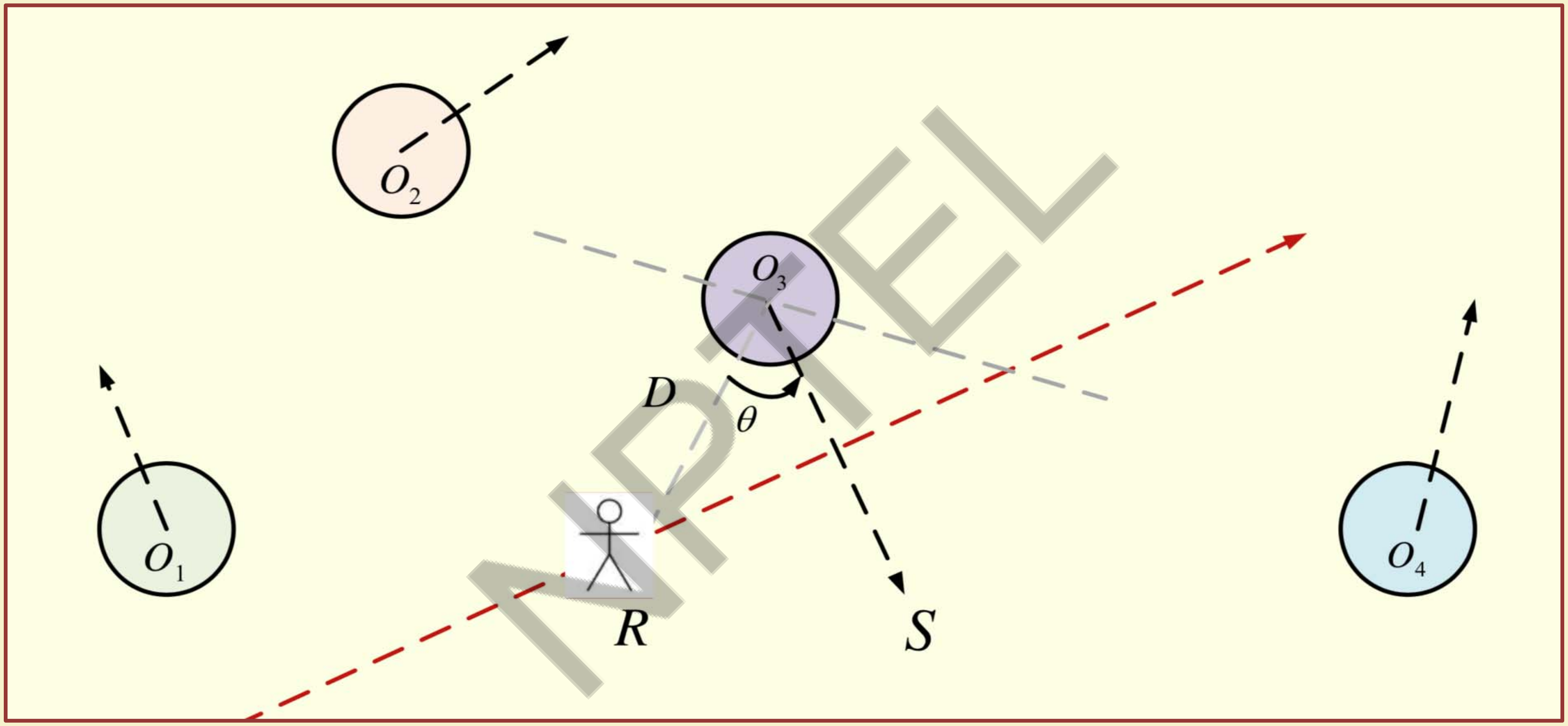
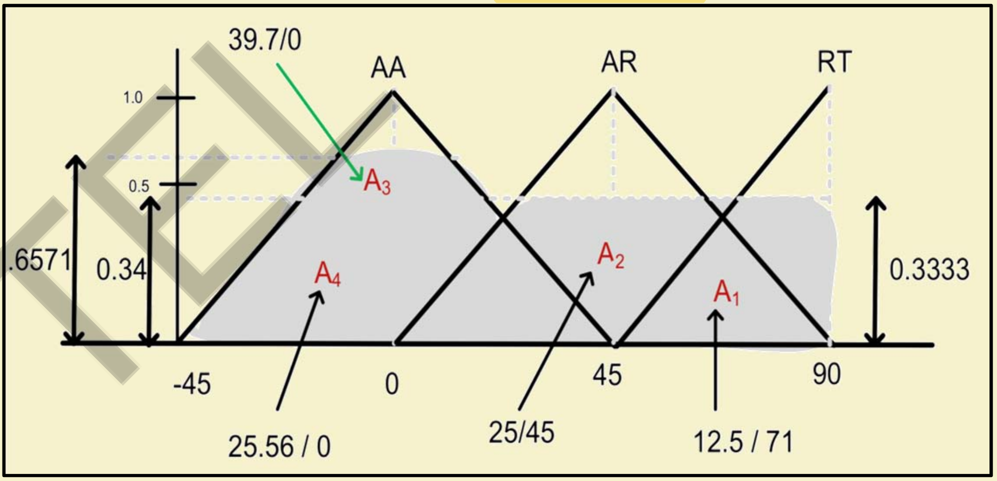


Illustration : Mobile Robot

From the combined fuzzified output for all four fired rules, we get the crisp value using Center of Sum method as follows.

$$v = \frac{12.5\times71+25\times45+25.56\times0+25.56\times0}{12.5+39.79+25+25.56} = 19.59$$



Conclusion : Therefore, the robot should deviate by 19.58089 degree towards the right with respect to the line joining to the move of direction to avoid collision with the obstacle O_3.

Thank You!!

IIT KHARAGPUR

NPTEL ONLINE CERTIFICATION COURSES

NPTEL

Introduction to Soft Computing

Solving optimization problems

Debasis Samanta

Department of Computer Science and Engineering

IIT KHARAGPUR

Concept of optimization problem

Optimization : Optimum value that is either minimum or maximum value.

$$y = F(x)$$

Example:

$$2x - 6y = 11 \text{ or } y = \frac{2x-11}{6}$$

Can we determine an optimum value for y?

Similarly, in the following case

$$3x + 4y \geq 56$$

These are really not related to optimization problem!

Defining an optimization problem

Suppose, we are to design an optimal pointer made of some material with density ρ. The pointer should be as large as possible, no mechanical breakage and deflection of pointing at end should be negligible.

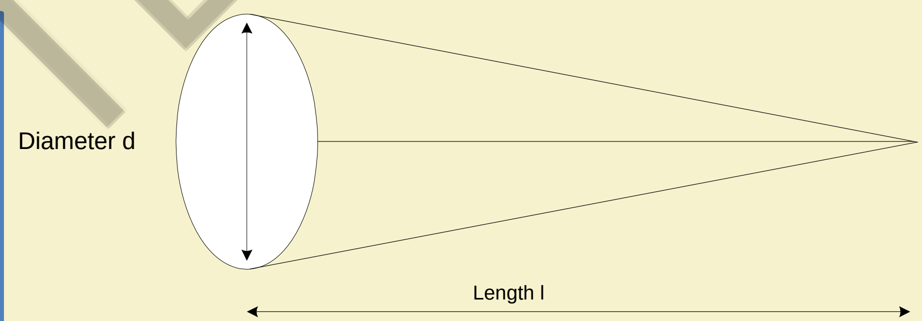
The task is to select the best pointer out of many all possible pointers.

Suppose, s is the strength of the pointer.

- ✓ Mass of the stick is denoted by:

$$M = \frac{1}{3}\prod\left(\frac{d}{2}\right)^2 \times l \times \rho = \frac{1}{12}\prod d^2 \times l \times \rho$$

- ✓ Deflection $\delta = f_1(d, l, \rho)$
- ✓ Strength $s = f_2(d, l, \rho)$



Defining an optimization problem

The problem can be stated as

- ✓ Objective function

$$\text{Minimize } M = \frac{1}{12} \prod d^2 \times l \times \rho$$

- ✓ Subject to

$\delta \leq \delta_{th}$, where, δ_{th} = allowable deflection

$s \leq s_{th}$, where, s_{th} = required strength

and

$$d_{min} \leq d \leq d_{max}$$
$$I_{min} \leq I \leq I_{max}$$

Defining an optimization problem

An optimization problem can be formally defined as follows:

- ✓ Maximize (or Minimize)

$$y_i = f_i(x_1, x_2 \dots \dots, x_n)$$
$$\text{where } i = 1,2, \dots \dots k, k \geq 1$$

- ✓ Subject to

$$g_i(x_1, x_2 \dots \dots, x_n) ROP_i \; c_i$$

Where $i = 1,2, \dots \dots j$, j ≥ 0 ROP_i denotes some relational operator and $c_i = 1,2, \dots \dots j$ are some constants

and

$x_i \;\; ROP \;\; d_i$, for all $i = 1,2 \dots . n \; (n \geq 1)$

Here, x_i denotes a design parameter and d_i is some constant.

Some Benchmark Optimization Problems

Exercises: Mathematically define the following optimization problems.

- ✓ Traveling Salesman Problem
- ✓ Knapsack Problem
- ✓ Graph Colouring Problem
- ✓ Job Machine Assignment Problem
- ✓ Coin Change Problem
- ✓ Binary search tree construction problem

Types of Optimization Problem

Unconstrained optimization problem

Problem is without any functional constraint.

Example:

Minimize $y = f(x_1, x_2) = (x_1 - 5)^2 + (x_2 - 3)^3$

where, $x_1, x_2 \geq 0$

Note:

Here $g_j = NULL$

Types of Optimization Problem

Constrained optimization problem

Optimization problem with at one or more functional constraint(s).

Example:

Maximize $y = f(x_1, x_2, \ldots \ldots x_n)$

Subject to

$$g_i(x_1, x_2, \ldots \ldots x_{ni}) \geq c_i$$

where $i = 1,2,3 \ldots . k$ and $k > 0$

and $x_1, x_2, \ldots \ldots x_n$ are design parameters.

Types of Optimization Problem

Integer Programming problem

If all the design variables take some integer values.

Example:

Minimize $y = f(x_1, x_2) = 2x_1 + x_2$

Subject to

$$x_1 + x_2 \leq 3$$
$$5x_1 + 2x_2 \leq 9$$

and x_1, x_2 are integer variables.

Types of Optimization Problem

- ✓ **Real-valued problem**
  If all the design variables are bound to take real values.

- ✓ **Mixed-integer programming problem**
  Some of the design variables are integers and the rest of the variables take real values.

Types of Optimization Problem

Linear optimization problem

Both objective functions as well as all constraints are found to be some linear functions of design variables.

Example:

Maximize $y = f(x_1, x_2) = 2x_1 + x_2$

Subject to

$$x_1 + x_2 \leq 3$$
$$5x_1 + 2x_2 \leq 10$$

and $x_1, x_2 \geq 0$

Types of Optimization Problem

Non-Linear optimization problem

If either the objective function or any one of the functional constraints are non-linear function of design variables.

Example:

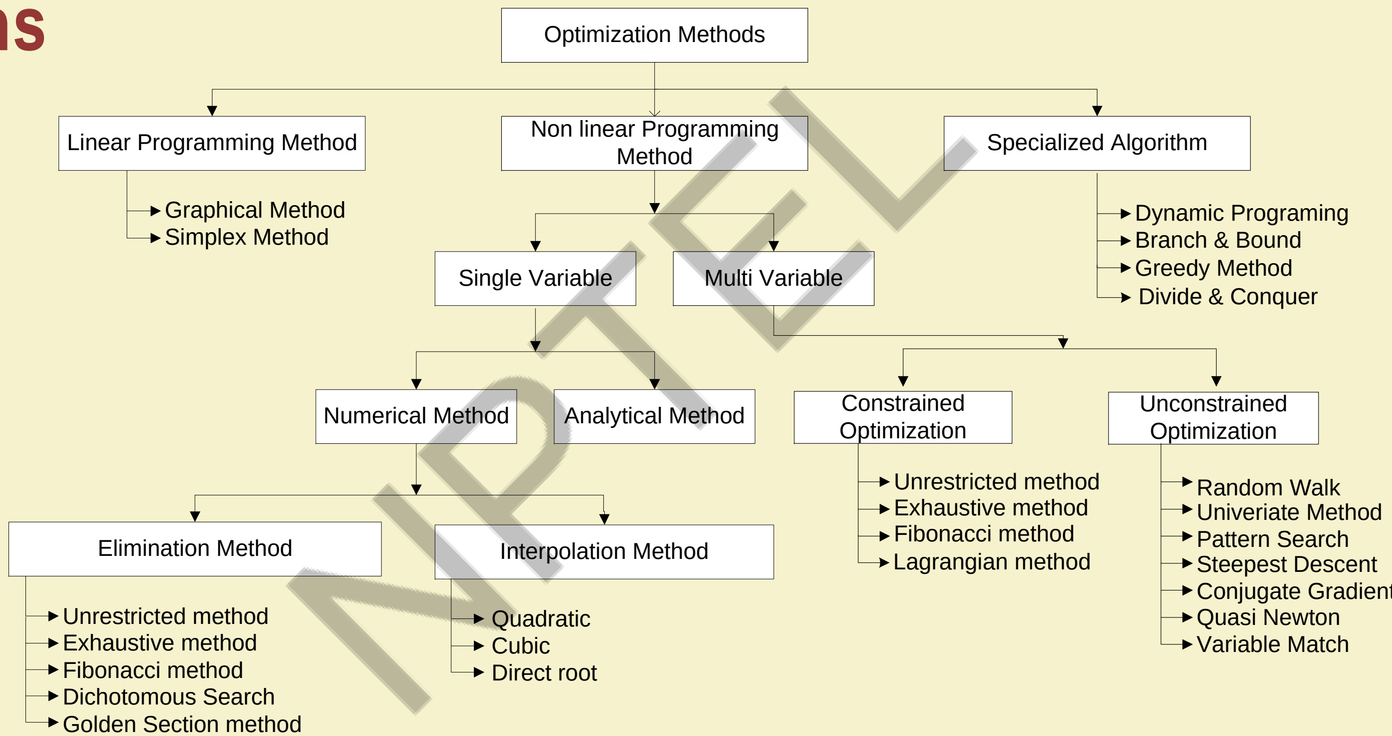
Maximize $y = f(x_1, x_2) = {x_1}^2 + 5{x_2}^3$

Subject to

$$ {x_1}^4 + 3{x_2}^2 \leq 629 $$
$$ 2{x_1}^3 + 4{x_2}^3 \leq 133 $$

and $x_1, x_2 \geq 0$

Traditional approaches to solve optimization problems



Example : Analytical Method

Suppose, the objective function: $y = f(x)$. Let $f(x)$ be a polynomial of degree m and $(m > 0)$

If $y' = f'(x) = 0$ for some $x = x^*$, then we say that y is optimum (i.e. either minimum or maximum point exist) at the point $x = x^*$.

If $y' = f'(x) \neq 0$ for some $x = x^*$, then we say that there is no optimum value at $x = x^*$ (i.e. $x = x^*$ is an inflection point)

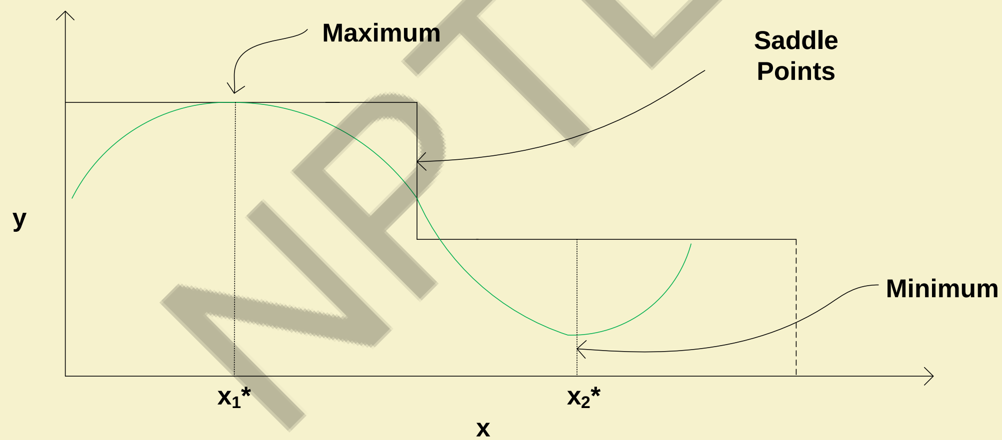
An inflection point is also called a **saddle point**.

Example : Analytical Method

Note:

An inflection point is a point, that is, neither a maximum nor a minimum at that point.

Following figure explains the concepts of minimum, maximum and saddle point.



Example : Analytical Method

Let us generalize the concept of $Analytical\ method$.

If $y = f(x)$ is a polynomial of degree m, then there are m number of candidate points to be checked for optimum or saddle points.

Suppose, y^n is the n^{th} derivative of y.

To further investigate the nature of the point, we determine (first non-zero) $(n-th)$ higher order derivative

$$y^n = f^n(x = x^*)$$

There are two cases.

Case 1: If $y^n \neq 0$ for $n =$ odd number, then x^* is an inflection point.

Case 2: If $y^n = 0$ for $n =$ odd number, then there exist an optimum point at x^*.

Example : Analytical Method

In order to decide the point x as minimum or maximum, we have to find the next higher order derivative, that is $y^{n+1} = f^{n+1}(x = x^*)$.

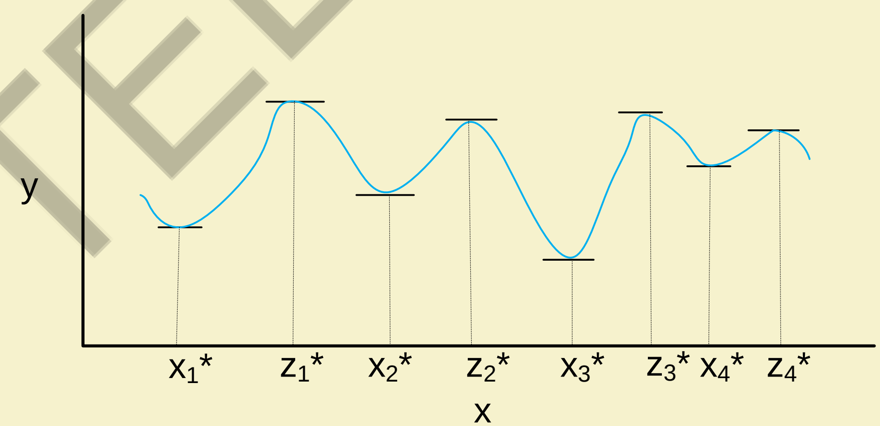
There are two sub cases:

Case 2.1:

If $y^{n+1} = f^{n+1}(x = x^*)$ is positive then x is a local minimum point.

Case 2.2:

If $y^{n+1} = f^{n+1}(x = x^*)$ is negative then x is a local maximum point.



If $y^{n+1} = f^{n+1}(x = x^*) = 0$ then we are to repeat the next higher order derivative.

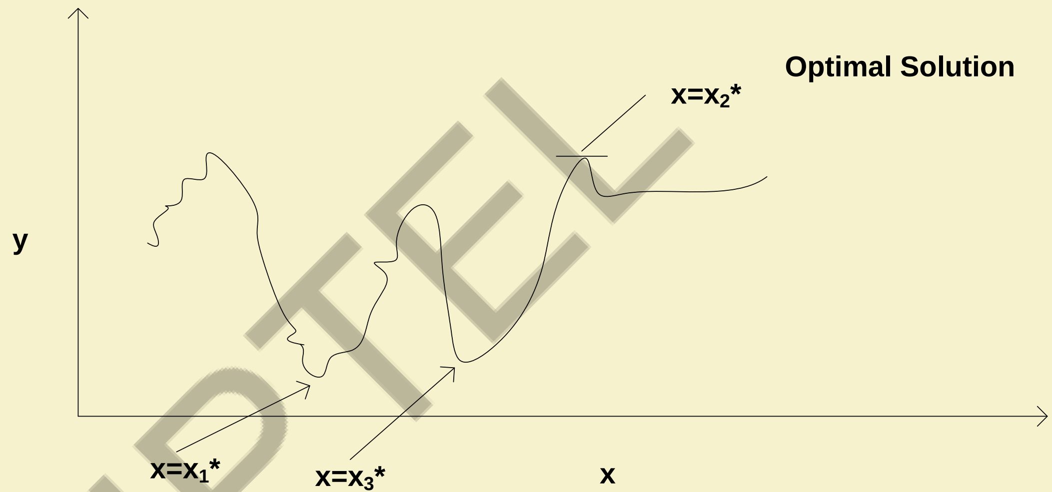
Question

$$y = f(x)$$

$$\frac{d^2y}{dx} = +ve \implies x = x_1^*$$

$$\frac{d^4y}{dx} = -ve \implies x = x_2^*$$

$$\frac{d^6y}{dx} = \pm ve \implies x = x_3^*$$



Is the analytical method solves optimization problem with multiple input variables?

1) If "Yes", than how?
2) If "No", than why not?

Exercise

Determine the minimum or maximum or saddle points, if any for the following single variable function $f\ (x)$

$$f(x) = \frac{x^2}{2} + \frac{125}{x}$$

for some real values of x.

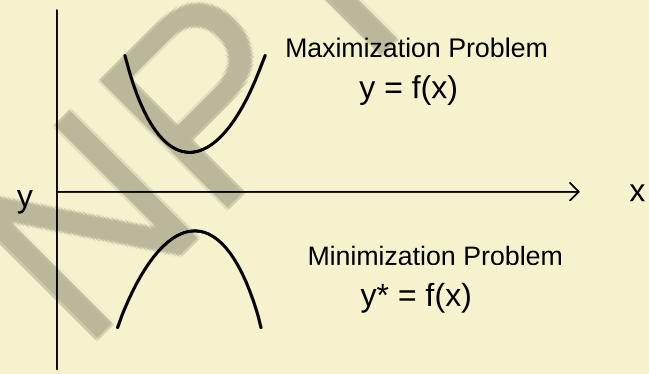
Duality Principle

Principle

A Minimization (Maximization) problem is said to have dual problem if it is converted to the maximization (Minimization) problem.

The usual conversion from maximization⇔ minimization

$$y = f(x) \Leftrightarrow y^* = -f(x) \qquad\qquad y = f(x) \Leftrightarrow y^* = \frac{1}{f(x)}$$



Limitations of the traditional optimization approach

- ✓ Computationally expensive.
- ✓ For a discontinuous objective function, methods may fail.
- ✓ Method may not be suitable for parallel computing.
- ✓ Discrete (integer) variables are difficult to handle.
- ✓ Methods may not necessarily adaptive.

Soft Computing techniques have been evolved to address the above mentioned limitations of solving optimization problem with traditional approaches.

Thank You!!

IIT KHARAGPUR

NPTEL ONLINE CERTIFICATION COURSES

NPTEL

Introduction to Soft Computing

Concept of GA

Debasis Samanta

Department of Computer Science and Engineering

IIT KHARAGPUR

Limitations of the traditional optimization approaches

Limitations:

- Computationally expensive.
- For a discontinuous objective function, methods may fail.
- Method may not be suitable for parallel computing.
- Discrete (integer) variables are difficult to handle.
- Methods may not necessarily adaptive.

Evolutionary algorithms have been evolved to address the above mentioned limitations of solving optimization problems with traditional approaches.

Evolutionary Algorithms

The algorithms, which follow some biological and physical behaviours:

Biologic behaviours:

1) Genetics and Evolution – > Genetic Algorithms (GA)
2) Behaviour of ant colony – > Ant Colony Optimization (ACO)
3) Human nervous system – > Artificial Neural Network (ANN)

In addition to that there are some algorithms inspired by some physical behaviours:

Physical behaviours:

1) Annealing process – > Simulated Annealing (SA)
2) Swarming of particle – > Particle Swarming Optimization (PSO)
3) Learning – > Fuzzy Logic (FL)

Genetic Algorithm

It is a subset of evolutionary algorithm:

1) Ant Colony optimization
2) Swarm Particle Optimization

Models biological processes:

1) Genetics
2) Evolution

To optimize highly complex objective functions:

1) Very difficult to model mathematically
2) NP-Hard (also called combinatorial optimization) problems (which are computationally very expensive)
3) Involves large number of parameters (discrete and/or continuous)

Background of Genetic Algorithm

Firs time introduced by Prof. John Holland (of Michigan University, USA, 1965). But, the first article on GA was published in 1975.

Principles of GA based on two fundamental biological processes:

1) **Genetics:** Gregor Johan Mendel (1865)
2) **Evolution:** Charles Darwin (1875)

Genetics

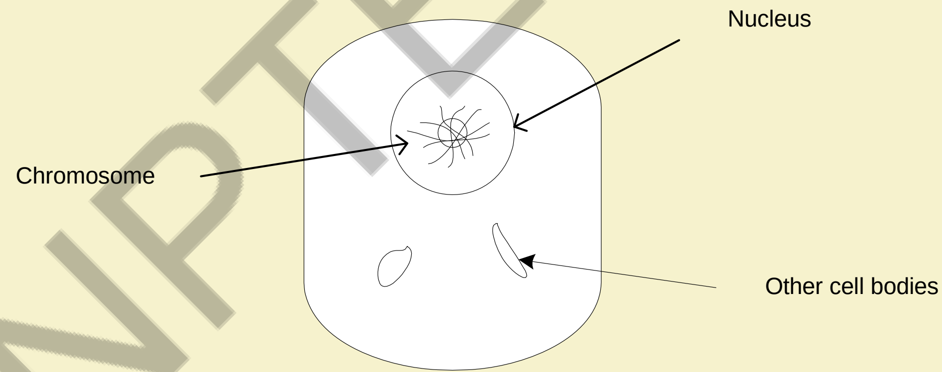
A brief account on genetics

The basic building blocks in living bodies are cells. Each cell carries the basic unit of heredity, called gene

For a particular specie, number of chromosomes is fixed.

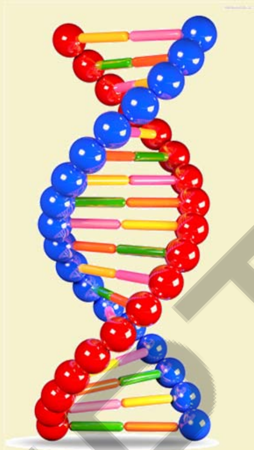
Examples

- Mosquito: 6
- Frogs: 26
- Human: 46
- Goldfish: 94



A brief account on genetics

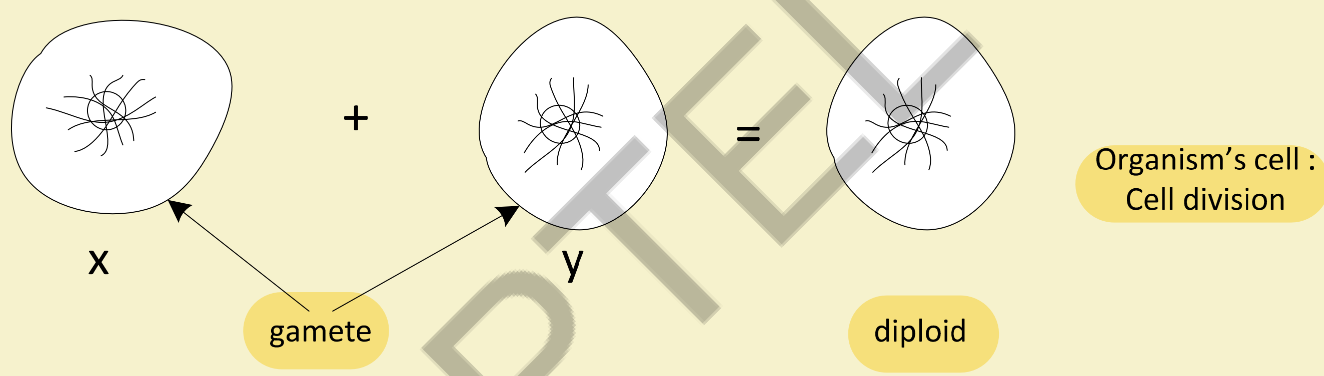
Genetic code



- Spiral helix of protein substance is called DNA.
- For a species, DNA code is unique, that is, vary uniquely from one to other.
- DNA code (inherits some characteristics from one generation to next generation) is used as biometric trait.

A brief account on genetics

Reproduction

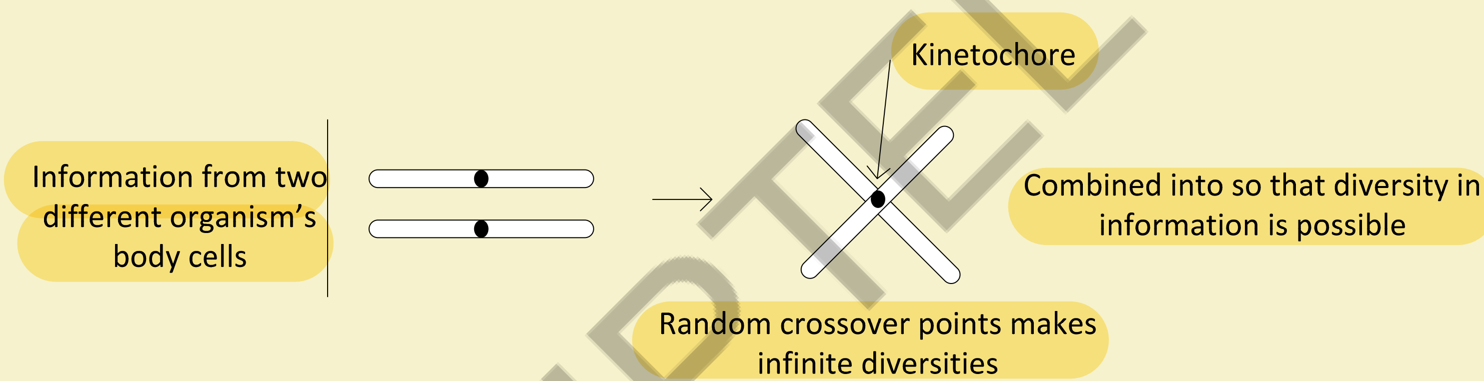


haploid (Reproductive cell has half the number of chromosomes)

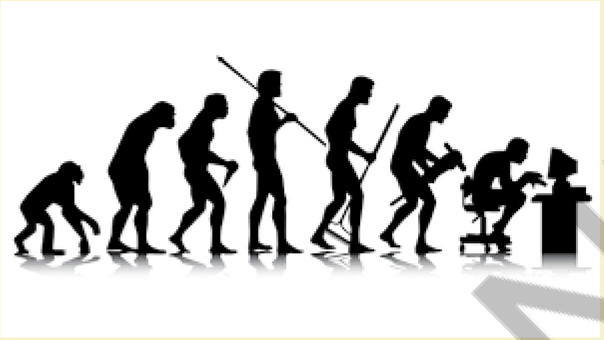
diploid Each chromosome from both haploids are combined to have full numbers

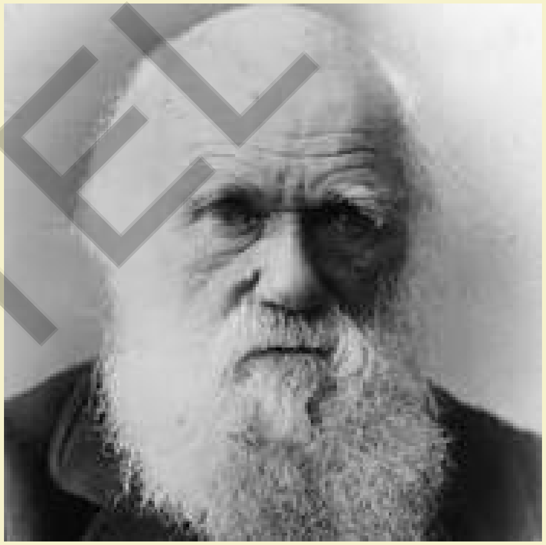
A brief account on genetics

Crossing over



Evolution





A brief account on genetics

Evolution : Natural Selection

Four primary premises:

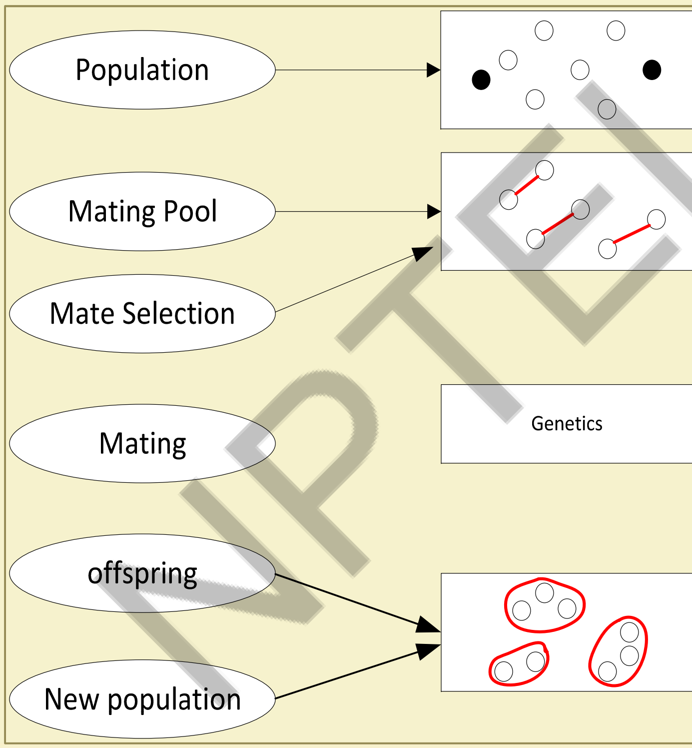
1) **Information propagation:** An offspring has many of its characteristics of its parents (i.e. information passes from parent to its offspring). **[Heredity]**
2) **Population diversity:** Variation in characteristics in the next generation. **[Diversity]**
3) **Survival for existence:** Only a small percentage of the offspring produced survive to adulthood. **[Selection]**
4) **Survival of the best:** Offspring survived depends on their inherited characteristics. **[Ranking]**

A brief account on genetics

Mutation:

To make the process forcefully dynamic when variations in population going to stable.

Biological process : A quick overview

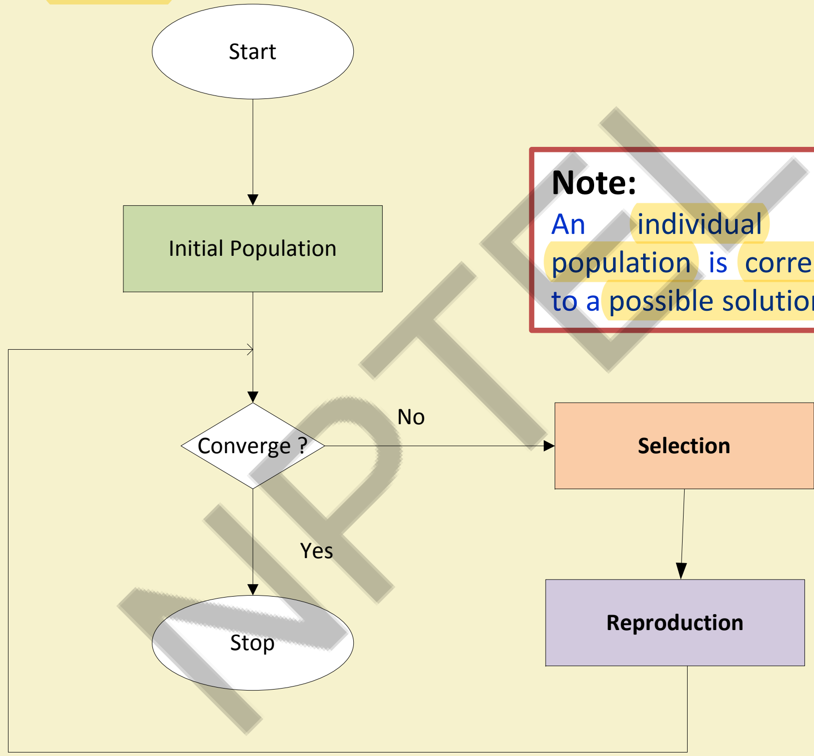


Working of Genetic Algorithm

Definition of GA:

Genetic algorithm is a population-based probabilistic search and optimization techniques, which works based on the mechanisms of natural genetics and natural evaluation.

Framework of GA



Note:
An individual in the population is corresponding to a possible solution

Thank You!!

IIT KHARAGPUR

NPTEL ONLINE CERTIFICATION COURSES

NPTEL

Introduction to Soft Computing

Concept of GA

Debasis Samanta

Department of Computer Science and Engineering

IIT KHARAGPUR

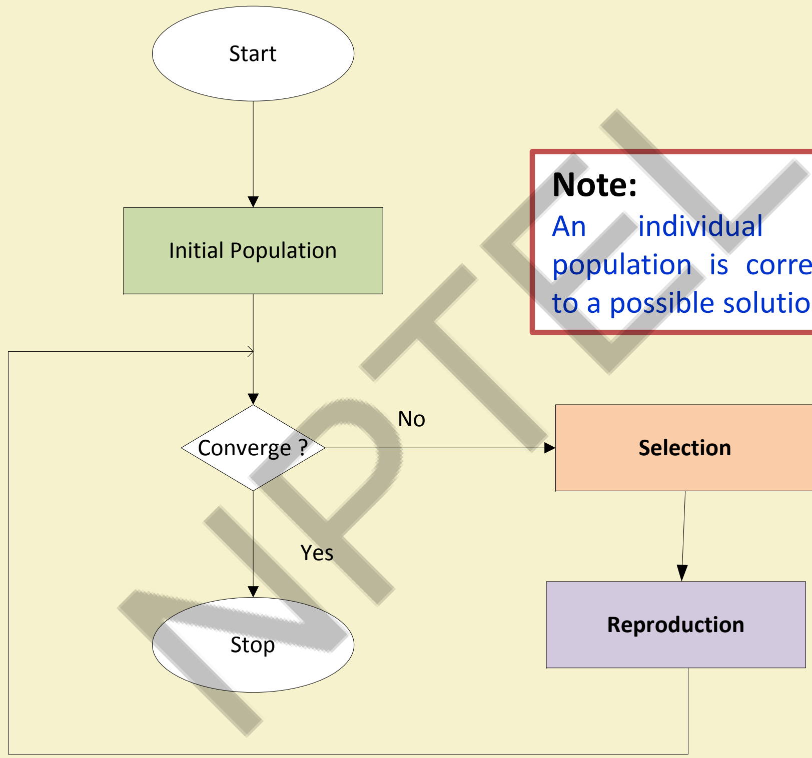
Working of Genetic Algorithm

Definition of GA:

Genetic algorithm is a population-based probabilistic search and optimization techniques, which works based on the mechanisms of natural genetics and natural evaluation.

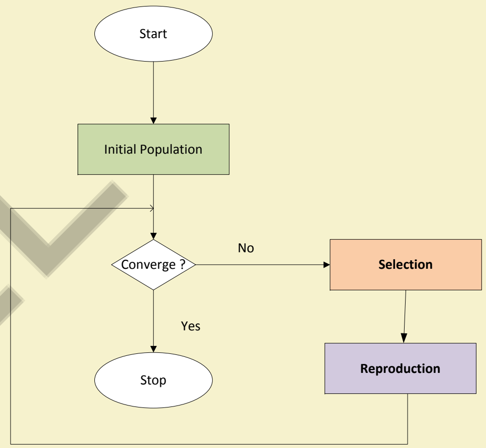
Debasis Samanta
CSE
IIT KHARAGPUR

Framework of GA



Note:
An individual in the population is corresponding to a possible solution

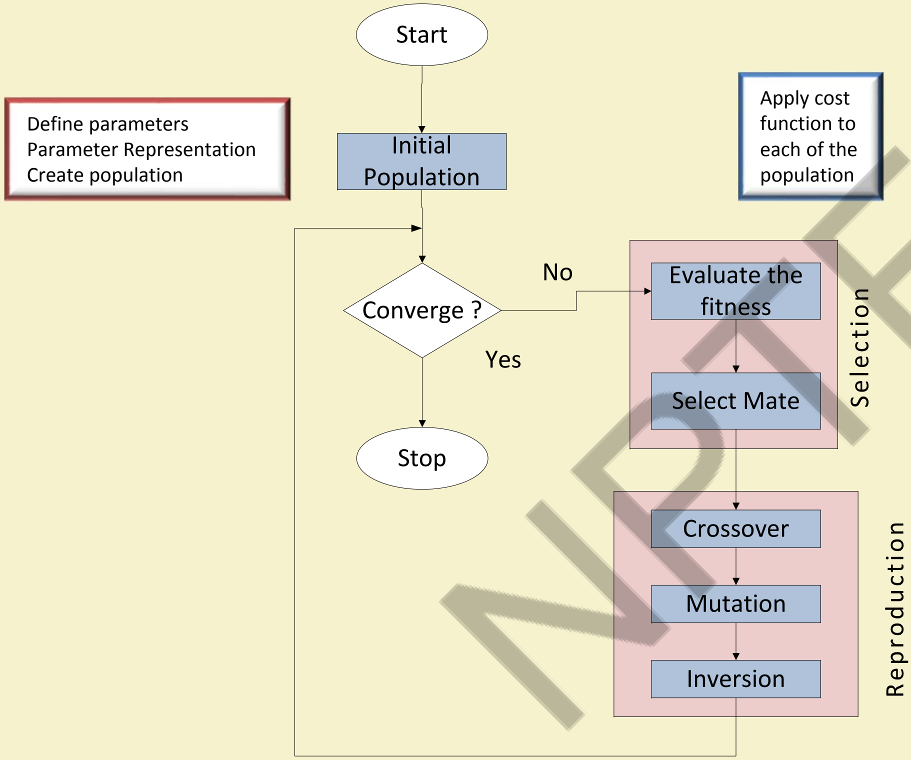
Working of Genetic Algorithm



Note:

1) GA is an iterative process.
2) It is a searching technique.
3) Working cycle with / without convergence.
4) Solution is not necessarily guaranteed. Usually, terminated with a local optima.

Framework of GA: A detail view



Optimization problem solving with GA

For the optimization problem, identify the following:

1) Objective function(s)
2) Constraint(s)
3) Input parameters
4) Fitness evaluation (it may be algorithm or mathematical formula)
5) Encoding
6) Decoding

GA Operators

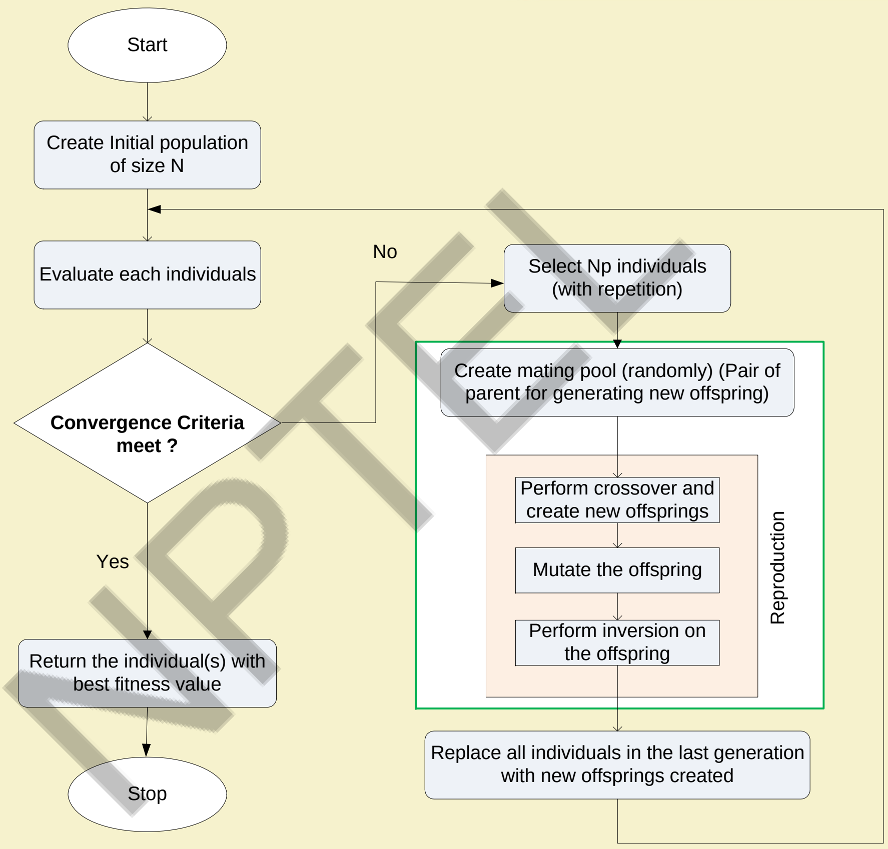
In fact, a GA implementation involved with the realization of the following operations.

1) **Encoding:** How to represent a solution to fit with GA framework.
2) **Convergence:** How to decide the termination criterion.
3) **Mating pool:** How to generate next solutions.
4) **Fitness Evaluation:** How to evaluate a solution.
5) **Crossover:** How to make the diverse set of next solutions.
6) **Mutation:** To explore other solution(s).
7) **Inversion:** To move from one optima to other.

Different GA Strategies

1) Simple Genetic Algorithm (SGA)
2) Steady State Genetic Algorithm (SSGA)
3) Messy Genetic Algorithm (MGA)

Simple GA



Start
Create Initial population of size N
Evaluate each individuals
Convergence Criteria meet ?
No
Yes
Select Np individuals (with repetition)
Create mating pool (randomly) (Pair of parent for generating new offspring)
Perform crossover and create new offsprings
Mutate the offspring
Perform inversion on the offspring
Reproduction
Return the individual(s) with best fitness value
Replace all individuals in the last generation with new offsprings created
Stop

Important parameters involved in Simple GA

SGA Parameters

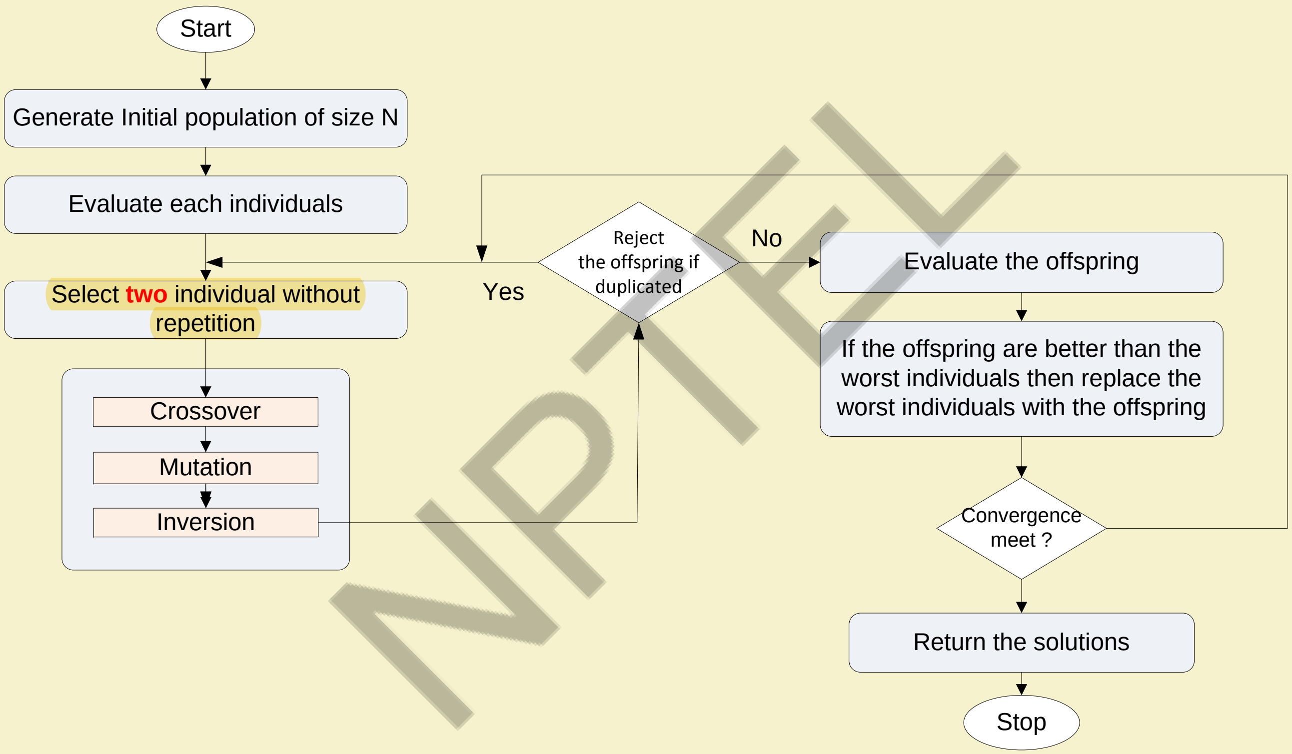
- ✓ Initial population size : N
- ✓ Size of mating pool, N_P: $N_P = P\%$ of N
- Convergence threshold δ
- Mutation μ
- Inversion η
- Crossover ρ

Salient features in SGA

Simple GA features:

- ✓ Have overlapping generation (Only fraction of individuals are replaced).
- ✓ Computationally expensive.
- ✓ Good when initial population size is large.
- ✓ In general, gives better results.
- ✓ Selection is biased toward more highly fit individuals; Hence, the average fitness (of overall population) is expected to increase in succession.
- ✓ The best individual may appear in any iteration.

Steady State Genetic Algorithm (SSGA)



Salient features in Steady-state GA

SSGA Features:

- ✓ Generation gap is small.
  Only two offspring are produced in one generation.
- ✓ It is applicable when
  - ➢ Population size is small
  - ➢ Chromosomes are of longer length
  - ➢ Evaluation operation is less computationally expensive (compare to duplicate checking)

Salient features in Steady-state GA

Limitations in SSGA:

- ✓ There is a chance of stuck at local optima, if crossover/mutation/inversion is not strong enough to diversify the population).

- ✓ Premature convergence may result.

- ✓ It is susceptible to stagnation. Inferiors are neglected or removed and keeps making more trials for very long period of time without any gain (i.e. long period of localized search).

Thank You!!