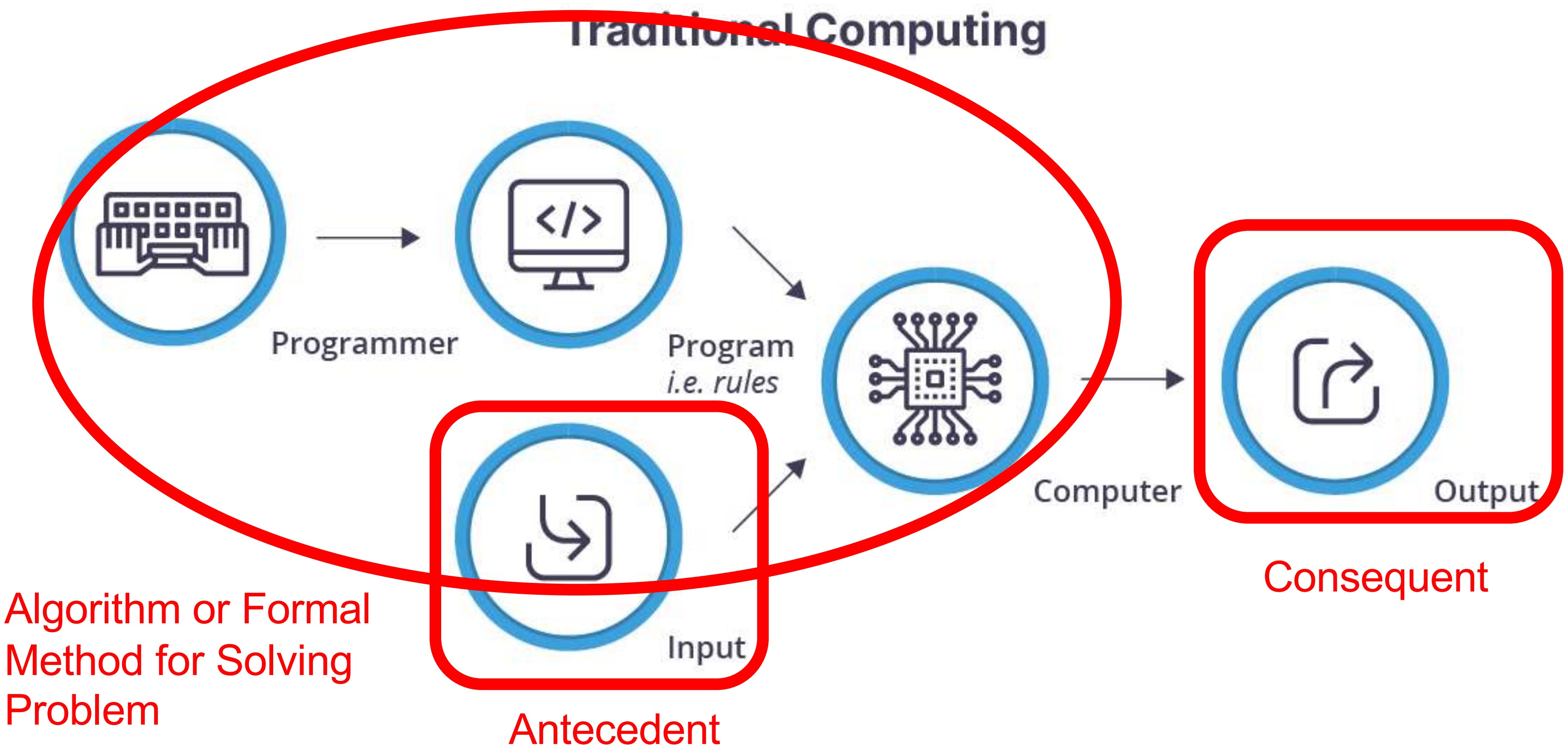
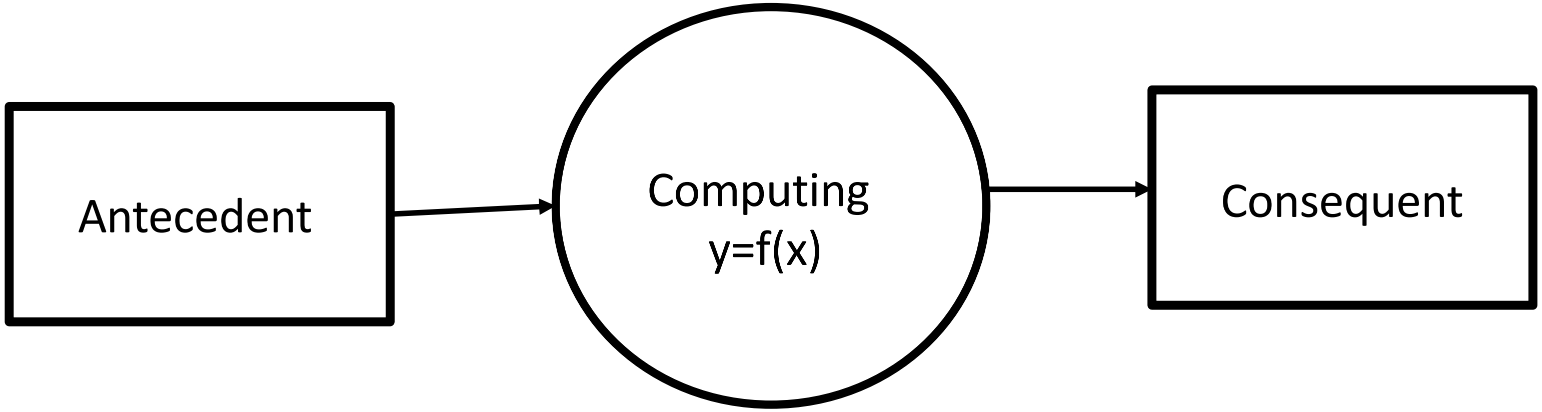
Today’s Topics

- **Course Introduction**
- **Basics of Computing**

What is Computing?

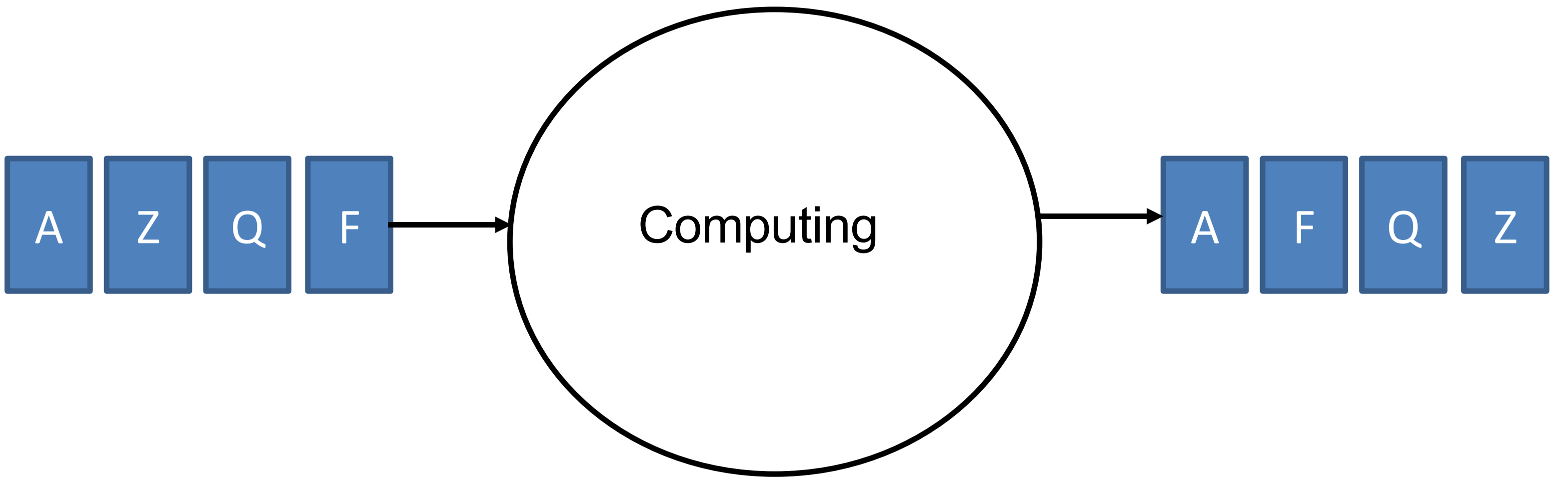


What is Computing?



x: antecedent
y: consequent
f : a mapping function or formal method or an algorithm

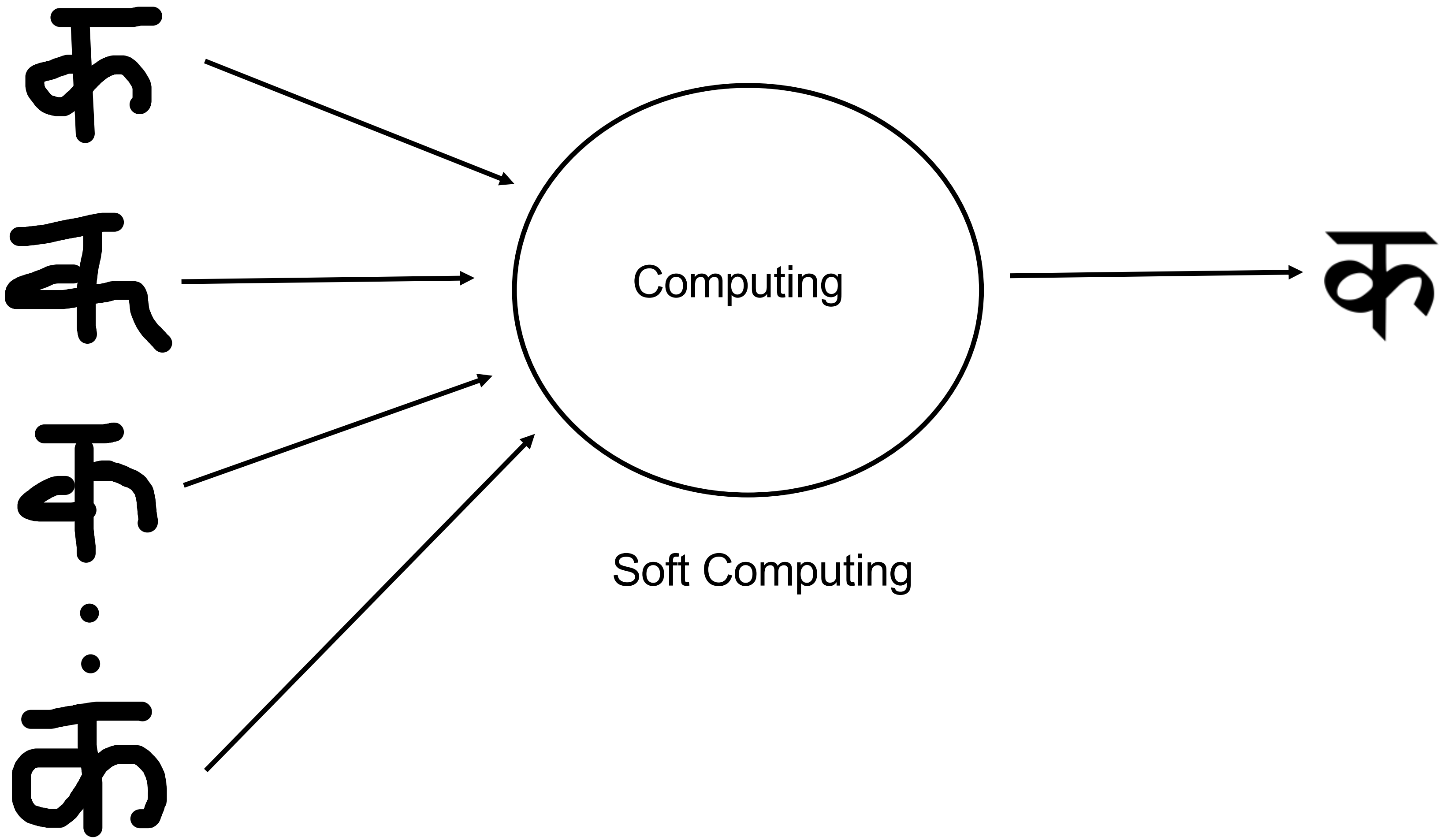
How many types of Computing?



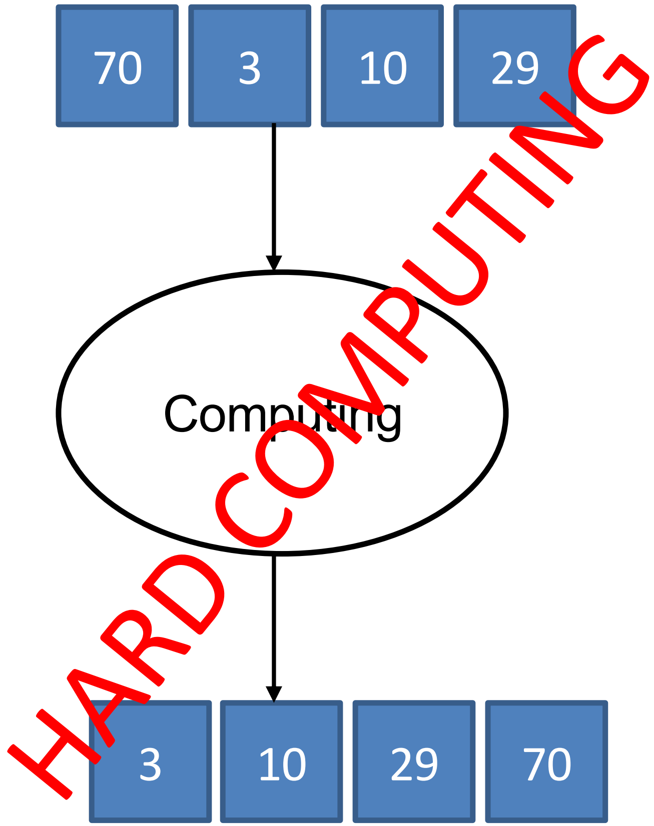
Hard Computing

How many types of Computing?

Recognizing Handwriting



Identify the type of Computing



Questions??

1. List few examples of Hard and Soft Computing

2. Identify the characteristics of Hard and Soft Computing

3. Differentiate between Hard and Soft Computing

Today’s Topics

- Hard & Soft Computing
  - Examples
  - Characteristics
  - Differences

- Need for Soft Computing

Few Examples of Hard Computing

- **Solving numerical problems (e.g. Roots of polynomials)**

```
# Solving quadratic equation of form ax**2 + bx + c = 0

# import complex math module
import cmath

a = 1
b = -9
c = 20

# calculate the discriminant
d = (b**2) - (4*a*c)

# find two solutions
root1 = (-b-cmath.sqrt(d))/(2*a)
root2 = (-b+cmath.sqrt(d))/(2*a)

print('The solution are {0} and {1}'.format(root1,root2))
```

Few Examples of Hard Computing

CHITKARA UNIVERSITY

- **Searching and sorting techniques**

```
unsorted_list = [10, 2, 38, -14, -8, 0, 9, 6]

for i in range(len(unsorted_list)):
    for j in range(i + 1, len(unsorted_list)):

        if unsorted_list[i] > unsorted_list[j]:
            unsorted_list[i], unsorted_list[j] = unsorted_list[j], unsorted_list[i]

print("The sorted list is", unsorted_list)
```

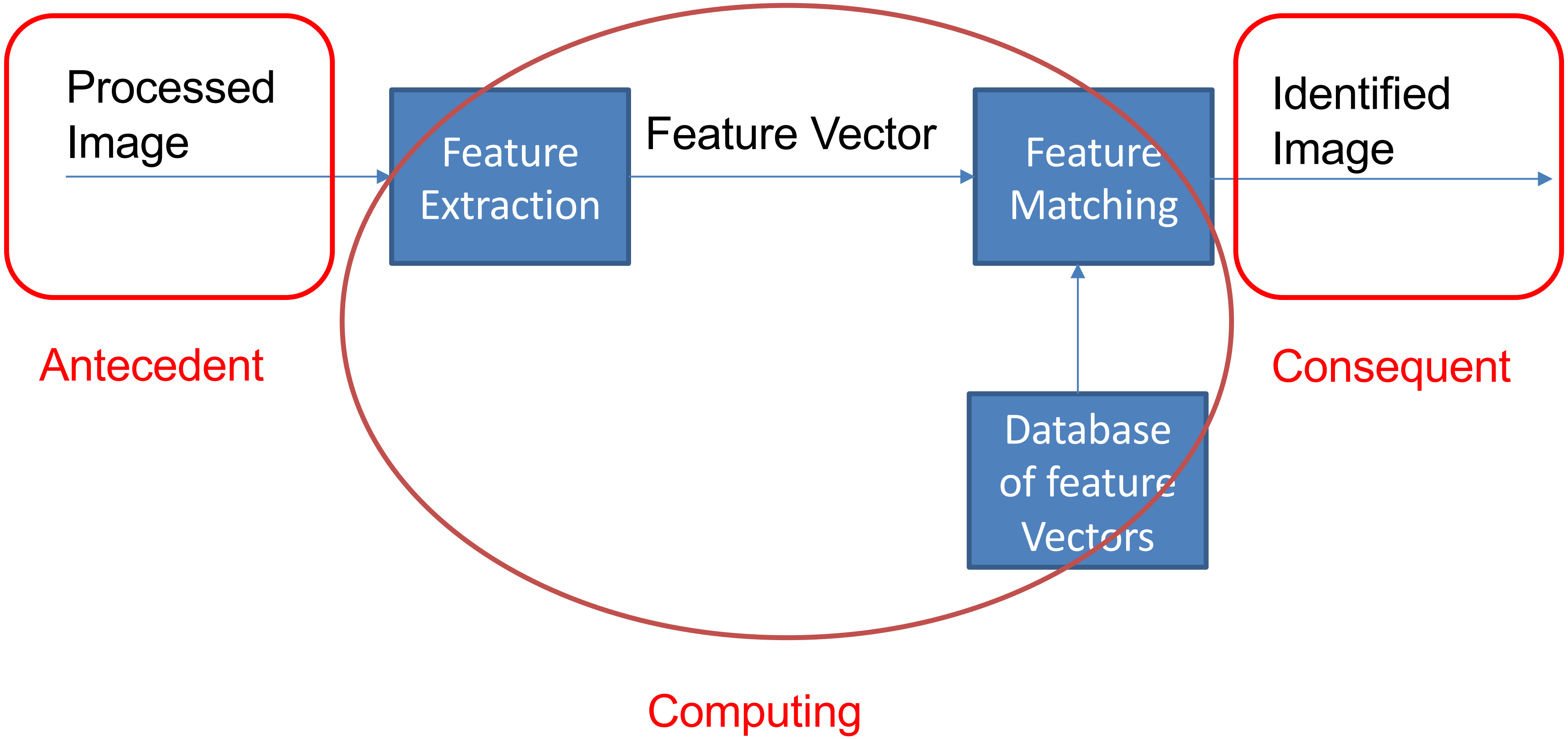
Few Examples of Hard Computing

- **Solving Computational Geometry problems**

  - **Algorithms for shortest path in Graph theory**
    - Bellman-Ford
    - Dijkstra's
    - Floyd-Warshall
  - **Algorithm for finding closest pair of points**
    - Brute Force
    - Split-Conquer

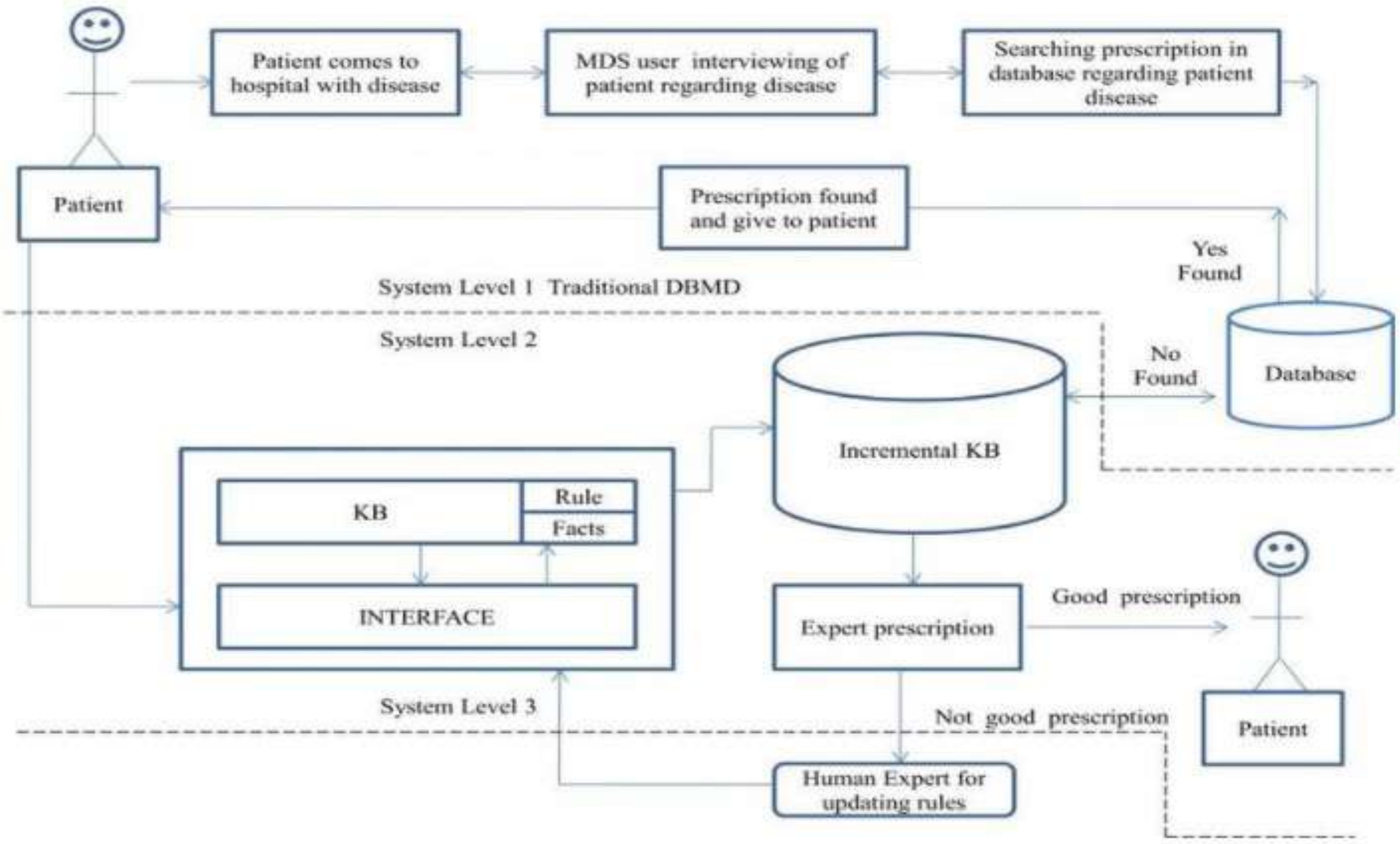
Few Examples of Soft Computing

- Person identification / Computer vision



Few Examples of Soft Computing

Medical diagnosis



Adoptive Medical Diagnosis System Using Expert System

Ref: Ansari, Gufran. (2013). Journal of Emerging Trends in Computing and Information Sciences An Adoptive Medical Diagnosis System Using Expert System with Applications. 4. 303-308.

Few Examples of Soft Computing

CHITKARA UNIVERSITY

- Person identification / Computer vision
- Hand written character recognition
- Weather forecasting
- Network optimization

Characteristics - Hard Computing

In 1996, LA Zadeh (LAZ) introduced the term hard computing. According to LAZ: We term a computing as ”Hard” computing, if

- **Precise solution is guaranteed**
- **Control action is unambiguous and accurate**
- **Control action is formally defined (i.e. with a mathematical model)**

Characteristics – Soft Computing

- **Provides approximate solution for real-life problems**
- **Process is not affected by any kind of change in the environment as it works on experimental data**
- **Does not require mathematical model to solve problem**

Soft computing vs hard computing

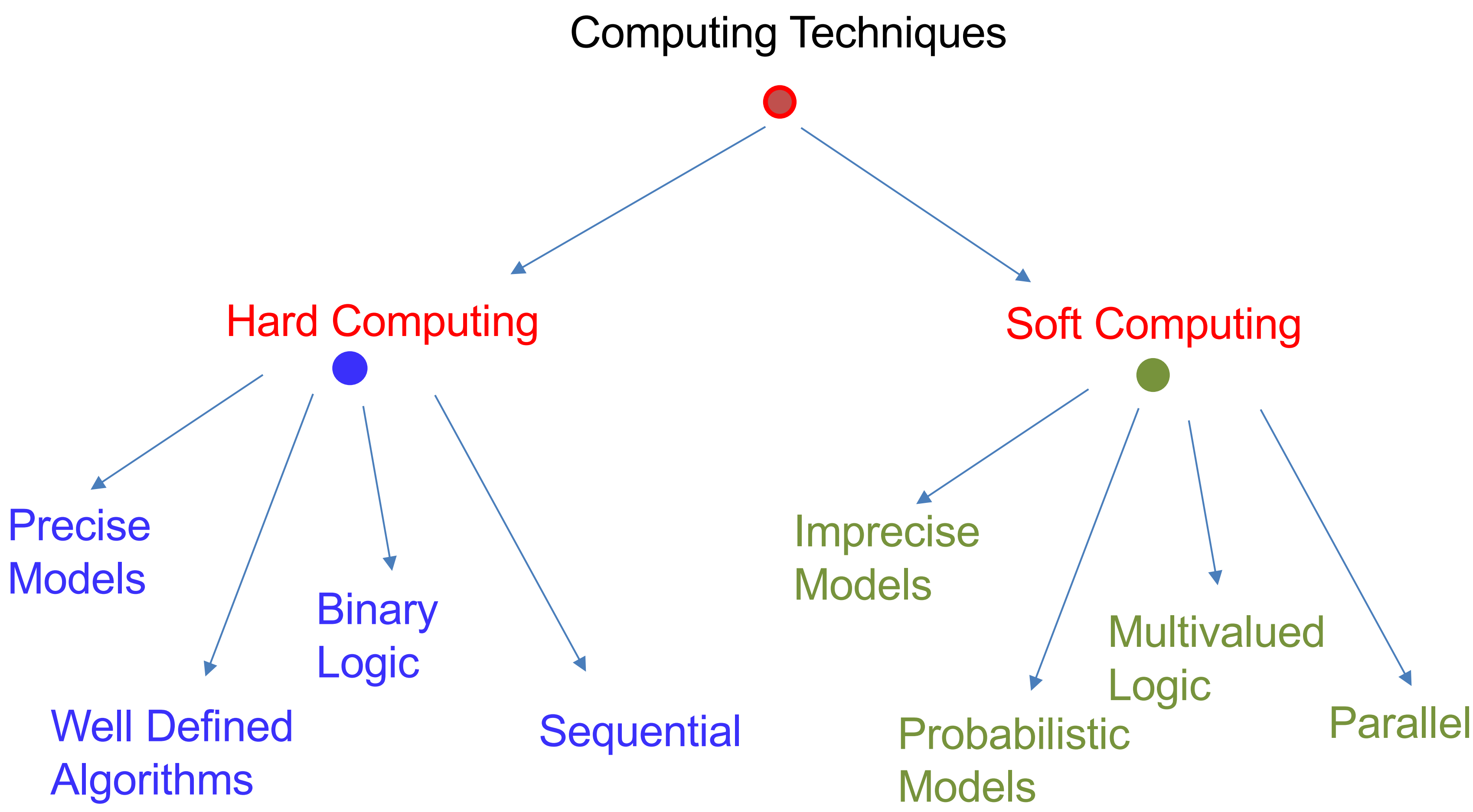
Parameters	Soft Computing	Hard Computing
Computation time	Takes less computation time.	Takes more computation time.
Dependency	It depends on approximation and dispositional.	It is mainly based on binary logic and numerical systems.
Computation type	Parallel computation	Sequential computation
Result/Output	Approximate result	Exact and precise result

Questions???

CHITKARA UNIVERSITY

- What are the major applications of AI?
- Identify major techniques of Soft Computing.

Summarizing Computing Techniques



Solutions through Soft Computing

Q: Is the suspect giving honest reply?

A: Hard Computing : Yes / No

Soft Computing : Extremely honest, honest, somewhat honest, somewhat dishonest, dishonest

Q: Will it rain tomorrow?

A: Hard Computing : Yes/ No

Soft Computing : 100% chance, 90% chance, 80% chance……

Q: Is soft computing an interesting course?

A: Hard Computing: Yes/ No

Soft Computing: Very interesting, interesting, somewhat interesting, difficult, very difficult

Need for soft computing

- Hard Computing fails to provide solutions for some real-life problems.
- When conventional mathematical and analytical models fail, soft computing helps
- Real-world problems do not have an ideal case, there exist a non-ideal environment
- Soft computing is not only limited to theory; it also gives insights into real-life problems
- Like all the above reasons, Soft computing helps to map the human mind, which cannot be possible with conventional mathematical and analytical models.

Soft Computing - Some insights

Zadeh coined the term of soft computing in 1992.

- The objective of soft computing is to provide precise approximation and quick solutions for complex real-life problems for which hard computing solution does not exist
- It refers to a group of computational techniques that are based on artificial intelligence (AI) and natural selection.
- The algorithms of soft computing are adaptive, so the current process is not affected by any kind of change in the environment.
- The concept of soft computing is based on **learning from experimental data**.
- It is based on Fuzzy logic, genetic algorithms, machine learning, Artificial Neural Networks, and expert systems.

Techniques in Soft Computing

- Fuzzy Logic
- Artificial Neural Network (ANN)
- Genetic Algorithms

- Associative Memory
- Adaptive Resonance Theory

Fuzzy Logic (FL)

- Fuzzy logic - a multivalued mathematical logic that allows intermediate values to be defined between conventional solutions with values 0 and 1.

- solves problems with an open and imprecise spectrum of data.

- makes it easy to obtain an array of precise conclusions.

- basically designed to achieve the best possible solution to complex problems from all the available information and input data.

Artificial Neural Network (ANN)

- Artificial neural network (ANN) - the concept inspired from human brain and the way the neurons in the human brain works.

- computational learning system that uses a network of functions to understand and translate a data input of one form into another form.

- contains large number of interconnected processing elements called as neuron.

- neurons operate in parallel and every neuron is connected with other neurons by a connection link.

Genetic Algorithms (GA)

- Genetic Algorithms - initiated and developed in the early 1970's by John Holland mimic some of the process of natural evolution.

- perform directed random search through a given set of alternative with the aim of finding the best alternative with respect to the given criteria of goodness.

- criteria is expressed in terms of an object function which is usually referred to as a fitness function.

- based on nature and take all inspirations from it.

Associative Memory

- Associative memory - a content-addressable structure that maps a set of input patterns to a set of output patterns.

- two types : auto- associative and hetero-associative

- **auto-associative memory** retrieves a previously stored pattern that most closely resembles the current pattern

- **hetero-associative memory**, the retrieved pattern is different from the input pattern not only in content but possibly also in type and format

Adaptive Resonance Theory

- Adaptive resonance theory - a type of neural network technique developed by Stephen Grossberg and Gail Carpenter in 1987.
- term **“adaptive”** and **“resonance”** used in this suggests that they are open to new learning without discarding the previous or the old information
- Stable and are flexible to gain new information
- ART networks implement a clustering algorithm

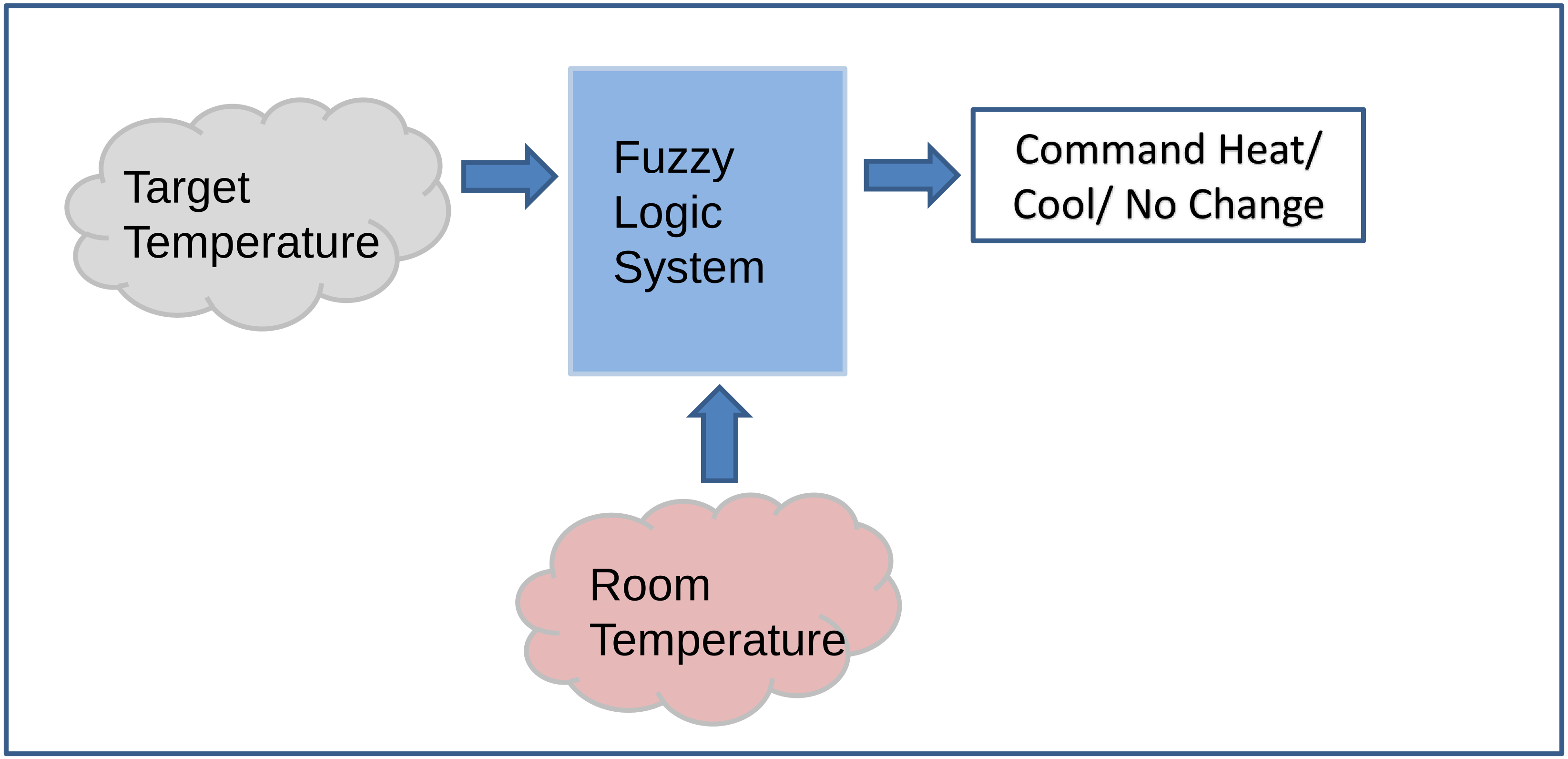
Questions??

List few examples of Fuzzy Logic, Artificial Neural Network and Genetic Algorithms

Identify real time problem that can be resolved through Soft Computing

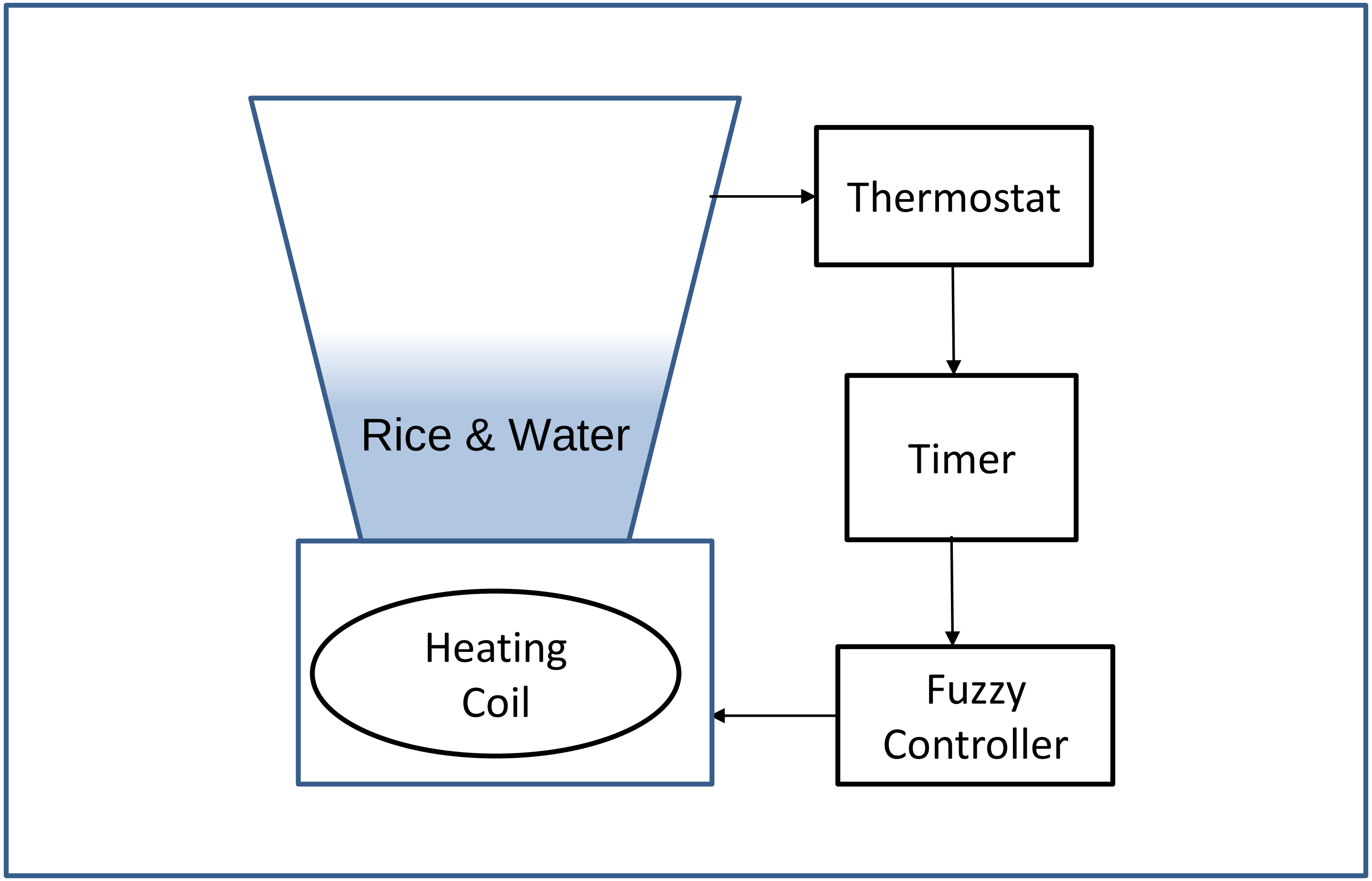
Fuzzy Logic (FL)

CHITKARA UNIVERSITY



Air Conditioning System

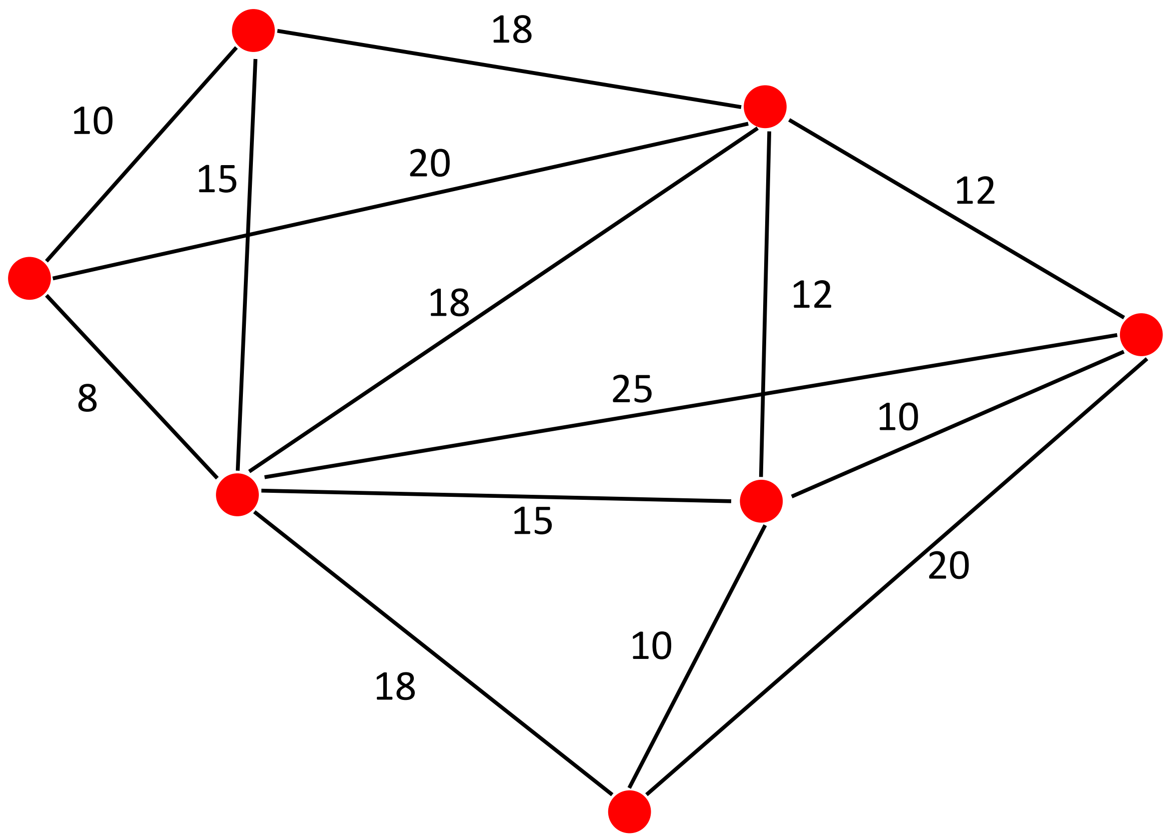
Fuzzy Logic (FL)



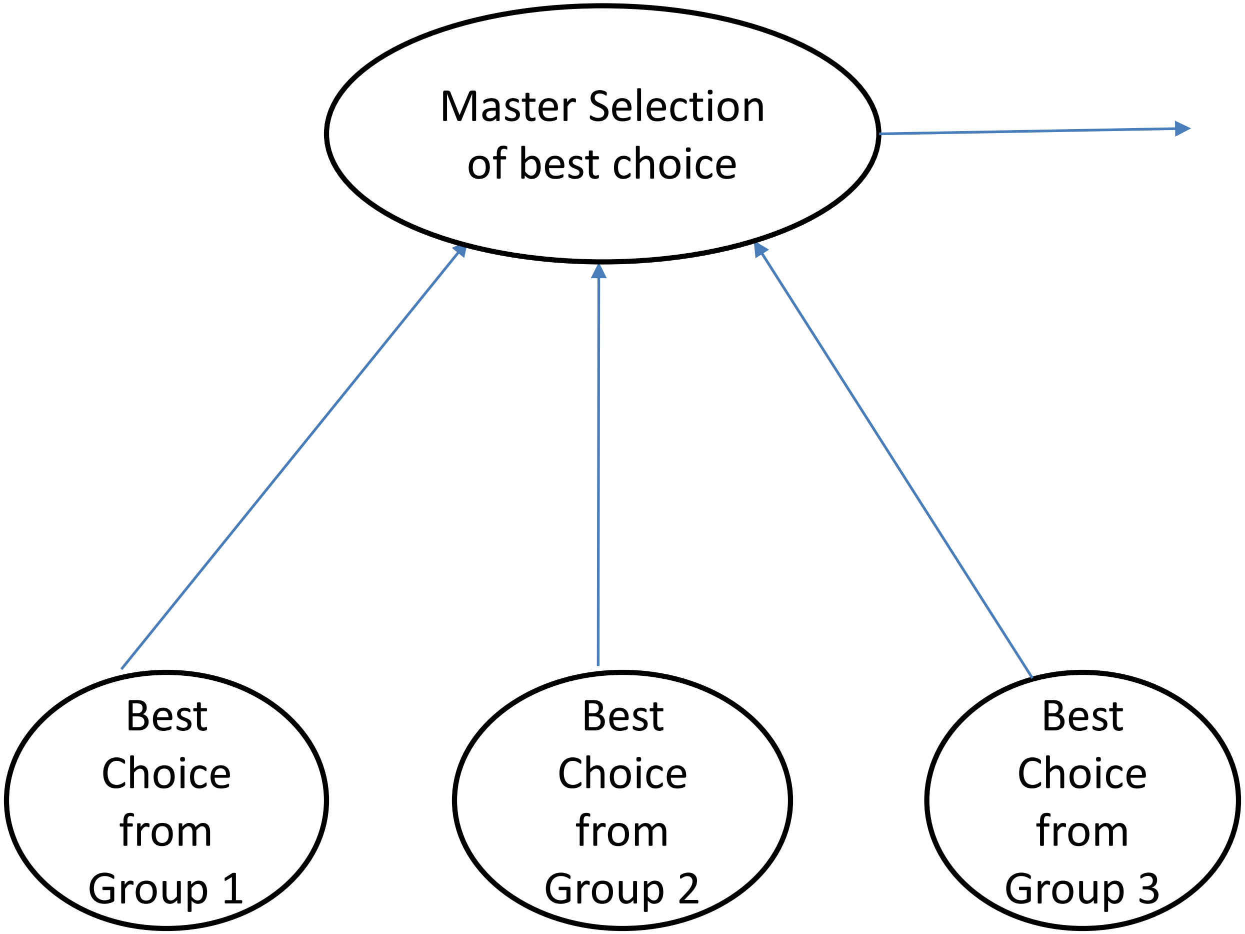
Rice Cooker

CHITKARA UNIVERSITY

Travelling Salesman Problem



Genetic Algorithms (GA)



Questions??

List few examples of Fuzzy Logic, Artificial Neural Network and Genetic Algorithms

Identify real time problem that can be resolved through Soft Computing

Fuzzy Logic Basics

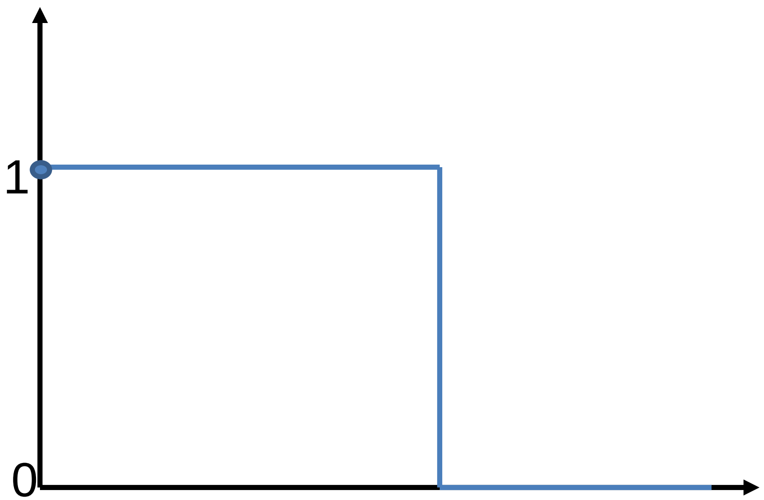
Fuzzy Logic

- Introduced by Prof. Lotfi A Zadeh in 1965.
- An organized method for dealing with imprecise data
- A multi-valued logic where values can be any real number between 0 & 1
- A mathematical language to express something.
- It deals with Fuzzy set .

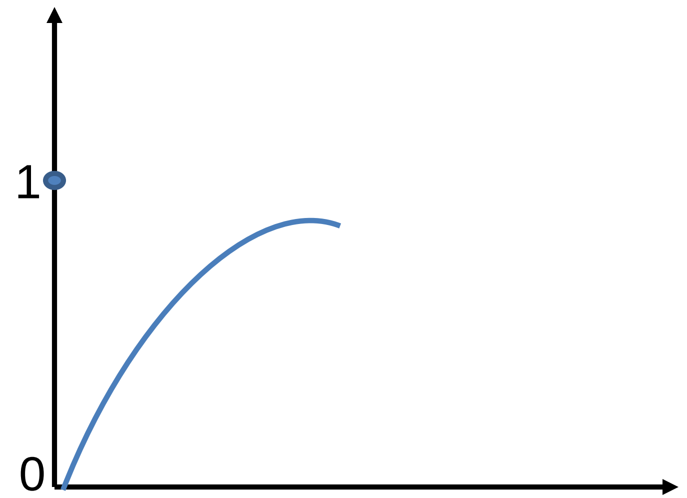
Fuzzy Logic vs. Crisp Logic

Crisp logic - a boolean logic (either 0 for false or 1 for true)

Fuzzy logic - captures the degree to which something is true.



Crisp Logic



Fuzzy Logic

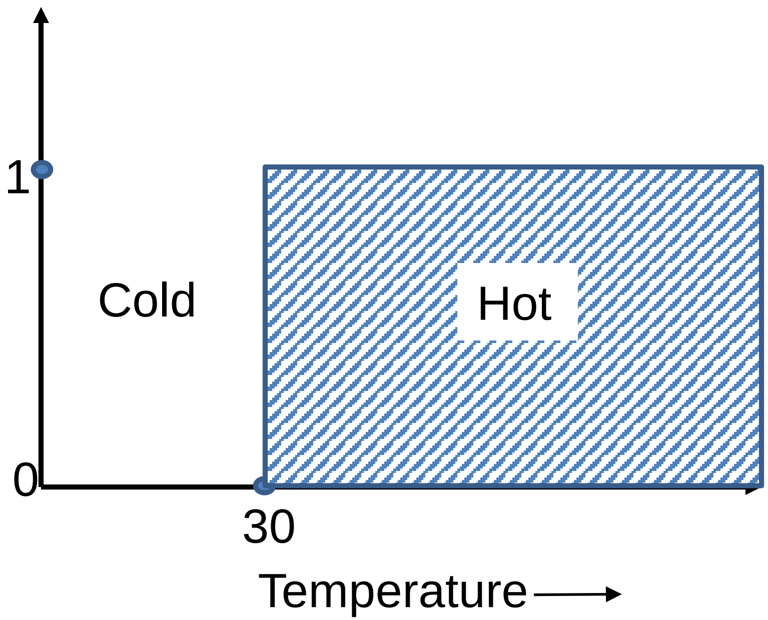
Examples

Rule : Temperature > 30^0C , Weather – Hot

Temperature < 30^0C , Weather – Cold

Crisp logic - a boolean logic (either 0 for cold or 1 for hot)

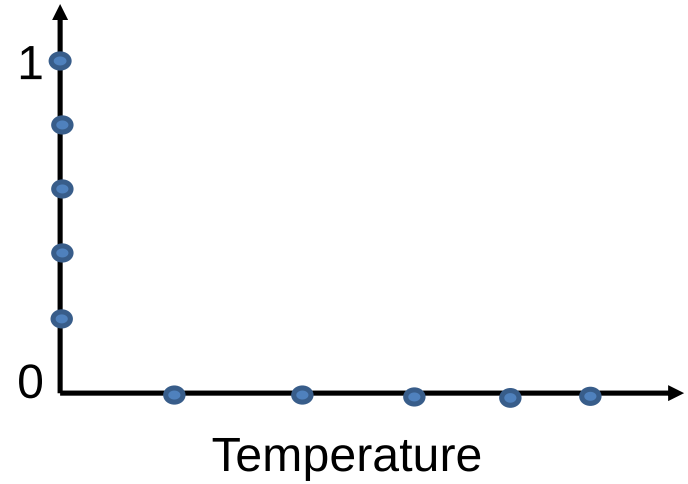
- Sharp boundaries, classical sets



Examples

Fuzzy Logic: Values from 0 to 1

- Unsharp boundaries, fuzzy set



Crisp (Classical) Sets

Classical or Crisp sets are useful in expressing classical or boolean logic

These sets:

- have collection of distinct objects.
- are binary
- have element either belonging to the set or not
- Have members or non-members

{True, false} {1, 0}

- N (set of positive integers) ={0,1,2,3,…….}
- Z (set of integers)={…,-2,-1,0,1,2,…}

Set – collection of objects sharing certain characteristics

Crisp (Classical) Sets

- X = Universe of discourse - set of all objects with the same characteristics.
- Let n_x = cardinal number - total number of elements in X.
- For crisp sets A in X, we define two cases:
  - $x \in A \rightarrow$ x belongs to A – x is a member of crisp set A
  - $x \notin A \rightarrow$ x does not belong to A – x is non-member of set A

Crisp (Classical) Sets

- The list of members of set A may be given in the following manner
  - List of members — A = {2,4,6,8,10}
  - Properties of set elements — A={x|x is even number <12}
  - Formula — A={x_i = i*2, where i= 1 to 5}
  - Logical Operation — A={x|x belongs to M OR Q}
  - Membership function

$$\mu_A(x) = \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{if } x \notin A \end{cases}$$

Crisp (Classical) Sets

- The set with no elements is empty set or null set
- Symbol for null set - $\varnothing$
- Set is null when occurrence of impossible event is denoted
- Occurrence of certain event indicates whole set

Relation of two classical sets

For sets A and B on X, the common notations are:

- $A \subset B \rightarrow$ if $x \in A$ then $x \in B$ (A is completely contained in B)
- $A \subseteq B \rightarrow$ A is contained in or is equivalent to B.
- $A = B \rightarrow A \subseteq B$ and $B \subseteq A$.

Operations on Crisp (Classical) Sets

- Union:
  - $A \cup B = \{x \mid x \in A \text{ or } x \in B\}$.
- Intersection:
  - $A \cap B = \{x \mid x \in A \text{ and } x \in B\}$.
- Complement:
  - $A^c = \{x \mid x \notin A, x \in X\}$.
- Difference (Subtraction)
  - $A|B$ or $(A\text{-}B) = \{x \mid x \in A \text{ and } x \notin B\}$

    $= A - (A \cap B)$

Properties of Crisp (Classical) Sets

- Commutativity
- Associativity
- Idempotency
- Transitivity
- Identity
- Involution
- Law of excluded middle
- Law of contradiction
- DeMorgan’s Law

Questions??

Ven Diagrams of Operations on Classical Sets

Examples of Properties of Classical Sets

Today’s Topic

CHITKARA UNIVERSITY

– Properties of Classical (Crisp) Sets
– Operations on Classical (Crisp) Sets
– Fuzzy Sets vs Crisp Sets

Crisp (Classical) Sets

- The set with no elements is empty set or null set
- Symbol for null set - $\varnothing$
- Set is null when occurrence of impossible event is denoted
- Occurrence of certain event indicates whole set

Operations on Crisp (Classical) Sets

- Union:
  - $A \cup B = \{x \mid x \in A \text{ or } x \in B\}$.
- Intersection:
  - $A \cap B = \{x \mid x \in A \text{ and } x \in B\}$.
- Complement:
  - $A^c = \{x \mid x \notin A, x \in X\}$.
- Difference (Subtraction)
  - $A|B$ or $(A\text{-}B) = \{x \mid x \in A \text{ and } x \notin B\}$

    $= A - (A \cap B)$

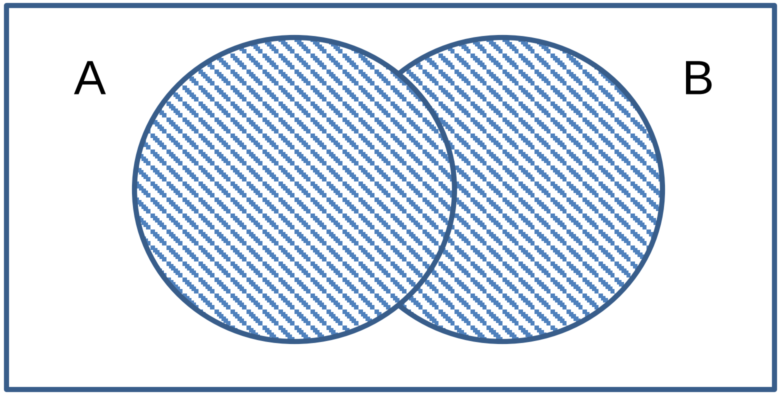
Operations on Crisp (Classical) Sets

- Union:
  - $A \cup B = \{x \mid x \in A \text{ or } x \in B\}$.

$A = \{1,2,3,4,5,6,7,8,9,10\}$

$B = \{2,4,6,8,10,12,14,16,18,20\}$

$A \cup B = \{1,2,3,4,5,6,7,8,9,10,12,14,16,18,20\}$



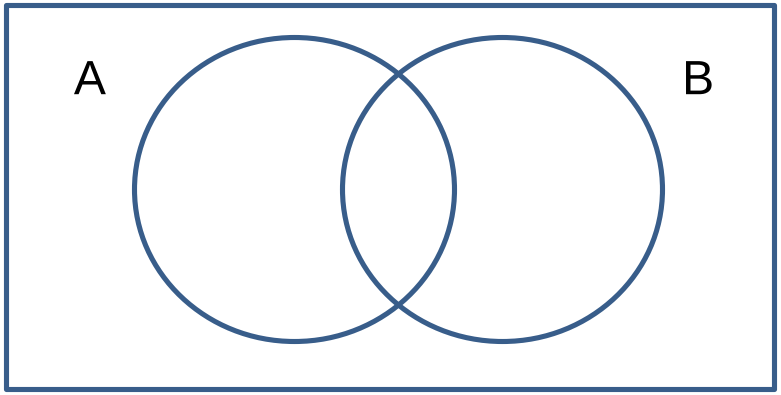
Operations on Crisp (Classical) Sets

- Intersection:
  - $A \cap B = \{x \mid x \in A \text{ and } x \in B\}$.

$A = \{1,2,3,4,5,6,7,8,9,10\}$

$B = \{2,4,6,8,10,12,14,16,18, 20\}$

$A \cap B = \{2,4,6,8,10\}$



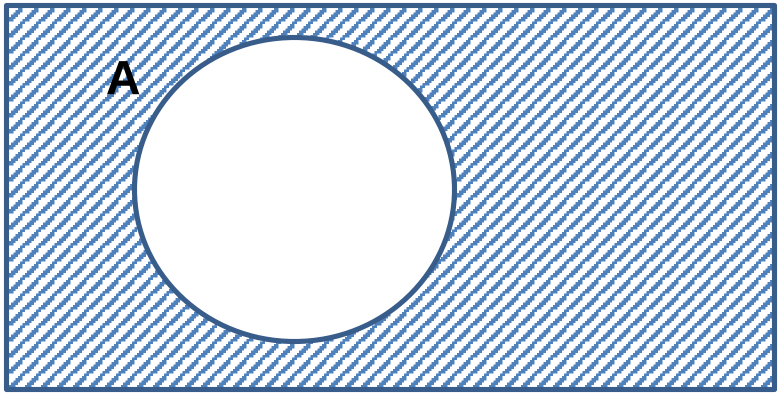
Operations on Crisp (Classical) Sets

- Complement:
  - $A^c = \{x \mid x \notin A, x \in X\}$.

A= {1,2,3,4,5,6,7,8,9,10};

A= {x | x is a natural number ≤ 10}

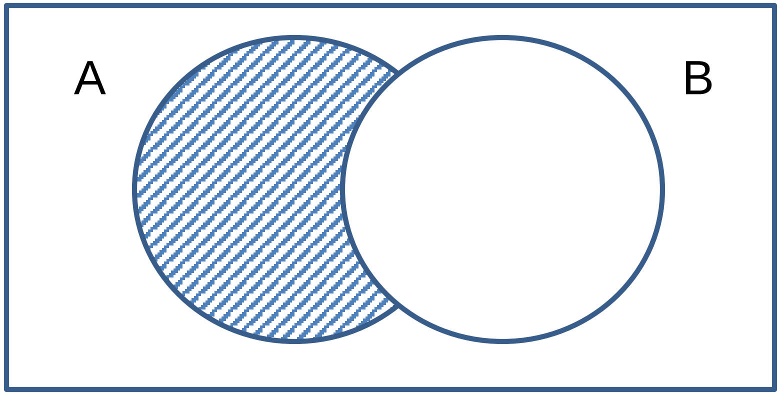
A^c = (x| x is a natural number > 10}



Operations on Crisp (Classical) Sets

- Difference (Subtraction)
  - $A|B$ or $(A-B) = \{x \mid x \in A \text{ and } x \notin B\}$

$$= A - (A \cap B)$$



Properties of Crisp (Classical) Sets

- Commutativity
- Associativity
- Distributivity
- Idempotency
- Transitivity
- Identity
- Involution
- Law of excluded middle
- Law of contradiction
- DeMorgan’s Law

Properties of Crisp (Classical) Sets

- Commutativity
  - $A \cup B = B \cup A$
  - $A \cap B = B \cap A$

- Associativity
  - $A \cup (B \cup C) = (A \cup B) \cup C$
  - $A \cap (B \cap C) = (A \cap B) \cap C$

- Distributivity
  - $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$
  - $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$

Properties of Crisp (Classical) Sets

- Idempotency
  - $A \cup A = A$
  - $A \cap A = A$

- Transitivity
  - If $A \subseteq B \subseteq C = A \subseteq B$

- Identity
  - $A \cup \varnothing = A, A \cap \varnothing = \varnothing$
  - $A \cup X = X, A \cap X = A$

Properties of Crisp (Classical) Sets

- Involution
  - $(A^c)^c = A$
- Law of Excluded Middle
  - $A \cup A^c = X$
- Law of Contradiction
  - $A \cap A^c = \varnothing$
- DeMorgan's Law
  - $\overline{A \cap B} = \overline{A} \cup \overline{B}$
  - $\overline{A \cup B} = \overline{A} \cap \overline{B}$

Mapping of Classical (Crisp) Sets

- X- Universe of Discourse
- Y- Universe of Discourse

$f : X \longrightarrow Y$

$$X_A(x) = \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{if } x \notin A \end{cases}$$

Where X_A is membership in set A for elements x in Universe

Mapping of Classical (Crisp) Sets

Consider two sets A & B on the Universe X

- Union of two sets in terms of function form
  - $A \cup B \longrightarrow X_{A \cup B}(x) = X_A(x) \vee X_B(x)$
    $= \max(X_A(x), X_B(x))$

    V indicates maximum operator

- Intersection of two sets in terms of function form
  - $A \cap B \longrightarrow X_{A \cap B}(x) = X_A(x) \wedge X_B(x)$
    $= \min(X_A(x), X_B(x))$

    $\wedge$ indicates minimum operator

Mapping of Classical (Crisp) Sets

- Complement of a set in terms of function form
  - A^c $\quad X_{A^c}(x) = 1 - X_A(x)$

- Containment of set in another in terms of function form
  - $A \subseteq B \longrightarrow \quad X_A(x) \leq X_B(x)$

Fuzzy set

- **Fuzzy sets** are sets whose elements have degrees of membership.
- In normal sets, membership is binary
  – An item is either in the set or not in the set
- In fuzzy sets, membership is based on a degree between 0 and 1
  – 0 if item not in set
  – 1 if item is in set
  – If degree is between 0 and 1, then this degree is the degree to which the item is thought to be in the set

Example of Fuzzy Set

Let X be the countries in the world

S = { (S, h) | $h \in X$ } and h(s) is a measurement of happiness quotient

S= All happy countries

S= {(Finland, 1), (Denmark, 0.9), (Switzerland, 0.9), (Iceland, 0.8)}

Crisp Set Vs. Fuzzy Set

- crisp set A of X is defined as function $f_A(x)$ called the characteristic function of A

$f_A(x) : X \in \{0, 1\}$, where

$$f_A(x) = \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{if } x \notin A \end{cases}$$

- fuzzy set A of universe X is defined by function $\mu_A(x)$ called the membership function of set A

$\mu_A(x) : X \in [0, 1]$, where $\mu_A(x) = 1$ if x is totally in A;
$\mu_A(x) = 0$ if x is not in A;
$0 < \mu_A(x) < 1$ if x is partly in A.

Crisp Set Vs. Fuzzy Set

Crisp Set	Fuzzy Set
$S = \{ s \mid s \in X \}$	1. $F = (s, \mu) \mid s \in X$ and $\mu(s)$ is the degree of membership.
It is a collection of elements.	It is collection of ordered pairs.
Inclusion of an element $s \in X$ into S has strict boundary Yes or No.	Inclusion of an element $s \in X$ into F is fuzzy, with a degree of Membership.

Questions??

- Can Crisp set be a fuzzy set or vice versa?
- How to find the power set and cardinality of given Set. Also find cardinality of Power Set?
- How is the degree of memberships decided?

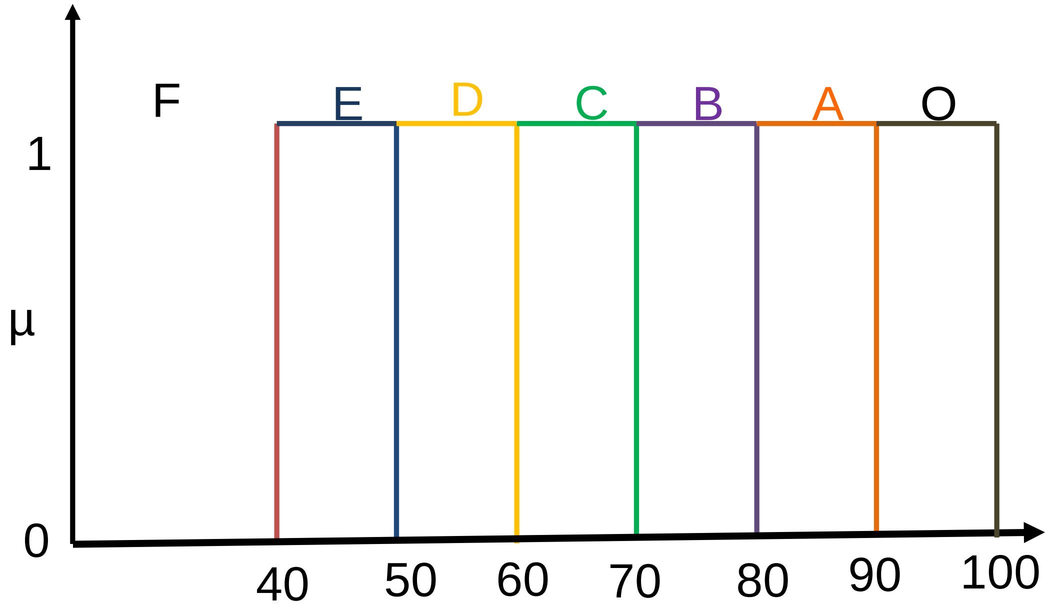
– Fuzzy Set Operations (Union, Intersection, Complement, Equality)

Grading Scheme in Crisp Way

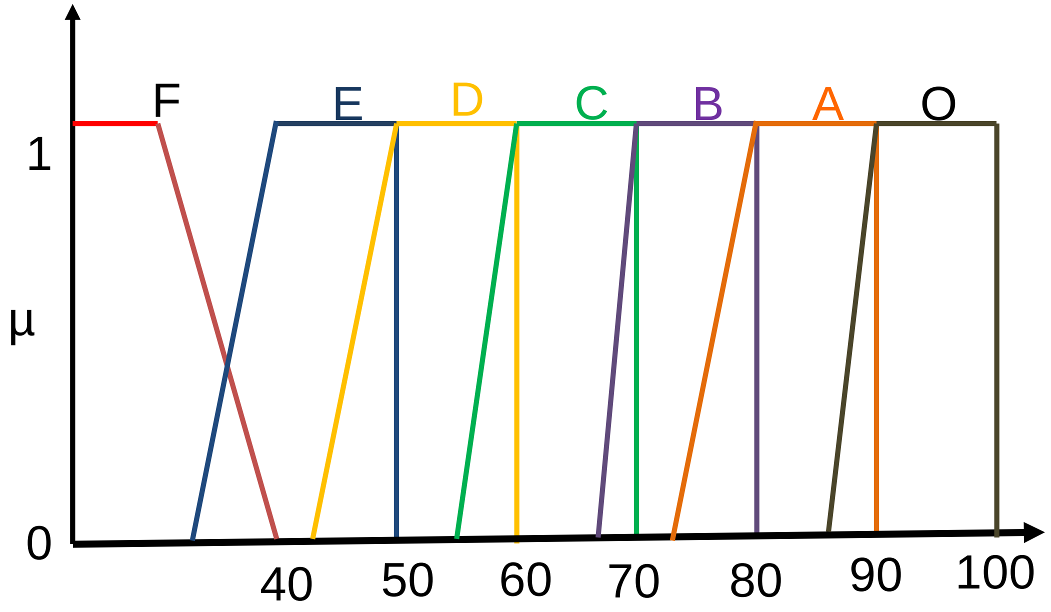
Marks are represented by m

$0 \leq m \leq 39$	Grade F
$40 \leq m \leq 49$	Grade E
$50 \leq m \leq 59$	Grade D
$60 \leq m \leq 69$	Grade C
$70 \leq m \leq 79$	Grade B
$80 \leq m \leq 89$	Grade A
$90 \leq m \leq 100$	Grade O

Crisp vs Fuzzy Representation of Grading Scheme



Crisp vs Fuzzy Representation of Grading Scheme



Membership function and Fuzzy set

If X is a Universe of discourse and $x \in X$, then a fuzzy set A in X is defined as a set of ordered pairs, that is

$A = \{(x, \mu_A(x)) | x \in X\}$ where $\mu_A(x)$ is called the membership function for the fuzzy set A.

**$\mu_A(x)$ map each element of X onto a membership value between 0 and 1 (both inclusive)

Fuzzy Set Representation

$$A = \left\{\frac{\mu_A(x_1)}{x_1} + \frac{\mu_A(x_2)}{x_2} + \frac{\mu_A(x_3)}{x_3} + \cdots\right\}$$

$$A = \left\{\sum_{i=1}^{n} \frac{\mu_A(x_i)}{x_i}\right\}$$

$A = \{(x, \mu_A(x)) \mid x \in X\}$ where μ_A(x) is called the membership function for the fuzzy set *A*.

**μ_A(x) map each element of *X* onto a membership value between 0 and 1 (both inclusive)

Operations on Fuzzy Sets

Consider two sets A & B on the Universe *X*

- Union of two sets in function form

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

Example:

$A = \{(x_1, 0.4), (x_2, 0.1), (x_3, 0.5), (x_4, 0.1)\}$

$B = \{(x_1, 0.2), (x_2, 0.6), (x_3, 0.3), (x_4, 0.2)\}$

$C = A \cup B = \{(x_1, 0.4), (x_2, 0.6), (x_3, 0.5), (x_4, 0.2)\}$

Operations on Fuzzy Sets

- Intersection of two sets in function form

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

Example:

$A = \{(x_1, 0.4), (x_2, 0.1), (x_3, 0.5), (x_4, 0.1)\}$

$B = \{(x_1, 0.2), (x_2, 0.6), (x_3, 0.3), (x_4, 0.2)\}$

$C = A \cap B = \{(x_1, 0.2), (x_2, 0.1), (x_3, 0.3), (x_4, 0.1)\}$

Operations on Fuzzy Sets

- Complement of a set in function form

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

Example:

$A = \{(x_1, 0.4), (x_2, 0.1), (x_3, 0.5), (x_4, 0.1)\}$

$C = A^C = \{(x_1, 0.6), (x_2, 0.9), (x_3, 0.5), (x_4, 0.9)\}$

- Algebraic product or Vector product (A•B)

$$\mu_{A \bullet B}(x) = \mu_A(x) \bullet \mu_B(x)$$

Example:

$A = \{(x_1, 0.4), (x_2, 0.1), (x_3, 0.5), (x_4, 0.1)\}$

$B = \{(x_1, 0.2), (x_2, 0.6), (x_3, 0.3), (x_4, 0.2)\}$

$A \bullet B = \{(x_1, 0.08), (x_2, 0.06), (x_3, 0.15), (x_4, 0.02)\}$

Operations on Fuzzy Sets

- Scalar product ($\alpha \times A$)

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

$A = \{(x_1, 0.4), (x_2, 0.1), (x_3, 0.5), (x_4, 0.1)\}$

$0.5*A = \{(x_1, 0.2), (x_2, 0.05), (x_3, 0.25), (x_4, 0.05)\}$

- Sum (A + B)

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

Example:

$A = \{(x_1, 0.4), (x_2, 0.1), (x_3, 0.5), (x_4, 0.1)\}$

$B = \{(x_1, 0.2), (x_2, 0.6), (x_3, 0.3), (x_4, 0.2)\}$

$A+B = \{(x_1, 0.52), (x_2, 0.64), (x_3, 0.65), (x_4, 0.28)\}$

Operations on Fuzzy Sets

- Difference ($A - B = A \cap B^C$)

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

Example:

$A = \{(x_1, 0.4), (x_2, 0.1), (x_3, 0.5), (x_4, 0.1)\}$

$B = \{(x_1, 0.2), (x_2, 0.6), (x_3, 0.3), (x_4, 0.2)\}$

$B^C = \{(x_1, 0.8), (x_2, 0.4), (x_3, 0.7), (x_4, 0.8)\}$

$A \cap B^C = \{(x_1, 0.4), (x_2, 0.1), (x_3, 0.5), (x_4, 0.1)\}$

Operations on Fuzzy Sets

- Disjunctive sum $A \oplus B = (A^C \cap B) \cup (A \cap B^C)$

Example:

$A = \{(x_1, 0.4), (x_2, 0.1), (x_3, 0.5), (x_4, 0.1)\}$

$B = \{(x_1, 0.2), (x_2, 0.6), (x_3, 0.3), (x_4, 0.2)\}$

$A^C = \{(x_1, 0.6), (x_2, 0.9), (x_3, 0.5), (x_4, 0.9)\}$

$B^C = \{(x_1, 0.8), (x_2, 0.4), (x_3, 0.7), (x_4, 0.8)\}$

$A^C \cap B = \{(x_1, 0.2), (x_2, 0.6), (x_3, 0.3), (x_4, 0.2)\}$

$A \cap B^C = \{(x_1, 0.4), (x_2, 0.1), (x_3, 0.5), (x_4, 0.1)\}$

$(A^C \cap B) \cup (A \cap B^C) = \{(x_1, 0.4), (x_2, 0.6), (x_3, 0.5), (x_4, 0.2)\}$

- Bounded Sum $| A(x) \oplus B(x) |$

$$\mu_{|A(x)\oplus B(x)|} = \min\{1, \mu_A(x) + \mu_B(x)\}$$

Example:

$A = \{(x_1, 0.4), (x_2, 0.5), (x_3, 0.5), (x_4, 0.1)\}$

$B = \{(x_1, 0.2), (x_2, 0.6), (x_3, 0.3), (x_4, 0.2)\}$

$\mu_A(x) + \mu_B(x) = \{(x_1, 0.6), (x_2, 1.1), (x_3, 0.8), (x_4, 0.3)\}$

$\min\{1, \mu_A(x) + \mu_B(x)\} = \{(x_1, 0.6), (x_2, 1), (x_3, 0.8), (x_4, 0.3)\}$

- Bounded Difference $| A(x) \odot B(x) |$

$$\mu_{|A(x) \odot B(x)|} = \max\{0, \mu_A(x) - \mu_B(x)\}$$

Example:

$A = \{(x_1, 0.4), (x_2, 0.1), (x_3, 0.5), (x_4, 0.1)\}$

$B = \{(x_1, 0.2), (x_2, 0.6), (x_3, 0.3), (x_4, 0.2)\}$

$\mu_A(x) - \mu_B(x) = \{(x_1, 0.2), (x_2, -0.5), (x_3, 0.2), (x_4, -0.1)\}$

$\max\{0, \mu_A(x) - \mu_B(x)\} = \{(x_1, 0.2), (x_2, 0), (x_3, 0.2), (x_4, 0)\}$

Operations on Fuzzy Sets

Eg: Consider two fuzzy sets

$$A = \left\{ \frac{1}{2} + \frac{0.4}{2} + \frac{0.5}{2} + \frac{0.3}{2} \right\}$$

$$B = \left\{ \frac{0.5}{2} + \frac{0.3}{2} + \frac{0.2}{2} + \frac{1}{2} \right\}$$

Find union, intersection, complement and difference

Operations on Fuzzy Sets

Eg: Compare the two sensors based upon their detection levels and gain settings the values are provided in the table below

Gain Setting	Detection Level Sensor I (A)	Detection Level Sensor II (B)
0	0	0
10	0.1	0.25
20	0.4	0.35
30	0.65	0.7
40	0.75	0.85
50	1	1

Operations on Fuzzy Sets

Find the following membership functions

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

Operations on Fuzzy Sets

Equality ($A = B$): $\mu_A(x) = \mu_B(x)$

Power of a fuzzy set A^{α}: $\mu_A{}^{\alpha}(x) = \{\mu_A(x)\}^{\alpha}$

If $\alpha < 1$, then it is called *dilation*

If $\alpha > 1$, then it is called *concentration*

Cartesian Product : A(x) X B(y) = $\mu_{(AXB)}(x,y) = \min\{\mu_A(x), \mu_B(y)\}$

A(x) and B(y) are defined on the Universal sets X and Y, respectively.

Cartesian Product of Fuzzy Sets Cont..

Eg: $A(x)=\{(x_1,0.2),(x_2,0.3),(x_3,0.5),(x_4,0.6)\}$

$B(y)=\{(y_1,0.6),(y_2,0.1),(y_3,0.3)\}$

$\min\{\mu_A(x_1),\mu_B(y_1)\} = \min\{0.2,0.6\} = 0.2$ $\min\{\mu_A(x_1),\mu_B(y_2)\} = \min\{0.2,0.1\} = 0.1$
$\min\{\mu_A(x_1),\mu_B(y_3)\} = \min\{0.2,0.3\} = 0.2$

$\min\{\mu_A(x_2),\mu_B(y_1)\} = \min\{0.3,0.6\} = 0.3$ $\min\{\mu_A(x_2),\mu_B(y_2)\} = \min\{0.3,0.1\} = 0.1$
$\min\{\mu_A(x_2),\mu_B(y_3)\} = \min\{0.3,0.3\} = 0.3$

$\min\{\mu_A(x_3),\mu_B(y_1)\} = \min\{0.5,0.6\} = 0.5$ $\min\{\mu_A(x_3),\mu_B(y_2)\} = \min\{0.5,0.1\} = 0.1$
$\min\{\mu_A(x_3),\mu_B(y_3)\} = \min\{0.5,0.3\} = 0.3$

$\min\{\mu_A(x_4),\mu_B(y_1)\} = \min\{0.6,0.6\} = 0.6$ $\min\{\mu_A(x_4),\mu_B(y_2)\} = \min\{0.6,0.1\} = 0.1$
$\min\{\mu_A(x_4),\mu_B(y_3)\} = \min\{0.6,0.3\} = 0.3$

$$AxB = \begin{bmatrix} 0.2 & 0.1 & 0.2 \\ 0.3 & 0.1 & 0.3 \\ 0.5 & 0.1 & 0.3 \\ 0.6 & 0.1 & 0.3 \end{bmatrix}$$

Properties of Fuzzy Sets

Fuzzy set follows the same properties as of crisp sets

- Commutativity
- Associativity
- Distributivity
- Idempotency
- Transitivity
- Identity
- Involution
- DeMorgan's Law

Cardinality of Fuzzy Set

The cardinality of the fuzzy set is sum of memberships values of fuzzy set A

$$|A| = \sum_{i=1}^{n} \frac{\mu_A(x_i)}{x_i}$$

Relative Cardinality = |A|/ No. of ordered pairs in the fuzzy set

Classical/ Crisp Relations

- An ordered sequence of r elements in the form (a_1, $a_{2,}$ a_3, $a_{4........}$ $a_{\,r}$) is an ordered r-tuple.
- An unordered r-tuple is collection of r elements without restrictions on order
- For crisp sets A_1, $A_{2,}$ A_3, $A_{4........}$ $A_{\,r,}$ the set of all r-tuples (a_1, $a_{2,}$ a_3, $a_{4........}$ $a_{\,r}$), where $a_1 \in A_{1,}$ $a_2 \in A_2$, $a_3 \in A_3$,…… $a_r \in A_r$, is called Cartesian product of A_1, $A_{2,}$ A_3, $A_{4........}$ A_r and ids denoted by A_1x A_2 x A_3 x $A_{4........}$ X $A_{\,r}$. It is not same as arithmetic product.
- An r-ary relation over A_1, $A_{2,}$ A_3, $A_{4........}$ $A_{\,r,}$ is a subset of Cartesian product.
- When r =2, the cartesian product is A_1x A_2 which is a binary relation

Classical/ Crisp Relations

Eg: $A=\{0,1\}$

$B=\{a,b,c\}$

$A \times B = \{(0,a), (0,b), (0,c), (1,a), (1,b), (1,c)\}$

The Cartesian product of two universes X and Y is determined as $X \times Y = \{(x, y) \mid x \in X, y \in Y\}$ which forms an ordered pair of every $x \in X$ with every $y \in Y$.

That is, every element in universe X is related completely to every element in universe Y.

Classical/ Crisp Relations

The strength of this relationship between ordered pairs of elements in each universe is measured by the characteristic function, denoted by χ.

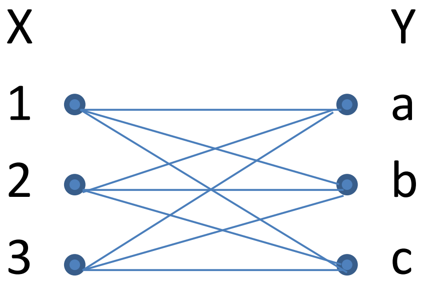
χ is 1 in case of complete relationship and 0 in case of no relationship.

$$\chi_{X \times Y}(x,y) = \begin{cases} 1 \text{ if } (x,y) \in X \times Y \\ 0 \text{ if } (x,y) \notin X \times Y \end{cases}$$

Eg: A={ 1,2,3}

B= {a,b,c}

Strength of relationship is



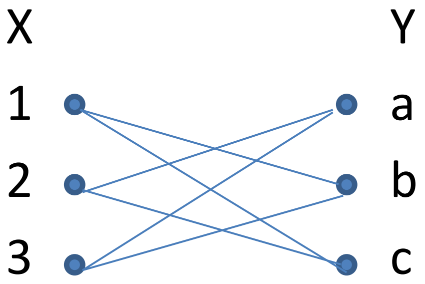
$$R = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{ccc} a & b & c \end{array} \\ \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \end{array}$$

Classical/ Crisp Relations

Eg: A={ 1,2,3}

B= {a,b,c}

Constrained Binary Relationship



$$
R = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array}
\begin{array}{c} \begin{array}{ccc} a & b & c \end{array} \\ \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \end{array}
$$

$R \subseteq A \times B$

Identity Relation and Universal Relation

Special cases of the constrained and the unconstrained Cartesian product for sets where r = 2 (i.e., for A_2) are called the identity relation and the universal relation.

Eg: A={ 1,2,3}

B= {a,b,c}

U_A= {(1,a), (1,b), (1,c), (2,a), (2,b), (2,c), (3,a), (3,b), (3,c)}

I_A= {(1,a), (2,b), (3,c)}

Cardinality of Crisp Relations

- Suppose n elements of the universe X are related to m elements of the universe Y.
- If the cardinality of X is n_X and the cardinality of Y is n_Y,

  then the cardinality of the relation, R, between these two universes is

  $$n_{X \times Y} = n_X * n_Y.$$

- The cardinality of the power set describing this relation, P (X × Y)

  is then $n_{P(X \times Y)} = 2^{n_X * n_Y}$.

Operations on Crisp Relations

- R and S are defined as two relations on Cartesian Universe X x Y
- O and E are null and complete relation respectively, Eg. of 3 x 3 set is :

$$O = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad E = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

The following operations are defined on R and S

Union	$R \cup S \longrightarrow \chi_{R\cup S}(x,y) : \chi_{R\cup S}(x,y) = \max[\chi_R(x,y), \chi_S(x,y)]$
Intersection	$R \cap S \longrightarrow \chi_{R\cap S}(x,y) : \chi_{R\cap S}(x,y) = \min[\chi_R(x,y), \chi_S(x,y)]$
Complement	$R^C = \chi_{R^c}(x,y) : \chi_{R^c}(x,y) = 1 - \chi_R(x,y)$
Containment	$R \subset S \longrightarrow \chi_R(x,y) : \chi_R(x,y) \le \chi_s(x,y)$
Identity	$\varnothing \longrightarrow$ O and X $\longrightarrow$ E

Properties of Crisp Relations

- The properties :
- Commutativity
- Associativity
- Distributivity
- Involution
- Idempotency
- De Morgan’s principles

Composition

- Let R be a relation that relates elements from universe X to universe Y, and S from universe Y to universe Z.
  - R = {(x1, y1), (x1, y3), (x2, y4)}
  - S = {(y1, z2), (y3, z2)}

- The max–min composition is defined by the set theoretic and membership function-theoretic expressions

  T= RoS

  $$\chi_T(x,y) = \vee_{y \in Y} (\chi_R(x, y) \wedge \chi_S(y, z))$$

•max–product (sometimes called max–dot) composition

$$\chi_T(x,y) = \vee_{y \in Y} (\chi_R(x, y) \bullet \chi_S(y, z))$$

Fuzzy Relations

- X – Universe of discourse
- Y – Universe of Discourse
- Fuzzy Relation maps X and Y through cartesian product
- Strength of relation is measured with membership function
- fuzzy relation is a mapping from the Cartesian space $X \times Y$ to the interval [0,1],

Fuzzy Relation Cont..

- Fuzzy Relation between sets X and Y is known as binary fuzzy relation and is denoted by

  R(X, Y).It is referred to as bipartite graph when $X \neq Y$.

- The binary relation on single set X is called directed graph or digraph when X= Y. It is denoted as R (X,X) or R(X^2)

Fuzzy Relation Cont..

Let X = {x_1, x_2, x_3, x_4........ x_n)

Let Y = {y_1, y_2, y_3, y_4........ y_m)

The relation R(X,Y) is represented as fuzzy matrix

$$R(X,Y) = \begin{bmatrix} \mu_R(x_1,y_1) & \mu_R(x_1,y_2) & \ldots\ldots \mu_R(x_1,y_m) \\ \mu_R(x_2,y_1) & \mu_R(x_2,y_2) & \ldots\ldots \mu_R(x_2,y_m) \\ & . & \\ & . & \\ & . & \\ \mu_R(x_n,y_1) & \mu_R(x_n,y_2) & \ldots\ldots \mu_R(x_n,y_m) \end{bmatrix}$$

Example of Fuzzy Relation

- Consider universe X = {{x_1, $x_{2,}$ x_3, x_4}, the binary fuzzy relation on X

$$R(X,X) = \begin{bmatrix} 0.2 & 0.0 & 0.4 & 0 \\ 0.0 & 0.3 & 0.6 & 0 \\ 0.1 & 0.0 & 0.5 & 0 \\ 0.0 & 0.6 & 0.0 & 1 \end{bmatrix}$$

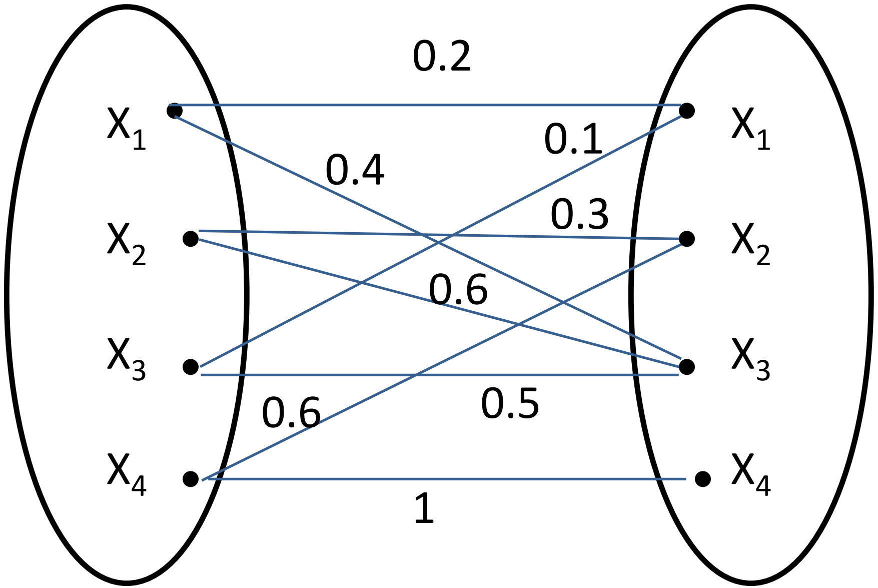
Two types of Graphical representations

- Bipartite Graph
- Simple Fuzzy Graph

Example of Fuzzy Relation Cont..

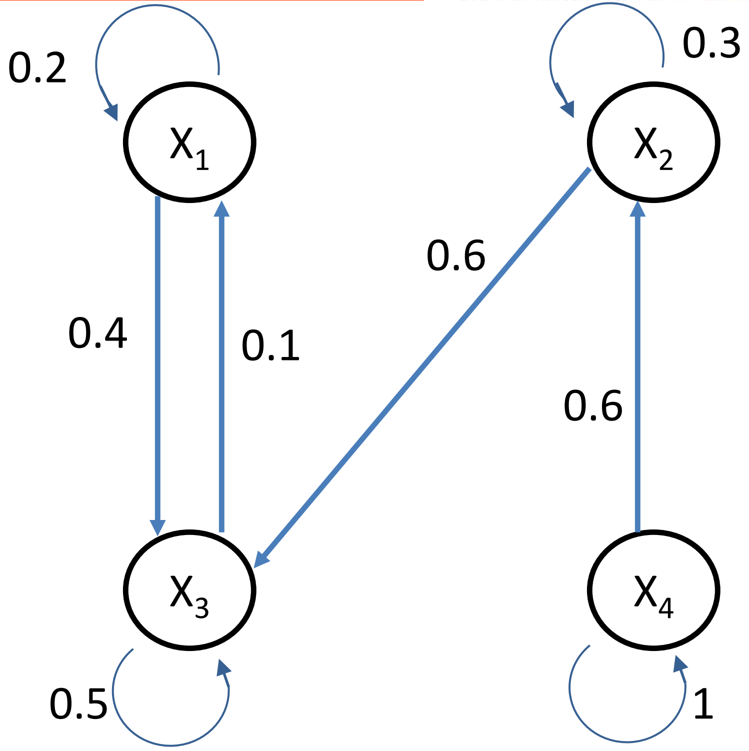
$$R(X,X) = \begin{bmatrix} 0.2 & 0.0 & 0.4 & 0 \\ 0.0 & 0.3 & 0.6 & 0 \\ 0.1 & 0.0 & 0.5 & 0 \\ 0.0 & 0.6 & 0.0 & 1 \end{bmatrix}$$

Simple Fuzzy Graph



Bipartite Graph

$$R(X,X) = \begin{bmatrix} 0.2 & 0.0 & 0.4 & 0 \\ 0.0 & 0.3 & 0.6 & 0 \\ 0.1 & 0.0 & 0.5 & 0 \\ 0.0 & 0.6 & 0.0 & 1 \end{bmatrix}$$



Simple Fuzzy Graph

Today’s Topics

– Fuzzy relations, Membership functions, Fuzzy If-then Rules

Classical Tolerance and Equivalence Relations

Crisp/ Classical Equivalence Relation:

Let relation R on Universe X is relation from X to X. Relation R will be equivalent relation if it follows

- Reflexivity
- Symmetry
- Transitivity

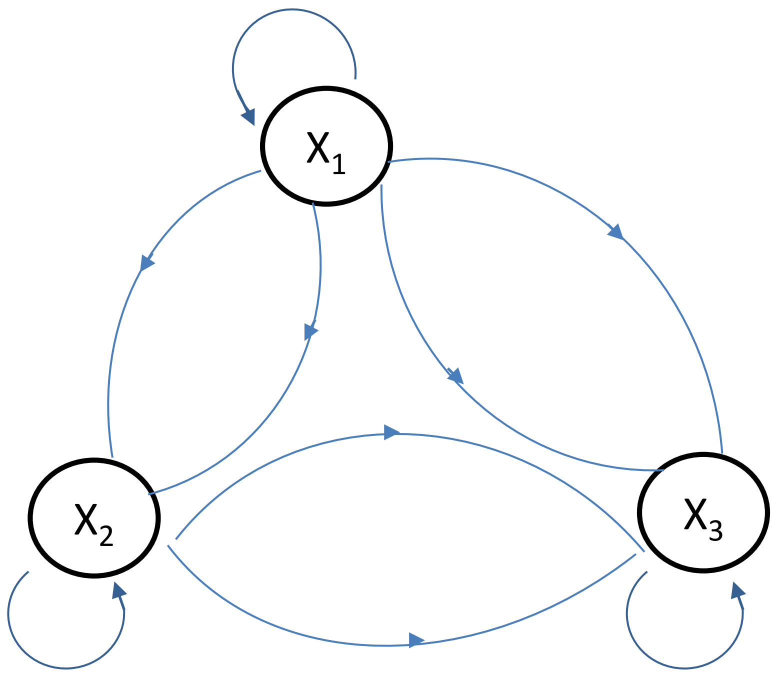
Reflexivity: $X_R(x_i, x_i) = 1$ or $(x_i, x_i) \in R$

Symmetry: $X_R(x_i, x_j) = X_R(x_j, x_i)$; $(x_i, x_j) \in R \Longrightarrow (x_j, x_i) \in R$

Transitivity: $X_R(x_i, x_j)$ and $X_R(x_j, x_k) = 1$, then $X_R(x_i, x_k) = 1$

$(x_i, x_j) \in R$, $(x_j, x_k) \in R$, then $(x_i, x_k) \in R$

Crisp/ Classical Equivalence Relation:



Classical Tolerance and Equivalence Relations....

Crisp/ Classical Tolerance Relation:

Relation R_1 on Universe X is tolerance relation also known as proximity relation if the following two properties are satisfied:

- Reflexivity
- Symmetry

Reflexivity: $X_R(x_i, x_i) = 1$ or $(x_i, x_i) \in R$

Symmetry: $X_R(x_i, x_j) = X_R(x_j, x_i)$; $(x_i, x_j) \in R \Longrightarrow (x_j, x_i) \in R$

Fuzzy Equivalence Relation

Let relation R be fuzzy relation on Universe X mapping elements from X to X. Relation R will be equivalent relation if the following properties are satisfied

- Reflexivity
- Symmetry
- Transitivity

Reflexivity: $\mu_R(x_i, x_i) = 1 \ \forall\ x \in X$

Symmetry: $\mu_R(x_i, x_j) = \mu_R(x_j, x_i)$ for all $(x_i, x_j) \in X$

Transitivity: $\mu_R(x_i, x_j) = \lambda_1$ and $\mu_R(x_j, x_k) = \lambda_2$, then $\mu_R(x_i, x_k) = \lambda$ where $\lambda = \min[\lambda_1, \lambda_2]$

$$\mu_R(x_i, x_j) \geq \max_{x_j \in X} \min[\mu_R(xi, x_j) \cdot \mu_R(x_j, x_k)] \ \forall\ (x_i, x_k) \in X^2$$

Fuzzy Tolerance Relation

Let relation R be fuzzy relation on Universe X mapping elements from X to X. Relation R will be tolerance relation if the following properties are satisfied

- Reflexivity
- Symmetry

Reflexivity: $\mu_R(x_i, x_i) = 1 \ \forall \ x \in X$

Symmetry: $\mu_R(x_i, x_j) = \mu_R(x_j, x_i)$ for all $(x_i, x_j) \in X$

•

Membership Functions

Membership Functions

1. defines fuzziness in the fuzzy set
2. generally represented in a graphical form
3. Multiple ways to characterize fuzziness

Features of Membership Functions

Fuzzy set : A= {(x, μ_A(x)) | x ∈ X}

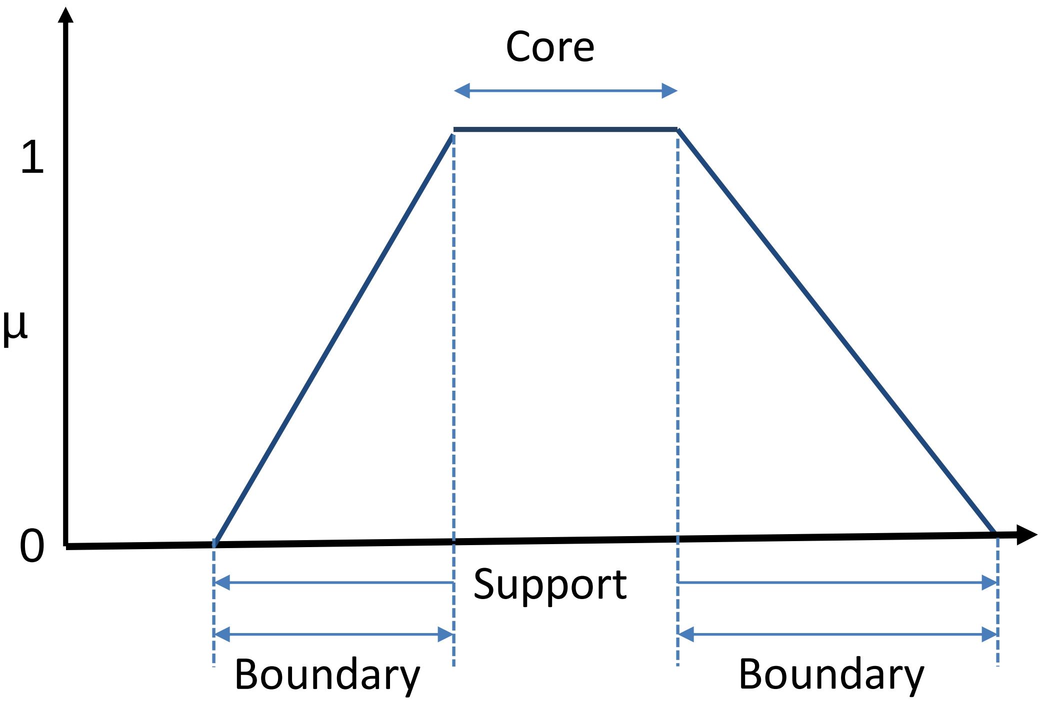
μ_A(.) is a membership function mapping X to membership space M.

Membership value ranges in [0,1]

Features:

- Core: Region of universe characterized by complete membership in set A; $\mu_A(x) = 1$
- Support: Region of universe characterized by nonzero membership in set A; $\mu_A(x) > 0$
- Boundary: Region of universe characterized by nonzero membership but not complete membership in set A; $0 < \mu_A(x) < 1$

Graphical Representation of Features of Membership Functions



Types of Fuzzy Sets

- Three Major types: Core, Support & Boundary
- Other types:
  - Normal Fuzzy Set : membership function has atleast one element with membership value as unity
  - Convex Fuzzy Set : membership function values strictly monotonically increasing or decreasing
  - Nonconvex fuzzy Set : membership function values are not strictly monotonically increasing or decreasing

Types of Fuzzy Sets

- The element in the universe for which a particular fuzzy set has value equal to 0.5 is called crossover point.
- There can be more than one crossover point.
- The maximum value of membership function in a fuzzy set is called the height of the fuzzy set.
- Normal fuzzy set has height as 1.
- Fuzzy set with height less than 1 is subnormal fuzzy set.

Membership Value Assignments

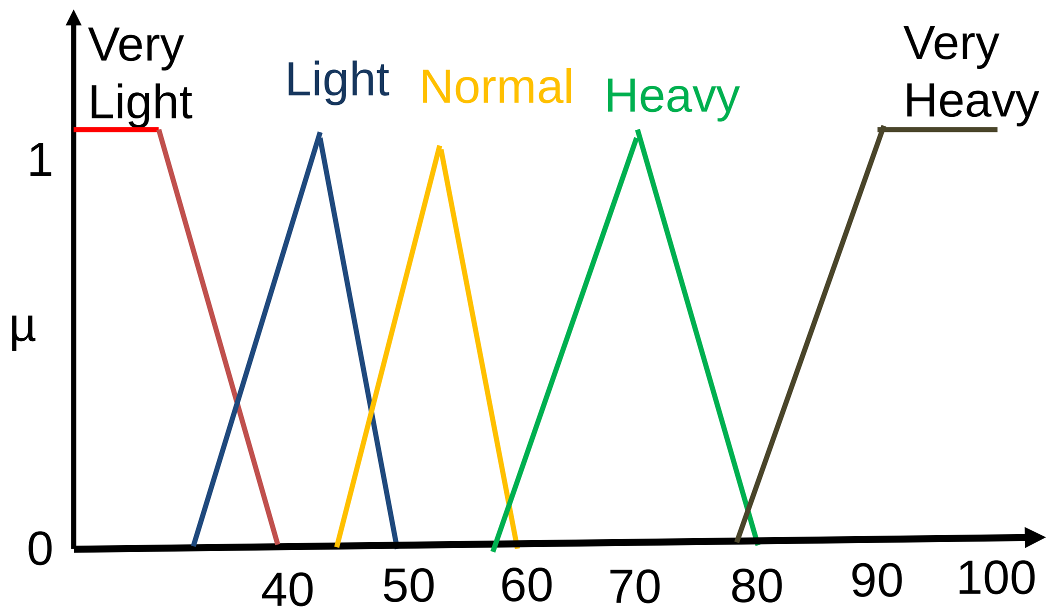
Methods for assigning membership value:

- Intuition
- Inference
- Rank Ordering
- Angular Fuzzy Sets
- Neural Networks
- Genetic Algorithms
- Inductive Reasoning

Membership Value Assignments

- Intuition : Developing membership functions on the basis of human intelligence

Eg. Categorizing people on basis of weights



Membership Value Assignments

- Inference : deductive reasoning by means of forward inference

Various shapes triangular, trapezoidal, bell shaped etc. can define membership function.

Eg: Inference method based on triangular shape

Let U be universe of Triangles and X, Y, Z are angles

$X \geq Y \geq Z \geq 0$; U= $\{(X,Y,Z) \mid X \geq Y \geq Z \geq 0;\ X+Y+Z=180\}$

Let I= isosceles triangle

E=equilateral triangle

R= right-angled triangle

IR = isosceles right-angled triangle

T = Other Triangles

Membership Value Assignments

- **Inference :** membership values of isosceles triangles can be

  $\mu_I(X,Y,Z) = 1\text{-}1/60^0 \min(X\text{-}Y, Y\text{-}Z)$

  If X=Y or Y=Z, the membership function value is 1

  If $X=120^0$, $Y=60^0$, $Z=0^0$, membership function value is 0

membership values of right-angle triangle can be

$\mu_R(X,Y,Z) = 1\text{-}1/90^0 \, |X\text{-}90|$

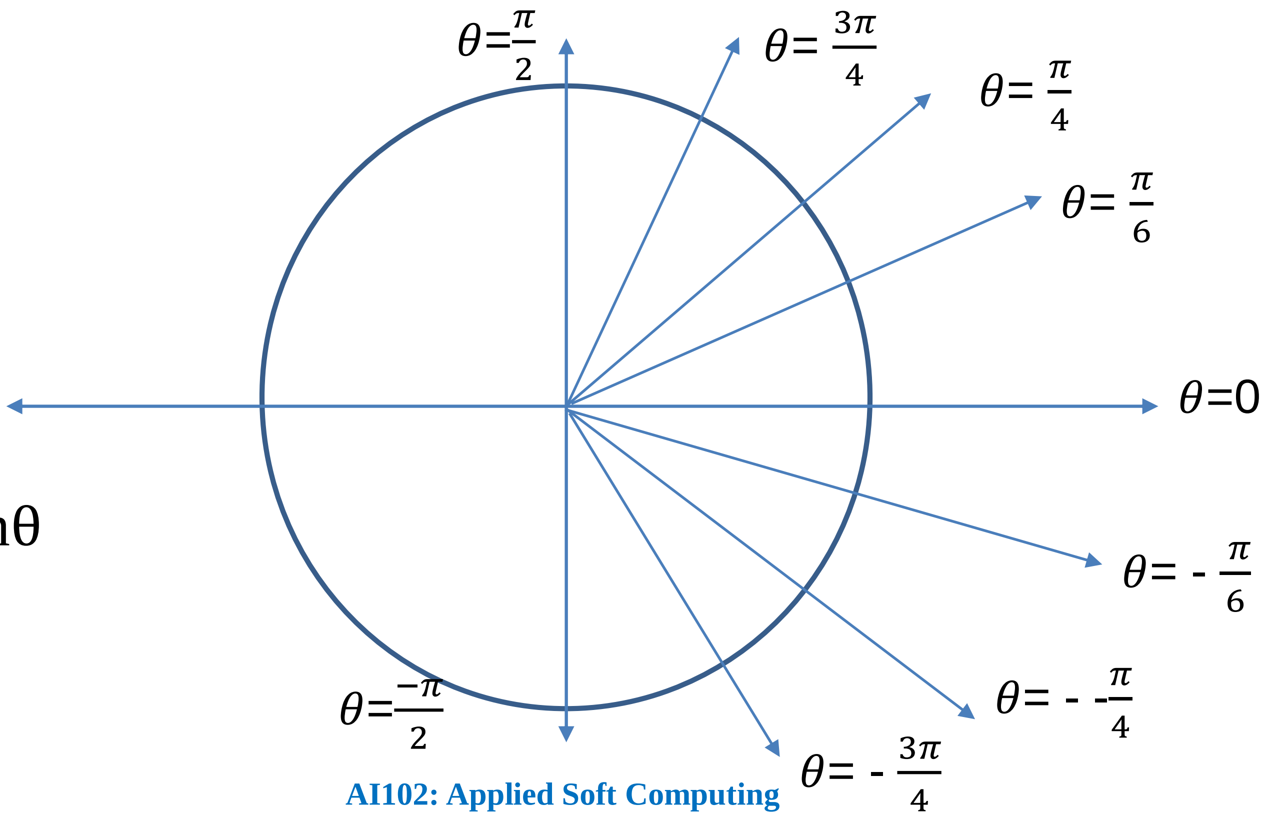
If $X=90^0$, the membership function value is 1

If $X=180^0$, membership function value is 0

membership values of right-angle triangle can be $I \cap R$

Membership Value Assignments

- **Rank Ordering :** Ranking based on preferences made by the individual, committee or by polling.
- **Angular Fuzzy Sets :** Fuzzy sets defined on universe of angles. It is based on linguistic values/ truth values.
- Eg. pH value 7 is neutral, value > 7 is basic (B) and < 7 is acidic (A)



- $\mu_t = t \cdot \tan\theta$

Membership Value Assignments

- **Neural Networks:** The collected data points are divided into classes by clustering techniques.
  - Data point lying in the region defined for class has membership of
  - Next level of data points are evaluated based on coordinates and assigned membership values
  - Similarly all data points are covered

Membership Value Assignments

- **Genetic Algorithms:**
  - Assume the same membership functions and shapes for fuzzy variables
  - Code the membership functions into bit strings
  - Concatenate the bit strings
  - The fitness function is used to evaluate the fitness of each set of membership function
  - The membership functions define the functional mapping of the system.
  - Obtain membership functions with best fitness value.

Membership Value Assignments

- **Induction Reasoning:** used to deduce causes through backward inference
  - Requires well defined database for input-output relationship
  - Applied for complex systems where data is in abundance and static
  - Establish fuzzy threshold between classes of data
  - Determine threshold line using entropy minimization method
  - Start segmentation into two classes
  - Again partition the two classes to get three different classes
  - Repeat the process to get number of classes
  - Membership function is determined based on classes

Fuzzy If Then Rules

Natural language expression for representing human knowledge given by

If antecedent **then** consequent

Fuzzy logic uses linguistic variables characterized by

- Name of the variable (x)
- Term set of the variable t(x)
- Syntactic rules for generating values of x
- Semantic rules for associating each value of x with its meaning

Fuzzy logic uses linguistic hedges

very, highly, slightly, moderately, fairly

Fuzzy If Then Rules

Natural language expression for representing human knowledge given by

If antecedent **then** consequent

General forms for linguistic variable:

Assignment Statements: Temperature=hot, y=tall

Conditional Statements: **If** person's IQ is high **then** person is intelligent

Unconditional Statements: Set the temperature low

Canonical form of fuzzy rule:

If condition C_1 then restriction R_1

Decomposition of Rules

The compound rule can be decomposed to number of simple canonical rule forms.

- **Multiple Conjunctive Antecedents**

If x is A_1, A_2, A_3……. A_n **then** y is B_m

Assume new fuzzy set is A_m; $A_m = A_1 \cap A_2 \cap A_3 \ldots\ldots \cap A_n$ and is expressed by membership function as

$\mu_{A^m}(x) = \min[\mu_{A^1}(x), \mu_{A^2}(x) \ldots\ldots \mu_{A^n}(x)]$

The rule can be rewritten as **If** A_m **then** B_m

Decomposition of Rules

- **Multiple Disjunctive Antecedents**

If x is A_1 OR A_2 OR A_3....... A_n **then** y is B_m

Rewritten as

If x is A_n **then** y is B_m

Assume new fuzzy set is A_m; $A_m = A_1 \cup A_2 \cup A_3 \ldots\ldots \cup A_n$ and is expressed by membership function as

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

Decomposition of Rules

Conditional statements

- **If** A_1 **then** B_1 else B_2

  Rewritten as

  If A_1 **then** B_1

  OR

  If not A_1 **then** B_2

- **If** A_1 (**then** B_1) **unless** A_2

  Rewritten as

  If A_1 **then** B_1

  OR

  If A_2 **then** not B_1

Decomposition of Rules

Nested If-then rules

- **If** A_1 **then** (**If** A_2 **then** B_1)

  Rewritten as

  If A_1 and A_2 **then** B_1

Fuzzy Inference System (FIS)

Fuzzy Inference system – key units of Fuzzy Logic System

– Fuzzy rule-based systems
– Fuzzy Models
– Fuzzy Expert Systems

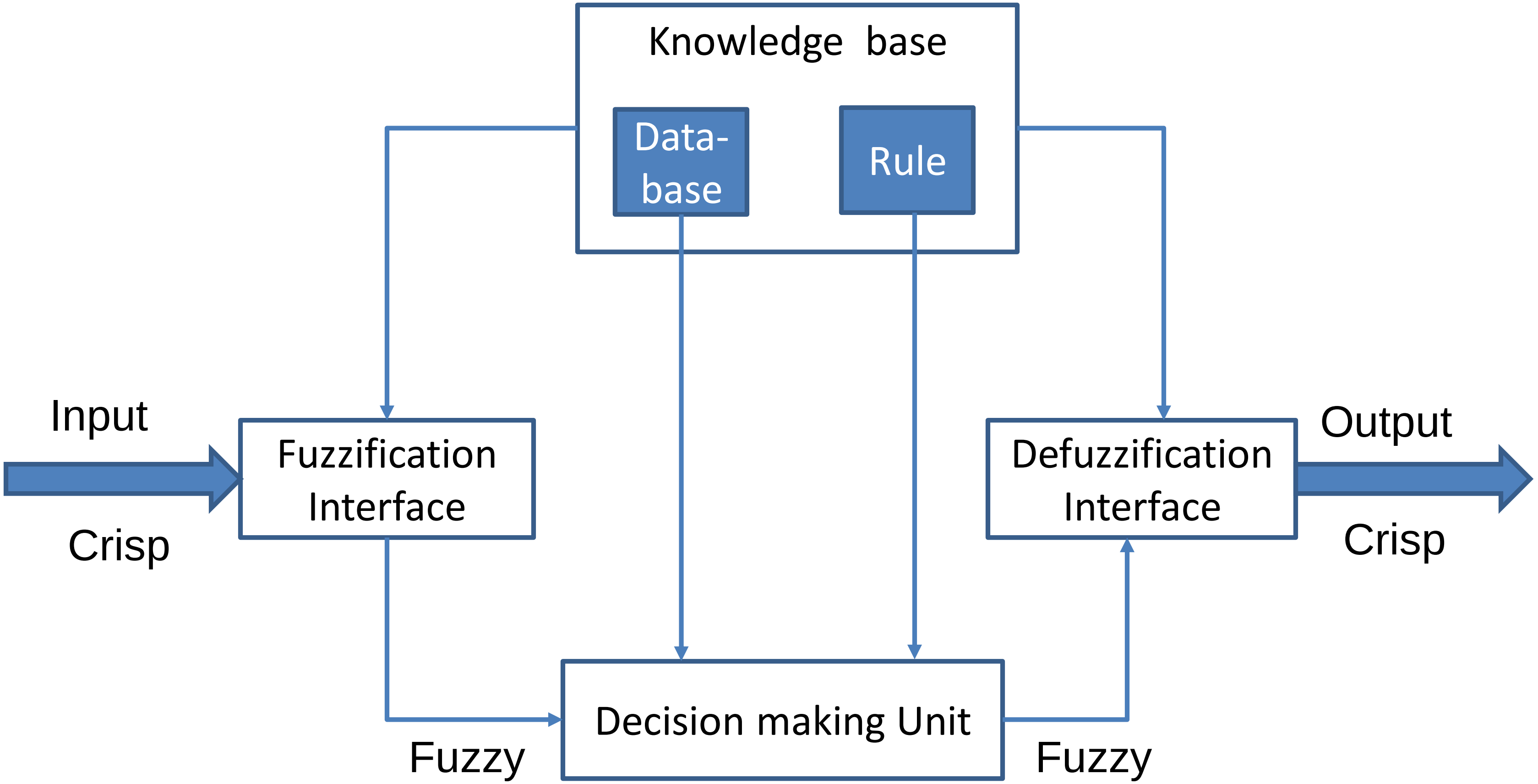
- FIS uses IF-THEN rules along with connectors “OR” or “AND”
- Input to FIS may be fuzzy or crisp but output is always a fuzzy set.
- Crisp output is required when FIS is used as controller
- To convert fuzzy output to crisp output, defuzzification is required

Construction & Working of FIS

FIS has five functional blocks

- Rule base containing fuzzy IF-THEN rules
- Database defining the membership functions of fuzzy sets used in fuzzy rules
- Decision making unit performing operation on rules
- Fuzzification interface unit, converting crisp quantities into fuzzy quantities
- Defuzzification interface unit, converting fuzzy quantities into crisp quantities

FIS Model



Knowledge base
Data-base
Rule
Input
Crisp
Fuzzification Interface
Decision making Unit
Fuzzy
Defuzzification Interface
Fuzzy
Output
Crisp

Inference Procedures

Two important inferring procedures are

Generalized Modus Ponens (GMP)

IF x is A THEN y is B

Antecedent: x is A’

Consequent y is B’

Generalized Modus Tollens (GMT)

IF x is A THEN y is B

Antecedent: y is B’

Consequent: x is A’

Inference Procedures

Two important inferring procedures are

Generalized Modus Ponens (GMP)

IF x is A THEN y is B

Antecedent: x is A’

Consequent y is B’

Generalized Modus Tollens (GMT)

IF x is A THEN y is B

Antecedent: y is B’

Consequent: x is A’

Methods of FIS

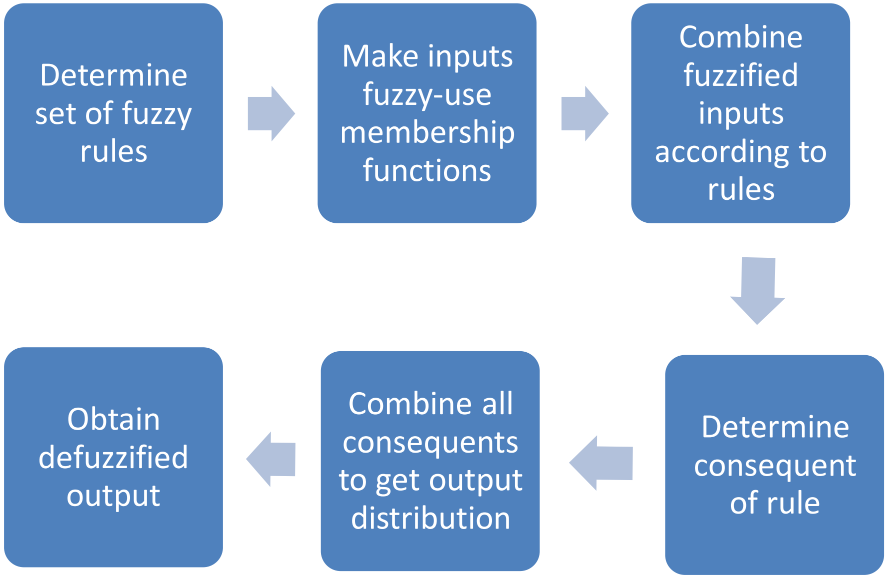
Two important methods of FIS:

– Mamdani FIS

– Sugeno FIS

Mamdani FIS

- Proposed by Ebsahim Mamdani in 1975
- To Control steam engine and boiler combination by defining fuzzy rules
- Output membership functions are fuzzy sets



Takagi-Sugeno FIS

- Proposed by Takagi, Sugeno & Kang in 1985
- Fuzzy Rule: IF x is A and y is B THEN $z=f(x,y)$
- A,B are fuzzy sets in antecedents, z is crisp function in consequent

Comparison of Mamdani and TSK Model

TSK Model:

- Output membership function in TSK model is linear or constant
- TSK Model is computationally efficient
- Works well with linear, adaptive and optimization techniques
- Suited for mathematical analysis
- Guaranteed continuity over output surface

Mamdani Model:

- Intuitive
- Widespread acceptance
- Well suited to human input

Defuzzification Techniques

Converting fuzzy set into crisp single-valued quantity

Techniques:

- **Lambda-Cut Method**
- **Maxima Methods**
  - Height (Max-membership principle)
  - First of Maxima
  - Last of maxima
  - Mean of Maxima (Mean-Max Membership)
- **Centroid Methods**
  - Center of gravity method (CoG)
  - Center of sum method (CoS)
  - Center of area method (CoA)
- **Weighted Average Method**

Lambda-cut (Alpha-cut) method

Applicable to Fuzzy Sets and Fuzzy Relations:

- fuzzy set A transformed into a crisp set A_λ for a given value of λ $(0 \le \lambda \le 1)$
- $A_\lambda = \{x \mid \mu_A(x) \ge \lambda\}$ – weak cut
- $A_\lambda = \{x \mid \mu_A(x) > \lambda\}$ – strong cut
- value of Lambda-cut set A_λ is x , when the membership value corresponding to x is greater than or equal to the specified value of λ.

Lambda-cut (Alpha-cut) method

Eg. Lambda-cut on sets

- $A_1 = \{(x_1, 0.8), (x_2, 0.4), (x_3, 0.1), (x_4, 0.3), (x_5, 0.6)\}$
- $A_{0.6} = \{(x_1, 1), (x_2, 0), (x_3, 0), (x_4, 0), (x_5, 1)\} = \{x_1, x_5\}$
- $A_{0.2} = \{(x_1, 1), (x_2, 1), (x_3, 0), (x_4, 1), (x_5, 1)\} = \{x_1, x_2, x_4, x_5\}$

Eg. Lambda-cut on relations

$$R = \begin{bmatrix} 0.1 & 0.3 & 0.6 \\ 0.8 & 0.4 & 0.5 \\ 0.2 & 0.9 & 0.7 \end{bmatrix} \quad R_{0.5} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \quad R_{0.8} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

Properties of Lambda-cut (Alpha-cut) method

- $(A \cup B)_\lambda = A_\lambda \cup B_\lambda$
- $(A \cap B)_\lambda = A_\lambda \cap B_\lambda$
- $(A^c)_\lambda \neq (A_\lambda)^c$ except $\lambda = 0.5$
- For any $\lambda \leq \beta,\ where\ 0 \leq \beta \leq 1, A_\beta \subseteq A_\lambda$

Lambda-cut for fuzzy relations follow the same properties

Maxima Methods of Defuzzification

- Height Method (Max-membership principle)

  $\mu_C(x^*) \geq \mu_C(x)$ for all $x \in X$

  x^* is the height of output of fuzzy set

  method is applicable when height is unique

- First of Maxima

  $x^* = \min\{x \,|\, C(x) = \max_w C\{w\}\}$

- Last of Maxima

  $x^* = \max\{x \,|\, C(x) = \max_w C\{w\}\}$

- Mean of Maxima

  Mean (average) of values of x* obtained from first of maxima and last of maxima

Centroid Methods of Defuzzification

- Center of Gravity (CoG) : The basic principle in CoG method is to find the point x^* where a vertical line would divide the aggregate into two equal masses.

$$x^* = \frac{\int \mu_C(\mathrm{x}) \,.\, \mathrm{xdx}}{\int \mu_C(\mathrm{x})\, \mathrm{dx}}$$

- Center of Sums (CoS) : It is the sum of individual; fuzzy subsets instead of their union (common part is considered twice)

$$\mathrm{x}^* = \frac{\int x \ \sum_{i=1}^{n} \mu_{c}(\mathrm{x}) \,.\, \mathrm{dx}}{\int \sum_{i=1}^{n} \mu_{c}(\mathrm{x}) \,.\, \mathrm{dx}}$$

- Center of Area (CoA) : When the output consists of atleast two convex non-overlapping sunsets, area of the largest fuzzy subset is taken

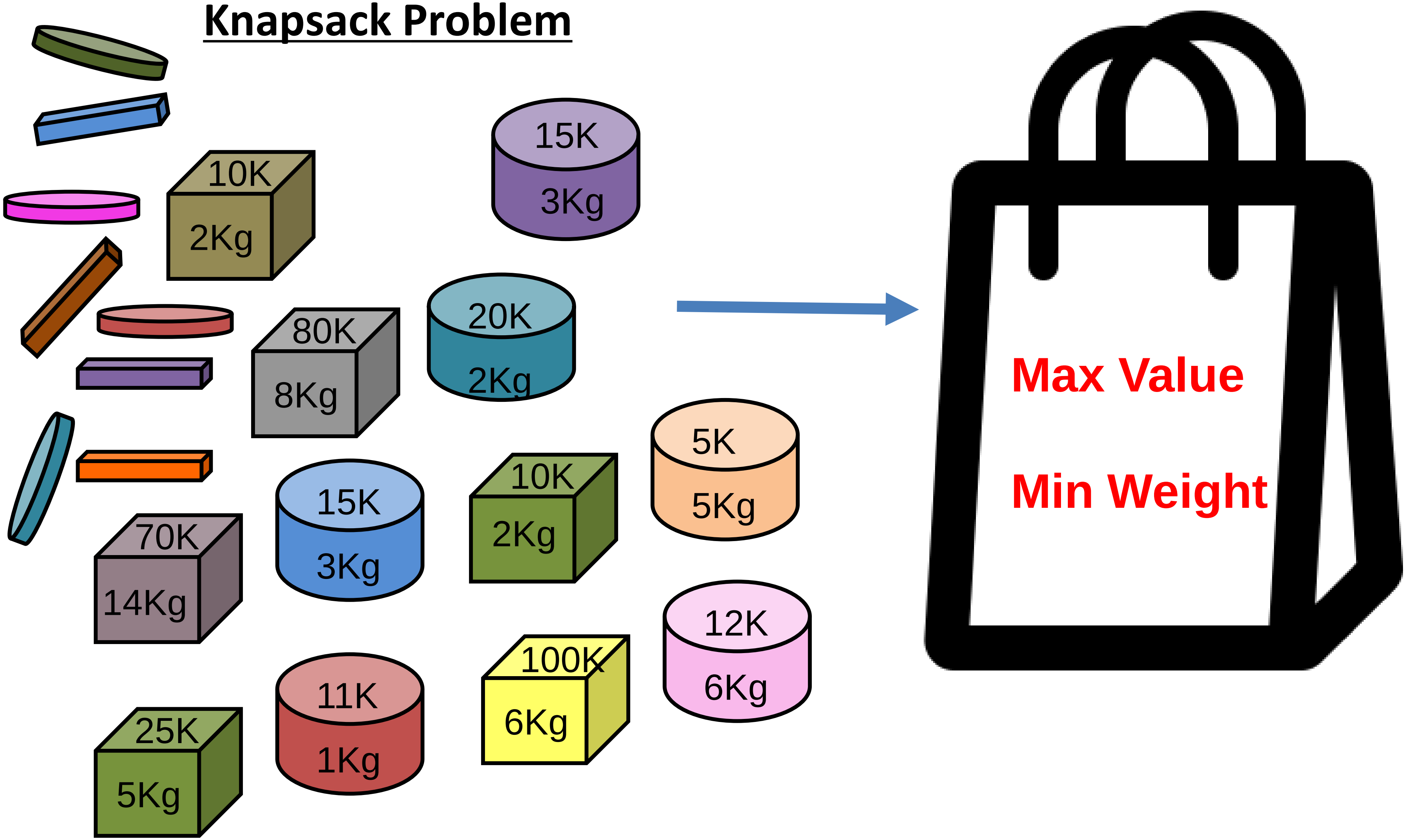
Weighted Average Method of Defuzzification

- Also termed as Sugeno Defuzzification method
- Used only for symmetrical output membership function

$$x^* = \frac{x.\sum_{i=1}^{n} \mu_c(\mathrm{x})}{\sum_{i=1}^{n} \mu_c(\mathrm{x})}$$

Why Genetic Algorithms?

Knapsack Problem



Why Genetic Algorithms?

Travelling Salesman Problem

Maximum destinations
Minimum Cost
Minimum Time

What are Genetic Algorithms (GAs)?

- **adaptive heuristic** search algorithms based on the evolutionary ideas of natural selection and genetics.
- represent an intelligent exploitation of a random search to solve **optimization** problems.
- exploit historical information to direct search within search space into region of better performance.
- designed to simulate processes in natural systems necessary for evolution **("survival of the fittest")**

In 1975, Holland developed this idea in **Adaptation in Natural and Artificial Systems** by applying principles of natural evolution to optimization problems.

Understanding biological terminologies

- The **genes** code the individual's characteristics.
- Genes can have different values or **alleles.** Eg. possible alleles for eye color can be black, brown, blue, green.
- **Gene Pool** is set of possible alleles present in a particular population.
- **Genome** is set of all the genes of specific species.
- Position of gene in genome is termed as **locus**.
- Genotype: set of genes of an individual.
- Phenotype: physical aspect of an individual.
- **Chromosomes** store genome and has two sets of genes (diploidy).
- Dominant one will determine phenotype.

Terminologies in GAs

GAs – based on an analogy with genetic structure and behaviour of chromosomes.

Foundations for GAs:

- Individuals in a population compete for resources and mates.
- Most successful individuals will produce more offsprings.
- Genes from good individuals propagate to produce offspring better than either of parents.
- Each successive generation will become more suited to environment.

Terminologies in GAs

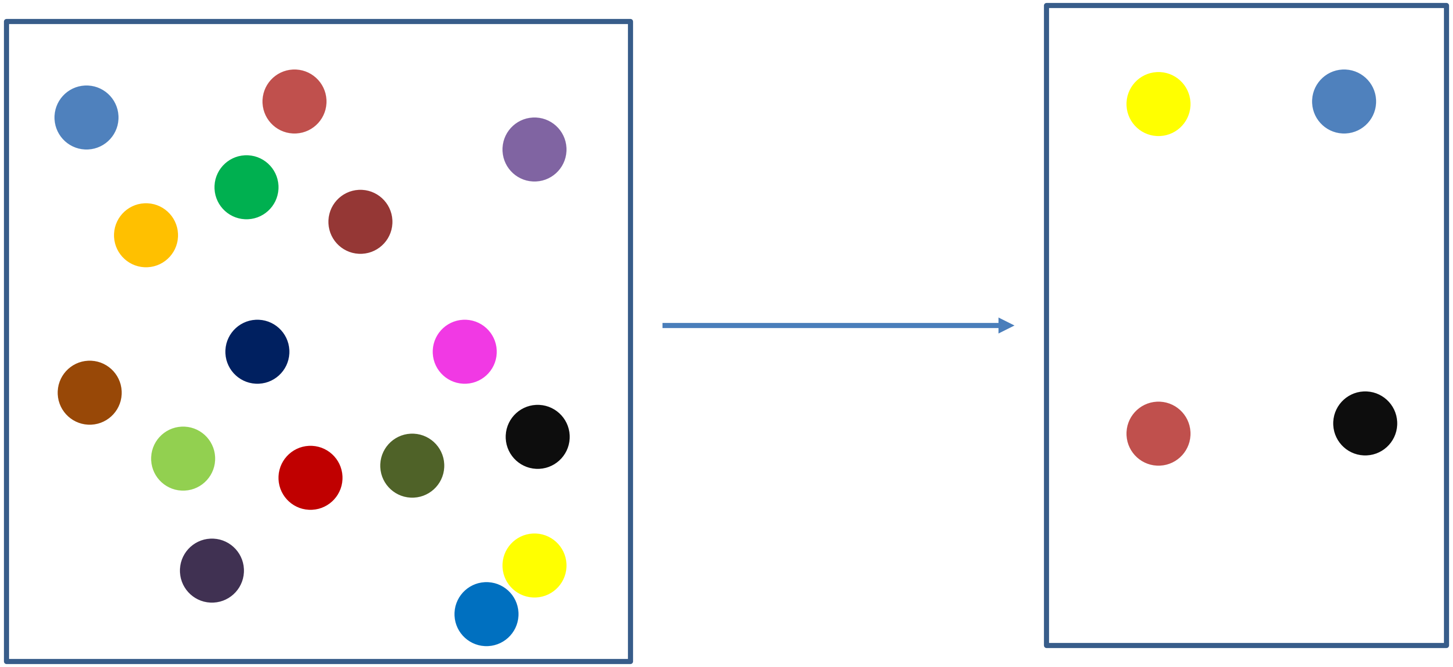
Search Space: Population of individuals is maintained within search space for GA.

Individual

- refers to the possible solution to given problem.
- is coded as finite length of vector of variables generally binary {0,1}.
- is assigned a fitness score representing its ability to compete.

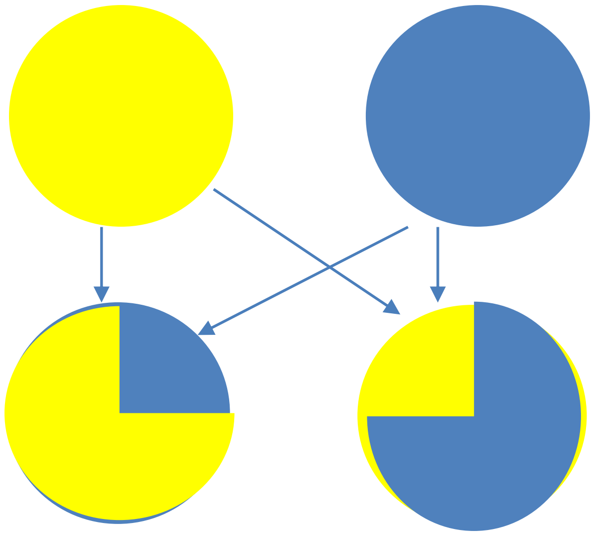
- Individuals are linked to chromosomes and variables to genes.
- Individuals with optimal fitness score are sought.
- GA aims for selective breeding of individuals for producing offspring better than parents.
- Offsprings are added to population and to keep the population size static old individuals are removed.
- It is expected to get the best fit solution over successive generations.

Selection Operator

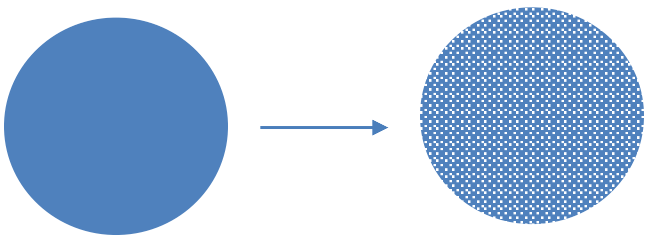


Selection based on fitness function

Crossover and Mutation Operator

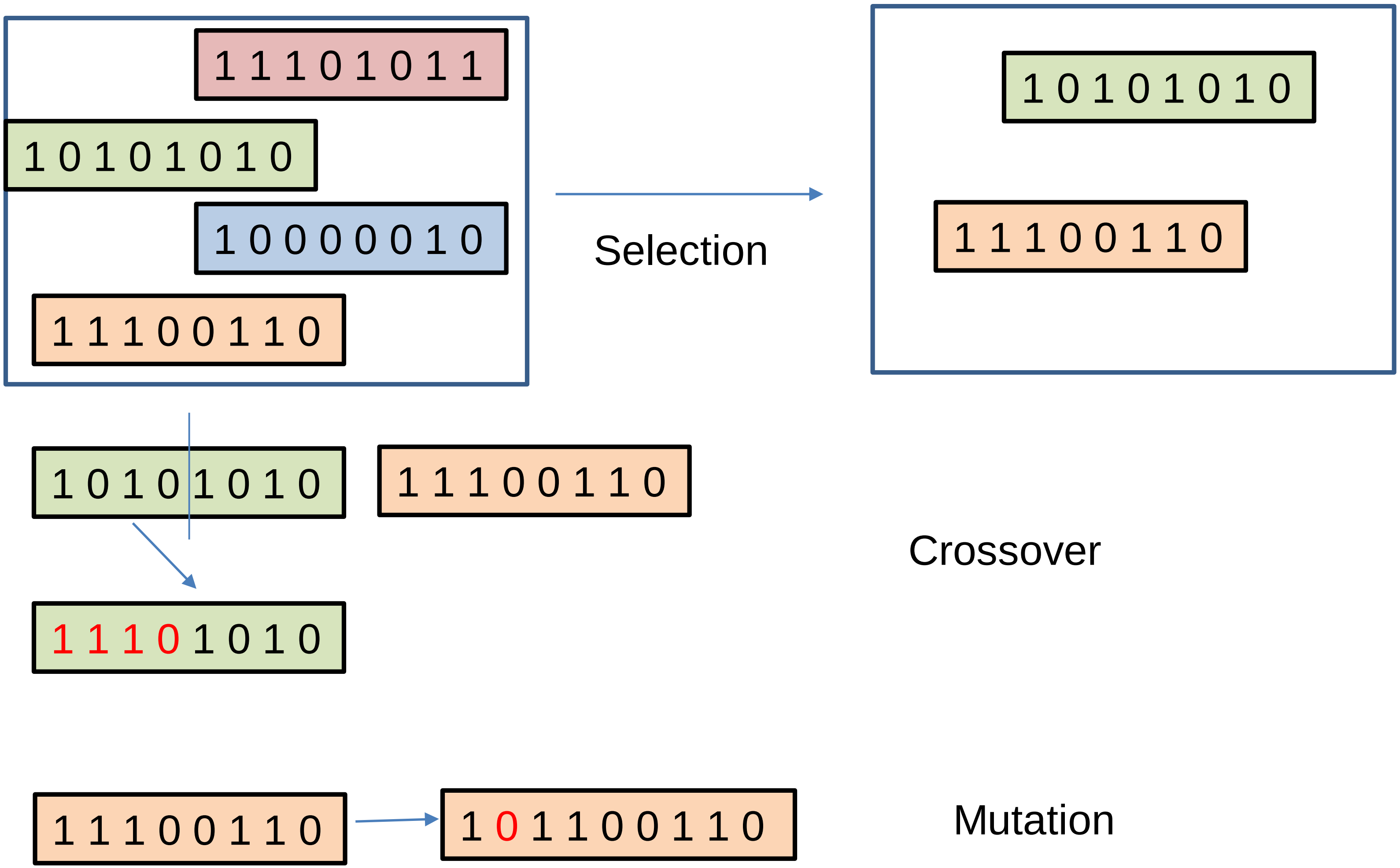


Crossover

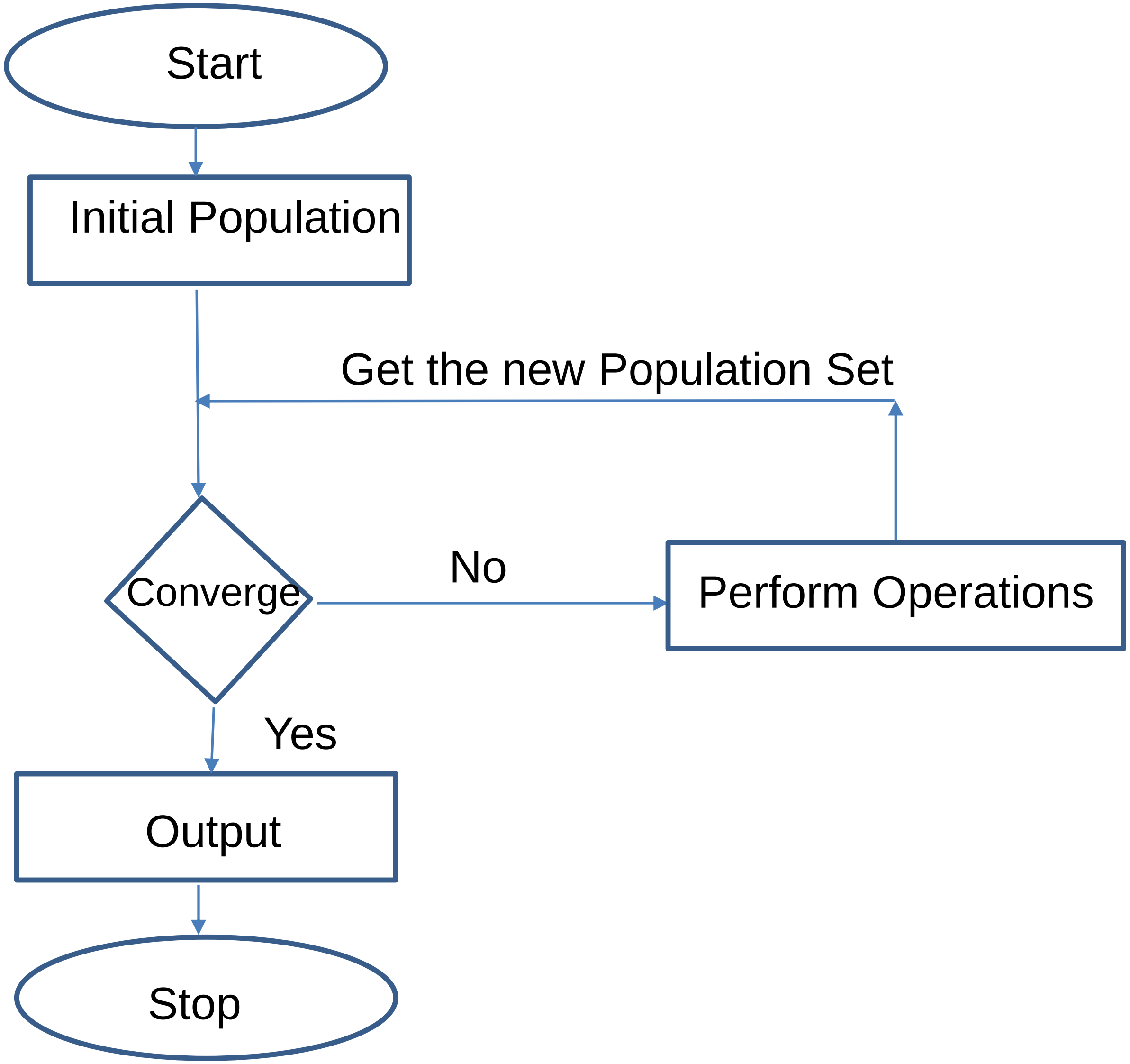


Mutation

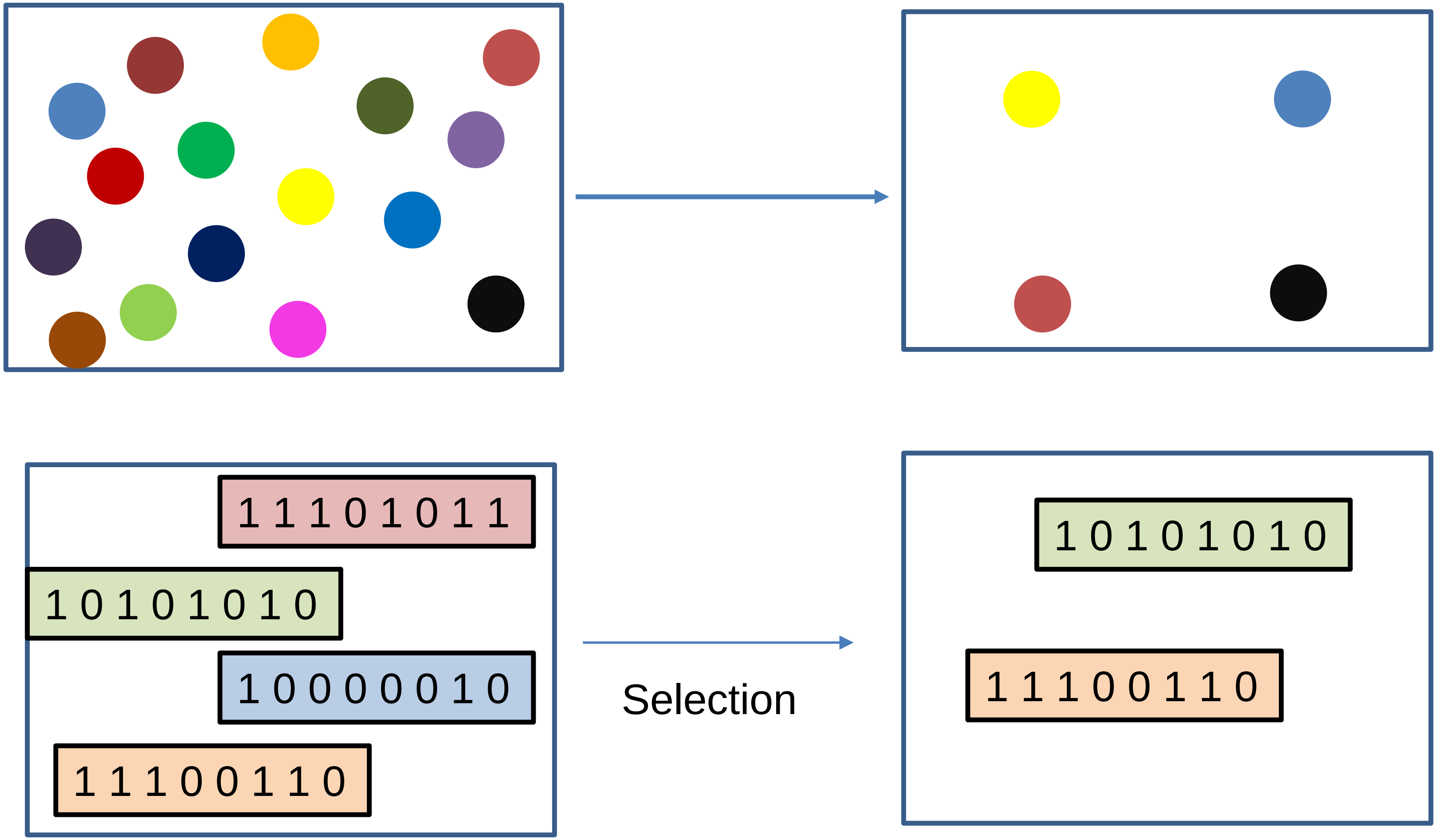
Operators in GA



Flowchart of GA



Selection Operator



Selection Operator

- Give preference to better individuals, allowing them to pass their genes to the next generation.
- Goodness of individual depends on its fitness.
- Fitness can be determined by an objective function or by subjective judgement.

How is selection done?

- Randomly picking chromosomes out of population.
- Criteria for picking is defined by evaluation function (fitness function).
- Higher the fitness function higher is the probability of selection.
- Degree to which better individuals are favored is selection pressure.
- Selection pressure drive GA to improve population fitness over several generations.
- Lower the Selection pressure, slower will be the convergence of GA and longer will be the time taken.
- Higher the Selection pressure, more are the chances of premature convergence of GA leading to sub-optimal solution.

Selection Schemes

- **Proportion-based selection** : selecting the individual on the basis of fitness value in comparison to other individuals in the population.

- **Ordinal based selection** : selecting the individuals based on their ranks in the population.

Proportion-based selection Scheme

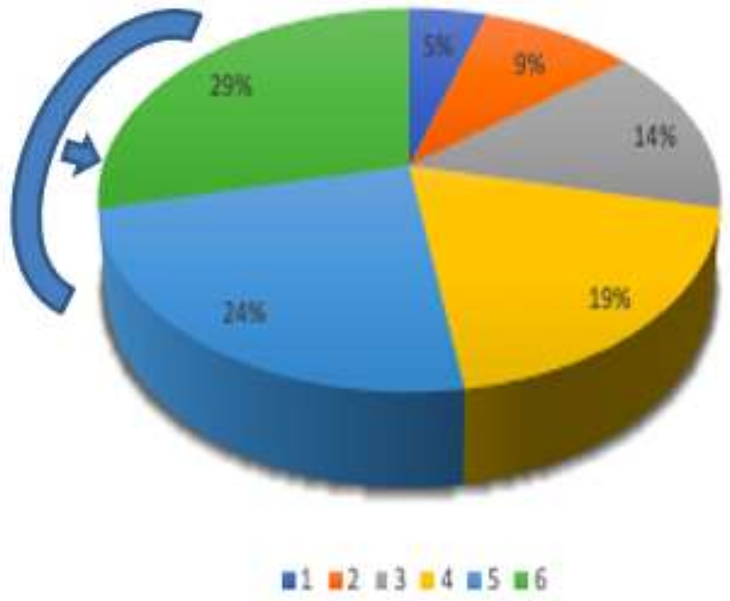
- Canonical Selection: the probability that the individual is selected is proportionate to its fitness.

Fitness = individual evaluation/ average evaluation of the entire population

Proportion-based selection Scheme

- Roulette – Wheel Selection

Individual	Fitness Value	Probability	%age
1	1.5	0.093	9
2	2.8	0.175	19
3	0.7	0.044	5
4	4.8	0.3	29
5	2.3	0.144	14
6	3.9	0.244	24



Ordinal-based Selection Scheme

- Rank-based Selection
  - Individuals are arranged in ascending order of respective fitness values (1 for least value)
  - Proportion-based scheme is followed based on assigned ranks

Individual	Fitness Value	Probability	%age value (R-W)	Rank Assigned	%age value based on ranks
1	1.8	0.128	13	2	20
2	3.2	0.228	23	3	30
3	1.6	0.115	11	1	10
4	7.4	0.529	53	4	40

Tournament Selection

- Sample a small group of individuals from the population.
- Individual with best fitness is selected.

Individual	1	2	3	4	5	6
Fitness	2.1	2.6	1.8	3.4	2.9	2.5

Trials	Group of Individuals	Selection based on fitness
1	1,3	1
2	2,5	5
3	4,6	4
4	1,6	6
5	2,4	4
6	3,5	5

Elitism Selection

- Used in conjunction with earlier schemes
- Elite class is identified and directly selected and for remaining individual other techniques are followed.

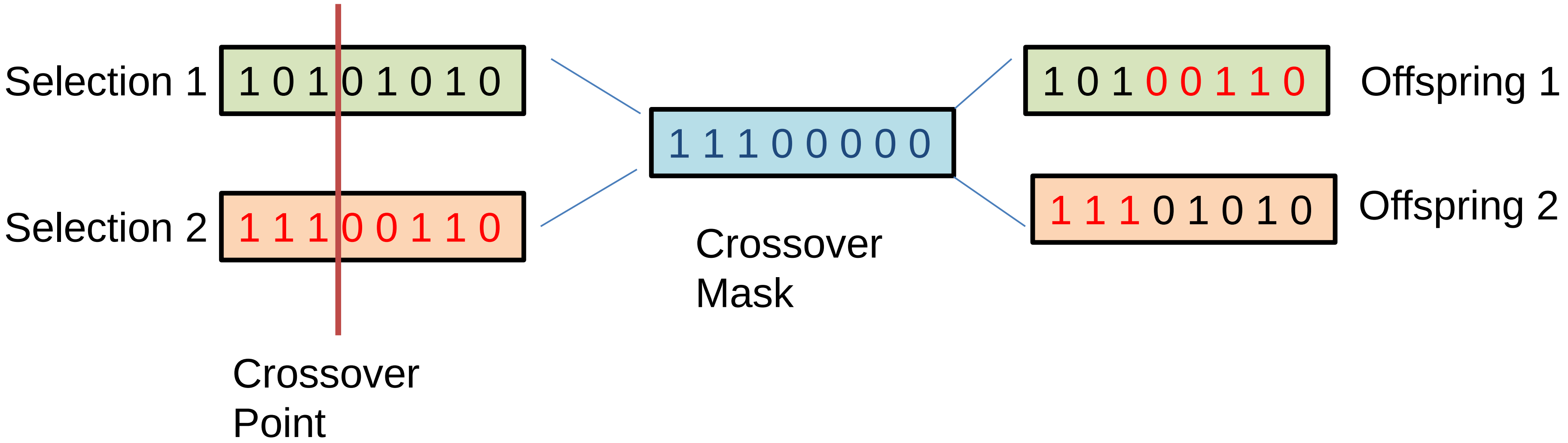
Crossover Operator

- Prime distinguished Factor of GA
- Two individuals are chosen from the population using selection operator.
- A crossover site is randomly chosen.

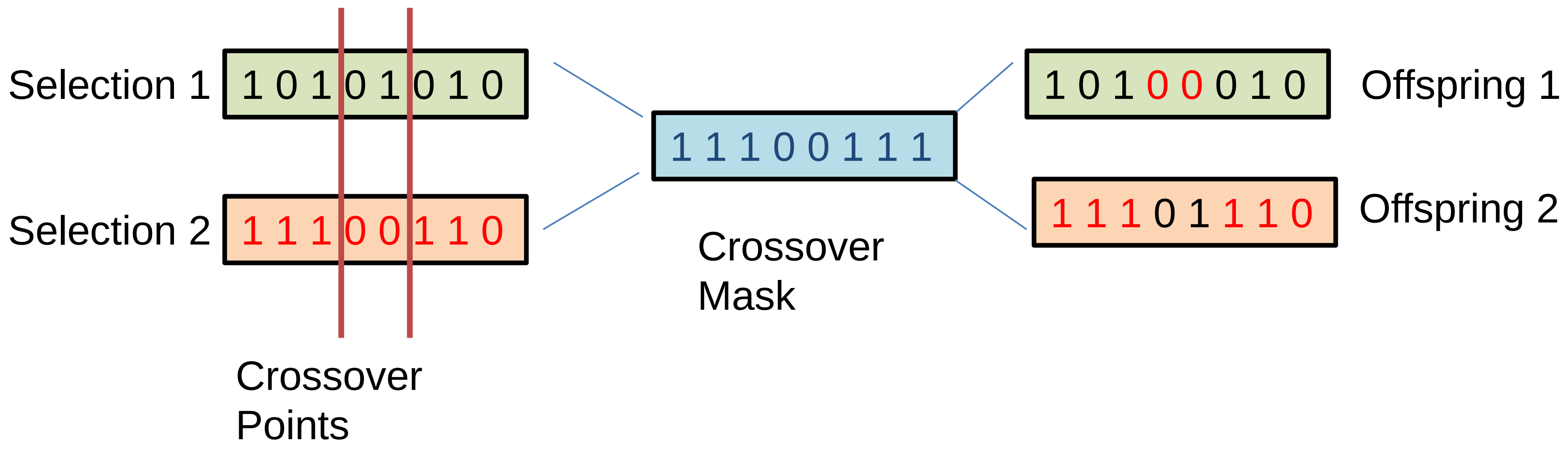
Types of Crossover Schemes

- Single Point Crossover
- Two Point Crossover
- Multi-point crossover
- Uniform Crossover
- Shuffle Crossover

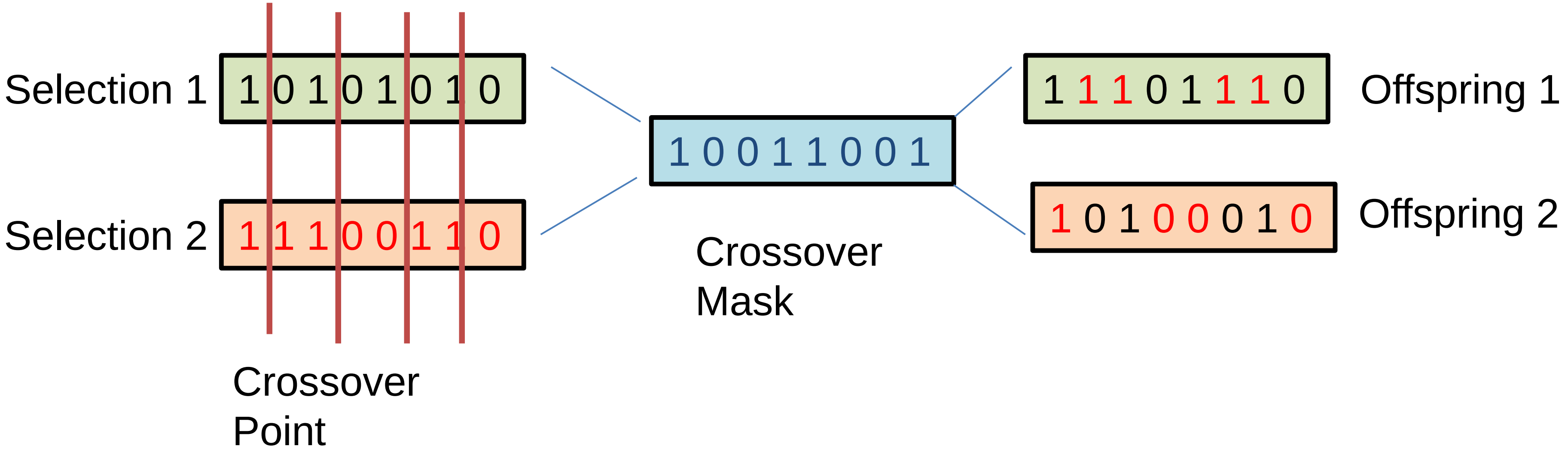
Single Point Crossover



Two Point Crossover



Multipoint Point Crossover



Uniform Crossover

Selection 1 | 1 0 1 0 1 0 1 0

Selection 2 | 1 1 1 0 0 1 1 0

0 0 0 1 1 0 0 1

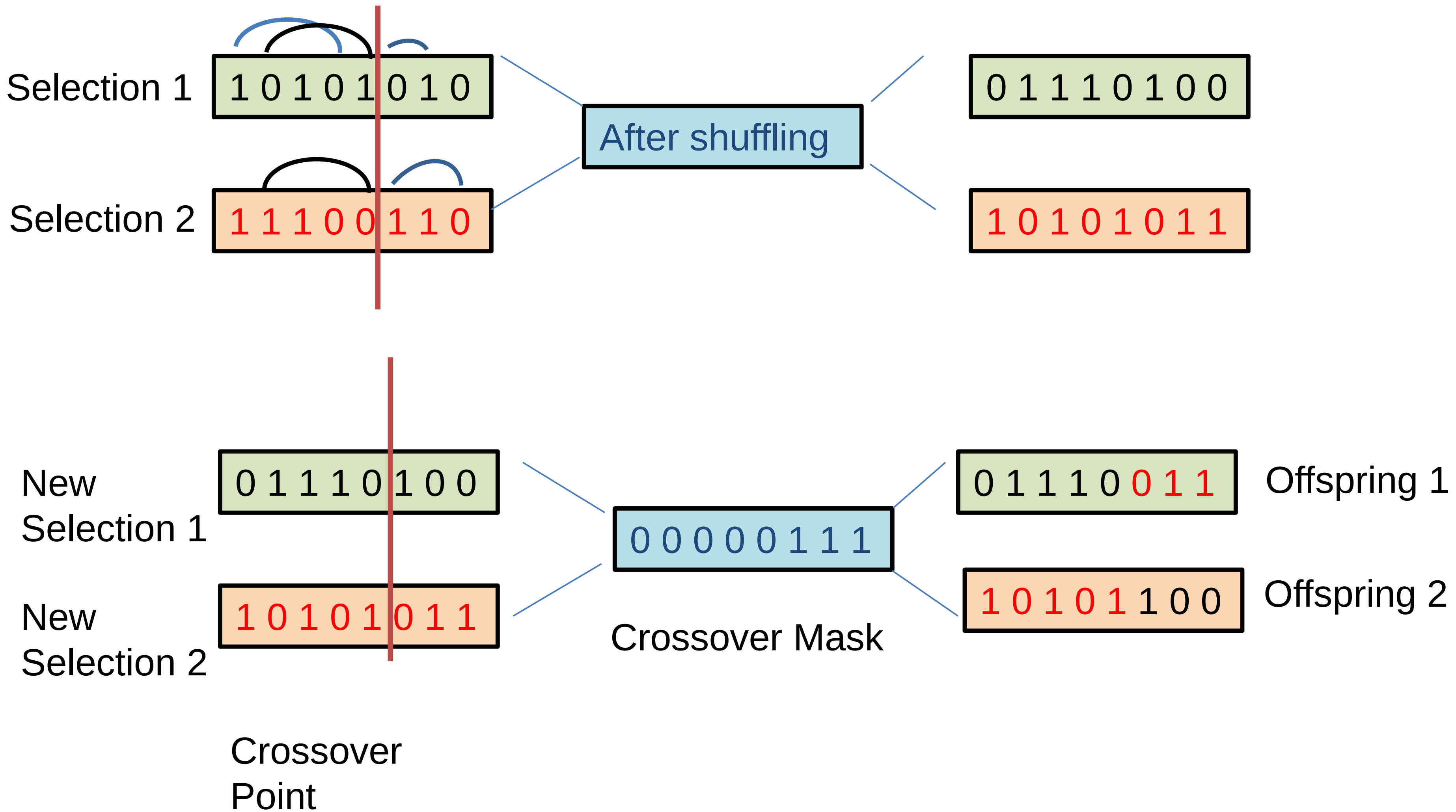
1 1 1 0 1 1 1 0 | Offspring 1

1 0 1 0 0 0 1 0 | Offspring 2

Crossover Point

Decide the mask based on coin tossing i.e. with probability p

Shuffle Crossover



Mutation Operation

- Maintains genetic diversity from one generation to next
- Similar to biological mutation
- Brings local change over present solutions.

Types of Mutations Operators

pM is the probability that a single gene is modified in mutation.

Mutation chromosome of same length as individual chromosome is generated based on pM.

- Binary Coded GA
  - Flipping : for 1 in mutation chromosome the corresponding bit in the individual selection is flipped from 0 to 1 or vice-versa
  - Interchanging : two randomly chosen bits are interchanged
  - Reversing : bit next to the chosen one is reversed

Mutation Operations in GA

CHITKARA UNIVERSITY

Mutation in Binary Coded GA:

	Flipping	**Interchanging**	**Reversing**
Offspring	1 0 1 0 1 0 1 0	1 0 1 0 1 0 1 0	1 1 1 0 1 1 1 0
Mutation Chromosome	1 0 1 0 0 0 1 0		
Mutated Offspring	0 0 0 0 1 0 0 0	1 1 1 0 0 0 1 0	1 1 1 0 1 1 0 0

CHITKARA UNIVERSITY

Mutation in Real Coded GA:

Random Mutation:

Mutated Offspring = Original Offspring + (r-0.5) x Δ

Where r is random number $0.0 \leq r \leq 1.0$

Δ is value of perturbation decided by user

Polynomial Mutation:

Mutated Offspring = Original Offspring + δ x Δ

δ is perturbation factor given $\begin{cases} 2r^{\frac{1}{q+1}} & \text{if } r < 0.5 \\ 1 - [2(1-r)]^{\frac{1}{q+1}} & \text{if } r \geq 0.5 \end{cases}$

Where r is random number $0.0 \leq r \leq 1.0$, q is positive or negative exponent

Encoding

String representation of the chromosomes

Types of Encoding

- Binary Encoding
- Real Value Encoding
- Order Encoding
- Tree Encoding

Encoding

String representation of the chromosomes

Types of Encoding

- Binary Encoding
- Real Value Encoding
- Order Encoding
- Tree Encoding

Encoding Techniques

Binary Encoding: Genes are represented in terms of strings of 0s and 1s. Length of the string can be fixed or variable.

Eg. in Knapsack problem there are n items and the item can be included or not in the sack with a limited capacity subject to maximization of total cost. It can be represented with a binary bit string of length n, with included item represented as 1 and others as 0.

0 1 0 1 1 1 0 0 0 1 1 1 0 0 1… 0 1 0

Encoding Techniques

Binary Encoding:

Pros:

GA Operations are faster if the representation is in binary form.

Cons:

Needs effort for conversion to binary string and also accuracy depends on the binary representation.

Encoding Techniques

Real Value Encoding: for optimization in continuous search space where binary values cannot suffice and real values are to be used.

Chromosome 1 : 6.234 5.32 6.865 1.832 3.69

Chromosome 2 : AA BC DF MZ CP KX WL

Chromosome 3 : blue green yellow red black orange white

Eg: Finding weights for neural network

Pros:

Needed for crossover and mutation

Cons:

Complex

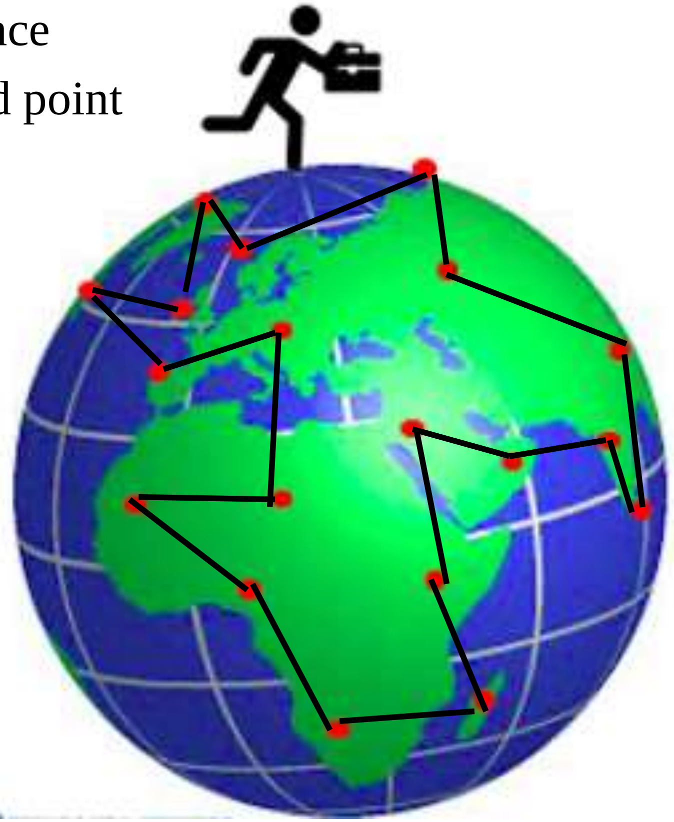
Encoding Techniques

Order Encoding: Used for problems requiring various permutations and combinations like Travelling Salesman Problem (TSP)

Constraints:

Visit all cities only once

Same starting and end point



Chromosome:

Number specified to the city in order of visit

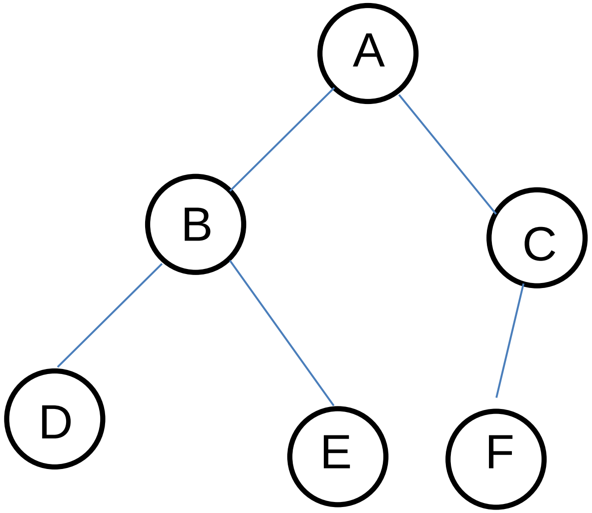
Eg.

1 15 7 8 12….

Possible Tour

Encoding Techniques

Tree Encoding : in the form of binary tree. Used mainly for evolving programs or expressions. Eg: floor planning in VLSI design, representing mathematical functions



Every chromosome is presented in the form of tree. L – Left subtree, R- Right Subtree

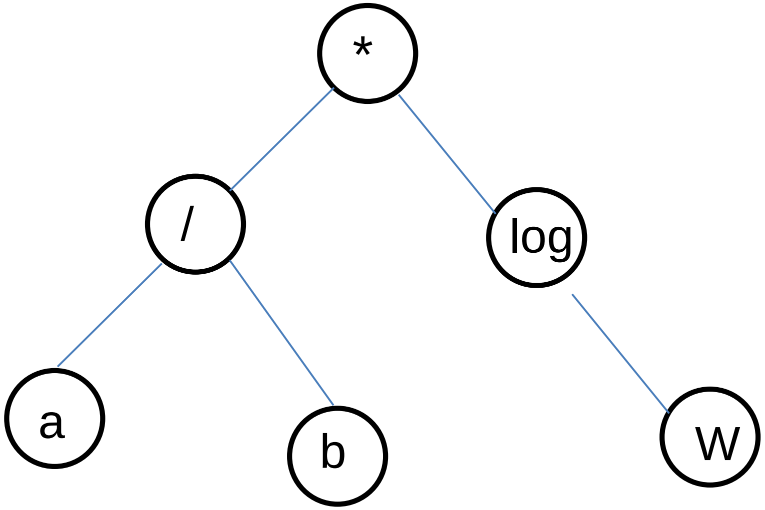
Any type of traversal can be done i.e.

inorder (L, Root, R) (D B E A F C)

preorder (Root, L, R) (A B D E C F)

postorder (L, R, Root) is (D E B F C A)

Examples of Tree Encoding :

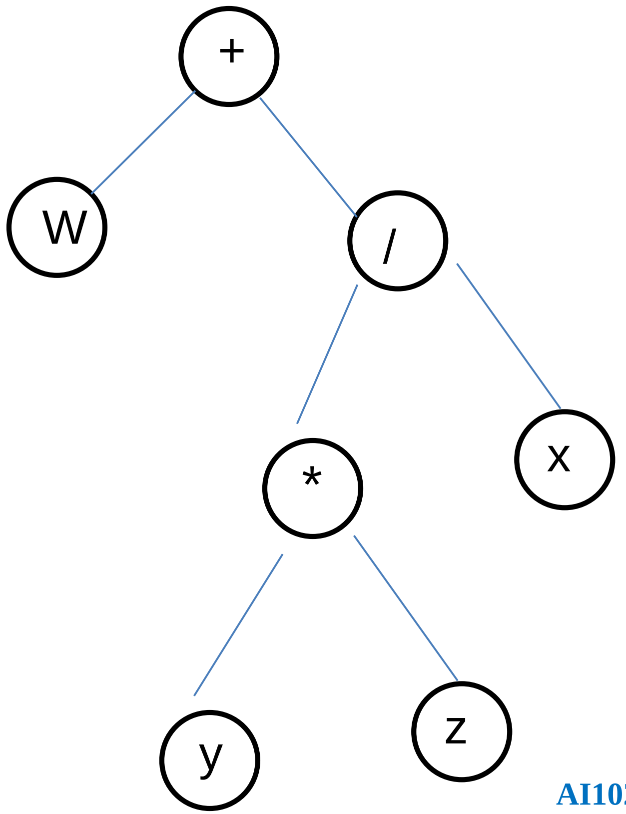


inorder (Left Subtree, Root, Right Subtree)

(a/b)*log(w)

Tree Encoding

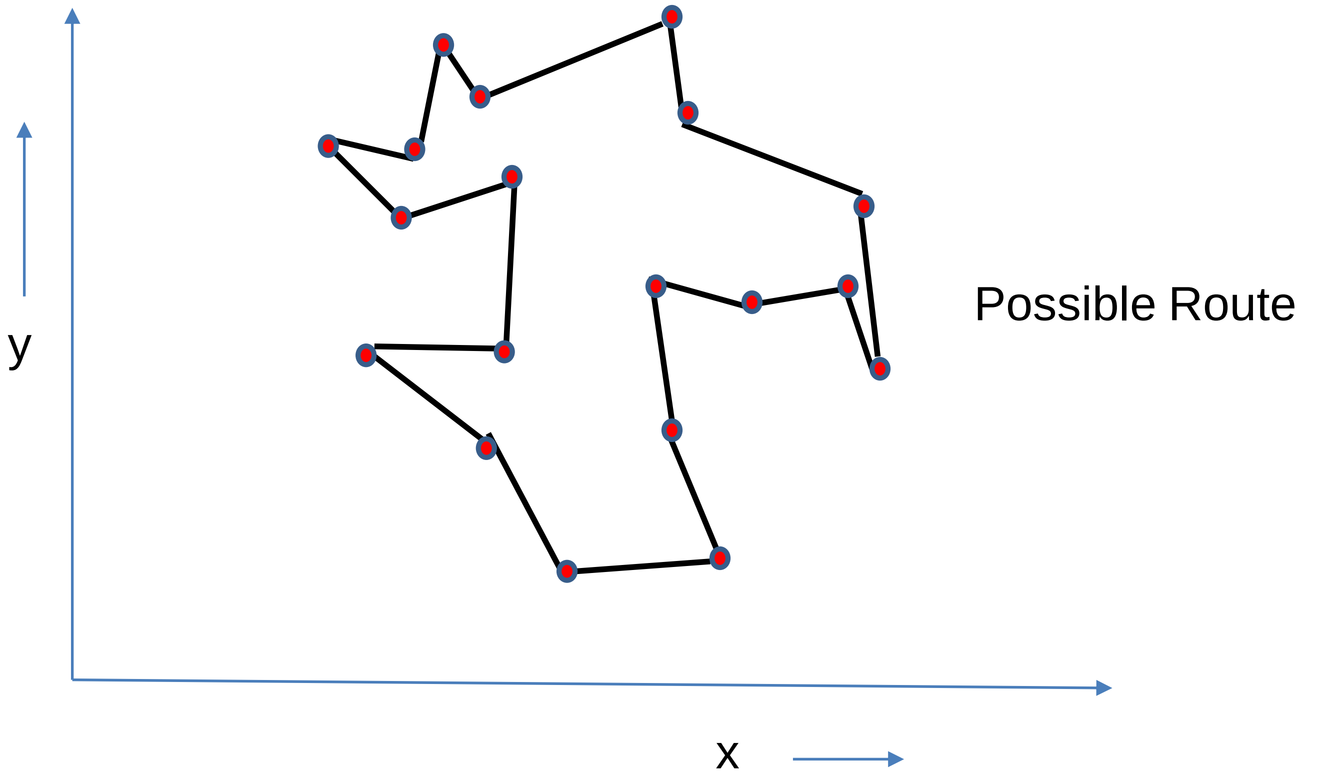
Examples of Tree Encoding :



inorder : w +(y*z)/x
preorder : + w / * y z x
postorder : w y z * x / +

Travelling Salesman Problem

The location in terms of x, y coordinates of cities is given. Find the shortest possible route such that each city is visited only once and route ends at the origin city.



Location of Cities in x and y coordinates

Algorithm for TSA

Step 1 (Encoding) : Generate random population of suitable solutions for the problem (chromosomes)

Step 2 : Evaluate the fitness of each solution (chromosome) in the population

Step 3: (Create a new population)

Repeat steps 4-7

Step 4 (selection) : Select suitable solutions (parent chromosomes) from a population based on fitness

Step 5 (crossover): Perform Crossover operation between selected solutions (parent chromosomes) to get new solution (offspring)

Step 6 (mutation): Perform mutation of new solution (offspring)

Step 7 (acceptance): Put the new solution (offspring) in population and remove any old solution from population to maintain the fixed population count

Step 8: Find solution from newly generated population. If not the best solution or if the end condition is not met. Go to step 3

Algorithm for TSA

Terminologies of GA in TSP

Gene: City (with x, y coordinates)

Individual (Chromosome): single route satisfying the criteria

Population : Set of possible routes (set of individuals)

Parent chromosomes : Two selected routes that can be operated to create new route

Fitness: Function to tell how good each route is.

Algorithm for TSA

Data: Six cities namely c1, c2, c3, c4, c5, c6

(x, y) coordinates of cities [(10,10), (20,20), (15, 35), (25,40), (30,50), (5, 50)]

Fitness Function of route starting from any city is the least route distance

Population Sample : Consider 4 routes out of total possible routes

[c1,c2,c3,c4,c5,c6]

[c2,c5,c3,c4,c6,c5]

[c4,c3,c2,c4,c6,c1]

[c6,c4,c2,c5,c3,c1]

Algorithm for TSA

Assign Ranks to the routes (eg.)

[c1,c2,c3,c4,c5,c6] – Rank 4

[c2,c5,c3,c4,c6,c6] – Rank 2

[c4,c3,c2,c4,c6,c1] – Rank 3

[c6,c4,c2,c5,c3,c1] – Rank 1

Select the parents (old routes) for performing operations to get new generation.

Perform Crossover get the offsprings (new routes), check fitness of route, if better then add in the population.

Perform mutation to get better results.

Maximizing function using GA

Maximize function $f(x) = x^2+1; \ 0 \leq x \leq 15$

Consider Population Size : 4

Maximizing function using GA

Solution: To represent the 16 different values of x, we need four bits, Search space will have population of 16

Step 1:

Initial Population (selected randomly		Value of X	f(x) (x^2+1)	Probability (P)	P in %	Expected Selection count of Individual	Rounded off Count
No.	String Representation	Decimal Notation		$\frac{f_i}{\sum_{i=1}^{4} f_i}$		$\frac{f(x)}{Avg. of\ f(x)}$	Round Off (val<0.5 ~ 0, val ≥ 0.5~1)
1	0 1 1 0	6	37	0.18	18	0.7	1
2	0 0 0 1	1	2	0.098	0.98	0.04	0
3	1 0 0 0	8	65	0.3175	31.8	1.27	1
4	1 0 1 0	10	101*	0.0493	49.3	1.97	2
SUM			205				
AVG			57.25				

* Maximum Value of function

Maximizing function using GA

Step 2: Single Point Crossover Operation on Population selected after Roulette Wheel Selection

Mating Pool (Selected from step 1)		Crossover Point	Offspring	Decimal Value of X	f(x) (x^2+1)
No.	String Representation	Crossover (1,2) and (3,4)			
1	0 1 1 0	1	0 0 1 0	2	5
2	1 0 1 0	1	1 1 1 0	14	197*
3	1 0 0 0	2	1 0 1 0	10	101
4	1 0 1 0	2	1 0 0 0	8	65

* Maximum Value of function

Maximizing function using GA

Step 3: Mutation of new Population of offspring obtained from Step 2

New Population (offsprings)		Mutation Chromosome	Offspring	Decimal Value of X	f(x) (x^2+1)
No.	String Representation				
1	0 0 1 0	1 0 0 0	1 0 1 0	10	101
2	1 1 1 0	0 0 0 1	1 1 1 1	15	226*
3	1 0 1 0	0 0 0 1	1 0 1 1	11	101
4	1 0 0 0	0 1 0 0	1 1 0 0	12	145

* Maximum Value of function

Maximizing function using GA

Maximum value obtained of the function is 226

Comparing the maximum values obtained from three steps

Step No.	Maximum Value of f(x)
1 (Selection)	101
2 (Crossover)	197
3 (Mutation)	226