M.S. Ramaiah Institute of Technology
(Autonomous Institute, Affiliated to VTU)
Department of Computer Science and Engineering

Course Name: Artificial Intelligence
Course Code: CSE551
Credits: 3:0:0:0

Term: September – December 2020

Faculty:
Dr. Annapurna P Patil
Professor and Chief Proctor
Mr. Shreekant Jere
Assistant Professor

References

1. Stuart Russel, Peter Norvig: Artificial Intelligence - A Modern Approach, 2nd Edition, Pearson Education, 2012.
2. Elaine Rich, Kevin Knight, Shivashankar B Nair: Artificial Intelligence, 3rd Edition, Tata McGraw Hill, 2011.
3. Nils J. Nilsson: Principles of Artificial Intelligence, First Edition, Elsevier, 2002.
4. Luger, G. F., & Stubblefield, W. A., Artificial Intelligence - Structures and Strategies for Complex Problem Solving. New York, NY: Addison Wesley, 5th edition (2005).
5. http://aima.cs.Berkeley.edu

Acknowledgement:

We acknowledge the authors listed above and all the course materials available on the Internet in the area of Artificial Intelligence and Machine Learning.

Our Next discussion : UNIT V

UNIT V

Genetic Algorithms: Genetic Algorithms Introduction, Significance of Genetic Operators, Termination Parameters, Niching and Speciation, Evolving Neural Networks, Theoretical Grounding, Ant Algorithms.

Robotics: Introduction, Hardware, Perception, Planning to Move, Planning Uncertain Movement, Moving, Robotic Software Architecture, Application Domains.

Philosophical Foundations: Weak and Strong AI, The Ethics and Risks of Developing AI,

AI: The present and Future.

(Chapter 23 of Text Book 2, Chapter 25, 26, 27 of Text Book 1)

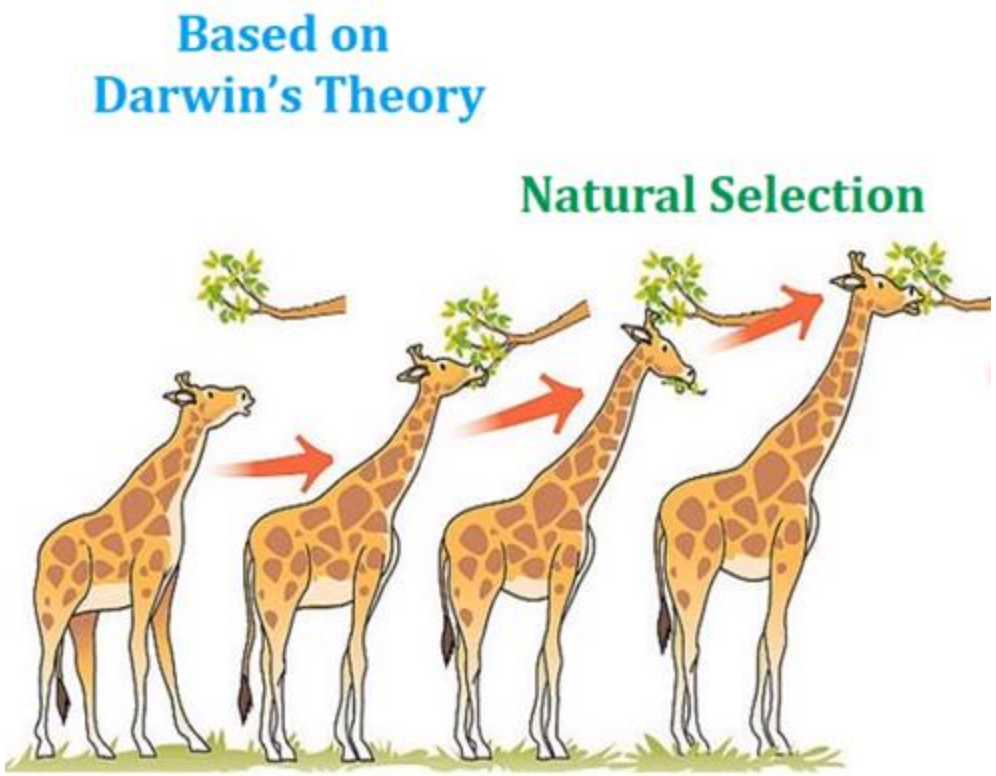
Task to be completed

Submission of Assignments and the study Paper.

Genetic Algorithms: Outline : 16.12.2020

- Introduction
- Motivation
- Applications
- Basic structure of Genetic Algorithm
- Basic terminology of Genetic Algorithm
- Knapsack problem by using Genetic Algorithm
- Advantages of Genetic Algorithm
- Niching, Speciation
- Introducing ANT Algorithms

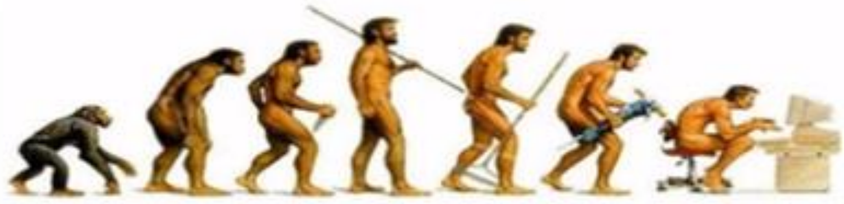
Genetic Algorithm



- John Holland introduced in 1975.
- Search-based optimization technique.
- Uses concept from Evolutionary Biology (Natural Genetics and Natural Selection)
- Pool or a population of possible solutions to the given problem.

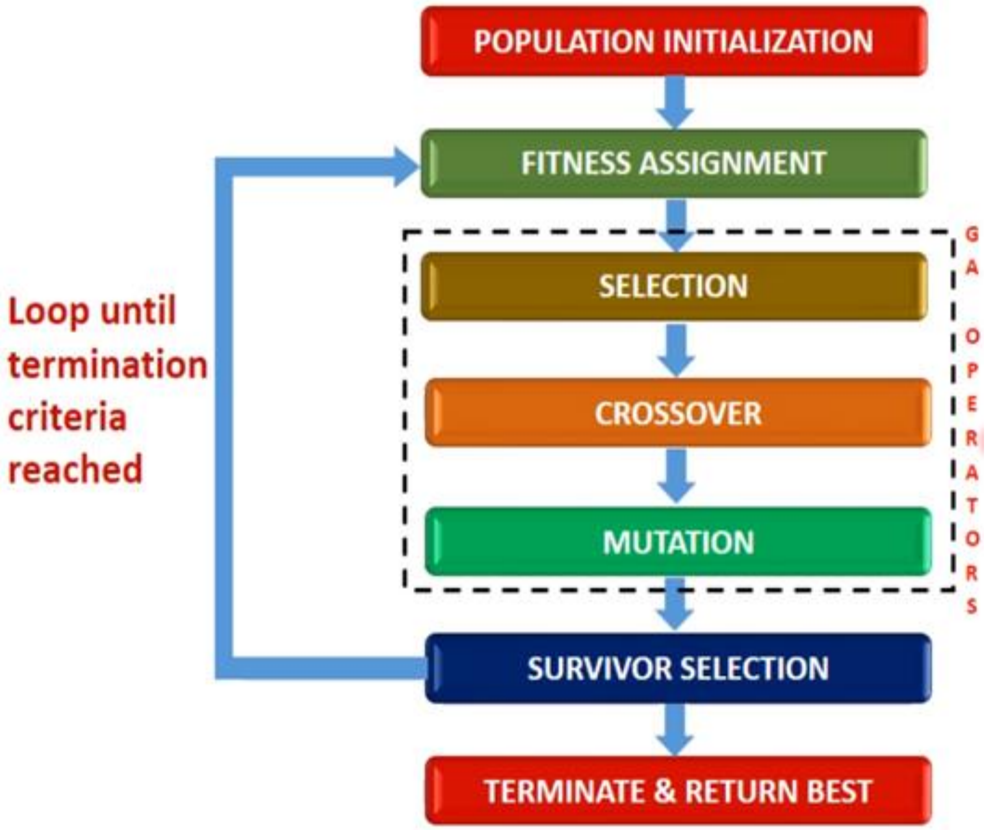
Motivation

- Genetic algorithm have the ability to deliver a “good-enough” solution “fast enough”.
- Evolution is known to be successful robust method for adaptation with biological systems.
- Solving Difficult Problems:
  - GAs prove to be an efficient tool to provide usable near-optimal solutions in a short amount of time.



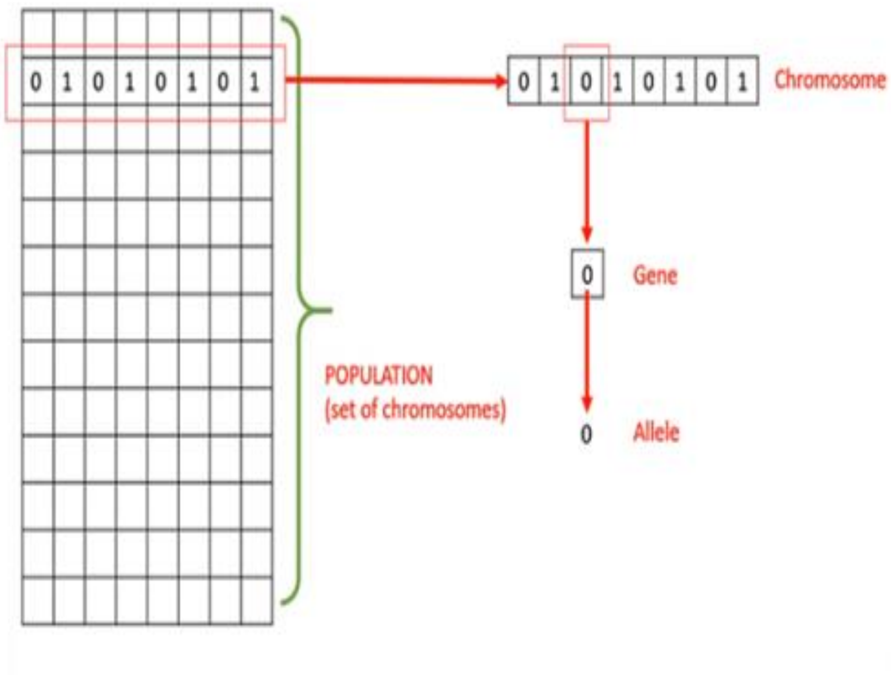

DNA Analysis
Robotics
Game Playing
Business
APPLICATIONS
Machine Learning
Image Processing
Vehicle Routing
Neural Network

Basic Structure of Genetic Algorithm



- Each iteration in the cycle produces a new "generation" of chromosomes.
- The entire set of generations is called a run.
- Typical GA run is from 50 to 500 or more generations.
- At the end of a run often there is at least one highly fit chromosome in the population.

Basic terminology of GA



- Population: subset of all the possible solutions to the given problem.
- Chromosomes: one such solution to given problem.
- Gene: one element position of a chromosome.
- Allele: value a gene takes for particular chromosome.
- Genotype: population in the computation space.
- Phenotype: population in the actual real world solution space.
- Decoding: transforming a solution from the genotype to the phenotype space.
- Encoding: transforming from the phenotype to genotype space.

Knapsack problem by using GA

ITEM	WEIGHT	VALUE
A	5 kg	$12
B	3 kg	$ 5
C	7 kg	$ 10
D	2 kg	$ 7

There are four items. Each item is associated with some weight (W) and value at item (V)

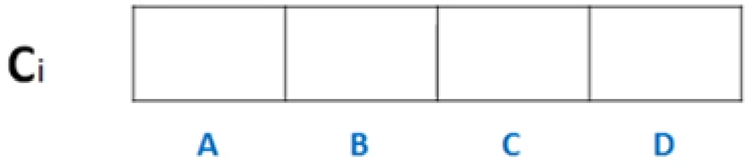
There is a knapsack (k) with limited capacity that can hold 12 kg.

Problem:

The problem is that which item should be kept in the knapsack so as it will maximize knapsack value without breaking knapsack.

Knapsack problem by using GA

STEP-1: Chromosomes Encoding



ITEM	WEIGHT	VALUE
A	5 kg	$12
B	3 kg	$ 5
C	7 kg	$ 10
D	2 kg	$ 7

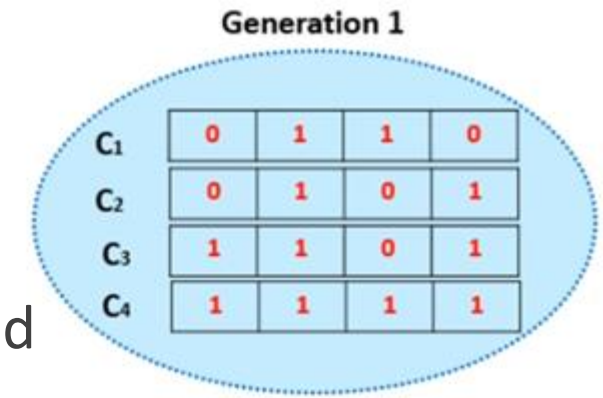
Gene: 0 – represents absence of item in the knapsack

1 – represents presence of item in the knapsack

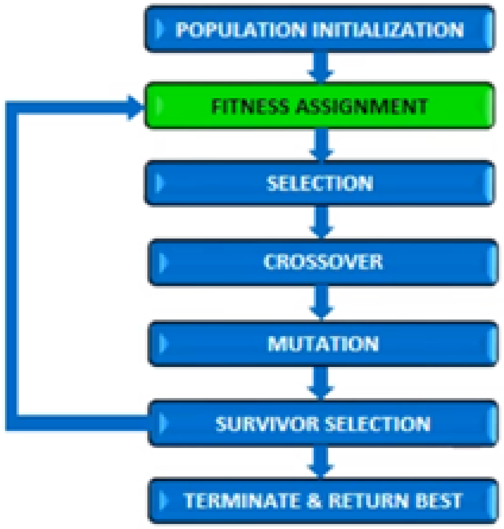
4 bits are requested to represent chromosomes encoding

Set space = 2 ^ 4 = 16

Initial population is created and chromosomes randomly created



Knapsack problem by using GA



STEP-2: Fitness Function

Next step is to determine fitness function which is used to evaluate how good particular solution is. Lets take C1.

C_1	0	1	1	0
	A	B	C	D

represent that knapsack has presence of item B & C and absence of item A & D.

Value of knapsack = value of B + value of C = 5 + 10 = 15

Weight of knapsack = weight of item B + weight of item C = 3 + 7 = 10 kg

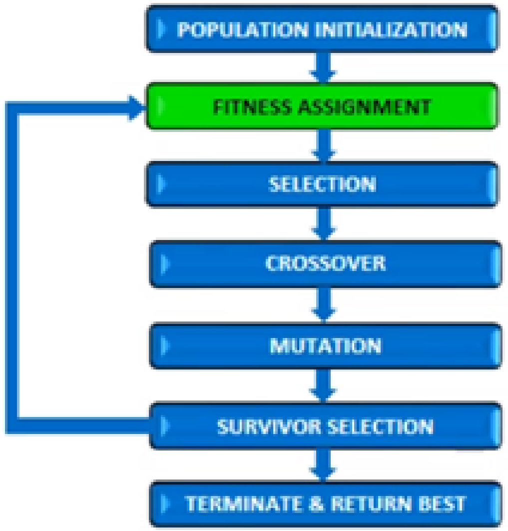
Knapsack capacity = 12 kg. 12 kg > 10 kg. **So C1 is accepted.**

ITEM	WEIGHT	VALUE
A	5 kg	$12
B	3 kg	$ 5
C	7 kg	$ 10
D	2 kg	$ 7

Chromosome Encoding

C1	0	1	1	0
C2	0	1	0	1
C3	1	1	0	1
C4	1	1	1	1

Knapsack problem by using GA



STEP-2: Fitness Function

Lets take C2.

	A	B	C	D
C_2	0	1	0	1

represent that knapsack has presence of item B & D and absence of item A & C.

Value of knapsack = value of B + value of D = 5 + 7 = 12

Weight of knapsack = weight of item B + weight of item D = 3 + 2 = 5 kg

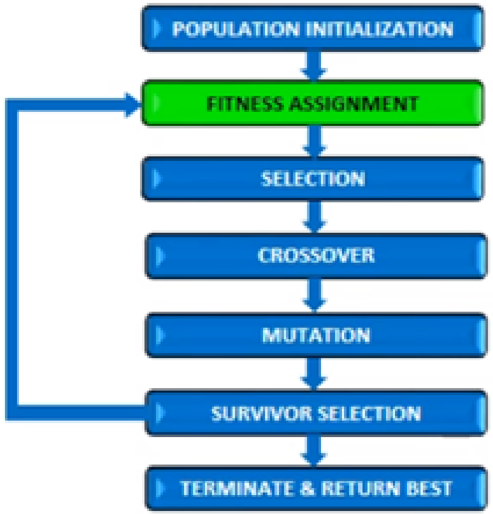
Knapsack capacity = 12 kg. 12 kg > 5 kg. **So C2 is accepted.**

ITEM	WEIGHT	VALUE
A	5 kg	$12
B	3 kg	$ 5
C	7 kg	$ 10
D	2 kg	$ 7

Chromosome Encoding

C1	0	1	1	0
C2	0	1	0	1
C3	1	1	0	1
C4	1	1	1	1

Knapsack problem by using GA



STEP-2: Fitness Function

Lets take C3.

	A	B	C	D
C3	1	1	0	1

represent that knapsack has presence of item A, B & D and absence of item C.

Value of knapsack = value of A + value of B + value of D = 12 + 5 + 7 = 24

Weight of knapsack = weight of item A + weight of item B + weight of item D = 5 + 3 + 2 = 10 kg

Knapsack capacity = 12 kg. 12 kg > 10 kg. **So C3 is accepted.**

ITEM	WEIGHT	VALUE
A	5 kg	$12
B	3 kg	$ 5
C	7 kg	$ 10
D	2 kg	$ 7

Chromosome Encoding

C1	0	1	1	0
C2	0	1	0	1
C3	1	1	0	1
C4	1	1	1	1

RAMAIAH
Institute of Technology

Knapsack problem by using GA

STEP-2: Fitness Function

Lets take C4.

C4	1	1	1	1
	A	B	C	D

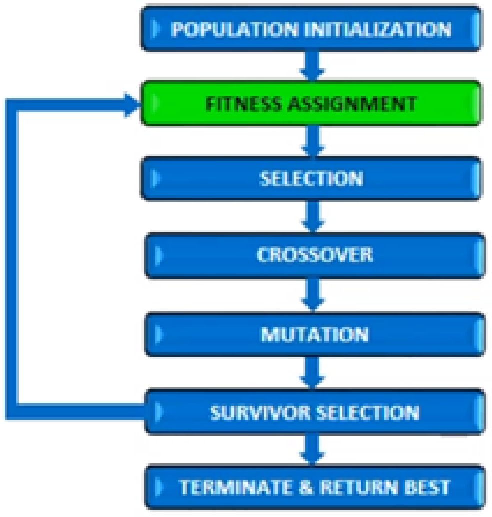
represent that knapsack has presence of all item A, B, C & D.

Value of knapsack = value of A + value of B + value of C+ value of D

= 12 + 5 + 10 + 7 = 34

Weight of knapsack = weight of item A + weight of item B + weight of item C + weight of item D = 5 + 3 + 7 + 2 = 17 kg

Knapsack capacity = 12 kg. 12 kg < 17 kg. **So C4 is discarded.**



ITEM	WEIGHT	VALUE
A	5 kg	$12
B	3 kg	$ 5
C	7 kg	$ 10
D	2 kg	$ 7

Chromosome Encoding

C1	0	1	1	0
C2	0	1	0	1
C3	1	1	0	1
C4	1	1	1	1

Knapsack problem by using GA

STEP-3: Selection

Next step is to collect the filter individual and wake up the next generation, Chromosome

By using Roulette wheel Selection:

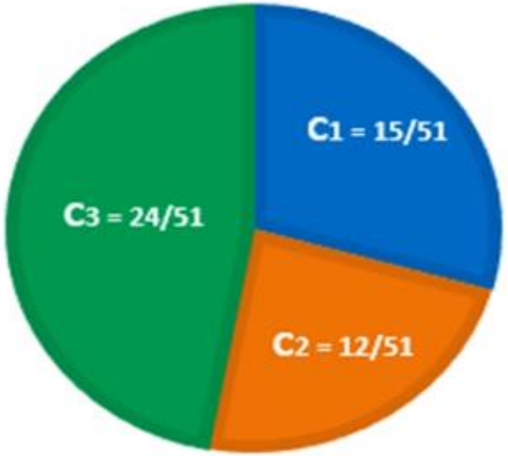
- Spin the Roulette Wheel and whenever the wheel stops, the individual gets selected at that point.
- The individual that has highest fitness value gets larger share of the wheel.

e. g. total fitness value = 15 + 12+ 24 + 0 =51

Fitness value of C3 = 24, largest fitness value

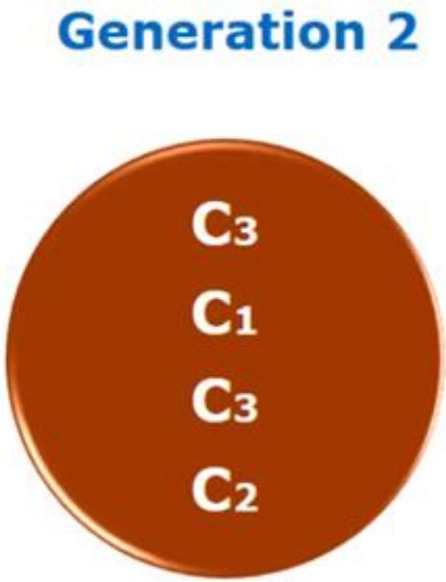
So, C3 occupies half of the wheel as 24/51

C4 has zero chance of winning, C3 has the highest probability of getting selected in the next generation



Knapsack problem by using GA

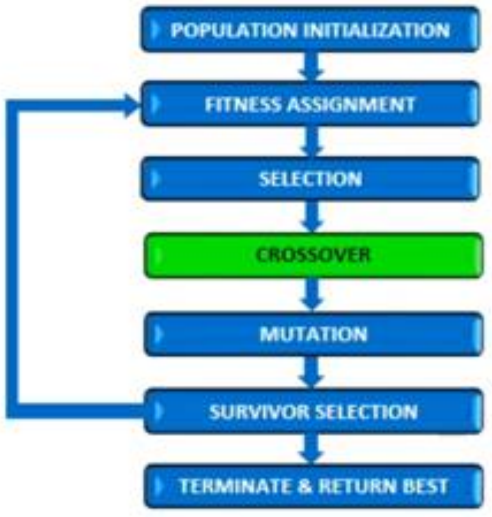
After spinning Roulette wheel, in the first spin C3 will be selected and then C1 after that C3 and C2.
Individuals of next generation are selected as follows:



ITEM	WEIGHT	VALUE
A	5 kg	$12
B	3 kg	$ 5
C	7 kg	$ 10
D	2 kg	$ 7

RAMAIAH
Institute of Technology

Knapsack problem by using GA



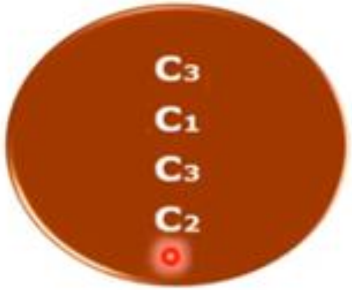
STEP-4: Crossover

The crossover operation takes the selected chromosomes, and mixes the genetic material to produce offspring.

	A	B	C	D
C3	1	1	0	1
C2	0	1	1	0

ITEM	WEIGHT	VALUE
A	5 kg	$12
B	3 kg	$ 5
C	7 kg	$ 10
D	2 kg	$ 7

Generation 2

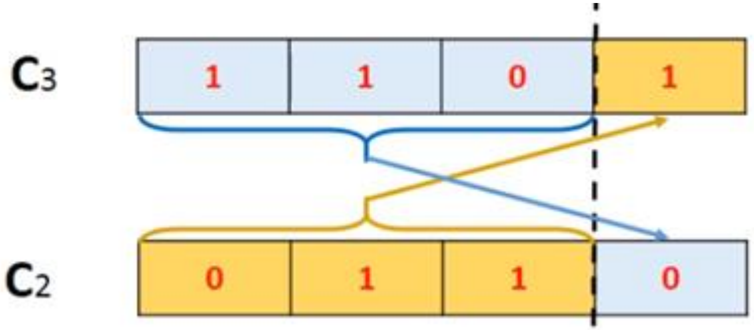


One point crossover – Randomly select the position on the chromosomes about which gene would be exchange.

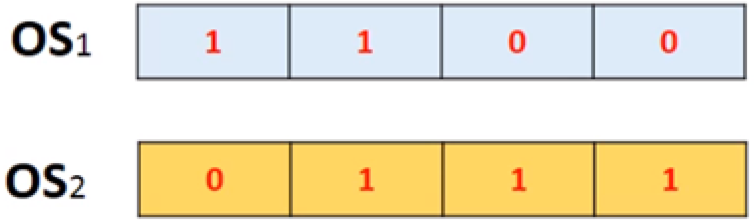
C3	1	1	0	1
C2	0	1	1	0

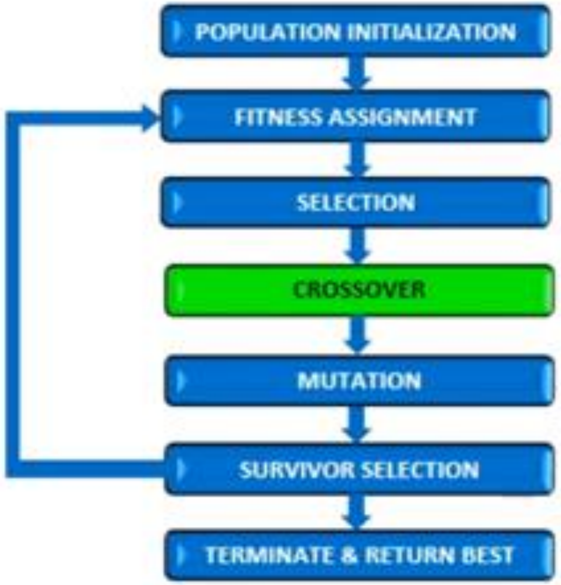
Knapsack problem by using GA

One-point crossover – Randomly select the position on the chromosomes about which gene would be exchange.



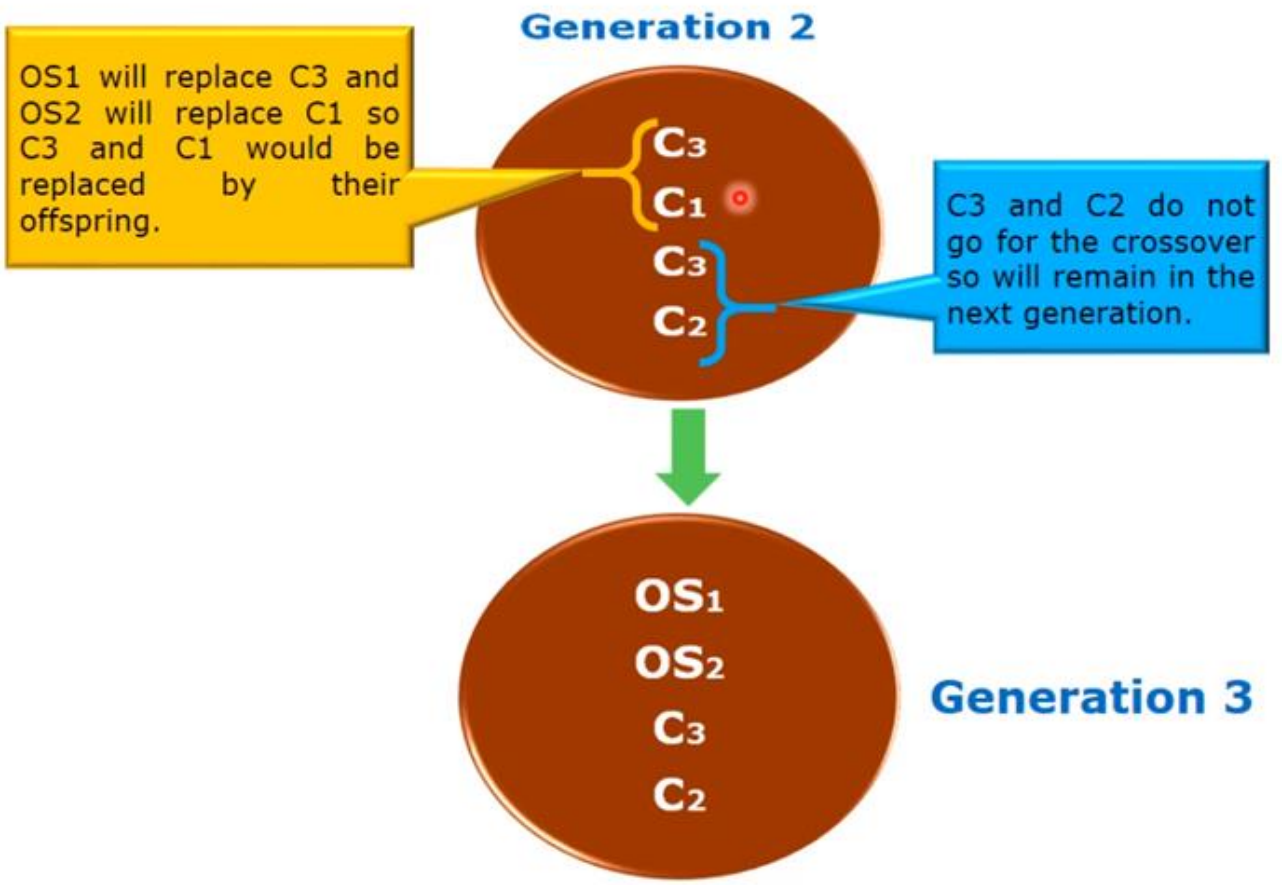
Result of one point crossover i.e. produced offspring

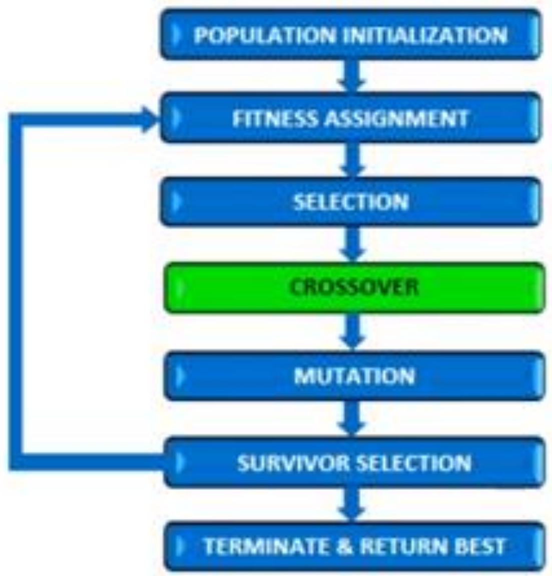




ITEM	WEIGHT	VALUE
A	5 kg	$12
B	3 kg	$ 5
C	7 kg	$ 10
D	2 kg	$ 7

Knapsack problem by using GA

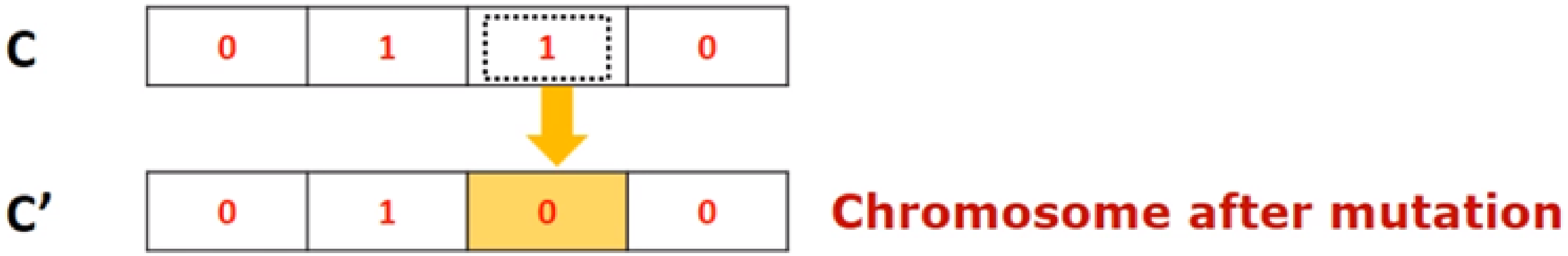




Knapsack problem by using GA

STEP-4: Mutation

Let us consider chromosomes C. Now randomly select a gene from C. A flip happens at the selected genes and zero becomes one and one becomes zero.



RAMAIAH Institute of Technology

Knapsack problem by using GA

Example 2:

Items	1	2	3	4
Weights	5	4	6	3
Values	10	40	30	50

Capacity : 10

RAMAIAH
Institute of Technology

Genotype Representation

- One of the most important decisions to make while implementing a genetic algorithm is deciding the representation that we will use to represent our solutions.
- It has been observed that improper representation can lead to poor performance of the GA.
- Therefore, choosing a proper representation, having a proper definition of the mappings between the phenotype and genotype spaces is essential for the success of a GA.
- In this section, we discuss some of the most commonly used representations for genetic algorithms. However, representation is highly problem specific and the reader might find that another representation or a mix of the representations mentioned here might suit the problem better.

Genotype Representation

1. Binary Representation

- This is one of the simplest and most widely used representation in GAs.
- In this type of representation the genotype consists of bit strings.
- For some problems when the solution space consists of Boolean decision variables – yes or no, the binary representation is natural.
- Take for example the 0/1 Knapsack Problem.
- If there are n items, we can represent a solution by a binary string of n elements, where the x^{th} element tells whether the item x is picked (1) or not (0).

0	0	1	0	1	1	1	0	0	1

Genotype Representation

2. Real Valued Representation

- For problems where we want to define the genes using continuous rather than discrete variables, the real valued representation is the most natural.
- The precision of these real valued or floating point numbers is however limited to the computer.

0.5	0.2	0.6	0.8	0.7	0.4	0.3	0.2	0.1	0.9

Genotype Representation

3. Integer Representation

- For discrete valued genes, we cannot always limit the solution space to binary ‘yes’ or ‘no’.
- For example, if we want to encode the four distances – North, South, East and West, we can encode them as **{0,1,2,3}**.
- In such cases, integer representation is desirable.

1	2	3	4	3	2	4	1	2	1

Genotype Representation

4. Permutation Representation

- In many problems, the solution is represented by an order of elements. In such cases permutation representation is the most suited.
- A classic example of this representation is the travelling salesman problem (TSP).
- In this the salesman has to take a tour of all the cities, visiting each city exactly once and come back to the starting city. The total distance of the tour has to be minimized. The solution to this TSP is naturally an ordering or permutation of all the cities and therefore using a permutation representation makes sense for this problem.

1	5	9	8	7	4	2	3	6	0

Population Initialization

There are two primary methods to initialize a population in a GA. They are −

- **Random Initialization** − Populate the initial population with completely random solutions.
- **Heuristic Initialization** − Populate the initial population using a known heuristic for the problem.

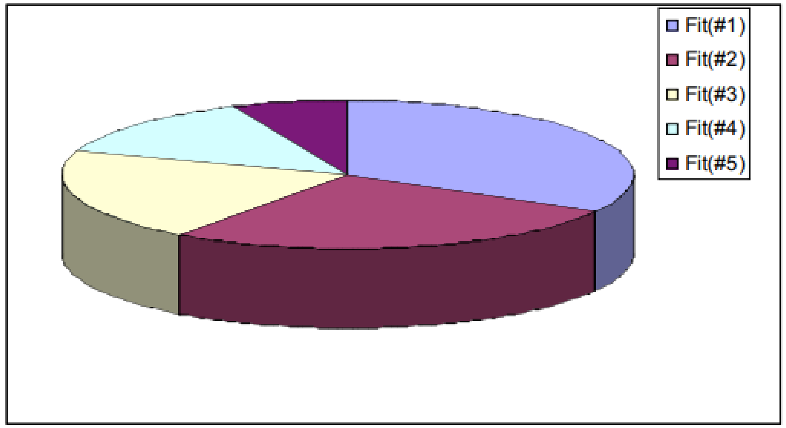
Characteristics of Fitness Function

A fitness function should possess the following characteristics –

- The fitness function should be sufficiently fast to compute.
- It must quantitatively measure how fit a given solution is or how fit individuals can be produced from the given solution.

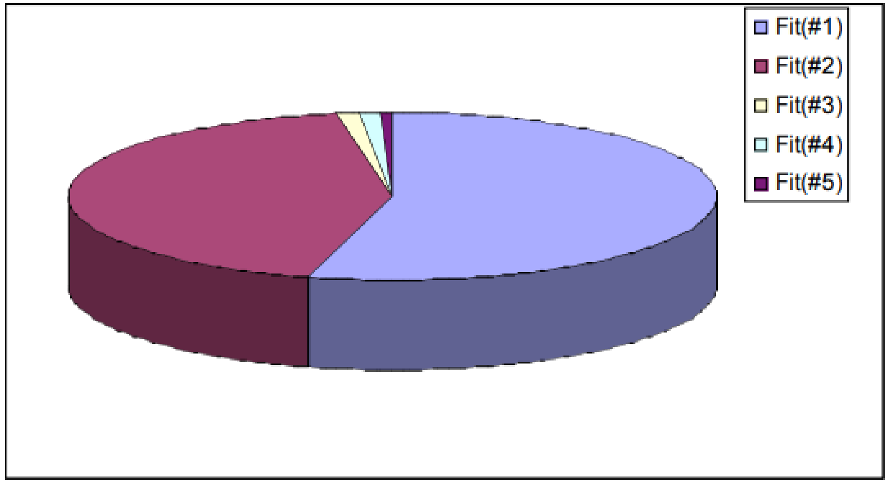
Roulette Wheel Selection

- Probability of parenthood is proportional to fitness.
- The wheel is spun until two parents are selected.
- The two parents create one offspring.
- The process is repeated to create a new population for the next generation.



Roulette Wheel Selection

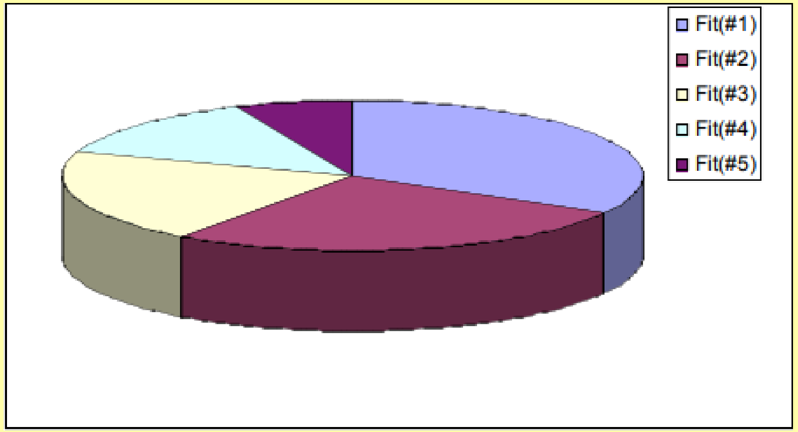
- Probability of parenthood is proportional to fitness.
- Roulette wheel selection has problems if the fitness changes by orders of magnitude.
- If two individuals have a much higher fitness, they could be the parents for every child in the next generation.



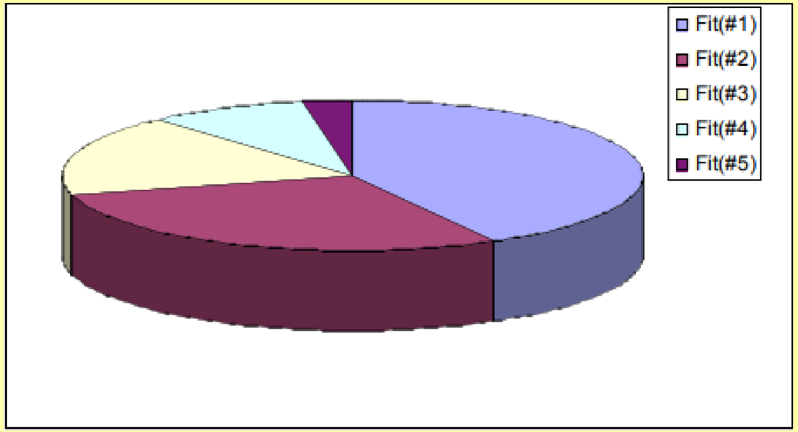
Rank Selection

- All individuals in the population are ranked according to fitness.
- Each individual is assigned a weight inversely proportional to the rank (or other similar scheme).

$$\frac{1}{rank}$$

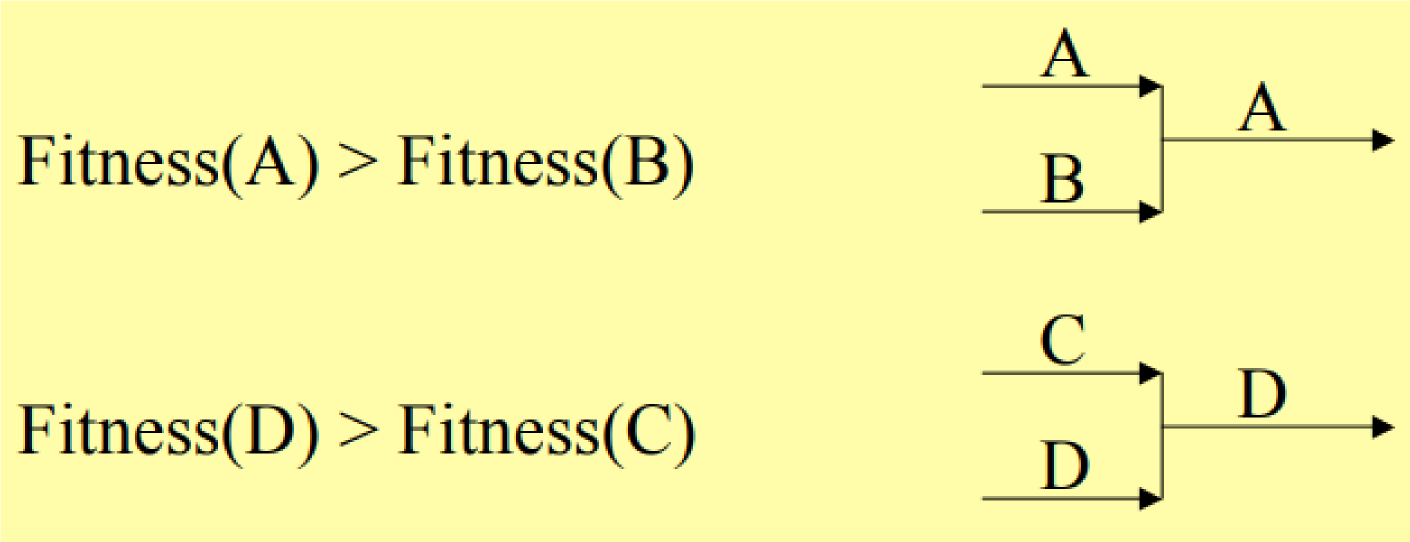


$$\frac{1}{(rank)^{1.7}}$$



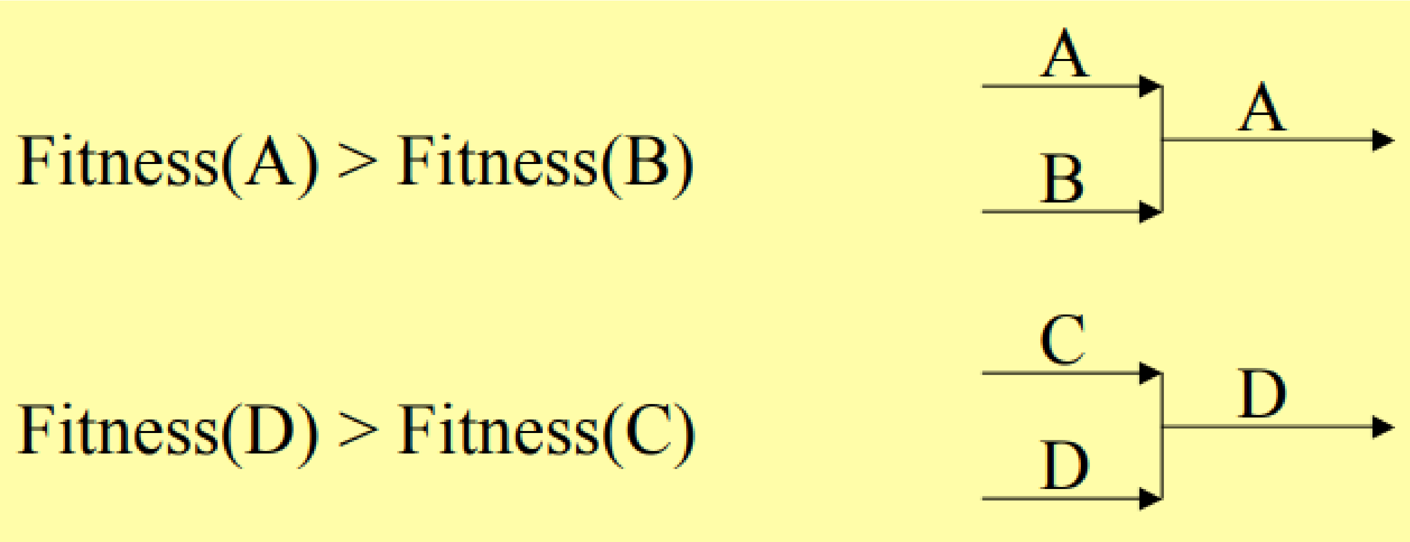
Tournament Selection

- 4 individuals (A,B,C,D) are randomly selected from the population.
- Two are eliminated and two become the parents of a child in the next generation



Tournament Selection

- Selection of parents continues until a new population is completed.
- Individuals might be the parent to several children, or no children.



Advantages of Genetic Algorithm

- Faster and more efficient as compared to the traditional methods.
- Optimizes both continuous and discrete functions and also multi-objective problems.
- Provides a list of “good” solutions and not just a single solution. Always gets an answer which gets better over the time.
- Useful when the search space is very large and there are a large number of parameters involved.

RAMAIAH Institute of Technology

Limitations of Genetic Algorithm

- Fitness value is calculated repeatedly which might be computationally expensive for some problems.
- Being stochastic, there are no guarantees on the optimality or the quality of the solution.
- If not implemented properly, the GA may not converge to the optimal solution.

Niching

- So far we have discussed that look into optimization of just one parameter (unimodal).
- There are several complex problems where in a set of parameters (multimodal) that may be dependent on each other is to be optimized.
- Such are known to use niching. Multi-objective optimization problems.
- Niching is a concept based on the nature’s ecosystems.
- In the real world for every niche, the physical resources available remain finite. Thus, they have to be shared among the individuals of the population that infest niche.

Niching

- Likewise in a **multi optima (multi-modal) search space**, a niche forms one optimum point and the concept of **fitness** refers to the **resources available**.
- Thus all individuals within this niche have to share their fitness with others (niching).
- **Fitness sharing** is a form of **niching strategy**.
- Another form of niching strategy termed as **Crowding**, an **offspring replaces the most similar individual** taken from a randomly drawn subpopulation of size Cf called the **crowding factor**.

Speciation

- Speciation is the development of one or more species from existing ones.
- One may imagine the formation of diverse yet stable populations that can occupy different regions of a search space.
- This would mean that part of the population is near one optimum while the other portions have hit similar ones elsewhere in the search space.
- Speciation is commonly realized by imposing restriction in crossover. A distance is found between two individuals. If this distance is less than the predetermined threshold value, the two are allowed to crossover.

Evolving Neural Networks

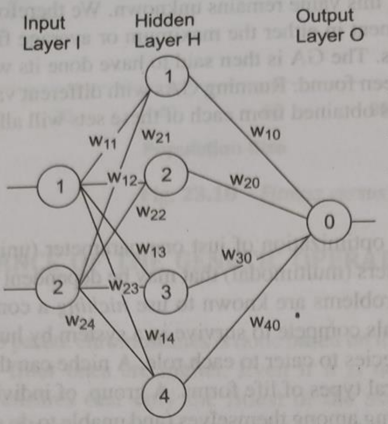
- Though neural networks have been used in a wide variety of applications, they suffer from some basic drawbacks.
- We look at two of the major problems faced while using ANN and see how genetic algorithms can be used to overcome them.

a) Network Topology Selection

- The very first problem one encounters while dealing with ANN is that of topology selection.
- How do we know that a particular network configuration (in terms of layers and neurons per layer) is the best?
- Most of the time the network configuration is decided based on some personal thumb rules or heuristics pertaining to the problem at hand.

Evolving Neural Networks

- As always, the search for a solution in the world of GA begins with some known solutions, which in this case is a set of network topologies which are encoded in a chromosome.
- One possible way of structuring the chromosome is shown in the figure.



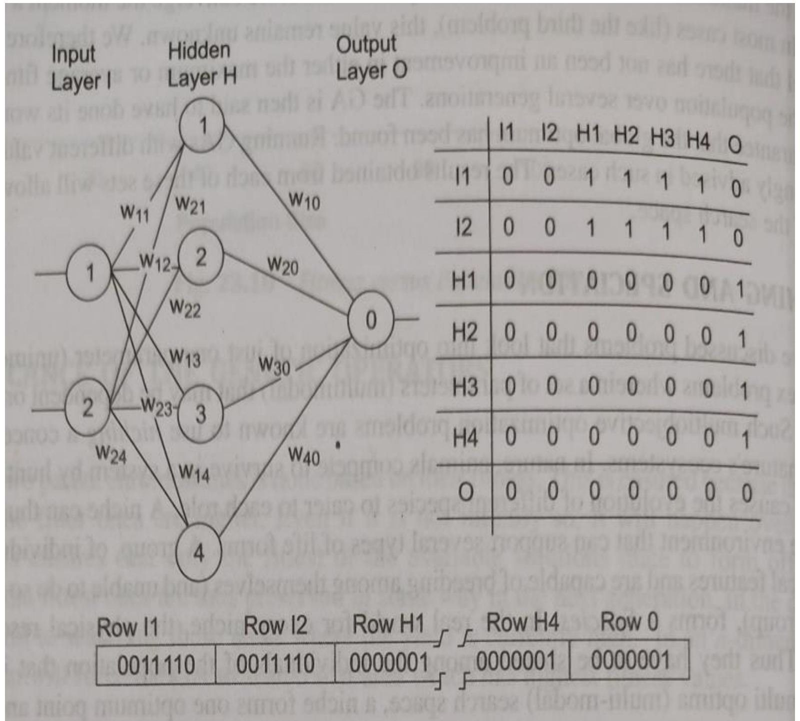
	I1	I2	H1	H2	H3	H4	O
I1	0	0	1	1	1	1	0
I2	0	0	1	1	1	1	0
H1	0	0	0	0	0	0	1
H2	0	0	0	0	0	0	1
H3	0	0	0	0	0	0	1
H4	0	0	0	0	0	0	1
O	0	0	0	0	0	0	0

Row I1	Row I2	Row H1	Row H4	Row 0
0011110	0011110	0000001	0000001	0000001

Fig. 23.11 *A Typical Artificial Neural Network and a Possible Method of Representing its Weights as a Chromosome*

a) Network Topology Selection

- Figure shows the manner in which the various neurons in the layers are connected.
- A ‘1’ indicates a connection while a ‘0’ indicates otherwise. Note that use of such a chromosome does limit the search to a finite number of neurons.
- One may increase the number of layers which will make the chromosome to contain longer sequences.
- We must bear in mind that the more the information contained within the chromosome, the more the computational time required to complete the search.



	I1	I2	H1	H2	H3	H4	O
I1	0	0	1	1	1	1	0
I2	0	0	1	1	1	1	0
H1	0	0	0	0	0	0	1
H2	0	0	0	0	0	0	1
H3	0	0	0	0	0	0	1
H4	0	0	0	0	0	0	1
O	0	0	0	0	0	0	0

Fig. 23.11 *A Typical Artificial Neural Network and a Possible Method of Representing its Weights as a Chromosome*

a) Network Topology Selection

- Now that the chromosomes is in place, we need to find a fitness function to evaluate them.
- A good fitness function should take into account network compactness, accuracy and learning rate.
- Considering all of them also contribute to computation costs.

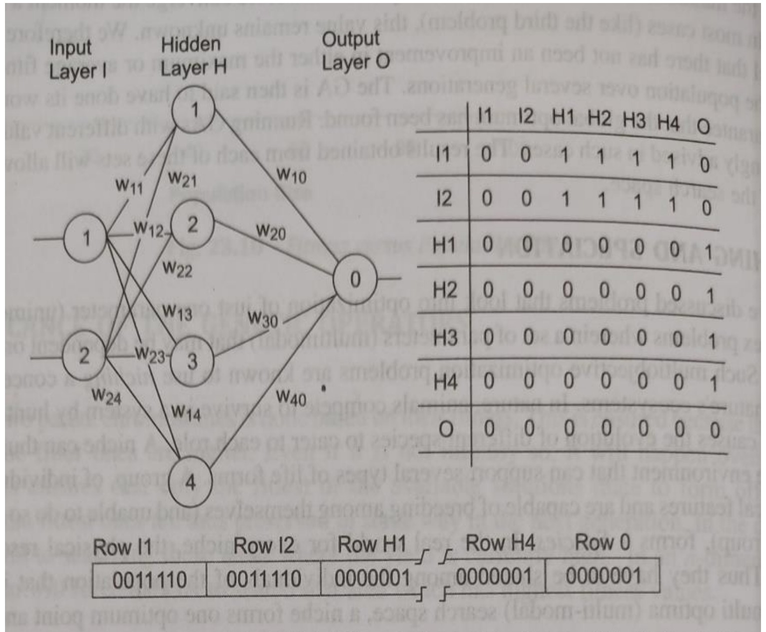


Fig. 23.11 *A Typical Artificial Neural Network and a Possible Method of Representing its Weights as a Chromosome*

b) Finding the optimal set of weights

- One of the serious limitations of the most commonly used backpropagation algorithm is that it cannot guarantee an optimal convergence.
- In quite a few cases the algorithm manages to provide weights that lead to only a sub-optimal solution. Further, there is no method to recover from such local optima. Using a GA to find the set of optimal weights can help solve this problem to a great extent.

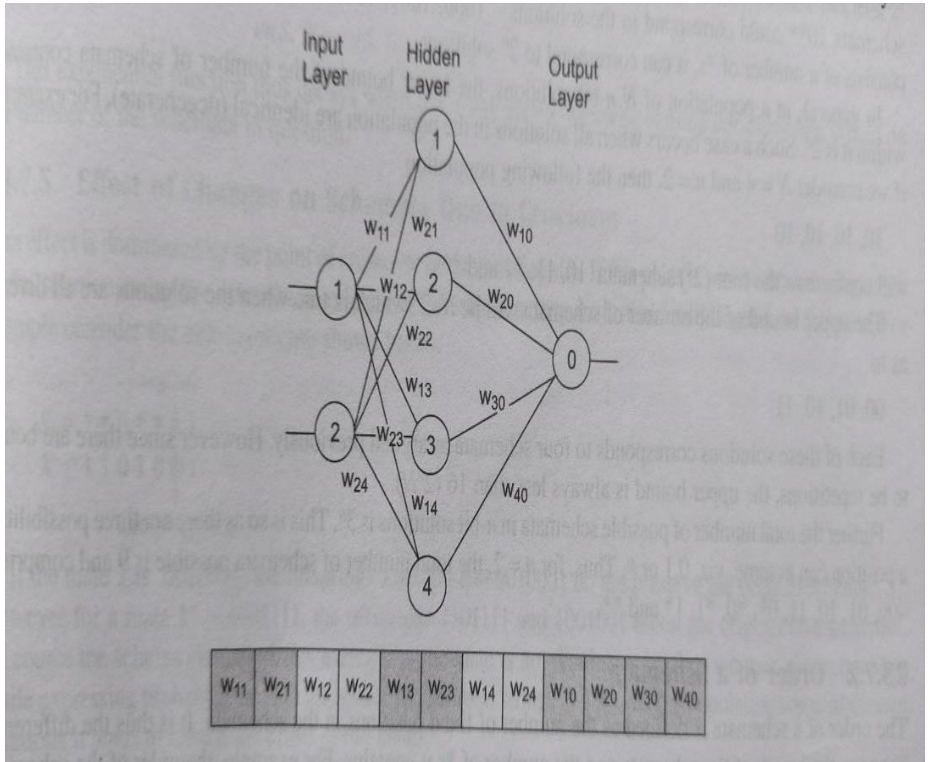


Figure 23.12 A Typical Artificial Neural Network and its Corresponding Chromosome Representation (The darkened lines indicate the gene boundaries)

b) Finding the optimal set of weights

- The chromosome in this case could be an ordered chain of weights. One such is depicted in figure for the network shown.
- Notice that each gene comprises the weights of the arcs that connect the neuron of a layer to those of its previous layer.

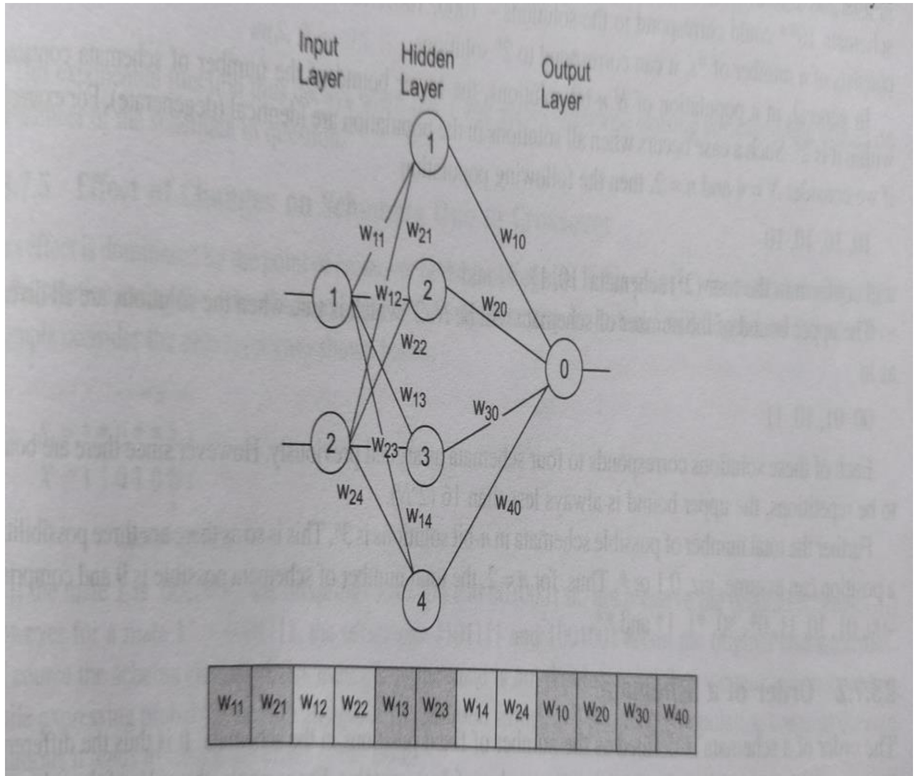


Figure 23.12 *A Typical Artificial Neural Network and its Corresponding Chromosome Representation (The darkened lines indicate the gene boundaries)*

Theoretical Grounding

- Though the basic rudiments have been copied from nature, a theoretical proof always helps comprehension and refinement of these algorithms.
- It is thus essential to understand the theory on which these algorithms rest.
- Below are the few pertinent terms:

1. Schemata:

- Sometimes referred to as similarity templates, schemata are basically the patterns of the solutions.
- For instance, if two solutions were represented in binary as 11011 and 01010, we observe that the inner three bits are the same in both the solutions.
- This means that both these solutions conform to the schemata *101* where the asterisks indicate don't cares (0 or 1).

Theoretical Grounding

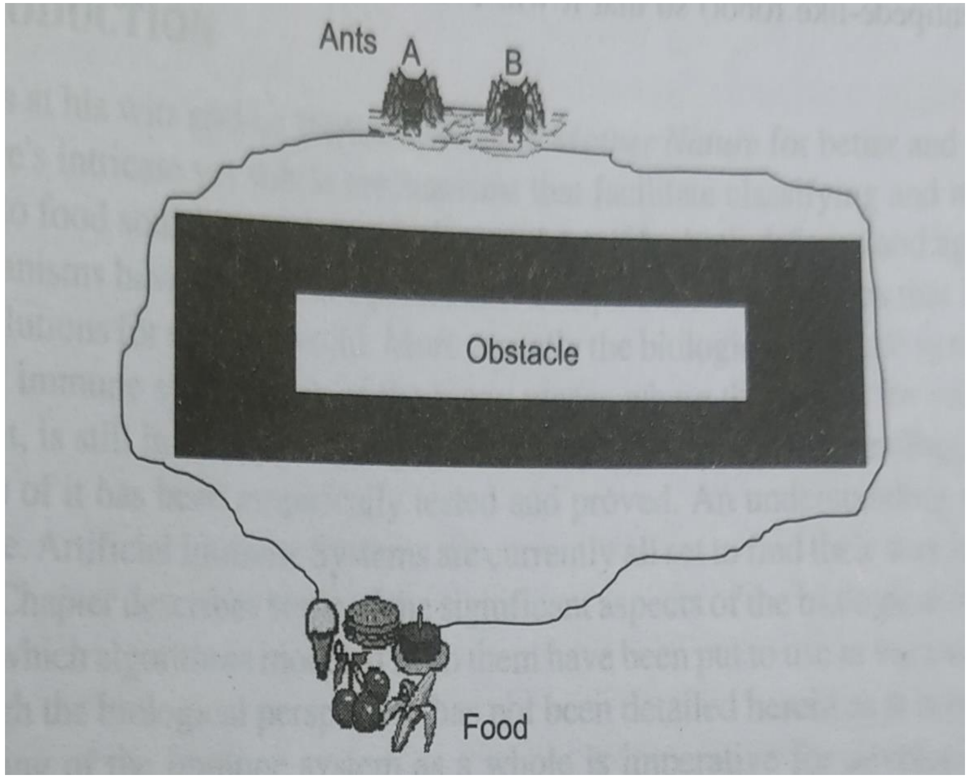
2. Order of Schemata:

- The order of a schemata is defined as the number of fixed positions in the schemata. It is thus the difference between the length of the schemata and the number of *s it contains.
- For example, the order of the schemata 10**1 is given by o(10**1)=5-2=3

RAMAIAH
Institute of Technology

Ant algorithm

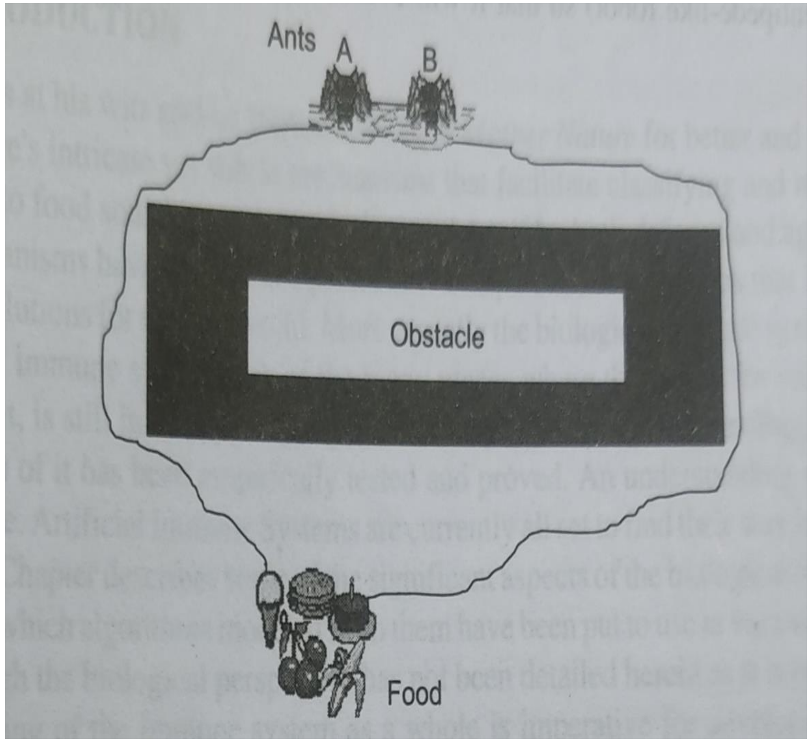
- Here we take a brief look at the manner in which one of nature's creations – Ants – can trigger our search for new algorithms.
- Ants are capable of navigating complex terrains in search of food.
- They also find their way back to the nest.
- Over a period of time a colony of ants are able to find the best or shortest path between the food source and the nest.
- So how do they achieve this?



RAMAIAH
Institute of Technology

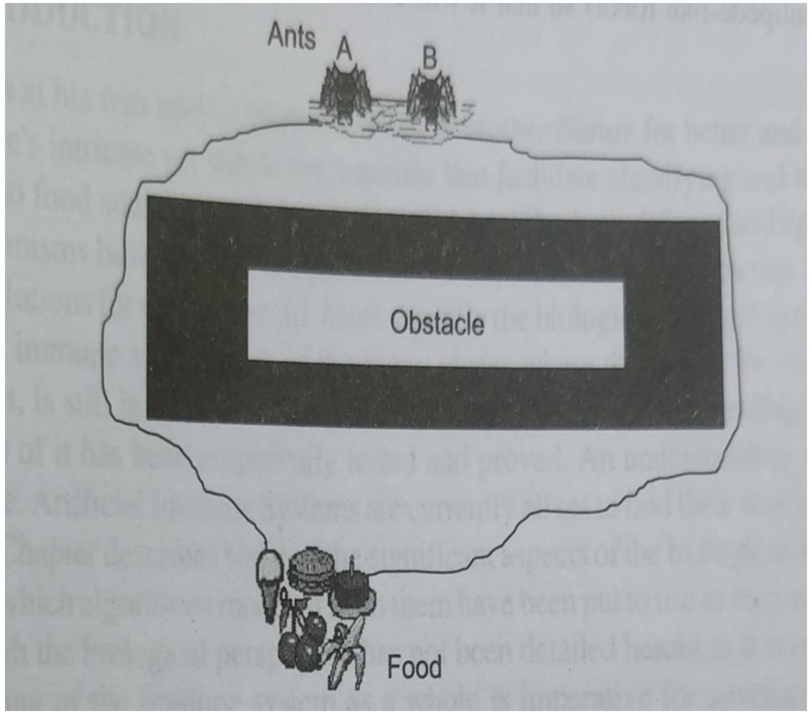
Ant algorithm

- As they navigate they keep laying pheromones which tend to modify their environment and serves as a means for communication amongst them in the colony.
- Pheromones are chemicals that are volatile and give way over a period of time.
- All ants choose to move over tracks of high pheromone concentration.
- In the beginning each ant goes in search of food and as they move the pheromone is laid along the path. When an ant finds the food source it starts its return journey along the same path and adds the pheromone concentration along it.



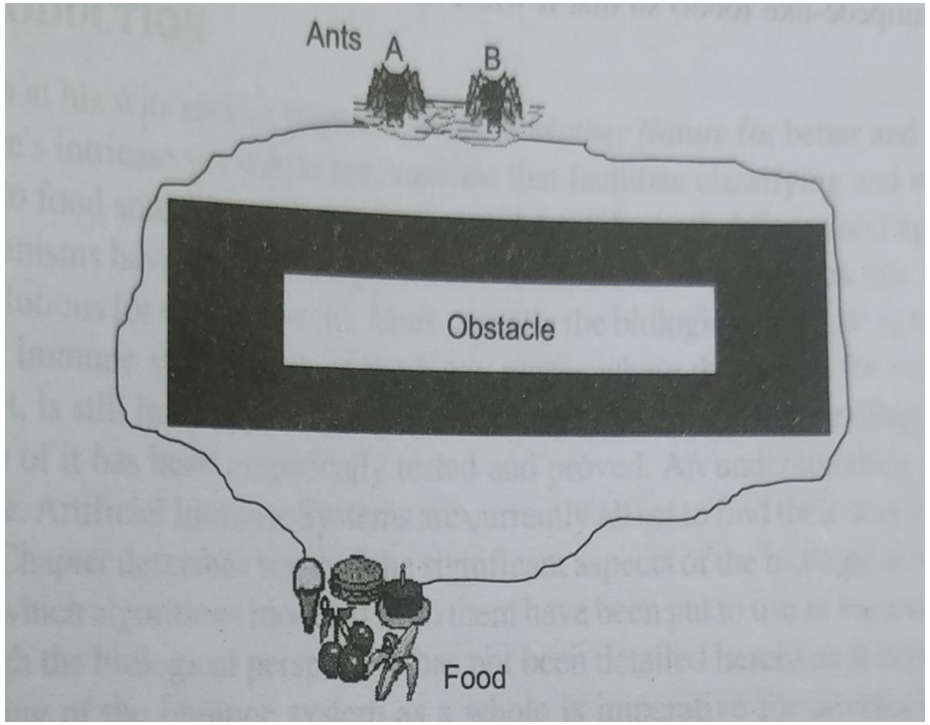
Ant algorithm

- Since the colony comprises a large number of ants, a parallel search ensues.
- Chances are that several of them discover the food source through different paths.
- Naturally the ant that found the closest path would over a period of time shuttle up and down more number of times than its counterparts.
- This increases the pheromone concentration of the shortest path and forces other ants too choose it.
- Over a period of time only the shortest path exists while other paths fade away due to pheromone volatility.



Ant algorithm

- Figure depicts two ants A and B using this technique.
- Ant A will definitely reach back to the starting point faster than ant B, thereby depositing more pheromone over a period of time forcing the other to also follow the shortest route discovered.
- This algorithm can be easily ported onto the well known Travelling Salesman problem.



Summary

- GA is a stochastic search algorithm based on principles of natural competition between individuals for appropriating limited natural sources.
- Genetic Algorithms with ACO (Ant Algorithms) are an example of Mimicking the Biological World.
- They have many proven applications in Computational Domains. (Job-Shop, Optimization Problems : Skill Based employee Allocation etc.
- There are three basic Operations of GA: Reproduction, Crossover, Mutation.
- Though Neural Networks have been used in a wide variety of Applications, they suffer from some basic drawbacks. These definitions can be handled by the GA techniques.
- **Ant system** (AS) was the first ACO algorithm to be proposed in the literature (Dorigo et al. 1991, Dorigo 1992, Dorigo et al. 1996). Its main characteristic is that the pheromone values are updated by all the **ants** that have completed the tour.
- **Swarm intelligence** is the discipline that deals with natural and artificial systems composed of many individuals that coordinate using decentralized control and self-organization

Sample Questions

1. Give an example of combinatorial problem. What is the most difficult in solving these problems?
2. What two requirements should a problem satisfy in order to be suitable for solving it by a GA?
3. Consider the problem of finding the shortest route through several cities, such that each city is visited only once and in the end return to the starting city (the Travelling Salesman problem). Suppose that in order to solve this problem we use a genetic algorithm, in which genes represent links between pairs of cities. For example, a link between London and Paris is represented by a single gene 'LP'. Let also assume that the direction in which we travel is not important, so that LP = PL.
   - How many genes will be used in a chromosome of each individual if the number of cities is 10?
   - How many genes will be in the alphabet of the algorithm?

Sample Questions

4. Suppose a genetic algorithm uses chromosomes of the form x = abcdefgh with a fixed length of eight genes. Each gene can be any digit between 0 and 9. Let the fitness of individual x be calculated as: $f(x) = (a + b) - (c + d) + (e + f) - (g + h)$, and let the initial population consist of four individuals with the following chromosomes:

   x1 = 6 5 4 1 3 5 3 2

   x2 = 8 7 1 2 6 6 0 1

   x3 = 2 3 9 2 1 2 8 5

   x4 = 4 1 8 5 2 0 9 4

a) Evaluate the fitness of each individual, showing all your workings, and arrange them in order with the fittest first and the least fit last.

b) Perform the following crossover operations:

i) Cross the fittest two individuals using one–point crossover at the middle point.

ii) Cross the second and third fittest individuals using a two–point crossover (points b and f).

iii) Cross the first and third fittest individuals (ranked 1st and 3rd) using a uniform crossover.

Thank you