## LAB # 13

##### IMPLEMENTING METAHEURISTIC ALGORITHM

(Genetic Algorithm)

###### OBJECTIVE

Solving Coin Toss Problem by implementing Genetic Algorithm (GA).

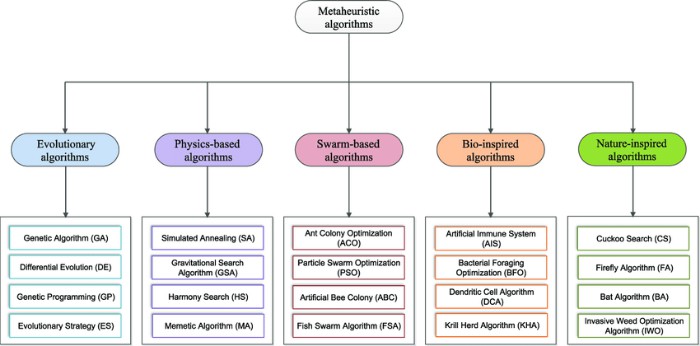
###### THEORY

META-HEURISTIC ALGORITHMS:

In computer science a **meta-heuristic** is a higher-level procedure to find, generate, or select a heuristic (partial search algorithm) that may provide a sufficiently good solution to an optimization problem, especially within complete or imperfect information or limited computation capacity.

Meta-heuristics sample, a set of solutions which is too large to be completely sampled. In combinatorial optimization, by searching over a large set of feasible solutions, meta-heuristics can often find good solutions with less computational effort than optimization algorithms, iterative methods, or simple heuristics. They are useful approaches for optimization problems. It operates on those optimization problems in which the set of solutions is discrete or can be reduced to discrete, and in which the goal is to find the best solution. Some common problems are:

1. Coin Toss Problem (CTP).
2. Knap sack problem.



**Fig 1: Types of Metaheuristic Algorithms**

EVOLUTIONARY ALGORITHM:

* + An evolutionary algorithm is an [**evolutionary AI-based**](https://www.cognizant.com/glossary/evolutionary-computation.html)computer application that solves problems by employing processes that mimic the behaviors of living things. As such, it uses mechanisms that are typically associated with biological evolution, such as reproduction, mutation and recombination.
  + Evolutionary algorithms function in a Darwinian-like natural selection process; the weakest solutions are eliminated while stronger, more viable options are retained and re- evaluated in the next evolution—with the goal being to arrive at optimal actions to achieve the desired outcomes.

GENETIC ALGORITHM (GA):

Genetic Algorithm (GA) is a search-based optimization technique based on the principles of Genetics and Natural Selection. It is frequently used to find optimal or near-optimal solutions to difficult problems which otherwise would take a lifetime to solve. It is frequently used to solve optimization problems, in research, and in machine learning. Genetic algorithms have collections of solutions that are collided with each other to make new solutions, eventually returning the best solution. The genetic algorithm repeatedly modifies a population of individual solutions. At each step, the genetic algorithm selects individuals at random from the current population to be parents and uses them to produce the children for the next generation. Over successive generations, the population "evolves" toward an optimal solution.

Generates a population of points at each iteration. The best point in the population approaches an optimal solution. Selects the next population by computation which uses random number generators.

Five phases are considered in a genetic algorithm.

1. Initial population
2. Fitness function
3. Selection
4. Crossover
5. Mutation
6. Initial Population:

The process begins with a set of individuals which is called a Population. Each individual is a solution to the problem you want to solve.

An individual is characterized by a set of parameters (variables) known as Genes. Genes are joined into a string to form a Chromosome (solution).

In a genetic algorithm, the set of genes of an individual is represented using a string, in terms of an alphabet. Usually, binary values are used (string of 1s and 0s). We say that we encode the genes in a chromosome.

1. Fitness Function:

The fitness function determines how fit an individual is (the ability of an individual to compete with other individuals). It gives a fitness score to each individual. The probability that an individual will be selected for reproduction is based on its fitness score.

1. Selection:

The idea of selection phase is to select the fittest individuals and let them pass their genes to the next generation.

Two pairs of individuals (parents) are selected based on their fitness scores. Individuals with high fitness have more chance to be selected for reproduction.

1. Crossover:

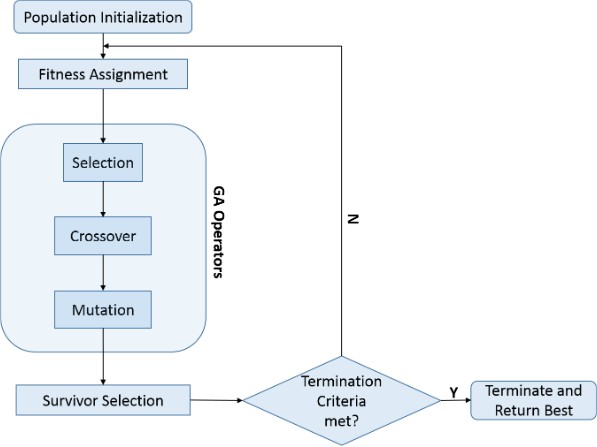
Crossover is the most significant phase in a genetic algorithm. For each pair of parents to be mated, a crossover point is chosen at random from within the genes.

Offspring are created by exchanging the genes of parents among themselves until the crossover point is reached.

1. Mutation:

In certain new offspring formed, some of their genes can be subjected to a mutation with a low random probability. This implies that some of the bits in the bit string can be flipped.

GA Flow:

****

**Fig 2: Flow of Genetic Algorithms**

GA Code (Coin Toss Problem):

S=[[0,0,1,1,1,1,1,1,1,1],[0,0,1,1,1,1,1,1,1,1],[0,1,1,0,1,1,0,1,1,1],[0,1,0,1,1,1,0,0,1,1],[0,0,1,0,1,1, 1,1,1,1],[0,0,1,0,1,1,0,1,1,1]]

print("S=",S) def calfitness(S):

fit=[] total=0

print("Fitness") for i in S:

fit.append(i.count(1)) total+=i.count(1) print(fit)

print(total) return fit;

fit=calfitness(S)

print("Arranging in Descending order") desc=S

for i in range(len(desc)): j=i + 1

for j in range(len(desc)):

if desc[i].count(1)>desc[j].count(1):

desc[j],desc[i]=desc[i],desc[j]

print("s =" , desc)

print("Cross over after 2 points") for i in range(3):

desc[0][i],desc[3][i]=desc[3][i],desc[0][i]

desc[4][i],desc[5][i]=desc[5][i],desc[4][i] desc[0],desc[3]=desc[3],desc[0]

desc[4],desc[5]=desc[5],desc[4] print("s1 and s4")

print("s1=", desc[0], "s4=", desc[3])

print("s5=", desc[4], "s6=", desc[5]) print("s=", desc)

print("Mutation") for i in range(6):

for j in range(2): if(desc[i][j]==0):

desc[i][j]=1 else:

desc[i][j]=0

print("s=",desc) fit2=calfitness(desc) if fit<fit2:

print("Fitness is greater after applying Genetic Algorithm") else:

print("Fitness is greater before applying Genetic Algorithm")

Lab Tasks:

1. Run the given code of Genetic Algorithm (Coin Toss Problem) and show the output.
2. In given code there are crossover functions implemented between S1, S4 and S5, S6 after 3rd point. You can perform crossover between S2 and S3 after 4th point.
3. In given code the values of chromosomes are hardcoded, you may take input values of chromosomes at run time.