Page No.
Date

ASSIGNMENT No. 1

Aim:
To apply the artificial immune pattern recognition to perform a task of structure damage classification.

Objective:
- To apply artificial immune pattern recognition techniques to classify & identify structural damage.
- The aim of improving the accuracy & reliability of damage detection in engineered systems.

Theory:
An Artificial Immune system (AIS) for the task of structural damage classification, we will focus on a simplified scenario where our goal is to classify structures are either 'damaged' or 'not damaged'. Artificial Immune systems are computational systems inspired by the principles and process of the biological immune system. They can particularly effective in pattern recognition tasks, including anomaly detection & classification change.

Step 1: Generating Data
Let's first generate some synthetic data for our task. For simplicity, we will create a dataset with features that might relate to structural integrity such as stress levels, vibrations frequencies & temperature readings.

Step 2 : Implementing the NSA
We will then implement a basic Negative Selection Algorithms, where we will generate doctors that do not match the normal data points. These doctors will then be used to identify 'damaged' structures.

Limitations :
The simplicity of this model means it may not handle complex structural damage scenarios without modifications & additional features engineering.

Conclusion :
An artificial immune pattern recognition to perform a task of structure damage classification proves to be promising approach.

Page No.
Date

ASSIGNMENT NO. 2

Aim:
Implement DEAP (Distributed Evolutionary Algorithms) using python.

Objectives:
- Create custom evolutionary algorithms, such as genetic algorithm, genetic programming, or evolutionary strategies.
- Evaluate the performance of algorithms through experimentation & benchmarking
- Explore parallel & distributed computing to improve computational efficiency.

Theory:
Distributed Evolutionary Algorithms are computational models inspired by natural evolution. They aim to solve optimization problems by mimicking the process of selection, mutation and crossover. This algorithms often involve multiple subpopulations that evolve in parallel and occasionally exchange individuals, making the system scalable and efficient.
The DEAP framework is a popular python library for implementing evolutionary algorithms. It provides tools to create genetic algorithms, evolutionary strategies, & other population-based optimization methods

Conclusion :

By implementing Distributed Evolutionary Algorithms in Python using DEAP, the experiment demonstrates the potential of evolutionary algorithms to solve optimization problems efficiently.

ASSIGNMENT No. 3

Page No.
Date

Aim: Design & develop a distributed application to find the coolest / hotest year from the available weather data. Use weather data from the Internet & process it using MapReduce.

Objectives:
- To design and develop a distributed applications that process weather data using the MapReduce framework to determine the coolest & hottest years based on temperature records.
- The application should efficiently handle large datasets & extract meaningful insights from weather data retrieved from the internet.

Theory:

1. MapReduce Framework:
   - MapReduce is a programming model for processing and generating large datasets in a distributed manner.
   - It consist of two main functions:
     - Map Function: Process input data & produces intermediate key-value pairs.
     - Reduce Function: Aggregates & process the intermediate data to produce the final result.

2. Weather Data Analysis:
   - weather data typically contains fields like date, temperature, location & other attributes.
   - To determine the coolest & hottest years:

- Extract year & temperature fields from the dataset.
- For each year, compute the maximum & minimum temperatures.

3. Distributed Processing :
   - In a distributed system, large datasets are divided into smaller chunks & processed on multiple nodes.
   - MapReduce ensures scalability, fault tolerance & efficient computations by parallelizing tasks across these nodes.

Conclusion :
Using MapReduce enables processing large-scale weather datasets efficiently in a distributed environment.

Page No.
Date

ASSIGNMENT NO. 4

Aim: Implement Ant colony optimization by solving the travelling salesman problem using python
problem statement - A salesman needs to visit a set of cities exactly once & return to the original city. The task is to find the shortest possible route that the salesman can take to visit all the cities & return to the starting city.

Objectives:
- To implement the Ant colony optimization algorithm to solve the Travelling salesman problem using python.
- The goal is to find the shortest possible route for the salesman to visit all cities exactly once & return to the starting city.

Theory:
1. Travelling Salesman Problem:
- A combinatorial optimization problem where a salesman must visit a given set of cities exactly once & return to the starting city.

2. Ant Colony Optimization:
- ACO is a bio-inspired optimization technique based on the behaviours of ants.
- Ants deposite pheromones on paths they traverse. Over time, shorter paths accumulate more pheromones, attracting more ants & reinforcing the solution.

Conclusion :

The ACO algorithm successfully finds a near-optimal solution to the TSP problem by iteratively refining the pheromone trails.