

**GENERAL SIR JOHN KOTELAWALA DEFENCE UNIVERSITY**

**Nature Inspired Computing**

**CS – 3192**

**Reflective Journal**

**Name : G.S.M.Jayasoriya**

**Reg No. : D / BCS / 22 / 0015**

TABLE OF CONTENTS

[INTRODUCTION 3](#_Toc185508722)

[GENETIC ALGORITHM 3](#_Toc185508723)

# **INTRODUCTION**

Nature-Inspired Computing (NIC) is a field of computer science where we create algorithms and systems based on how nature works. It takes ideas from things like how animals behave, how evolution happens, or how physical processes work, to solve complex problems efficiently.

Nature-Inspired Optimization Algorithms (NIOS) are algorithms designed to find the best solution to a problem by mimicking natural processes or behaviors. They are particularly useful for solving complex optimization problems where traditional methods might fail. In this journal, let’s explore how NIOAs are applied in various industries, reflecting on their impact and potential to drive innovation.

This journal focuses on real world applications of four widely used Nature-Inspired Optimization Algorithms:

* Genetic Algorithm (GA)
* Simulated Annealing (SA)
* Particle Swarm Optimization (PSO)
* Ant Colony Optimization (ACO)

These algorithms have been successfully applied in real-world problems, such as optimizing transportation routes, improving resource allocation, designing efficient networks, and scheduling complex tasks. Their ability to find near-optimal solutions in challenging scenarios highlights their importance and relevance in modern problem-solving.

# 

# **GENETIC ALGORITHM (GA)**

Genetic Algorithm (GA) is a heuristic optimization technique inspired by the process of natural selection in biological evolution. It is part of a class of evolutionary algorithms that solve complex problems by mimicking the principles of survival of the fittest. As work by generating a population of candidate solutions and evolving them over generations to find the optimal or near-optimal solution.

The core steps of GA include:

* ***Initialization:*** Generate an initial population of solutions randomly.
* ***Evaluation:*** Assess each solution using a fitness function.
* ***Selection:*** Choose the best solutions for reproduction.
* ***Crossover:*** Combine pairs of solutions to create offspring.
* ***Mutation:*** Introduce small random changes to offspring for diversity.
* ***Termination:*** Repeat the process until a stopping criterion is met

Selection is a crucial step in GAs that determines which individuals from the current population will contribute to the next generation. Common selection methods include:

* Roulette Wheel Selection
* Tournament Selection
* Rank-Based Selection

For the same real world application of GA, I plan to solve the same problem using ***two different selection methods***, Roulette Wheel Selection and Tournament Selection.

***PROBLEM STATEMENT***