

# Enhanced Whale Optimization Algorithm for Dependent Tasks Offloading Problem in Multi-edge Cloud Computing

Atluri Charitha Sri (B220213CS)  
Bandapalli Venkata Durga Sahithi (B220753CS)  
K Triveni (B220935CS)

CS4037D Cloud Computing

15<sup>th</sup> November 2025

B.Tech, Computer Science and Engineering  
National Institute of Technology, Calicut

# Overview

- ① Introduction
- ② Problem Definition
- ③ System Model
- ④ Methodology
- ⑤ Algorithms
- ⑥ Flowchart of EWA Algorithm
- ⑦ Implementation
- ⑧ Numerical Experiments and Results
- ⑨ Conclusion

# Introduction

- High processing power is required for the growing use of applications like Computer Vision, Virtual Reality etc.
- The battery life and resources of Internet of Things (IoT) and Internet of Vehicles (IoV) devices are limited.
- Although Cloud Computing is powerful, it has high latency.

# Introduction

- By processing tasks close to devices, Mobile Edge Computing (MEC) lowers latency.
- In order to reduce Latency, Cost, and Energy Consumption, the proposed Enhanced Whale Optimization Algorithm (EWA) enhances task offloading across MECs and Cloud.

# Problem Definition

- In a Mobile Edge Computing (MEC) environment, applications are often composed of multiple interdependent tasks.
- Efficiently deciding which tasks to offload to edge servers or the central cloud, while minimizing latency, energy consumption, and resource utilization cost, is a complex multi-objective optimization problem.
- Therefore, the problem is to develop an enhanced optimization algorithm (EWA) capable of determining optimal task-offloading strategies for interdependent tasks in a MEC and cloud system, achieving lower delay, reduced energy use, and balanced resource utilization.

# System Model

- Tasks can run locally, on MECs, or in the Cloud.
- Smartphones receive data from IoT devices and process dependent tasks.

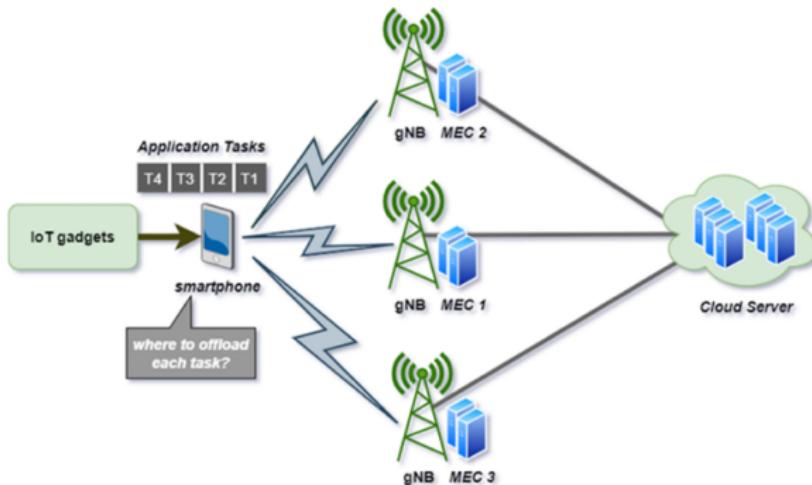


Figure 1: Proposed System Architecture.

# Methodology

- To maximize the offloading of interdependent tasks across cloud servers and multi-edge MEC, the Enhanced Whale Optimization Algorithm (EWA) was proposed.
- A Directed Acyclic Graph (DAG) of tasks was created to represent dependencies.
- EWA uses enhanced WOA operations, such as Frameshifting (FS) and Load Redistribution Strategy (LRS), to dynamically optimize latency, energy, and cost (L, E, UC).
- Introduced linear scaling and TLRS as normalization techniques to transform continuous vectors into discrete values.

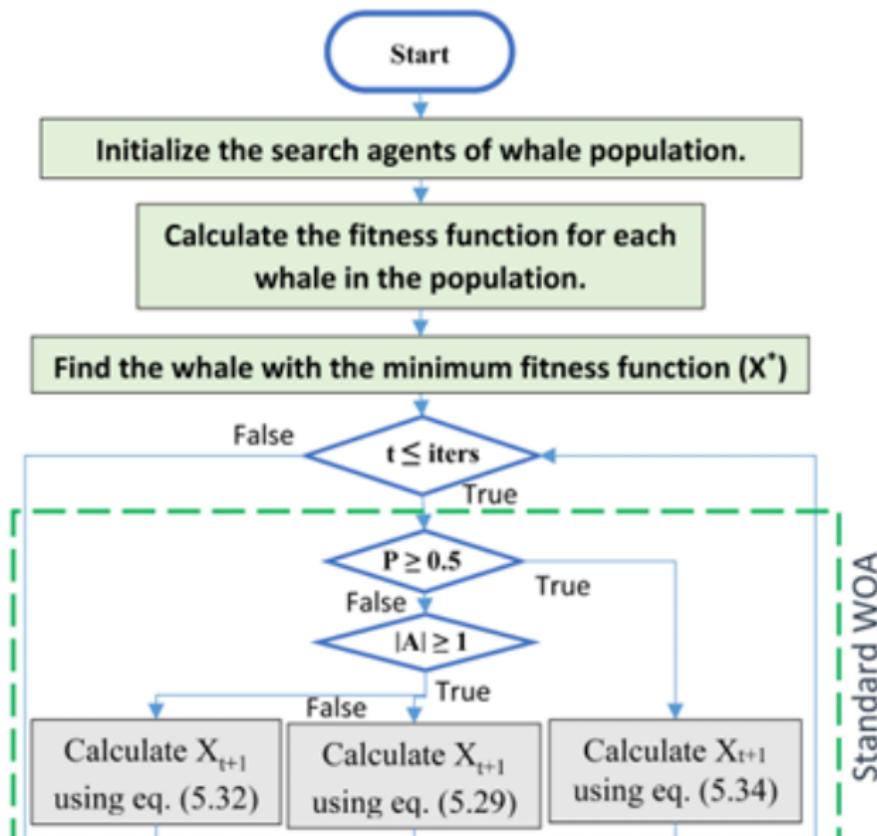
# Flow of the Execution

- Step 1: Initialize population: generate random continuous whales.
- Step 2: Convert to discrete numbers: TLRS for task order and linear scale for locations.
- Step 3: Calculate objective  $Z = w_1 \cdot L + w_2 \cdot E + w_3 \cdot UC$  for ranking.
- Step 4: Apply WOA updates (explore/encircle/spiral) on continuous population.
- Step 5: Normalize updated whales back to discrete agents (TLRS + scaling).
- Step 6: Apply enhancements: Frameshifting and Load Redistribution.
- Step 7: Replace population entries with improved best agent and repeat.
- Step 8: Record convergence: store best  $Z$  per iteration for plotting.
- Step 9: Compare EWA vs baseline (WOA) and produce latency, energy, cost, and fitness plots.

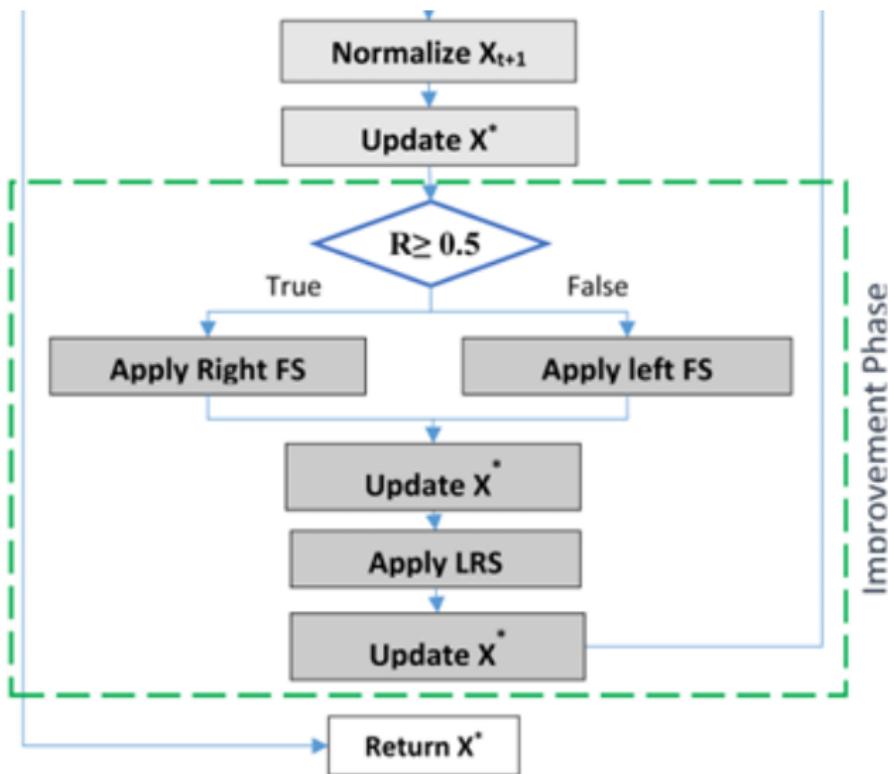
# Algorithms Used

- The initialization algorithm maintains task dependencies while creating random whale agents (solutions) with tasks and processing locations.
- By moving jobs from high-load servers to low-load servers, the load redistribution strategy (LRS) balances workload and reduces overall execution time.
- By adding Frameshifting and LRS to Whale Optimization, the Enhanced Whale Optimization Algorithm (EWA) enhances task offloading.

# Flowchart of EWA Algorithm



# Flowchart of EWA Algorithm



# Implementation

- Implemented the Enhanced Whale Optimization Algorithm (EWA) in Python, simulating the experimental setup and workflow.
- Tested synthetic task-offloading DAGs.
- Evaluation metrics: Fitness function for thorough performance analysis, execution latency, energy consumption, and usage cost.

# Numerical Experiments and Results

- The execution latency is improved by 16.43%, the energy consumption is reduced by 78.28%, the imposed cost is minimized by 35.02% by using FS and LRS Operations.

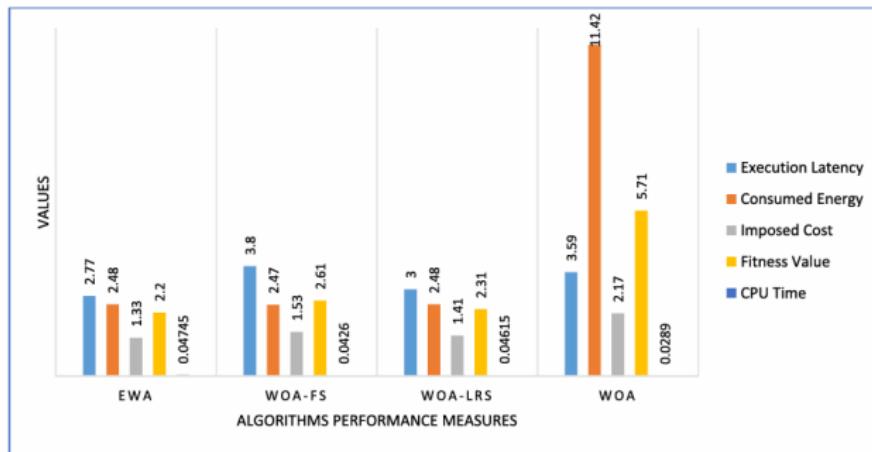


Figure 2: Performance measures values for running different derivatives of the WOA algorithm.

# Numerical Experiments and Results

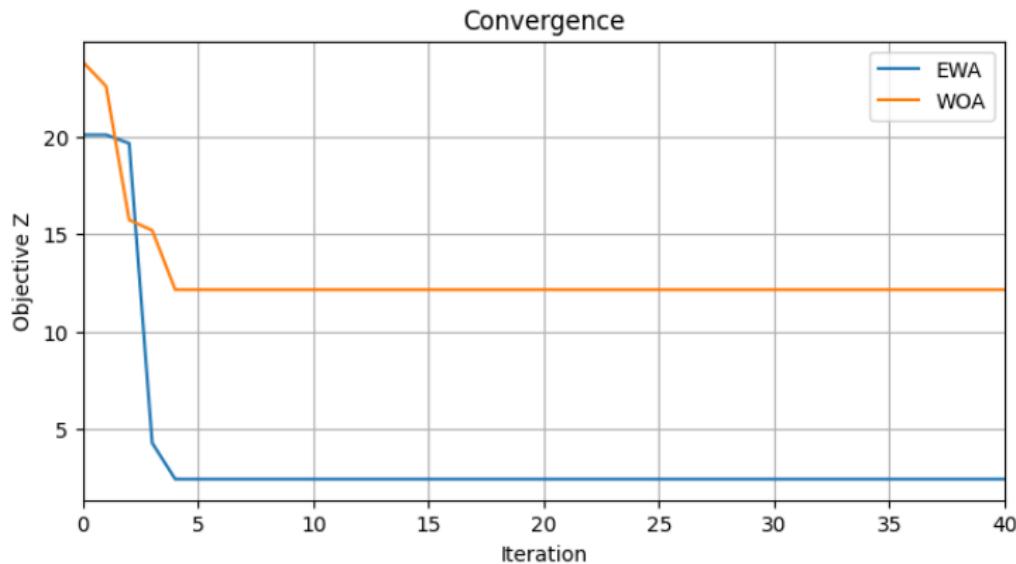


Figure 3: The convergence graph for Z value of both EWA and WOA

# Extension

- Real mobile users change position over time, so the distance to the servers changes.
- Channel quality ( $G_k$ ), which is a function of the distance, becomes dynamic instead of static.
- As a result, transmission time fluctuates, which affects offloading decisions.
- Original EWA assumes fixed channel gain, which is unrealistic for mobile scenarios.

# Conclusion

- In Multi-Edge Cloud environments with dependent tasks, the suggested Enhanced Whale Optimization Algorithm (EWA) optimized Cost, Energy Consumption, and Latency.
- To increase task scheduling efficiency and load balance, EWA combined Frame Shifting (FS) and Load Redistribution Strategy (LRS).

# Thank You