Parallel Grey Wolf Optimizer

High Performance Computing for Data Science Project
2024/2025A

Introduction

What is Optimization?

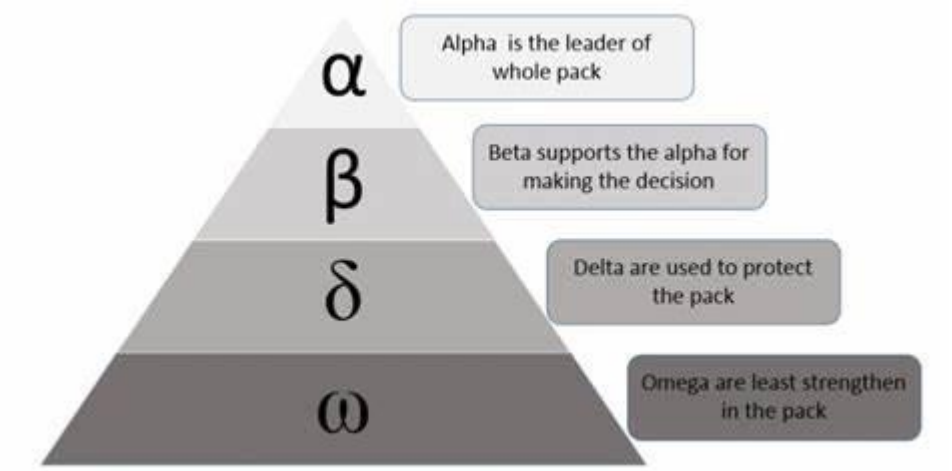
- Process of finding the best solution from a set of possible solutions.

Why Metaheuristic Algorithms?

- Sutable for complex, high-dimensional, or non-differentiable problems.

Grey Wolf Optimizer (GWO)

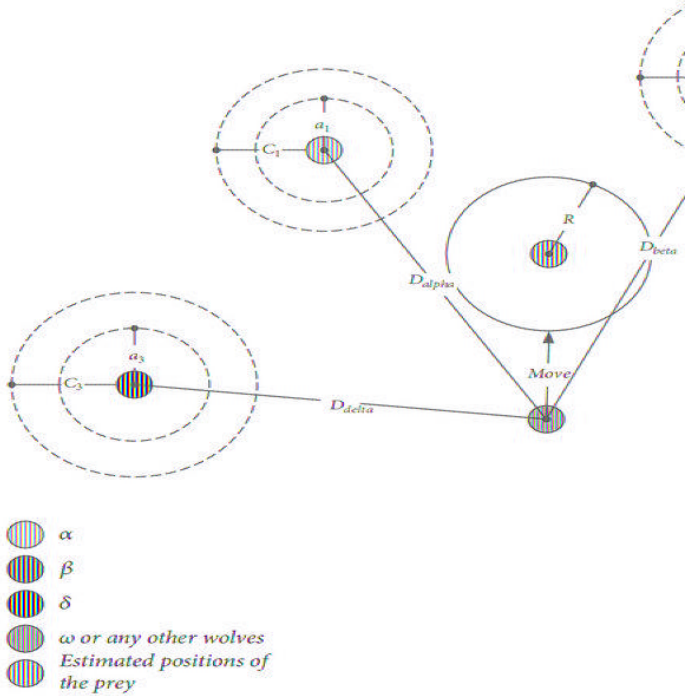
- Inspired by the social hierarchy, hunting behavior of grey wolves.
- Balances exploration and exploitation effectively.
- Useful in machine learning, engineering, and feature selection.



Grey Wolf Optimization (GWO) Overview

Key Components:

- Alpha (α), Beta (β), Delta (δ), and Omega (ω) wolves.
- Simulates hunting behavior: Encircling, Hunting, and Attacking Prey.
- Position updates using mathematical equations to improve solutions.



Why Parallel GWO?

Challenges in Sequential GWO:

- Computationally expensive for high-dimensional problems.

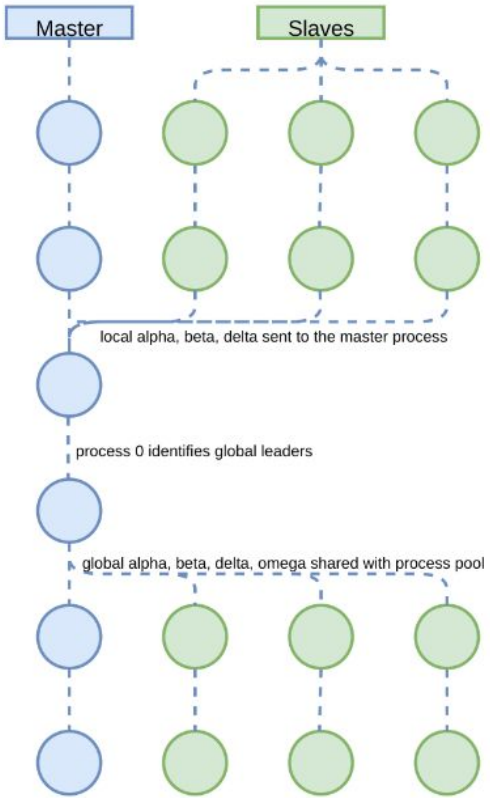
Parallelization Approach:

- Uses **MPI framework** to distribute computations across multiple processors.
- Improves scalability and efficiency.

Parallelization Strategy (1)

Master-Slave Model:

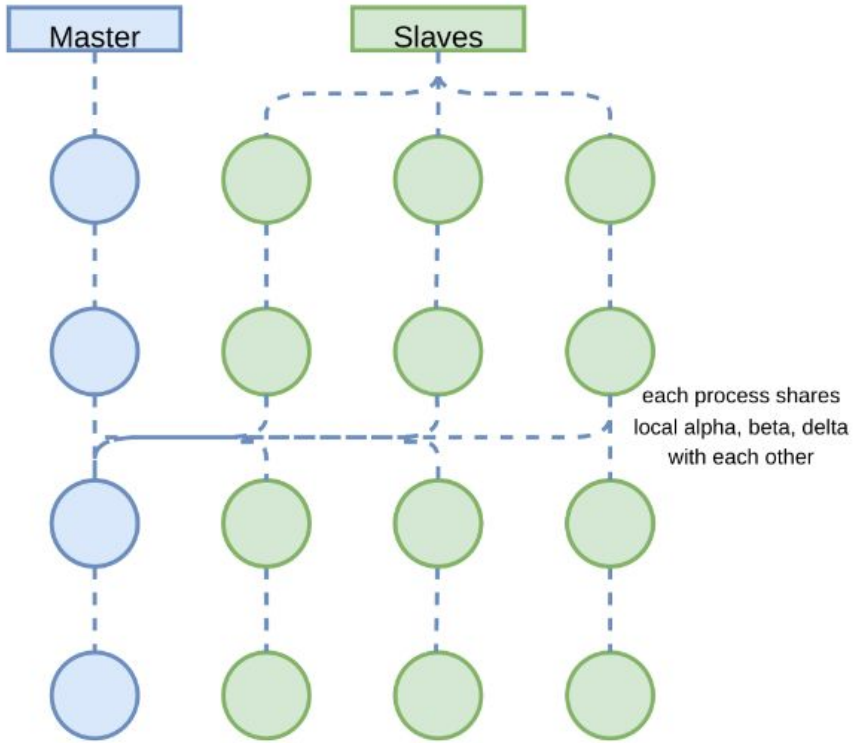
- Process 0 collects best solutions and distributes information.
- Uses MPI Gather and Broadcast for communication.
- Bottleneck on process 0



Parallelization Strategy (2)

Fully Decentralized Model:

- All processes share and determine global best solutions independently.
- Uses MPI Allgather for improved performance.



Experimental Setup

Hardware:

- University of Trento HPC Cluster: 126 nodes, 6092 CPU cores, 53 TB RAM, 10Gb/s network.

Benchmark Problems:

- Sphere function (unimodal, measures convergence speed).
- Rastrigin function (multimodal, tests ability to escape local minima).

Other settings

- Hyperparameter tuning
- Warm-up run
- 10 runs executed per experiment for robistness

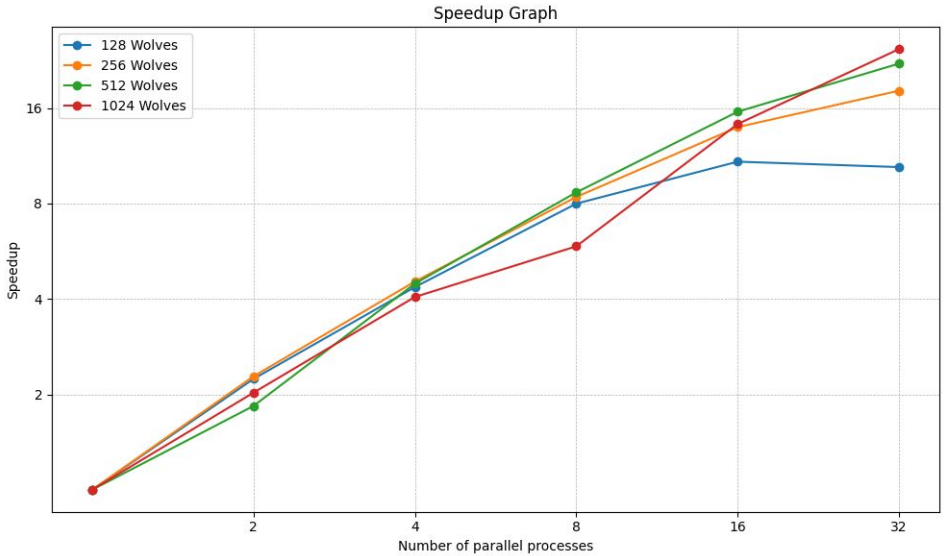
Results & Performance Analysis

Speedup Analysis:

- Near-linear speedup up to 16 processes
- Communication overhead limits gains beyond 16-32 processes.

Efficiency Trends:

- High efficiency with large workloads
- Significant drop for small problem sizes



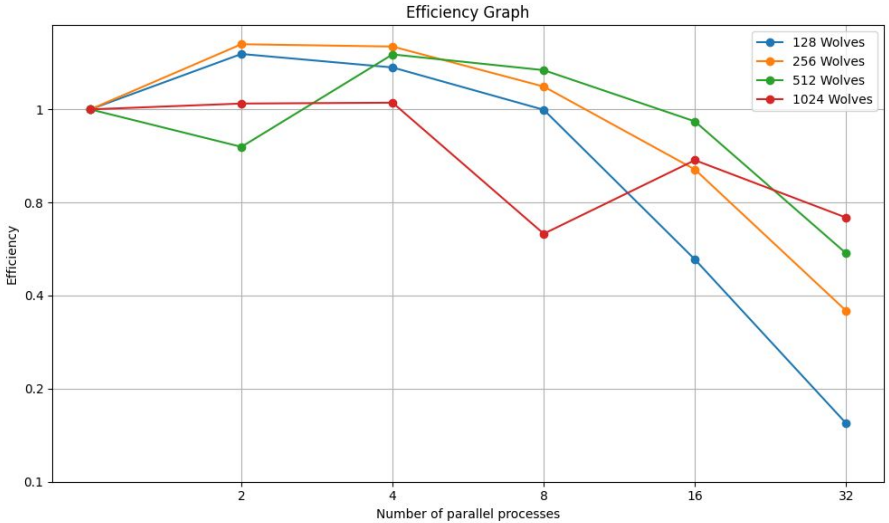
Scalability Analysis

Strong Scalability:

- Measures how well performance improves when adding more processors while keeping the problem size fixed.
- Performance gains diminish beyond a certain number of processors due to communication overhead.

Weak Scalability:

- Evaluates how performance holds when both problem size and number of processors increase proportionally.
- Parallel GWO demonstrates better weak scalability in large-scale problems.



Data dependencies

What are Data Dependencies?

- Occur when one computation relies on the results of another.

Types of Dependencies in GWO:

- Flow Dependency: Sequential updates of alpha, beta, and delta wolves.
- Anti-Dependency: Variables being read before they are updated.
- Output Dependency: Multiple processes updating the same variable.

Impact on Parallelization:

- Can cause synchronization issues and slow down execution.
- Need for careful ordering and optimized communication strategies

Conclusion & Future Work

Key Takeaways:

- Parallel GWO significantly reduces execution time.
- Works best for large-scale optimization problems.

Future Improvements:

- Hybrid OpenMP-MPI for better load balancing.
- GPU acceleration for further speedup