Travelling Salesman Problem Using Whale Optimization Algorithm

Prepared by:
Aditya Bisht(20114013)
Deepanshu Meena(20114031)
Sonali Gupta (20114094)

Problem Statement

- Travelling Salesman Problem: Given a set of cities and the distances between each pair of cities, the problem asks for the determination of the shortest possible route that starts at one city, visits each city exactly once, and returns to the origin city.
- Find a solution using the Whale Optimization Algorithm (WOA) and apply parallelization to enhance the efficiency of the algorithm

Datasets

- We have used two dataset. For our problem statement, we are utilizing two datasets:
    - Dataset 1: The first dataset is available at the following link: att48_d.txt. It contains data of 26 cities, distance matrix of 26*26 which represent distance between each pair of cities.
    - Dataset 2: The second dataset can be accessed from the following link: gr17_d.txt. It contains data of 17 cities, distance matrix of 17*17 which represent distance between each pair of cities.

TSP using brute-force

- This problem is a NP-hard problem. A simple brute-force solution can be outlined as follows:
  - Consider city 1 as the starting and ending point.
  - Generate all (n-1)! permutations of cities.
  - Calculate the cost of every permutation and keep track of the minimum cost permutation.
  - Return the permutation with minimum cost.

Whale optimization algorithm on TSP

- Steps to solve the problem using WOA:-
  - Generate random solutions for the TSP.
  - Set variables to track `bestPath` and `bestDistance`.
  - Evaluate each solution and update `bestPath` if necessary.
  - Apply WOA logic to potentially improve solutions.
  - Evaluate fitness of each solution.
  - Update solutions by combining strategies:
    - moving closer to the best solution and performing local optimizations.
  - Repeat process for specified iterations.
  - Return `bestPath` and `bestDistance`.

Fitness Function

- The fitness function for the Traveling Salesman Problem (TSP) is defined as the total distance travelled by salesman.
- It is calculated as sum of distances between every consecutive cities visited.
- The goal is to minimize this total distance.
- $f(x) = \sum_{n=1}^{n-1}\left(d(x_i, x_{i\ +\ 1})\right) + \ d(x_n, x_1)$

n= no. of cities

x_i= ith city in sol. Order

$d(x_i, xj)$ = distance b/w
city i to city j

Updating the solution

- Encircling Prey Strategy:
  - Probability of 0.5 to choose this strategy.
  - For each city:
    - 10% chance to swap it with the corresponding city in the `bestSolution`.
- Bubble-net Attacking Strategy:
  - Probability of 0.5 to choose this strategy.
  - Randomly select two cities and swap them.
  - If resulting solution is worse than the best solution:
    - Revert the swap to restore the original solution.

Parallelized Whale Optimization Algo. on TSP

- Parallel For Directive:
  - Parallelise loop iterations of calculating fitness function.
  - Each thread evaluates fitness of solution subset independently.
  - Ensures efficient parallel evaluation of solutions.
- Critical Section:
  - In this section, we update bestDistance and bestPath.
  - Updates occur if solution's distance is better than current best
- We have used upto 64 threads.

Results

Dataset 2	**Brute-force**	**Using WOA**	**Parallelised WOA with 2 thread**	**Parallelised WOA with 4 thread**	**Parallelised WOA with 64 thread**
Total distance	291	291	291	291	291
Time	too long	10.226	6.291	6.254	5.437

Results

Dataset 2	Population 1000	Population 900	Population 500
Total distance	291	307	321

Thank You