Team Members: Fernando Camurugy Notari (fc2672), Maximo Jalife (mj2994)
Project Proposal: Parallel Genetic Algorithm Solver for the Traveling Salesman Problem

1. Introduction

- Background: The Traveling Salesman Problem (TSP) is a classic problem that is NP-Hard. The problem involves finding the shortest possible route that visits a set of cities and returns to the origin city.
  The Traveling Salesman Problem also has a decision version, which is NP-Complete. This is the case because while it is still difficult to find a solution to this version, once one is found it is easily verifiable. The problem statement is: "*given a set of N cities and a unit measure of distance, is it possible to find a path that visits all N cities and returns to the starting point in less than X distance units?*"
- Objective: The objective of this project is to develop an efficient solver for the NP-Complete version of the TSP using genetic algorithms, and also an efficient estimator of its NP-Hard counterpart. Both versions will be implemented in Haskell and make use of the language's parallelization capabilities. Both objectives will be addressed concurrently in the process of estimating the NP-Hard solution.

2. Algorithm

- Genetic Algorithm Overview: Genetic algorithms are a class of evolutionary algorithms that mimic the process of natural selection. They operate through a cycle of selection, crossover, mutation, and a termination criterion. These algorithms are particularly suitable for optimization problems like TSP due to their ability to explore a vast search space, which is what we need.
- Adapting Genetic Algorithms for TSP: For the TSP, each individual in the population represents a possible solution (i.e., a specific route visiting all cities). The fitness function will evaluate the total distance of the route, with shorter routes being more favorable. Selection will be based on fitness, favoring shorter routes for reproduction. Crossover and mutation operations will be designed to generate valid routes.
- Improvements and Optimizations: The algorithm will incorporate elitism, maintaining a portion of the best solutions in each generation. The purpose of this is to prune excessive paths and prevent an exponential blowup in the population. An adaptive mutation rate will be used, where the rate changes based on the diversity of the population, encouraging exploration of new solutions. This will prevent the algorithm from getting stuck in low-performing solutions which represent local minimums of the overall distance as opposed to the global minimum, which is the desired output. This technique is not fail-proof but should prove effective nonetheless.

3. Parallelism:

Parallelizing a genetic algorithm for the Traveling Salesman Problem (TSP): Genetic algorithms have inherent parallelism in some of their operations, such as the evaluation of individuals in the population and the application of crossover and mutation to different individuals. This is our planned approach:

- Evaluation: The fitness evaluation step, where we calculate the total distance for each tour, can be parallelized. We can evaluate the fitness of different individuals concurrently.
- Crossover: The crossover operation can be parallelized by applying crossover to multiple pairs of parents simultaneously. Each pair of parents produces a pair of offspring.
- Mutation: Apply mutation to different individuals in parallel.

4. Performance Evaluation

- Benchmarking Strategy: The solver will be benchmarked against standard TSP instances from the TSPLIB library.
- Performance Metrics: Metrics will include the quality of the solution (total distance), computation time.
- Comparative Analysis: Results will be analyzed to demonstrate the results of using a parallel genetic algorithm approach in Haskell.