Applying Genetic Algorithms to the Traveling Salesperson Problem : Project Proposal

Abby Butel-Fry

Introduction

One of the most famous problems in computational complexity theory is the traveling salesperson problem (TSP). The TSP is an example of a routing problem, which is a very common type of real-world optimization problem ("Vehicle Routing", 2025). The goal of the traveling salesperson problem is, given a list of locations for a salesperson to visit, construct the shortest possible route for them which visits every location and which starts and ends at the same point. This means, except for the start/endpoint, the salesperson should visit every location on the list exactly once.

Finding the guaranteed optimal solution for the traveling salesperson problem (especially for large instances), can be extremely computationally expensive. However, if we only need to find a reasonably good (but probably not optimal) solution within a relatively short length of time, we might opt to use a discrete search method called a genetic algorithm. Genetic algorithms are a heuristic method that is inspired by natural selection. Candidate solutions are conceptualized as organisms, and the fittest organisms (i.e., the highest quality solutions) will be the most likely to survive and pass on their advantageous characteristics to subsequent generations.

These interesting algorithms "can be applied to any problem that has these two characteristics: (i) a solution can be expressed as a string, and (ii) a value representing the worth of the string can be calculated" (Chennick, 2020). The first step of applying a genetic algorithm to the TSP is generating an initial population, i.e., a set of random lists which include every destination (other than the start/endpoint) exactly once. Next, some subset of these organisms would be selected to breed with one another. When two organisms from the mating pool breed with one another, they will generate a child organism which contains information from both parents, and which is a feasible solution to the given instance of the TSP. This stage within the genetic algorithm is called the crossover operation, and one approach that is suited to problems like the TSP is the partially matched crossover operation (Chennick, 2020). Next, some proportion of the newly generated organisms will undergo a random mutation (i.e., a change to their genome). Finally, we may optionally get rid of some of the less fit parents and/or offspring, then the breeding and subsequent operations are repeated for the desired number of generations.

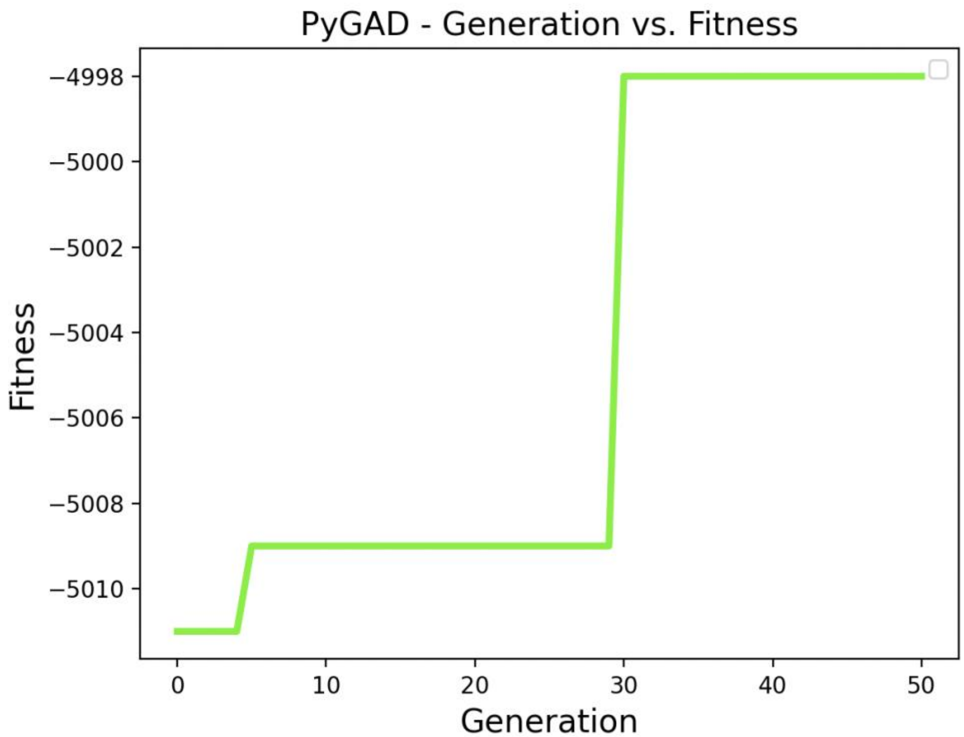
Proposed Work

We will use the USA13509 dataset found on TSPLIB, and we will randomly select subsets of coordinates of different sizes (e.g., 10, 100, 1000, 10000, and 13509). Next, we will use Google's OR-Tools solver to generate a reasonably good guess for each of these instances of the traveling salesperson problem. The solutions generated by this solver can be compared with the results of the genetic algorithms to assess the quality of the results. Additionally, we could opt to include these solutions in the initial populations in an effort to reduce the number of generations needed to generate the fittest possible offspring.

Our overall goal will be to examine how the value of the shortest tour seen so far changes with each generation within a genetic algorithm. In addition to varying the number of distinct nodes in each TSP instance, we will examine the effects of varying the initial population sizes, mutation rates, and the number of solutions carried over from one generation to the next.

Sketches of Graphs

Map depicting an example solution for a TSP with ten random nodes selected from USA 13509 dataset



Fitness = -1 x distance in miles

Relevant Journals & Trade Publications for Possible Publication

IAENG International Journal of Applied Mathematics

IEEE Transactions on Evolutionary Computation

International Journal of Computer Science Issues

International Journal of Computational and Experimental Science and Engineering

Resources/References

Chenneck, J. W. (2020, December 7). Chapter 14: Heuristics for Discrete Search: Genetic Algorithms and Simulated Annealing. In *Practical Optimization: a Gentle Introduction*. essay. Retrieved from https://drive.google.com/file/d/1hHiNpSFt2Kb5pjbMRXSdEpxXXTDEYIvR/view.

Google. (2025, January 22). *Vehicle Routing*. Vehicle Routing | OR-Tools | Google for Developers. https://developers.google.com/optimization/routing