Algorithms for the Travelling Salesman Problem

Partial Report

Eric Gastineau
College of Computing
Georgia Institute of Technology
Atlanta, GA, US
egastineau3@gatech.edu

Jacob Beel
College of Computing
Georgia Institute of Technology
Atlanta, GA, US
jacobdbeel@outlook.com

Yiwen Bu
College of Computing
Georgia Institute of Technology
Atlanta, GA, US
yjquek@gatech.edu

Yong Jian Quek
College of Computing
Georgia Institute of Technology
Atlanta, GA, US
ybu@gatech.edu

INTRODUCTION

The Travelling Salesman Problem (TSP) is a NP-complete problem that has no known optimal algorithm that runs in polynomial time. This report will examine various methods to cope with this, some offering an approximate solution, while others providing a heuristic for a local search. The four algorithms that we will examine will be

1. Exact algorithm using Branch-and-Bound
2. Construction Heuristics using Nearest Neighbor
3. Local Search using Simulated Annealing
4. Local Search using Genetic Algorithm

1 Problem Definition

Given a list of coordinates, find the shortest path that goes through each city exactly once and then returns to the origin.

2 Branch-and-Bound

For Branch and Bound, the goal is the find a perfect result. It takes much more times as the time complexity for our problem is exponential. However, we can get some results for small datasets. The lower bound that has been chosen for now is the sum of the partial path + the sum of the minimum cost of leaving every remaining city.

We also used a heuristic to choose the next city. We sort the cities according to their distances to the last chosen city and we choose the closest. With this heuristic we can find the exact solution for Cincinnati in 0.84 second and for Kansas State University in 0.2 second. However, we find the exact solution for Atlanta in a much greater time (few hours) and we can't find the exact solution for other datasets. This is because the time complexity is exponential and that each time, we add a new point we need to do much more operations.

3 Nearest Neighbor

The algorithm is very simple, adding the nearest neighbor to the tour at any given step. However, we can do some experiments with starting points, since we're given the freedom to start at any point. I propose we look at random, first in a sorted list of nodes, and the node with the highest degree.

3 Simulated Annealing

Simulated annealing (SA) algorithm is another local search strategy we are going to use for this project. the simulated annealing algorithm is not guaranteed to get the optimal solution. However, it is useful to speed up convergence at the beginning of the process with a high T, and when it cools down, the probability of optimal solution increases. As it closes to 0, it has the highest probability to get the optimal solutions [1]. In this case, when T is large, it is easy to accept neighboring solutions even it is worse than current solution, but it will be strict when T is getting small. For SA algorithm, it is important to know how to update temperature and what should be stop criterion. Sometimes, it is also needed to restart the algorithm.

4 Genetic Algorithm

Based on the theory of evolution, Genetic Algorithm (GA) uses the core components of the theory to determine the neighborhood and the heuristics for a local search [2].

1. Selection of the fittest
2. Crossover
3. Mutation

GA is initialized with a base population of randomly selected DNA. It then evaluates the fitness of each DNA and has a probability of selecting a DNA in proportion to its fitness. Two selected DNA act as the parents and will be their genes will be passed onto a child DNA in crossover process. The child DNA will have a chance to mutation, before being introduced into the next generation of the population.

For TSP, the genes are the individual sets of coordinates, the DNA is a route, and the fitness is evaluated by the reciprocal of the length of the tour. Hyperparameters being optimized are the base population, the probability of mutation, the number of generations, and the crossover method.

5 Progress

All utility functions to handle input and output files have been completed and progress has been made, in parallel, on all algorithms. The next step would be to complete the testing and debugging, before moving on to collecting the qualitative data.

REFERENCES

[1] Jon Kleinberg and Éva Tardos. 2014. Algorithm design, Boston: Pearson.

[2] Melanie Mitchell. 1998. An Introduction to Genetic Algorithms, Cambridge, MA: MIT.