Using Genetic Algorithms to Optimize the Traveling Salesman Problem

By: Ryan Honig

Abstract

My goal is to create a program that can solve the Traveling Salesman Problem, finding near-optimal solutions for any set of points. I will use genetic algorithms to try to find the optimal paths between the points. I would also like to expand my algorithm so that it can solve both symmetric and asymmetric problems. In the end, after I create a working algorithm that will find near optimal paths, I hope to create a graphic interface that will display the chosen points and the paths through those points as the algorithm runs.

What is the Traveling Salesman Problem

Traveling Salesman Problem (TSP) - a set of points is given. Try to find the shortest path that travels between each point once and returns to the starting point

Symmetric TSP - distance between towns A and B is the same as distance between towns B and A.

Asymmetric TSP - distance between towns A and B is different from distance between towns B and A.

Background

- Purely genetic approaches can find near optimal solutions, but take a long time
- Purely heuristic approaches can run very efficiently, but don't find very optimal solutions
- Many of the current best known solution algorithms use a combination of heuristics and genetic algorithms

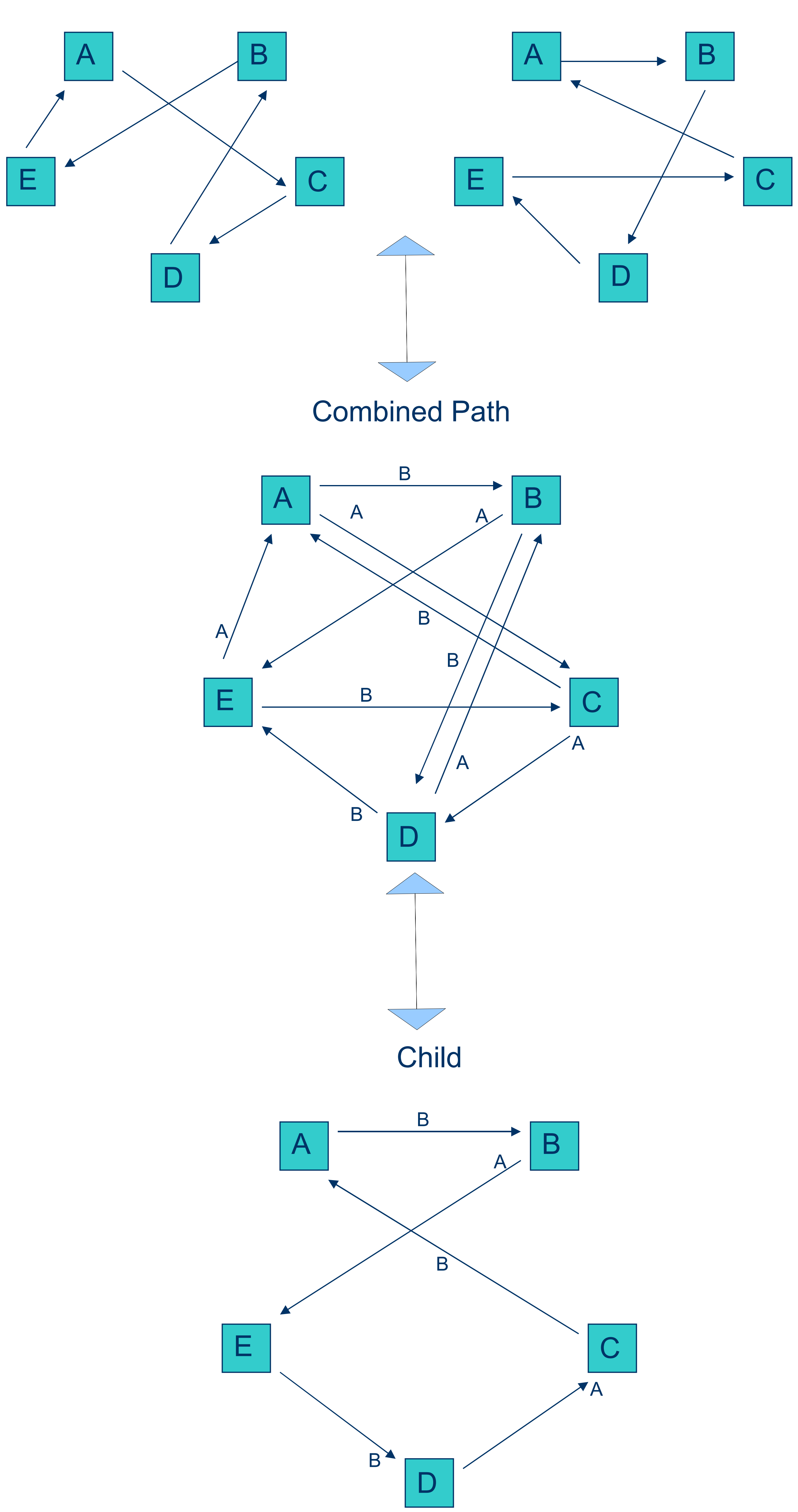
Development

- Currently working on an algorithm that randomizes a pool, and then uses genetic crossovers within the pool to find the best solution
- Later, I will use a heuristic to generate the initial pool
- I will also add in mutations so that the pool doesn't get stopped on a single path

Results

- There is currently a bug in the section of the code that fixes broken paths, creating duplicates of sections of the path.

How My Algorithm Works


Parent A
A
B
E
C
D
Parent B
A
B
E
C
D
Combined Path
A
B
B
A
A
A
B
B
E
B
C
A
A
B
D
Child
A
B
B
A
B
E
C
A
B
D