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

- I have a genetic algorithm that creates a pool, and then uses genetic crossovers within the pool to find the best solution
- I also have a mutation function that has a one in fifty chance of adding a variation into the pool by reversing a segment of a path, this helps to keep the pool from getting filled by copies of the same path
- I also created a heuristic that creates a better pool than the randomized pool

Results

Random pool

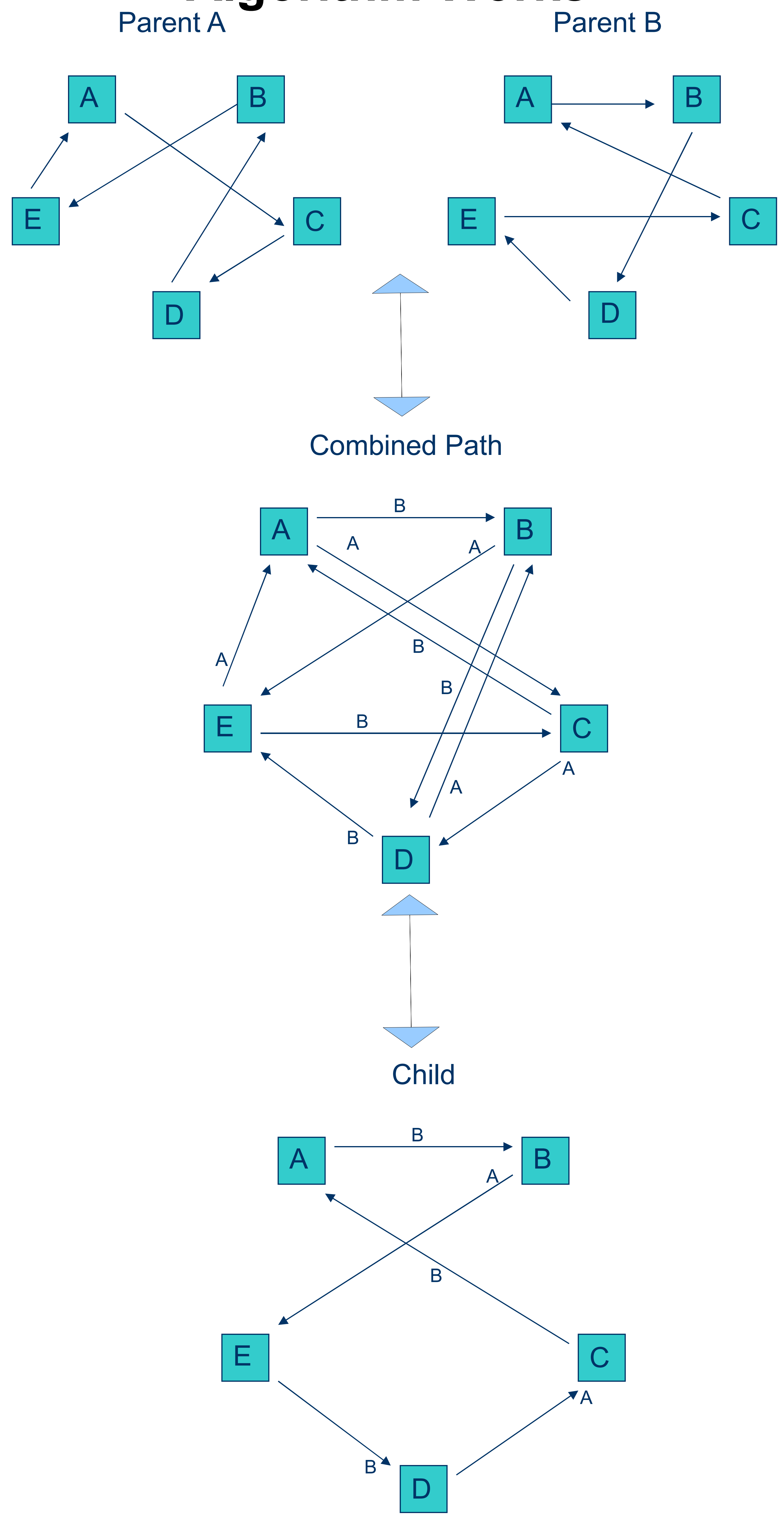
- data set a280: 2608.837612, which has an error of just 1.16 percent from best known of 2579, average running time 2.15 seconds.
- att48 data set: 10820.248365, which has an error of just 1.81 percent from best known of 10628, average running time of 3.52 seconds.

Heuristic pool

- data set a280, 2597.401845, which has an error of just .72 percent from best known of 2579, average running time of 5.03 seconds.
- att48 data set, 10751.542837, which has an error of just 1.16 percent from best known of 10628, average running time of 7.31 seconds.

- So although the heuristic finds slightly better solutions, it takes much more time to run, so I am not sure whether to continue with it.

How My Genetic Algorithm Works


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