

Name: Samuel Cook

User-ID: gsgw38

Algorithm A: Genetic Algorithm

Algorithm B: Particle Swarm Optimisation

Description of enhancement of Algorithm A:

The most significant change that I made to my algorithm was the use of a function called newTourNN that generates Nearest Neighbour tours to use in the initial population. I also experimented with using Minimum Spanning Trees in the initial population but found this was ineffective.

My enhanced algorithm uses a different mutation function (mutateChild) which reverses a section of the tour instead of making a random swap.

I changed the way my tours were stored in order to keep a record of the length of the tour and the ordering of the cities, this meant that I could rewrite the function for choosing parents and introduce elitism. Elitism is introduced by using the 'fittest' tours from the old population as a starting point for the new population. The enhanced version also has a new function for choosing the parents which does not require calculating the tour length for each member of the population.

I also increase the probability of mutation each time a new shortest tour is found, this tactic increases the level of diversity in the population to avoid local minima.

I settled on my final parameters (a population size of 100 and an initial probability of mutation equal to 0.1) after a large amount of experimentation.

Description of enhancement of Algorithm B:

For my enhanced version of the Particle Swarm Optimisation (PSO) I used the nearest neighbour tours as initial positions for the particles. To avoid my particles starting at positions which are too similar I use a percentage (80%) of nearest neighbour tours and the rest are randomly generated.

The enhanced version uses a different discretization. This discretization uses a new function for finding the velocity between the positions of two particles and does not require tours to be in canonical form. Instead of using bubble sort it iterates through the first tour and records the swaps required to reorder it according to the linear order defined by the second tour. This method of 'sorting' takes advantage of the knowledge that a tour contains one example of every city and is much faster, allowing for more iterations in a set time.

My enhanced version also has a new function for editing velocities according to the epsilon value. A random swap is added instead of being removed as in the basic version. This keeps the particles moving to avoid local minima.

I also experimented with a new way of calculating the velocity but found that this was less effective than the basic one.