

Name: Kamil Hepak

Algorithm A: A\* Search

Algorithm B: Genetic Algorithm

Description of enhancement of Algorithm A:

I have implemented an A\* Search algorithm with the heuristic of the remaining path cost to a goal node found using the basic greedy algorithm - that is, the path cost of a greedy search from the last city in any state's tour to a valid goal state. To enhance this basic algorithm, I implemented a 2-opt local search algorithm to improve the tour my A\* search finds. My 2-opt works by reversing every possible subsequence in the tour to simulate "uncrossing" overlapping edges in the hope that their "straightening-out" yields a shorter overall tour - every time such a sequence reversal does in fact yield a shorter tour, the algorithm continues running with this new, shorter tour as its input. This process keeps going until no more improvements can be found.

2-opt improves every tour my basic algorithm finds by between 2% and 16% using increases in time that are small in absolute terms (eg 100ths of a second) but can be large in relative terms (eg 200%). Overall, it's well worth it.

Description of enhancement of Algorithm B:

I have implemented a Genetic Algorithm as described in lecture, with a one-point crossover (modified to fix repeated cities) and a mutation that swaps two cities in the tour. To enhance this basic algorithm, I changed my mutation to reversing a random subsequence of a tour, generated two of my initial chromosomes using a greedy algorithm, introduced elitism by keeping my best two chromosomes between each generation, and changed my crossover to an adapted two-point crossover.

These enhancements together improve my tours considerably - 535 cities shaved about half of its tour length between basic and enhanced, though the improvement is less pronounced in other city sets.