Name: Owen Shaw User-ID: rmvc61

Algorithm A: Basic Greedy

Algorithm B: Rank-Based Ant Colony Optimization

Description of enhancement of Algorithm A:

Algorithm A is a basic greedy algorithm and was enhanced by having a salesman start at every possible city, instead of just the first one. This was done by creating a list of salesman objects and then calculating their tour and then returning the shortest tour done by the salesmen.

This improved the results of the basic algorithm on most city files.

Description of enhancement of Algorithm B:

The enhancement of algorithm B expands upon the rank-based ant colony optimization by introducing diversity control. Diversity control works to prevent the ants from getting stuck in local optima [1]. It does this by selecting an ant at random after every iteration and either making it undergo simulated annealing or mutation, dependent upon the diversity score of the ants. The diversity score is calculated using the equation below. Where $\bar{d}$ *is the average difference between the tour length of the best ant and the remaining ants,* d_{min} *and* d_{max} *are the distances of the tour length of the worst ant and second best ant, respectively.*

$$Diversity = \frac{\bar{d} - d_{min}}{d_{ma} - d_{min}}$$

If the diversity is larger than 0.5 then the chosen ant will undergo simulated annealing, which should bring the diversity down, or if the diversity is lower than 0.5 the chosen ant will undergo mutation at a specified mutation rate.

This improved or equalled the results of the algorithm on most city files, however the extra steps involved were computationally expensive and for large city files, each iteration took longer than the basic algorithm.

References:

[1] Abdulqader M. Mohsen, "Annealing Ant Colony Optimization with Mutation Operator for Solving TSP", Computational Intelligence and Neuroscience, vol. 2016, Article ID 8932896, 13 pages, 2016. https://doi.org/10.1155/2016/8932896