Name: Robbie Goodall User-ID: cnkz75

Algorithm A: Genetic Algorithm (GA)

Algorithm B: Ant Colony Optimisation (AC)

Description of enhancement of Algorithm A:

The first enhancement I made was to change the definition of crossover when creating children. I used the idea of Order Crossover from Genetic Algorithms for the Travelling Salesman Problem: A Review of Representations and Operators, Artificial Intelligence Review 13 (1999) pages 140-141. It works on the premise that the order of cities and not their positions in the tour is more important. To produce a child from parents X and Y, it selects a subtour of length 5 inside the tour of X, and fixes its position. Then, starting from the index after the subtour ends, it adds cities not in the subtour in the order in which they appear in the tour of Y (starting from the same index). Note the algorithm 'wraps around' the tour of Y and the new tour for the child, meaning once the end of the tour is reached, it goes to index 0. The process is repeated for fixing a subtour from Y and then adding in cities from the tour of X.
The second enhancement I made was changing the definition of mutation. I used the notion of Insertion Mutation, from the same paper mentioned above on pages 149-150. The algorithm selects a city at random from the tour, removes it, and then re-inserts it into a random point into the tour. Note that the re-insertion point is selected such that the mutated tour is different from the original tour (i.e., the city is not re-inserted into the same position it originally was in).
In general, these enhancements were definitely an improvement and in all cases found a tour that was significantly shorter than or the same length as those found in the basic implementation. The only case where the tour found was the same length as in the basic implementation was for the 12 city tour.

Description of enhancement of Algorithm B:

I did some research online and found an adaptation of the basic Ant System algorithm called the Max-Min Ant System. The key differences between the Max-Min Ant System and the basic Ant System is that pheromone levels on each edge are bounded between a minimum and maximum value, and at the end of each iteration, only a single ant is allowed to update pheromone levels on its trail. This ant is either the best ant from the iteration (the one with the shortest tour), or the best ant overall from any previous iteration, depending on how many iterations have been done. In the first 25 iterations, the iteration best ant is used. After that, the global best ant is used every 5 iterations between iterations 26-75, every 3 between iterations 76-125, every 2 between iterations 126-250, and after that point, it is used every iteration. I also played around with some of the parameters and used $\beta = 2$, $\rho = 0.2$, as opposed to 3 and 0.5 respectively in the basic implementation.
The bounding values for the pheromone levels (τ_{min} and τ_{max}) are chosen with respect to certain parameter values. τ_{max} is also dependent on the length of the best tour, and is updated every time a new best tour is found. The pheromone levels for each edge are initialized to τ_{max}. I found this paper https://www.researchgate.net/publication/277284831_MAX-MIN_ant_system to be extremely useful and it is where I got the idea and algorithm from.
The best tour found from this enhanced version was always at least as good as the basic implementation, and in most cases was better (i.e. shorter).