Name: Tala Eagling-Vose User-ID: djgv37

Algorithm A: GA

Algorithm B: AC

Description of enhancement of Algorithm A:

To find the optimum values of population size and mutation chance across the test sets I performed a grid search of different combinations of values for each. For population size I tested values between 50 and 2000 and for mutation probabilities between 1% and 30%. I then plotted graphs of Both average and minimum tours for the given parameters, this led me to use 150 and 0.08.

*Additionally, taking inspiration from **An Improved Immune Genetic Algorithm and its Application on TSP** (A. S. Ghorab) I chose to implement vaccination algorithm on weak members of the population. The vaccination is found by identifying the shortest edge in the graph and then greedily appending to the front or back of the tour depending on what is shorter, I trilled several lengths for this vaccine but found 4 for tours with more than 10 cities* else 2 *worked effectively. The chance of vaccination is inversely proportional to its fitness and if selected the vaccine will be inserted into tour at the first instance of a member of the vaccine (e.g., tour- [0,5,2,3,4,6,1], vaccine- [1,2,3], vaccinated- [0,5,1,2,3,4,6]). The notion prevention of weakness has been used meaning if the fitness is reduced with the vaccine the original tour is retained. Immune selection is also used meaning given a successful vaccination (one where fitness is increased), with a small probability proportional to the fitness of the vaccinated tour the vaccination will be ignored.*

My further experimentation with both probabilities for vaccination and immunity found successful values of 0.5(bestFitness-childFitness)/bestFitness and 0.1*(bestFitness-vaccinatedFitness)/bestFitness. This, when averaged across all test sets (not including 012) lead to an improvement in best tour of 3.46%.*

Description of enhancement of Algorithm B:

To explore and enhance my Ant colony optimisation I performed grid search on different parameters including alpha, beta, p and the number of ants before comparing both shortest and average tours for each combination. From this I found the best were 0.95, 4.6, 0.45 and 200 respectively.

Additionally, taking inspiration from **Annealing Ant Colony Optimization with Mutation Operator for Solving TSP** (Abdulqader M. Mohsen) I created a genetic and ant colony hybrid algorithm. This makes use of diversity calculation outlined in the paper to measure concentration of ants around the best tour.

When this diversity is less than 0.5 there is a probability proportional to (0.5-divercity) that the ant with shortest tour in that iteration will crossover with a random ant and then replace that ant. When greater than 0.5 there is a probability proportional to (divercity-0.5) that an ant in the population will be changed to a random tour.

When averaged across all test sets (not including 012) lead to an improvement in best tour of 1.03%.

DESCRIPTION OF ALGORITHM ONLY IF THE ALGORITHM IS NOT COVERED IN LECTURES

Description of *non-standard* Algorithm A:

*Describe any non-standard algorithms you have implemented that **have not been covered in lectures** (otherwise these boxes should be blank) You need to convince me that your implementation is indeed that of the named algorithm and you need to **provide a full reference to the source for your algorithm**. You should **include a pseudocode description**. You can vary the sizes of these boxes but not the font (Calabri), font size (11) or paragraph properties (single space), and everything should fit onto one side of A4. (You can delete these instructions.)*

Description of *non-standard* Algorithm B:

Type here.