ECM3412 Nature-Inspired Computation Coursework Report

Question 1: Which combination of parameters produces the best results?

To answer this question, I initially set the parameters of my ant colony algorithm to a generic combination: alpha = 1, beta = 1, number of ants = 10, evaporation rate = 0.02, Q = 1 and the file to 'Burma.xml' (the reduced number of cities allowed for more executions of the algorithm to be recorded and analysed). I now varied each parameter one at a time over a range of values to compare its performance on the algorithm. As the best routes found by the ants often varied, I executed the algorithm 10 times to get a spread of results for each parameter value. I have chosen a box plot as my method of analysing the produced data because, while the best route found in all the runs is a very important aspect, the spread of the of the outputs clearly shown by a box plot illustrates the exploitation vs exploration element of the algorithm.

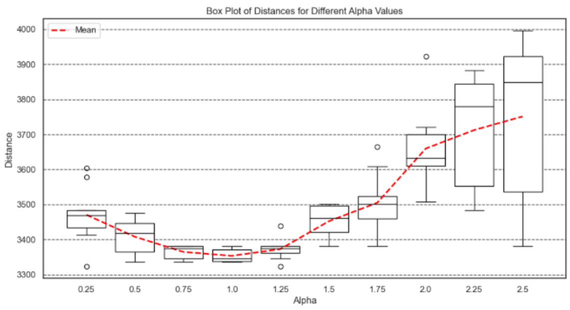

Figure 1: Box Plot for Different Alpha Values

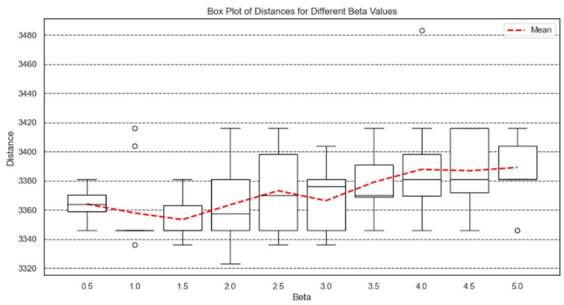

Figure 2: Box Plot for Different Beta Values

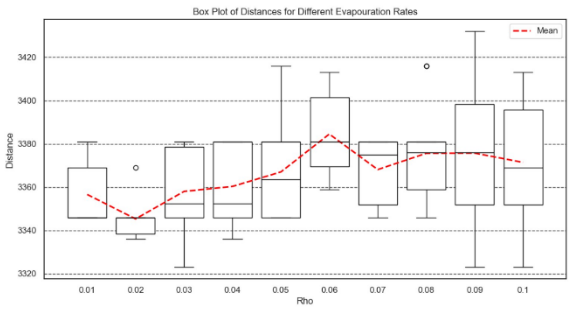

Figure 3: Box Plot for Different Evaporation Rates

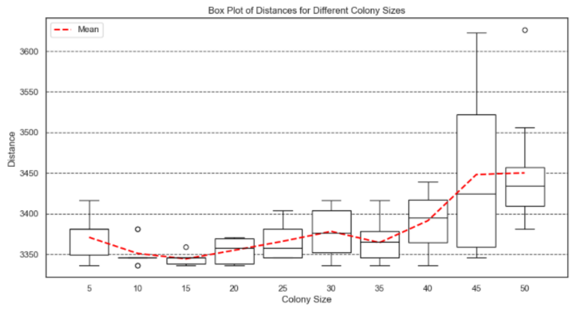

Figure 4: Box Plot for Different Colony Sizes

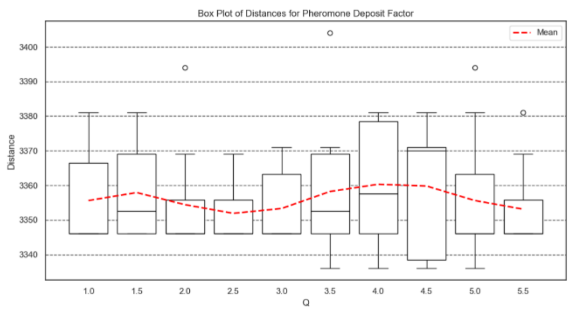

Figure 5: Box Plot for Different Pheromone Deposit Factors

These box plots and mean show the best parameters for the algorithm. Alpha and Beta should both be kept close to 1, the evaporation rate should be set to 0.02. A colony size of 15 had the most consistent and best solutions. The pheromone deposit factor has a smaller effect on the algorithm's performance but setting its value to between 2 and 2.5 reduces the spread of the results, making the outputs far more consistent.

From here I tested the use of Elitism and MMAS within the ant colony. I used a population size of 30 for these tests to increase the impact of these variations but kept the other parameters the same as before. The MMAS variation has been implemented as described in [3], where only the best ant from each iteration deposits pheromones, all values in the pheromone matrix are kept within the set pheromone range and initialised at the upper boundary.

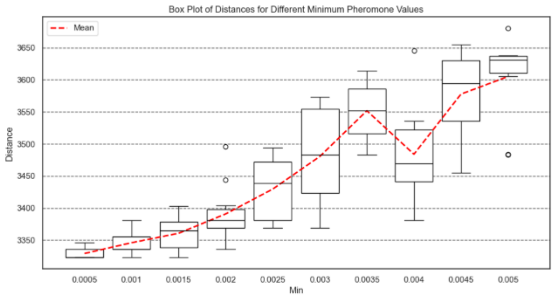

Figure 6: Box Plot of Different Minimum Pheromone Values

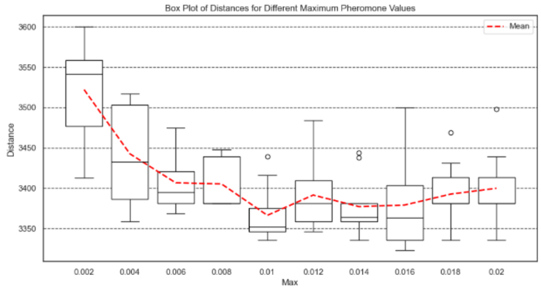

Figure 7: Box Plot of Different Maximum Pheromone Values

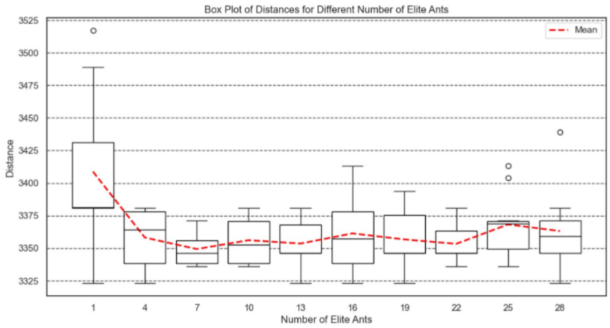

Figure 8: Box Plot of Different Number of Elite Ants

The number of ants selected each iteration doesn't appear to have a large effect on the average performace of the algorithm however when selecting between 7 and 10 ants the algorithm did produce more consistent results.

Increasing the minimum number of pheromones on each edge seems to only reduce the performance of the algorithm however Figure 7 clearly shows that setting the max pheromone value of an edge to 0.01 results in consistently good results. Unfortunately compared to the results for the other parameters, no significant improvement was shown.

Using the optimal parameters found, I ran the algorithm on both Burma and Brazil to attempt to find the optimal solution to the travelling salesman problem in these cities. The best routes found for each city were:

Burma: [1, 2, 14, 3, 4, 5, 6, 12, 7, 13, 8, 11, 9, 10] –> length = 3323

Brazil: [1, 30, 40, 25, 9, 13, 32, 20, 53, 50, 4, 22, 8, 55, 54, 2, 41, 35, 10, 52, 51, 47, 49, 43, 27, 12, 57, 23, 5, 48, 39, 3, 29, 17, 36, 26, 6, 19, 28, 14, 33, 45, 46, 56, 34, 15, 37, 21, 11, 16, 38, 42, 7, 31, 58, 24, 44, 18] -> length = 27581

Question 2 and Question 3 (merged): What do you think the reasons are for your findings in Question 1 and how does each parameter setting influence the performance of the algorithm?

Alpha – Used when calculating the probability of an ant visiting a city. Alpha affects how strongly the current pheromone matrix impacts the probabilities, increasing the value of Alpha will result in a greater pheromone impact so the ant colony will favour pre-discovered promising routes. However, this can lead to pre-mature convergence to local minima which Figure 1 demonstrates by a large variance in the results.
Beta – Again used when calculating the probability of an ant visiting a city. Beta affects how strongly the heuristic matrix impacts the probabilities, however, as the values in the heuristic matrix are all smaller than one, increasing Beta will lower the impact of the heuristic matrix. The local heuristic used for these results was 1/Length of Path, meaning that shorter edges are given a greater value.

Figure 2 shows that as Beta increases and the impact of the heuristic matrix weakens, the results found by the ants worsen as they are not favouring shorter paths enough.
Colony Size – The number of ants traversing the graph in parallel influences the exploration exploitation balance of the algorithm as increasing the colony size allows more paths to be explored before the pheromone matrix is updated and the colony gets closer to a convergence. Figure 4 demonstrates that while increasing the colony size is beneficial at first, when the colony starts becoming too large the algorithm no longer executes enough iterations for potential solutions to be adequately explored, resulting in a worse average performance.
Evaporation Rate – Determines how fast the pheromone values decrease during runtime which has a great effect on the exploration exploitation balance of the colony. An increased evaporation rate means previous paths found by the ants are quickly forgotten and only repeatedly travelled routes are retaken, this leads to a faster convergence to a local minimum. On the other hand, a lower evaporation rate leads to suboptimal paths explored in the early stages of the algorithm keeping a high pheromone value. This means that current iterations are less likely to explore more deeply the currently favoured paths. Figure 3 reflects these properties, after finding the optimum balance between exploration and exploitation an increase in the evaporation rate leads to more variance in the outcomes due to the faster convergence.
Pheromone Deposit Factor – This parameter is a constant number in the pheromone deposit formula that has a linear correlation to the pheromone deposit. Increasing this factor will intensify the exploitation element of the algorithm as ants will favour previously travelled paths, however increasing this constant too much will lead to pre-mature convergence. While the box plot in Figure 5 shows little effect on the average, the wider range of the inter-quartile range that appears as Q increases reflects the early convergence to early found routes.
Elitism – When Elitism is introduced to the algorithm only a specified number of the ants with the best fitness function during that iteration will be selected to deposit pheromones. A low number of selected ants leads to very little pheromones being added to the matrix so promising paths are not explored deeply enough. Figure 8 illustrates that a lower number of selected ants leads to more variance in the results and on average perform worse. When the optimal proportion of ants are selected to deposit pheromones, we see consistently good results because promising paths are being reinforced by the ant selection leading to the correct amount of exploitation.
MMAS – Max-Min Ant System combines strict elitism with a boundary to the values of the pheromone matrix [3]. Allowing only one ant per iteration to deposit pheromones leads to quick convergence which is counteracted by the upper bound on any one edge's pheromone value. Figure 6 shows how an increase in the lower bound for the pheromone values reduces the performance of the algorithm while also increasing the spread in the results. This is due to infrequently visited paths still maintaining a high pheromone value and as such a high chance of an ant choosing this path, reducing the number of ants that further explore the current promising paths.

Question 4: Can you think of a local heuristic function to add?

The current local heuristic used is 1 / length of edge. This makes immediate shorter routes more desirable to the ants, it would be interesting to see the results if the heuristic matrix used 1/ (length of edge)^2. This would greatly decrease the probability of the ants choosing a long edge. In addition, the opposite would also be useful to observe, using 1 / $\sqrt{\text{length of edge}}$ would reduce the undesirability of a longer path.

To explore this path further, I adapted the heuristic matrix so each value is now calculated by: 1 / (length of edge)^{Theta} where Theta is a predefined parameter. I tested varying the value of Theta using the optimal parameters found in Q1 (excluding Elitism and MMAS for simplicity). Figure 9 displays the results found when varying Theta, the general performance of the algorithm worsens as Theta exceeds 1.5.

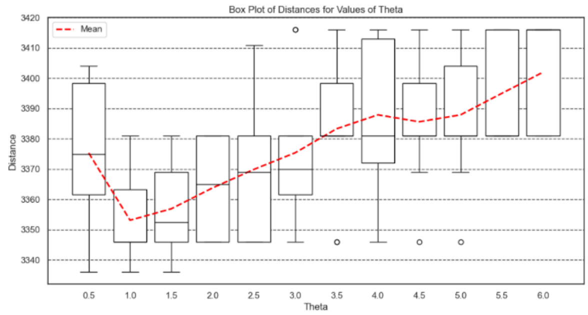

Figure 9: Box Plot for Different Values of Theta

Question 5: Can you think if any variation for this algorithm to improve your results?

To improve the results, the parameters looked at in Q1 could be dynamically changed during runtime (excluding the number of ants). Increasing the evaporation rate during the early stages of the algorithm will encourage exploration as previous iterations' pheromones will have a smaller impact on the current ants, avoiding premature convergence to unideal solutions. As the algorithm progresses the evaporation rate can then be lowered to allow the ants to exploit the discovered promising paths. Increasing the evaporation rate can also be used when the ants are repeatedly converging to the same path, this would disrupt the current pheromone paths allowing the ants to explore alternative paths outside the current local minima. The other parameters such as alpha and beta can be altered in a similar manner to shift the exploration-exploitation balance during the runtime.

Question 6: Do you think of any other nature inspired algorithms that might have provided better results?

Genetic algorithms (GA) can be used to solve the travelling salesman problem where a phenotype can represent the order of a path taken. The population of phenotypes will undergo a series of selection and mutation processes, mimicking the idea of real-life evolution, in order to increase the fitness of the population. Over multiple iteration, the population's fitness values will converge to a solution.

No. Cities	GA		ACO	
	Time	Best Solution	Time	Best Solution
13	1.0	77892	1.0	67950
25	3.0	147178	3.0	80321
35	8.0	127886	16.0	49998

Figure 10: Performance of GA vs ACO [2]

The papers [1][2] provide comparative analysis for the performance of GA and ACO. Figure 10 presents the consistent better results obtained from using ACO however it is worth noting that these solutions require more time to compute.

Overall, the situation in which the two algorithms are applied will determine which one is better suited. With the current problem, computational time is not a large issue as ACO still solves Brazil (58 cities) in a reasonable amount of time. If the problem contained a very large number of cities and ACO became ineffective to use, genetic algorithms would be of more use as a solution would still be acquired.

References

[1] Quoc Anh, L. (2019). Comparing the effectiveness of the genetic algorithm and ant colony optimization algorithms for the traveling salesman problem. *Vinh University Journal of Science*.

[2] Adib, M.Y., Razia, J., & Rahman, M.T. (2021). Experimental Comparison between Genetic Algorithm and Ant Colony Optimization on Traveling Salesman Problem. *International journal of scientific research in science, engineering and technology, 8*, 155-162.

[3] Dorigo, M., & Stützle, T. (2004). Ant colony optimization. *MIT Press.*