ECM3412 Ant Colony Optimisation Algorithm Report

Student Number: 710017661

November 2023

1 Graphs and Experiments Analysis

To truly optimise the ant colony optimisation (ACO [1]) algorithm being used, each of the parameters had to be systematically identified and tested in isolation. Doing this enables the conditions required for both converging on a value and finding the optimal value to be found and applied. Experiments were therefore carried out for the following parameters, *colony size*, *pheromone evaporation rate*, *pheromone importance factor* (alpha), *heuristic importance factor* (beta) and a *local heuristic* (Q) which was used to reward better reward solutions with a better fitness value. Furthermore, in each of the experiments base values were used for each of the parameters not being tested, these values were, a colony size of **10**, a pheromone evaporation rate of **0.5**, an alpha value of **1**, a beta value of **2** and a Q value of **1**. These values were chosen as they provide decent results across the board and were the starting values provided to us in this module [2].

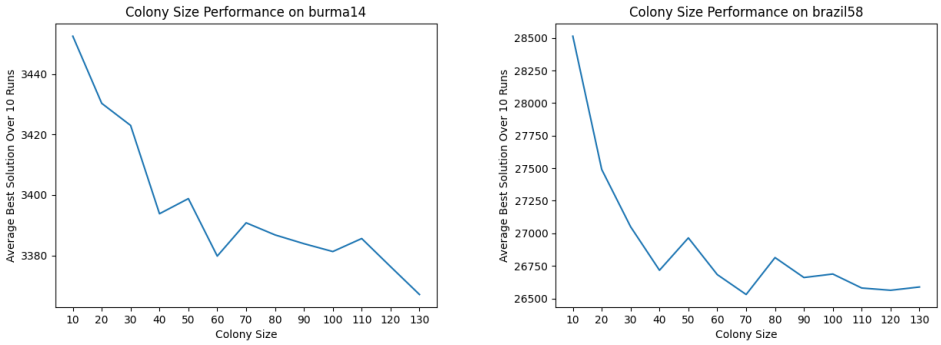

Figure 1: The Impact of Colony Size on the ACO Algorithm

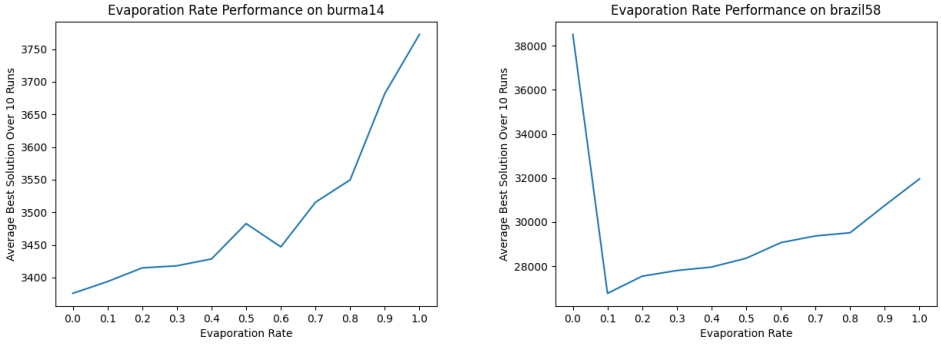

Figure 2: The Impact of Evaporation Rate on the ACO Algorithm

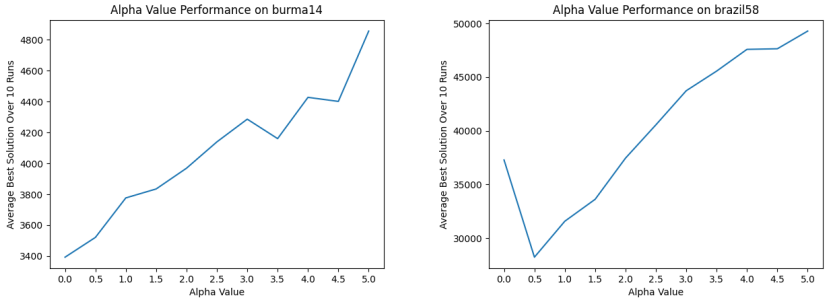

Figure 3: The Impact of a Pheromone Importance Factor on the ACO Algorithm

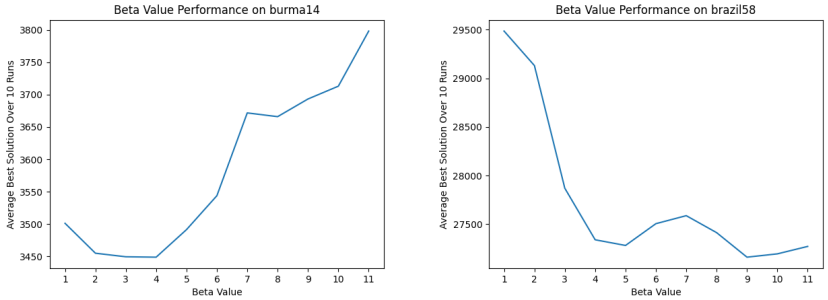

Figure 4: The Impact of a Heuristic Importance Factor on the ACO Algorithm

2 Questions

1. Which combination of parameters produces the best results?
   For both data sets a colony size of 130, alpha value of 0.5 and an evaporation rate of 0.3 produced the best results. A smaller beta value of 3 worked better on burma14 and a higher beta of 9 worked better on brazil58; additionally a smaller Q of 100 worked best on burma14 and a higher Q of 500 worked better on brazil58. With these parameters, optimal results for burma14 (path length of 3323) and close to optimal results for brazil58 were achieved (path length of 26238). The path taken for both of these can be found in the appendix.

   From the graphs above a pheromone evaporation rate of ≤ 0.1 appears to be optimal, although this stifles exploration and prevents the optimal solution from being found by getting the algorithm stuck in a local optima. In the case of the burma14 data set using an evaporation rate of 0.1 (with all other parameters optimised) leads to great but not optimal solutions (on average paths would be a length of 23 longer than the optimal, for a length of 3346).

2. What do you think is the reason for your findings in Question 1?
   A sufficiently large colony size, 130 in this case, and a lower evaporation rate ensured the state space was explored adequately with many different edges getting a higher chance of being taken. However, for this algorithm to be effective it must also exploit certain solutions that are found to have a better fitness. Using a local heuristic (Q) ensures this is done by rewarding shorter paths. Additionally using a lower alpha value (0.5) and higher beta value (3 and 9) helped direct the search by calculating the edge desirability of every edge from the current node.

3. How does each of the parameter settings influence the performance of the algorithm?

   - Colony Size: The amount of ants traversing the graph in a given iteration. More ants traversing the graph increases exploration as edges with a lower probability have a higher chance (more

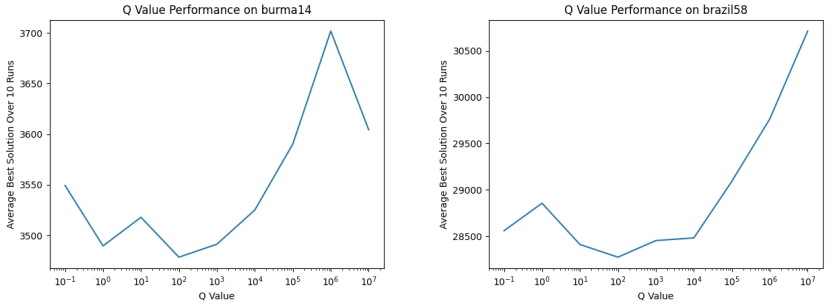

Figure 5: The Impact of a Local Heuristic on the ACO Algorithm

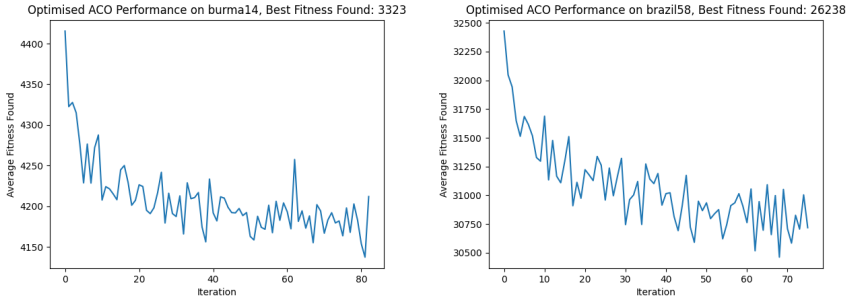

Figure 6: Results for the ACO using Optimised Parameters

ants = more paths taken = higher probability a given path is taken) of being picked and more pheromone is globally added after each iteration, reducing the importance of pheromone (giving less visited edges a higher desirability) and reducing the liklihood the algorithm gets stuck in a local optima as the ants are further spread about the state space.

- Pheromone evaporation rate: Dictates the amount of pheromone that leaves each edge after an iteration, evaporating more makes each edge less desirable at a higher rate favouring edges which are taken more frequently (lower exploration). This is done by in-directly decreasing the desirability value [1] the edge has, at a higher rate. A lower evaporation rate keeps unvisited/ less visited edges 'desirable' for longer by slowing the rate at which they lose desirability.
- Alpha value: Increases the desirability of edges which are visited more frequently by ants, exploiting the search space and any good solutions which have been found recently. This is done by directly increasing the amount of weight an edge's pheromone has in its desirability value. Conversely, decreasing this value allows for greater exploration of the search space by reducing the amount of influence pheromone has on desirability.
- Beta value: Increases the desirability of edges which have a shorter distance, exploring the search space more by acting against the pheromone importance value and adding more weight to the distance of an edge. This is done by directly increasing the desirability for a given edge (smaller edges get larger values as the 'distance' is actually the 1/ distance value found the in heuristic matrix used).
- Q value: Proportionally increases the amount of pheromone ants with better a fitness value deposit, rewarding shorter paths and downplaying the importance of longer more exploitative paths (pheromone is still deposited, enabling exploration). This is done by directly influencing the amount of pheromone deposited on each edge in the path taken by an ant (done according to the Q/fitness), say Q = 1, longer paths, e.g. fitness = 5000, deposit less, 1/5000, as compared to shorter paths, e.g. 3000, which would deposit 1/3000 on each edge.

[1] $edgepheromone^{alpha} * edgeheuristic^{beta}$

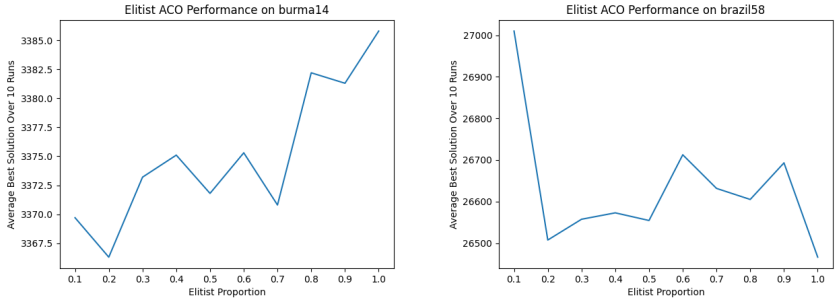

Figure 7: Results for different rates of Elitism on an Elitist ACO Algorithm

4. Can any local heuristic functions be added?
   A local heuristic function (Q/fitness, where Q is a fixed parameter) was applied to help direct the search and reward paths proportionally to their length (ants who took better paths deposited more pheromone). Supplemental to this, a heuristic matrix was applied to ensure pheromone wasn't the only parameter dictating the desirability of an edge (the values in this matrix were set to 1/edge length). Further local heuristics could be used to expand upon or in place of these, i.e. the nearest neighbour heuristic, where distance is the only metric used to select the next city; MST-based heuristics, which constructs a minimum spanning tree to guide the search for the next city, and many more.

   In addition to this local search methods could be added to further direct the search and improve results, e.g. Tabu Search (check the values of the immediate neighbours of a value to see if it can be improved), simulated annealing search (the probability of accepting worse solutions decreases over time), hill climbing search, etc.

5. Can you think of any variations of this algorithm that may improve results? Explain your answer.
   An elitist ACO variation was implemented and produced slightly better results on average when compared to the normal ACO (average solution length of just under 26000 when using optimised parameters). The affect of the elitist parameters can be seen in 7, showing that favouring better solutions to a higher degree (letting fewer, longer solutions deposit pheromone) lead to improved results. This algorithm also appeared to converge on a solution faster but was more prone to getting stuck in local optima (this effect can be seen in the print outs when running the algorithm source code).

   Additionally, the Max-Min Ant System (MMAS) and a hybrid Elitist MMAS were implemented but did not appear to produce significantly better solutions. The results and parameters used can be investigated in the source code.

6. Are there any other nature inspired algorithms that could have produced better results? Explain your answer
   Genetic Algorithms have long been applied to the travelling salesperson problem and have showed quite remarkable success [3]. Genetic algorithms are a subsection of optimisation algorithms which are inspired by the process of survival of the fittest (natural selection), here the fittest solutions are stochastically selected after each iteration to create new solutions that make up the population for the next iteration. Each solution's characteristics have a genetic representation and are altered by operators such as mutation, crossover, etc.

   Here each individual could represent a solution to the TSP, operators such as mutation could then randomly permute the solution and cross-over could take parts of a path and splice them into the path of another solution. Care needs to be taken here to ensure that the resulting individuals are still valid solutions. This process would repeat until a full population of solutions is created or found. These solutions could then be evaluated according to their length and the process repeats until there is convergence or the optimal solution is reached.

.

References

[1] M. L. Carmosino, R. Impagliazzo, V. Kabanets, and A. Kolokolova, "Learning algorithms from natural proofs," in *31st Conference on Computational Complexity (CCC 2016)*, Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2016.

[2] D. A. Helal, "Nature inspired computation," 2023. [Online]. Available: `https://emps.exeter.ac.uk/modules/ECM3412`.

[3] H. Braun, "On solving travelling salesman problems by genetic algorithms," in *International Conference on Parallel Problem Solving from Nature*, Springer, 1990, pp. 129–133.

3 Appendix A

$0 \to 9 \to 8 \to 10 \to 7 \to 12 \to 6 \to 11 \to 5 \to 4 \to 3 \to 2 \to 13 \to 1 \to 0$

4 Appendix B

$0 \to 29 \to 12 \to 39 \to 24 \to 8 \to 31 \to 19 \to 52 \to 49 \to 3 \to 7 \to 21 \to 54 \to 53 \to 1 \to 40 \to 34 \to 9 \to 51 \to 50 \to 46 \to 48 \to 2 \to 47 \to 28 \to 32 \to 44 \to 45 \to 55 \to 33 \to 14 \to 36 \to 13 \to 27 \to 5 \to 18 \to 25 \to 16 \to 35 \to 20 \to 38 \to 10 \to 15 \to 37 \to 41 \to 30 \to 6 \to 4 \to 26 \to 42 \to 11 \to 56 \to 22 \to 23 \to 57 \to 43 \to 17 \to 0$