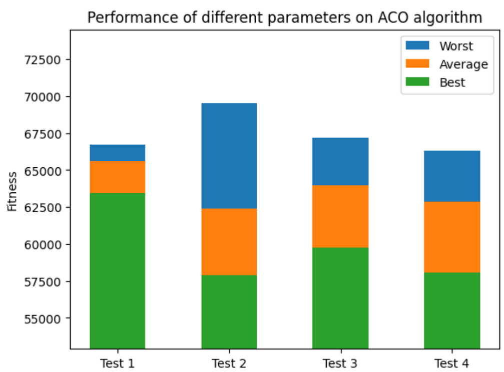
Nature-inspired Computing - Ant Colony Optimisation (ACO)

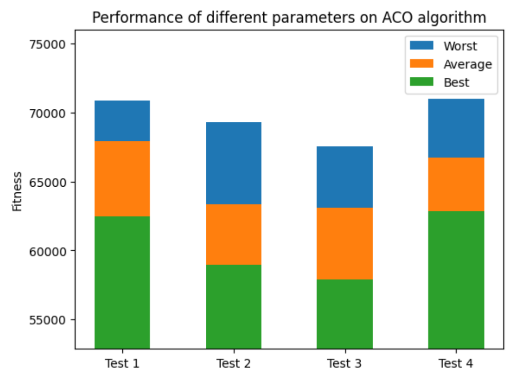
Theodore Proctor

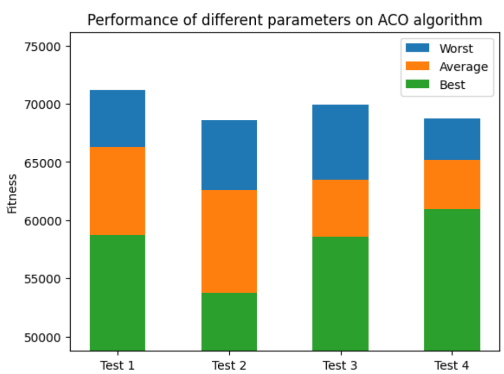
December 2022

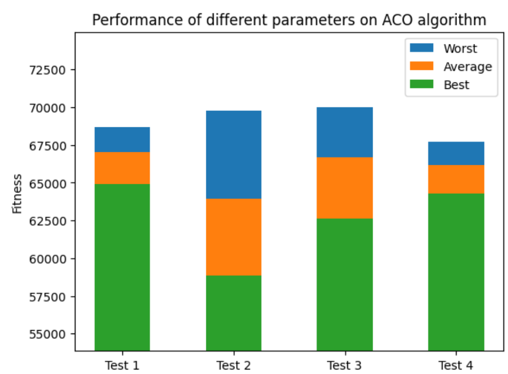
1 Question 1

Although each test aggregates five simulations and produces the best result, there was still some major discrepancy between independent executions of the tests. The following figures display the same tests being run four separate times, with variable random seeds.









Although somewhat inconsistent, we can determine that Test 2 (m = 100, e = 0.5) performs consistently well, acquiring the best solution 3/4 times. Neither Test 3 nor Test 4 performed consistently, with optimal solutions ranging from the best of the set to amongst the worst. Test 1 usually had the least variation between the best, average and worst fitness scores, although this was not always the case. Overall, Test 2 was the most successful, since it consistently scored the best and managed to reduce a path's fitness to 53772, outperforming the other optimal results by 4000-5000.

2 Question 2

Two parameters were changed between Tests 1-4. These are m (population size) and e (pheromone evaporation rate). Test had m = 100, and e = 0.5. This was the larger value for m and e.

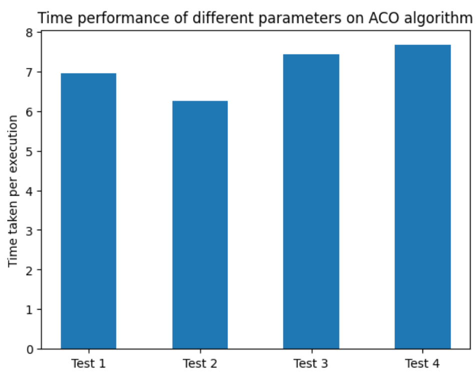
A larger population size can be beneficial in ACO because it allows for a greater number of ants to explore the search space and find good solutions. This can increase the chances of finding a high-quality solution and can also prevent the ants from getting stuck in a local optima. The drawback is that a larger population is typically associated with a higher computational cost. This downside may be irrelevant in this instance of the algorithm since the program is limited by 10000 fitness evaluations, not some other metric.

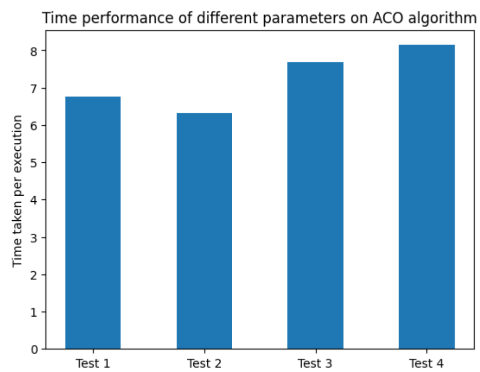
A higher evaporation rate means that the pheromone trails will evaporate more quickly, which can have a number of consequences for the search process. On one hand, a higher evaporation rate can help prevent the ants from getting stuck in a local optima and encourage them to explore a wider range of solutions. On the other hand, a higher evaporation rate can also make it more difficult for the ants to coordinate their search, which can lead to a less efficient search process.

This information is compounded by the evidence provided in question 1. Local optima seem to be avoided, usually resulting in the best fitness score, and the higher than average disparity between the best, worst and average solutions suggest that the search space is being covered more comprehensively. The increase in computational complexity can be ignored due to prerequisites set by the specification, and the inconsistency of results could be attributed to a less coordinated search.

3 Question 3

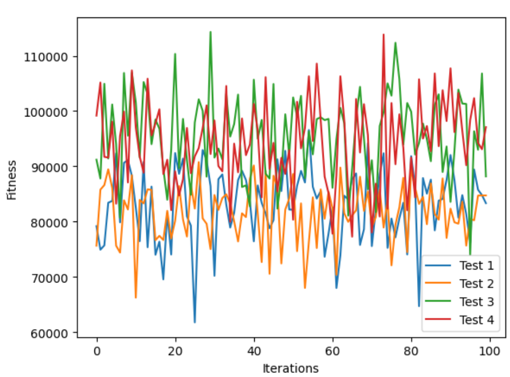
One affect that the different parameter values might have had on the algorithm is the time taken to run.

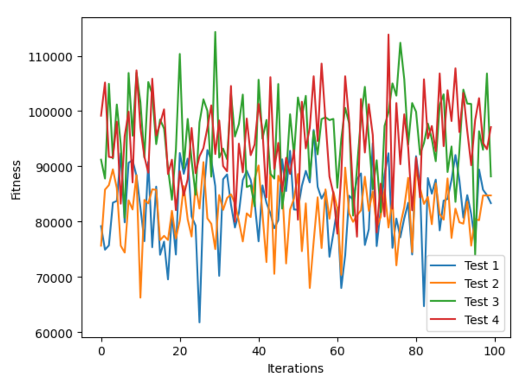




The parameters used in Test 2 have the best result regarding execution time. Tests where m = 10 were expected to perform more slowly, since search algorithms and similarly intensive functions have to be completed more regularly (albeit on a smaller set), resulting in more calls and operations being performed. Test 1 executing more slowly than Test 2 is more surprising, although this could be due to less rounding computations having to be completed when updating the pheromone matrix.

Testing was conducted to examine whether fitness improved over the number of iterations. Although the best solution was almost never found on the first iteration, after the pheromone matrix had adjusted slightly the quality of the solution was seemingly random, as displayed below:





This data does support the notion that the parameters used in Test 2 were optimal, since they scored lower than the others in terms of average fitness.

Another hyperparameter that was investigated was the fitness evaluation method. Two options were tested, a multiplicative formula and a summative one. The summative one was tested since the multiplicative formula would produce very large values for fitness (often in excess of 100,000), which resulted in less effect once the pheromone matrix was updated. On the one hand, the summative formula resulted in no bias between flow and distance (since the range of accepted values are the same for both), meaning that whatever value was produced was not biased. On the other hand, the multiplicative formula meant that very troublesome arrangements (such as high flow over a large distance) would be more likely to be avoided. The benefits of the multiplicative formula outweigh the summative one, but need additional tuning.

Consequently, another hyperparameter that was subject to amelioration was the scalar applied to the fitness value when updating the pheromone matrix. This scalar has been labelled "tuner" within the code. A low value (10,000) for tuner results in less variation between the tests, as well as a higher optimal fitness score. When tuner was very large (100,000), the best solution was often still limited, and tests with a larger population performed less well. I found that approximately 50,000 was the optimal value for tuner, although more research can be conducted. Additionally, more research can be conducted regarding the initial placement (start node) of the path. In this implementation it was decided randomly, but tests could be administered to discover if evenly distributing the ants over all possible start nodes would be preferable. Another area of further research could be the investigation of a different range for the pheromone matrix initialisation.

4 Question 4

A local heuristic function that might improve the efficiency of the ACO algorithm is by considering the distance and flow as well as the pheromone strength. By considering the previous distance a more appropriate decision could be made when considering flow (or vice versa), especially if one of the two is large. Alternatively, a local search algorithm could be implemented, so that a better solution that lies nearby in the solution space may be derived. One such local search algorithm is variable neighbourhood search [5], which is an algorithm that uses a sequence of neighbourhoods to explore the search space. It starts with a small neighbourhood and gradually increases the size of the neighbourhood as the optimisation process progresses. Thus allowing for a better solution to be located and a local optima being breached.

5 Question 5

There are several variations of ACO algorithms that have been proposed for solving the quadratic assignment problem (QAP). Some of the main variations include:

- Max-Min Ant System [8]: This algorithm introduces a "max-min" mechanism to control the amount of pheromone deposited on each solution, which helps to balance exploration and exploitation in the search process.

- Ant Colony System (ACS) [2]: This is another variation of ACO that is similar to the standard algorithm, but it uses a more sophisticated pheromone update rule. This algorithm uses a global pheromone update rule that takes into account the quality of all of the solutions found by the ants, rather than just the best solution.

- Rank-Based Ant System [1]: This variation of ACO uses a rank-based pheromone update rule to guide the search for a solution. In the RBAS algorithm, the pheromone trail is updated based on the rank of the solution rather than its quality. This can help to prevent premature convergence and encourage the exploration of a wider range of solutions.

- Max-Min Ant System with Local Search [7]: This variation of ACO combines the standard algorithm with a local search procedure. The local search procedure is used to improve the quality of the solutions found by the ants. The local search procedure is used to explore the neighborhood of the current solution and find a better solution that is closer to the global optimum. This allows for local optima to be more easily broken, although it is very computationally expensive.

- Ant Colony System with Local Search [3]: This variation of ACO combines the ACS algorithm with a local search procedure. The local search procedure is used to improve the quality of the solutions found by the ants.

Overall, the choice of ACO algorithm will depend on the specific requirements of the problem and the desired trade-off between solution quality and computational efficiency. However, I believe a number of the aforementioned techniques could improve the results, a prime example being any permutation with a local search. Although run-time may be increased, a better solution is almost guaranteed. Alternatively, the maximum number of fitness evaluations could be increased. This would be less effective than adding local search since the best solution appears randomly (as demonstrated in question 3), rather than the solutions improving over time.

6 Question 6

There are several nature-inspired optimisation algorithms that have been applied to the quadratic assignment problem. Genetic algorithms [4] are one type of algorithm. Genetic algorithms are a type of optimisation algorithm that are inspired by the process of natural selection. They involve the use of a population of candidate solutions that evolve over time through the application of genetic operators such as crossover and mutation. They have been applied to the quadratic assignment problem and have been shown to be effective at finding good, approximate solutions to the problem. Unfortunately, this type of algorithm will usually perform less well than ACO's, since the training/improvement methods are vigorous, due to their reliance on randomisation. Although a similarly good solution can be found using a genetic algorithm, it will likely take longer.

Particle swarm optimisation [6] is a type of optimisation algorithm that is inspired by the behaviour of swarms of birds or fish. It involves the use of a population of particles that move through the search space and adjust their movement based on their own experience and the experience of their neighbours. Particle swarm optimisation has been applied to the quadratic assignment problem before and has been shown to be effective. This algorithm shares many similarities with ACO, but since it incorporates the experience of neighbouring particles it is more alike to an ACO variant with a local search, such as the Max-Min Ant System with Local Search. This will likely be an improvement over ACO, although it will be more computationally expensive and more complex to implement.

References

[1] Bernd Bullnheimer, Richard F Hartl, and Christine Strauss. "A new rank based version of the Ant System. A computational study." In: (1997).

[2] Marco Dorigo and Luca Maria Gambardella. "Ant colony system: a cooperative learning approach to the traveling salesman problem". In: *IEEE Transactions on evolutionary computation* 1.1 (1997), pp. 53–66.

[3] Luca Maria Gambardella and Marco Dorigo. "An ant colony system hybridized with a new local search for the sequential ordering problem". In: *INFORMS Journal on Computing* 12.3 (2000), pp. 237–255.

[4] John H Holland. "Genetic algorithms". In: *Scientific american* 267.1 (1992), pp. 66–73.

[5] Nenad Mladenović and Pierre Hansen. "Variable neighborhood search". In: *Computers & operations research* 24.11 (1997), pp. 1097–1100.

[6] Riccardo Poli. "Analysis of the publications on the applications of particle swarm optimisation". In: *Journal of Artificial Evolution and Applications* 2008 (2008).

[7] Thomas Stutzle and Holger Hoos. "MAX-MIN ant system and local search for the traveling salesman problem". In: *Proceedings of 1997 IEEE international conference on evolutionary computation (ICEC'97)*. IEEE. 1997, pp. 309–314.

[8] Thomas Stützle and Holger H Hoos. "MAX–MIN ant system". In: *Future generation computer systems* 16.8 (2000), pp. 889–914.