Ant Colony Optimisation, ECM3412

Nathan Baker

November 2022

1 Problem

There is a set of n university facilities and n locations for them. Between each pair of locations, a distance d is specified, and between each pair of facilities, a flow value f of students is specified. These values are stored inside matrices L and D. I will use ant colony optimisation to assign all facilities to different locations, with the goal of minimizing the sum of the distances. The solution is represented by a vector p in which each entry $p[i]$ takes an unique integer value from 1 to n corresponding to the location values, and the index i denotes the ith facility.

2 Method

L and D were read from disk, and padded with random numbers ranging from the minimum and maximum values from L and D to represent the start node transition values. 0 was then inserted at position 0,0 for the starting position.
1: Numerator matrix, 2: Transition Probability Function, 3: Update Function, 4: Pheromone Evaporation Function, 5: Dorigo et al.'s Edge Selection function [1].

$$NM_{ij} = [d_{ij}(t)]^{\alpha} \times [f_{ij}(t)]^{\beta} \tag{1}$$

$$p_{ij}^{k}(t) = \frac{NM_{ij}}{\sum_{l \in J_i^k} NM_{il}} \tag{2}$$

$$NM_{ij} = NM_{ij} + \frac{p}{l} \tag{3}$$

$$NM_1 = NM_0 \times e \tag{4}$$

$$p_{ij}^{k}(t) = \begin{cases} \frac{[\tau_{ij}(t)]^{\alpha} \cdot [\eta_{ij}]^{\beta}}{\sum_{k \in \text{allowed}_k} [\tau_{ik}(t)]^{\alpha} \cdot [\eta_{ik}]^{\beta}} \\ 0 \end{cases} \tag{5}$$

Next a numerator matrix NM consisting of numerator values (fitness values), derived from the edge selection function described by Dorigo et al. 1996 [1](equation 5) was computed via (equation 1). In my function however, there is no local heuristic η, and τ is given by $d_{ij} \times f_{ij}$, which are values taken from matrices L and D. α and β were set to 1.

Next, over 5 iterations, m ants each make a copy of NM. Then, until all nodes have been visited, the probability of the ant moving to each neighbour from the previous node via (equation 2) is calculated. A neighbour will be chosen based on its probability and the ant will move to it. Previous nodes are set to 0 in the NM copy to prevent backtracking, and a new NM is calculated to refresh the fitness values. Once the ants tour is completed, the fitness of that route f is calculated by the sum of the NM values they chose.

After all ants have completed each iteration, the global NM is updated with new pheromones via (equation 3), where l = the length of the path traversed by the ant, and p = the pheromone strength, given by $\frac{1}{f}$ *(A high fitness in this experiment is considered bad)*. A division by l was included to spread the pheromones out equally across all steps in the path taken. This does not affect the performance of the algorithm however.

Afterwards, the global NM is recalculated and pheromones are evaporated by (equation 4), where e = the evaporation hyper-parameter. The average fitness for all ants in the iteration is then saved.

3 Results

Results were computed on Pop!OS 22.04 with AMD Ryzen 5 3550H CPU. Each experiment having the following hyper-parameters:

- Experiment 1: Five iterations with m=100 and e=0.90
- Experiment 2: Five iterations with m=100 and e=0.50
- Experiment 3: Five iterations with m=10, and e=0.90
- Experiment 4: Five iterations with m=10, and e=0.50

Each experiment was repeated 4 times. Experiment 1 and 2 completed with mean time of 2m 40s, whereas experiment 3 and 4 completed with a mean time of 16s.

Best path from experiment 1: [0, 36, 11, 30, 21, 33, 49, 7, 34, 35, 17, 14, 16, 46, 27, 4, 38, 22, 2, 48, 39, 29, 23, 45, 3, 42, 6, 9, 44, 32, 43, 1, 31, 25, 5, 20, 19, 47, 10, 15, 8, 13, 28, 18, 12, 37, 41, 40, 26, 24, 50]
Traversed in experiment 1, 3
Fitness: 5.6530

Best path from experiment 2: [0, 28, 15, 43, 22, 38, 12, 23, 29, 19, 30, 35, 47, 49, 31, 1, 13, 21, 3, 4, 16, 46, 17, 44, 2, 14, 41, 24, 8, 5, 50, 9, 18, 25, 36, 33, 27, 6, 48, 45, 7, 34, 10, 11, 39, 40, 20, 37, 26, 42, 32]
Traversed in experiment 2, 2
Fitness: 5.7170

Best path from experiment 3: [0, 46, 42, 27, 30, 48, 7, 26, 23, 4, 40, 49, 43, 21, 16, 2, 9, 3, 50, 10, 6, 36, 11, 47, 37, 12, 1, 13, 19, 44, 31, 5, 28, 32, 45, 15, 24, 8, 41, 17, 39, 29, 38, 33, 34, 25, 22, 14, 35, 20, 18]
Traversed in experiment 3, 2
Fitness: 6.0164

Best path from experiment 4: [0, 20, 42, 40, 4, 3, 33, 24, 29, 2, 28, 37, 44, 26, 15, 49, 47, 32, 17, 14, 31, 35, 27, 7, 18, 19, 8, 50, 43, 41, 21, 6, 34, 10, 13, 1, 46, 38, 12, 39, 30, 36, 9, 45, 11, 22, 5, 48, 25, 23, 16]
Traversed in experiment 4, 4
Fitness: 5.4652

Full results, fitness landscapes and construction graph are on page 4.

4 Questions

Question1: Which combination of parameters produces the best results?
Experiment 4 produced the best results out of the 4 experiments, with a best fitness of 5.4652 and mean fitness of 5.7154. Experiment 1 was second, with a best fitness of mean fitness of 5.6530 and a mean fitness of 5.7498.

Question 2: What do you think is the reason for your findings in Question 1?
The results are unexpected, and most likely due to the random nature of the algorithm. Experiment 1 was expected to perform the highest out of the 4, due to it having a population size of 100. This means the search space is 10x that in experiment 3 and 4, thus more area of the fitness landscape could be covered. Experiment 1 (and 3) also had larger e values. This means the amount of pheromones remaining on the trail is grater (see equation 4), thus increasing exploration as less desirable paths' probability of traversal remains comparatively high. I believe experiment 4 producing the best results and not experiment 1, despite having a lower e and m, to simply be a product of the random nature of this algorithm.

Question 3: How do each of the parameter settings influence the performance of the algorithm?
Both e and m are directly proportional to the performance of the algorithm. A higher e, lead to greater exploration, as more paths are desirable. A higher m means a higher population, and therefore a greater search space of the fitness landscape. Balancing α and β had no positive effects on the results.

		fitness			
alpha	beta	1	2	3	mean
2	1	6.399	7.172	6.618	6.729
1	2	6.950	6.757	6.991	6.899

These results were obtained with the hyper parameters outlined in experiment 4, and resulted in a decrease in mean performance from experiment 1 by 1.0642, and 1.0986 from experiment 4. This is most likely due to, without the inclusion of a local heuristic (outlined in the next question), that the inclusion of β or α powers scale the fitness of the path equally, even if β and α are distinct values. This is because d and f are multiplied together in (equation 1), to compute the fitness in the formation of the numerator matrix.

Question 4: Can you think of a local heuristic function to add?
A modification to our already existing (equation 1) would be a simple addition of a local heuristic, which encourages paths with a smaller distance, but higher flow to be traversed:

$$NM_{ij} = \frac{1}{[d_{ij}(t)]^{\alpha}} \times [f_{ij}(t)]^{\beta}$$

Here the inverse of the distance is taken to discourage larger distances, and allow for seperate scaling of d and f via β and α hyperparameters.

Question 5: Can you think of any variation for this algorithm to improve your results? Explain your answer.
Setting MIN and MAX pheromone amounts on paths could increase exploration and avoid local optima, as no path's probability of traversal will tend to 0. We could also implement elitism to decrease randomness, and steer the ants in the right direction.

Here are the results after an implementation of min and max pheromones on experiment 1, with max=5000, min=100:

Experiment 1 - with min & max pheromones

	1	2	3	4	mean
Fitness	5.6422	5.8278	5.7835	5.6791	5.7498

This experiment resulted in an identical mean to experiment 1, which was expected to be the most successful experiment. It did not however manage to beat the performance exhibited by experiment 4, and had a mean difference of 0.0344 from it. I do believe however after more iterations, this experiment would of performed the best due to the increase in exploration it encourages over the other experiments, and the results from 4 were very 'lucky'.

An experiment with identical hyperparameters to experiment 4 was then completed, with the addition of minimum and maximum pheromone strength and elitism. Elitism was accomplished by the fittest ant surviving through each generation.

Experiment4 - with elitism

	1	2	3	4	mean
Fitness	5.7604	6.7572	6.2809	6.9509	6.33735

This experiment did not improve over Experiment 1 or 4, in either mean or lowest value. The best path traversed by this experiment was:

[0, 45, 47, 34, 19, 25, 6, 27, 12, 5, 31, 15, 35, 11, 13, 1, 22, 37, 7, 32, 17, 9, 4, 30, 49, 16, 8, 3, 33, 41, 43, 21, 29, 42, 38, 40, 26, 23, 2, 50, 14, 18, 20, 24, 44, 39, 36, 28, 10, 46, 48]

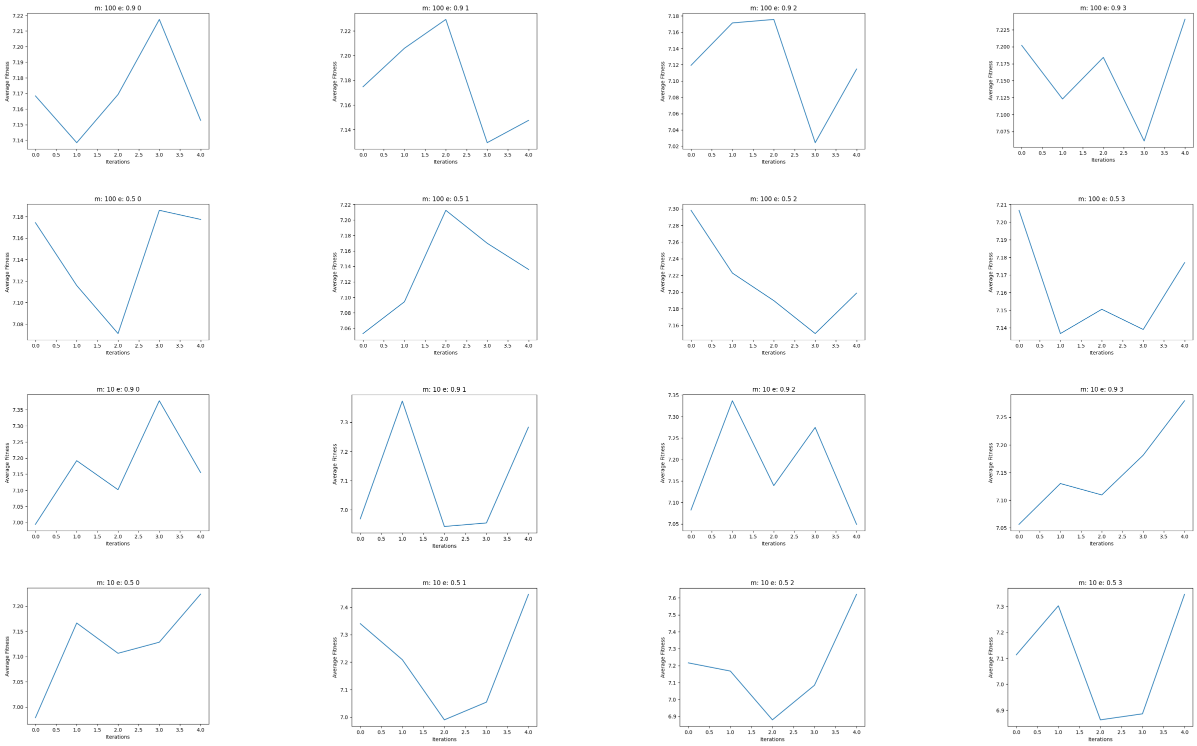
Achieving a fitness of 5.7604. I do believe this to be another manifestation of the random nature of the algorithm however, and that more iterations than the 5 specified would have resulted in more stable and expectable results (although this would of greatly increased processing time); likely achieving better results than all other experiments.

The affects of incorporating α and β were discussed in question 3 and 4.

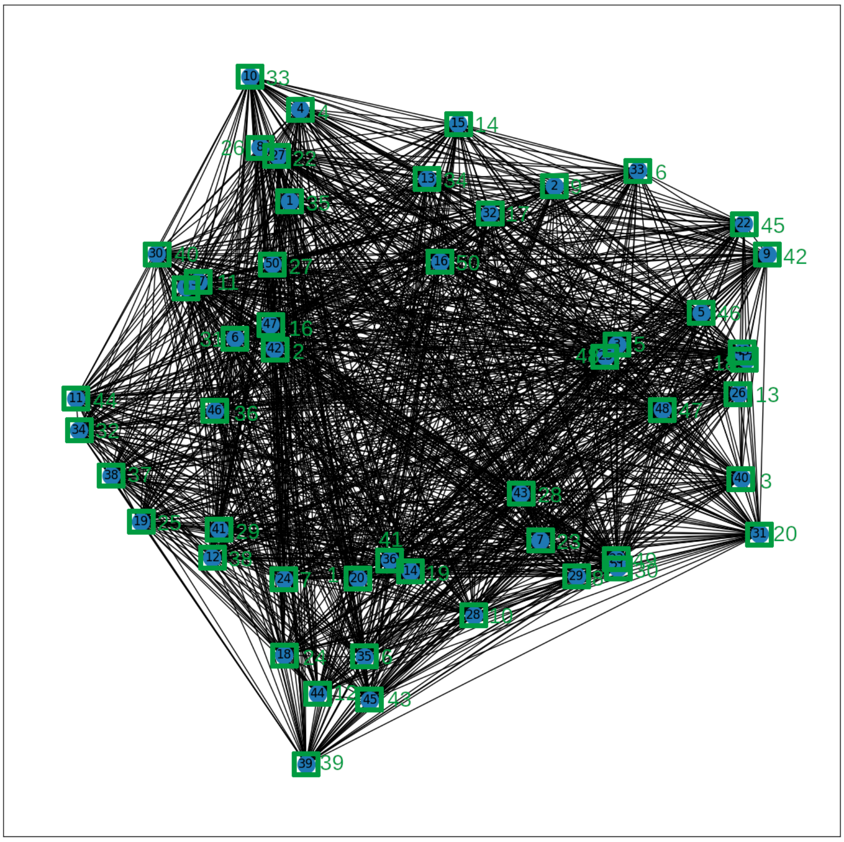
Question 6: Do you think of any any other nature inspired algorithms that might have provided better results? Explain your answer
Particle swarm offers the inclusion of gbest and pbest into its algorithm as appose to stigmergy. And after interpreting velocities as probabilities via an activation function, PSO could be applied to this discrete problem. Comparative to ACO, PCO falls less frequently in local optima, which could potentially lead to better results. It does however impose problems with parameter selection due to poor exploration [2].

Genetic algorithms offer another solution to this problem, having the ability to solve the problem of local optima via gene mutation. However early convergence is a common problem in this algorithm, hence finding the best solution or a better solution than PCO and ACO is not guaranteed [2]. An example of a potential GA for the problem tackled in this paper is a permutation-based genetic algorithm that could be constructed with x-point crossovers, and a mutation via a swap between values to maintain the set.

experiment	1					2					3					4				
	1	2	3	4	mean	1	2	3	4	mean	1	2	3	4	mean	1	2	3	4	mean
Fitness	5.7810	5.7430	5.6530	5.8223	5.7498	5.9809	5.7170	5.9667	6.0855	5.9375	6.7500	6.0164	6.4856	6.5797	6.4579	5.7582	6.0389	5.5996	5.4652	5.7154



A construction graph was created from the results of experiment 4,4. The blue value represent the location, the green values represent the facility, and the distance between each node represents the size of the numerator matrix value (distance d × flow f) between them - a larger distance relates to a higher numerator value:



References

[1] Marco Dorigo, Vittorio Maniezzo, and Alberto Colorni. “Ant sytem: Optimization by a colony of cooperating agents”. In: *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics* 26 (1) (Jan. 1996), pp. 29–41.

[2] Sherylaidah Samsuddin, Mohd Othman, and Lizawati Mi Yusuf. “A REVIEW OF SINGLE AND POPULATION-BASED METAHEURISTIC ALGORITHMS SOLVING MULTI DEPOT VEHICLE ROUTING PROBLEM”. In: *International Journal of Software Engineering and Computer Systems* 4 (Aug. 2018), pp. 80–93. DOI: `10.15282/ijsecs.4.2.2018.6.0050`.