ECM3412 - Coursework exercise

1. Introduction

This report is to analyse an ant colony optimization algorithm on a quadratic assessment problem. A quadratic assessment problem is an optimization problem that has an objective to find minimum cost allocation between a set of facilities to a set of locations and taking the cost as sum distance-flow products[1]. Ant Colony Optimization is a metaheuristic algorithm that has an approach in constructing solutions based on the quality of previously obtained results[2].

In the experiments, there are 4 experiments to run. There are three parameters setting that are included in the experiments which are m which is the total number of ants, e is evaporation rate and number of iteration. Each experiments contains different parameters:

1. Experiment 1: ACO with m = 10, e = 0.90, iteration = 10000
2. Experiment 2: ACO with m = 100, e = 0.50, iteration = 10000
3. Experiment 1: ACO with m = 100, e = 0.90, iteration = 10000
4. Experiment 1: ACO with m = 10, e = 0.50, iteration = 10000

2. Results and Analysis

I have implemented the algorithm using the python modules. The tables below show different fitness values as a solution to the experiment.

Explanation:

Due to lack of understanding of the assignment, I could not manage to run the experiment 1 and 2 with iteration 10000 and also another trial for experiment 3 and 4. What I could learn from experiment 1 and 2 is that they took a long time to produce outputs.

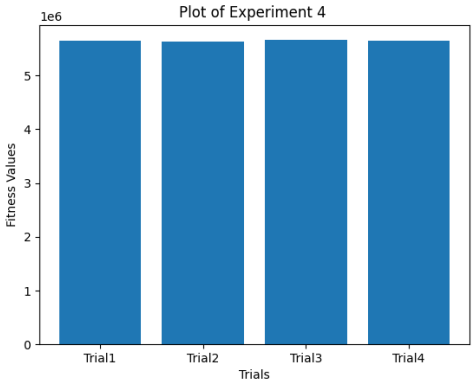
Improvement:

- Since I do not have fitness for 1 and 2, I would compute the average of fitness for 5 trials and compare the values
- I would make bar charts for each experiment and each trial and use that to observe clear results from the algorithm.

2.1 Tables of Results

	Trial1	**Trial2**	**Trial3**	**Trial4**	**Average**
Experiment3	5665964	5651568	5661394	5648236	5656790.5
Experiment4	5643544	5629152	5661394	5648236	5645581.5

2.2 Graph



3. Questions

Question1: Which combination of parameters produces the best results?

I think the best combination of parameters should be m= 10. e = 0.90. Iteration = 10.000

Question 2: What do you think is the reason for your findings in Question 1?

It is because when you compare the average of fitness value, experiment 4 has lower fitness. It means that the parameters give solutions in shorter distances between facilities and location. I think the reason is that there is a less number of ants and the evaporation rate is faster which could help the ants to eliminate the pheromone trails quickly and search the optimum solutions efficiently.

Question 3: How do each of the parameter settings influence the performance of the algorithm?

- M: I could not manage to implement a time function to compute the execution time. However, as I ran the code, I noticed that the number of ants affects the execution time. As the value increases, the execution time also increases. The algorithm needs to analyse more ants and their paths, which leads to increase in processing time
- E: influence the ability to search suboptimal path and delays faster convergence of ants

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

Local heuristic function is a good solution to prevent having local optimum solution. In the ACO algorithm, the ant pants is generated by choosing random selection for the next path. The solution: initialise random solution by swapping facilities on

several locations where locations are chosen based on a probability, choose higher value of fitness from swapping new location to choose previous locations as best solution.

Question 5: Can you think of any variation for this algorithm to improve your results? Explain your answer.

Another variation of ACO that can improve the results is Max-Min Ant System (MMAS). MMAS allows the best solutions for pheromone during the pheromone trail update. This variation is better because it includes two different local search algorithms which are 2-opt algorithm and tabu search. It helps to alternate the pheromone trail update, analyse more solutions quickly, and avoid getting stuck in local optimum solutions.

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

Another nature inspired algorithm that can be used and show better performance is Genetic Algorithm. A genetic algorithm is done by generating a population of random solutions (chromosomes) are created, evaluates each individual by their fitness function, and next iteration creates a new population, then evaluates them until termination criteria is reached. This algorithm would perform better because it analyses all possible solutions as whole and include them in the search space which lead to optimal solutions. Moreover, research has found that GA performs faster than ECO[3].

4. Further Experiment

Many researches have shown that hybrid approaches, a combination of two algorithms would show better performances in finding the best solutions. For example, the ideas are implementing this algorithm with simulated annealing or genetic algorithm(Cooperative Genetic Ant System) and implementing a local heuristic function. Several steps could be added to the hybrid algorithm:

1. Create a random initial permutation
2. Generate initial solution by using local search using Simulated Annealing (number of iteration is one of the factor)
3. Create of the pheromone matrix for the solution
4. Evaluates solution
5. Terminate criteria

5. Bibliography

1. Loiola, Eliane Maria, et al. "A survey for the quadratic assignment problem." *European journal of operational research* 176.2 (2007): 657-690.

2. Maniezzo, Vittorio, Luca Maria Gambardella, and Fabio de Luigi. "Ant colony optimization." *New optimization techniques in engineering*. Springer, Berlin, Heidelberg, 2004. 102.

3. Mukhairez, Hosam H., and Ashraf YA Maghari. "Performance comparison of simulated annealing, GA and ACO applied to TSP." *International Journal of Intelligent Computing Research (IJICR)* 6.4 (2015).