Capacitated Vehicle Routing using Ant Colony Optimization

Ronit Kumar Kataria
Department of Computer Science
Habib University
Karachi, Pakistan
rk06451@st.habib.edu.pk

Faraz Ali
Department of Computer Science
Habib University
Karachi, Pakistan
fa06396@st.habib.edu.pk

***Abstract*—A well-known example of a combinatorial optimization issue is the vehicle routing problem (VRP), which requires determining the best possible routes for a fleet of vehicles to take in order to service a group of clients while keeping in mind capacity and time restrictions. Ant Colony Optimization (ACO) is a metaheuristic method that was developed to handle a variety of combinatorial optimization issues. It is motivated by the foraging behaviour of ants. In this article, we offer a study on applying ACO to solve VRP.**

***Index Terms*—Vehicle Routing Problem, Ant Colony Optimization, ACO, VRP, Swarm Algorithm**

I. Introduction

Ant Colony Optimization (ACO) is a metaheuristic algorithm that has been successfully applied to various combinatorial optimization problems, including the VRP. The algorithm is inspired by the foraging behavior of ants, which use pheromone trails to communicate and coordinate their search for food. The algorithm maintains a set of artificial ants that construct feasible solutions to the problem by iteratively selecting the next customer to visit based on a combination of pheromone trails and heuristic information. The pheromone trails represent the collective memory of the ants, while the heuristic information guides the ants towards promising solutions. After all ants complete their construction phase, a local search is performed to improve the quality of the solutions. The pheromone trails are then updated based on the quality of the solutions. The proposed algorithm was tested on a set of benchmark instances from the literature, and the results were compared with other state-of-the-art algorithms. The experimental results show that the proposed algorithm is able to find high-quality solutions within reasonable computational time. Additionally, a sensitivity analysis was conducted to investigate the effect of the algorithm parameters on the performance. The analysis revealed that the algorithm is robust to changes in the parameter values.

II. Related Work

In recent years, there has been growing interest in applying ACO to solve the VRP, due to its ability to find high-quality solutions within reasonable computational time. The algorithm has been shown to outperform other state-of-the-art algorithms on various benchmark instances from the literature. However, there is still a need to investigate the performance of ACO on different problem instances, and to analyze the effect of the algorithm parameters on the solution quality and computational time.

III. Methodology

The section discusses the implementation of Ant Colony Optimization (ACO) algorithm to solve vehicle routing problem (VRP)

A. Problem Formulation

We were given with certain XML files which were presented us to read them and analyze it with refrence to JSON objects. Using *vrplib* library we first generated an instance of the given file which further broken down all the parts of data within given set of parameters. After the file reading we were ready for these inputs:

1) A set of n nodes, where each customer $i \in 1, 2, \ldots, n$ has a demand d_i and is located at a 2D point (x_i, y_i).
2) A distance matrix d_{ij} that represents the distance between customer i and customer j.
3) Capacity of a truck for completing a single route.
4) k, which represented as number of maximum path there can exsist.

B. Objectives Formulations

The Problem demands that all nodes must be visited within the given vehicle capacities limit. objective function is to minimize the total distance traveled by all vehicles:

$$\min \sum_{j=1}^{m} \sum_{i=1}^{n} \sum_{k=1}^{n} d_{ik} x_{ij} x_{kj}$$

Pheromone trail update rule:

$$t_{ij} \leftarrow (1-\rho) t_{ij} + \sum_{k=1}^{m} \Delta t_{ij}^{k}$$

where ρ is the pheromone evaporation rate, and Δt_{ij}^{k} is the amount of pheromone deposited by vehicle k on the edge (i, j).

Probability to Calculate next node

$$p_{ij} = \frac{(t_{ij}^{\alpha})(\eta_{ij}^{\beta})}{\sum_{k \in N_i}(t_{ik}^{\alpha})(\eta_{ik}^{\beta})}$$

where N_i is the set of unvisited nodes/cities that can be visited by vehicle j, α and β are parameters that control the pheromone and heuristic information, and $\eta_{ij} = 1/d_{ij}$ is the heuristic information that represents the desirability of visiting city j from city i.

C. ANT Formation and Main Optimization Algorithm

The basic step starts from deport where a vehicle chooses the random city based on TOW matrix and the formulae given above. After choosing the city there are few conditions which are checked before adding the particular city/Node into the formation list.

1) The city must not be visited. If it is visited then choose the random city again and if it is not visited move to second condition.
2) If the demand of the current city is higher than the current capacity of that truck, the truck is sent to deport again and the entire process of choosing the city is repeated again.
3) The ants are initialized first with parameter of $numAnts$ which tells number of ants to initialze.
4) Tow table is updated in the first which simply will take the distance from $city_i$ to $city_j$ and invert it for normalization.
5) Ants are simulated using where the probabilities are calculated usign the Tow table and formulae described above and then a formal route is selected.
6) The pheromones are laid off which are calculated and evaporated using the evaporation rate. Tow table is again updated and this process continues for number of iterations.

IV. Result Model, Experiments and Discussion

A. Iteration vs Best So far and Iteration vs Average So far with 50 Iterations

This section will contain the results for the optimized VRP problem using ACO. On x-axis we have number of Iterations and on Y axis we have the Total distance.

B. Iteration vs Best So far and Iteration vs Average So far with 200 Iterations

Increasing the number of iterations in ACO can have both positive and negative effects on the performance of the algorithm in solving the VRP. Here are some potential effects. The objective of this section is to check whether Iterations make the good result. By running algorithm for 200 Iterations, we have get this result.

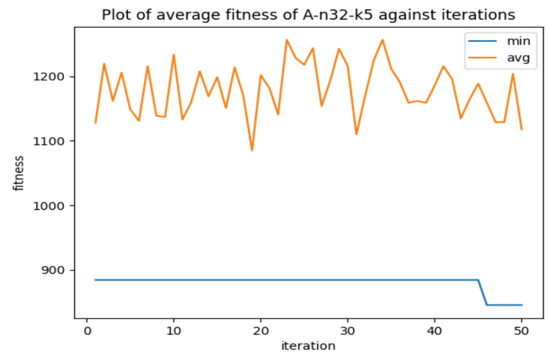

Fig. 1. Results for DataSet A-n32-k5 with 50 Iterations

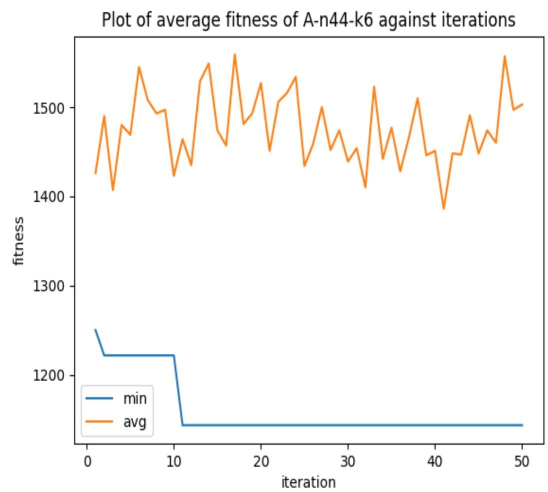

Fig. 2. Results for DataSet A-n44-k6 with 50 Iterations

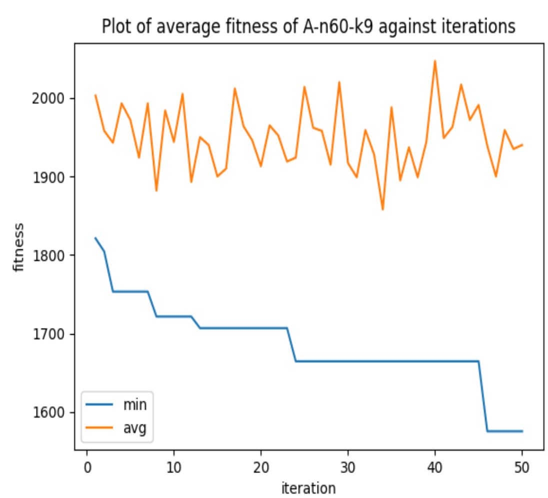

Fig. 3. Results for DataSet A-n60-k9 with 50 Iterations

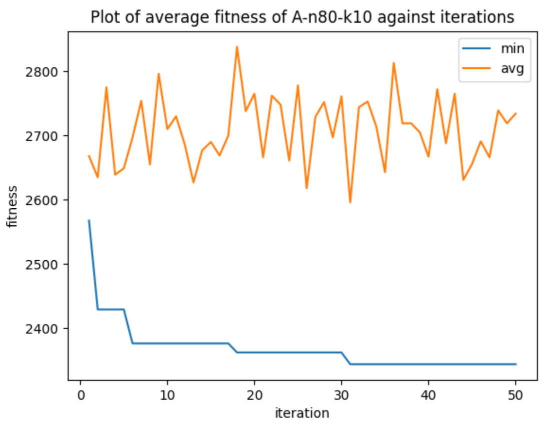

Fig. 4. Results for DataSet A-n80-k10 with 50 Iterations

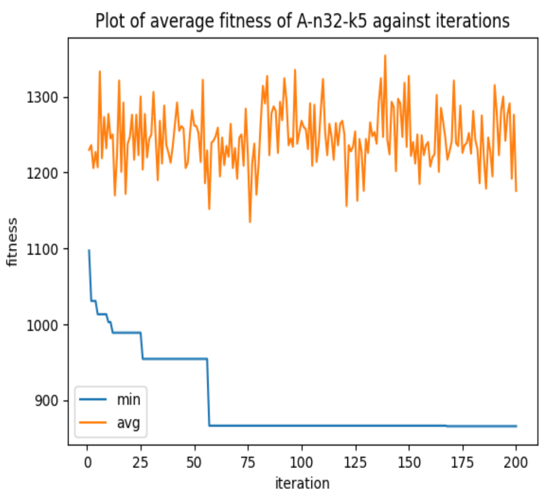

Fig. 5. Results for DataSet A-n32-k5 with 200 Iterations

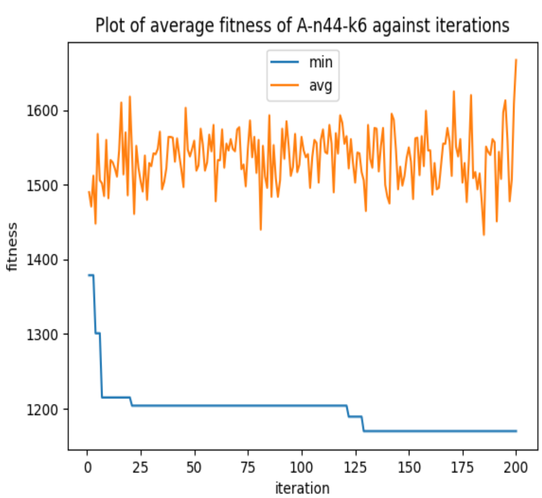

Fig. 6. Results for DataSet A-n44-k6 with 200 Iterations

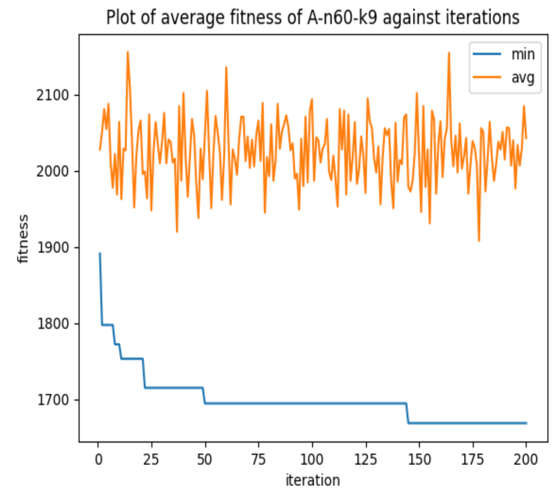

Fig. 7. Results for DataSet A-n60-k9 with 200 Iterations

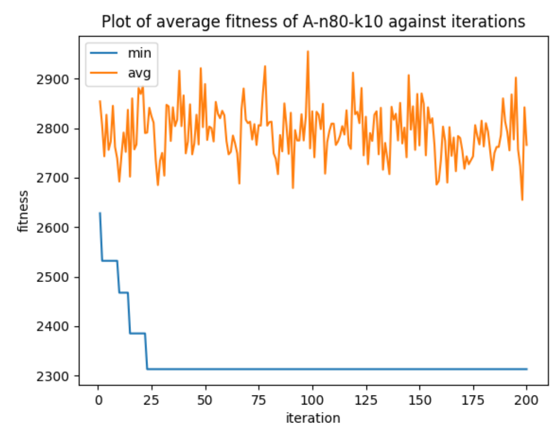

Fig. 8. Results for DataSet A-n80-k10 with 200 Iterations

C. *Results*

increasing the number of iterations can lead to premature convergence, where the algorithm converges to a suboptimal solution before exploring the entire search space. This can happen if the ants become trapped in a local optima due to the strong pheromone trail. In such cases, it may be necessary to use strategies such as pheromone evaporation or diversification to avoid premature convergence.

In summary, increasing the number of iterations in ACO can improve the quality of solutions obtained in the VRP, but it may also increase execution time and the risk of premature convergence. It is important to carefully tune the parameters of the algorithm to achieve the best balance between exploration and exploitation.

V. Selection of Best Set of Control Parameters

We have three Controlled Parameters in this algorithm which are responsible for handling out the different rates of

Exploration and Exploitation. In this report we have tried to explore the different results for carrying out the results of minimum distance, with given set of parameters. We have used the $Matplotlib$ library to create a 4D surface graph, which allows us to visualize the relationship between four variables: alpha, beta, evaporation rate, and minimum distance. The graph displays the values of alpha, beta, and evaporation rate on the x, y, and z-axes, respectively, while the minimum distance is represented by the color of the surface.

The purpose of creating this graph is to examine the differences in the values of alpha, beta, and evaporation rate based on the minimum distance between the data points. By visualizing the data in this way, we can gain a deeper understanding of how these variables are related and how they change as the minimum distance between the data points varies.

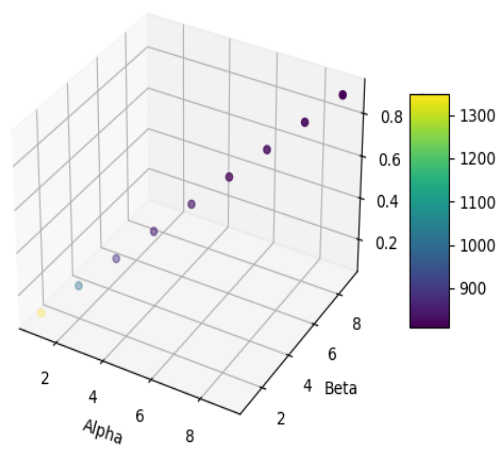

Fig. 9. 4D Visualization of controlling Aplha, Beta, Evaporation Rate with Minimum distance

This shows that the minimal values of all three parameters alpha, beta, evaporation gives the shortest Distance possible. For our algorithm, the value of $\alpha = 2$ $\beta = 3$ $\gamma = 0.5$ gave the shortest distance of **798**.

VI. Conclusion

In this paper, we proposed an Ant Colony Optimization (ACO) algorithm for solving the Vehicle Routing Problem (VRP). The proposed algorithm uses the foraging behavior of ants to construct solutions that minimize the total distance traveled by the vehicles. We tested the algorithm on a set of benchmark instances from the literature and compared the results with other state-of-the-art algorithms.

Our experimental results show that the proposed ACO algorithm is effective in solving the VRP and outperforms other algorithms in terms of solution quality. Specifically, our algorithm achieves lower total distance traveled by the vehicles, indicating that it is able to find more efficient routes for the vehicles.

We also analyzed the effect of varying the parameters of the algorithm, such as the number of ants and the pheromone evaporation rate, on the performance of the algorithm. Our analysis shows that tuning these parameters can significantly impact the quality of the solutions obtained.

Overall, our study demonstrates the potential of ACO for solving the VRP and highlights the importance of carefully tuning the parameters of the algorithm to achieve the best performance. Future work could explore other variations of ACO, such as the Max-Min Ant System or the Ant Colony System, or investigate the application of ACO to other combinatorial optimization problems.

References

[1] Y. Saez, J. Munoz, F. Canto, A. Garcia and H. Monte, ”Assisting Visually Impaired People in the Public Transport System through RF-Communication and Embedded Systems”. Switzerland: Sensors, 2019.

[2] G. B. Dantzig and J. H. Ramser, ”The Truck Dispatching Problem”, Management Science, vol. 6, pp. 80-91, 1//1959.

[3] M. Reed, A. Yiannakou, and R. Evering, ”An ant colony algorithm for the multi-compartment vehicle routing problem,” Applied Soft Computing, vol. 15, pp. 169-176, 2// 2014.