NETAJI SUBHAS UNIVERSITY OF TECHNOLOGY

OPTIMIZATION TECHNIQUES PROJECT ON VEHICLE ROUTING PROBLEM WITH ANT COLONY OPTIMIZATION ALGORITHM

SEMESTER-6
ECAM-2

Submitted to:

Ms. Aarti Jain

Mr. Shivam Ahuja

Submitted by:
Amogh Alone - 6584
Nikhil Vashisht – 6594
Robert - 6562
Dhruv - 6564

INDEX

ABSTRACT

This research applies the meta-heuristic method of ant colony optimization (ACO) to an established set of vehicle routing problems (VRP). The procedure simulates the decision-making processes of ant colonies as they forage for food and is similar to other adaptive learning and artificial intelligence techniques such as Tabu Search, Simulated Annealing and Genetic Algorithms. Modifications are made to the ACO algorithm used to solve the traditional traveling salesman problem in order to allow the search of the multiple routes of the VRP. Experimentation shows that the algorithm is successful in finding solutions within 1% of known optimal solutions and the use of multiple ant colonies is found to provide a comparatively competitive solution technique especially for larger problems. Additionally, the size of the candidate lists used within the algorithm is a significant factor in finding improved solutions, and the computational times for the algorithm compare favorably with other solution methods.

In computer science and operations research, the ant colony optimization algorithm (ACO) is a probabilistic technique for solving computational problems which can be reduced to finding good paths through graphs. Artificial ants stand for multi-agent methods inspired by the behaviour of real ants. The pheromone-based communication of biological ants is often the predominant paradigm used.[2] Combinations of artificial ants and local search algorithms have become a method of choice for numerous optimization tasks involving some sort of graph, e.g., vehicle routing and internet routing.

As an example, ant colony optimization[3] is a class of optimization algorithms modelled on the actions of an ant colony.[4] Artificial 'ants' (e.g. simulation agents) locate optimal solutions by moving through a parameter space representing all possible solutions. Real ants lay down pheromones directing each other to resources while exploring their environment. The simulated 'ants' similarly record their positions and the quality of their solutions, so that in later simulation iterations more ants locate better solutions.[5] One variation on this approach is the bees algorithm, which is more analogous to the foraging patterns of the honey bee, another social insect.

INTRODUCTION

In the **Ant Colony Optimization** algorithms, an artificial ant is a simple computational agent that searches for good solutions to a given optimization problem. To apply an ant colony algorithm, the optimization problem needs to be converted into the problem of finding the shortest path on a weighted graph. In the first step of each iteration, each ant stochastically constructs a solution, i.e. the order in which the edges in the graph should be followed. In the second step, the paths found by the different ants are compared. The last step consists of updating the pheromone levels on each edge.

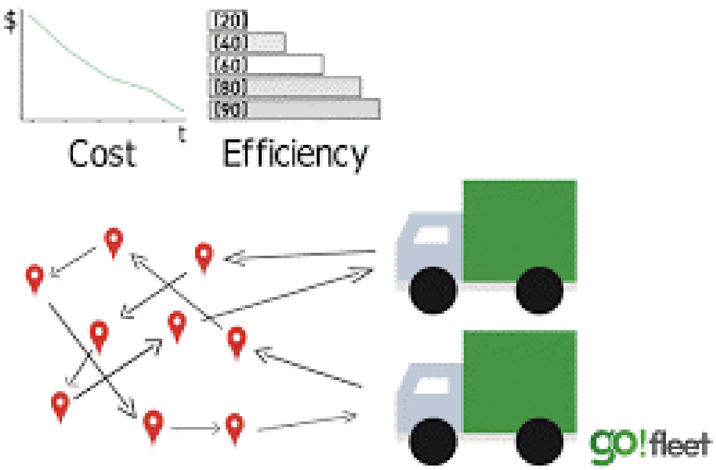
In the **Vehicle Routing Problem (VRP)**, the goal is to find optimal routes for multiple vehicles visiting a set of locations. (When there's only one vehicle, it reduces to the Traveling Salesperson Problem.)

But what do we mean by "optimal routes" for a VRP? One answer is the routes with the least total distance. However, if there are no other constraints, the optimal solution is to assign just one vehicle to visit all locations, and find the shortest route for that vehicle. This is essentially the same problem as the TSP.

A better way to define optimal routes is to minimize the length of the longest single route among all vehicles. This is the right definition if the goal is to complete all deliveries as soon as possible. The VRP example below finds optimal routes defined this way.

In later sections, we'll describe other ways of generalizing the TSP by adding constraints on the vehicles, including:

- Capacity constraints: the vehicles need to pick up items at each location they visit, but have a maximum carrying capacity.
- Time windows: each location must be visited within a specific time window.

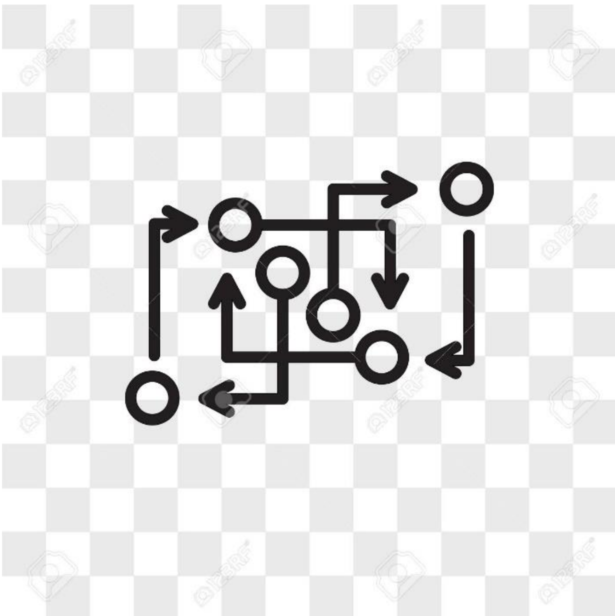


METHODOLOGY

- The methodology for solving the Vehicle Routing Problem (VRP) using Ant Colony Optimization (ACO) typically involves the following steps:
- Problem formulation: Define the VRP problem, including the number of vehicles, capacity constraints, customer locations, and associated demands.
- Graph representation: Represent the VRP problem as a graph, where nodes represent customers and edges represent distances or costs between customers.
- Initialization: Initialize the pheromone levels on the edges of the graph. Typically, this is done by setting all pheromone levels to a constant value or a small random value.
- Ant solution construction: For each ant, construct a solution by iteratively selecting the next customer to visit based on a combination of pheromone levels and heuristic information (such as distance or cost). This is usually done using a probabilistic rule, such as the probability proportional to the product of pheromone level and heuristic information.

- Local pheromone update: After each ant completes its solution, update the pheromone levels on the edges of the graph locally, typically using a local update rule that increases the pheromone level of edges used by the ant and evaporates the pheromone on all edges.
- Global pheromone update: After all ants have completed their solutions, update the pheromone levels on the edges globally, typically using a global update rule that deposits additional pheromone on edges that were used by better solutions and evaporates the pheromone on all edges.
- Iteration: Repeat steps 4-6 for a certain number of iterations or until a termination condition is met, such as reaching a maximum number of iterations or a convergence criteria.
- Solution evaluation: Evaluate the quality of the solutions generated by the ants based on objective criteria, such as total cost, total distance, or number of vehicles used.
- Termination: Terminate the algorithm when the termination condition is met, and output the best solution found during the iterations.

- Post-processing: Optionally, apply any post-processing techniques, such as local search or solution improvement heuristics, to further improve the quality of the solution.
- Output: Output the final solution, which represents the optimized vehicle routes that satisfy the VRP constraints.



DATASET & OUTPUT

Here are links we used for dataset –

1. https://www.dca.fee.unicamp.br/projects/infobiosys/vrp/E-n22-k4.txt

2. https://www.dca.fee.unicamp.br/projects/infobiosys/vrp/E-n33-k4.txt

3. https://www.dca.fee.unicamp.br/projects/infobiosys/vrp/E-n51-k5.txt

Here is the link for code –

https://github.com/nikhilvashisht/VehicleRoutingProblem

OUTPUTS →

- For 1st dataset –

```
Route #1: 17 20 18 15 12
Route #2: 16 19 21 14
Route #3: 13 11 4 3 8 10
Route #4: 9 7 5 2 1 6
Cost 375
```

- For 2nd dataset –

```
Route #1: 1 15 26 27 16 28 29
Route #2: 30 14 31
Route #3: 3 5 6 10 18 19 22 21 20 23 24
25 17 13
Route #4: 2 12 11 32 8 9 7 4
Cost 835
```

- For 3rd dataset –

```
Route #1: 5 49 10 39 33 45 15 44 37 17 12
Route #2: 47 4 42 19 40 41 13 18
Route #3: 46 32 1 22 20 35 36 3 28 31 26 8
Route #4: 6 14 25 24 43 7 23 48 27
Route #5: 11 16 2 29 21 50 34 30 9 38
cost 521
```

REFERENCES

https://www.sciencedirect.com/science/article/abs/pii/S1474034604000060

https://en.wikipedia.org/wiki/Vehicle_routing_problem

https://en.wikipedia.org/wiki/Ant_colony_optimization_algorithms

https://github.com/nikhilvashisht/VehicleRoutingProblem