A Project Report

On

HOSPITAL ROUTING PROBLEM

USING ACO-VRPTW ALGORITHM

BY

B Shreyas Bhat

2019B1A80969G

Under the supervision of

Dr. Sumit Biswas

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS OF

BIO F266: STUDY PROJECT

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE PILANI (RAJASTHAN)

GOA CAMPUS

(APRIL 2021)

ACKNOWLEDGMENTS

I am extremely grateful to my supervisor, Dr. Sumit Biswas, for his invaluable advice, continuous support, and patience during my study-oriented project. His encouragement and ample experience have encouraged me in my research, even in challenging times to overcome roadblocks. There were a lot of learning curves in the journey of this project which will help me in advancing my research career

Birla Institute of Technology and Science-Pilani,

Goa Campus

Certificate

This is to certify that the project report entitled "**HOSPITAL ROUTING PROBLEM USING ACO-VRPTW ALGORITHM**" was submitted by Mr. B Shreyas Bhat (ID No. 2019B1A80969G) in fulfillment of the requirements of the course BIO F266, Study Project Course embodies the work done by him under my supervision and guidance.

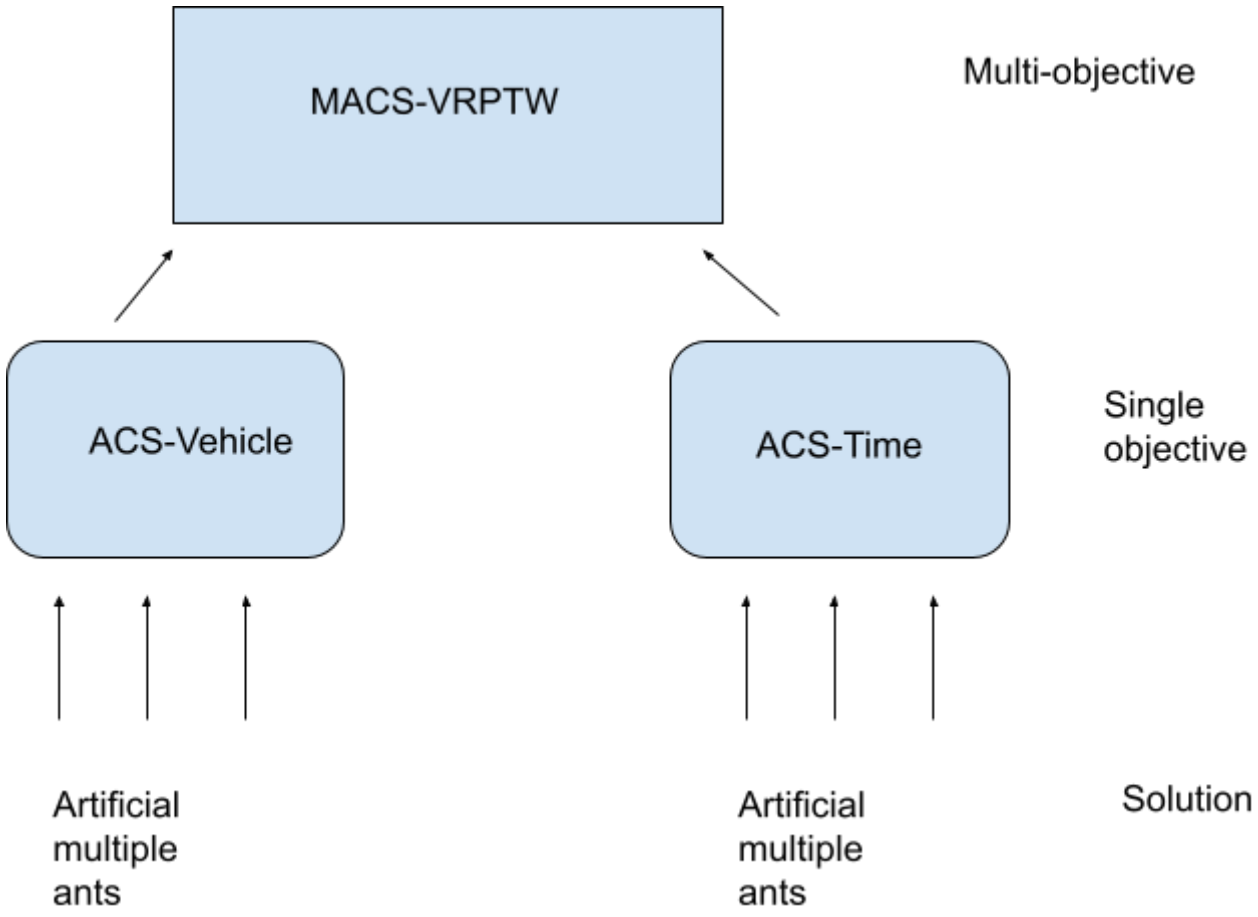
Date: 26-04-2021

(Dr. Sumit Biswas)

BITS- Pilani, Goa Campus

INTRODUCTION

The hospital routing problem is a contextual variation of the traveling salesman problem, an NP-hard optimization problem asking the question, "What is the optimal set of routes for a fleet of vehicles to traverse to deliver to a given set of customers?". I have adopted a multi ant colony system - vehicle routing problem with time windows(MACS-VRPTW) for this problem which is based on the Ant Colony Optimization(ACO), the multiple ants, are organized in a manner to solve a multi-objective problem which is - i) the minimization of the number of tours(or vehicles) and ii) minimization of total travel time. In my approach, I have prioritized the number of tours as compared to the latter objective.



Solution Model

For the solution model, the ant builds a tour in this manner: A depot is created with all its edges going to/fro from the patients. This makes the problem close to a traveling salesman problem. An artificial ant chooses any depot at random and at each iteration checks whether the ant violates the vehicle capacity condition or the time availability condition. Current time and load are initialized to 0 an ant at position i check all of this for all neighboring nodes j. When parameters are violated, we break out of the loop and begin a new local search.

For the data, each hospital has the following columns- the x coordinate of the patients, y coordinate of the patients, the demand of each hospital, ready time of the patients, due time of the patients, and the service time required by the ambulances. I created different classes for different purposes, and each class with specific objects. Meta-heuristics and other math modeling are constructed by referring to my literature survey, such as the pheromone matrix, which stores the pheromone values of each path, heuristic information matrix, which is the inverse of the distance between two nodes. The pheromone matrix is updated using the local and global update pheromone heuristic.

Whenever a patient is picked up, it is checked whether the ambulance can return to the hospital within the service time and the distance and wait time. The distance matrix, pheromone matrix, and nearest neighbor index are recalculated. when an ant moves to the next node, the following index demand is added to the vehicle load and the time saved in reaching the patient before due time is subtracted from the total travel time of the ambulance

A path list is maintained to update the ant path whenever it reaches base cases such as reaching hospital would result in emptying the ambulance, or if the arrival time is earlier than the ready time of the patient, the ambulance would have to wait or explore the closest node for an optimized result.

The node id of an index is only added into the current path if the inserted position doesn't violate the load, time, and travel distance and if there are multiple locations, find the best one.

To implement the local search procedure, find locations of all the depots in the path, add them to a list, and divide the depot list into multiple segments. Each segment that starts with a depot and ends with a depot is called a route. Then keep finding the distance between every two starting and ending nodes (this can be done using a cross-search) and store the shortest route(route a and route b) for the particular depot and determine which route is more feasible and delete the latter this is carried out over all nodes and then finally print the travel path and the total distance.

Pseudocode

procedure MACS-VRPTW

/*initialization*/

/* main loop */

repeat

v:=active_vehicles()

Activate ACS-VEI(v-1)

Activate ACS-TIME(v)

while ACS-VEI and ACS-TIME are active:

wait for an improved solution

end while after a base case is met

ACS-VEI: minimizing the number of vehicles

procedure ACS-VEI

/* some utility functions have already been initialized to find the number of patients covered in the solution ϕ, calculate the nearest neighbour heuristic, and the distance covered */

/* cycle */

for each ant k:

construct a solution ϕ^k

new_ant()

/*update for best solution if it is improved*/

if visited_patients(ϕ^k) > visited_patients($\phi^{ACS\text{-}VEI}$)

$\phi^{ACS\text{-}VEI} := \phi^k$

/* next. send for VRPTW */

/* perform global update on both ϕ^k and $\phi^{ACS\text{-}VEI}$ */

$\tau_{ij} = \mathbf{(1-\rho)}\tau_{ij} + \rho/J^k$

$\tau_{ij} = \mathbf{(1-\rho)}\tau_{ij} + \rho/J^{ACS\text{-}VEI}$

until a stopping condition is met

Data Set

The efficiency of this algorithm is going to be tested on 10 different datasets, where each dataset represents a different hospital with 100 patients' random locations.
The hospitals I have chosen are from the Satara district and their details can be found in the dataset folder present in the GitHub repository. Each file contains- the number of ambulances the hospital has, the name of the hospital, its capacity, customer number, x coordinate, y coordinate of the customer, the demand of each depot, ready time, due time, service time. All the values of the patients are randomly generated and all the data is collected with the consent of the hospitals. The entire code is written in python using jupyter notebook.

d1 Hospital: Siddhivinayak Hospital, Lonand

VEHICLE

NUMBER	CAPACITY
1	50

CUSTOMER

CUST NO.	XCOORD.	YCOORD.	DEMAND	READY TIME	DUE DATE	SERVICE
0	25	20	30	196	1889	0
1	39	31	19	617	1941	90
2	22	26	13	27	1190	90
3	20	30	22	805	1503	90
4	40	28	13	172	1081	90
5	39	36	28	386	1247	90
6	22	22	20	945	1216	90
7	37	29	26	179	1785	90
8	20	25	24	494	1018	90
9	28	24	19	4	1236	90
10	29	32	30	491	1750	90
11	18	27	17	992	1829	90
12	20	13	16	682	1642	90
13	34	22	27	421	1598	90
14	35	10	29	469	1774	90
15	36	31	11	475	1387	90
16	28	33	28	882	1024	90
17	35	39	10	97	1104	90

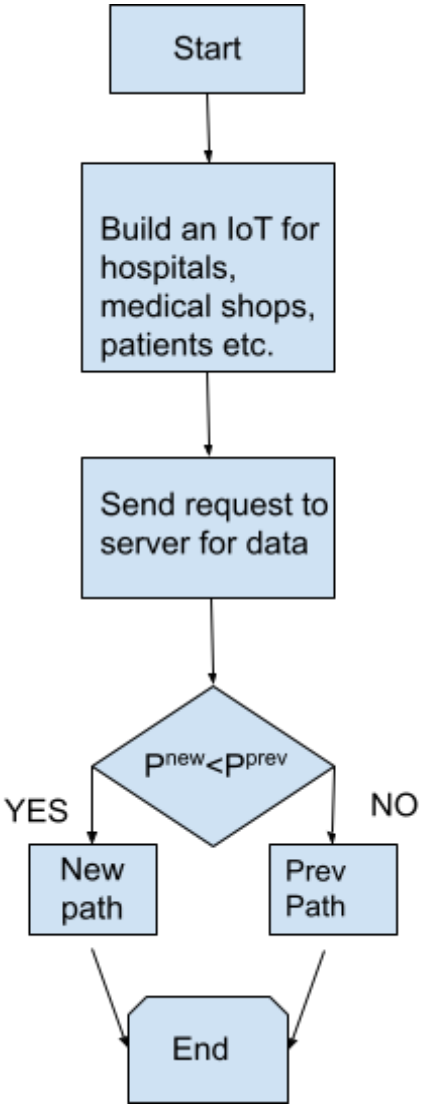
Results

I was able to implement ACS-VEI as discussed above which is the main focus of Ant colony optimization - vehicle routing problem using time windows and can be executed on the datasets in a fixed time which is the main problem of solving an NP-hard problem.

Future Scope

MACS-VRPTW introduces a new methodology to solve optimizing multi-objective problems amongst many competitors such as genetic algorithms, Dynamic Programming, and other variations of Ant Colony Optimization. Further, ACS-TIME needs to be implemented for a fully functional

model and an appropriate user interface to display the results of ant moving from node to node. In the code, I have only been able to dispatch 1 ambulance per hospital which can be improved to multiple ambulances in the entire location. Another drawback may arise if the hospitals of different locations form clusters, then the vehicles in some cases may not exist from the clusters to explore other depots due to low pheromone concentration, this may arise due to poor initialization and the system is not able to get out of a sub-optimal solution. There has been a significant role of ACO in other optimization problems, and we can utilize other engineering fields such as the Internet of Things for hospital scheduling(in this case) with respect to various parameters such as path length, energy expended, efficiency over a large area.



Annexure

My code and the datasets: Click Here

References

An ACO algorithm for a dynamic regional nurse-scheduling problem in Austria, Gutjahr, University of Vienna

Ant Colony Optimization Marco Dorigo, Thomas Stu, the MIT Press

Ambulance Routing with Ant Colony Optimization, Javidaneh et al. KNT University

Ambulance routing in disaster response scenario considering different types of ambulances and semi-soft time windows, Hamid Tikani, Mostafa Setak, KNT University

Ph.D. Thesis of Yihan Liu (BUAA)

Ant colony optimization model with characterization-based speed and multi-driver for the refilling system in the hospital. Jin et al., advances in Mechanical engineering. 2017

A comprehensive survey on the ambulance routing and location problems. Joseph Tassone, Saimur Choudary. Department of Computer Science, Lakehead University, 2020

A multi-objective ACO for operating room scheduling optimization, Wei Xiang. Springer 2017

MACS-VRPTW: A multiple Ant Colony System for vehicle routing problem with time windows Luca Maria Gambardella, Éric Taillard, and Giovanni Agazzi. McGraw-Hill, London, UK, 1999

Applying the Ant System to the Vehicle Routing Problem, in Metaheuristics: Advances and Trends in Local Search for Optimization, S.Voss, S. Martello, I.H. Osman and C. Roucairol (eds.), Kluwer Academic Publishers, Boston, 1999