

Vehicle Routing problem

Keval Savaliya

BTech - ICT

DAIICT

Gandhinagar, India

201901006@daiict.ac.in

Arik Dholiya

BTech - ICT

DAIICT

Gandhinagar, India

201901009@daiict.ac.in

Sanny Dhameliya

BTech - ICT

DAIICT

Gandhinagar, India

201901031@daiict.ac.in

Purav Kansara

BTech - ICT

DAIICT

Gandhinagar, India

201901032@daiict.ac.in

Abhi Dholiya

BTech - ICT

DAIICT

Gandhinagar, India

201901069@daiict.ac.in

Aditya Ramanuj

BTech - ICT

DAIICT

Gandhinagar, India

201901138@daiict.ac.in

I. ABSTRACT

Distribution companies face greater challenges in organizing their fleets efficiently. So they solve the problem by using VRT. The vehicle routing problem is a combinatorial optimization problem. The optimal set of routes for a group of vehicles to traverse is used to deliver to a given set of customers. The VRP has several variants depending on tasks performed and restrictions, such as time windows, multiple vehicles, backhauls, simultaneous delivery, pickup, etc. This paper establishes a model for a large-scale vehicle routing problem with simultaneous pickup and delivery, and the objective is to minimize total transportation costs. The possible preliminary answer is acquired through a miles exceptional transportation efficiency-primarily based on a set of rules and progressed via way of means of a nearby seek set of rules primarily based totally on variable community seek. The computational consequences suggest that the proposed set of regulations computes with the experimental techniques withinside the articles and improves numerous best-acknowledged solutions.

II. INTRODUCTION

Physical distribution is one of the key capabilities withinside the logistics industry. A very costly and time-consuming function, especially for the distribution industries, is called a Vehicle routing problem(VRP). Dantzig and Ramser first proposed this problem in 1959 as a generalized problem of traveling salesman problem. Since then, many loads of studies works have been posted on numerous editions of VRP. Vehicle routing problems are beneficial in lots of real-international programs like items distribution, mail delivery, schools bus routing, garbage collections, private travel operations, courier service applications, newspaper distribution, etc. Even though VRP is classed into many editions, the most essential and important variations of VRP are capacitated VRP (CVRP), VRP with time windows (VRPTW), VRP with pickup and delivery (VRPPD), and

Open VRP (OVRP). The problem taken up in this work is that of a private logistics service provider, who has numerous depots throughout the location with clients spread throughout all of the depots. CVRP with simultaneous pickup and delivery between depots with numerous nodes spread in between them is considered. In this problem, we use VRPPD (Vehicle Routing Problem with pickup and delivery).

III. AIM AND OBJECTIVE

We prefer the expression VRP with pickups and deliveries (rather than deliveries and pickup) to signify that all goods come from the depot or arrive there. There is no inter-customer transportation of goods. This paper looks at pickup-and delivery problems, where goods are taken directly from one customer to another. While that is a crucial problem, it isn't always an extension to the VRP because it lacks the multi-stop tour planning aspect. Customers who receive goods from the depot are called linehaul or deliveries; customers who send goods are backhauls or pickups. A customer may also wish to both send and receive goods. This case is called combined demands. If all clients are both natural deliveries or pure pickups, it is called single demands. Since the seminal paper of Casco, Golden, and Wasil (1988), initially, most VRPDP studies targeted three essential problem classes. The first one is The VRP with Backhauling (VRPB) is a selected case of single demands, constructed on the idea that all deliveries ought to be served before pickups can begin. We provide an explanation for the motivation for this in a few details later on.

Second is The VRP with Mixed Deliveries and Pickups (VRPMDP), on the alternative hand, does now no longer make the above assumption and permits deliveries and pickups to arise in any order on a vehicle route. The remaining third one is The VRP with Simultaneous Deliveries and Pickups (VRPSDP) is a case of combined demands in which the vehicle makes a single stop to both deliver and pick up

goods. However, this expression also denotes that customers simultaneously have delivery and pickup demand, i.e., a synonym for combined needs. It is profitable to note that the VRPMDP can be modeled as a VRPSDP by including a pickup of zero to every linehaul and delivery of zero to every backhaul. This does not make the model unduly more complicated. However, the VRPSDP can't be modeled as a VRPMDP. One can create fictitious co-placed customers (one purely linehaul one in simple terms backhaul) to transform the problem from combined demands to single demands. However, if this is solved as a VRPMDP, the requirement of a simultaneous visit might not be satisfied.

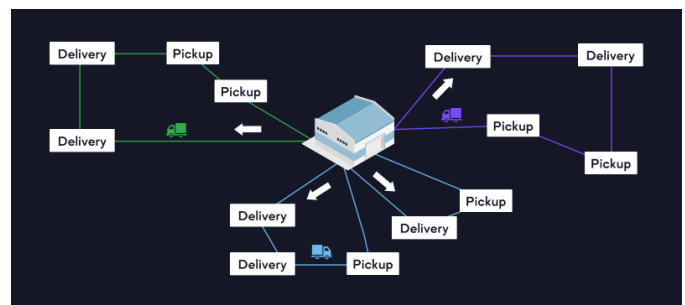
IV. PROBLEM STATEMENT

Multiple Travelling Salesman Problems combined with Together which is called The Vehicle Routing Problem. So, now we understand how default solves this problem as opposed to a different NP-complete problem. The goal is not just to find the optimal overall path but also to dynamically find the optimal approach for each vehicle and its deliveries. The purpose of the VRP is to find the optimal approach for each vehicle's route. Our main motto in this project is to solve the problem of some organization or a big transportation company having to transport goods to different locations using different vehicles every day. The problem is that the transportation is very costly and very time-consuming, So the overall efficiency of the delivery system will be reduced. So, using the Vehicle Routing Algorithm, we can achieve our efficiency and Make more Profit. Determining the route can be done by several different methods like time, priority, and demand in the VRT problem. In practice, the basic vehicle routing problem is extended with constraints, for example, on the allowed capacity of the vehicle, length of route, arrival, departure and repair time, time of assortment, and delivery of products. Several variations and specializations of the vehicle routing problem exist. So, We pick up the specific type of problem in the vast VRT problem: the pickup and delivery problem.

The pickup and delivery problem (PDP) is a problem of finding a set of optimal routes for a fleet of vehicles to serve a group of transportation requests. Each vehicle from the fleet has a given capacity, a start location, and an end location. Each transportation request is specified by a load to be transported, a source, and a destination location. In other words, the pickup and delivery problem deals with the construction of optimal routes to visit all pickup and delivery locations and satisfy precedence and pairing constraints. Precedence constraints deal with the restriction, in which each pickup location has to be visited before visiting the corresponding delivery location. Pairing constraints restrict the set of admissible routes so that one vehicle has to do both the pickup and the delivery of the load of one transportation request. A review of various approaches to the solution of VRPPD is presented in Desaulniers et al. (2000), Cordeau et

al. (2007), and Parragh et al. (2008). So Our main real-life application to solve using this algorithm is,

“We are given N different **locations** (here given coordinates of each location) and V different **vehicles** (let's suppose given Trucks). For each location , d_i is **demand** and location is (x_i, y_i) with their **pick-up point** (x_{ip}, y_{ip}) and **delivery point** (x_{id}, y_{id}) and the **capacity** of each vehicle is C . So, here we want to **Minimizing the total distance** by each of vehicle to it's pick-up and delivery locations. The real life application of it is Goods distribution .In this problem, each of vehicle departs from the vehicle depart they belong to and back the same depot after visiting several locations with a series of loading and unloading operations the objective is to minimizing the trip total cost used to satisfy all delivery requests while meeting loading constraints and time limit. ”



V. BRIEF LITERATURE REVIEW

A. CAPACITATED VEHICLE ROUTING problem WITH PICK-UP AND ALTERNATIVE DELIVERY(CVRPPAD) : MODEL

Various forms of Vehicle Routing Problems occur in the field of product distribution/shipments. Solving these problems within optimum time and costs is becoming a major issue in modern distribution systems. The vehicle routing problem aims to answer the question, "What is the optimal set of routes for a set of vehicles/transportation modes to travel to deliver it to the given set of customers at the appropriate timing." The main objective of VRP is generally to minimize the route cost. The objective functions may vary with the particular application and its variants. But the most common objectives are given below; with the used means of transport (e.g., vehicles) and the distance traveled, we have to minimize all transportation costs associated with it Cover all the given delivery points with the minimum number of transport vehicles available; To minimize the overrun in the situation when estimated travel time and vehicle have been exceeded;

B. LITERATURE REVIEW ON THE VEHICLE ROUTING problem IN THE GREEN TRANSPORTATION CONTEXT, Luna Azul no.42 Manizales Jan./June 2016

The article explains the importance of transportation in the supply chain. Since the vehicle routing problem's First method

in 1959, there have been many courses and has elevated its scope. Initial research had been carried out for analyzing distribution management. Another component that has gained interest is the inclusion of technological improvements in the VRP. About the environment, transportation is one of the most seen within the supply chain aspects. A growing interest is likewise visible by companies to reduce the environmental effect in their products and services (Quariguasi et al., 2009). VRP problems can be classified by developing a taxonomy or developing a generalized framework that summarizes the existing models, the targets pursued and the theories related to the evaluation of the problem (Reisman, 1992). The VRP in its one-of-a-kind forms was initially addressed using actual strategies together with Branch and Bound algorithm proposed through Laporte and Nobert (1987); Fischetti, Toth, and Vigo (1994) and Baldacci and Mingozzi (2004), Branch and Cut taken into consideration through Cordeau (2006), Branch and Price turned into posed by Pessoa et al., 2008; Martinelli et al., 2011, and Branch and Cut and Price proposed through Baldacci et al. (2010).

C. APPROACHES TO SOLVE THE VEHICLE ROUTING problem IN THE VALUABLES DELIVERY DOMAIN : Vladimir Korablev, Ivan Makeev, Evgeny Kharitonov, Boris Tshukin and Ilya Romanov

This PDF includes the Description of developed algorithms. The various extensions of the vehicle routing problem with time windows (VRPTW) are considered. In addition to the VRPTW, the authors present a method to solve the SDVRPTW – the version of the project allowing separate objects to be delivered to the customers. The features of this problem solved are additional route restrictions, such as the maximum time, the number of clients and cost, and the most range of vehicles required for delivery. This article is dedicated to valuable delivery problems and strategies to clear them up. The algorithms defined in this text use an evolutionary approach. First of all, the set of solutions (population) is initialized. It is represented schematically. Further, regular development takes region iteratively at the made populations. At a certain iteration, the stop condition is met.

D. A SURVEY ON VEHICLE ROUTING problem AND ITS VARIANTS

This paper consists of a literature assessment of the latest tendencies and guides regarding the vehicle routing problem and its variants, specifically vehicle routing problem with time windows (VRPTW) and the capacitated vehicle routing problem (CVRP) and additionally their variants. There are different classes or variations of VRP similar to the capacitated VRP (CVRP), VRP with Time Windows (VRPTW). In the CVRP, a fleet of identical vehicles positioned at a significant depot, needs to be optimally routed to supply a set of customers with known demands. The goal of

the VRPTW is to serve a number of clients inside predefined time windows at minimal cost (in terms of distance traveled) without violating the capacity and total trip time constraints for every vehicle. Exact algorithms to solve VRP, especially the capacitated VRP (CVRP), include the branch-and-bound, the branch-and-cut, and the branch-and-price algorithms.

VI. CURRENT STATE OF ART

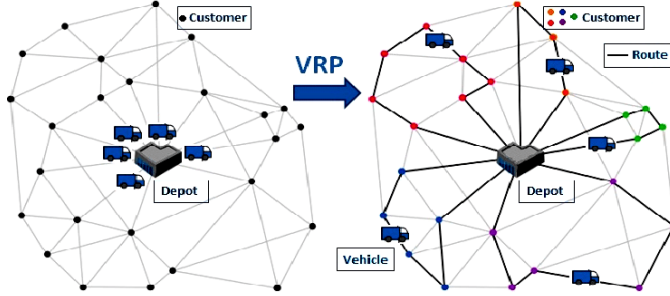
The problem of vehicle routing is very useful for transportation, but it is very big and complicated. This problem has many several variations, which are divided using different constraints like capacity of the vehicle, length of route, arrival, departure and repair time, time of assortment, and delivery of products. But here we will consider only the problem of Pickup and Deliveries of goods using vehicles and try to minimize the total cost of the trip. In the VRP with Pickup and Delivery (VRPPD), goods must be picked up from a certain location and dropped off at their destination. The pickup and drop-off must be done by the same vehicle, which is why the pickup location and drop-off location must be included in the same route. Here one of the advanced versions of pickup and delivery. The related problem is the VRP with backhauls where a vehicle does delivery and pickups in one route. Some customers require deliveries and others pickups. We can also add one more constraint in the VRPPD problem with the time window, which is denoted minimum time and minimum cost to get all the tours completed. Recently, some of these variants have been combined into so-called 'Rich' vehicle routing problems, simultaneously including multiple real-life aspects. Let's have some discussion about our current problem and also mathematically define our problem.



A. problem Description

The problem could be started as follows, there is a set of homogeneous vehicles $V=1,2,...,|V|$. Each vehicle pick-up and delivery of different kinds of goods to different locations and after they are done working, they go back to the same depot position. Here is given the set of data of N different locations using their (x_i, y_i) corresponding coordinates. For each of the locations we also gave the demand of goods which denoted using d_i . Also here given the capacity of each of the trucks is C . Using this information we will be able

to find the different sequences of delivery for V_i vehicle which is denoted by T_i (which is denoted Tour sequences, i -th vehicle travel different location in they current tour.) and try to minimize the trip total cost used to satisfy all delivery requests. hear the total cost which means the distance traveled by vehicle is minimized.



B. Equations

- 1) $|T_i|$ = Length of the current traveled locations sequence of i the vehicle
- 2) V_i = denoted i th vehicle
- 3) D_i = demand of i th location
- 4) x_i, y_i = coordinate of pick-up or delivery point
- 5) C = capacity of the vehicles

$$Z = \sum_{i \in V} ((dist(0, T(i, 0))) + \sum_{(j,k) \in T_i} dist(j, k) +$$

$$dist(T(i, |T_i| - 1), 0))$$

Subject to ,

$$\sum_{j \in T_i} d_j \leq C (i \in V)$$

Above equation is denoted that the in sequences of i the vehicle sum of all the demand in sequences will less then the capacity of vehicle.

$$\sum_{i \in V} (j \in T_i) = 1 (j \in N / \{0\})$$

This equation denoted that all the locations are visited using the given vehicles.

VII. PLAN OF WORK

A. Goal:1

1) **Intentions:** Finding good quality and more quantity research paper or reference material. Formulating clear abstract and problem statements

2) **Timeline:** 16th Sept - 22nd Sept 2021.

3) **Work progress:** First of all we decide to find a relevant research paper or any article on our topic. So we all find one or two articles or papers on it. Then we discuss all of our problems . From this paper and articles, we made our problem statement.

B. Goal:2

1) **Intentions:** Understand the theory involved in problem solving about VRP(vehicle routing problem) and decide the subtopic which this project is about.

2) **Timeline:** 3rd oct - 11th oct 2021.

3) **Work progress:** For better understanding of our problem statement and theory behind it, we watched a YouTube video. Now, we know about our problem very well. In this problem we have many variants of VRT. Then we decided to go on with VRPPD(vehicle routing problem with pick-up and delivery) nicely. Also we put Video link and research paper or reference link in our reference part.

C. Goal:3

1) **Intentions:** Formulating equations and research about algorithm behind this problem

2) **Timeline:** 14th oct - 18th oct 2021, 22nd oct - 25th oct 2021.

3) **Work progress:** we discussed our views and solved the query we understood from the theory. Also we built equations for our problem and also made changes from previous algo and built new algo and implemented it very well and easily worked for the problem. For this , we refer research paper which mentioned in our reference section

D. Goal:4

1) **Intentions:** To implementing the final version of our solution in Python (Python compiler)

2) **Timeline:** 6th Nov to 13th Nov 2021.

3) **Strategy:** In this we will try to understand the code(problem) from online resources and we will try to implement the code.

E. Goal:5

1) **Intentions:** Describe some application of our project in real life problems.

2) **Timeline:** 16th Nov to 20th Nov 2021

3) **Strategy:** At the end of this project our final aim is to use this into real life applications such as delivery or Transportation systems.

VIII. CURRENT PROGRESS

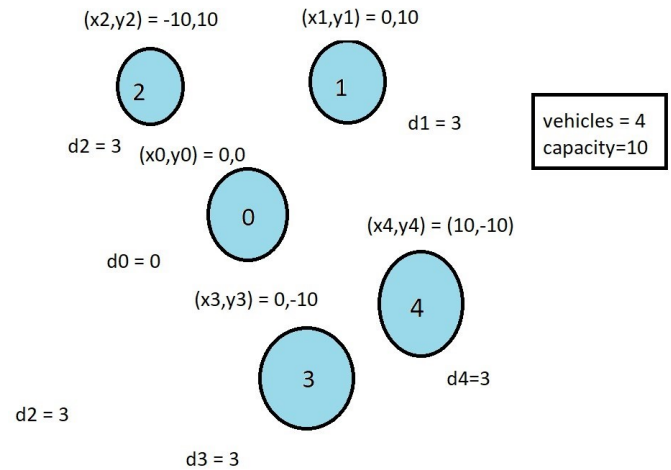
First, we went through multiple different articles and problem papers, then we found our problem statement, which is the Vehicle Routing Problem, of which all the research papers were read by each of our group members. After discussing the problem among us, We found the theory and various algorithms as Vehicle Routing Problem consists of sub-problems. As we understood the importance of Linear programming in our Vehicle Routing Problem, we decided to implement it in python.

IX. TECHNOLOGY AND LITERATURE REVIEW

Modern vehicle routing tools use machine learning-based solutions, which are used to increase efficiency and also help reduce fuel costs. It is also used to boost consumer satisfaction to ensure profitability per order. This VRPPD is a very complex problem, so we can't solve it without the use of any computerized technology. VRPPD is easily solved using machine-learning principles and tools that thoroughly analyze collected historical data. Each delivery made by a vehicle routing software run on machine-learning-based route planning can help companies save money and time.

X. SYSTEM DESCRIPTION

The vehicle routing problem is very complicated, and also it has many variants, so we can't solve it using hand analysis. In our project, we take one of the most difficult and complex variants of VRP, which is pickup and delivery. Let's take a simple formulation for a better understanding problem of VRPPD. Here for simplicity lets, we pick up at the central position and delivery for any other point. Here (x_0, y_0) is denoted its central position, and all other locations are denoted using (x_i, y_i) . For every position, some demand is denoted using d_i . Also, we have V vehicles that have the same capacity C . For every tour, we start with a central position. we take four vehicles, and each of the vehicles has a capacity of 10

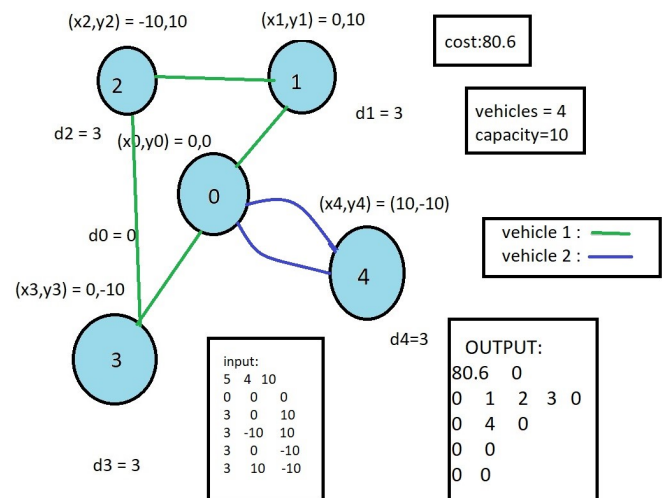


Let's solve the vehicle routing problem with simultaneous pick-up and delivery. Here we give some additional inputs to the computerized VRPPD algorithms then we get additional output.

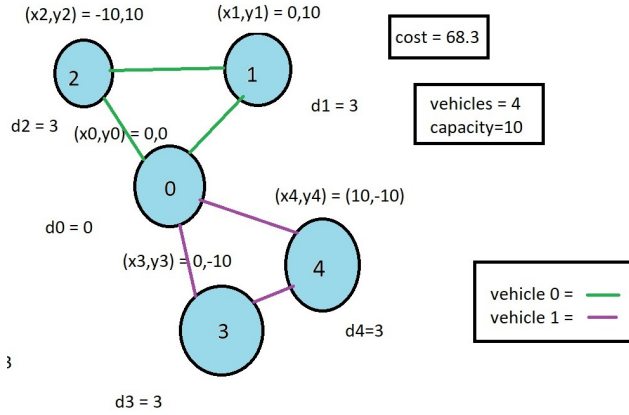
INPUT				OUTPUT			
N	V	C		obj	opt		
d_0	x_0	y_0		0	t_0_1	t_0_20
d_1	x_1	y_1		0	t_1_1	t_1_20
.....						
.....						
d_ N -1	x_ N -1	y_ N -1		0	t_ V -1_1	t_ V -1_2 0

In our first tour The minimum cost is 80.6. Here vehicle 1 is denoted using green lines and vehicle 2 path is denoted with purple color. Here the main constrain is that every demand should be less than the total capacity of the vehicle and also demand should satisfy one's. we have given the sequence of deliveries of vehicle i is denoted using T_i

Path of vehicle 1 (T_1): $0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 0$
 Path of vehicle 2 (T_2): $0 \rightarrow 4 \rightarrow 0$



In the second interaction we minimize our cost and it is better than the previous one. here we get minimum cost is 68.3 which is optimizing value compare to previous one.



Path of vehicle 0 (T0) : 0 → 1 → 2 → 0

Path of vehicle 1 (T1) : 0 → 4 → 3 → 0

XI. CONCLUSION

In this paper, we have considered the Vehicle Routing Problem for pickup and delivery problem (PDP). For solving the problem, we generated the LPP for the graph. Then by using the Routing Index Manager, we solved the problem. The advantage of this method is that modern vehicle routing tools use machine learning-based solutions, which are used to increase efficiency and reduce fuel costs. We also developed code for solving these types of problems in PyCharm. The problem for finding the complexity is an NP-complete problem.

Our computations show that this problem can also be converted into a Distribution Problem. This problem has many several variations, which are divided using different constraints like capacity of the vehicle, length of route and repair time, delivery of products, and time of assortment. But here we will consider only the problem of Pickup and Deliveries of goods using vehicles and try to minimize the total cost of transport. Many modifications can be made in our problem to solve several problems in real life.

XII. ACKNOWLEDGMENT

We are thankful to Prof. Manish Kumar for providing us the opportunity to research on such topics and have the experience of applying what we study to real-life scenarios. Also thankful to TAs for guiding us throughout the semester to carry out our research in the correct direction.

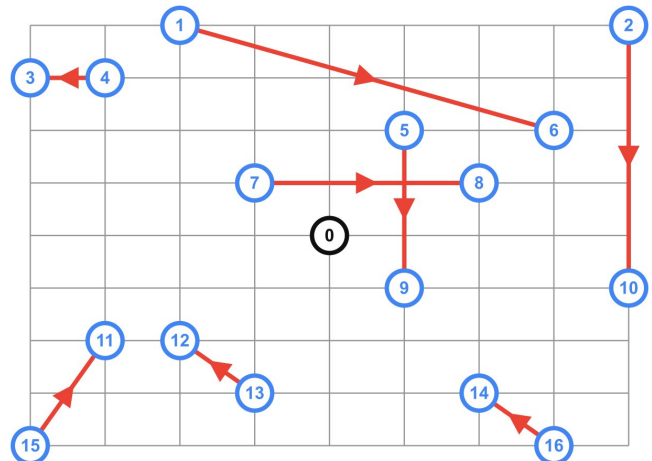
REFERENCES

- [1] <https://www.researchgate.net/profile/Ashima-Gupta-5/publication/327192553/figure/fig1/AS:703224415805440@154467317432/Vehicle-Routing-problem.png>

- [2] F. A. Tang Montané and R. D. Galvão, "A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service," *Computers Operations Research*, vol. 33, no. 3, pp. 595–619, 2006
- [3] https://www.researchgate.net/figure/Classical-Vehicle-Routing-problem_fig13/27192553
- [4] K. Fleszar, I. H. Osman, and K. S. Hindi, "A variable neighbourhood search algorithm for the open vehicle routing problem," *European Journal of Operational Research*, vol. 195, no. 3, pp. 803–809, 2009.
- [5] H. Hernández-Pérez and J.-J. Salazar-González, "A branch-and-cut algorithm for a traveling salesman problem with pick-up and delivery," *Discrete Applied Mathematics*, vol. 145, no. 1, pp. 126–139, 2004
- [6] http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S1909-24742016000100021f1
- [7] S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.
- [8] <https://link.springer.com/article/10.1007/s10479-017-2722-xSec3>
- [9] Beck, J.C.; Prosser, P.; Selensky, E. (2003). "Vehicle routing and job shop scheduling: What's the difference?" (PDF). *Proceedings of the 13th International Conference on Artificial Intelligence Planning and Scheduling*
- [10] <https://neo.lcc.uma.es/vrp/vrp-instances/>
- [11] <https://core.ac.uk/download/19477982.pdf>
- [12] https://en.wikipedia.org/wiki/Vehicle_routing_problem
- [13] G. Mosheiov, "Vehicle routing with pick-up and delivery: tour-partitioning heuristics," *Computers Industrial Engineering*, vol. 34, no. 3, pp. 669–684, 1998.
- [14] <http://webpages.ull.es/users/hhperez/PDsite/index.html>

XIII. APPENDIX

A. Input



```

"""Simple Pickup Delivery Problem (PDP)."""

from ortools.constraint_solver import routing_enums_pb2
from ortools.constraint_solver import pywrapcp

def create_data_model():
    """Stores the data for the problem."""
    data = {}
    data['distance_matrix'] = [
        [
            0, 548, 776, 696, 582, 274, 502, 194, 308, 194, 536, 502, 388, 354,
            468, 776, 662
        ],
        [
            548, 0, 684, 388, 194, 502, 738, 354, 696, 742, 1084, 594, 480, 674,
            1016, 868, 1210
        ],
        [
            776, 684, 0, 992, 878, 502, 274, 810, 468, 742, 480, 1278, 1164,
            1130, 788, 1552, 754
        ],
        [
            696, 388, 992, 0, 114, 650, 878, 502, 844, 890, 1232, 514, 628, 822,
            1164, 560, 1358
        ],
        [
            582, 194, 878, 114, 0, 536, 764, 388, 730, 776, 1118, 400, 514, 708,
            1050, 674, 1244
        ],
        [
            274, 502, 502, 650, 536, 0, 228, 308, 194, 240, 582, 776, 662, 628,
            514, 1050, 708
        ],
        [
            502, 738, 274, 878, 764, 228, 0, 536, 194, 468, 354, 1004, 890, 856,
            514, 1278, 480
        ],
        [
            194, 354, 810, 502, 388, 308, 536, 0, 342, 388, 730, 468, 354, 320,
            662, 742, 856
        ],
        [
            308, 696, 468, 844, 730, 194, 194, 342, 0, 274, 388, 810, 696, 662,
            320, 1084, 514
        ],
        [
            194, 742, 742, 890, 776, 240, 468, 388, 274, 0, 342, 536, 422, 388,
            274, 810, 468
        ],
        [
            536, 1084, 400, 1232, 1118, 582, 354, 730, 388, 342, 0, 878, 764,
            730, 388, 1152, 354
        ],
        [
            502, 594, 1278, 514, 400, 776, 1004, 468, 810, 536, 878, 0, 114,
            308, 650, 274, 844
        ],
        [
            388, 480, 1164, 628, 514, 662, 890, 354, 696, 422, 764, 114, 0, 194,
            536, 388, 730
        ],
    ],

```

B. Main code

```

def main():
    """Entry point of the program."""
    # Instantiate the data problem.
    data = create_data_model()

    # Create the routing index manager.
    manager = pywrapcp.RoutingIndexManager(len(data['distance_matrix']),
                                           data['num_vehicles'], data['depot'])

    # Create Routing Model.
    routing = pywrapcp.RoutingModel(manager)

    # Define cost of each arc.
    def distance_callback(from_index, to_index):
        """Returns the manhattan distance between the two nodes."""
        # Convert from routing variable Index to distance matrix NodeIndex.
        from_node = manager.IndexToNode(from_index)
        to_node = manager.IndexToNode(to_index)
        return data['distance_matrix'][from_node][to_node]

    transit_callback_index = routing.RegisterTransitCallback(distance_callback)
    routing.SetArcCostEvaluatorOfAllVehicles(transit_callback_index)

    # Add Distance constraint.
    dimension_name = 'Distance'
    routing.AddDimension(
        transit_callback_index,
        0, # no slack
        3000, # vehicle maximum travel distance
        True, # start cumul to zero
        dimension_name)

```

```

    ],
    [
        274, 502, 502, 650, 536, 0, 228, 308, 194, 240, 582, 776, 662, 628,
        514, 1050, 708
    ],
    [
        502, 738, 274, 878, 764, 228, 0, 536, 194, 468, 354, 1004, 890, 856,
        514, 1278, 480
    ],
    [
        194, 354, 810, 502, 388, 308, 536, 0, 342, 388, 730, 468, 354, 320,
        662, 742, 856
    ],
    [
        308, 696, 468, 844, 730, 194, 194, 342, 0, 274, 388, 810, 696, 662,
        320, 1084, 514
    ],
    [
        194, 742, 742, 890, 776, 240, 468, 388, 274, 0, 342, 536, 422, 388,
        274, 810, 468
    ],
    [
        536, 1084, 400, 1232, 1118, 582, 354, 730, 388, 342, 0, 878, 764,
        730, 388, 1152, 354
    ],
    [
        502, 594, 1278, 514, 400, 776, 1004, 468, 810, 536, 878, 0, 114,
        308, 650, 274, 844
    ],
    [
        388, 480, 1164, 628, 514, 662, 890, 354, 696, 422, 764, 114, 0, 194,
        536, 388, 730
    ],
],

```

```

    dimension_name)
    distance_dimension = routing.GetDimensionOrDie(dimension_name)
    distance_dimension.SetGlobalSpanCostCoefficient(100)

    # Define Transportation Requests.
    for request in data['pickups_deliveries']:
        pickup_index = manager.NodeToIndex(request[0])
        delivery_index = manager.NodeToIndex(request[1])
        routing.AddPickupAndDelivery(pickup_index, delivery_index)
        routing.solver().Add(
            routing.VehicleVar(pickup_index) == routing.VehicleVar(
                delivery_index))
        routing.solver().Add(
            distance_dimension.CumulVar(pickup_index) <=
            distance_dimension.CumulVar(delivery_index))

    # Setting first solution heuristic.
    search_parameters = pywrapcp.DefaultRoutingSearchParameters()
    search_parameters.first_solution_strategy = (
        routing_enums_pb2.FirstSolutionStrategy.PARALLEL_CHEAPEST_INSERTION)

    # Solve the problem.
    solution = routing.SolveWithParameters(search_parameters)

    # Print solution on console.
    if solution:
        print_solution(data, manager, routing, solution)

if __name__ == '__main__':
    main()

```

```

    ],
    data['pickups_deliveries'] = [
        [1, 6],
        [2, 10],
        [4, 3],
        [5, 9],
        [7, 8],
        [15, 11],
        [13, 12],
        [16, 14],
    ]
    data['num_vehicles'] = 4
    data['depot'] = 0
    return data

def print_solution(data, manager, routing, solution):
    """Prints solution on console."""
    print(f'Objective: {solution.ObjectiveValue()}')
    total_distance = 0
    for vehicle_id in range(data['num_vehicles']):
        index = routing.Start(vehicle_id)
        plan_output = 'Route for vehicle {}: \n'.format(vehicle_id)
        route_distance = 0
        while not routing.IsEnd(index):
            plan_output += ' {} -> '.format(manager.IndexToNode(index))
            previous_index = index
            index = solution.Value(routing.NextVar(index))
            route_distance += routing.GetArcCostForVehicle(
                previous_index, index, vehicle_id)
            plan_output += '({}) \n'.format(manager.IndexToNode(index))
        plan_output += 'Distance of the route: {} \n'.format(route_distance)
        print(plan_output)
        total_distance += route_distance

```

C. Output

```

Objective: 226116
Route for vehicle 0:
0 -> 13 -> 15 -> 11 -> 12 -> 0
Distance of the route: 1552m

```

```

Route for vehicle 1:
0 -> 5 -> 2 -> 10 -> 16 -> 14 -> 9 -> 0
Distance of the route: 2192m

```

```

Route for vehicle 2:
0 -> 4 -> 3 -> 0
Distance of the route: 1392m

```

```

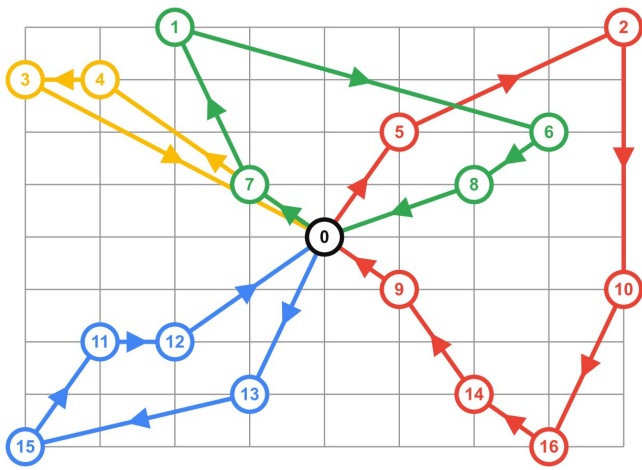
Route for vehicle 3:
0 -> 7 -> 1 -> 6 -> 8 -> 0
Distance of the route: 1780m

```

```

Total Distance of all routes: 6916m

```



D. solution Code link

<https://bit.ly/3r7MkHBCode>