Heuristic Optimization Methods
Project instructions 2022/23

Capacitated Vehicle Routing Problem with Time Windows

Problem description

These instructions describe the problem of capacitated vehicle routing with time windows (CVRPTW). CVRPTW belongs to the class of Vehicle Routing Problems (VRP), but, on top of the generic VRP formulation, defines additional constraints that occur in real-life scenarios. In VRP, the objective is to find a set of routes for a fleet of vehicles serving customers from a depot. Figure 1 illustrates the VRP with an example problem input and one of the possible outputs.

The *capacity constraint* in CVRP refers to the capacity of the vehicle: the total demand of all customers supplied on a single route must not exceed the vehicle capacity. On the other hand, the *time window* constraint in VRPTW refers to the interval in which a customer must be supplied (often called the "scheduling horizon"): all customers in a route need to be reached within their scheduling horizons, and the whole route needs

[*]Figure taken from `http://neo.lcc.uma.es/vrp/vehicle-routing-problem`.

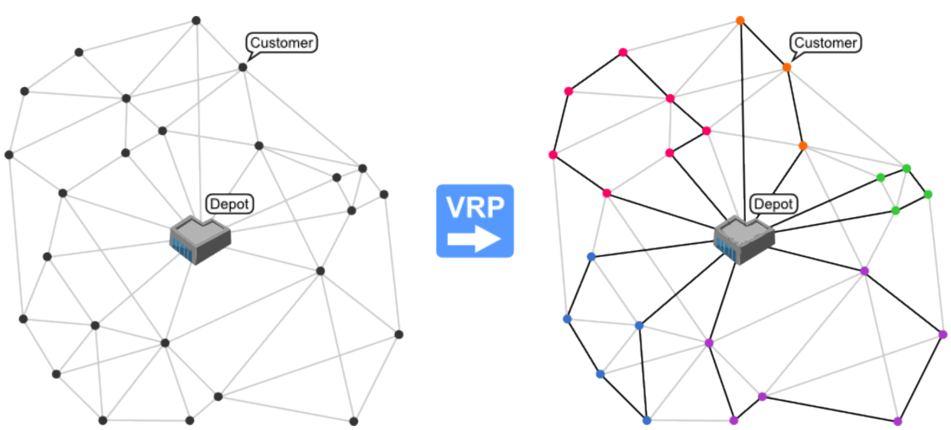

Figure 1: An instance of a VRP (left) and its solution (right)[*]

to be started and finished within the working hours of the depot. CVRPTW includes both constraints.

The following text describes the format of problem instances (input files), formulates the objectives and constraints, and describes your project task related to this problem.

Problem instance

Problem instances define the number of vehicles, single vehicle capacity, and data about each customer: X and Y coordinates, resource demands, schedule horizon, and the duration of the service time. The data provided in each instance file is defined as follows:

- VEHICLE -	
NUMBER	Total number of available vehicles.
CAPACITY	Capacity of a single vehicle. All vehicles have the same capacity. Capacity is "spent" on fulfilling customer demands on a route serviced by that vehicle.
- CUSTOMER -	
CUST NO.	Customer index. Index 0 corresponds to the depot, not a customer.
XCOORD.	X coordinate of the customer location.
YCOORD.	Y coordinate of the customer location.
DEMAND	Amount of resources that need to be delivered to a given customer by a vehicle on a route that services that customer. Demands "spend" the capacity of the vehicle on that route. For customer no. 0 (depot) demand equals 0.
READY TIME	The earliest time at which the start of the service can happen for a certain customer. For the depot (index 0) indicates the opening of the depot.
DUE DATE	The latest time at which the start of the service can happen for a certain customer. For the depot (index 0) indicates the closing of the depot.
SERVICE TIME	The duration of the service – the amount of time which the service vehicle needs to spend at the customer location. Note: Service only needs to start between READY TIME and DUE DATE, it can finish after the DUE DATE.

```
VEHICLE
NUMBER     CAPACITY
  50          200

CUSTOMER
CUST NO. XCOORD.   YCOORD.   DEMAND  READY TIME  DUE DATE  SERVICE TIME

    0       70        70         0          0       1351            0
    1       33        78        20        750        809           90
    2       59        52        20          0       1240           90
    3       10       137        30          0       1172           90
    4        4        28        10          0       1183           90
    5       25        26        20        128        179           90
    6       86        37        10        478        531           90
    7        1       109        10          0       1182           90
    8        6       135        40        351        386           90
    9       32        79        20        655        721           90
   10       24        26        20          0       1198           90
   ...
```

Problem formulation

The problem defines the following **constrains**:

1. Each customer is served by exactly one vehicle/route, with the resource amounts that equal their demands.
2. The demand on each route must not exceed the capacity of the vehicle.
3. The vehicle servicing a certain customer must arrive at the customer location within the interval given for that customer. The duration of the service can exceed the interval.
4. Each vehicle starts and finishes its route in node 0 (customer 0 location; depot), within the time interval given for customer 0.

The time at which a vehicle starts the service at the location of a certain customer in a route equals the sum of: (1) the time at which previous customer was serviced (or the time at which the depot was left, for the first customer), (2) SERVICE TIME at previous location, and (3) the ceiling of the distance between previous and current location. The distance is calculated as Euclidean distance. If that value is smaller than

READY TIME for the current customer, then the service of the current customer starts at READY TIME.

The primary **objective** is to minimize the number of vehicles by which all the customers can be serviced, while the secondary objective is to minimize the sum of distances on all routes. In this formulation, one vehicle can only service one route – each route in the solution file assumes a new vehicle being used (i.e., a vehicle cannot return to depot and continue on another route). The distance between two customers is calculated as the Euclidean distance. For example: $d(c_0, c_1) = \sqrt{(33-70)^2 + (78-70)^2}$. The solutions that use a smaller number of vehicles **are considered to be better** than solutions with a smaller total distance. Distance is considered in solution comparison only in solutions using the same number of vehicles.

Project task

Students may work in teams (max. 2 persons) or individually.

1. Design and implement a heuristic algorithm to solve the given problem.
2. Execute your algorithm for given instances of the problem.
3. Save **3 solutions for each instance**: solution obtained after 1 minute of algorithm execution, 5 minutes of algorithm execution and without time constraints.
4. For each solution, save the value of both objective functions (number of vehicles and distance traveled), and the number of iterations in which you evaluated the objective function until obtaining the saved solution.
5. Create a report that describes your implemented heuristic algorithm. The report should include the following:
    - A description of the problem.
    - A description of the implemented heuristic algorithm (solution representation, objective/fitness function, way of construction of an initial solution, iteration size and termination criterion, heuristic specific design elements, etc.).
    - Pseudocode of the implemented algorithm.
    - Analysis of results and discussion (e.g., influence of certain parameters of the heuristic algorithm on the quality of solutions, execution time of the algorithm)
    - Conclusion (e.g., discussion of potential further improvements of the implemented algorithm).

6. **Due date and project submission**: the project is **due on January 10, 2022 at 12:00 noon**. Submission involves uploading a ZIP archive to Moodle ("Project 2021/2022").

   The archive should contain:

   (a) the report (including what is listed in bullets on page 4 of the instructions),

   (b) solution files, named and formatted as is given in the instructions,

   (c) source code.

Files with obtained solutions must:

- be named "res-time-instance.txt", where "time" equals "1m", "5m" or "un" (unlimited), and "instance" equals "i1", "i2", etc.
- contain information about the routes and servicing schedule, in the following format:

```
21
1: 0(0)->53(85)->58(180)->2(200)->0(228)
2: 0(0)->27(27)->28(44)->12(81)->80(168)->77(185)->50(203)->0(230)
...
19: 0(0)->44(59)->38(80)->0(133)
20: 0(0)->39(34)->23(58)->67(80)->0(134)
21: 0(0)->65(50)->0(110)
1836,87
```

  The first row in the solution file indicates the number of routes/vehicles. Following rows define the routes. Each cycle is defined by ordered customer indices (and the time of the service start). The last row indicates the total distance of all routes.