Sabancı University Industrial Engineering
Optimization Challenge 2024
Problem Description

Delivering goods efficiently is a pivotal challenge in urban logistics, particularly within the rapidly growing domain of online grocery shopping. FreshCart is a popular online grocery delivery service that prides itself on delivering fresh produce and pantry staples right to its customers' doorsteps. With a fleet of delivery vehicles and a network of suppliers, FreshCart has revolutionized the way people shop for groceries. However, ensuring timely deliveries in a city teeming with traffic and diverse delivery locations is not an easy task.

Every morning, FreshCart's operations team faces the daunting task of planning efficient delivery routes for their fleet of delivery vehicles. Each vehicle has a limited capacity and must adhere to strict time windows for each delivery. For example, perishable items like dairy and meat must be delivered within a specific timeframe to maintain freshness. Moreover, customers can state preferences on when they would like to receive their orders. The team analyzes orders received overnight, taking into account delivery addresses, time windows, and product availability. The information gathered by this analysis serves as an input when planning the vehicle routes.

Suppose that you are a member of FreshCart's operations team and that you are responsible for determining the delivery routes having all the necessary information about the customer orders to be fulfilled on a given day. To achieve this, you need to solve a well-known combinatorial optimization problem, namely, the vehicle routing problem with time windows (VRPTW).

The vehicle routing problem (VRP) seeks to identify a least cost set of routes for a fleet of vehicles to satisfy the given demands of a set of geographically dispersed customers. All vehicles must start and end their routes at a central depot, and the total demand assigned to any vehicle's route cannot exceed its capacity. Every customer must be visited and served by exactly one vehicle. In the classical VRP, the vehicles are assumed to be identical (they have equal capacity) and the customers can be served in any order. In the VRPTW, each customer has an associated time window within which service should take place. A time window is represented by a closed time interval and customers cannot be served outside this time interval. Note that this implies that if a vehicle arrives at a customer location earlier than the opening of the customer's time window, it has to wait until the opening of the time window. The VRPTW can be simply defined as a variant of the VRP with the additional constraint that each customer must be served within his/her time window.

For the first round of this optimization challenge, you are expected to come up with a solution approach for the VRPTW, implement it using a programming language of your choice, and use it to solve a set of benchmark instances that will be provided to you.

What you need to submit via email (opt-challenge@sabanciuniv.edu) by April 19:

1. A short report presenting
- your solution approach
- relevant implementation details including the choice of programming language, solver or any additional software (if applicable), algorithm-specific parameter choices, where you ran the experiments etc.

2. Your code file(s).
3. Your solution files in a zip folder (see the instance folder for a sample solution file, you must have a solution file in the same format for every instance you solve).
4. Proof of student certificate for each group member (recall that only undergraduate students are eligible).

Note that you need to register in order to access the problem instances.

Evaluations will be based on the quality of the solutions submitted. The quality of a solution is measured by its cost; solutions having lower cost values are considered to be of higher quality. For different solutions of the same cost, the one involving the fewest number of routes will be favored over the others. The cost of a solution is defined by two components: (1) the total distance/travel time of the routes and (2) the fixed cost per each vehicle used. You need to calculate the Euclidean distances (you may use one decimal point and truncation) between all pairs of locations for each problem instance, and take them as your distance/travel time values. For a problem instance with n customers, set the fixed cost of using a vehicle to be $2 * n * c_{max}$ where c_{max} is the maximum of the distance values connecting any pair of locations.

Further Clarifications:

- The travel time along an arc is equal to the distance of that arc.
- The time window of a customer represents the earliest and latest times to begin (not to complete) service. Thus, service times may be longer than the length of the time window.
- Since the due date (the upper bound of the time window) represents the latest time to start service at that node, it is possible to leave a node after its time window closes.
- All arcs should be considered while determining the value of c_{max}.
- It is not allowed to use a vehicle for more than one route.