

Discrete Optimization and Decision Making Project

Last-mile Delivery Problem

Patrick Hamzaj
VR474246

Federico Leonardi
VR

1. Introduction

This report presents the solution to the project assignment for the course Discrete Optimization and Decision Making. In particular, the central objective of this project is to devise a solution that efficiently assigns a fleet of delivery vans to serve a known set of customers, each requiring delivery of packages with known weights and scheduled delivery times. Vehicles depart from a common depot, deliver packages while respecting their maximum carrying capacity, and must complete all deliveries within a specified maximum time.

To model this scenario, we consider a mathematical optimization problem structured around a complete directed graph, representing customers and the depot as nodes and the feasible paths between them as arcs. Each arc has an associated travel time that satisfies the triangle inequality. The main challenge involves determining routes for each vehicle that minimize the total completion time of deliveries, adhering to constraints such as vehicle capacity, customer-specific delivery times, and operational limits.

This report presents the mathematical formulation of the described optimization problem, completing in particular the baseline problem (Module 1) and two extensions—namely Module 2, which requires to restrain deliveries within a delivery time window along with a newly introduced set of triplets of incompatible customers served by a single vehicle and Module 4, that introduces an additional objective function focused on fairness and equity among drivers.

2. Problem Description

We consider a set of customers $C = \{1, \dots, \bar{c}\}$ and the depot (node 0). Each customer requires delivery of a package with weight $w = \{w_1, w_2, \dots, w_c\}$ and a delivery time $s = \{s_1, s_2, \dots, s_c\}$. Deliveries are performed by a fleet $K = \{1, \dots, \bar{k}\}$ of vehicles each with maximum capacity W , which must start and end at the depot, returning within a maximum allowed time t_{\max} .

The delivery network is modeled as a complete directed graph $G = (V, A)$, with:

- Nodes $V = C \cup \{0\}$ representing customers and depot.
- Arcs $A = \{(i, j) \mid i, j \in V, i \neq j\}$ indicating direct travel possibilities with associated travel times $t_{i,j}$.

The company's goal is to minimize the total time needed to complete all deliveries, defined as the latest return time among all vehicles.

3. Mathematical Formulation

3. 1. Decision Variables:

1. Deciding if an arch is chosen or not.

$$x_{i,j,k} = \begin{cases} 1 & \text{if vehicle } k \text{ traverses the arch } (i,j) \in A \\ 0 & \text{otherwise} \end{cases}$$

with $i \neq j \in V$ and $k \in K$.

$x_{i,k,k}$ is a binary variable that equals 1 if vehicle k travels from customer i to customer j (with $i \neq j$); it takes 0 if it does not.

2. Deciding the starting time of the service.

$$y_{i,k} \geq 0, \forall i \in V, k \in K$$

$y_{i,k}$ is a continuous variable representing the time at which vehicle k starts the service at customer i . This variable serves as a labeling system in order to avoid subtours.

3. Deciding the overall route duration.

$$d_{j,k} \geq 0, \forall j \in V, k \in K$$

It is a continuous variable that tracks the total duration for vehicle k when departing from node j toward the depot.

3. 2. Objective function

In this problem, the task is to optimize the overall service time. This can be done by minimizing the sum over the travel times of the routes travelled by the vehicles.

$$\min \sum_{k \in K} \sum_{(i,j) \in A} t_{i,j} x_{i,j,k}$$

where:

- $t_{i,j}$ is the travel time over the arc (i,j)
- $x_{i,j,k} \in \{0,1\}$ indicates if vehicle k uses $(i \rightarrow j)$

3. 3. Constraints

1. Each customer is served exactly once.

$$\sum_{\substack{j \in V \\ j \neq i}} \sum_{k \in K} x_{j,i,k} = 1, \forall i \in C.$$

This formulation ensures that every customer i in the set of customers C is visited exactly once by summing over all vehicles $k \in K$ and all potential preceding nodes $j \in V$ (excluding i itself).

2. Every vehicle must depart.

$$\sum_{j \in C} x_{0,j,k} = 1, \forall k \in K.$$

It guarantees that each vehicle $k \in K$ departs from the depot (node 0) exactly once by defining the sum of flows from the depot to all customers $j \in C$ is 1.

3. Flow conservation: incoming arch=outgoing arch.

$$\sum_{\substack{j \in V \\ j \neq i}} x_{j,i,k} = \sum_{\substack{j \in V \\ j \neq i}} x_{i,j,k}, \forall i \in C, \forall k \in K.$$

For every customer i (excluding the depot), the number of arcs entering i equals the number of arcs leaving. In other words, if a node is visited, it must have both an incoming and an outgoing arc, thereby maintaining the consistency of the vehicle route.

4. Depot flow: each vehicle starts and ends at depot.

$$\sum_{j \in C} x_{0,j,k} = \sum_{j \in C} x_{j,0,k}, \forall k \in K.$$

This constraint enforces that for every $k \in K$, the total flow from the depot (node 0) to all customers $j \in C$ equals the total flow from all customers $j \in C$ back to the depot, ensuring a balanced departure and return for each vehicle.

5. Vehicle capacity constraint.

$$\sum_{i \in C} w_i \left(\sum_{\substack{j \in V \\ j \neq i}} x_{j,i,k} \right) \leq W, \forall k \in K.$$

With this constraint, we ensure that for each vehicle $k \in K$, the sum of the weights w_i of the customers $i \in C$ it serves does not exceed the vehicle's capacity. The inner summation determines whether customer i is visited by vehicle k —with a value of 1 if visited, 0 otherwise—therefore including w_i only if the customer is served by that vehicle.

6. Time constraint with progression of the vehicles to avoid subtours

$$y_{i,k} \geq y_{i,j} + t_{i,j} + s_i - M(1 - x_{i,j,k}), \forall i, j \in V \text{ with } i \neq j \text{ and } j \neq 0.$$

This constraint avoids subtours $i \rightarrow j \rightarrow i$ by forcing a time progression when vehicle k travels from customer i to customer j (with j not being the depot). It ensures that if arch (i, j) is traversed (so $x_{i,j,k} = 1$) the start time at the next customer $y_{j,k}$ must be at least the sum of the start time at the current customer $y_{i,k}$ plus the travel time from i to j and the service time at i . The Big-M notation is used to deactivate the constraint when $x_{i,j,k} = 0$.

7. Route duration constraint.

$$d_{j,k} \geq y_{j,k} + s_j + t_{j,0} - M(1 - x_{j,0,k}), \forall k \in K, \forall j \in V \text{ with } j \neq 0$$

If vehicle k returns to the depot from customer j (with $j \neq 0$)—so this is the last node of the route—then the route duration $d_{j,k}$ is at least the sum of the vehicle's starting time at j ($y_{j,k}$) plus the service time and the travel time from j to the depot. Again, the Big-M constraint deactivates this constraint when the arch in question is not the last travel for vehicle k .

8. **The starting time from the depot is always 0.**

$$y_{0,k} = 0, \forall k \in K$$

This sets the initial departure time at the depot to be always 0 for every vehicle $k \in K$, establishing a common starting point for all routes.

9. **Maximum route duration.**

$$d_{j,k} \leq t_{\max} x_{j,0,k}, \forall k \in K, \forall j \in V \text{ with } j \neq 0$$

If a vehicle k returns to depot from customer j , then the route duration $d_{j,k}$ must not exceed the maximum duration t_{\max} . If the arc $j \rightarrow 0$ is not chosen, the constraint is effectively deactivated.

4. Time Windows and Incompatible Triplets (module 2)

4. 1. Problem Description

The second module extends the previous scenario by adding some constraint to enhance service quality and operational efficiency. Each customer specifies a preferred time window $[a_c, b_c]$, within which the delivery must start: drivers cannot wait at customer's location.

Additionally, to ensure balanced workloads among drivers, it has introduced constraints defining incompatibilities between customers, in the form of a set of tuples $R = \{(i, j, l) | i, j, l \in C, i \neq j, j \neq l\}$. Each tuple indicates a restriction that a customer l can not be served by the same delivery vehicle if customers i and j are also served by that vehicle.

The goal remains minimizing total deliver time.

4. 2. Decision Variables

A customer assignment variable has been introduced, in order to later being able to build the constraint of the incompatible tuples.

$$z_{i,k} = \begin{cases} 1 & \text{if customer } i \text{ is served by vehicle } k \\ 0 & \text{otherwise} \end{cases}$$

$\forall i \in C$ and $\forall k \in K$.

This binary decision variable is equal to 1 if customer i is served by vehicle k , 0 otherwise.

4. 3. Constraints

10. **Link customer assignment variables to route variables.**

The purpose of this constraint is to link the customer assignment variable $z_{i,k}$ with the routing variables $x_{i,j,k}$, for which if customer i is assigned to the route of vehicle