A tailored Benders decomposition approach for last-mile delivery with autonomous robots

Laurent Alfandari[a], Ivana Ljubić[a,*], Marcos de Melo da Silva[a]

[a] *ESSEC Business School, Cergy-Pontoise, France*

Abstract

This work addresses an operational problem of a logistics service provider that consists of finding an optimal route for a vehicle carrying customer parcels from a central depot to selected facilities, from where autonomous devices like robots are launched to perform last-mile deliveries. The objective is to minimize a tardiness indicator based on the customer delivery deadlines. This article provides a better understanding of how three major tardiness indicators can be used to improve the quality of service by minimizing the maximum tardiness, the total tardiness, or the number of late deliveries. We study the problem complexity, devise a unifying Mixed Integer Programming formulation and propose an efficient branch-and-Benders-cut scheme to deal with instances of realistic size. Numerical results show that this novel Benders approach with a tailored combinatorial algorithm for generating Benders cuts largely outperforms all other alternatives. In our managerial study, we vary the number of available facilities, the coverage radius of autonomous robots and their speed, to assess their impact on the quality of service and environmental costs.

Keywords: Integer Programming, Last-mile delivery, self-driving robots, Benders decomposition

1. Introduction

The growth of the world population living in urban areas, reaching 54%, poses many logistic challenges. Designing efficient and effective transportation systems for both goods and passengers, while ensuring mobility, safety and sustainability, is the main challenge for modern city logistics. While both environmental and social factors are increasingly considered in new transportation technologies and models, the pressure for efficient distribution of goods also increases, with users often requiring the so-called same-day deliveries (Bertsimas et al., 2019; Savelsbergh & Woensel, 2016; Taniguchi, 2014; Taniguchi et al., 2014, 2016).

In 2016, the cost of global parcel delivery, excluding pickup, line-haul, and sorting, amounted to approximately EUR 70 billion (Joerss et al., 2016). According to the aforementioned McKinsey report, over the next ten years, market volumes in Germany and the US might reach 5 billion and 25 billion parcels per

[*] Corresponding author

Email addresses: `alfandari@essec.edu` (Laurent Alfandari), `ivana.ljubic@essec.edu` (Ivana Ljubić), `mrcsmlsilva@gmail.com` (Marcos de Melo da Silva) year, respectively. The biggest share (often higher than 50%) in total parcel delivery cost goes to last-mile delivery. Innovative and disruptive last-mile concepts have been proposed to cope with the increasing demand for logistic efficiency and competitive prices: pickup points networks, integrated public and freight transportation, deliveries directly into the customer car's trunk, crowd-shipping, and more recently, the use of unmanned aerial vehicles (drones) and self-driving autonomous robots (Qi et al., 2018; Savelsbergh & Woensel, 2016).

The use of drones for performing deliveries has gained increasing interest recently, with many authors investigating new mathematical models, exact and heuristic algorithms, expanding the literature on classical Traveling Salesman Problems (TSP) and the Vehicle Routing Problems (VRP). However, from a regulation point of view, the adoption of drones on practical delivery scenarios has been rendered increasingly difficult due to stricter rules concerning their operation and safety, especially in urban areas. In this context, self-driving robots have an advantage as they are designed to operate at low speeds, e.g., pedestrian speed, so that they can safely share existing sidewalks and bike lanes with people. Self-driving delivery robots were introduced much later than drones, however many initiatives can now be found such as the self-driving robots developed by e-novia (2020), Starship (2020), and Twinswheel (2020) that have been tested in many cities around the world. More recently, Amazon also announced the development of their own self-driving delivery robots, called Scout (Amazon, 2020). FedEx also tested its six-wheeled autonomous robot, called the Roxo SameDay Bot (FedEx, 2020) and is testing the Ford Digit robot for delivery. The robots by e-novia, Startship and Amazon are depicted in Fig. 1. These are sidewalk robots, with a capacity to perform a single delivery before returning to the assigned facility. There exist other recent studies (see, e.g., Jennings & Figliozzi (2019, 2020)) in which the routing of road autonomous robots with a larger capacity is considered. The latter allows robots to deliver multiple customers within the same route, but this is out of scope of the current paper.

From the modeling and planning perspective, drone-based and robot-based deliveries share some similarities. As pointed out by Boysen et al. (2018), a major difference (which also distinguishes our research from the existing literature) is the fact that drones allow for unattended delivery, which is (currently) not possible with robots. This is why the literature on drones focuses on minimizing the makespan, rather than on the attended-delivery tardiness indicators studied in this paper. Moreover, drones travel at a higher speed, so that the trucks can collect them en-route and reuse them for later deliveries, which does not apply to self-driving robots. Further important differences are highlighted in Boysen et al. (2018), Clarke & Moses (2014), Goodchild & Toy (2018), and Jones (2017).

Our contribution. This work addresses the problem of a logistics service provider (LSP) of finding an optimal route for a vehicle carrying customer parcels from a central depot to selected facilities, from where autonomous devices like self-driving robots are launched to perform *attended* last-mile deliveries. We call

(a) Yape (source: Yape-image) (b) Starship (source: Starship-image) (c) Scout (source: Amazon (2020))

Figure 1: Self-driving robots currently being employed in last-mile delivery tests.

that problem the Uncapacitated Routing-Scheduling Problem (URSP). Due dates for customer delivery are agreed beforehand. The objective is to serve all customers in a timely fashion.

Our contributions can be summarized as follows:

- As delays in the last-mile delivery may be unavoidable, we introduce three problem variants in which we minimize the following tardiness measures: the maximum tardiness, the total tardiness, or the number of late deliveries. While similar tardiness indicators have been considered in the humanitarian relief operations (Campbell et al., 2008; Huang et al., 2012), they have been mostly neglected for last-mile delivery.

- We study the complexity of all these problem variants and devise a generic Mixed Integer Linear Programming (MILP) formulation for them.

- The resulting MILP model involves a large number of variables and becomes intractable for instances of realistic size. We therefore propose a new problem-tailored Benders decomposition framework capable of handling all three URSP variants in a unifying way.

- This new exact method is based on a normalization approach which guarantees that the generated Benders cuts are sparse and numerically stable and that they can be separated using an efficient combinatorial procedure.

- The method is implemented in a branch-and-cut fashion using a modern general purpose MILP solver, which significantly improves the scalability of our approach.

- We use our new mathematical model and the exact algorithm applied in a realistic setting to answer the following questions:

– What is the impact of the coverage radius or the speed of self-driving robots (or alternatively, the number of available facilities) on the quality of service (QoS) measured by one of the three tardiness indicators?
– How are the robot speed and coverage radius affecting the environment in terms of: a) the distance traveled by the delivery truck, and b) the CO_2 emissions?
– How is the structure of the optimal solution affected by the choice of the tardiness indicator?

While VRPs combined with drone deliveries can be seen as an established concept in the literature, to the best of our knowledge our paper provides a first exact solution framework to optimize three different tardiness indicators in the context of robot-based last-mile operations. The recent work by Boysen et al. (2018) also exploits self-driving robots in combination with classical transportation problems in a more general variant where robots can also be collected en-route and launched from the trucks. They propose a mathematical model with a single tardiness indicator and provide optimal solutions for instances with less than 10 customers, and a multi-start heuristic for larger instances. Our exact method provides optimal solutions for instances with up to 100 customers, which allows an in-depth managerial study involving three different tardiness indicators. It is complementary to the one provided by Boysen et al. (2018) as the robots cannot be carried on board on the latter one.

Outline of the paper. In the remainder of this section we provide a literature overview. A formal definition for the URSP is given in Section 2, where we also investigate its complexity. A multi-commodity network flow MILP formulation is presented in Section 3. Due to the complexity of the model, we propose to solve it by a Benders Decomposition approach outlined in Section 4. Details on the separation of Benders cuts and subtour-elimination inequalities, and the computation of lower and upper bounds, are given in Section 5. In Section 6, the three variants of our Benders approach are implemented and their performance is analyzed. We assess the impact of varying the number of available facilities, the robots speed and the robots coverage radius. Concluding remarks and future works are discussed in Section 7.

1.1. Related literature

To the best of our knowledge, the work of Boysen et al. (2018) (see below) is the only one that combines scheduling and routing aspects, in the context of two-tier urban logistics for parcel deliveries using robots. The remaining literature focuses on the minimization of operational costs, which may include facility opening and/or allocation costs, while (optionally) respecting time windows. Moreover, most of the articles provide heuristic methods, sometimes combined with simulations. In this section we briefly overview these recent related works and point out the differences to some classical network design and routing problems.

Boysen et al. (2018) studied a capacitated single vehicle scheduling problem with truck-based autonomous robots for last-mile deliveries. Different from the URSP, in their problem setting the truck is loaded with both customer packets and robots. Also, besides the central depot, the customers and the robot facilities, the network has additional drop-off points from where robots in truck can be launched to perform deliveries. The truck can be reloaded with new robots in the robot facilities. The aim is to find a feasible route for the vehicle such that the number of customers served late is minimized. The authors analyze the complexity of the problem and propose a MILP formulation (which is capable of solving only small instances with up to 10 customers) and a heuristic method for larger instances, evaluated on randomly generated instances.

Related Network Design and Routing Problems. The URSP resembles in some aspects the ring-star problem (Labbé et al., 2004), the median cycle problem (Labbé et al., 2005), the Steiner ring-star problem (Xu et al., 1999), or the Traveling Purchaser Problem (Manerba et al., 2017), which are all routing-location-allocation problems with an underlying ring-star structure. Traditional applications of these problems are in the design of circular shaped transportation infrastructure (e.g, a metro line or a motorway), in the design of telecommunication networks, or routing.

The major difference between the URSP and the above mentioned problems is in the nature of the objective function, which minimizes the setup costs of the selected nodes and edges in the ring and the assignment cost of those nodes not in the ring. Hence, these are pure network *network-design* problems that mainly address strategic decisions with no scheduling aspects included. On the other hand, the URSP is a *scheduling* problem in which we assume that the infrastructure is already available, and hence, the major goal is to minimize the indicators for late deliveries. Indeed, in the URSP, we have facility nodes and customer nodes, the ring is composed of facility nodes only, but not necessarily all of them will be used and there is no construction or setup cost as no ring is actually constructed.

In another family of two-echelon Vehicle Routing Problems (2E-VRP), the second leg of delivery corresponds to routes instead of stars (see, e.g., the recent surveys by Cuda et al. (2015) and Guastaroba et al. (2016)). Among the 2E-VRP studies that focus on sustainable applications for e-commerce and city logistics, we highlight the work of Enthoven et al. (2020) who recently introduced the 2E-VRP with Covering Options. In this problem, the first echelon consists of truck routes departing from a single depot to visit two type of locations: covering locations or satellite locations, from where goods are picked up by or delivered to the customers. At covering locations with parcel lockers, customers can pick up goods themselves, whereas at satellite locations, goods are transferred to zero-emission vehicles (such as cargo bikes) that perform last-mile deliveries. The objective is to design routes covering the demands at minimum cost.

Related Two-Tier Urban Logistics Problems. We now highlight several recent studies that model last-mile deliveries using self-driving robots. The major difference to our approach is in the nature of the objective function (these studies focus on the minimization of operational costs) and in the methodology (they are all heuristics). Many of these studies also analyse the trade-offs and the pros and cons of the green last-mile deliveries, whereas in our work we primarily focus on the scheduling aspects, assuming that the necessary infrastructure is already provided.

Bakach et al. (2021) study a robot-based urban delivery by applying a sequential approach: in the first phase, they solve a facility location MILP formulation to find the minimum number p of locations of robot depots to open. In the second phase, they solve a p-median MILP formulation in which the operational cost for robot deliveries are minimized. An advantage of such a two-phase method (as opposed to an integrated approach, like ours) is that larger instances can be tackled. On the other hand, the approach of Bakach et al. (2021) is a heuristic and there is no guarantee that a globally optimal solution can be found.

Using a similar two-tier network for performing deliveries with robots, Poeting et al. (2019a) analyze two time slot selection policies for the LSP: in the first one, the customer chooses a due window, whereas in the second one, the customer triggers the delivery on demand. The authors use a TSP formulation to find the tour of the truck, and a Simulated Annealing heuristic to find the reassignment, allocation, and scheduling of the delivery robots. An agent-based model is used to simulate several days of parcel deliveries, based on the geographical data for the city of Cologne in Germany. The agent-based simulation approach is also used in Poeting et al. (2019b), where the authors compare the last-mile delivery of parcels using conventional truck-based deliveries against robot-based deliveries. For the truck only optimization problem, they propose a heuristic based on an MILP model for solving the TSP with Precedence Constraints. The robot-based delivery problem is modeled as an Orienteering Problem with Multiple Time Windows and a heuristic is presented.

Sonneberg et al. (2019) study the Electric Location Routing Problem with Time-Windows, which employs autonomous robots for last-mile delivery of parcels. The objective is to select the best location of the robot depots and the corresponding robot routes, while minimizing daily operational costs. The authors analyze how the number of compartments in the robot affects the solution costs and the amount of robots used.

Another simulation-optimization framework that focuses on the trade-off between costs and operations for multi-modal last-mile deliveries can be found in Brotcorne et al. (2019); Perboli & Rosano (2019); Perboli et al. (2018). A very recent survey on the routing problems for e-commerce and last-mile delivery is given by Archetti & Bertazzi (2021).

Drone-based deliveries. As pointed out in the introduction, drone-based deliveries allow for unattended delivery, and therefore the underlying models do not need to take into account scheduling constraints. Concerning the related literature on drone-based deliveries, the closest setting to ours has been considered by Kim & Moon (2018), in which a delivery truck loaded with parcels leaves the depot and has to serve all the customers. However, contrary to our assumptions, only a single drone station is allowed (see our central facility policy in Section 2.2) and deliveries are also allowed directly from the truck without using drones. Chauhan et al. (2019) investigate the Maximum Coverage Facility Location Problem with Drones

(MCFLPD) which seeks to locate a given number of facilities and assign drones to them, which will serve the demand points. The authors describe an MILP formulation which incorporates drone energy consumption and range constraints. They introduce greedy and three-stage heuristics for solving large instances. Multiple scenarios are considered to show the impact of the drone battery capabilities on coverage. Even though the URSP resembles the MCFLPD in the way that it also has to select a subset of robot facilities, no scheduling aspects are incorporated into the MCFLPD. Kitjacharoenchai et al. (2020) study a synchronized truck-drone 2E-VRP which allows multiple drones to fly from a truck, serve one or multiple customers, and return to the same truck for battery swap and package retrieval. Truck routes start and finish at a main depot and customers are visited either by a truck or a drone. In the second echelon of the problem, the trucks can be seen as moveable intermediate depots. The objective minimizes the total arrival time at the depot after completing the deliveries.

For a broader overview, we refer the interested reader to recent surveys on drone-delivery problems which can be found in Chung et al. (2020) and Macrina et al. (2020).

2. Problem statement and complexity analysis

In this section we provide a formal problem definition and show that finding optimal routes and allocations of customers to distribution facilities is an $\mathcal{NP}$-hard problem, whatever the tardiness indicator. We also show that if the automated vehicle is at least as fast as the truck and the coverage radius of each device is large enough, the problem becomes tractable. In this case, we show that a truck delivery to a "central" facility represents the optimal delivery policy.

2.1. The three variants of the Uncapacitated Routing-Scheduling Problem (URSP)

In the Uncapacitated Routing-Scheduling Problem (URSP) a vehicle departing from a central distribution depot at a predefined time, loaded with the packets for a group of customers, has to visit a subset of facilities where the packets are unloaded and then transported to their respective customers by autonomous robots. In addition, each delivery is supposed to take place before a given deadline. The objective is to minimize a tardiness indicator concerning the packets delivered after the deadline. Fig. 2 shows an instance of the problem and a feasible solution. The delivery deadlines for each customer are in brackets, and the travel times between two points are on the arcs. On the solution proposed, 6 clients are served late, with a maximum tardiness of 3 units, and a cumulative tardiness of 10 units.

More formally, we consider a distribution network composed of a central depot, denoted by 0, a set F of local distribution facilities and a set of customers C. Let $F_0 = F \cup \{0\}$ denote the set of facilities including the depot, then set $A_F = \{(i, j) \in F_0 \times F_0, i \neq j\}$ and $A_C = F \times C$. The network is modeled by a directed graph $G = (V, A)$, where $V = C \cup F_0$ is the set of nodes, and $A = A_F \cup A_C$, is the set of arcs. For each arc

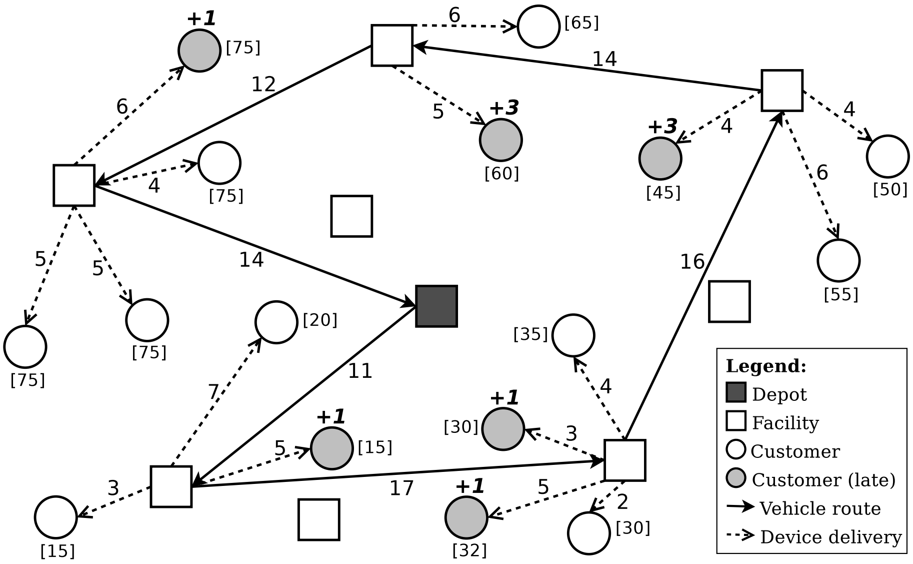


Figure 2: Routing-Scheduling feasible solution with 6 late customers, maximum tardiness 3 and tardiness sum 10.

$(i,j) \in A$, let t_{ij} be the time required for traveling from i to j. Observe that if some customer $k \in C$ is not connected to facility $i \in F$, one can still assign a time value $t_{ik} \geq L$ on arc (i,k), where L is a sufficiently large value. Therefore, we can assume all arcs exist between C and F. The following assumptions are made:

- A single vehicle is used to transfer the parcels for a set C of customers from the depot to a subset $F' \subseteq F$ of the open facilities.
- At each facility $i \in F'$, autonomous robots are launched to deliver the packets to some subset of customers $C_i \subseteq C$ assigned to it. Each robot can deliver a single customer, and then it returns to its facility.
- No customer is directly served by the depot. This is not a limiting assumption as a facility could have the same position as the depot.
- A sufficient number of autonomous robots is available at each open facility.
- Without loss of generality, service times and preparation times are integrated into the travel times t_{ik}.
- The range of robots allows to reach every customer from at least one of the facilities (otherwise, the customers that cannot be reached from any of the available locations can be preprocessed and removed from the model).
- The delivery to each customer $k \in C$ has to take place before a due date u_k. A penalty w_k may be applied per late time unit. These penalty weights w_k, if different than one, can be used to express priorities among customers.

Let $T \subset A$ be the tour which starts at 0 and visits all facilities in $F' \subset F$. We denote by $t_T(i,j)$ the traversal time from node i to node j in tour T, for $i,j \in F' \cup \{0\}$. Assuming that the truck departs from the depot at time 0, for each $k \in C$ let $t^*_T(0,k) = \min_{i \in F'} \{t_T(0,i) + t_{ik}\}$ be the traversal time of the shortest path, denoted $P^*_T(0,k)$, connecting the depot to client k via arcs of tour T. Then, we define the positive slack between the travel time along the routing path $P^*_T(0,k)$ and due date time as:

$$[t^*_T(0,k) - u_k]^+ = \begin{cases} t^*_T(0,k) - u_k, \text{ if } t^*_T(0,k) \geq u_k \\ 0, \text{ otherwise.} \end{cases}$$

Additionally, for each $k \in C$, let $l_k(T) = 1$ if k is served late in tour T, i.e., $[t^*_T(0,k) - u_k]^+ > 0$, and $l_k(T) = 0$ otherwise. The following objective functions (tardiness indicators) are considered in this paper:

- (`min-max`) Minimize the (weighted) maximum tardiness:

$$\min_T \left(\max_{k \in C} w_k [t^*_T(0,k) - u_k]^+ \right)$$

- (`min-sum`) Minimize the sum of (weighted) tardiness:

$$\min_T \left(\sum_{k \in C} w_k [t^*_T(0,k) - u_k]^+ \right)$$

- (`min-num`) Minimize the (weighted) number of late deliveries :

$$\min_T \left(\sum_{k \in C} w_k l_k(T) \right)$$

These objective functions allow to capture different preferences of decision makers. Objective `min-num` is the most restrictive as it focuses on minimizing the number of late deliveries, thus making no differentiation between deliveries which are one minute late or one hour late. Thus, it tends to preserve the overall quality of service and to reduce the number of potential customer complaints. On the other hand, objectives `min-max` and `min-sum` minimize the amount of time by which a customer is served after the deadline. Nevertheless, among the latter two functions, objective `min-max` does not directly reduce the number of customers served late as it only focuses on a single value which is the maximum delay and all other late deliveries can reach that maximum delay with no impact on the objective.

We also make use of the following notation. Given a customer $k \in C$, let $F(k) \subseteq F$ be the subset of facilities that can serve k (i.e., facilities i with $t_{ik} < L$). Given the subset $S \subset V$, $\delta^-(S) = \{(i,j) \in A \mid i \in V \setminus S,\ j \in S\}$ denotes the set of arcs entering subset S, and $\delta^+(S) = \{(i,j) \in A \mid i \in S,\ j \in V \setminus S\}$ denotes the set of arcs leaving subset S. For simplicity, if $S = \{i\}$, we write $\delta(i)$ instead of $\delta(\{i\})$. For a subset $B \subseteq A$, we define $x(B) = \sum_{(i,j) \in B} x_{ij}$.

2.2. Complexity and polynomially solvable cases

So far we have not made any particular assumption related to the speed of the delivery truck versus the speed of the autonomous robots, i.e., regarding the structure of the travel times t_{ij}, $(i,j) \in A$. Nevertheless, the URSP is a two-echelon transportation problem in which different types of vehicles are used in each delivery stage. Such vehicles are clearly traveling at different speeds, with the autonomous robots being significantly slower than the truck Consequently, the travel times t_{ij}, $(i, j) \in A$, can be decomposed into two components: times on arcs in A_F between facilities (referring to the travel times of the truck), and times on arcs in A_C between facilities and customers (referring to the travel times of autonomous vehicles).

For the special case of the URSP in which the travel matrix satisfies the triangle inequality, which happens, e.g., when the robots are at least as fast as the truck, we define the *central-facility policy* as the strategy in which the truck is sent directly from the depot to a single facility $i^* \in F$ from where all customers $k \in C$ will be served by autonomous robots, before turning back to the depot. Under this policy we can show that all three problem variants can be solved in polynomial time. Notice that if we assume that the robots have a limited range coverage the triangle inequality may be violated, as not all customers could be reached from each facility.

Proposition 1. *The central-facility policy is optimal for all three variants of the URSP if the travel times t_{ij}, $(i, j) \in A$, satisfy the triangle inequality and the robots have an unlimited range coverage. In this case, the central facility policy can be found in $O(|F||C|)$ polynomial time.*

Corollary 1. *If robots are at least as fast as the truck and both travel at constant speed, then the central-facility policy solves the problem to optimality in polynomial time.*

Proof. Assume robot speed v is $v \geq v'$, where v' is the speed of the truck. Let us assume that shortest path distances d_{ij} are pre-computed for each pair of distinct nodes $i, j \in F \cup C$. These distances d_{ij} satisfy the triangle inequality, so on A_F travel times $t_{ij} = d_{ij}/v'$ also satisfy the triangle inequality. Now for $i, j \in F$ and $k \in C$, we have $t_{ik} = \frac{d_{ik}}{v} \leq \frac{d_{ij}+d_{jk}}{v} = \frac{d_{ij}}{v} + t_{jk} \leq \frac{d_{ij}}{v'} + t_{jk} = t_{ij} + t_{jk}$. Hence travel times satisfy the triangle inequality on A, so the conditions of Proposition 1 are satisfied. □

The following proposition shows that, in general, the URSP is $\mathcal{NP}$-hard to solve. All related proofs are given in Appendix B.

Proposition 2. *If the travel times t_{ij}, $(i, j) \in A$, do not satisfy the triangle inequality, then all three variants of the URSP studied in this paper are $\mathcal{NP}$-hard, even if travel times on A_F satisfy the triangle inequality.*

We point out that Boysen et al. (2018) prove $\mathcal{NP}$-hardness for the `min-num` variant of the problem, in which the truck is allowed to carry robots along the route. If we assume that the robot-capacity of the truck is zero, then their proof of NP-hardness is also valid for our `min-num` URSP. This is why in the Appendix B we only provide a detailed proof for the `min-max` and `min-sum` URSP.

The next section describes an extended MILP formulation for the problem. The formulation uses flow variables to model delivery paths for each single customer, i.e., it belongs to a family of multi-commodity flow (MCF) formulations for routing problems (see, e.g., Letchford & Salazar-González (2015)).

3. MILP formulations

In the URSP, we assume that there are enough autonomous robots available at the distribution facilities so that all the delivery requests can be covered. The model is based on the following property:

Property 3. *For all three URSP variants, there always exists an optimal solution such that the delivery truck stops at each distribution facility at most once.*

We now devise a multi-commodity flow MILP formulation for the problem. The proposed model uses the following decision variables:

$$f_{ij}^k = \begin{cases} 1, & \text{if arc } (i,j) \text{ is on the delivery truck path to customer } k \\ 0, & \text{otherwise} \end{cases} \qquad (i,j) \in A_F,\ k \in C$$

$$z_{ik} = \begin{cases} 1, & \text{if } k \text{ is served by autonomous device launched from facility } i \\ 0, & \text{otherwise} \end{cases} \qquad k \in C,\ i \in F(k)$$

$$x_{ij} = \begin{cases} 1, & \text{if arc } (i,j) \text{ is on the tour of the delivery truck} \\ 0, & \text{otherwise} \end{cases} \qquad (i,j) \in A_F$$

$$s_k = \begin{cases} 0, & \text{if customer } k \text{ is served in time} \le u_k \\ \text{number of late time units}, & \text{otherwise} \end{cases} \qquad k \in C$$

Notice that variables f_{ij}^k can be used to model both arcs defining the vehicle tour and assignment of customers to facilities. However, to avoid redundancies, variables f^k are defined only on A_F and for the arcs on A_C we use the additional set of variables z_{ik} to represent the assignment of customer $k \in C$ to facility $i \in F(k)$.

The set of URSP feasible solutions can now be modeled by the following set of constraints:

$$\sum_{(i,j)\in A_F} t_{ij} f_{ij}^k + \sum_{i\in F(k)} t_{ik} z_{ik} \leq u_k + s_k \qquad k \in C \tag{1}$$

$$\sum_{j\in F_0} f_{ji}^k - \sum_{j\in F_0} f_{ij}^k = \begin{cases} -1, \text{ if } i = 0 \\ z_{ik}, \text{ if } i \in F(k) \\ 0, \text{ otherwise} \end{cases} \qquad i \in F,\ k \in C \tag{2}$$

$$f_{ij}^k \leq x_{ij} \qquad (i,j) \in A_F,\ k \in C \tag{3}$$

$$z_{ik} \leq x(\delta^-(i)) \qquad k \in C,\ i \in F(k) \tag{4}$$

$$\sum_{i\in F(k)} z_{ik} = 1 \qquad k \in C \tag{5}$$

$$z_{ik} \in \{0,1\} \qquad k \in C,\ i \in F(k) \tag{6}$$

$$f_{ij}^k \geq 0 \qquad k \in C,\ (i,j) \in A_F \tag{7}$$

$$s_k \geq 0 \qquad k \in C \tag{8}$$

$$x \in X \tag{9}$$

The left hand side of constraints (1) computes the travel time to reach a customer $k \in C$. Constraints (2) enforce flow conservation, while constraints (3) and (4) forbid flow through arcs which are not in the tour. Constraints (5) require each customer to be served, and constraints (6)-(8) define the variables domain. Finally, with (9), we guarantee that the arcs chosen from A_F define a tour composed of facility locations and the depot. The set of possible tours X for the delivery truck must satisfy the degree constraints (10)-(12) and the subtour elimination constraints (13):

$$X = \{x \in \{0,1\}^{|A_F|} :$$

$$x(\delta^-(i)) = x(\delta^+(i)) \qquad i \in F \tag{10}$$

$$x(\delta^-(i)) \leq 1 \qquad i \in F \tag{11}$$

$$x(\delta^+(0)) = x(\delta^-(0)) = 1 \tag{12}$$

$$x(A(S)) \leq |S| - 1 \qquad S \subseteq F, |S| \geq 3\} \tag{13}$$

To model the `min-max` objective function, we introduce an auxiliary variable $t \geq 0$ which represents the objective value. The overall model for the `min-max` URSP reads as follows:

$$\text{Minimize } \{t : t \geq w_k s_k,\ k \in C,\ (x, s, z, f) \text{ satisfy (1)-(9)}\} \tag{14}$$

We notice that constraints (1)-(9) may not be sufficient to prevent that facilities with no assigned customers are included in a tour. These empty facilities could only appear at the end of the tour, when the delivery truck is empty. Since it has no influence on the solution cost, they can be removed in a post-processing phase. Otherwise, if the travel times on A_F satisfy the triangle inequality, additional constraints (15) can be added to ensure that a selected facility $i \in F$ serves at least one customer:

$$x(\delta^-(i)) \leq \sum_{k \in C_i} z_{ik} \qquad i \in F \tag{15}$$

Similarly, constraints (13) are not redundant in our model. This is a difference between the URSP and some classical network design or routing problems (see Section 1.1). In the ring-star problems for example, constraints (2)-(3), together with non-negative route costs guarantee that optimal solutions will contain no subtours. However, because there is no cost associated to variables x in the URSP, and the objective function focuses on the tardiness indicators, if we leave out constraints (13), redundant subtours could be created, and they would need to be eliminated in a post-processing phase.

The other two variants of the URSP can be modeled in a similar way. First, the `min-sum` URSP, which aims at minimizing the sum of (weighted) tardiness, can be obtained as:

$$\text{Minimize } \{\sum_{k \in C} w_k s_k : (x, s, z, f) \text{ satisfy (1)-(9)}\}. \tag{16}$$

The minimization of the (weighted) number of late deliveries in `min-num` requires new binary variables: l_k equals 1 if customer k is served late, and 0, otherwise. Then, `min-num` URSP is modeled as:

$$\begin{aligned} \text{Minimize } \{\sum_{k \in C} w_k l_k : s_k \leq (M_k - u_k) l_k, \; k \in C, \\ (x, s, z, f, l) \text{ satisfy (1)-(9)}, \; l \in \{0,1\}^{|C|}\}. \end{aligned} \tag{17}$$

where M_k is an upper bound on the length of the path connecting 0 to k, computed as $M_k = (|F| * \max_{(i,j) \in A_F}\{t_{ij}\} + \max_{(i,k) \in A_C}\{t_{ik}\}), \forall k \in C$. Variables l_k are used to count the number of late deliveries and the validity of the bound for M_k follows from Property 3, i.e., the number of stops made by the delivery truck is at most $|F|$.

In all three models, variables s_k, $k \in C$ assume positive values whenever the traversal time of the path $P_T(0, k)$ chosen to reach customer k is longer than the due date u_k, as defined in constraints (1). Initial lower bounds for the values of s_k, $k \in C$ can be obtained in a combinatorial way, using the following proposition.

Proposition 4. *For each $k \in C$, let $P^*_G(0, k)$ be the shortest path from 0 to k on graph G, and $t^*_G(0, k)$ its value, then $s_k \geq [t^*_G(0, k)) - u_k]^+$.*

Proof. We have $s_k = [t^*_T(0, k) - u_k]^+ \geq [t^*_G(0, k) - u_k]^+$ as the shortest path from 0 to k in G is shorter than the shortest path in the subgraph of G induced by T. □

It is not difficult to see that variables z_{ik} do not have to be binary in the above models. We have:

Observation 5. *In all three models described above, constraints $z_{ik} \in \{0, 1\}$ can be replaced by $z_{ik} \geq 0$, for all $k \in C$, $i \in F(k)$, as flow conservation constraints (2) will force them to assume integral values.*

MILP model (1)-(9) provides stronger lower bounds than some compact formulations typically used in vehicle routing problems (see, e.g., the Miller-Tucker-Zemlin formulations in Toth & Vigo (2014)). However, with the increasing number of customers, the model becomes intractable for many modern MILP solvers, as the number of decision variables is in $O(|C| \cdot |A|)$. Fortunately, Observation 5 allows us to project out flow and assignment variables (f and z, respectively) in a *Benders-like fashion.* Hence, at the cost of introducing a family of *Benders cuts* (which is exponential in size), the number of variables can be reduced to $O(|A_F| + |C|)$, making it intrinsically more scalable for larger input data. This, along with an efficient combinatorial algorithm for separating Benders cuts, will allow us to derive a highly effective exact solution method based on Branch-and-Benders-cuts, as detailed in the next section.

4. Benders decomposition

In the standard decomposition approach by Benders (1962), a mixed integer linear program can be solved in an iterative fashion by keeping the "complicating variables" (whose values must be integer) in the *relaxed master problem* and projecting out continuous variables (and the associated constraints) by replacing them with two families of Benders cuts: *optimality* and *feasibility* cuts. The former ones are used to properly bound the contribution of continuous variables in the objective, and the latter ones are added to guarantee that any solution of the relaxed master problem remains feasible with respect to original constraints. For each solution of the relaxed master problem, these cuts are separated by solving linear programs (LP), and hence, the process is repeated by alternating between solving integer linear programs (relaxed master problem) and LPs (Benders subproblems), until an optimal solution is found.

In this section we propose to solve the three variants of the URSP using a Benders decomposition approach in which routes for the distribution truck are generated at the master level (described by x variables), along with the delays s_k (in time units) for serving customers $k \in C$. Flow and assignment variables are eliminated from the model, and hence, a vector (x, s) obtained by solving the relaxed master problem does not necessarily correspond to a feasible solution. In the jargon of Benders decomposition, this means that *Benders feasibility* cuts need to be generated to discard any infeasible combination (x^*, s^*) in which delays s^* cannot be guaranteed by letting the delivery truck follow the route described by $x^* \in X$.

Our approach deviates from the above described classical Benders decomposition approach in two important aspects:

1. Benders feasibility cuts are typically separated by choosing (extreme) rays of the unbounded dual of the Benders subproblem. Many authors have observed that such implementations suffer from slow-convergence and instability, due to degeneracy of the underlying Benders subproblem and the fact that the rays returned by the LP solvers do not even have to be extreme (Fischetti et al., 2010; Magnanti & Wong, 1981; Wentges, 1996). Some of the proposed techniques to alleviate these issues are the

generation of *Pareto-optimal* cuts (Magnanti & Wong, 1981; Papadakos, 2008), *facet-defining* cuts (Conforti & Wolsey, 2018) or a *normalization* of the dual of the Benders subproblem (Fischetti et al., 2010). Contrary to Fischetti et al. (2010), who propose to intersect the unbounded dual cone with a simplex, we propose an alternative and *problem-tailored normalization* approach. Moreover, instead of solving an LP, our normalization allows to exploit an efficient combinatorial procedure inspired by the one of Magnanti et al. (1986) to obtain the coefficients of the Benders cut.

2. Instead of solving each relaxed master problem as an integer linear program (ILP) as done in traditional implementations, we follow the line of research in which Benders cuts are separated on the fly in a branch-and-cut fashion (also called Branch-and-Benders-cut). Recent studies have shown that the latter allows for a significant boost in the performance of MILP solvers (Contreras & Fernández, 2014; Fischetti et al., 2016, 2017b).

4.1. Problem reformulation using problem-tailored normalized Benders cuts

For the ease of explanation, we will focus on the `min-max` URSP variant, the other two variants can be modeled analogously. After projecting out f and z variables from the MILP model shown in Section 3, the *relaxed master problem* (RMP) is initialized as follows:

$$\min_{(t,s)\geq 0} \{t \ : t \ \geq w_k s_k, k \in C, \quad x \in X\}$$

Once the solution (s^*, x^*) of the RMP is found, it has to be checked whether there exists a possibility to ship the parcels through the network defined by the values of x^*, and whether the tardiness for each customer does not exceed s^*. In a standard Benders decomposition approach this is done by checking the feasibility of the LP defined by (1)-(5) in which the values of (x, s) variables are fixed to (x^*, s^*). By Farkas Lemma, an (extreme) ray of the unbounded dual of this LP is taken to generate a Benders feasibility cut that will be added to the RMP to cut off the point (x^*, s^*) and the process is repeated until a feasible (and hence optimal) (x^*, s^*) is found.

However, we can look at the problem from a different perspective. There are namely two possible sources of infeasibility of the solution of the RMP: a) there exists a customer $k \in C$ such that there is no path between the depot and k in the network defined by x^*, and hence the delivery time is equal to $+\infty$, or b) all customers are connected to the depot, but there exists $k \in C$ such that the parcel cannot be delivered within the deadline $u_k + s_k^*$. The first source of infeasibility can be resolved by explicitly imposing connectivity between the depot and each customer, adding constraints:

$$\sum_{i \in F(k)} x(\delta^-(i)) \geq 1 \qquad\qquad k \in C$$

which can be interpreted as Benders feasibility cuts, together with the subtour elimination constraints (which are part of the description of the set X).

To deal with the second source of infeasibility, we propose a problem-tailored normalization approach. We first observe that, after adding the feasibility cuts of the first type, the `min-max` URSP can be reformulated as follows:

$$\min \quad t \tag{18}$$

$$\text{s.t.} \qquad t \geq w_k s_k \qquad k \in C \tag{19}$$

$$s_k + u_k \geq \theta_k(x) \qquad k \in C \tag{20}$$

$$\sum_{i \in F(k)} x(\delta^-(i)) \geq 1 \qquad k \in C \tag{21}$$

$$x \in X \tag{22}$$

$$t, s_k \geq 0 \qquad k \in C \tag{23}$$

where, for a given route $x^* \in X$, the function $\theta_k(x^*)$ calculates the serving time for a customer $k \in C$:

$$\theta_k(x^*) = \min \quad \sum_{(i,j) \in A_F} t_{ij} f_{ij}^k + \sum_{i \in F(k)} t_{ik} z_{ik} \tag{24}$$

$$\text{s.t.} \qquad \sum_{j \in F_0} f_{ji}^k - \sum_{j \in F_0} f_{ij}^k = \begin{cases} -1, & \text{if } i = 0 \\ z_{ik}, & \text{if } i \in F(k) \\ 0, & \text{otherwise} \end{cases} \qquad i \in F \tag{25}$$

$$f_{ij}^k \leq x_{ij}^* \qquad (i,j) \in A_F \tag{26}$$

$$z_{ik} \leq x^*(\delta^-(i)) \qquad i \in F(k) \tag{27}$$

$$\sum_{i \in F(k)} z_{ik} = 1 \tag{28}$$

$$z_{ik}, f_{ij}^k \geq 0 \qquad (i,j) \in A_F, i \in F(k) \tag{29}$$

In the jargon of Benders decomposition, we observe that for any vector (x, s), Benders decomposition requires a solution of $|C|$ independent subproblems of the form (24)-(29), called *Benders subproblem associated to customer* $k \in C$ (given in its primal form).

Lemma 6. *For any solution* $x^* \in X$ *that satisfies constraints* (21)*, the Benders subproblem* (24)-(29) *is feasible.*

Proof. Each $x^* \in X$ represents a non-trivial tour starting at the depot. Constraints (21) ensure that at least one among the facilities from which customer $k \in C$ can be reached, must be visited by the delivery truck. Hence, the values x^* provide a network along which one unit of flow can be sent from the depot to each $k \in C$, which is required to ensure that (24)-(29) is feasible. □

We observe that constraints (21) are implied by the compact model, but once the flow variables are projected out, they need to be imposed explicitly. Furthermore, we observe that the results of Lemma 6 hold, even for fractional values of x^*.

The function $\theta_k(x)$ is convex in x, and to derive a valid Benders reformulation of the problem, we will replace the *value function reformulation* constraints (20) with an exponential family of linear under-estimators of $\theta_k(x)$ that we will refer to as *normalized Benders cuts.* “Normalization” comes from the fact that the polytope of the unbounded dual of the Benders subproblem (when the problem is formulated in a traditional way, for separating Benders feasibility cuts) is intersected with a hyperplane in which the dual variable associated to the constraint (1) is fixed to one. We point out that our approach is different from a standard normalization recipe proposed by Fischetti et al. (2010), in which the dual cone is intersected with a simplex (i.e., the sum of all dual variables is imposed to be equal to one). The major advantage of our problem-tailored normalization is summarized in the following proposition which gives a characterization of the Benders subproblem (24)-(29).

Proposition 7. *Let $x^* \in X$ be a binary vector representing a subtour T that contains node 0 and also satisfies (21). Let $F_T \subset F_0$ and $A_T \subset F_T \times F_T$ represent the node set and arc set of T. Let $G_T = (F_T \cup \{k\},\ A_T \cup \{(i,k) :\ i \in F_T \cap F(k)\})$ be the support graph that extends this subtour to arcs connecting k. Then the primal Benders subproblem (24)-(29) is the shortest path problem (SPP) on G_T, whose value is $t^*_T(0,k)$.*

Proof. It is easy to see that objective function (24) together with constraints (25), (28)-(29) provide a valid formulation for the SPP on the original graph G. Then, the inclusion of constraints (26)-(27) forbid flow on those arcs for which the master solution $x^*_{ij} = 0$, $(i,j) \in A_F$, and also prevent facilities disconnected from 0 from serving customers. Consequently, only arcs in A_T can be used to send flow from the depot to the last facility. Hence, the SPP is not longer solved on G, but on G_T, which is a subgraph of G. □

Consequently, given a customer $k \in C$, and a solution of the relaxed master problem (x^*, s^*), a path $P_T(0,k)$ from 0 to customer k in subgraph G_T defines a valid Benders cut if $t^*_T(0,k) > u_k + s^*_k$. Moreover, valid Benders cuts can also be derived at fractional points x that satisfy all the constraints from the set X. It is not difficult to see that for fractional solutions x^* of the master problem, the primal Benders subproblem becomes the minimum cost flow problem on graph G, in which arc capacities are defined as x^*_{ij} for $(i,j) \in A_F$, and $x^*(\delta^-(i))$ for $(i,k) \in A_C$.

To derive appropriate Benders cuts, we exploit the LP-duality theory. The value of $\theta_k(x^*)$ can be equivalently obtained by solving the dual of (24)-(29). As the following lemma shows, this dual can be slightly simplified. Let us associate the dual variables α_i to constraints (25) for $i \in F_0$, $-\beta_{ij}$ with $\beta_{ij} \geq 0$ to constraints (26) for $(i,j) \in A_F$ and α_k to constraint (28).

Lemma 8. *The* dual Benders subproblem *associated to customer $k \in C$ can be formulated as:*

$$\max \quad -\sum_{(i,j)\in A_F} x^*_{ij}\beta_{ij} \qquad -\alpha_0 + \alpha_k \tag{30}$$

$$s.t. \qquad -\beta_{ij} - \alpha_i + \alpha_j \le t_{ij} \qquad (i,j) \in A_F \tag{31}$$

$$-\alpha_i + \alpha_k \le t_{ik} \qquad (i,k) \in A_C \tag{32}$$

$$\alpha_i \in \mathbb{R},\ \beta_{ij} \ge 0 \qquad i \in V,\ (i,j) \in A_F \tag{33}$$

Proof. This dual formulation is obtained by eliminating constraints (27) from the primal Benders subproblem as we will show that they are redundant. Consider a facility node $i \in F$, and its incoming flow $\sum_{(j,i)\in\delta^-(i)} f^k_{ji}$. By the flow preservation constraints, the incoming flow is equal to the outgoing flow. By construction of graph G_T and the nature of the objective function, either the total incoming flow will be routed within the facility subnetwork, or towards customer k. In the latter case, the flow routed from i to k corresponds to z_{ik}, and we have $z_{ik} = \sum_{(j,i)\in\delta^-(i)} f^k_{ji} \le x^*(\delta^-(i))$, where the last inequality follows from constraints (26). □

We observe that the polytope

$$\mathcal{P}_k = \{(\alpha,\beta) \in \mathbb{R}^{|F_0|+|A_F|+1} : (\alpha,\beta) \text{ satisfy (31)-(33)}\}$$

does not depend on x^* anymore, hence, by enumerating all extreme points (α^*,β^*) of $\mathcal{P}_k$, we can replace the non-linear constraint (20) with the following exponential family of *normalized Benders cuts* associated to customer $k \in C$:

$$s_k + u_k \ge \alpha^*_k - \alpha^*_0 - \sum_{(i,j)\in A_F} \beta^*_{ij} x_{ij} \qquad (\alpha^*,\beta^*) \in \text{ext}(\mathcal{P}_k) \tag{34}$$

Expression (34) can be further simplified by setting $\alpha_0 = 0$ when solving formulation (30)-(33). Moreover, if $\beta_{ij} = 0$, $\forall (i,j) \in A_F$, then α_k is expected to be equal to the length of the shortest path from the depot to customer k on the support graph G_T, as we will show later.

4.2. Combinatorial algorithm for separating normalized Benders cuts

Feasible values for the dual variables (α,β) can be obtained by solving the linear program (30)-(33) for each $k \in C$ with any available LP solver. However, the impact on the total computational time caused by the successive calls to the solver on the Benders subproblems is expected to grow with the problem size. As the number of facilities increases, so does the number of possible routes, which in turn requires more cuts to be generated. So as to avoid the computational burden of solving a LP for each subproblem, one may want to compute the values for (α,β) in a combinatorial (and faster) way.

From Proposition 7 and Lemma 8, we can solve the dual problem (30)-(33) by exploiting the structure of the classical shortest path problem (SPP). Indeed, observe that the well known dual formulation for the SPP over the original graph G can be obtained from model (30)-(33) by completely removing the β variables. Nonetheless, if not properly penalized in the objective function (e.g., if $x_{ij} < 1,\ \forall (i,j) \in A_F$), the presence of the β variables may allow the path length constraints (31) to be relaxed, thus allowing the model to find artificially long paths. Consequently, the role of the master solution x^* on the dual objective function (30) is to restrain some of the β variables so that instead of solving the SPP on the original graph G, model (30)-(33) is equivalent to solving the problem on the support graph G_T induced by x^*, as discussed in Proposition 7.

The optimal shortest paths $P^*_T(0,k)$ over the support graph G_T, for all $k \in C$, can be obtained by applying a labelling algorithm, similar to Dijkstra's algorithm on G_T, with the advantage that it takes only $O(|F_T|)$ time because G_T is a tour. As a result, we have that $\alpha^*_0 = 0$ and $\alpha^*_k = t^*_T(0,k)$, i.e., the traversal time of $P^*_T(0,k)$. We furthermore denote by $t_T(0,i)$ the length of the path from the depot to facility $i \in F_T$ along this tour.

When the Benders subproblem is a shortest path, Magnanti et al. (1986) showed (in the context of the uncapacitated network design problem) that the coefficients of Benders cuts can be computed as indicated in Proposition 9, which adapts their result to our problem's specificities and notations. Then we show in Proposition 11 that this particular setting provides Benders cuts that are particularly sparse for our problem.

Proposition 9. *For a master solution $x^* \in \{0,1\}^{|A_F|}$ and customer $k \in C$, an optimal (α^*, β^*) solution for the Benders subproblem (30)-(33) can be computed as:*

$$\alpha^*_k = t^*_T(0,k) = \min_{i \in T}\{t_T(0,i) + t_{ik}\} \tag{35}$$

and for facility nodes $i \in F$:

$$\begin{array}{lcl} (i) & \alpha^*_i = t_T(0,i) & \textbf{\textit{if}}\ \ i \in F_T, i \in P^*_T(0,k) \\ (ii) & \alpha^*_i = \min\{t_T(0,i),\ \alpha^*_k - t^*_G(i,k)\} & \textbf{\textit{if}}\ \ i \in F_T,\ i \notin P^*_T(0,k) \\ (iii) & \alpha^*_i = \alpha^*_k - t^*_G(i,k) & \textbf{\textit{if}}\ \ i \in F_{\bar{T}} = F \setminus F_T \end{array}$$

and finally

$$\beta^*_{ij} = \max\{0,\ \alpha^*_j - \alpha^*_i - t_{ij}\},\ \forall (i,j) \in A_F \tag{36}$$

As a corollary, the separation of Benders cuts for $x^* \in X$ and $k \in C$ can be performed in time $O(|F_T|)$.

Note that in the strengthened Benders cuts of Magnanti et al. (1986), there is a single formula of α^*_i for *any* node i, which would correspond to $\alpha^*_i = \min(t_T(0,i), \alpha^*_k - t^*_G(i,k))$ with our time notation, i.e. formula (ii). Since in our problem a node i is not reachable if $i \notin T$, we need to distinguish three cases (i)-(iii), also for the proof of Proposition 11. The proofs for this and the following propositions can be found in the Appendix C.

The major implication of the above result is that we can avoid solving an LP (or finding a min-cost flow) for calculating a violated Benders cut.

The following results provide additional theoretical arguments for choosing this specific calibration of Benders cuts. We show that the optimal dual multipliers (α^*, β^*) computed in Proposition 9 result in sparse and numerically stable cuts for our specific problem. We show below that more than 50% of β^*_{ij} multipliers are equal to zero. If the values of t_{ij} are integer, the cuts are numerically stable because the dual multipliers are calculated in a combinatorial way so that they also remain integer. On the contrary, if one would use cut-generating LPs as an alternative way to derive Benders cuts (as e.g., for deriving Pareto-optimal or facet-defining cuts, see Magnanti & Wong (1981) and Conforti & Wolsey (2018), respectively), this numerical stability would be lost.

Lemma 10. *In the Benders cut $s_k + \sum_{(i,j)\in A_F} \beta^*_{ij} x_{ij} \geq \alpha^*_k - u_k$, the optimal β^* vector of Proposition 9 satisfies $\beta^*_{ij} = 0$ for:*

(a) $(i,j) \in A_T$

(b) $(i,j) \in F_T \times F_T$*, i is after j in the tour induced by A_T,*

(c) $(i,j) \in F_{\bar{T}} \times F_T$*,*

(d) $(i,j) \in F_{\bar{T}} \times F_{\bar{T}}$

Proposition 11. *If the optimal solution (α^*, β^*) of the Benders subproblem is computed as in Proposition 9, and $\rho = |F_T|/|F_0|$ denotes the proportion of facility nodes, including 0, that are in tour T ($\rho \in [\frac{2}{m}, 1]$), where $m = |F_0|$), then the fraction of variables β^*_{ij} equal to zero in the Benders cut is at least*

$$g(\rho) = 1 - \rho + \frac{\rho^2}{2} + \frac{1.5\rho - 1}{m} - \frac{1}{m^2} \geq \frac{1}{2}. \tag{37}$$

Moreover, we have $\lim_{\rho\to 2/m} g(\rho) = 1 - o(1/m)$.

We thus get that the fraction of β^* variables equal to zero in our Benders cuts is always at least 0.5, and tends to 1 when the tour is small, the smallest tour being composed of a single direct trip from the depot to one facility ($\rho \to 2/m$, i.e., $|F_T| \to 2$). As mentioned earlier, sparser Benders cuts increase the computational performance of the method, which could not be necessarily achieved without a combinatorial algorithm for the cut generation.

5. Implementation details

In this section we provide implementation details and explain how feasible solutions are obtained. In the initialization phase, we calculate combinatorial lower bounds according to Proposition 4 and insert these values as default lower bounds for s_k, $k \in C$. In addition, we add constraints (10)-(12).

5.1. Separation algorithms

There are two types of subtour elimination constraints that we consider in our implementation:

$$x(\delta^-(W)) \geq 1, \qquad k \in C,\ F(k) \subseteq W \subseteq F,\ 0 \notin W \tag{38}$$

and

$$x(\delta^-(W)) \geq x(\delta^-(i)), \qquad i \in W,\ 0 \notin W. \tag{39}$$

Constraints (38) are separated only in the case $F(k) \neq F$. To this end, for a given *fractional* solution x^* of the master and a given $k \in C$, we generate an auxiliary graph $G_k = (F_0 \cup \{k\}, A_k)$, where $A_k = A_F \cup \{(i,k) : i \in F(k)\}$ and set the arc capacities to:

$$cap(i,j) = \begin{cases} x^*_{ij}, \text{ if } (i,j) \in A_F \\ \infty, \text{ otherwise} \end{cases} \qquad (i,j) \in A_k.$$

If the maximum 0-k flow on G_k is smaller than one, let W be a subset of facilities inducing the associated min-cut, such that $0 \in W$, and let W' be another subset of nodes from $F_0 \cup \{k\}$ inducing the min-cut, such that $k \in W'$. Assuming that $W \cup W' \neq F_0 \cup \{k\}$, two valid cuts are then added to the model: $x(\delta^+(W)) \geq 1$ and $x(\delta^-(W' \cap F_0)) \geq 1$, referred to as *forward* and *backward* cut, respectively. The maximum flow is calculated using the preflow-push implementation by Cherkassky & Goldberg (1997). See also (Gollowitzer & Ljubić, 2011; Manerba et al., 2017) for the separation of these cuts for related routing and network design problems. Alternatively, to separate an *integer* solution x^*, we perform a graph traversal on G_k starting from 0, and add a violated cut for a customer that cannot be reached (if any). For the separation of constraints (39), see (Ljubić et al., 2006; Fischetti et al., 2017a).

Finally, normalized Benders cuts (34) are separated at integer points only (in the so-called *lazy cut fashion*), and only when no violated cuts of types (38)-(39) are found. For a given solution (x^*, s^*) of the relaxed master problem and each $k \in C$, we find the 0-k shortest path on the support graph induced by x^*, calculate the values of (α^*, β^*) according to Proposition 9, and add the corresponding cut if $s^*_k < \alpha^*_k - \sum_{(i,j)\in A_F} \beta^*_{ij} x^*_{ij} - u_k$.

5.2. Obtaining feasible solutions: A local search heuristic

Feasible solutions are obtained by applying a simple constructive heuristic followed by a local search phase. Besides computing initial upper bounds, this heuristic is also used as a primal heuristic within the branch-and-bound tree. It takes as input a set of *available facilities* F': when the heuristic is called for the first time to initialize upper bounds we have $F' = F$, and within the branching nodes we define F' as a set of facilities visited by the current LP-optimal solution (x^*, s^*), i.e., $F' = \{i \in F : x^*(\delta^-(i)) > 0\}$.

In a pre-processing phase, the shortest paths $P^*_G(0,k)$ from the depot to each customer $k \in C$ in G are pre-computed. The values $u_k - t^*_G(0,k)$ (difference between the due date and the length of the shortest path for each customer) are then stored in a sorted list $\mathcal{L}$ in non-decreasing order.

Construction phase: we initialize a new route T' with the depot only. We then apply the following two steps:

- *Greedy insertion following the most-urgent-deadline-first policy:* We select a customer from the top of the list $\mathcal{L}$ and follow the 0-k shortest path to insert all the facilities along this path to T' (the insertion consider only the facilities in F' which are not already in T'). We apply the *best insertion policy with respect to the total tardiness criterion*, i.e., we choose the insertion position so as to minimize the sum of tardiness for all reachable customers. The customer k is then removed from the list and this step is repeated until $\mathcal{L}$ is empty.
- *Greedy insertion of the remaining available facilities:* If T' does not visit all the facilities from F', we insert the remaining ones using the same best insertion policy from above.

Local search phase: In the local search phase, we now try to improve the current tour T' by applying some of the standard moves typically used in the vehicle routing literature (Gendreau et al., 1994; Vidal et al., 2012; Vidal, 2017). More precisely, we perform *node re-insertions* and *swaps* in a Variable Neighborhood Descent fashion (Hansen & Mladenović, 2003), and finally remove unused facilities.

This local search heuristic is used to initialize feasible solutions for all exact methods tested in our computational study.

6. Computational experiments

This section presents the computational results obtained with the proposed formulations and decomposition methods. The goal is two-fold: 1) to demonstrate the computational efficiency of the proposed branch-and-Benders-cut approach when compared to alternative exact methods, and 2) to conduct a managerial study analyzing how some of the major features (like the number of available facilities, the coverage radius or the travel speed of self-driving robots) affect the late deliveries.

In what follows, we first present the computational setting. Second, we describe how the benchmark instances are generated. Third, the numerical results for each method are discussed and compared. Finally, managerial insights are provided.

The formulations, decomposition methods and other algorithms were coded in C/C++ (compiled with the g++ 8.2 compiler) and executed on an Intel® i7-7820X$^{\text{TM}}$ 4.0 GHz CPU, with 64.0 GB of RAM running under GNU/Linux Arch (*kernel* 4.20). IBM CPLEX® 12.8 was used as the LP and MILP solver. A single computation thread was used during all the experiments.

In addition to the proposed URSP MILP formulation given in Section 3, three Benders decomposition procedures were implemented;

- *Auto-Benders*, obtained by solving the URSP model with the built-in Benders decomposition implemented by CPLEX. In practice, we relaxed the integrality constraints on the flow and assignment variables and select the fully automated strategy.
- *LP-Benders*, implemented as described in Section 4, and the Benders cuts are obtained by solving the subproblems (24)-(29) as LPs.
- *SP-Benders*, where the Benders cuts are computed using the combinatorial shortest path-based (SP) procedure described in Subsection 4.2.

A computation time limit of one hour (3600 s) was imposed for each instance. All the other parameters of CPLEX were left at their default values. Additionally, all the methods are initialized with a feasible solution obtained by applying the local search heuristic described in Section 5.2.

6.1. Instance generation

The instances were generated by a procedure similar to the one described by Boysen et al. (2018). Both facilities and customers coordinates are uniformly generated on a 10 km square grid. Each benchmark set contains either 25 or 50 symmetric Euclidean problems which are classified according to the number of facilities and customers.

The travel times among facilities are based on an average truck speed of 30 km/h as in Boysen et al. (2018). Likewise, in the default scenarios, the travel times between facilities and customers were defined for autonomous robots moving at 5 km/h, i.e., at pedestrian speed. For each customer $k \in C$, the due date is computed as:

$$u_k = t^*_G(0,k) \times (\rho_{min} + \delta_k(\rho_{max} - \rho_{min}))$$

The parameters $\{\rho_{min}, \rho_{max}\}$ control the tightness and the width of the due dates time horizon, and the random number δ_k, drawn uniformly from the interval $(0, 1]$, defines how the due dates are spread. Based on preliminary experiments, the more challenging problems are those with tighter due dates. Such instances can be obtained by setting $\rho_{min} = 1$.

During our experiments the most restrictive scenarios in terms of facility-customer coverage were obtained by setting the robot speed to 5 km/h and the robot coverage radius to 30 minutes. To generate feasible instances we made sure that each customer can be reached by at least one facility and that each facility can reach at least one customer.

6.2. Analysis of computational performance

In this section we discuss the numerical results obtained with the four proposed methods for solving the URSP (MILP Formulation from Section 3, Auto-Benders, LP-Benders and our SP-Benders approach) and compare their performance. When running the *compact* MILP formulation from Section 3, we leave out the subtour elimination constraints (as they will not affect the solution value, and redundant subtours could be removed in a post-processing phase).

The experiments were carried on benchmark problems with $|C| \in \{25, 50, 75, 100\}$ customers, and $|F| = 20$ facilities. The problems are grouped according to the number of customers and for each group 25 instances were generated. The customer deadlines were generated using $\rho_{min} = 1$ and $\rho_{max} = 5$. The speed of autonomous robots was set to 5 km/h with no coverage radius limit. The three tardiness objective functions were used with uniform weights $w_k = 1.0$, for all $k \in C$. The results obtained by each of the four methods are detailed in Table 1. For each instance group and objective function, we provide the number of instances solved to optimality within the time limit (#Solved), and the corresponding average CPU time in seconds. For those instances which were not solved within the time limit, the average gap between the best integer solution found (among the four methods) and the best lower bound (for each method) is reported.

In total, SP-Benders was able to solve to optimality 231 instances out of 300 (100 instances per each objective function). At the same time, the compact formulation and automatic Benders approach find optimal solutions for only 112 instances, whereas for LP-Benders this figure drops to 70. As expected, the compact MILP formulation scales very badly with the increasing number of commodities: already for $|C| = 75$ and the `min-num` objective function not a single instance could be solved to optimality, and the average gap for the unsolved instances is higher than 95%. The remaining three Benders decomposition methods scale better with the increasing size of C, but there are still significant differences between them. SP-Benders drastically outperforms all other methods both in terms of computational time and average gap. SP-Benders shows an average gap lower than 12% over all unsolved instances and all three objective functions, whereas for the second best approach, Auto-Benders, this gap is above 60%. The weak performance of LP-Benders can be explained by the fact that the dual Benders subproblem is highly degenerate, so that the optimal dual solution (α, β) obtained from the LP solver produces numerically unstable, possibly too dense and shallow cuts. Auto-Benders uses sophisticated stabilization techniques (see, e.g., in-out approach described in (Fischetti et al., 2017a)) to overcome these issues, whereas our SP-Benders relies on normalization, sparsity of derived cuts and their numerical stability thanks to the combinatorial procedure for calculating the dual multipliers (cf. Propositions 9 and 11).

Table 1 also shows that the `min-num` and `min-sum` objective functions are generally more computationally expensive than the `min-max` objective function, and that the amount of instances solved to optimality decreases as the number of customers increases. Nevertheless, the performance of SP-Benders remains stable even with the increasing number of customers, and shows the potential of our method to be applied

Table 1: Comparison of the four URSP solution methods using the three objective functions and tight delivery deadlines.

	Compact Formulation (cf. Section 3)								
	`min-max`			`min-num`			`min-sum`		
Instances	#Solved	Time (s)	Gap (%)	#Solved	Time (s)	Gap (%)	#Solved	Time (s)	Gap (%)
ursp_20_25	23	626.50	1.36	23	1090.87	6.66	21	1016.41	13.09
ursp_20_50	17	1714.56	20.65	6	3074.85	74.50	11	2640.79	46.45
ursp_20_75	10	3005.53	46.31	0	3600.00	95.86	1	3459.05	82.59
ursp_20_100	0	3600.00	95.90	0	3600.00	99.83	0	3600.00	95.26
Total/Avg.	50		41.06	29		69.21	33		59.35

	Automatic Benders								
	`min-max`			`min-num`			`min-sum`		
Instances	#Solved	Time (s)	Gap (%)	#Solved	Time (s)	Gap (%)	#Solved	Time (s)	Gap (%)
ursp_20_25	15	1568.36	35.84	19	944.43	20.00	12	1989.87	40.20
ursp_20_50	14	1643.19	28.58	11	2192.42	44.13	9	2577.34	49.58
ursp_20_75	13	1932.53	22.24	4	3092.88	63.81	2	3332.49	77.41
ursp_20_100	9	2617.63	35.64	4	3117.09	68.97	0	3600.00	87.29
Total/Avg.	51		30.58	38		49.23	23		63.62

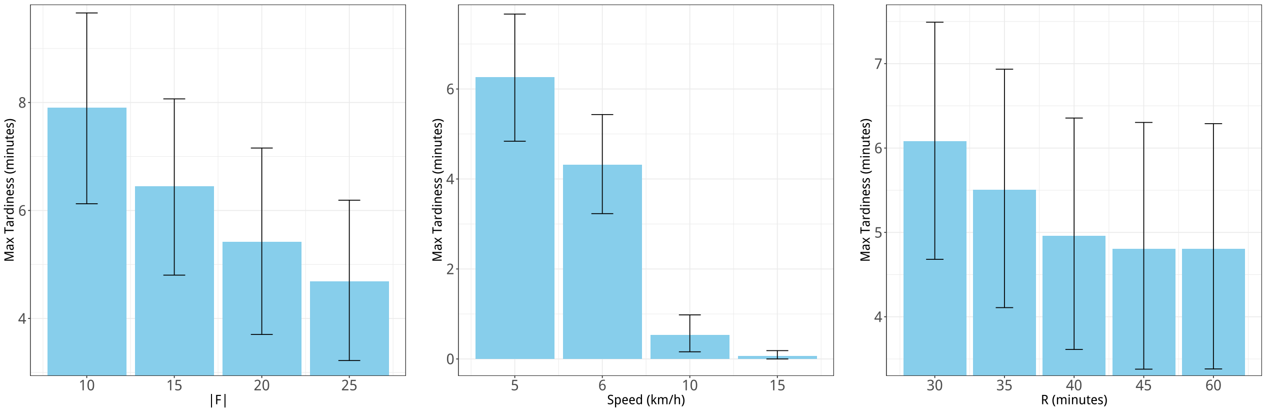
	LP-Benders								
	`min-max`			`min-num`			`min-sum`		
Instances	#Solved	Time (s)	Gap (%)	#Solved	Time (s)	Gap (%)	#Solved	Time (s)	Gap (%)
ursp_20_25	10	2317.31	40.55	15	1645.92	32.00	10	2333.23	47.75
ursp_20_50	10	2844.46	47.44	9	2693.18	51.86	4	3180.70	58.39
ursp_20_75	3	3272.03	51.52	3	3271.28	64.86	2	3448.40	73.02
ursp_20_100	2	3454.41	62.68	2	3373.58	74.51	0	3600.01	81.88
Total/Avg.	25		50.55	29		55.81	16		65.26

	SP-Benders								
	`min-max`			`min-num`			`min-sum`		
Instances	#Solved	Time (s)	Gap (%)	#Solved	Time (s)	Gap (%)	#Solved	Time (s)	Gap (%)
ursp_20_25	25	253.65	0.00	23	416.10	4.00	21	770.75	3.68
ursp_20_50	21	669.23	8.31	20	934.39	8.93	18	1137.66	12.52
ursp_20_75	23	525.43	5.14	15	1572.36	14.84	17	1609.58	11.66
ursp_20_100	21	1020.25	5.48	12	2175.63	18.82	15	1982.82	15.86
Total/Avg.	90		4.73	70		11.65	71		10.93

to even larger instances. Additional analysis of the computational performance for the `min-max` objective function is provided in the Appendix D.

6.3. Managerial insights

In this section we analyze how the QoS is affected by the number of available facilities, the speed of robots and their coverage radius. In addition, to assess the environmental impacts, we focus on the savings in km (and respectively the CO_2 emissions) for the delivery trucks. This emission analysis is the most suitable for our purposes, given the differences in electricity generation (for powering the robots) between different cities/countries. Assuming that the delivery trucks do not belong to the last-generation vehicles, we estimate emission of 200 grams of CO_2 per kilometer and estimate annual CO_2 emissions assuming 300 working days per year. We also report the length of the paths traversed by the robots. We are not analyzing operational costs, as done in e.g., (Brotcorne et al., 2019) and (Perboli & Rosano, 2019) or (Bakach et al., 2021), for several reasons: 1) Our model deals with operational (scheduling) decisions while assuming that the underlying infrastructure (robots, facilities, delivery truck with a sufficiently large capacity) is available. 2) Investment cost data (e.g., operating costs related to fleet management and maintenance, personnel costs, costs per stop, etc) vary between cities and organizations; 3) In a case study conducted by Bakach et al. (2021) the authors show that, compared with conventional truck-based deliveries, robot-based deliveries can save about 70% of operational cost. Therefore, in this section we focus on the impact of the available infrastructure and technology on: a) the quality of service, and b) the environmental costs.



(a) Influence of $|F|$ (b) Influence of device speed (c) Influence of device radius

Figure 3: Effect of increasing the number of facilities, device speed and radius to the quality of solutions.

The obtained results are summarized in Table 3 and Fig. 3. Each row in Table 3 reports average values across 50 instances from our benchmark set. When varying the number of available facilities (respectively, the robots' speed or coverage radius), we report the average number of *used* facilities in an optimal solution (second column), the average distance (in km) traveled by the delivery truck (third column), the average distance (in km) traveled by all robots (fourth column), and the average distance (in km) traveled by each robot (last column). In Fig. 3 we vary the same parameters and report the average solution value (max-tardiness in our case), together with the bars corresponding to the 95% confidence interval.

Table 2: Parameters used for the managerial study.

Parameters	Values
`Number of Customers` ($\|C\|$)	50
`Number of Facilities` ($\|F\|$)	10, 15, 20*, 25
`Device speed` (km/h)	5*, 6, 10, 15
`Device coverage radius` (R min.)	30, 35, 40, 45, 60*
`Objective function`	`min-max`*, `min-sum`, `min-num`

(*) default values.

By default, we consider the maximum tardiness indicator as the objective function. Table 2 summarizes the values of the test parameters. The selected intervals are based on the technical capabilities of existing delivery robots. For example, on the most conservative side we have the autonomous robots produced by Starship (2020), with a maximum speed of 6 km/h and maximum coverage range of 6 km, on the other side we have the robots produced by e-novia (2020), which have a maximum coverage radius of 80 km and different speed modes: it imposes a maximum speed of 6 km/h on sidewalks (pedestrian speed), and can travel at up to 20 km/h using bicycle lanes.

Effect of increasing the number of facilities. In this setting, we iteratively expanded the set of available facilities from $|F| = 10$ to $|F| \in \{10, 15, 20, 25\}$ by adding facilities to existing ones, with a uniform distribution in the space. The chart given in Fig. 3a shows that the maximum tardiness improves in a rather linear way with the increasing number of facilities available for robot delivery, with a gain of 30% when doubling the number of facilities. Interestingly, Table 3 shows that the average number of the facilities visited by the truck increases very little (from 5.84 for $|F| = 10$, to 6.18 for $|F| = 25$) indicating that the underlying cost per stop might not be significantly affected, even if a larger number of facilities is available. From the same table we observe that the length of the average truck route can be shortened by less than 10%, without a significant increase in the distances traveled by robots. Overall, these results indicate that the larger number of available facilities enables the delivery truck to get closer to customers and thus, to reduce the overall tardiness, while visiting ≈ 6 facilities on average. We conclude that a denser infrastructure in terms of available facilities has a high impact on the QoS and on-time delivery indicators. However, when it comes to environmental costs, only moderate savings of CO_2 emissions could be achieved: increasing the number of available facilities from 10 to 25 results in annual savings of ≈ 147 kg CO_2, for a single urban area represented by a 10 km square grid.

Table 3: Solution properties when varying number of available facilities, robots' speed and their coverage radius.

	Varying Number of Facilities			
$\lvert F\rvert$	Avg. tour #Facilities	Avg. truck travel distances (km)	Avg. all robots travel distances (km)	Avg. single robot travel distance (km)
10	5.84	28.91	183.87	3.68
15	6.00	27.95	183.87	3.68
20	6.06	27.26	183.29	3.67
25	6.18	26.46	184.59	3.69

	Varying Robots' Speed			
Speed (km/h)	Avg. tour #Facilities	Avg. truck travel distances (km)	Avg. all robots travel distances (km)	Avg. single robot travel distance (km)
5	6.06	27.26	183.29	3.67
6	5.46	24.17	202.12	4.04
10	4.28	19.68	264.10	5.28
15	2.90	16.01	360.67	7.21

	Varying Robots' Coverage Radius			
Radius (minutes)	Avg. tour #Facilities	Avg. truck travel distances (km)	Avg. all robots travel distances (km)	Avg. single robot travel distance (km)
30	9.08	39.79	143.04	2.86
35	7.90	35.14	154.56	3.09
40	7.46	32.46	161.21	3.22
45	6.92	30.33	169.29	3.39
60	6.06	27.26	183.29	3.67

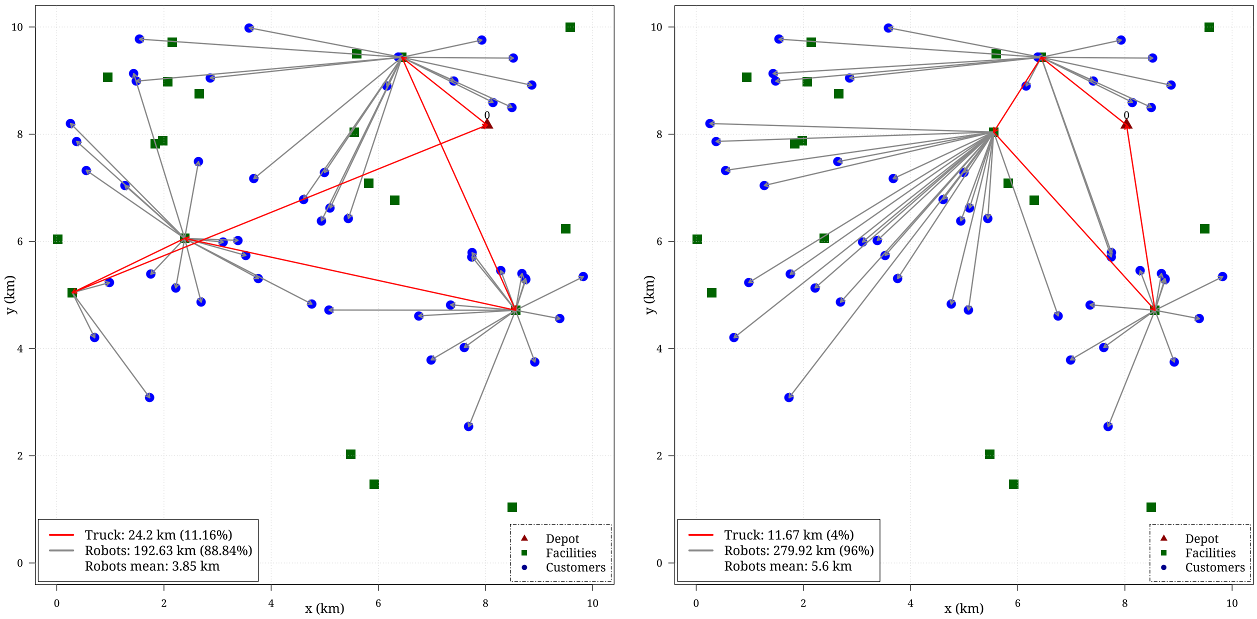
Let us now focus on a logistics service provider and compare two possible solutions: 1) vertical integration of the last-mile delivery using own infrastructure (i.e., distribution facilities and self-driving robots) versus 2) outsourcing to one or several third-party logistics (3PL) providers that offer the last-mile delivery using their own distribution facilities and self-driving robots. As we can see from Fig. 3a, having a larger set of available facilities is highly beneficial for reducing the late deliveries. On the other hand, the upfront investment cost for constructing/renting distribution facilities may be quite large for a single company. Therefore, it can be interesting for the LSP to expand the infrastructure by adding the facilities of another 3PL provider as soon as it significantly improves the coverage of the service area, and potentially sign contracts with multiple 3PL providers, sorting them by marginal improvement of the QoS. A full economic analysis and trade-off cost study, introducing the costs per stop of standard delivery, could be further conducted, following the findings of, e.g., Brotcorne et al. (2019) and Perboli et al. (2018).

Effect of increasing the device speed. In this experiment, we change the robot speed from 5 to 6, 10 and 15 km/h, respectively, while keeping $|F| = 20$, $R = 60$ minutes, $|C| = 50$. Fig. 3b shows that a significant gain in the QoS is obtained by employing robots with higher speed (the max-tardiness of more than 6 minutes drops down to near zero when robots with 5 km/h speed are replaced by ones with the speed of 10 km/h). Surprisingly, faster robots seem to be useful only up a certain threshold. For example, on our benchmark set the max-tardiness can no longer be improved by employing robots whose speed is 15 km/h, which can be explained by the fact that already with the speed of 10 km/h, most of the customers could be reached in due time. Naturally, this threshold switch varies with the problem data. The optimal selection of the speed of robots depends on the structure of the underlying city network, and hence needs to be numerically tested on the specific case study.

Table 3 provides complementary information with respect to Fig. 3b. Besides the significant reduction of the maximum tardiness, we observe another general trend: by increasing the speed of robots, less facilities are being visited (the average number of stops drops down from 6 to 2.9), the routes of the delivery truck are becoming shorter (the average length drops down from 27.3 km to 16 km), which in turn leads to longer distances traversed by self-driving robots (the average distance traversed by a robot doubles from 3.7 km to 7.2 km). In terms of annual CO_2 emission savings, increasing the speed of robots from 5 km/h to 15 km/h, results in annual savings of ≈ 675 kg CO_2, for a single urban area represented by a 10 km square grid considered in our study.

Fig. 4 shows optimal solutions for an URSP instance in which the robot speeds are set to 5 and 15 km/h, respectively. Besides the significant reduction of the maximum tardiness (from ≈ 11 to ≈ 0.5 minutes), the example also shows that for the given instance the truck route is reduced by more than 50%, whereas the average distance traveled by robots increases by approximately 45%, and fewer facilities are visited.

Effect of increasing the coverage radius. In this experiment we change the coverage radius of robots from 60 to 45, 40, 35 and 30 minutes, while keeping $|F| = 20$, $|C| = 50$, and the speed of 5 km/h. Again, the gain in the QoS is significant up to the point where the radius is large enough to cover all customers in due time. The curve of Fig. 3c has a convex shape, which indicates that the lower the radius of the current technology, the higher the marginal impact of increasing it. Table 3 provides more detailed information for all instances from our benchmark set. As in the previous case, when increasing the radius we notice a general trend that less facilities are visited (the number of stops drops from 9 to 6), the truck routes are shorter (the average



(a) Robot speed = 5 km/h, Max. Tardiness = 11.33 min.

(b) Robot speed = 15 km/h, Max. Tardiness = 0.46 min.

Figure 4: URSP selected solutions for robot speed $\in \{5, 15\}$ km/h.

length drops from $\approx$ 40 km to 27.3 km) and the distances traversed by robots are longer (on average, the length of a robot's route increases from 2.9 km to 3.7 km).

Hence, among the three factors analyzed in this work, increasing the coverage radius of robots from 30 to 60 minutes, has the highest environmental impact in terms of annual CO_2 emissions savings. Annual savings of $\approx$ 750 kg CO_2 can be achieved for a single urban area represented by a 10 km square grid considered in our study.

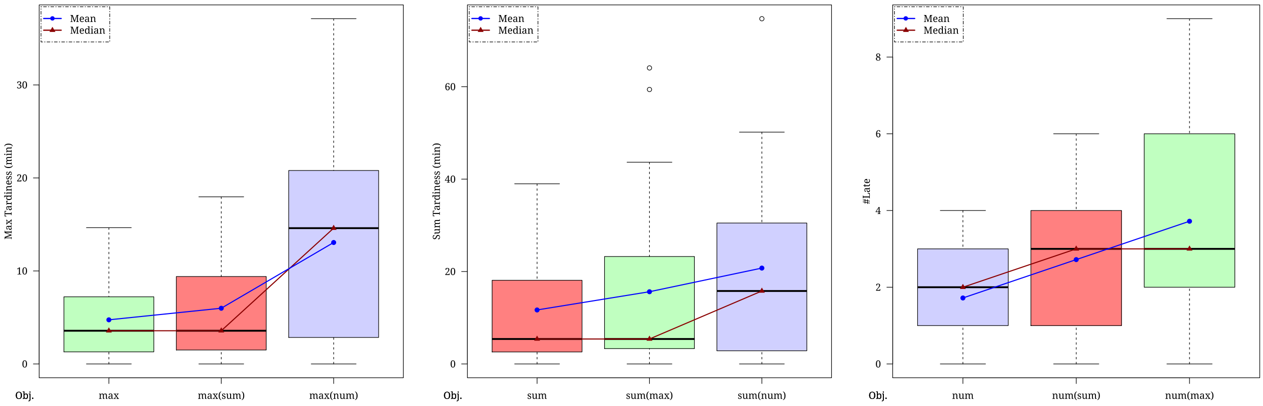
6.3.1. Comparing how the choice of the KPI affects the resulting solution

In this article, we have suggested three tardiness KPIs as possible objective functions: `min-max`, `min-sum` and `min-num`. We now attempt to answer the question: how does the choice of the KPI influence the quality of the obtained solution, i.e., if we were to find an optimal schedule according to, say, `min-max` criterion, how does such solution perform with respect to the other two criteria, namely `min-sum` and `min-num`?

To answer this question, in Figure 5 and Table 4 we compare how the value of the objective function changes when the same solution is evaluated using an alternative lateness indicator. For example, we take the solutions found by our method using the `min-sum` objective function and evaluate them using the `min-num` and `min-max` objectives, and vice-versa. That way, we indirectly measure how different the solutions are. For this analysis, we consider all the instances from our benchmark set with the default setting $|F| = 20$,

$|C| = 50$, the speed of 5 km/h, and the coverage radius of 60 minutes. The obtained results are summarized in Fig. 5 and Table 4.

In Fig. 5a, the boxplots show the distribution of the maximum tardiness by evaluating the cost of the optimal solutions obtained by all three objective functions using the `min-max` objective function. Similar re-evaluations of optimal solutions using the `min-sum` and `min-num` KPIs are shown in Figs. 5b and 5c, respectively.



(a) `min-max` objective

(b) `min-sum` objective

(c) `min-num` objective

Figure 5: Trade-offs among the three objective functions studied.

From Fig. 5a and Fig. 5b, we observe that the `min-max` and `min-sum` objective functions behave in a similar way, with the same median values obtained when switching from `min-max` to `min-sum` and vice-versa. The average optimal `min-max` tardiness is 4.74 minutes (cf. Table 4), and it increases to 6 minutes (on average) if an optimal `min-sum` solution is taken instead. Similarly, the average optimal `min-sum` tardiness is 11.7 minutes, and this value increases to 15.6 minutes (on average) if an optimal `min-max` solution is taken instead.

These deteriorations of the average objective value are much more pronounced for the `min-num` KPI that turns out to be significantly more disruptive compared to the other two KPIs. Indeed, the `min-num` objective induces a deterioration of max-tardiness of 175 % and sum-tardiness of 77 % on average, with a factor 3 to 4 for the median. This can be explained as follows. To minimize the number of late customers (`min-num`), the model sacrifices a few customers that will be served very late as tardiness measured in time units is no more controlled, in order to serve more customers in due time, and vice-versa.

Overall, the obtained results indicate that `min-sum` and `min-max` solutions are very similar, but that they can be quite different from the solutions obtained when `min-num` KPI is optimized. We therefore conclude that, if the decision-maker wishes to control both conflicting criteria (which are the number of late customers

Table 4: Comparing the quality of optimal solutions with respect to three different KPIs. Each row corresponds to optimal solutions for 25 instances obtained by one of the three objective functions. Each column shows the KPI evaluation of these solutions.

	max		sum		num		
Objective	Median	Mean	Median	Mean	Median	Mean	#Opt
min-max	(3.57)	(4.74)	3.57	5.99	14.60	13.06	25/50
min-sum	5.40	15.63	(5.40)	(11.69)	15.79	20.75	25/50
min-num	3.00	3.72	3.00	2.72	(2.00)	(1.72)	25/50

and the total/max tardiness), both of them should be included in the decision model. This can be achieved for example by fixing an upper bound on one of these criteria while optimizing another.

7. Concluding remarks

Last-mile delivery is currently being disrupted by the introduction of innovative technologies like self-driving robots, drones and other autonomous devices which impact delivery cost and safety. In this paper we studied a scheduling problem in this disruptive environment from the perspective of a logistics service provider: The problem optimizes the quality of service (on-time deliveries) by combining routing and last-mile delivery decisions, where the last-mile delivery is performed by self-driving robots. Given the limited capacities of an LSP (namely, the existing logistics network, current resources and available technology), there is potentially a large number of customers that will be served late. We proposed to optimize the delivery schedule by controlling one of the three key performance indicators: maximum tardiness over all customers, total tardiness, or the number of late deliveries. We provided a compact MILP formulation and a problem reformulation based on normalized Benders cuts. We showed that the separation of normalized Benders cuts is tractable and can be solved in a combinatorial fashion by employing a labelling algorithm. Our normalized Benders cuts provide sparse and numerically more stable cuts than their generic counterparts obtained by solving a linear program. Our exact method based on branch-and-Benders-cut allowed us to efficiently solve the problem on realistic instances of larger size.

Several managerial insights have been also derived. One of the key take-aways is that all three tardiness indicators are very sensitive to the number of available facilities. This should drive the LSP to sign contracts with several third party providers that use robot-deliveries, if available, to improve the coverage of the network and get closer to customers. The three tardiness KPIs are also highly sensitive to the speed and coverage radius of the self-driving robots, up to the point where speed or radius get high enough to cover all customers in due time. Finally, our tests showed that the total/maximum tardiness and the number of late deliveries are somewhat conflicting criteria that do not necessarily behave in the same way. These key performance indicators could be controlled in the same decision model to improve the quality of service. The problem defined in this paper is generic and could be extended to many real-case applications and variants.

When it comes to future work, several interesting directions are possible. For example, for VRP applications with large scale input data, the state-of-the-art methods are typically based on math/metaheuriscs (Vidal et al., 2012; Vidal, 2017; Gendreau et al., 1994). We therefore believe that developing matheuristics derived from the proposed mathematical framework could be a promising direction for future research, in particular for the logistics providers whose business model is based on the same-day delivery. Another promising direction would be to replace the standard delivery truck in our model by (a fleet of) electric vehicles (Desaulniers et al., 2016). Also, the available number of self-driving robots at any given facility may be limited, which leads to natural extensions of the proposed problem to the variants of the Capacitated Routing-Scheduling Problem, that cannot be handled directly with the same techniques developed in this paper. Stochastic problem variants that capture the uncertainty on travel times, would result in more challenging but highly interesting models for future research.

In addition, extending our concept of due dates to time-windows and comparing different delivery policies (e.g., same-day delivery versus next-day delivery with a time-window – see Manerba et al. (2018) for a related VRP study) could provide interesting insights concerning the environmental impacts and trade-offs between different delivery policies. Finally, our model could also be a useful tool for conducting a full economic analysis that addresses strategic questions such as: which facilities to open, with which capacity, and what should be the size of the robot fleet. Such economic analysis would need to be done in a simulation-optimization environment, similarly as done in Perboli et al. (2018); Perboli & Rosano (2019), in which our new algorithm could be plugged in as a black-box exact solution method.

Acknowledgments

This project is supported by the Initiative d'Excellence Paris Seine.

References

Afrati, F., Cosmadakis, S., Papadimitriou, C. H., Papageorgiou, G., & Papakostantinou, N. (1986). The complexity of the travelling repairman problem. *RAIRO - Theoretical Informatics and Applications*, *20*, 79–87.

Amazon (2020). Scout - amazon's self-driving robot. `http://blog.aboutamazon.com/transportation/meet-scout`. Last accessed: 2020-04-13.

Archetti, C., & Bertazzi, L. (2021). Recent challenges in routing and inventory routing: E-commerce and last-mile delivery. *Networks*, *77*, 255–268.

Bakach, I., Campbell, A. M., & Ehmke, J. F. (2021). A two-tier urban delivery network with robot-based deliveries. *Networks*, *n/a*, 1–23.

Benders, J. F. (1962). Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, *4*, 238–252.

Bertsimas, D., Jaillet, P., & Martin, S. (2019). Online vehicle routing: The edge of optimization in large-scale applications. *Operations Research*, *67*, 143–162.

Boysen, N., Schwerdfeger, S., & Weidinger, F. (2018). Scheduling last-mile deliveries with truck-based autonomous robots. *European Journal of Operational Research*, *271*, 1085–1099.

Brotcorne, L., Perboli, G., Rosano, M., & Wei, Q. (2019). A managerial analysis of urban parcel delivery: A lean business approach. *Sustainability*, *11*, 3439.

Campbell, A. M., Vandenbussche, D., & Hermann, W. (2008). Routing for relief efforts. *Transportation Science*, *42*, 127–145.

Chauhan, D., Unnikrishnan, A., & Figliozzi, M. (2019). Maximum coverage capacitated facility location problem with range constrained drones. *Transportation Research Part C: Emerging Technologies*, *99*, 1–18.

Cherkassky, B. V., & Goldberg, A. V. (1997). On implementing the push-relabel method for the maximum flow problem. *Algorithmica*, *19*, 390–410.

Chung, S. H., Sah, B., & Lee, J. (2020). Optimization for drone and drone-truck combined operations: A review of the state of the art and future directions. *Computers & Operations Research*, *123*.

Clarke, R., & Moses, L. B. (2014). The regulation of civilian drones' impacts on public safety. *Computer Law & Security Review*, *30*, 263–285.

Conforti, M., & Wolsey, L. A. (2018). "Facet" separation with one linear program. *Mathematical Programming*, . To appear.

Contreras, I. A., & Fernández, E. (2014). Hub location as the minimization of a supermodular set function. *Operations Research*, *62*, 557–570.

Cuda, R., Guastaroba, G., & Speranza, M. (2015). A survey on two-echelon routing problems. *Computers & Operations Research*, *55*, 185–199.

Desaulniers, G., Errico, F., Irnich, S., & Schneider, M. (2016). Exact algorithms for electric vehicle-routing problems with time windows. *Operations Research*, *64*, 1388–1405.

e-novia (2020). Yape - e-novia's self-driving robot. `http://e-novia.it/startup/yape`. Last accessed: 2020-04-13.

Enthoven, D. L., Jargalsaikhan, B., Roodbergen, K. J., uit het Broek, M. A. J., & Schrotenboer, A. H. (2020). The two-echelon vehicle routing problem with covering options: City logistics with cargo bikes and parcel lockers. *Computers & Operations Research*, *118*.

FedEx (2020). Sameday bot - fedex's self-driving robot. `http://about.van.fedex.com/newsroom/thefuturefedex/`. Last accessed: 2020-04-13.

Fischetti, M., Leitner, M., Ljubić, I., Luipersbeck, M., Monaci, M., Resch, M., Salvagnin, D., & Sinnl, M. (2017a). Thinning out steiner trees: a node-based model for uniform edge costs. *Mathematical Programming Computation*, *9*, 203–229.

Fischetti, M., Ljubić, I., & Sinnl, M. (2016). Benders decomposition without separability: A computational study for capacitated facility location problems. *European Journal of Operational Research*, *253*, 557–569.

Fischetti, M., Ljubić, I., & Sinnl, M. (2017b). Redesigning Benders decomposition for large-scale facility location. *Management Science*, *63*, 2146–2162.

Fischetti, M., Salvagnin, D., & Zanette, A. (2010). A note on the selection of Benders' cuts. *Mathematical Programming*, *124*, 175–182.

Gendreau, M., Hertz, A., & Laporte, G. (1994). A tabu search heuristic for the vehicle routing problem. *Management Science*, *40*, 1276–1290.

Gollowitzer, S., & Ljubić, I. (2011). MIP models for connected facility location: A theoretical and computational study. *Computers & Operations Research*, *38*, 435–449.

Goodchild, A., & Toy, J. (2018). Delivery by drone: An evaluation of unmanned aerial vehicle technology in reducing CO2 emissions in the delivery service industry. *Transportation Research Part D: Transport and Environment*, *61*, 58–67.

Guastaroba, G., Speranza, M. G., & Vigo, D. (2016). Intermediate facilities in freight transportation planning: A survey. *Transportation Science*, *50*, 763–789.

Hansen, P., & Mladenović, N. (2003). Variable neighborhood search. In F. Glover, & G. A. Kochenberger (Eds.), *Handbook of Metaheuristics* (pp. 145–184). Boston, MA: Springer US.

Huang, M., Smilowitz, K., & Balcik, B. (2012). Models for relief routing: Equity, efficiency and efficacy. *Transportation research part E: logistics and transportation review*, *48*, 2–18.

Jennings, D., & Figliozzi, M. (2019). Study of sidewalk autonomous delivery robots and their potential impacts on freight efficiency and travel. *Transportation Research Record: Journal of the Transportation Research Board*, *2673*, 317–326.

Jennings, D., & Figliozzi, M. (2020). Study of road autonomous delivery robots and their potential effects on freight efficiency and travel. *Transportation Research Record: Journal of the Transportation Research Board*, *2674*, 1019–1029.

Joerss, M., Schröder, J., Neuhaus, F., Klink, C., & Mann, F. (2016). Parcel delivery: The future of last mile. *McKinsey&Company*, . McKinsey Report on Travel, Transport and Logistics.

Jones, T. (2017). *International Commercial Drone Regulation and Drone Delivery Services*. RAND Corporation.

Kim, S., & Moon, I. (2018). Traveling salesman problem with a drone station. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *49*, 42–52.

Kitjacharoenchai, P., Min, B.-C., & Lee, S. (2020). Two echelon vehicle routing problem with drones in last mile delivery. *International Journal of Production Economics*, *225*.

Labbé, M., Laporte, G., Martín, I. R., & Salazar-González, J. J. (2004). The ring star problem: Polyhedral analysis and exact algorithm. *Networks*, *43*, 177–189.

Labbé, M., Laporte, G., Martín, I. R., & Salazar-González, J. J. (2005). Locating median cycles in networks. *European Journal of Operational Research*, *160*, 457–470.

Letchford, A. N., & Salazar-González, J.-J. (2015). Stronger multi-commodity flow formulations of the capacitated vehicle routing problem. *European Journal of Operational Research*, *244*, 730–738.

Ljubić, I., Weiskircher, R., Pferschy, U., Klau, G. W., Mutzel, P., & Fischetti, M. (2006). An algorithmic framework for the exact solution of the prize-collecting steiner tree problem. *Mathematical Programming*, *105*, 427–449.

Macrina, G., Pugliese, L. D. P., Guerriero, F., & Laporte, G. (2020). Drone-aided routing: A literature review. *Transportation Research Part C: Emerging Technologies*, *120*.

Magnanti, T. L., Mireault, P., & Wong, R. T. (1986). Tailoring benders decomposition for uncapacitated network design. In *Netflow at Pisa* (pp. 112–154). Springer.

Magnanti, T. L., & Wong, R. T. (1981). Accelerating Benders decomposition: Algorithmic enhancement and model selection criteria. *Operations Research*, *29*, 464–484.

Manerba, D., Mansini, R., & Riera-Ledesma, J. (2017). The traveling purchaser problem and its variants. *European Journal of Operational Research*, *259*, 1–18.

Manerba, D., Mansini, R., & Zanotti, R. (2018). Attended home delivery: reducing last-mile environmental impact by changing customer habits. *IFAC-PapersOnLine*, *51*, 55–60.

Papadakos, N. (2008). Practical enhancements to the Magnanti-Wong method. *Operations Research Letters*, *36*, 444–449.

Perboli, G., & Rosano, M. (2019). Parcel delivery in urban areas: Opportunities and threats for the mix of traditional and green business models. *Transportation Research Part C: Emerging Technologies*, *99*, 19–36.

Perboli, G., Rosano, M., Saint-Guillain, M., & Rizzo, P. (2018). Simulation–optimisation framework for city logistics: an application on multimodal last-mile delivery. *IET Intelligent Transport Systems*, *12*, 262–269.

Poeting, M., Schaudt, S., & Clausen, U. (2019a). A comprehensive case study in last-mile delivery concepts for parcel robots. In *2019 Winter Simulation Conference (WSC)*. IEEE.

Poeting, M., Schaudt, S., & Clausen, U. (2019b). Simulation of an optimized last-mile parcel delivery network involving delivery robots. In *Advances in Production, Logistics and Traffic* (pp. 1–19). Springer International Publishing.

Qi, W., Li, L., Liu, S., & Shen, Z.-J. M. (2018). Shared mobility for last-mile delivery: Design, operational prescriptions, and environmental impact. *Manufacturing & Service Operations Management*, *20*, 737–751.

Sahni, S., & Gonzalez, T. (1976). P-complete approximation problems. *Journal of the ACM*, *23*, 555–565.

Savelsbergh, M., & Woensel, T. V. (2016). 50th anniversary invited article - city logistics: Challenges and opportunities. *Transportation Science*, *50*, 579–590.

Sonneberg, M.-O., Leyerer, M., Kleinschmidt, A., Knigge, F., & Breitner, M. H. (2019). Autonomous unmanned ground vehicles for urban logistics: Optimization of last mile delivery operations. In *Proceedings of the 52nd Hawaii International Conference on System Sciences*. Hawaii International Conference on System Sciences.

Starship (2020). Starship's self-driving robot. `http://www.starship.xyz`. Last accessed: 2020-04-13.

Starship-image (2020). Photo of starship's self-driving robot. `https://www.instagram.com/p/BEvw7Dtic0D/`. Last accessed: 2020-04-13.

Taniguchi, E. (2014). Concepts of city logistics for sustainable and liveable cities. *Procedia - Social and Behavioral Sciences*, *151*, 310–317.

Taniguchi, E., Thompson, R. G., & Yamada, T. (2014). Recent trends and innovations in modelling city logistics. *Procedia - Social and Behavioral Sciences*, *125*, 4–14.

Taniguchi, E., Thompson, R. G., & Yamada, T. (2016). New opportunities and challenges for city logistics. *Transportation Research Procedia*, *12*, 5–13.

Toth, P., & Vigo, D. (Eds.) (2014). *Vehicle Routing: Problems, Methods, and Applications*. MOS-SIAM Series on Optimization (2nd ed.). Society for Industrial and Applied Mathematics.

Twinswheel (2020). Twinswheel's self-driving robot. `http://www.twinswheel.fr/`. Last accessed: 2020-04-13.

Vidal, T. (2017). Node, edge, arc routing and turn penalties: Multiple problems - one neighborhood extension. *Operations Research*, *65*, 992–1010.

Vidal, T., Crainic, T. G., Gendreau, M., Lahrichi, N., & Rei, W. (2012). A hybrid genetic algorithm for multidepot and periodic vehicle routing problems. *Operations Research*, *60*, 611–624.

Wentges, P. (1996). Accelerating Benders' decomposition for the capacitated facility location problem. *Mathematical Methods of Operations Research*, *44*, 267–290.

Xu, J., Chiu, S. Y., & Glover, F. (1999). Optimizing a ring-based private line telecommunication network using tabu search. *Management Science*, *45*, 330–345.

Yape-image (2020). Photo of yape self-driving robot. `https://www.instagram.com/p/Bf00f2dnxbE/`. Last accessed: 2020-04-13.

Appendix A. Notation

Table A.5: Summary of notation.

Symbol	Meaning
0	Depot
F	Set of facilities
$F(k)$	Set of facilities that can serve client k, $k \in C$
$F_0 = F \cup \{0\}$	Set of facilities including depot
C	Set of clients to deliver
$V = C \cup F_0$	Set of all nodes in the network
A_F	Set of arcs between nodes from F_0, $A_F = \{(i,j) \in F_0 \times F_0, i \neq j\}$
$A_C = F \times C$	Set of arcs between nodes from F and C
$A = A_F \cup A_C$	Set of all arcs in the logistic network
$G = (V, A)$	Graph representing the logistic network
t_{ij}	Time required for traveling from node i to node j, $i, j \in V$
u_k	Delivery due date for client $k \in C$
w_k	Penalty per late time unit for client $k \in C$
T	Vehicle tour connecting a subset of distribution facilities with the depot
$P^*_T(0,k)$	Shortest path connecting the depot to client k via arcs of tour T
$t_T(i,j)$	Traversal time from node i to node j via arcs of tour T
$t^*_T(0,k)$	Traversal time from the depot to client k via the shortest path $P^*_T(0,k)$
$P^*_G(0,k)$	Shortest path connecting the depot to client k in the network G
$t^*_G(0,k)$	Shortest traversal time from the depot to client k in the network G
R	Device's coverage radius (in minutes)
$\delta^-(S)$	Set of arcs entering a set of nodes $S \subset V$
$\delta^+(S)$	Set of arcs leaving a set of nodes $S \subset V$

Appendix B. Complexity results

Proof of Proposition 1. Assume first that the optimal solution is a tour connecting at least two facilities. We will show that by creating a new tour that visits only a single facility, at least as good travel times can be obtained, and thus, no matter which objective function is considered, the optimal solution value will be preserved.

Let T be an optimal tour visiting two or more facilities, and let i be the first facility visited in that tour. Then, a new solution T' can be created by stopping at i, and serving all customers from i. Let $k \in C$ be a customer that was served from some facility j, $j \neq i$ in the optimal solution. Due to triangle inequality, we have: $t^*_T(0,k) = t_T(0,j) + t_{jk} = t_T(0,i) + t_T(i,j) + t_{jk} \geq t_T(0,i) + t_{ik} = t^*_{T'}(0,k)$, so potentially travel times to k can be improved, and hence, the new solution associated with T' will be at least as good as the starting one.

To find the single-facility optimal solution, one can simply evaluate all single-facility solutions and take the best among them.

□

Proof of Proposition 2. Note that for the `min-num` objective, our problem is a particular case of Boysen et al. (2018) where their number δ of robots initially on the truck is zero. Even though their $\mathcal{NP}$-hardness proof assumes δ is equal to the number of customers, with a slight adaptation, a reduction from the decision version of the Hamiltonian Path problem ("is there a Hamiltonian path of length $\leq L$?") can be used to prove the $\mathcal{NP}$-hardness of the `min-num` URSP.

We now provide a detailed proof for the `min-max` URSP, and will provide a sketch of the proof for the other `min-sum` variant.

The polynomial reduction for the `min-max` URSP is from the metric Shortest Hamiltonian Path problem, which is known to be $\mathcal{NP}$-hard. Consider an instance $G = (V, E)$ of the metric Shortest Hamiltonian Path problem with $V = \{1, \ldots, n\}$, and c_{ij} the cost (or distance) between nodes i and j which satisfy the triangle inequality. We construct an instance of the `min-max` URSP as follows. Define the set of potential facilities as $F_0 = \{0, f_1, \ldots, f_n\}$, and the set of customers as $C = \{k_1, \ldots, k_n\}$ and $u_{k_i} = 0$ for $i = 1, \ldots, n$. Set $M = \max_{(i,j)\in V\times V, i\neq j} c_{ij}$. For travel times on arcs inside F_0, set $t_{0,f_i} = t_{f_i,0} = Mn/2$ for $i = 1, \ldots, n$, and $t_{f_i,f_j} = c_{ij}$ for $i, j = 1, \ldots, n$, $i \neq j$. This way, the triangle inequality holds on A_F because any path from f_i to f_j has length at most $nM = t_{f_i,0} + t_{0,f_j}$.

For times on arcs from F to C, set $t_{f_i,k_i} = 0$ for $i = 1, \ldots, n$, and $t_{f_i,k_j} = \infty$ for $j \neq i$. Fig. B.6 illustrates a transformed `min-max` URSP instance for $V = \{1, 2, 3, 4\}$. Now, because the only arc that reaches customer k_i with no infinite time is arc (f_i, k_i) (with time 0), the tour T in the URSP solution necessarily passes through each node f_i, $i = 1, \ldots, n$, which defines a Hamiltonian path over V in the original instance. The time to reach customer k_i will be the value of the path in tour T connecting 0 to k_i. Then, there exists a Hamiltonian path of length at most L in G if and only if there exists a `min-max` URSP solution with the objective value at most $Mn/2 + L$. Hence, if the decision version of `min-max` URSP was solvable in polynomial time, one could solve the decision version of the Shortest Hamiltonian Path Problem in polynomial time, which completes the $\mathcal{NP}$-hardness proof.

To prove the $\mathcal{NP}$-hardness for the `min-sum` URSP, we use a similar construction, with $u_k = 0$, $k \in C$,

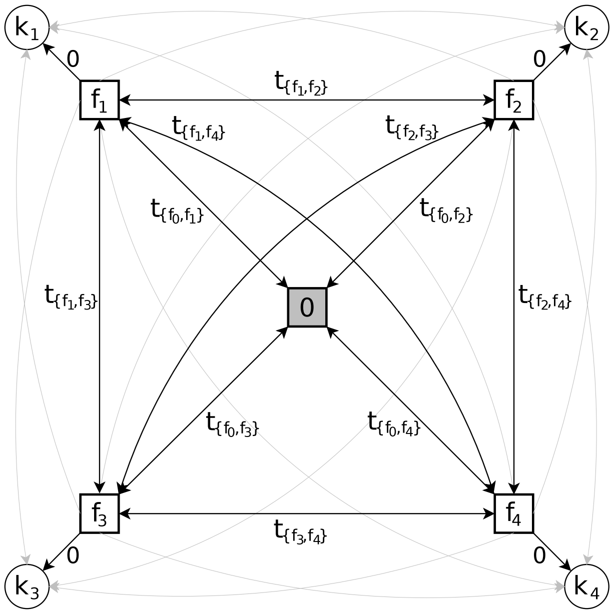


Figure B.6: Transformed URSP instance for the metric Hamiltonian shortest path with $V = \{1, 2, 3, 4\}$. The grey arcs have cost $t = \infty$.

as well, and the reduction follows from the Traveling Repairman Problem (TRP) (Afrati et al., 1986; Sahni & Gonzalez, 1976).

□

Appendix C. Structure of Benders Cuts

Proof of Proposition 9. First, let us show that (α^*, β^*) is feasible for constraints (31)-(33). Observe that as $\beta^*_{ij} \geq 0$ by (36), constraints (33) are satisfied. Also, (36) implies that $\beta^*_{ij} \geq \alpha^*_j - \alpha^*_i - t_{ij}$, i.e., $-\beta^*_{ij} - \alpha^*_i + \alpha^*_j \leq t_{ij}$ which is exactly constraints (31). As for constraints (32), there are two cases. If $i \in T$ and $\alpha^*_i = t_T(0, i)$ (cases (i) or (ii)), we indeed have $\alpha^*_k \leq \alpha^*_i + t_{ik}$ because α^*_k is the value of a shortest path from 0 to k in T which is not longer than the path that passes through node i in G_T. On the other hand, if $\alpha^*_i = \alpha^*_k - t^*_G(i, k)$ (cases (ii) or (iii)), then $\alpha^*_k = \alpha^*_i + t^*_G(i, k) \leq \alpha^*_i + t_{ik}$, so constraints (32) are satisfied in both cases. Overall, we have proven that (α^*, β^*) is feasible for the dual subproblem.

Second, we show that the objective function of the dual subproblem (30) is indeed the shortest path value from 0 to k in T, $\alpha^*_k = t^*_T(0, k)$. As $\alpha^*_0 = 0$, for that it suffices to show that $\sum_{(i,j) \in A_F} \beta^*_{ij} x^*_{ij} = 0$. We show that if $x^*_{ij} = 1$ then $\beta^*_{ij} = 0$, which is a sufficient condition. Consider an arc (i, j) such that $x^*_{ij} = 1$, i.e., $(i, j) \in A_T$. We need to distinguish the following three cases:

(a) Arc (i, j) belongs to $P^*_T(0, k)$, i.e., $\alpha^*_i = t_T(0, i), \alpha^*_j = t_T(0, j)$: in that case, because i and j are on the shortest path $P^*_T(0, k)$ from 0 to k, we have that $\alpha^*_j - \alpha^*_i = t_{ij}$ and hence, $\beta^*_{ij} = 0$.

(b) Arc (i,j) does not belong to $P_T^*(0,k)$, but node i does, i.e., $\alpha_i^* = t_T(0,i)$, $\alpha_j^* = \min(t_T(0,j), \alpha_k^* - t_G^*(j,k))$. Then $\alpha_j^* - \alpha_i^* - t_{ij} \leq t_T(0,j) - t_T(0,i) - t_{ij} = 0$ so also in that last case $\beta_{ij}^* = 0$.

(c) Arc (i,j) and both nodes i and j do not belong to $P_T^*(0,k)$, i.e., $\alpha_i^* = \min(t_T(0,i), \alpha_k^* - t_G^*(i,k)), \alpha_j^* = \min(t_T(0,j), \alpha_k^* - t_G^*(j,k))$. Then we have four subcases:

1. $\alpha_j^* - \alpha_i^* = t_T(0,j) - t_T(0,i)$, then $\alpha_j^* - \alpha_i^* - t_{ij} = 0$ and $\beta_{ij}^* = 0$ as in case (a);
2. $\alpha_j^* - \alpha_i^* = (\alpha_k^* - t_G^*(j,k)) - (\alpha_k^* - t_G^*(i,k))$, then $\alpha_j^* - \alpha_i^* = t_G^*(i,k) - t_G^*(j,k) < t_{ij}$ as $t_G^*(i,k) < t_{ij} + t_G^*(j,k)$, so $\beta_{ij}^* = 0$.
3. $\alpha_j^* - \alpha_i^* = (\alpha_k^* - t_G^*(j,k)) - t_T(0,i)$ then $\alpha_j^* - \alpha_i^* \leq t_T(0,j) - t_T(0,i) = t_{ij}$ so $\beta_{ij}^* = 0$.
4. $\alpha_j^* - \alpha_i^* = t_T(0,j) - (\alpha_k^* - t_G^*(i,k))$ then $\alpha_j^* - \alpha_i^* \leq (\alpha_k^* - t_G^*(j,k)) - (\alpha_k^* - t_G^*(i,k)) = t_G^*(i,k) - t_G^*(j,k) \leq t_{ij} + t_T^*(j,k) - t_T^*(j,k) = t_{ij}$ which leads to $\beta_{ij}^* = 0$.

□

Proof of Lemma 10. Let us show that $\beta_{ij}^* = 0$ for each case (a)-(d) one by one:

(a) if $(i,j) \in A_T$ then there are three subcases:

(a1) if $\alpha_i^* = t_T(0,i)$ and $\alpha_j^* = t_T(0,j)$ then $\alpha_j^* = \alpha_i^* + t_{ij}$ so $\beta_{ij}^* = 0$,

(a2) if $\alpha_i^* = t_T(0,i)$ and $\alpha_j^* = t_T^*(0,k) - t_G^*(j,k) \leq t_T(0,j)$ then

$$\begin{aligned}\beta_{ij}^* &= \max(0, t_T^*(0,k) - t_G^*(j,k) - t_T(0,i) - t_{ij}) \\ &\leq \max(0, t_T(0,j) - t_T(0,i) - t_{ij}) \\ &= \max(0, t_{ij} - t_{ij}) \\ &= 0\end{aligned}$$

(a3) if $\alpha_i^* = t_T^*(0,k) - t_G^*(i,k)$ and $\alpha_j^* = t_T^*(0,k) - t_G^*(j,k)$ then

$$\begin{aligned}\beta_{ij}^* &= \max(0, t_T^*(0,k) - t_G^*(j,k) - t_T^*(0,k) + t_G^*(i,k) - t_{ij}) \\ &= \max(0, t_G^*(i,k) - (t_{ij} + t_G^*(j,k)) \\ &= 0\end{aligned}$$

because the shortest path from i to k in G is shorter than a path that passes through node j (for $j \in P_T^*(0,k)$, we note $t_T^*(j,k) = t_T^*(0,k) - t_T(0,j)$ the value of the subpath connecting j to k in the shortest path to k) .

(b) If both $i,j \in F_T$ and are counter-clockwise, i.e., i is after j in tour T, then again we have three subcases:

(b1) if $\alpha_i^* = t_T(0,i)$ and $\alpha_j^* = t_T(0,j)$ then $\beta_{ij}^* = \max(0, t_T(0,j) - t_T(0,i) - t_{ij}) = 0$ because $t_T(0,i) \geq t_T(0,j)$,

(b2) if $\alpha_i^* = t_T^*(0,k) - t_G^*(i,k) \leq t_T(0,i)$ and $\alpha_j^* = t_T(0,j)$ then

$$\begin{aligned}\beta_{ij}^* &= \max(0, t_T(0,j) - t_T^*(0,k) + t_G^*(i,k) - t_{ij}) \\ &= \max(0, t_G^*(i,k) - (t_{ij} + t_T^*(0,k) - t_T(0,j))) \\ &= 0\end{aligned}$$

because $t_G^*(i,k) \leq t_{ij} + t_T^*(j,k) = t_{ij} + t_T^*(0,k) - t_T^*(0,j)$,

(b3) if $\alpha_i^* = t_T^*(0,k) - t_G^*(i,k)$ and $\alpha_j^* = t_T^*(0,k) - t_G^*(j,k)$ then we are in the case (a3) for which $\beta_{ij}^* = 0$.

(c) if $(i,j) \in F_{\bar{T}} \times F_T$ then there are two subcases:

(c1) if $\alpha_j^* = t_T^*(0,j)$ then

$$\begin{aligned}\beta_{ij}^* &= \max(0, t_T(0,j) - t_T^*(0,k) + t_G^*(i,k) - t_{ij}) \\ &= \max(0, t_G^*(i,k) - (t_{ij} + t_T^*(j,k)) \\ &= 0\end{aligned}$$

as $t_G^*(i,k) \leq t_{ij} + t_T^*(j,k)$ because the shortest path from i to k in G is shorter than any other path.

(c2) if $\alpha_j^* = t_T^*(0,k) - t_G^*(j,k)$ then

$$\begin{aligned}\beta_{ij}^* &= \max(0, t_T^*(0,k) - t_G^*(j,k) - t_T^*(0,k) + t_G^*(i,k) - t_{ij}) \\ &= \max(0, t_G^*(i,k) - (t_{ij} + t_G^*(j,k)) \\ &= 0\end{aligned}$$

for the same reason as (c1).

(d) if both $i,j \in F_{\bar{T}}$ then

$$\begin{aligned}\beta_{ij}^* &= \max(0, t_T(0,j) - t_G^*(j,k) - t_T^*(0,k) + t_G^*(i,k) - t_{ij}) \\ &= \max(0, t_G^*(i,k) - (t_{ij} + t_G^*(j,k)) \\ &= 0\end{aligned}$$

This completes the proof of the lemma. □

Proof of Proposition 11. The proof follows from Lemma 10. Summing (a) to (d) in that order, and given that the number of β^* variables is $|A_F|$ $=m(m-1)$ and $|F_T| = \rho m$, we obtain:

$$
\begin{aligned}
\%_{[\beta^*=0]} &\geq \frac{1}{m(m-1)}\left(|F_T| + (\frac{|F_T|(|F_T|-1)}{2} - 1) + |F_{\bar{T}}||F_T| + |F_{\bar{T}}|(|F_{\bar{T}}|-1)\right) \\
&\geq \frac{1}{m^2}\left(\rho m + \frac{\rho m(\rho m - 1)}{2} - 1 + (m-\rho m)(\rho m) + (m - \rho m)(m - \rho m - 1)\right) \\
&= \frac{\rho^2}{2} + (1-\rho)\rho + (1-\rho)^2 + \frac{1.5\rho - 1}{m} - \frac{1}{m^2} \\
&= 1 - \rho + 0.5\rho^2 + \frac{1.5\rho - 1}{m} - \frac{1}{m^2} = g(\rho)
\end{aligned}
$$

Given that $\rho \geq 2/m$, we get $\lim_{\rho \to 2/m} g(\rho) = g(2/m) = 1 - \frac{3}{m} + \frac{4}{m^2} = 1 - o(1/m)$. About the lower bound 0.5, to show this we distinguish two cases:

(i) If $1.5\rho - 1 \geq \frac{1}{m}$, which is equivalent to $\rho \geq \frac{2}{3}(1 + \frac{1}{m})$, then $g(\rho) \geq 1 - \rho + \frac{\rho^2}{2} \geq 0.5$ for $\rho \in [0, 1]$. This holds because the function $1 - \rho + \frac{\rho^2}{2}$ is decreasing over $[0, 1]$ and thus reaches its minimum 0.5 at $\rho = 1$.

(ii) Otherwise, if $\rho < \frac{2}{3}(1 + \frac{1}{m})$ then $g(\rho)$ is decreasing over this interval for $m \geq 7$), so in that case

$$
\begin{aligned}
g(\rho) &\geq g\left(\frac{2}{3}(1 + \frac{1}{m})\right) \\
&= \frac{1}{3} - \frac{2}{3m} + \frac{2}{9}(1 + \frac{1}{m})^2 \\
&= 0.5 + \frac{1}{18} - \frac{2}{9m} + \frac{2}{9m^2} \\
&= 0.5 + \frac{1}{18}\left(1 - \frac{2}{m}\right)^2 \geq 0.5
\end{aligned}
$$

Overall, from (i) and (ii) we have proven that the proportion of β^*_{ij} equal to zero is at least 0.5 at worst case. □

Appendix D. Additional computational results

Appendix D.1. Comparison of the computational performance

The overall performance of the four tested methods (SP-Benders, compact MILP formulation, Auto-Benders and LP-Benders) for the `min-max` objective is summarized in the graph depicted in Fig. D.7. Analyzing the cumulative amount of instances for which the integrality gap was equal to a given percentage by the end of the search time, one can see that SP-Benders outperforms all other approaches both in terms of number of problems solved optimally and average integrality gap. SP-Benders proved optimality for 90% of the instances, followed by Auto-Benders and the MILP Formulation, which solved 51% and 50% of instances, respectively. The worst performance is obtained by LP-Benders, with only 25% of the instances being solved within the time limit.

Counting the instances for which the integrality gap was greater than 50% when the search was interrupted, we have 4% of instances for SP-Benders, 25% for Auto-Benders, more than 40% for the compact model, and more than 50% for LP-Benders. Similar observations can be derived for the other two objective functions.

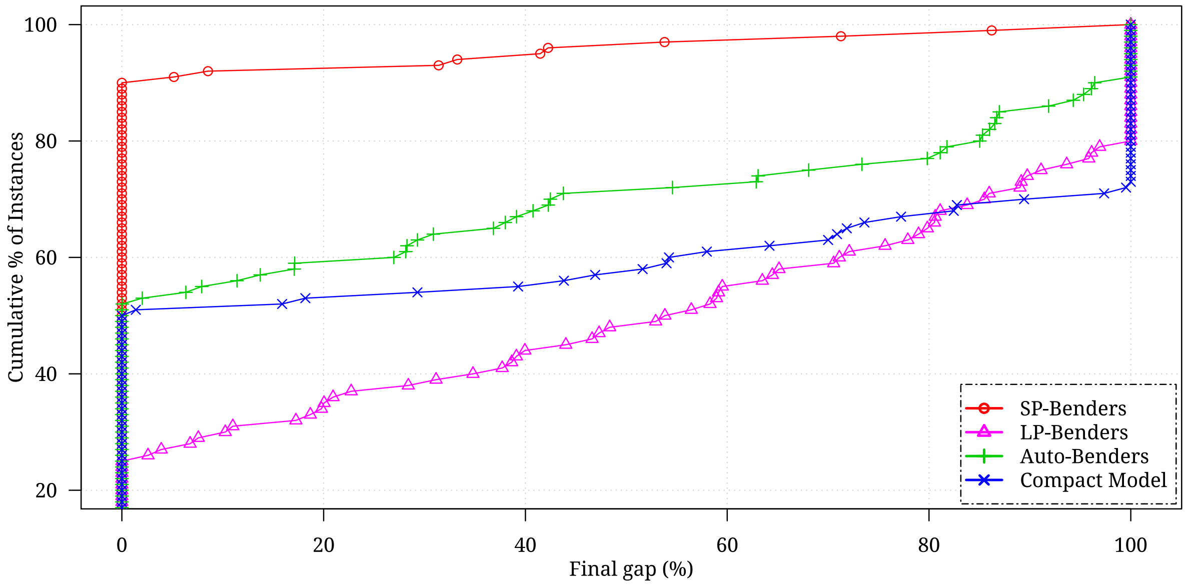


Figure D.7: Final gap comparison among the four methods (SP-Benders, LP-Benders, Auto-Benders, and MILP formulation) proposed for the URSP using the `min-max` objective function.

The boxplot graph depicted in Fig. D.8 provides more detailed computational results for each one of the proposed methods. We point out that in Fig. D.8 the running times, $t(s)$, are shown in logarithmic scale.

Appendix D.2. Examples

To illustrate the effect of increasing the number of facilities on the max-tardiness, two optimal solutions for an URSP instance with $|F| = 10$ and $|F| = 25$ are depicted in Fig. D.9. We observe that by incorporating more facilities, all customers could be served on time.

Similarly, the effect of the increased coverage radius is illustrated in Fig. D.10. We show two optimal solutions for an URSP instance in which the coverage radius is set to 30 and 60 min, respectively. When increasing the radius we notice that for the given instance the truck route is reduced by more than 40%, whereas the average distance traveled by robots increases by 20%, and fewer facilities are visited.

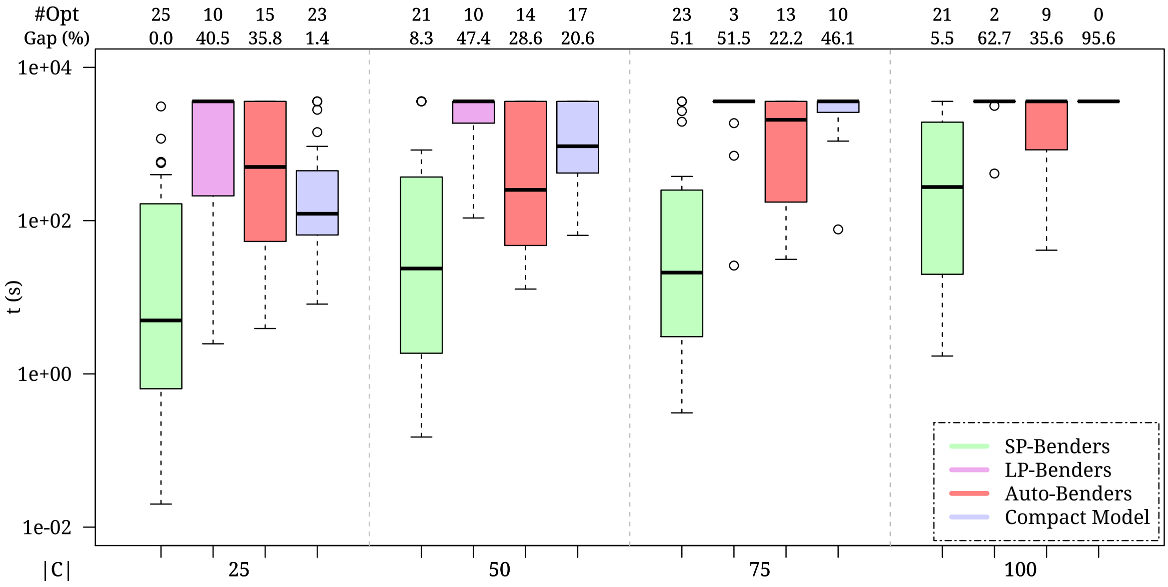
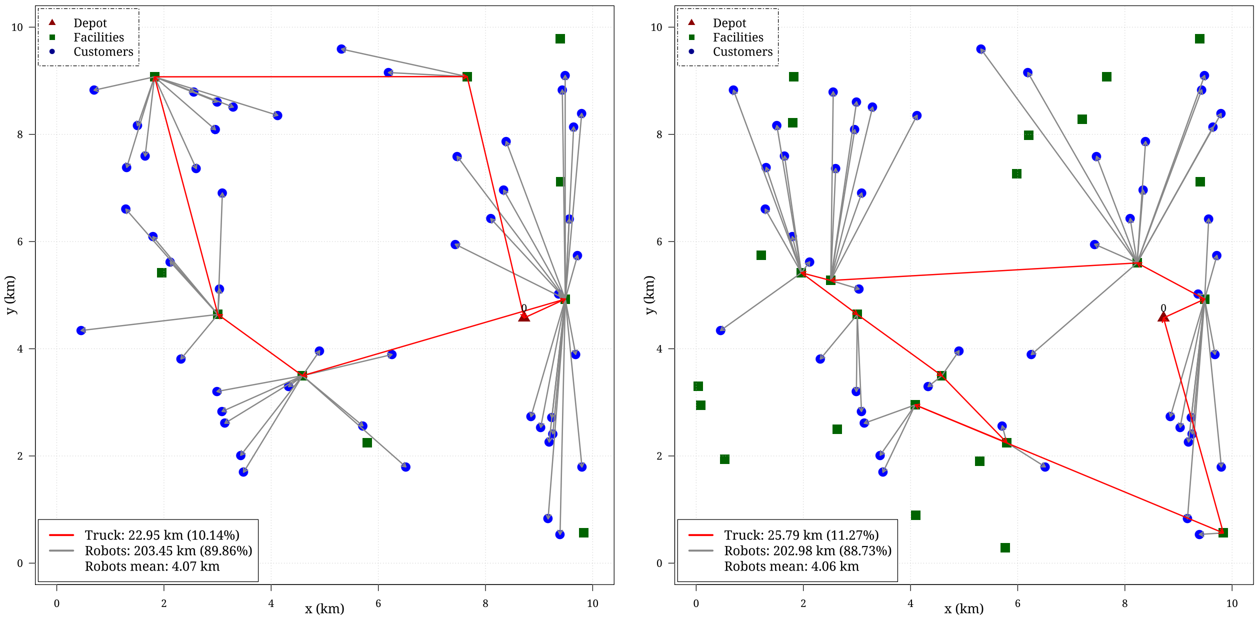


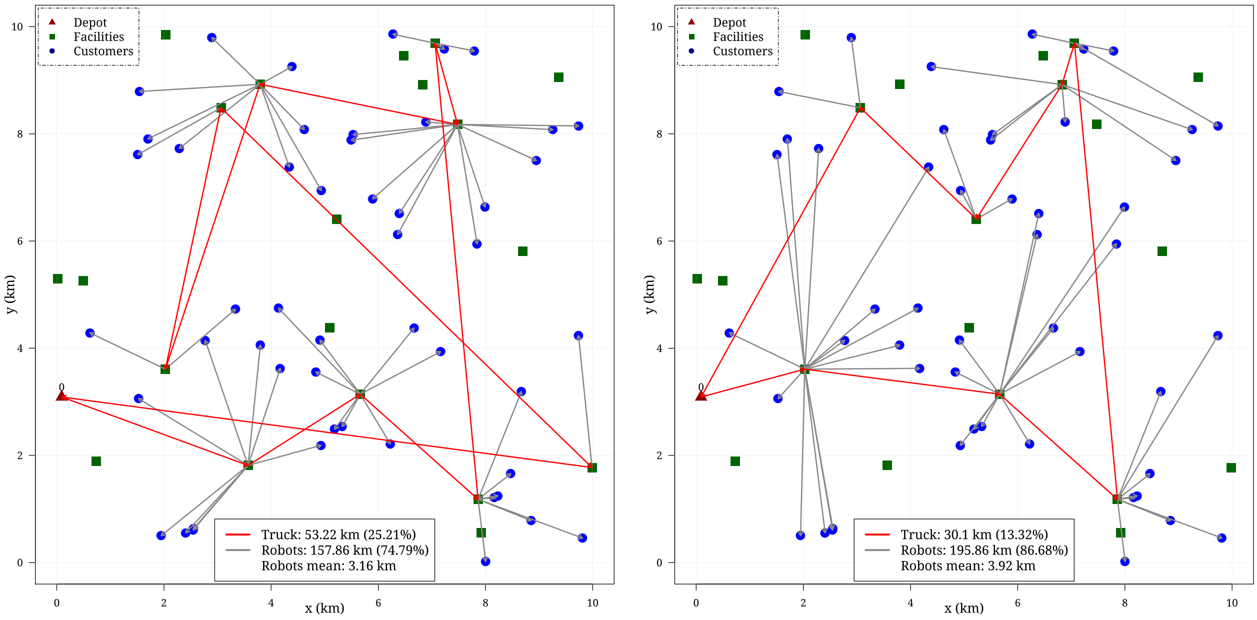
Figure D.8: Running time comparison among the four methods (SP-Benders, LP-Benders, Auto-Benders, and MILP formulation) proposed for the URSP using the `min-max` objective function.



(a) $|F| = 10$, Max. Tardiness = 3.39 min.

(b) $|F| = 25$, Max. Tardiness = 0.00 min.

Figure D.9: URSP selected solutions for $|F| \in \{10, 25\}$.



(a) R = 30 min, Max. Tardiness = 6.84 min.

(b) R = 60 min, Max. Tardiness = 5.81 min.

Figure D.10: URSP selected solutions for coverage radius $R \in \{30, 60\}$ min.