Last mile deliveries with lockers: formulations and algorithms

Giovanni Buzzega1 · Stefano Novellani2

Accepted: 17 November 2021 / Published online: 27 January 2022
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2021

Abstract
In this paper, we consider the use of lockers in parcel delivery, a recent method used in last mile logistics. Lockers are pickup points made of several cells that are located in several points of a city where customers can collect their parcels as an alternative to home delivery. We study routing problems in which one or multiple vehicles are used to deliver parcels directly to customers or lockers. We also study the influence of the introduction of lockers when these problems include time windows. We propose a set of novel formulations for these problems, some valid inequalities, and a branch-and-cut algorithm. Moreover, we investigate the difference between the routing problems with lockers and the classical routing problems.

Keywords Last mile · Lockers · Branch-and-cut · Formulations · Routing

1 Introduction
Retail e-commerce sales worldwide have experienced a continuous growth from 2014 to 2020, the year in which the sales value has more than doubled with respect to 2016, despite the current pandemic, and forecasts expect the sales value to be more than three times as high in 2023 (see, e.g., Statista 2020). One of the main challenges introduced by the e-commerce boom that retailers and logistics companies must overcome is the quick delivery of a large number of parcels at minimum cost, with a particular focus on the last mile delivery. To do so, they are implementing alternative delivery methods with respect to home delivery. One of the strategies put in place by these companies is the use of pickup points (see, e.g., Amazon 2020) located in places that are convenient to the customers in which couriers deliver the parcels that will be collected by the customers at a later time. The introduction of pickup points for the last mile deliveries as an alternative to home delivery offers two main advantages: the reduction in traveling costs and external costs due to the fact that more parcels are delivered in the same place and thus the couriers can travel shorter routes; moreover, the lack of need of synchronization between the couriers and the customers allows more flexibility in both delivery and collection times that hence decreases the amount of failed deliveries.

Pickup points can be divided into two types: counters and lockers (see, e.g., Amazon 2020). Counters are pickup points located in shops, in which parcels are handled manually by the shopkeeper and thus are subject to opening hours. Lockers are (automated) self-service facilities which are located externally and allow customers to autonomously collect parcels 24 hours a day, seven days a week.

The number of papers considering logistics and optimization problems that include pickup points is limited but increasing. In this paper, we tackle a last mile problem in which one or more identical vehicles can either deliver parcels directly to the customers or to nearby lockers (the adaptation to the counter case is straightforward and can be done by setting more restrictive time windows to the pickup points in order to respect the shopkeepers working hours). We study the cases with and without time windows for which we propose a set of formulations and branch-and-cut algorithms.

The contributions of this paper are the following: (i) We propose enhanced exact methods to solve the traveling salesman problem with lockers, the traveling salesman problem with lockers and time windows, the vehicle routing problem with lockers, and the vehicle routing problem with lockers and time windows. Note that most of the papers in the related literature propose heuristic or metaheuristic algorithms, and
that the mathematical formulations are sometimes used just as a detailed description of the problem. (ii) We define a set of valid inequalities for the proposed formulations of the studied problems. (iii) We consider the capacity of the trucks as a constraint that limits the routes, despite this is not considered in the relevant literature. (iv) We provide a comparison among several exact methods for the proposed problems. (v) We improve the results of other methods from the literature that solve similar problems: our methods were able to obtain new optimal solutions for instances of the literature for the first time and improve several best-known solutions. (vi) We provide a discussion on the costs and benefits linked to the introduction of lockers and time windows in the last mile delivery.

In the remainder of the paper, we firstly propose a detailed literature review in Sect. 2. Section 3 provides an explanation on the considered problems, while in Sect. 4 several formulations are proposed to model these problems. The implementation of the branch-and-cut algorithm is defined in Sect. 5, while Sect. 6 presents computational results comparing the proposed formulations with respect to each problem and provides a comment on the impact of the introduction of lockers in the last mile delivery process. Section 7 concludes the paper.

2 Literature review

The introduction of pickup points in logistics optimization problems is getting more and more attention; moreover, most of the studies that consider the use of lockers have been published in the last five years. For a wide classification on recent studies, we head the reader to the survey by Rohmer and Gendron (2020).

Jiang et al. (2020) study the traveling salesman problem with time windows for the last mile delivery in online shopping, a generalization of the traveling salesman problem (TSP) with time windows in which customers can either receive their parcels at home or pick them up from shared delivery facilities (lockers) nearby. Lockers have a maximum radius in which customers can be served and a maximum capacity. No preference between the two types of service is considered; however, the objective function aims at minimizing the truck traveling costs, the traveling cost of the customer to reach the locker, and the cost of opening a locker. The authors propose a Mixed Integer Linear Programming (MILP) formulation and a variable neighborhood descent. The MILP formulation models hard time windows; however, in order to improve the efficiency of the commercial solver, they only solve the case with soft time windows. The largest instance solved exactly counts 60 customers and 6 lockers.

Most of the other works consider multiple vehicles. Oliveira and dos Santos (2020) study a problem in which a limited number of lockers can be opened and used to serve customers. Two different sets of vehicles are employed to carry out the deliveries: one fleet for home delivery and one for servicing lockers. Customer requests are not necessarily unitary. Lockers have a maximum radius and a capacity, but no opening cost is considered; in fact, the problem minimizes the traveling costs. The authors propose a MILP formulation and solve instances with up to 75 customers and 10 lockers; they also introduce a variable neighborhood descent metaheuristic. Osaba et al. (2018) study a multi-objective optimization problem allowing for delivery at third-party drop-off points instead of direct delivery at the customer locations. The problem aims at optimizing bike routes under considerations of total profit, route distance, and the safety of the bikers. They propose six different evolutionary heuristics and solve three instances with 28 customers and 13 drop-off points. Mancini and Gansterer (2021) propose the VRP with private and shared delivery locations where customers can either be served at home within a set time window or at a locker with an unlimited homogeneous fleet of uncapped trucks. The truck routes are constrained by a maximum duration. Customers can select one of the two types or let the company choose between the two types of services. The service via locker implies a compensation to the customer and thus has a cost for the company. They define a maximum capacity for the lockers, but the used instances have lockers with a very large capacity. The authors propose a MILP model, a large neighborhood search (LNS)-based matheuristic, and an iterated local search algorithm. They study different time windows preferences, different radii of service, and different compensation schemes. They could solve exactly most of the 50 customer instances, and heuristic solutions are provided for the 75 customer instances.

Grabenschweiger et al. (2021) propose the VRP with heterogeneous locker boxes, where customers have multiple requests that can be delivered either at home (respecting a time window) or at a locker by an unlimited homogeneous set of uncapped trucks. Trucks must travel for at most a set maximum duration. Customers can be served by the closest locker (if they allow so) for which they receive compensation, that is a cost to the company. If the delivery happens at a locker, then all the parcels of the same customer are stored at the same locker. The authors consider the case with lockers having boxes (or cells) of a single dimension or of different dimensions. They propose two MILP models, each of which considers one of the two features just described and an Adaptive LNS-based metaheuristic. They could solve exactly most of the 25 customer instances, and heuristic solutions are provided for the 75 customer instances, improving some large solutions with respect to Mancini and Gansterer (2021). Orenstein et al. (2019) propose a vehicle routing problem (VRP), called flexible parcel delivery problem, for delivering parcels exclusively to lockers. Truck routes are limited
by a maximum shift length. Each parcel is associated with a set of possible destinations (lockers). Parcels have different sizes, and both lockers and trucks are divided into cells of different sizes. No more than one parcel can be accommodated in one cell, and a cell must be sufficiently large to contain a parcel. The objective is to minimize the traveling costs and the penalties incurred in case of failed delivery. The authors propose a MILP formulation and two metaheuristics based on the Clarke and Wright’s (C&W) Savings and the Petal heuristics. They consider instances with up to 50 lockers and 1500 parcels, but they solve to optimality only three instances with 20 lockers and 200 parcels. Wen and Li (2016) propose 1500 parcels, but they solve to optimality only three instances heuristics. They consider instances with up to 50 lockers and propose a MILP formulation and two metaheuristics based the penalties incurred in case of failed delivery. The authors parcel. The objective is to minimize the traveling costs and in one cell, and a cell must be sufficiently large to contain a different sizes. No more than one parcel can be accommodated set of possible destinations (lockers). Parcels have different sizes. The authors study the impact of the introduction of lockers. The problem aims at minimizing the variable transportation costs, the soft time windows penalty, the \( CO_2 \) emission costs, and the fixed cost of labor and vehicles. Once the lockers are introduced, the time windows constraints are not considered anymore, but the locker renting costs and the new small vans acquisition costs are included in the objective function. The authors model the problem with MILP formulations and present a C&W Savings-based heuristic to solve a real-world instance with 12 customers. Dumez et al. (2021) present the vehicle routing problem with delivery options (VRPDO), a VRP with time windows in which several delivery options such as home, workplace, locker, or car trunk are considered. The authors model customer preferences: the customers define their delivery preferences by ordering the options. Both vehicles and lockers are capacitated, and a maximum travel time is imposed on trucks. The objective is to define the truck routes minimizing the traveling costs and the number of used vehicles while guaranteeing the minimum satisfaction level of the customers. The authors propose a formulation, but they solve the problem with a dedicated LNS that heuristically solves instances with up to 200 customers. Tilk et al. (2021) propose a branch-and-price-and-cut algorithm to solve the VRPDO that solves instances with up to 60 customers.

Sitek and Wikarek (2019) consider both pickup and delivery proposing a capacitated VRP (CVRP) called CVRP with pick-up and alternative delivery in which heterogeneous vehicles are used to serve customers at home, at a postal office, or at a capacitated locker. Vehicles must respect a maximum shift duration. The objective is that of minimizing the distance traveled by the trucks and the penalties linked to customer preferences. The authors propose an ILP formulation and a heuristic algorithm, both enhanced by a constraint programming preprocessing. They could solve instances with 20 vehicles, 200 delivery points, and 2000 parcels.

Zhou et al. (2018) propose the multi-depot two-echelon VRP with delivery options for the last mile distribution; in the first echelon, the parcels are moved from the main depots to the intermediate ones (satellites), while the parcel delivery is performed in the second echelon. Customers can define a preference for a set of lockers or their home. Two different homogeneous capacitated fleets are used for the two echelons, and a maximum traveling time is imposed on each vehicle. Satellites have a maximum capacity, while lockers do not; however, both can be visited by several vehicles. The objective is to minimize the fixed costs of the depots and the fixed and variable costs of vehicles and satellites. The authors propose a hybrid multi-population genetic metaheuristic. The larger instance used includes two depots, 12 satellites, 30 lockers, and 200 customers. Enthoven et al. (2020) also solve a two-echelon problem: the first echelon accounts for deliveries at satellites or directly to lockers, while in the second echelon they consider deliveries from the satellites to customer houses with zero emission vehicles. The problem minimizes the traveling costs and the cost linked to the lockers. The authors propose a MILP formulation and an adaptive LNS. Their model can solve instances with up to 51 customers, four satellites, and four lockers.

Zhou et al. (2019) present the green VRP considering dual services (home delivery and customer pickup) with stochastic travel times. A fleet of capacitated identical vehicles is used to perform the deliveries to the customers that are associated with non-unitary requests and that are already divided into home delivery and self-collection at uncapacitated lockers. Time windows and a maximum traveling time for each vehicle are considered and are subject to stochastic traveling times. The objective is to minimize the transportation costs and the expected recourse cost that includes the violation of the time windows and of the maximum travel time. The problem is formulated as a two-stage stochastic optimization model with recourse strategy and solved by applying a two-stage heuristic algorithm. They consider instances with up to 120 customers.

In the following, we focus on the TSP with lockers and time windows and we extend it to the VRP case, as the majority of the papers that consider lockers. Most of the papers described in this section solve the proposed problems with heuristic or metaheuristic algorithms, sometimes only reporting a mathematical formulation to provide a detailed description of the problem. In the following, we propose a set of formulations to solve the tackled problems with exact methods, a set of valid inequalities, and a comparison between our methods and exact methods proposed in the literature to solve similar problems. Moreover, we analyze the obtained results and the impact of lockers and time windows on the solutions.

### 3 Problems definition

In this section, we describe the set of problems that we will tackle in the following. We firstly describe the traveling salesman problem with lockers (TSPL), which is a generalization
of the TSP in which some customers can be served by lockers. We thus consider the case with multiple capacitated vehicles, the vehicle routing problem with lockers (VRPL). Afterward, we consider the single and the multiple vehicle cases with time windows, the traveling salesman problem with locker and time windows (TSPLTW), and the vehicle routing problem with lockers and time windows (VRPLTW), respectively. The problem description follows this outline.

### 3.1 The single vehicle case: the TSPL

The TSPL is modeled on a digraph $G = (V, A)$. The set of vertices $V$ is partitioned as $V = \{0\} \cup V_C \cup V_L$, where 0 is the depot, $V_C = \{1, 2, \ldots, n\}$ is the set of the $n$ customers and $V_L = \{n + 1, n + 2, \ldots, n + m\}$ is the set of the $m$ available lockers. $A = \{(i, j) : i, j \in V\}$ is the given set of arcs, each of which is associated with a distance $d_{ij}$ and a traveling time $\tau_{ij}$.

The problem aims at delivering the parcels to the customers with a truck at minimum cost. The truck must leave from and return to the depot and it can serve the customers either directly at home (home service) or via a set of lockers (self-service). Each customer must be served, but both customers and lockers can be visited at most once by the truck. Each locker $b \in V_L$ is associated with a maximum capacity defined by the number of customers $v_b$, and with a radius $R_b$ beyond which it cannot serve any customer. Note that some customers may lay within the radius of more than one locker, but some may be located beyond the radius of all the lockers, and thus, no self-service is possible for them.

The objective function to be minimized is made of two components: (i) the truck transportation costs and (ii) the cost sustained because the customers must collect their parcels in a locker, each of which is multiplied by a coefficient: $\alpha_1$ and $\alpha_2$, respectively.

A feasible solution for the TSPL is presented in Fig. 1, in which the square is the depot, the circles are the customers, and the triangles are the lockers with their radius. The arrows represent the truck route, and the dashed arrows represent the customer’s path to collect their parcels at the lockers.

### 3.2 Extension to the multiple vehicle cases: the VRPL

In order to extend the TSPL to the case that includes multiple capacitated vehicles, the VRPL, we introduce a fleet composed of at most $n$ identical vehicles with capacity $Q$ that, starting and returning from the depot, deliver parcels by visiting lockers or customers. Each customer has a non-negative request $q_i$. Note that also for the VRPL the capacity of lockers is expressed by the number of customers and is not affected by their requests. All the other characteristics are inherited by the TSPL. Among these, note that each vertex but the depot can be visited at most once. This prevents a locker to be visited by more than one truck, that is a characteristic shared with other similar works in the literature, such as Mancini and Gansterer (2021).

### 3.3 Time windows: the TSPLTW and the VRPLTW

We extend the TSPL and the VRPL to include a time window $[e_i, l_i]$ for each customer $i \in V_C$ within which a customer must be served. Note that a truck can reach a customer $i$ before its earliest time window $e_i$, but it must wait until $e_i$ to serve it. The earliest visit time of the depot $e_0$ is set to the beginning of the time horizon 0 (if $e_0 \neq 0$ one can always shift all the time windows by $e_0$), while $l_0$ defines a maximum time that each route cannot exceed. Pickup points can have time windows $[e_b, l_b], b \in V_L$; however, no time window should be imposed to lockers, and for them we set $[e_b, l_b] = [e_0, l_0]$. Time windows constraints can be applied to both the TSPL and the VRPL, defining two new problems: the TSPLTW and the VRPLTW, respectively.

In Table 1, one can find the summary of all the symbols used in this paper.

### 4 Formulations

In this section, we first propose a set of different formulations for the VRPL and their adaptation to the VRPLTW. Afterward, we show their adaptation to the TSPL and the TSPLTW. A set of valid inequalities is introduced for all formulations.

#### 4.1 Formulations for the VRPL

In the following, we propose three formulations for the VRPL.
Table 1 Symbols used

| Sets       | Description                                      |
|------------|--------------------------------------------------|
| $V$        | Set of vertices, $V = \{0\} \cup V_C \cup V_L$  |
| $V_C$      | Set of customers, $V_C = \{1, 2, \ldots, n\}$   |
| $V_L$      | Set of lockers, $V_L = \{n + 1, n + 2, \ldots, n + m\}$ |
| $A$        | Set of arcs, $A = \{(i, j) : i, j \in V\}$       |

| Parameters | Description                                      |
|------------|--------------------------------------------------|
| $n$        | Number of customers                              |
| $m$        | Number of available lockers                     |
| $d_{ij}$   | Distance associated with each arc $(i, j) \in A$ |
| $\tau_{ij}$| Time needed to travel an arc $(i, j) \in A$       |
| $c_{ib}$   | Cost of serving customer $i \in V_C$ from locker $b \in V_L$ |
| $R_b$      | Radius of locker $b \in V_L$                    |
| $v_b$      | Capacity of locker $b \in V_L$                  |
| $Q$        | Vehicles capacity                                |
| $q_i$      | Request of customer $i \in V_C$                  |
| $[e_i, l_i]$| Time window for customer $i \in V_C$             |
| $\alpha_1$| Weight of the truck transportation costs         |
| $\alpha_2$| Weight of the self-service costs                 |

| Variables  | Description                                      |
|------------|--------------------------------------------------|
| $x_{ij}$   | Binary variable that equals 1 if a vehicle travels the arc $(i, j) \in A$; 0 otherwise |
| $y_{ib}$   | Binary variable that equals 1 if customer $i \in V_C$ is served via locker $b \in V_L$; 0 otherwise |
| $t_0$      | Non-negative variables defining the time in which the truck serving vertex $i \in V$ can leave it |
| $f_{ij}$   | Non-negative variables defining the commodity flow on arc $(i, j) \in A$ |
| $g_{ij}$   | Non-negative variables defining the residual capacity flow on the truck on arc $(i, j) \in A$ |

4.1.1 The vehicle flow formulation

The first formulation that we present is based on the well-known vehicle flow formulation for the VRP (see, e.g., Toth and Vigo 2002), we call it vehicle flow formulation (VFF), and it makes use of two sets of variables. Variables $x_{ij}$ equal 1 if a truck travels arc $(i, j) \in A$, 0 otherwise. Variables $y_{ib}$ equal 1 if the customer $i \in V_C$ travels to locker $b \in V_L$ to collect its parcel, 0 otherwise. Note that $x_{ii} = 0$, $i \in V$ and variables $y_{ib}$ are fixed to 0 if $d_{ib} > R_b$ for this formulation and those defined in the following.

$$
\min z = \alpha_1 \sum_{i \in V} \sum_{j \in V} d_{ij} x_{ij} + \alpha_2 \sum_{i \in V_C} \sum_{b \in V_L} c_{ib} y_{ib}
$$

$$
\sum_{j \in V} x_{ji} = \sum_{j \in V} x_{ij} \quad i \in V
$$

$$
\sum_{i \in V} x_{ij} + \sum_{b \in V_L} y_{jb} = 1 \quad j \in V_C
$$

$$
\sum_{j \in V} x_{jb} \leq 1 \quad b \in V_L
$$

$$
\sum_{j \in V} (x_{jb} + x_{bj}) \geq 2y_{ib} \quad i \in V_C, b \in V_L
$$

The objective function (1) minimizes two components, each one multiplied by its weight: the distance traveled by the truck and a penalty associated with self-service that depends on the customer and the locker used to serve it. Constraints (2) guarantee the flow conservation in every vertex, while constraints (3) impose that a customer is either served at home by a truck or at a locker. Constraints (4) impose that if a locker is not used, then no vehicle may visit the locker. Constraints (5) impose that if a locker is used, then one arc entering and one arc leaving the locker must be covered by a truck. Constraints (6) state that if no customer is served via a certain locker, then no truck should visit that locker.
Constraints (7) impose a maximum capacity to each locker. In (8), we adapted the generalized subtour elimination constraints (GSEC) to the VRPL. Variables $x$ and $y$ are defined binary in (9) and (10).

Inequalities A valid inequality imposing a lower bound on the number of vehicles that must leave the depot to serve all customers as in (11) can be added to strengthen the formulation. GSEC (8) are exponentially many and will be included only when violated in a branch-and-cut fashion (for details, see Sect. 5); however, one can statically include GSEC for all the arcs and the non-negativity constraint for the flow variables. This formulation differs from the traditional one-commodity formulation for the VRP because we need two separate flow conservation constraints for the customers subject to home delivery or self-service. Constraints (17) guarantee the flow conservation for each customer that is served directly at home, and thus, the request $q_i$ must be satisfied by this flow; if the customer is served with a locker, then no request is left by the truck. Constraints (18) guarantee the flow conservation for the lockers that must account for all the requests $q_i$ served by the considered locker.

Inequalities (11–14) can be included to strengthen OCF.

4.1.3 The two-commodity formulation

The third formulation that we present is based on the two-commodity formulation originally proposed for the CVRP by Baldacci et al. (2004). We call this formulation the two-commodity formulation (TCF). TCF inherits the variables $x$, $y$, and $f$ from OCF and uses a fourth variable $g_{ij}$ that gives the residual space (the empty space) on the vehicle traveling along arc $(i, j) \in A$, if any. Note that variables $f$ show the load flow along the routes traveled by vehicles from their start to their return to the depot, while variables $g$ show the residual flow in the opposite direction.

The TCF for the VRPL is given by (1–7), (9), (10), (15), (16), (19) and by the following constraints (20–25):

$$\sum_{i \in V_L} f_{i0} = 0 \quad (15)$$

$$\sum_{i \in V_L} f_{ib} = \sum_{i \in V_C} q_i \quad (16)$$

$$\sum_{j \in V_C} f_{ij} = \sum_{j \in V_C} f_{ji} + \sum_{i \in V_C} q_i y_{ib} \quad i \in V_C \quad (17)$$

$$\sum_{j \in V_L} f_{jb} = \sum_{j \in V_L} f_{bj} + \sum_{i \in V_C} q_i y_{ib} \quad b \in V_L \quad (18)$$

$$0 \leq f_{ij} \leq Q x_{ij} \quad i, j \in V \quad (19)$$

$$\sum_{i \in V_L} g_{i0} \leq n Q - \sum_{i \in V_C} q_i \quad (20)$$

$$\sum_{i \in V_L} g_{ib} \leq n Q \quad (21)$$

$$f_{ij} + g_{ji} = Q x_{ij} \quad (i, j) \in A \quad (22)$$

$$\sum_{j \in V} (f_{ji} - g_{ij}) = \sum_{j \in V} (f_{ij} - g_{ji}) + 2 q_i (1 - \sum_{b \in V_L} y_{ib}) \quad i \in V_C \quad (23)$$

$$\sum_{j \in V} (f_{jb} - g_{bj}) = \sum_{j \in V} (f_{bj} - g_{jb}) + 2 \sum_{i \in V_C} q_i y_{ib} \quad b \in V_L \quad (24)$$

$$0 \leq g_{ji} \leq Q x_{ji} \quad i, j \in V \quad (25)$$

Constraints (20) and (21) impose an upper bound on the amount of residual flow leaving and entering the depot, respectively. On each arc used by a truck, the sum of the load flow and the one of the residual flow must equal the capacity of the truck as per constraints (22). Constraints (23) impose that the difference between the load flow and residual
capacity flow that enter and leave a vertex is the double of the request if the customer is served directly. Constraints (24) guarantee the same flow conservation for the lockers. Non-negativity and a maximum capacity are imposed to variables \( g \) in (25).

Inequalities (11–14) can be included to strengthen the TCF for the VRPL.

### 4.2 Formulations for the VRPLTW

In the following, we define the components needed to include time windows in the previously presented formulations and thus to model the VRPLTW.

#### 4.2.1 The vehicle flow formulation

A non-negative continuous variable \( t_i \) is used to represent the time at which the truck visiting the vertex \( i \in V \setminus \{0\} \), that can be either a customer or a locker, can leave. \( t_0 \) represents the moment at which the last truck returns to the depot, since each shift begins at the depot at time \( e_0 \). Note that \( t \) does not model the arrival time at a customer vertex because the truck is allowed to arrive early and wait the beginning of its time window. The following constraints, added to (1–10), model the VRPLTW and represent its VFF.

\[
\begin{align*}
t_j & \geq t_i + \tau_{ij} - Z_{ij}(1-x_{ij}) & i & \in V_C \cup V_L, j & \in V \quad (26) \\
t_j & \geq e_0 + \tau_{0j} - Z_{0j}(1-x_{0j}) & j & \in V \quad (27) \\
\max(e_i, e_0 + \tau_{0i}) & \leq t_i \leq \min(l_i, l_0 - \tau_{0i}) & i & \in V_C \quad (28) \\
b_i & \leq t_b \leq \min(l_b, l_0 - \tau_{0b}) & b & \in V_L \quad (29) \\
t_i & \geq 0 & i & \in V \quad (30)
\end{align*}
\]

Constraints (26) update the time variables when an arc \((i, j) \in A, i \neq 0 \) is traveled and constraints (27) update the same variables when the arcs start from the depot. In all these constraints, \( Z_{ij} \) is a sufficiently large number that we compute as \( l_i + \tau_{ij} - e_j, (i, j) \in A \). Constraints (28) and (29) strengthen the classical time window constraints \( e_i \leq t_i \leq l_i \) for the customers and the lockers, respectively. In (30), the time variables are defined non-negative.

**Inequalities based on time windows** In the following, we write some valid inequalities to tighten or easily impose time windows in a VRPLTW.

\[
\begin{align*}
x_{ij} & = 0 & i, j & \in V_C \cup \{0\} : e_i + \tau_{ij} > l_j \quad (31) \\
x_{ij} + \sum_{k \in H_{ij}} x_{jk} & \leq 1 & i & \in V, j & \in V \setminus \{0\}, i \neq j \quad (32) \\
\sum_{k \in H_{ij}} x_{ki} + x_{ij} & \leq 1 & i & \in V \setminus \{0\}, j & \in V, i \neq j \quad (33)
\end{align*}
\]

Constraints (31) fix to zero all the arc variables starting from \( i \) arriving in \( j \) such that the minimum arrival time in \( j \) is larger than \( l_i \). Let us define for each couple of vertices \((i, j), i \in V, j \in V \setminus \{0\}, i \neq j \) the set \( H_{ij} = \{ k \in V : \max(e_j + \tau_{ij}, e_j) + \tau_{jk} > l_k \} \) that includes all the vertices \( k \) for which a path \((i, j, k)\) is infeasible due to the time windows. Let us also define for each couple of vertices \((i, j), i \in V \setminus \{0\}, j \in V, i \neq j \) the set \( H_{ij} = \{ k \in V : \max(e_k + \tau_{ki}, e_i) + \tau_{ij} > l_j \} \) that includes all the vertices \( k \) for which a path \((k, i, j)\) is infeasible due to the time windows. Then, inequalities (32) and (33) are valid for the VRPLTW.

\[
\begin{align*}
t_b & \leq (l_i - \tau_{bi} - Z_{bi}) x_{bi} + Z_{bi} & i & \in V_C, b & \in V_L \quad (34) \\
t_b & \geq (e_i + \tau_{ib}) x_{ib} & i & \in V_C, b & \in V_L \quad (35)
\end{align*}
\]

If a customer \( i \) is visited just after locker \( b \) on a route, then the time variable at the locker should be bounded so as to arrive before \( l_i \). This is expressed in inequalities (34). On the other hand, if a locker \( b \) is visited after vertex \( i \), then the time variable in \( b \) cannot be smaller than the time needed to travel from \( i \) to \( b \) plus the earliest time window of \( i \), \( e_i \). This is imposed by inequalities (35). Note that \( Z_{bi} = \min(l_{bi}, l_{b} - \tau_{bi}), b \in V_L \) is a sufficiently large number.

#### 4.2.2 The one-commodity formulation

The OCF for the VRPLTW can be obtained with (1–7), (9), (10), (15–19) and the time window constraints (26–30). Inequalities (11–14) and (31–35) also apply.

#### 4.2.3 The two-commodity formulation

The TCF for the VRPLTW can be obtained with (1–7), (9), (10), (15), (16), (19–25) and the time window constraints (26–30). Inequalities (11–14) and (31–35) also apply.

### 4.3 Adaptation to the TSPL

Previously presented formulations can be adapted to solve the TSPL and the TSPLTW. Let us define constraints (36) that impose the number of vehicle exiting and entering the depot to one and constraints (37) that represent the GSEC specific to the TSPL.

\[
\begin{align*}
\sum_{j \in V} x_{0j} & = \sum_{j \in V} x_{j0} = 1 \quad (36) \\
\sum_{i \in S} \sum_{j \in \mathcal{V}_C} \sum_{b \in \mathcal{V}_L} y_{ib} & \leq |S| - 1 \quad (37) \\
S & \subseteq V_C \cup V_L, S \neq \emptyset
\end{align*}
\]

The VFF for the TSPLTW can then be written as (1–7), (9), (10), (36), (37).
To adapt the OCF to solve the TSPL, constraints (16–19) must be updated as follows:

$$\sum_{i \in V_C \cup V_L} f_{0i} = n$$

(38)

$$\sum_{j \in V} f_{ji} = \sum_{j \in V} f_{ij} + \left( 1 - \sum_{b \in V_L} y_{ib} \right) \quad i \in V_C$$

(39)

$$\sum_{j \in V} f_{jb} = \sum_{j \in V} f_{bj} + \sum_{i \in V_C} y_{ib} \quad b \in V_L$$

(40)

$$0 \leq f_{ij} \leq nx_{ij} \quad i, j \in V$$

(41)

The OCF for the TSPL thus includes (1–7), (9), (10), (15), (36), (38–41).

The TCF can be adapted to solve the TSPL by using (1–7), (9), (10), (15), (36), (38), (41), and the following constraints (42–47).

$$\sum_{i \in V_C \cup V_L} g_{i0} = 0$$

(42)

$$\sum_{i \in V_C \cup V_L} g_{0i} = n$$

(43)

$$f_{ij} + g_{ji} = nx_{ij} \quad (i, j) \in A$$

(44)

$$\sum_{j \in V} (f_{ji} - g_{ij})$$

$$= \sum_{j \in V} (f_{ij} - g_{ji}) + 2 \left( 1 - \sum_{b \in V_L} y_{ib} \right) \quad i \in V_C$$

(45)

$$\sum_{j \in V} (f_{jb} - g_{bj})$$

$$= \sum_{j \in V} (f_{bj} - g_{jb}) + 2 \sum_{i \in V_C} y_{ib} \quad b \in V_L$$

(46)

$$0 \leq g_{ji} \leq nx_{ij} \quad i, j \in V$$

(47)

Inequalities (13), (14) can be added to strengthen all the formulations.

### 4.4 Adaptation to the TSPLTW

The VFF, the OCF, and the TCF for the TSPL can be adapted to solve the TSPLTW by including constraints (26–30). Inequalities (13), (14) and (31–35) also apply.

### 5 Branch-and-cut implementation

The VFF formulations include an exponential number of constraints, and hence, we solve them with a Branch-and-Cut (B&C) algorithm. Formulations OCF and TCF do not need a dynamic insertion of constraints; even if the introduction of the exponential constraints used in VFF and separated on fractional solutions is still possible, computational evidence showed that their use in a B&C fashion is not profitable. We use the B&C framework of CPLEX 20.10.0, which solves at every node of the enumeration tree the linear relaxation of a MILP model and then, invokes user developed separation procedures to possibly add cuts. Preliminary tests suggested to adopt strong branching as branching rule. At the root node of the B&C procedure, we obtain an initial solution by using a simple greedy heuristic. We first build a TSP or a VRP solution (ignoring the lockers) by using the well-known Nearest Neighborhood (NN) (see, e.g., Toth and Vigo 2002) and the C&W algorithm (see, e.g., Clarke and Wright 1964). We adapted the NN to solve the VRP by starting a new truck route when the maximum capacity of the current truck was reached, or no new customer could be included. The C&W algorithm is applied to solve the TSP without fixing the truck capacity until only one route remains. Moreover, we adapted NN and C&W to provide solutions to problems with time window constraints. For the NN, we check the vertices that can be visited without exceeding their latest time window, at each iteration. In case, the arrival at those vertices precedes their earliest time window, the truck needs to wait the so-called waiting time needed to respect them. The following vertex selected to be visited in the truck sequence minimizes both the distance traveled by the truck and the waiting time. For the C&W, we considered the time window constraints by performing only merging moves that allow respecting them. As in the NN, we have considered both the traveled distance and the waiting time in the evaluating function. Once the solutions of the NN and the C&W (with or without time windows) are obtained, lockers are included in two ways: by performing an assignment based on customers or an assignment based on lockers. In the first case, the algorithm tries to move a customer from the truck route to a locker so that the entire solution remains feasible and the objective function is minimized. A new customer move is checked at each iteration. In the second case, for each unused locker, the algorithm tries to find the smallest set of customers such that their assignment to the locker would improve the objective function. Among these feasible moves, we chose those using the locker that provides the larger improvement in the objective function and among these the one with a smaller number of customers. If possible, we reiterate. Among the possible solutions, we select the one with the smallest value to pass to the corresponding exact algorithm.

Note that it was not always possible to obtain a feasible initial solution for the TSPLTW because of the limitation of having one vehicle and the time window constraints. This is not the case for the TSPL, VRPL, and VRPLTW, for which an initial greedy solution was always at hand. For the TSPL, it is always possible to find a feasible initial solution because any solution that visits all the customers with one truck is
feasible, while for the VRPL and the VRPLTW it is always possible to find a feasible initial solution because the fleet is assumed to be as large as the number of customers in such problems.

5.1 Separation procedures

The aim of this section is to present the procedures that we use to determine if the valid inequalities that we proposed are violated by a given possibly fractional solution \((\bar{x}, \bar{y})\).

5.1.1 Separation S1

To separate GSEC (37), we first build a supporting graph \(\overline{G} = (V, \overline{A})\), where \(\overline{A} = \{(i, j) \in A : \overline{x}_{ij} > 0\} \cup \{(i, j) \in A : i \in V_C, j \in V_L, \overline{y}_{ij} > 0\}\), and a capacity \(\overline{x}_{ij} + \overline{y}_{ij}\) is assigned to arc \((i, j) \in \overline{A}\).

\(O(n)\) max-flows are computed using the depot as a source and any vertex \(i \in V_C\), in turn, as a sink. If the max-flow value obtained is lower than one, then \(i\) is disconnected from the depot, and hence, the cut that corresponds to the set \(S\) induced by the min-cut may be added to the model. Moreover, for the VRP case, we obtain a simple algorithmic improvement by checking, for any set \(S\) induced by a min-cut, if the minimum number of vehicles required to serve \(S\) is lower than the max-flow value obtained, and, in such a case, we add the corresponding (stronger) constraint (8). In this way, the exact separation of the subtour elimination constraints (37) is used as a heuristic for separating (8).

We refer to this procedure as separation S1.

5.1.2 Separation S2

To separate exactly constraints (8), we start by adding an auxiliary vertex \(a\) to graph \(\overline{G}\) obtaining a new supporting graph \(G'\) in which a set of arcs \((a, i)\) links the vertex \(a\) to every vertex \(i \in V_C\), with capacity \(q_i/\overline{Q}\). We thus compute the max-flow on \(G'\) with vertex \(a\) as the source and the depot as the sink. The constraint (8) corresponding to the set \(S\) induced by the min-cut is then checked, and, if violated, it is added to the model.

We refer to this procedure as separation S2.

In Table 2, we provide a summary of the presented formulations to solve the proposed problems. Note that, for each cell of the table, the first set of constraints is the one needed to obtain a feasible solution for the problem, while the set of constraints after a + includes the non-necessary inequalities that improve formulations performances. These additional inequalities are used to speed up each formulation, except for constraints (14), which are not included in VFF.

The described separation procedures are invoked at every node of the enumeration tree in case of integer solutions and at every node with depth not larger than 8 for fractional solutions, basing our decision on preliminary tests. S1 is used to separate constraints (37) in formulations VFF for the TSPL and TSPLTW. S1 and S2 are used to separate constraints (37) and (8) in formulations VFF for the VRPL and VRPLTW. On the basis of computational evidence on the instances that we tested, we stop the separation process as soon as we find and add a violated cut, if any. The order of the separation procedures, when both are used, follows the one in which they are presented, because S1 can separate exactly constraints (37) and heuristically constraints (8), and thus, separation S2 is invoked only in case no violated cut is found by S1, to separate constraints (8) exactly.

6 Computational results

In this section, we show the computational results obtained by running the proposed methods on a single thread of CPU Intel i7-6850K @ 3.60 GHz with 64 GB RAM. All our methods have been implemented in C++ and compiled with gcc 7.5.0. As already mentioned, the used solver is CPLEX 20.10.0.

6.1 Benchmark instances

Routing problems with lockers are very recent, and no standard set of benchmark instances has been established yet. We make use of the instances used by Jiang et al. (2020), who adapted 40 benchmark instances for the TSP with time windows to the TSPLTW by including a set of locker vertices to the graph. In particular, the original instances are divided into two sets: the first set with time windows of amplitude of 20 was initially proposed by Dumas et al. (1995); the second set with time windows of 100 was proposed by Gendreau et al. (1998) and Dumas et al. (1995). Each of these sets can be divided into four groups of five instances each, with the same number of customers \(n = 20, 40, 60, 100\). The number of added lockers is \(m = \lfloor n/10 \rfloor\). Each instance is modified to include \(m\) lockers by adding the first \(m\) vertices of the following set of coordinates: \((25.0, 25.0), (12.5, 12.5), (37.5, 37.5), (12.5, 25.0), (37.5, 12.5), (12.5, 25.0), (25.0, 37.5), (37.5, 25.0), (25.0, 12.5), (0, 0)\). Note that the instances include \(n + m + 1\) vertices. The distance \(d_{ij}\) is computed as an Euclidean distance between each couple of vertices \((i, j) \in A\), rounded to the second decimal. The distance matrix is corrected so as to respect the triangular inequality, as in Jiang et al. (2020). The following characteristics are also derived from the instances proposed by Jiang et al. (2020). The traveling time is set to equal the distance: \(t_{ij} = d_{ij}, (i, j) \in A\). The cost of serving customers via lockers also depends on the distance:
The best-known upper bound of each instance and VRPLTW, respectively. The results are displayed by instance size (n) and time window size (tw). For each formulation, we show the percentage gap (computed as explained above), the seconds needed to solve the instance to optimality or to reach the time limit of 2 hours, and the number of obtained optima. Note that the results are averaged on those instances for which an upper bound was available, that is not always the case for the TSPLTW, because small time windows and the number of vehicles limited to one makes it hard to find feasible solutions, if any; indeed, we suspect that few instances do not have one.

6.2 TSPL and VRPL

In Tables 3 and 4, we show the results obtained by running the proposed formulations to solve the TSPL and the VRPL, respectively. We grouped the results per instance size (number of customers) and for each formulation, we report the percentage gap (computed as 100 \( \cdot \) (UB - LB)/UB, being UB the best-known upper bound of each instance and LB the lower bound obtained at the end of the iterations), the seconds needed to reach the optimal solution or the time limit of 2 hours, and the number of optima.

The formulation providing the best results when solving the TSPL is OCF: it could solve almost all the instances to the proven optimality in less than 5 minutes, on average. TCF and VFF could solve, respectively, 34 and 32 out of 40 instances with very small gaps but with larger computing times with respect to OCF, on average. However, for the smaller instances, VFF is the fastest method among all.

The proposed formulations provide good results when solving the VRPL, showing small optimality gaps also for the largest instances; however, only OCF could solve one instance with \( n = 100 \) to optimality within the time limit. OCF is the best performing of our formulations, being able to solve 19 out of 40 instances and showing an average %gap of 1.64. Note that TCF shows better results than VFF, even if VFF is, once more, quicker in solving the smaller instances.

6.3 TSPLTW and VRPLTW

In Tables 5 and 6, we show the results for the TSPLTW and the VRPLTW, respectively. The results are displayed by instance

\[ c_{ib} = d_{ib}, i \in V_C, b \in V_L. \]  
For all instances, the parameters are set as follows: \( \alpha_1 = 1, \alpha_2 = 0.5, \) and \( R_b = 20, v_b = 5, \) \( b \in V_L. \) We extended time windows to the lockers imposing \( [e_b, l_b] = [e_l, l_l], b \in V_L. \) Note that by imposing tighter time windows one can model counters instead of lockers. For the multiple vehicle case, we considered \( q_i = 1, i \in V_C \) and a capacity \( Q = \lceil n/2 \rceil. \)

### Table 2 Summary of the presented formulations for the described problems

|        | VFF                  | OCF                  | TCF                  |
|--------|----------------------|----------------------|----------------------|
| VRPL   | (1–10), S1, S2+ (11–13) | (1–7), (9), (10), (15–19) + (11–14) | (1–7), (9), (10), (15), (16), (19–25) + (11–14) |
| VRPLTW | (1–10), (26–30), S1, S2 + (11–13), (31–35) | (1–7), (9), (10), (15–19), (26–30), + (11–14), (31–35) | (1–7), (9), (10), (15), (16), (19–30) + (11–14), (31–35) |
| TSPL   | (1–7), (9), (10), (36), (37), S1 + (13) | (1–7), (9), (10), (36), (38–41) + (13), (14) | (1–7), (9), (10), (36), (38), (41–47) + (13), (14) |
| TSPLTW | (1–7), (9), (10), (36), (37), (26–30), S1 + (13), (31–35) | (1–7), (9), (10), (36), (38–41), (26–30), + (13), (14), (31–35) | (1–7), (9), (10), (36), (38), (41–47), (26–30), + (13), (14), (31–35) |

For each instance and problem, we show the needed constraints and separation procedures. After the + sign, we report the valid inequalities.
Table 3 Results for the TSPL

| n   | m   | VFF %gap | Sec | opt | OCF %gap | Sec | opt | TCF %gap | Sec | opt |
|-----|-----|----------|-----|-----|----------|-----|-----|----------|-----|-----|
| 20  | 2   | 0.00     | 0.15| 10  | 0.00     | 0.53| 10  | 0.00     | 0.87| 10  |
| 40  | 4   | 0.00     | 204.9| 10  | 0.00     | 12.25| 10  | 0.00     | 61.28| 10  |
| 60  | 6   | 0.00     | 513.34| 10  | 0.00     | 48.92| 10  | 0.00     | 150.55| 10  |
| 100 | 10  | 1.44     | 6856.91| 2   | 0.06     | 1120.15| 9   | 0.47     | 5180.10| 4   |
| Avg/sum | 0.36 | 1893.22 | 32  | 0.02 | 295.46 | 39  | 0.12 | 1348.20 | 34  |

Table 4 Results for the VRPL

| n   | m   | VFF %gap | Sec | opt | OCF %gap | Sec | opt | TCF %gap | Sec | opt |
|-----|-----|----------|-----|-----|----------|-----|-----|----------|-----|-----|
| 20  | 2   | 0.00     | 1.96| 10  | 0.00     | 9.84 | 10  | 0.00     | 28.71| 10  |
| 40  | 4   | 4.89     | 5805.32| 2   | 1.30     | 4639.26| 6   | 2.47     | 5589.79| 4   |
| 60  | 6   | 7.90     | 7200.00| 0   | 2.12     | 6187.86| 2   | 3.12     | 6690.03| 1   |
| 100 | 10  | 8.29     | 7200.00| 0   | 3.16     | 6867.47| 1   | 5.28     | 7200.00| 0   |
| Avg/sum | 5.27 | 5051.82 | 12  | 1.64 | 4426.11| 19  | 2.72 | 4877.13| 15  |

Table 5 Results for the TSPLTW

| n   | m   | tw | VFF %gap | Sec | opt | OCF %gap | Sec | opt | TCF %gap | Sec | opt |
|-----|-----|----|----------|-----|-----|----------|-----|-----|----------|-----|-----|
| 20  | 2   | 20 | 0.00     | 0.24| 5   | 0.00     | 0.55| 5   | 0.00     | 0.66| 5   |
| 100 | 0.00| 4  | 0.00     | 4.83| 5   | 0.00     | 13.84| 5   | 0.00     | 18.21| 5   |
| Avg/sum | 0.00 | 2.54 | 10  | 0.00 | 7.19 | 10  | 0.00 | 9.44 | 10  |
| 40  | 20  | 0.25| 1505.60| 4   | 0.00 | 222.74| 5   | 0.71     | 1582.50| 4   |
| 100 | 2.34| 3037.53| 3   | 1.03     | 2343.65| 4   | 2.72     | 3147.31| 3   |
| Avg/sum | 1.30 | 2271.56 | 7   | 0.52     | 1283.19| 9   | 1.71     | 2364.90| 7   |
| 60  | 1.33| 3006.29| 3   | 0.95     | 2976.33| 3   | 1.48     | 3062.90| 3   |
| 100 | 7.49| 7200.00| 0   | 5.83     | 7200.00| 0   | 6.74     | 7200.00| 0   |
| Avg/sum | 4.41 | 5103.14 | 3   | 3.39     | 5088.16| 3   | 4.11     | 5131.45| 3   |
| 100 | 8.97| 7200.00| 0   | 7.89     | 7200.00| 0   | 8.90     | 7200.00| 0   |
| Avg/sum | 12.12| 7200.00 | 0   | 10.35     | 7200.00| 0   | 11.51     | 7200.00| 0   |
| 100 | 16.06| 7200.00| 0   | 14.78     | 7200.00| 0   | 14.78     | 7200.00| 0   |
| Avg/sum | 12.12| 7200.00 | 0   | 10.35     | 7200.00| 0   | 11.51     | 7200.00| 0   |

is imposed by constraints (48), which state that the time in which the customer \( i \) is served at a locker \( b \) is at least the time \( t_b \) in which the truck leaves the locker plus the time needed by the customer to reach the locker \( \tau_{ib} \).

\[
t_i \geq t_b + \tau_{ib} - Z_{ib}(1 - y_{ib}) \quad i \in V_C, b \in V_L
\]  

Moreover, their objective function computes locker setup costs that are proportional to the number of customer served by a locker: for each customer served via locker \( b \in V_L \) a penalty \( \pi_b \) must be paid. The authors solve the relaxation of the problem which they propose by including soft time windows constraints, i.e., allowing the latest time window to be exceeded at a cost of a penalty. In order to keep track of the violation of time windows, we introduced a new non-negative variable \( s_i, i \in V \). The new objective function is thus (49), and constraints (50) and (51) must be imposed to guarantee that \( s_i \) represents the exceeded time in each vertex, if any. Note that the two new components of the objective function are multiplied by two coefficients \( \alpha_3 = 5 \) and \( \alpha_4 = 1 \). The setup cost \( \pi_b = 1 \) for all \( b \in V_L \).

\[
\min z' = \alpha_1 \sum_{i \in V} \sum_{j \in V} d_{ij} x_{ij} + \alpha_2 \sum_{i \in V_C} \sum_{b \in V_L} c_{ib} y_{ib}
+ \alpha_3 \sum_{i \in V} \sum_{b \in V_L} \pi_{ib} y_{ib} + \alpha_4 \sum_{i \in V} s_i
\]

\[
s_i \geq t_i - l_i \quad i \in V
\]

\[
s_i \geq 0 \quad i \in V
\]
In addition, some time window constraints must be modified. Constraints (26)–(48) can be inherited by changing their ‘big-M’ value to a large enough number $Z$. One such large number is the value of the initial heuristic upper bound provided to the solver. Inequalities (35) can be inherited as they are while constraints (28) and (29) must be changed as follows:

$$\max(e_i, e_0 + \tau_0) \leq t_i \leq Z \quad i \in V$$

The following inequalities, aimed at bounding from above the value of variables $s$, can also be added:

$$s_i \leq Z \quad i \in V$$

$$s_b \leq Z \sum_{i \in V_C} y_{ib} \quad b \in V_L$$

Constraints (53) provide an upper bound on each variable $s_i$, $i \in V$, while constraints (54) bound variables $s_b$ to a large value for each locker used, while it sets the same variable to 0 if the locker is not used. Note that a valid large number $Z$ is, once more, the value of the initial heuristic upper bound on $z'$.

Despite the described modification, it was not possible to obtain the same results showed by Jiang et al. (2020), so we implemented their formulation, which we call MTZ in the following, to provide a comparison with our methods. The results are shown in Table 7 in which we display the instance size $n$, the time window size (tw), and, for each formulation, the percentage gap, the seconds needed to get to the end of iterations (proven optimality or the time limit of 2 hours), and the number of optima.

One can see that all the proposed formulations provide better results than MTZ, being able to solve to the proven optimality 18 instances compared to the 14 instances solved by MTZ. In particular, all instances of size 20 and 8 out of 10 of size 40 could be solved to optimality with our methods. VFF is the formulation with shorter times on average and the one that presents the best $\%$ gap for the instances with up to $n = 60$ and time windows of value 20. From $n = 60$, tw = 100, the best method is OCF. The percentage gaps shown for instances with $n = 100$ are large also because no very good quality solutions could be found.

### 6.4.2 Comparison with Mancini and Gansterer

Mancini and Gansterer (2021) propose the vehicle routing with private and shared delivery locations and solve it both exactly and heuristically. We are interested in their exact results. The problem that they solve is similar to the VRPLTW with some modifications. First of all, the objective function includes one more term that aims at minimizing the number of trucks used (multiplied by coefficient $\alpha_5$).

The new objective function can thus be rewritten as in (55).

$$\min z'' = \alpha_1 \sum_{i \in V} \sum_{j \in V} d_{ij} x_{ij} + \alpha_2 \sum_{i \in V_C} \sum_{b \in V_L} c_{ib} y_{ib} + \alpha_5 \sum_{j \in V_C \cup V_L} x_{0j}$$

The values of the used parameters are: $\alpha_1 = 1$, $\alpha_2 = 1$, $\alpha_5 = 1$, $c_{ib} = 5$, $i \in V_C$, $b \in V_L$.

Another modification is needed because Mancini and Gansterer consider a fixed service time $\tau_1 = 5$ for serving a customer $i \in V_C$ and an una tantum service time $\tau_b = 10$ when visiting a locker $b \in V_L$. To include that, we can update the service times as follows: $\tau_i \leftarrow \tau_i + \tau_1$, $i \in V$, $j \in V_C$ and $\tau_{ib} \leftarrow \tau_{ib} + \tau_b$, $i \in V, b \in V_L$. Another needed
Table 7 Comparison with Jiang et al.

| n  | m  | tw | MTZ %gap | sec     | #opt | VFF %gap | Sec     | #opt | OCF %gap | Sec     | #opt | TCF %gap | Sec     | #opt |
|----|----|----|---------|---------|------|---------|---------|------|---------|---------|------|---------|---------|------|
| 20 | 2  | 20 | 0.00    | 248.56  | 5    | 0.00    | 0.60    | 5    | 0.00    | 11.87   | 5    | 0.00    | 8.34    | 5    |
| 100| 0.00|648.08|5     |0.00    |29.47   |5    |0.00    |41.94   |5    |0.00    |76.08   |5    |
| Avg/sum | | | 0.00 | 448.32 | 10 | 0.00 | 15.03 | 10 | 0.00 | 26.90 | 10 | 0.00 | 42.21 | 10 |
| 40 | 4  | 20 | 3.25    | 5877.42 | 2   | 0.81    | 1745.03 | 4   | 1.73    | 2368.35 | 4   | 1.44    | 2478.08 | 4   |
| 100| 3.41|4888.22|2     |0.98    |2626.71 |4    |1.44    |3100.35 |4    |1.79    |3341.27 |4    |
| Avg/sum | | | 3.33 | 5382.82 | 4  | 0.89    | 2185.87 | 8  | 1.58    | 2734.35 | 8  | 1.62    | 2909.67 | 8  |
| 60 | 6  | 20 | 11.20   | 7200.00 | 0   | 5.25    | 7200.00 | 0   | 5.94    | 7200.00 | 0   | 6.32    | 7200.00 | 0   |
| 100| 14.23|7200.00|0     |8.75    |7200.00 |0    |8.49    |7200.00 |0    |8.95    |7200.00 |0    |
| Avg/sum | | | 12.71 | 7200.00 | 0  | 7.00    | 7200.00 | 0  | 7.22    | 7200.00 | 0  | 7.64    | 7200.00 | 0  |
| 100| 20.51|7200.00|0     |16.76   |7200.00 |0    |15.39   |7200.00 |0    |15.50   |7200.00 |0    |
| 100| 30.70|7200.00|0     |27.81   |7200.00 |0    |26.80   |7200.00 |0    |27.32   |7200.00 |0    |
| Avg/sum | | | 25.60 | 7200.00 | 0  | 22.28   | 7200.00 | 0  | 21.09   | 7200.00 | 0  | 21.41   | 7200.00 | 0  |
| Avg/sum | | | 10.41 | 5057.79 | 14 | 7.54    | 4150.23 | 18 | 7.47    | 4290.31 | 18 | 7.67    | 4337.97 | 18 |

modification is about the truck capacity: the authors use an unlimited set of uncapacitated trucks, and thus, we can set $Q = n$ and $q_i = 1, i \in V_C$. Mancini and Gansterer propose three sets of 10 instances available online (Mancini and Gansterer 2020) with 5 lockers and 25, 50, and 75 customers, respectively, randomly generated in a 10*10 km area. The maximum travel time for each truck is 12 h (thus $l_0 = 720$), and the time window of a customer can be one of the 12 hours. The travel time is equal to the travel distance that is computed as 3 times the Euclidean distance between two points. In our adaptation, the travel times will also include the service times as explained previously. All customers can be served either with a truck or via a nearby locker; indeed, only customers within a travel time radius of 15 can be served by that locker ($R_b = 15, b \in V_L$). Note that we must fix variables $y_{ib} = 0$ if $\tau_{ib} > R_b$, service times excluded.

Table 8 reports the results of the MILP proposed by Mancini and Gansterer (MG) and the results of our methods adapted to solve their version of the VRPLTW on their set of instances. We show the number of customers, the number of lockers, and the name of the instances. Moreover, we report the best-known solution (BKS) for each instance. The BKS is bold when it is the proven optimum and in italic when the optimum is shown for the first time in this paper. We also show, for each method, the percentage gap between the BKS and the lower bound at the end of the iterations, the seconds needed to prove optimality if the time limit of 1 hour is not reached, and the number of optima for each set of instances. Note that the %gap of MG is updated (improved) with respect to what is shown in Mancini and Gansterer (2021), because we used the improved BKS values.

Our methods could provide a larger number of proven optima with respect to MG and most of the times with shorter times. In particular, OCF can solve 28 out of 30 instances, obtaining two more 50 customer instances optima and eight new 75 customer instances optima. This is reflected by the average gap of our best formulation, OCF, which is 0.15, with respect to 3.74 of MG. Note that for smaller instances (25 customers) VFF is the fastest method.

6.5 Results analysis

In this section, we compare the solutions of the problems previously presented to inspect the impact of the lockers and time windows on the solutions and their costs.

In Figure 2, we compare the value of the solutions of different TSP-based problems on the same instances. We display all values in boxplots, in which the lower and the upper bounds are the first and the third quartile, the middle line shows the median, and the diamond shows the average.

The first comparison we perform is between the classical TSP and the TSPL; the %difference of the costs is computed as $100 \cdot \frac{(c_{TSP} - c_{TSPL})}{c_{TSP}}$. One can see in Fig. 2a that the introduction of lockers can improve the cost of the solutions, because it can simplify the path of the truck, by at most about 10% on the used instances, but on average the improvements on the total cost are between 3.5 and 7.2%, approximately. Moreover, this value seems to decrease with the size of the instance. On the other hand, the routing costs in the TSP are much higher than in the TSPL, with up to 30% difference, and this is due to the fact that more customers are served with lockers and that the routing cost is not the only one considered in the TSPL.

In Figure 2b, we show the percentage difference of the classical TSP with time windows (TSPTW) and of the costs of the TSPLTW computed as $100 \cdot \frac{(c_{TSP} - c_{TSPL})}{c_{TSP}}$. One can see in Fig. 2b that the introduction of lockers can improve the cost of the solutions, because it can simplify the path of the truck, by at most about 10% on the used instances, but on average the improvements on the total cost are between 3.5 and 7.2%, approximately. Moreover, this value seems to decrease with the size of the instance. On the other hand, the routing costs in the TSP are much higher than in the TSPL, with up to 30% difference, and this is due to the fact that more customers are served with lockers and that the routing cost is not the only one considered in the TSPL.
Table 8 Comparison with Mancini and Gansterer. The BKS are in bold when they are the proven optimum, while they are in italic when the proven optimum is shown for the first time.

| n  | m  | Inst. | BKS | MG %gap | Sec | #opt | VFF %gap | Sec | #opt | OCF %gap | Sec | #opt | TCF %gap | Sec | #opt |
|----|----|-------|-----|---------|-----|------|---------|-----|------|---------|-----|------|---------|-----|------|
| 25 | 5  | 1     | 161.37 | 0.00 | 1.60 | 0.00 | 0.48 | 0.00 | 1.36 | 0.00 | 2.07 |
|    |    | 2     | 166.63 | 0.00 | 127.70 | 0.00 | 23.45 | 0.00 | 100.44 | 0.00 | 94.30 |
|    |    | 3     | 146.56 | 0.00 | 11.20 | 0.00 | 4.97 | 0.00 | 1.04 | 0.00 | 2.97 |
|    |    | 4     | 161.04 | 0.00 | 36.16 | 0.00 | 9.56 | 0.00 | 15.57 | 0.00 | 72.66 |
|    |    | 5     | 157.95 | 0.00 | 2.40 | 0.00 | 6.22 | 0.00 | 9.56 | 0.00 | 10.31 |
|    |    | 6     | 160.83 | 0.00 | 4.90 | 0.00 | 23.45 | 0.00 | 100.44 | 0.00 | 94.30 |
|    |    | 7     | 152.69 | 0.00 | 9.12 | 0.00 | 1.33 | 0.00 | 2.97 | 0.00 | 7.79 |
|    |    | 8     | 165.16 | 0.00 | 59.36 | 0.00 | 9.56 | 0.00 | 10.31 | 0.00 | 7.79 |
|    |    | 9     | 151.54 | 0.00 | 4.34 | 0.00 | 1.36 | 0.00 | 4.79 | 0.00 | 4.79 |
|    |    | 10    | 151.99 | 0.00 | 5.16 | 0.00 | 3.14 | 0.00 | 7.67 | 0.00 | 7.67 |
| Avg/sum | | | 0.00 | 26.19 | 10 | 0.00 | 9.85 | 10 | 0.00 | 20.73 | 10 | 0.00 | 31.44 | 10 |

| 50 | 5  | 1     | 266.80 | 0.00 | 1112.00 | 0.00 | 490.18 | 0.00 | 276.22 | 0.00 | 418.06 |
|    |    | 2     | 267.97 | 0.00 | 1465.00 | 0.00 | 27.63 | 0.00 | 71.26 | 0.00 | 120.78 |
|    |    | 3     | 273.05 | 0.00 | 3600.00 | 1.40 | 3600.00 | 0.00 | 291.23 | 0.00 | 1241.90 |
|    |    | 4     | 268.89 | 0.00 | 526.00 | 0.00 | 9.21 | 0.00 | 11.78 | 0.00 | 22.47 |
|    |    | 5     | 271.32 | 0.00 | 3439.00 | 2.05 | 3600.00 | 0.00 | 212.38 | 0.00 | 1250.41 |
|    |    | 6     | 268.32 | 0.00 | 1592.00 | 0.00 | 23.51 | 0.00 | 83.78 | 0.00 | 52.71 |
|    |    | 7     | 253.18 | 0.00 | 480.00 | 0.00 | 15.12 | 0.00 | 29.28 | 0.00 | 20.83 |
|    |    | 8     | 266.68 | 0.00 | 3114.00 | 0.00 | 47.91 | 0.00 | 231.80 | 0.00 | 274.17 |
|    |    | 9     | 267.56 | 0.00 | 3600.00 | 0.00 | 3333.15 | 0.00 | 458.21 | 0.00 | 3227.73 |
|    |    | 10    | 273.60 | 0.00 | 3089.00 | 0.00 | 179.71 | 10 | 0.00 | 694.02 | 10 |
| Avg/sum | | | 1.21 | 2201.70 | 8 | 0.35 | 1132.78 | 8 | 0.00 | 179.71 | 10 | 0.00 | 694.02 | 10 |

| 75 | 5  | 1     | 356.34 | 8.18 | 3600.00 | 2.77 | 3600.00 | 0.00 | 967.99 | 1.48 | 3600.00 |
|    |    | 2     | 371.00 | 8.05 | 3600.00 | 0.00 | 1170.00 | 0.00 | 460.38 | 0.00 | 1832.20 |
|    |    | 3     | 373.91 | 12.70 | 3600.00 | 2.13 | 3600.00 | 0.00 | 458.94 | 1.95 | 1200.83 |
|    |    | 4     | 388.91 | 10.26 | 3600.00 | 3.77 | 3600.00 | 1.80 | 3600.00 | 3.01 | 3600.00 |
|    |    | 5     | 378.09 | 12.40 | 3600.00 | 0.00 | 1277.68 | 0.00 | 721.27 | 0.00 | 3460.81 |
|    |    | 6     | 377.11 | 12.37 | 3600.00 | 4.96 | 3600.00 | 2.78 | 3600.00 | 4.51 | 3600.00 |
|    |    | 7     | 373.77 | 8.02 | 3600.00 | 0.00 | 890.69 | 0.00 | 421.24 | 0.00 | 443.39 |
|    |    | 8     | 364.18 | 9.52 | 3600.00 | 0.00 | 2448.88 | 0.00 | 486.44 | 0.00 | 935.07 |
|    |    | 9     | 372.78 | 15.35 | 3600.00 | 5.28 | 3600.00 | 0.00 | 1838.23 | 4.48 | 3600.00 |
|    |    | 10    | 360.79 | 3.25 | 3600.00 | 0.00 | 280.16 | 0.00 | 825.47 | 0.00 | 997.43 |
| Avg/sum | | | 10.01 | 3600.00 | 0 | 1.89 | 2406.74 | 5 | 0.46 | 1338.01 | 8 | 1.54 | 2326.97 | 5 |
| Avg/sum | | | 3.74 | 1942.63 | 17 | 0.75 | 1183.12 | 23 | 0.15 | 512.82 | 28 | 0.51 | 1017.48 | 25 |

cTSPLTW)/cTSPTW, which means that we show the cost improvement that occurs when using lockers in a TSP with time windows. This improvement can reach more than 40%, which shows how the introduction of lockers can help in decoupling the times of customers and couriers and improve the solution cost overall. In fact, the improvement is more evident when the time windows are more restrictive; note, for instance, the very large difference between n20tw20 and n20tw100. The routing costs show an even higher improvement that is due to the fact that more customers are served with lockers and thus do not affect the routing component.

In Figure 3, we compare the value of the solutions of different VRP-based problems on the same instances. In Fig. 3a, we show the %difference in costs between the VRPL and the classical VRP, while in Figure 3b, we show the %difference on costs between the VRPTW (the classical VRP with time windows) and the VRPLTW. They present a trend very similar to what was shown in 2a, 2b; however, one can see that in the multiple vehicle case the cost improvement is lower.
than that of the single vehicle case, even if still very high. On average, the %difference is between 0 and 11% when considering the VRP and the VRPL and between 3% and 31% when considering the VRPTW and the VRPLTW. This is due to the fact that multiple vehicles can operate in parallel, and this makes it easier to serve customers within their time windows with less costly routing solutions.

In Figure 4, we show the solutions of the TSPs and VRPs on one instance with \( n = 20 \), \( m = 2 \), and \( tw = 20 \). In the graphical representation, the circles are the customers, the triangles are the lockers, and the square is the depot. Note the effect of the time windows on the solutions of the TSPTW and VRPTW: the trucks must travel back and forth, running on costly routes, in order to respect the time windows of the customers. One can see that the solutions of the TSPTW and VRPTW take advantage of the lockers and the truck routes are visibly less costly. Moreover, one can see that the TSPL and the VRPL solutions provide an improvement of the TSP and VRP solutions by using lockers to serve customers that are located far from the routes and making the routes effectively shorter.

In Table 9, we show, for all the studied problems, the percentage of lockers used (%lock) and the percentage of customers served via locker (%self) averaged on the number of customers and time window size. One can see that problems with time window constraints use more lockers, and this is also the case of instances with more restrictive time windows because lockers help in desynchronizing by allowing the courier to leave the parcel at the locker before the time window of a customer is opened. For the same reason, the number of lockers that are used is higher in TSPs solutions than in VRPs ones; indeed, the fewer the vehicles, the harder it is to match the time windows. Nevertheless, the number of used lockers increases with the size of the instance.

In Table 10, we show the number of trucks used in the optimal solutions of the several VRPs studied, averaged on the number of customers and the time window size. In the solutions of the problems including time windows, more than twice the number of customers is served via lockers than in the other problems, and this number increases with the increase in the instance size. More restrictive time windows also require more customers to be served via lockers.

The classical VRPTW, having no lockers to be used to serve customers, needs a larger number of vehicles in all cases with respect to the other problems, and this underlines once more the relevance of the use of lockers. On the other hand, the number of vehicles used is minimal when no time window is imposed.
Fig. 4 Graphical representation of the solutions of the studied problems on instance n20l2w20_002

(a) TSP

(b) VRP

(c) TSPTW

(d) VRPTW

(e) TSPL

(f) VRPL

(g) TSPLTW

(h) VRPLTW
Table 9 Percentage of lockers and customers served via lockers in the optimal solutions of the studied problems

| n   | m   | tw | TSPL | TSPLTW | VRPL | VRPLTW |
|-----|-----|----|------|--------|------|--------|
|     |     |    | %lock | %self | %lock | %self | %lock | %self | %lock | %self | %lock | %self |
| 20  | 2   | 20 | 70.00 | 14.00  | 100.00 | 39.00  | 70.00 | 14.00  | 100.00 | 33.00  |
|     | 100 |    | 70.00 | 25.00  | 90.00  | 41.00  | 50.00 | 14.00  | 80.00  | 31.00  |
| Avg |     |    | 70.00 | 19.50  | 95.00  | 40.00  | 60.00 | 14.00  | 90.00  | 32.00  |
| 40  | 4   | 20 | 60.00 | 16.00  | 100.00 | 42.00  | 62.50 | 12.50  | 95.00  | 35.50  |
|     | 100 |    | 65.00 | 21.00  | 87.50  | 33.75  | 50.00 | 18.13  | 80.00  | 24.50  |
| Avg |     |    | 62.50 | 18.50  | 94.44  | 38.33  | 54.17 | 16.25  | 87.50  | 30.00  |
| 60  | 6   | 20 | 60.00 | 20.00  | 88.89  | 40.00  | 66.67 | 15.00  | 95.83  | 35.83  |
|     | 100 |    | 46.67 | 17.67  | –      | –      | 33.33 | 10.00  | –      | –      |
| Avg |     |    | 53.33 | 18.83  | 88.89  | 40.00  | 50.00 | 12.50  | 95.83  | 35.83  |

6.5.1 Analysis on the number of lockers

In this section, we analyze the difference in the total costs of the optimal solutions when allowing only a subset of lockers to be open. In particular, we study the instances with n = 40 for all the problems (TSPL, TSPLTW, VRPL, VRPLTW) by allowing from 0 lockers (thus solving the TSP, TSPTW, VRP, VRPTW, respectively) up to 4 available lockers (as in the solutions presented in previous sections) and we analyze the difference in the optimal solutions. In Fig. 5, we analyze the difference in costs: we propose a boxplot that shows, for each problem, the percentage difference between the value of the optimal solution with 0 lockers and the value of the optimal solutions with 1, 2, 3, and 4 lockers computed as follows: \( \% \text{difference} = 100 \cdot (z_0 - z_i)/z_i \), being \( z_0 \) the solution with 0 lockers and \( z_i \) the value of the solution with several lockers, \( i = 1, \ldots, 4 \). Note that for a few instances, the optimal solution was not available and we did not include the \( \% \text{difference} \) for them. One can clearly see that the problems with time windows benefit greatly from the increase in available lockers, more evidently in the single

Table 10 Number of vehicles used in the optimal solutions of the VRPs

| n   | m   | tw | VRPL | VRPTW | VRPLTW |
|-----|-----|----|------|-------|--------|
|     |     |    | 2.0  | 2.8   | 2.2    |
| 20  | 2   | 20 | 2.0  | 2.4   | 2.0    |
|     | 100 |    | 2.0  | 2.6   | 2.1    |
| Avg |     |    | 2.0  | 3.3   | 2.3    |
| 40  | 4   | 20 | 2.0  | 3.6   | 2.2    |
|     | 100 |    | 2.0  | 3.0   | 2.4    |
| Avg |     |    | 2.0  | 3.3   | 2.3    |
| 60  | 6   | 20 | 2.0  | 3.4   | 2.0    |
|     | 100 |    | 2.0  | 4.0   | –      |
| Avg |     |    | 2.0  | 3.6   | 2.0    |
| 100 | 10  | 20 | 2.0  | 5.2   | –      |
|     | 100 |    | 2.0  | 5.0   | –      |
| Avg |     |    | 2.0  | 5.1   | –      |
| Avg |     |    | 2.0  | 3.5   | 2.2    |

Table 11 Percentage of lockers and customers served via lockers in the optimal solutions of the studied problems on instances with n = 40 with available lockers from 1 to 4

| tw | #lock | TSPL | TSPLTW | VRPL | VRPLTW |
|----|-------|------|--------|------|--------|
|    |       | %lock | %self | %lock | %self | %lock | %self | %lock | %self | %lock | %self |
| 20 | 1     | 100.00 | 6.50  | 100.00 | 10.50 | 50.00 | 3.13  | 100.00 | 11.50 |
|    | 2     | 80.00  | 9.00  | 100.00 | 23.00 | 50.00 | 2.50  | 100.00 | 21.00 |
|    | 3     | 73.33  | 16.00 | 100.00 | 33.13 | 83.33 | 12.50 | 100.00 | 31.00 |
|    | 4     | 60.00  | 16.00 | 100.00 | 42.00 | 62.50 | 12.50 | 95.00  | 35.50 |
| 100| 1     | 40.00  | 3.50  | 100.00 | 10.83 | 50.00 | 1.88  | 100.00 | 9.00  |
|    | 2     | 60.00  | 12.50 | 100.00 | 19.38 | 62.50 | 10.00 | 90.00  | 19.50 |
|    | 3     | 73.33  | 21.00 | 100.00 | 28.75 | 66.67 | 18.13 | 77.78 | 20.83 |
|    | 4     | 65.00  | 21.00 | 87.50  | 33.75 | 50.00 | 18.13 | 80.00  | 24.50 |
vehicle case, improving the solutions costs up to more than 40% when using 4 available lockers. On the other hand, the problems without time windows show a limited improvement in the solution costs with up to 10% when using up to 4 lockers. Moreover, the instances with more restrictive time windows benefit more than the others when the number of available lockers is increased. Note that we divided the instances between tw=20 and tw=100 also for the TSPL and the VRPL because they represent two different sets and one can easily compare the results with those of the TSPLTW and the VRPLTW.

In Table 11, we show the percentage number of lockers used (%lock) and the percentage number of customers served via lockers (%self) in the optimal solutions of the studied problems averaged on the number of available lockers (#lock, from 1 to 4) and on the time window size. The problems with time windows make use of a larger number of lockers and serve more customers via lockers, being the solutions of the TSPLTW those showing the highest numbers. The smaller the time windows, the more lockers are used and the more the customers are served via lockers. In the TSPL and the VRPL solutions, one can see that for the used instances the solutions with 3 and 4 lockers serve the same number of customers via lockers, while the more constrained problems keep benefiting from the increase in available lockers.

In Table 12, we show the number of trucks used in the optimal solution of the VRPL is always two, while it decreases with the increase in available lockers in the optimal solutions of the VRPLTW, because lockers help in decoupling the truck and the time windows, thus allowing to use fewer trucks to perform deliveries while respecting the time windows.

### 7 Conclusion

In this paper, we have considered single and multiple vehicle routing problems with lockers and with time windows. We proposed three mixed integer linear programming formulations, one of which needs a branch-and-cut implementation, that are applied to all the problems, and a set of valid inequalities. The proposed one-commodity formulation is the one that performed better when solving problems without time windows, being able to solve all the instances but one with up to 100 customers and 10 lockers for the single vehicle case, and many with up to 60 customers and 6 lockers for the multiple vehicles case. The same formulation provide the smallest gaps on problems with time windows, but the proposed vehicle flow formulation is faster when solving smaller instances.

We also improve the literature results on two related problems obtaining new optima and improved best-known solutions. By inspecting the solutions, we can conclude that the introduction of lockers diminishes the total costs and provides simpler routes for the vehicles; this is particularly true when time windows are considered. In such cases, the lockers help in decoupling the couriers and the customers’ times allowing less costly routes to be taken. The use of multiple vehicles has a similar, even if less evident, effect.

Possible future research is focused on the improvements of the newly proposed formulations by defining new valid inequalities to include in a branch-and-cut fashion. Moreover, we intend to foster the ideas of this paper to similar applications and problems, including the following features:
delivery preference for customers; lockers modeled with cells of different capacities; pickup and delivery; etc. Eventually, we intend to develop dedicated metaheuristic and branch-and-price algorithms.

**Author Contributions** All authors contributed to the manuscript equally. All authors read and approved the final manuscript.

**Declarations**

**Conflict of Interest** the authors declare that have no conflict of interest.

**Ethical approval** This article does not contain any studies with human participants or animals performed by any of the authors.

**References**

Amazon (2020) Amazon hub. https://www.amazon.it/b?node=16679461031

Baldacci R, Hadjiconstantinou E, Mingozzi A (2004) An exact algorithm for the capacitated vehicle routing problem based on a two-commodity network flow formulation. Oper Res 52(5):723–738

Clarke G, Wright JW (1964) Scheduling of vehicles from a central depot to a number of delivery points. Oper Res 12(4):568–581

Dumas Y, Desrosiers J, Gelinas E, Solomon MM (1995) An optimal algorithm for the traveling salesman problem with time windows. Oper Res 43(2):367–371

Dumée D, Lehuédé F, Péton O (2021) A large neighborhood search approach to the vehicle routing problem with delivery options. Transp Res Part B: Methodol 144:103–132

Enthoven DL, Jargalsaikhan B, Roodbergen KJ, uit het Broek MA, Schrottenhofer AH (2020) The two-echelon vehicle routing problem with covering options: City logistics with cargo bikes and parcel lockers. Comput & Oper Res 118:104919

Gendreau M, Hertz A, Laporte G, Stan M (1998) A generalized insertion heuristic for the traveling salesman problem with time windows. Oper Res 46(3):330–335

Grabenschweiger J, Doerner KF, Hartl RF, Savelsbergh MWP (2021) The vehicle routing problem with heterogeneous locker boxes. Cent Eur J Oper Res. https://doi.org/10.1007/s10100-020-00725-2

Jiang L, Dhiaf M, Dong J, Liang C, Zhao S (2020) A traveling salesman problem with time windows for the last mile delivery in online shopping. Int J Prod Res 58(16):5077–5088

Mancini S, Gansterer M (2020) Vrp with private and shared delivery locations (instances). Mendeley Data V1. https://doi.org/10.17632/w5mvmkn46h.1

Mancini S, Gansterer M (2021) Vehicle routing with private and shared delivery locations. Comput & Oper Res 133:105361. https://doi.org/10.1016/j.cor.2021.105361

Mitrea IA, Zenezini G, De Marco A, Ottaviani FM, Delpastro T, Botta C (2020) Estimating e-consumers’ attitude towards parcel locker usage. In: 2020 IEEE 44th annual computers, software, and applications conference (COMPSAC), IEEE, pp 1731–1736

Oliveira WJP, dos Santos AG (2020) Last mile delivery with lockers: formulation and heuristic. In: Proceedings of the 22nd international conference on enterprise information systems (ICEIS 2020), Vol 1, Science and Technology Publications, pp 460–467. https://doi.org/10.5220/0009580004600467

Orenstein I, Raviv T, Sadan E (2019) Flexible parcel delivery to automated parcel lockers: models, solution methods and analysis. EURO J Transp Logist 8(5):683–711

Osaba E, Del Ser J, Nebro AJ, Laña I, Bilbao MN, Sanchez-Medina JJ (2018) Multi-objective optimization of bike routes for last-mile package delivery with drop-offs. In: 2018 21st international conference on intelligent transportation systems (ITSC), IEEE, pp. 865–870

Rohmer S, Gendron B (2020) A guide to parcel lockers in last mile distribution—highlighting challenges and opportunities from an or perspective. CIRRELT

Sitko P, Wikarek J (2019) Capacitated vehicle routing problem with pick-up and alternative delivery (cvrppad): model and implementation using hybrid approach. Ann Oper Res 273(1–2):257–277

Statista (2020) Retail e-commerce sales worldwide from 2014 to 2021 (in billion u.s. dollars), https://www.statista.com/statistics/379046/worldwide-retail-e-commerce-sales/. (Accessed: 2021-09-10)

Tilk C, Olikis K, Imich S (2021) The last-mile vehicle routing problem with delivery options. OR Spect 43:877–904. https://doi.org/10.1007/s00291-021-00633-0

Toth P, Vigo D (eds) (2002) The vehicle routing problem. Society for Industrial and Applied Mathematics

Wen J, Li Y (2016) Vehicle routing optimization of urban distribution with self-pick-up lockers. In: 2016 international conference on logistics, informatics and service sciences (LISS), IEEE, pp. 1–6

Zhou F, He Y, Zhou L (2019) Last mile delivery with stochastic travel times considering dual services. IEEE Access 7:159013–159021

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.