Last-mile delivery with drone and lockers

Published Version:
Boschetti Marco Antonio, Novellani Stefano (2024). Last-mile delivery with drone and lockers. NETWORKS, 83(2), 213-235 [10.1002/net.22190].

Availability:
This version is available at: https://hdl.handle.net/11585/945394 since: 2024-02-08

Published:
DOI: http://doi.org/10.1002/net.22190

Noname manuscript No.
(will be inserted by the editor)

Last-mile delivery with drone and lockers

Marco Antonio Boschetti · Stefano Novellani

the date of receipt and acceptance should be inserted later

Abstract In this paper, we define a new routing problem that arises in the last-mile delivery of parcels, in which customers can be served either directly at home by a capacitated truck, or possibly with a drone carried on the truck, or in a self-service mode using one of the available lockers.

We investigate four different formulations, and for one of them, we propose a branch-and-cut approach. We also discuss some possible variants of the original problem.

In the computational experiments, we analyze and compare the performance of the four formulations for the problem and its variants, and we provide some useful managerial insights.

Keywords last-mile, routing, lockers, drones, formulations, branch-and-cut

1 Introduction

Global retail e-commerce sales have been increasing steadily since 2014; according to projections, they will more than double in 2023 compared to 2018 and almost triple in 2025 (see [79] and [68]). The rapid delivery of a large number of parcels at a minimum cost is one of the main challenges introduced by the e-commerce boom that retailers and logistics companies must solve. According to a recent McKinsey study (see [40]), 20 to 25% of customers are willing to pay more for same-day delivery, whereas 2% are willing to pay more for instant delivery. The same study confirms that autonomous vehicles could satisfy these requests and predicts that, within a few years, autonomous vehicles (not necessarily aerial) will deliver 80% of all parcels. In response, the major companies are implementing alternative delivery strategies, such as the use of unmanned air vehicles (also called *drones*) coupled with trucks (see, e.g., [57] and [4]) or the use of *lockers* (see, e.g., [53], and [3]).

M.A. Boschetti
Department of Mathematics, University of Bologna, Italy. E-mail: marco.boschetti@unibo.it

S. Novellani
Facultad de Ingeniería y Ciencias, Universidad Adolfo Ibáñez, Viña del Mar, Chile. E-mail: stefano.novellani@uai.cl

Several companies have considered, implemented, or are willing to implement parcel delivery by the means of drones. Among the others, we may mention Alibaba, Alphabet, Amazon, DHL, and JD.com (see, e.g., [4], [5], [2], [24], [36]). Moreover, the European Commission forecasts that the European drone sector will employ more than 100,000 people by 2035 and have an annual economic impact of over 10 billion euros (European Council 2019). As a result, it is not surprising that many hundreds of publications and surveys on the subject have been published in the last few years (see, e.g., [51]). In particular, the literature has focused on problems where one or more drones are equipped on one or more trucks and cooperate to deliver parcels in order to reduce the overall delivery time. On the other hand, current research has focused on lockers, which are pickup locations in areas that are convenient for the customers and where couriers deliver packages that customers will pick up later. The introduction of lockers is gaining popularity among logistics and e-commerce companies (see, e.g., [48]) and among customers because it improves the online shopping experience (see, e.g., [83]) by providing more control and greater flexibility to customers. Furthermore, retailers can enable an easier and seamless delivery process (see, e.g., [39] and [49]). For these reasons, companies such as Deutsche Post, DHL, Decathlon, and Amazon have implemented or expanded their locker offerings (see, e.g., [67], [54], [80], [11], [42], [3], and [44]).

In this paper, we introduce and tackle the traveling salesman problem with drone and lockers, a last-mile delivery problem in which a truck can perform deliveries either directly to the customers or to nearby lockers, or with the use of a drone equipped on the truck that is launched and retrieved from the truck when it is stationary. Because of customer preferences, physical constraints, or due to legislation, not all three types of delivery may be permitted; indeed, drone delivery may be performed primarily in rural areas where houses have gardens. We propose a set of formulations and a branch-and-cut algorithm. To the best of our knowledge, this is the first work that takes into account parcel deliveries via truck, drone, and lockers.

In the remainder of the paper, we firstly propose a detailed literature review in Section 2. We define the problem under consideration in Section 3, while in Section 4, we propose four formulations to model it. We discuss some variants in Section 5, and we briefly discuss the implementation of a branch-and-cut algorithm in Section 6. In Section 7, we report computational results comparing the proposed formulations and provide some comments on the impact of the combined use of drones and lockers in the last-mile delivery process. We draw conclusions in Section 8.

2 Literature Review

The current literature on routing problems where trucks are assisted by drones and the one concerning routing problems that include lockers as delivery points do not overlap. Hence, we provide two separate literature reviews in the following. To our knowledge, no previous paper has studied the use of trucks, drones, and lockers together.

2.1 Routing problems with lockers

Most of the works that consider the use of lockers have been published in the last five years. For a wide classification of recent studies, we head the reader to the survey by

Rohmer and Gendron [70]. In the following, we discuss works that consider lockers and vehicles: routing problems with lockers.

The early studies that consider the use of lockers to deliver packages only use one truck. Jiang et al. [38] 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 via truck or at nearby lockers, which are fulfilled when the truck passes by. The aim is to minimize 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 general variable neighborhood search (GVNS) that is based on a variable neighborhood descent (VND) heuristic. The MILP formulation models hard time windows but the authors solve the case with soft time windows, to increase efficiency. Buzzega and Novellani [12] propose improved MILP formulations and branch-and-cut (B&C) algorithms for a similar problem that they call the *vehicle routing problem* (VRP) *with lockers*, which may or may not consider (hard) time windows and multiple vehicles. The cost of servicing a customer via locker is included in the objective function as a penalty, together with the traveling costs, that must be minimized.

Most of the other works only consider multiple vehicles. Oliveira and dos Santos [59] present a multiple vehicle problem in which one set of trucks is devoted to home delivery only and another one to delivery by lockers. The problem aims to minimize routing costs, with the routing cost of the truck delivering to the lockers being cheaper than the others. The authors propose a MILP formulation and a VND based on the well-known Clarke and Wright (C&W) algorithm (see, e.g., [16]). Orenstein et al. [60] study a VRP called *flexible parcel delivery problem* for delivering parcels from the depot to the lockers. This work includes several differences with respect to the others: for instance, the parcels are delivered only via one of the lockers and home delivery is not allowed. Each parcel comes with a size, and both lockers and trucks are divided into cells of different sizes that must be sufficiently large to contain the assigned parcel. Moreover, failed deliveries are allowed. The objective is to minimize the traveling costs, the cost of using each truck, and the penalties incurred in the event of failed deliveries. The authors propose a MILP formulation and two metaheuristics based on the C&W and the Petal heuristics (see, e.g., [28]), whose solutions are improved by a Tabu Search algorithm.

The works that follow allow customers to express their service preferences to some extent, such as allowing delivery only to a specific set of lockers or choosing between home delivery and locker delivery. The *VRP with delivery options* (VRPDO), presented by Dumez et al. [26], is a VRP with time windows that takes into account a variety of delivery options, including home, workplace, locker, or automobile trunk. Customers define their delivery preferences among the options. The objective is to determine the truck routes that minimize the travel costs and ensure a minimum level of satisfaction for the customers. The authors propose a mixed integer non-linear formulation, but they solve the problem with a dedicated Large Neighborhood Search (LNS), in which a set-partitioning formulation is used to reassemble routes. To solve the VRPDO, Tilk et al. [81] propose a branch-and-cut-and-price algorithm. Mancini and Gansterer [52] propose the *VRP with private and shared delivery locations*, where customers can either be served at home within a preferred time window or at a locker, in both cases with an unlimited homogeneous fleet of uncapacitated trucks. Customers can select one of the two options or leave the choice between the two types of service to the company.

Customers must be compensated when the service is provided at the locker, which is a cost for the company. They propose a MILP model, a LNS-based matheuristic, and an iterated local search algorithm. Grabenschweiger et al. [31] propose the *VRP with heterogeneous locker boxes*, where the 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 uncapacitated trucks. Customers can be served by the closest locker, if they allow so, for which they receive compensation, which is a cost for the company. The authors consider the cases with lockers having boxes of a single dimension and having different dimensions. They propose two MILP models, each of which considers one of the two cases, and an Adaptive LNS-based (ALNS) metaheuristic.

Only a few papers consider using lockers to collect returned items, even though customers who shop online are becoming accustomed to returning packages. Sitek and Wikarek [77] solve a pickup and delivery VRP with lockers and proposed an ILP model and heuristic methods. Sitek et al. [78] extend their previous work by considering time windows. Yu et al. [87] solve the VRP with lockers where simultaneous pickups and deliveries are considered by using a MILP model and a simulated annealing. In a recent work, Dell'Amico et al. [21] consider different variants of the pickup and delivery routing problems with lockers and time windows, proposing MILP formulations, two of them solved via B&C algorithms, improving the previous results.

Three recent works solve two-echelon problems with lockers, namely those proposed by Zhou et al. [90], Enthoven et al. [27], and Li et al. [47]. Only one work, proposed by Zhou et al. [89], considers stochastic information in these problems that includes stochastic travel times.

2.2 Routing problems with truck(s) and drone(s)

The amount of literature that considers optimization problems related to drones or trucks and drones has been experiencing a boom in the last few years, and, as a result, the number of related surveys has rocketed. Here we list the most relevant ones in alphabetical order: Boysen et al. [10], Chung et al. [15], Khoufi et al. [43], Li et al. [46], Macrina et al. [51], Otto et al. [61], Rojas Viloria et al. [71]. Note also that the paper by Poikonen and Campbell [63] focused on future research directions.

In the following, we focus on those problems that consider the combined and synchronized use of truck(s) and drone(s) for deliveries. The first problem that considers a single drone and a single truck is the *flying sidekick TSP* (FSTSP), which was proposed in the seminal work of Murray and Chu [57]. In such a problem, the drone is installed on the truck and can be launched and retrieved to serve one customer at a time, while the truck can serve other customers in parallel. The drone cannot exceed its battery endurance. The two vehicles cooperate to serve customers, minimizing the completion time of the operations. In their work, Murray and Chu propose a MILP formulation and three heuristic methods based on well-known heuristics for the VRP.

The *TSP with drone* (TSP-D), presented by Agatz et al. [1] for the first time, is another problem that shares the main characteristics with the FSTSP. In the TSP-D, the truck can visit the customers multiple times if it is convenient for drone launching and returning; the launching and rendezvous locations of the same flight may coincide (the so-called loops can be performed); endurance is unlimited; and the launching and rendezvous times are considered negligible. The authors present an ILP model and propose route first-cluster second heuristics. Bouman et al. [9] extend the work of [1] by solving the TSP-D with dynamic programming and a dynamic programming-based heuristic.

Ha et al. [32] solve a type of FSTSP by proposing two heuristic algorithms: a route first-cluster second and a cluster first-route second. In Ha et al. [33], the same authors solve a similar problem with a different objective function, with the aim of minimizing the total distance traveled by the truck and by the drone, and the waiting time of the truck and of the drone. They present a MILP formulation based on Murray and Chu's one and two heuristics. Ha et al. [34] propose a hybrid genetic algorithm improved with local search procedures to solve both the minimum time and the minimum cost version of the FSTSP. In [29], de Freitas and Penna propose a randomized variable neighborhood descent for the FSTSP. In [30], the same authors propose a hybrid general variable neighborhood search algorithm for the FSTSP proposed by Murray and Chu [57] and the TSP-D proposed by Agatz et al. [1].

Poikonen et al. [65] propose a branch-and-bound (B&B) for a TSP-D version that considers maximum endurance for the drone flights and allows loops but not multiple visits to the same node. The studied problem is basically a FSTSP with loops in which no times for launching and collecting the drone are considered. They also propose three heuristic algorithms, two of which are derived from the B&B and one that is a divide-and-conquer method. Yurek and Ozmutlu [88] propose an exact and a heuristic iterative algorithm based on a decomposition approach for the FSTSP.

Dell'Amico et al. [23] solve two FSTSP variants: one where drones can wait at customer nodes while on the ground, and the other where they can only wait when hovering and thus using the energy of the battery. They propose improved MILP formulations and B&C algorithms. The same authors propose a random restart local search matheuristic for the second version (hovering) of the FSTSP in Dell'Amico et al. [19], and B&B-based algorithms (exact and heuristic) for the same version of the FSTSP, in Dell'Amico et al. [22]. Finally, they present enhanced MILP formulations and B&C algorithms adapted to solve several variants of the FSTPS and TSP-D, in Dell'Amico et al. [20].

Boccia et al. [7] solve the FSTSP with loops with a B&C algorithm. In [6], the same authors propose a column-and-row generation algorithm to solve the FSTSP.

Roberti and Ruthmair [69] propose exact solution approaches, based on MILP and branch-and-price (B&P), to solve TSP-D in which no loops, no endurance, and no multiple visits are allowed. Then they also show how to include these variants into their methods. Schermer et al. [75] propose MILP formulations and B&C algorithms to solve the TSP-D in which loops are allowed, endurance is limited, only single visits are allowed, and drones can wait on the ground without consuming battery. The authors also explain how they accommodate the other features in their methods but not multiple visits.

Recent works have investigated problems involving a single truck and multiple drones, to name a few: Tu et al. [82], Moshref-Javadi et al. [55], Seifried [76], Murray and Raj [58], Cavani et al. [13], and Dell'Amico et al. [18]. Other works consider that the truck can stop at so-called anchor points from which one or more drones are launched once or multiple times to serve customers, also in parallel with the truck; see, e.g., Peng et al. [62], Hu et al. [35], Moshref-Javadi et al. [56], Salama and Srinivas [73], and Chang and Lee [14]. Multiple visits along the same drone route are considered in Karak and Abdelghany [41], Luo et al. [50], Jeong and Lee [37], Poikonen and Golden [64], and Hu et al. [35].

Moreover, other studies further generalized the problem to include multiple vehicles. Wang et al. [84] define the *VRP with drones* (VRPD); Poikonen et al. [66] treat these problems theoretically; Daknama and Kraus [17] heuristically; Sacramento et al. [72] propose a MILP and an ALNS for the VRPD; Schermer et al. [74] propose a formulation and a matheuristic; Kitjacharoenchai et al. [45] a MILP formulation and an adaptive insertion heuristic. Wang and Sheu [85] solve a VRP with drones in which each drone can serve multiple customers during one flight because the parcels are parachuted. Di Puglia Pugliese et al. [25] propose and solve the VRPD including time windows.

3 Problem Definition

The *traveling salesman problem with drone and lockers* (TSPDL) is modeled on a digraph $G = (V, A)$. The set of vertices V is partitioned as $V = \{0\} \cup V_C \cup V_L \cup \{d\}$, where $V_C = \{1, 2, \dots, n\}$ is the set of the n customers and $V_L = \{n+1, n+2, \dots, n+m\}$ is the set of the m available lockers, whereas 0 and $d = n+m+1$ represent the depot's duplication at the beginning and end of the operations, respectively. $A = \{(i, j) : i, j \in V\}$ is the given set of arcs, each of which is associated with τ^T_{ij} and τ^D_{ij}, the times to travel the arc (i, j) required by the truck and the drone, respectively.

The problem is to deliver all the parcels with one of the three methods—the truck, the drone, or the lockers—at a minimum cost. The truck must leave from and return to the depot and can serve the customers either directly at home (home service) with the truck or the drone, or via a set of lockers selected by the customers (self-service). When a customer $i \in V_C$ is served by a locker $b \in V_L$, self-service results in a c_{ib} discount, and thus represents a cost of the same value that is thus included in the objective function. The maximum capacity of each locker $b \in V_L$ is determined by the number of cells it contains ν_b, which limits the number of customers it can serve. Moreover, the truck can visit the locker at most once.

The drone can perform so-called *sorties*, which are flights described as node triplets: the launching node, where the drone leaves the truck; the node of the customer to be served; and the rendezvous node, where the drone returns to the truck. Notice that we describe an *unmanned aerial vehicle* (UAV), but we later show how to modify the model to treat an *unmanned ground vehicle* (UGV). The drone can be launched from all the nodes of the graph and return to all of them, but only when the truck is stationary in those places. In our problem definition, we follow the common assumption of the truck and drone literature (see, e.g., Murray and Chu [57] or Dell'Amico et al. [23], to name a few) and impose that the launch and rendezvous nodes of the same sortie must be different and that nodes can be visited at most once. We are aware that visiting a node multiple times to launch and retrieve the drone or allowing it to perform loops may provide less expensive solutions, and this is a limitation of our method. Each flight has a maximum endurance E, which is the amount of time the battery guarantees. After each flight the drone battery is replaced with a new charged one. The times for launch and rendezvous are σ^L and σ^R, respectively. The rendezvous time consumes battery, while the launching time does not because, in such a case, the drone is not flying. We may include the time required for the battery exchange and the parcel preparation in the launch time. The truck must be stationary during the launch and the rendezvous. The service times at customers' and lockers' nodes are included in the traveling times, if

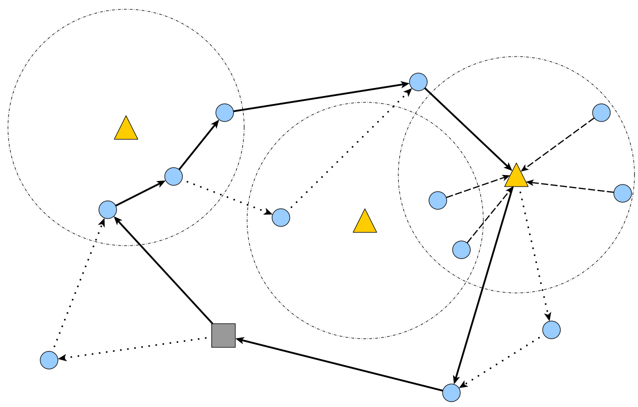

Fig. 1: A feasible solution for the TSPDL.

given. In our model, we assume that the service at customers homes and lockers is done before handling the drone; however, one can discount the service time at a rendezvous node from the drone flight in the following formulations to allow the rendezvous to be done after the collection of the drone. On the other hand, allowing the drone to be launched before the service at a node cannot be easily inserted in the proposed formulations, and it is thus a limitation of the proposed model.

Customers can specify their delivery requests or let the delivery company decide. We express this as follows. A binary constant δ_{ib} equals 1 if the customer $i \in V_C$ allows to be served by locker $b \in V_L$, 0 otherwise. Note that some customers may select more than one locker, but some may select none, and thus no self-service is possible for them. A binary constant γ_i equals 1 if the customer $i \in V_C$ allows to be served by the truck, 0 otherwise. A binary constant β_i equals 1 if the customer $i \in V_C$ allows or can be served (maybe due to the size of the parcel) by drone, 0 otherwise. Note that at least one delivery method must be available for each customer, and thus $\sum_{b \in V_L} \delta_{ib} + \gamma_i + \beta_i \geq 1$ for all the customers $i \in V_C$.

The objective function to be minimized consists of two components: the completion time and the cost due to the discount guaranteed to customers that collect their parcels in a locker. Each of these components is multiplied by a coefficient: α_1 and α_2, respectively.

A feasible solution for the TSPDL is presented in Figure 1, in which the square is the depot, the blue circles are the delivery customers, and the triangles are the lockers, each of which is associated with a set of customers that can be served by it, represented by the delivery customers included in the corresponding large circle. The arrows represent the truck route and the dashed arrows represent the customer's path to collect their parcels at the lockers. The dotted lines represent the sorties of the drone.

In Table 1, one can find a summary of all the symbols used.

Table 1: Used symbols

Sets	
V	Set of vertices, $V = \{0\} \cup V_C \cup V_L \cup \{d\}$
V_C	Set of customers, $V_C = \{1, 2, \dots, n\}$
V_L	Set of lockers, $V_L = \{n+1, n+2, \dots, n+m\}$
A	Set of arcs, $A = \{(i,j) : i, j \in V\}$
Parameters	
0, d	Depot node at the beginning and at the end of the operations, respectively
n	Number of customers
m	Number of available lockers
τ^T_{ij}	Time needed by the truck to travel an arc $(i,j) \in A$
τ^D_{ij}	Time needed by the drone to travel an arc $(i,j) \in A$
ν_b	Capacity of locker $b \in V_L$
δ_{ib}	Binary constant that equals 1 if customer $i \in V_C$ can be served by locker $b \in V_L$, 0 otherwise
γ_i	Binary constant that equals 1 if customer $i \in V_C$ can be served directly by the truck, 0 otherwise
β_i	Binary constant that equals 1 if customer $i \in V_C$ can be served by the drone, 0 otherwise
c_{ib}	Cost for serving customer $i \in V_C$ from locker $b \in V_L$
E	Endurance of the drone for each sortie
σ^L	Launching time of the drone
σ^R	Rendezvous time of the drone
α_1	Weight associated to the costs due to the completion time
α_2	Weight associated to self-service costs
b_i	Right hand side value for the flow conservation constraints: $b_i = 0$ if $i \in V_C \cup V_L$, $b_i = 1$ if $i = 0$, and $b_i = -1$ if $i = d$

4 Formulations

In this section, we propose four different TSPLD formulations, each of which has two main parts: a component for the drone and a component for the lockers. The first formulation that we suggest is inspired by some of the best formulations in the literature for the two subproblems related to those components. Despite the fact that the inspiring formulation of the drone component can be effective when solving some particular instances, it has a flaw that makes it ineffective in a more general case (see next section for details). To overcome that issue, we suggest how to fix the formulation to provide proven optimal solutions in any case, but this severely weakens it. As a result, we suggest a second formulation to find proven optimal solutions possibly more quickly. We propose a third formulation to lower the number of binary variables and possibly improve results. We then suggest a fourth formulation to further reduce the number of variables and constraints by introducing some inequalities in a cutting plane fashion to avoid infeasible solutions.

4.1 Formulation One

We first propose a formulation inspired by Roberti and Ruthmair's [69] TSP-D model and Buzzega and Novellani's [12] TSPL model, which uses the following sets of variables:

- x^T_{ij}, a binary variable that equals 1 if the truck travels arc $(i,j) \in A$, 0 otherwise.
- x^D_{ij}, a binary variable that takes value 1 if the drone travels the arc $(i,j) \in A$; 0 otherwise. Note that the variable x^D_{ij} may take value 1 also if the drone travels the

arc (i, j) on the truck. However, the model does not impose the variable $x_{ij}^D = 1$ every time the drone is on the truck when traveling that arc.

- y_i^T, a binary variable that takes value 1 if and only if the node $i \in V$ is visited by the truck only; 0 otherwise.
- y_i^D, a binary variable that takes value 1 if and only if the node $i \in V$ is visited by the drone only; 0 otherwise.
- y_i^C, a binary variable that takes value 1 if and only if the node $i \in V$ is visited by both the truck and the drone in a combined way, but served by the truck; 0 otherwise.
- y_{ib}, a binary variable that equals 1 if and only if the customer $i \in V_C$ travels to locker $b \in V_L$ to collect its parcel; 0 otherwise.
- t_i, a non-negative time variable that defines the earliest moment at which the truck can leave node $i \in V$. Hence, t_d is the completion time of all operations.

To help the reader, we first show the fixing of variables, and then we present the formulation by grouping its components with respect to what they represent.

Variables fixing We fix some of the variables as follows. First, we set $t_0 = 0$, $x_{i0}^T = x_{di}^T = x_{ii}^T = 0$, $i \in V$, and $x_{i0}^D = x_{di}^D = x_{ii}^D = 0$, $i \in V$. If $\gamma_i = 0$, then customer $i \in V_C$ cannot be served by the truck and thus $x_{ij}^T = x_{ji}^T = 0, j \in V$ and $x_{i0}^T = x_{id}^T = 0$. We set $y_i^D = 0$ if $\beta_i = 0$, $i \in V_C$, $y_b^D = 0$, $b \in V_L$, because no drone is required to serve a locker. Moreover, y_{ib} is set to 0 if $\delta_{ib} = 0$, $i \in V_C, b \in V_L$.

Objective function

$$\min z = \alpha_1(t_d + \sum_{i \in V}(\sigma^L + \sigma^R)y_i^D) + \alpha_2 \sum_{i \in V_C} \sum_{b \in V_L} c_{ib} y_{ib} \tag{1}$$

The objective function (1) is made of two components, one for the completion time and one for the penalty of customers served via lockers. Note that the time of arrival at the final depot (t_d) does not include the launch and rendezvous times, and thus we must include them separately. Notice that we decompose the completion time into t_d (the truck's arrival at the final depot; that includes the waiting times at the rendezvous nodes but not the launch and rendezvous times) and into the launch and rendezvous times (σ^L and σ^R), in case launches and rendezvous are performed. This decomposition is convenient because it improves the value of the linear relaxation. Indeed, it enables direct payment for launch and rendezvous times when performed, rather than including them in the time constraints that are big-M constraints (see, e.g., Dell'Amico et al. [23]).

Routing constraints

$$\sum_{j \in V} x_{ij}^T - \sum_{j \in V} x_{ji}^T = b_i \quad i \in V \tag{2}$$

$$\sum_{j \in V} x_{ij}^D - \sum_{j \in V} x_{ji}^D = b_i \quad i \in V \tag{3}$$

$$\sum_{j \in V} x_{ij}^T = y_i^T + y_i^C \quad i \in V, i \neq d \tag{4}$$

$$\sum_{j \in V} x_{ij}^D = y_i^D + y_i^C \quad i \in V, i \neq d \tag{5}$$

$$y_i^T + y_i^D + y_i^C + \sum_{b \in V_L} y_{bi} = 1 \quad i \in V_C \tag{6}$$

$$y_i^T + y_i^C \leq 1 \quad i \in V_L \tag{7}$$

Constraints (2) and (3) ensure flow conservation in every vertex for truck and drone arc binary variables, respectively, with $b_i = 0$ if $i \in V_C \cup V_L$, $b_i = 1$ if $i = 0$, and $b_i = -1$ if $i = d$. In constraints (4), we impose that the truck arc variable takes value 1 if the node is visited by the truck only ($y_i^T = 1$) or as a combined visit ($y_i^C = 1$). In constraints (5) we write a similar condition for the drone arcs. Constraints (6) state that a customer node can either be visited by the truck or drone only, by both combined, or served via one of the lockers. Constraints (7) state that the lockers can be visited at most once by either the truck or the truck carrying the drone.

Sortie constraints

$$x_{ij}^D + x_{ji}^D \leq y_i^C + y_j^C \quad i, j \in V : i < j \tag{8}$$

Constraints (8) imposes that the arc (i, j) can be traveled by the drone only if at least one of the two nodes is a combined node.

Locker constraints

$$\sum_{j \in V} x_{jb}^T \leq 1 \quad b \in V_L \tag{9}$$

$$\sum_{j \in V} x_{jb}^T \leq \sum_{i \in V_C} y_{ib} \quad b \in V_L \tag{10}$$

$$\sum_{j \in V} x_{jb}^T \geq y_{ib} \quad i \in V_C, b \in V_L \tag{11}$$

$$\sum_{i \in V_C} y_{ib} \leq \nu_b \quad b \in V_L \tag{12}$$

Constraints (9) impose that the locker can be visited at most once. Constraints (10) guarantee that no arc enters the locker if it is not used. Constraints (11) impose that if a locker is used, then one arc entering the locker must take value one. Constraints (12) impose a maximum capacity on the number of deliveries assigned to each locker.

Time constraints

$$t_j \geq t_i + \tau_{ij}^T - M(1 - x_{ij}^T) \quad (i, j) \in A, i \neq j \tag{13}$$

$$t_j \geq t_i + \tau_{ij}^D - M(1 - x_{ij}^D \overbrace{+ x_{ij}^T}^{\text{if } \tau_{ij}^D > \tau_{ij}^T}) \quad (i, j) \in A, i \neq j \tag{14}$$

Constraints (13) state that if the truck travels arc (i, j), the time in j is updated by the arc's traveling time with respect to the time in i. It is worth noting that by using these constraints, we can avoid including the well-known subtour elimination constraints in this and subsequent formulations. Constraints (14) update times when an arc is traveled by the drone. Note that these constraints require the travel time to be updated by considering the drone times only when the drone does not travel the arc on the truck. This prevents the need for extra time to traverse the arc in the event that $\tau_{ij}^D > \tau_{ij}^T$.

Endurance constraints

$$t_k - t_i + \sigma^R \leq E + M(3 - x_{ij}^D - x_{jk}^D - y_j^D) \quad i \in V, j \in V_C, k \in V, i \neq k \tag{15}$$

Constraints (15) guarantee that the endurance is not exceeded when a sortie is performed. It is important to note that we must include the y^D variable because $x_{ij}^D = x_{jk}^D = 1$ does not always imply that a sortie (i, j, k) is performed; in fact, we may also have $x_{ij}^D = x_{jk}^D = 1$ when the drone is returning from i to j and then leaving again to serve k.

Variable constraints

$$x_{ij}^T \in \{0, 1\} \quad i \in V, j \in V \tag{16}$$
$$x_{ij}^D \in \{0, 1\} \quad (i, j) \in A \tag{17}$$
$$y_{ib} \in \{0, 1\} \quad i \in V_C, b \in V_L \tag{18}$$
$$y_i^T, y_i^D, y_i^C \in \{0, 1\} \quad i \in V \tag{19}$$
$$t_i \geq 0 \quad i \in V \tag{20}$$

Variables x^T, x^D, y, y^T, y^D, and y^C are defined binary in (16), (17), (18), and (19), respectively. Variables t are set as non-negative variables in (20).

Polynomial-size families of valid inequalities In the following, we write some polynomial-size families of valid inequalities to help the convergence of Formulation One.

$$t_d \geq \sum_{(i,j) \in A} \tau_{ij}^T x_{ij}^T \tag{21}$$

$$t_d \geq \sum_{(i,j) \in A} (\tau_{ij}^D x_{ij}^D \overbrace{+ (\tau_{ij}^T - \tau_{ij}^D) x_{ij}^T}^{\text{if } \tau_{ij}^D > \tau_{ij}^T}) \tag{22}$$

$$x_{ij}^D \leq x_{ij}^T \quad (i, j) \in A : \tau_{ij}^D > E \tag{23}$$

$$\sum_{j \in V} \tau_{ij}^D x_{ij}^D + \sum_{j \in V} \tau_{ji}^D x_{ji}^D \leq E - \sigma^R + M(1 - y_i^D) \quad i \in V_C \tag{24}$$

$$x_{ij}^T + x_{ji}^T \leq 1 \quad (i, j) \in A \tag{25}$$

$$x_{ij}^D + x_{ji}^D \leq 1 \quad (i, j) \in A \tag{26}$$

Inequalities (21) and (22) provide lower bounds on the time of arrival at the final depot. It is worth noting that in order to have a good lower bound at the root node, we must include inequalities (21) and (22). This is due to the fact that the solution value of the linear relaxation without these inequalities is very weak, being that variables t are bounded by possibly fractional values of the x variables. By imposing constraint (21), on the other hand, we guarantee that all fractional values of x sum up and bound t_d

from below. Inequalities (23) ensure that every arc (i, j) with a travel time greater than the endurance E can be traveled by the drone only if it is on the truck. Inequalities (24) avoid certain infeasible sorties. Moreover, inequalities (25) and (26) avoid subtours of size two for variables x^T and x^D, respectively. It should be noted that all of the polynomial-size families of inequalities have been tested to demonstrate that their inclusion improves the average solving times (see, e.g., Section 7).

The part of Formulation One that takes into account the drone was inspired by the Roberti and Ruthmair model [69], which provides proven optimal solutions in case $\tau_{ij}^D \leq \tau_{ij}^T$ for all arcs $(i, j) \in A$. However, in case $\tau_{ij}^D > \tau_{ij}^T$, when both the truck and the drone travel the same arc ($x_{ij}^T = x_{ij}^D = 1$) and the drone is therefore carried by the truck, the truck's arc may still take longer than expected due to the drone's time constraints. In such a case, the solution provided by the model is suboptimal. In the following, we address this issue by introducing constraints (14) and, as a result, we provide a formulation capable of yielding the proven optimal solution even for instances that do not meet this condition: $\tau_{ij}^D \leq \tau_{ij}^T$ for all arcs $(i, j) \in A$.

On the other hand, the solution where $x_{0,1}^T = x_{1,2}^T = x_{0,2}^D = 1$ and $y_0^C = y_2^C = y_1^T = 1$ is possible for Formulation One because the model allows for a drone to travel an arc $(i, j) \in A$ between two nodes with the corresponding y^C variable set to 1, while the truck is traveling along a path from i to j. In such a case, the drone is not serving any customers and is considered to be carried by the truck (even if the variables do not state it explicitly). Due to the constraints (14), this does not result in suboptimal solutions because if τ_{ij}^D were larger than the time needed by the truck to travel along the path from i to j, then the objective function would avoid setting x_{ij}^D to 1. This could also be applied to all of the possible subpaths of the drone path from i to j. Hence, when the drone traveling time of an arc of the path is greater than the truck traveling time of the same arc, constraints (14) prevent the truck path time from being stretched.

4.2 Formulation Two

We define a new formulation called Formulation Two, which inherits variables x_{ij}^T, x_{ij}^D, y_{ib}, and t_i from the previous one. Additionally, the following variables are required:

- x_{ij}^L, a binary variable that equals 1 if the drone is launched from node $i \in V$ to serve customer $j \in V_C$, 0 otherwise.
- x_{ij}^R, a binary variable that equals 1 if the drone is collected at node $j \in V$ after serving customer $i \in V_C$, 0 otherwise.

Notice that x^R or x^L equal one only when the drone is not on the truck (i.e., if and only if $x_{ij}^T = 0$ and $x_{ij}^D = 1$).

In the following, we describe the variable fixing and the mathematical formulation grouping the components as done for Formulation One.

Variables fixing The variables x_{ij}^T, x_{ij}^D, y_{ib}, and t_0 are fixed in the same way as in the previous formulations. Variables $x_{ji}^L = x_{ij}^R = 0$ if $\beta_i = 0$, $i \in V_C, j \in V$, and $x_{di}^L = x_{0i}^R = x_{ii}^L = x_{ii}^R = 0$, $i \in V_C$. We also set $x_{ij}^L = 0$ if $\tau_{ij}^D > E$, for $i \in V$, $j \in V_C$, and $x_{ij}^R = 0$ if $\tau_{ij}^D + \sigma^R > E$, for $i \in V_C$, $j \in V$.

Objective function

$$\min z = \alpha_1 (t_d + \sum_{i \in V} \sum_{j \in V_C} \sigma^L x^L_{ij} + \sum_{i \in V_C} \sum_{j \in V} \sigma^R x^R_{ij}) + \alpha_2 \sum_{i \in V_C} \sum_{b \in V_L} c_{ib} y_{ib} \tag{27}$$

The objective function (27) is equivalent to that of Formulation One, but updated to the new variables. And thus it minimizes the same two components, each one multiplied by its coefficient: the time at which the operations are completed and the penalty to serve customers via lockers. The same consideration on inequality (21) is valid also for this formulation.

Routing constraints

$$\sum_{i \in V} x^T_{ij} + \sum_{i \in V} x^L_{ij} + \sum_{b \in V_L} y_{jb} = 1 \quad j \in V_C \tag{28}$$

$$\sum_{i \in V} x^T_{ji} + \sum_{i \in V} x^R_{ji} + \sum_{b \in V_L} y_{jb} = 1 \quad j \in V_C \tag{29}$$

Constraints (28) and (29) impose that a customer is served at home by a truck or a drone, or in a locker. The routing constraints (2) and (3) are inherited from the previous formulation.

Sortie constraints

$$\sum_{i \in V} x^L_{ij} = \sum_{i \in V} x^R_{ji} \quad j \in V_C \tag{30}$$

$$\sum_{j \in V_C} x^L_{ij} \leq \sum_{k \in V} x^T_{ik} \quad i \in V \tag{31}$$

$$\sum_{i \in V_C} x^R_{ij} \leq \sum_{k \in V} x^T_{kj} \quad j \in V \tag{32}$$

$$x^D_{ij} - x^T_{ij} \leq \overbrace{x^L_{ij}}^{\text{if } j \in V_C} + \overbrace{x^R_{ij}}^{\text{if } i \in V_C} \leq x^D_{ij} \quad (i,j) \in A \tag{33}$$

$$\overbrace{x^L_{ij}}^{\text{if } j \in V_C} + \overbrace{x^R_{ij}}^{\text{if } i \in V_C} \leq 1 - x^T_{ij} \quad (i,j) \in A \tag{34}$$

Constraints (30) state that, if the drone is launched to serve a customer node j, then it also needs to leave that node. It is trivial to prove that one of the two constraint sets between (30) and (29) is redundant, and thus could be removed. In fact, by subtracting (28) from (29) and then applying the flow conservation (2) to the resulting equality, (30) can be obtained. We report the constraint to make it clear that the flow conservation of the drone arcs is guaranteed as well. Constraints (31) and (32) avoid launches and rendezvous of the drone from and to nodes not visited by the truck, respectively. Constraints (33) and (34) require the drone not to fly along the arc $(i, j) \in A$ if it is also traveled by the truck (even when carrying the drone), while they impose the use of one of the two drone variables x^L_{ij} and x^R_{ij} (when defined) if the arc $(i, j) \in A$ is traveled by the drone alone. Note that variables x^L and x^R are only defined for a restricted set of arcs, and thus we make it explicit in (33) and (34), instead of writing three similar constraints each.

Locker constraints Constraints (9)–(12) are used to model the lockers.

Time constraints

$$t_j \geq t_i + \tau^D_{ij} - M(1 - \overbrace{x^L_{ij}}^{\text{if } j \in V_C} - \overbrace{x^R_{ij}}^{\text{if } i \in V_C}) \quad (i,j) \in A, i \neq j \tag{35}$$

Constraints (35) update time variables when the arcs are traversed by the drone. Note that x^L_{ij} (resp., x^R_{ij}) are only defined for $j \in V_C$ (resp., $i \in V_C$), but time variables must be updated for all nodes, and hence we indicate when variables are not defined in the constraints. It is worth noting, once more, that the launch and rendezvous times are not included in the time variables t_i, allowing us to compute them explicitly in the objective function. These constraints are needed to represent the travel times and to avoid subtours. Constraints (13) are inherited and included in this formulation and, with constraints (35), define the arrival time at each node. For the nodes not visited (i.e., customers served by a locker or locker not used), no value is imposed to the variables t, but this does not affect the actual travel times for the truck and the drone.

Endurance constraints

$$t_k - t_i + \sigma^R \leq E + M(2 - x^L_{ij} - x^R_{jk}) \quad i \in V, j \in V_C, k \in V, i \neq k \tag{36}$$

Constraints (36) guarantee that the time needed by the drone to perform the sortie (i, j, k) does not exceed the endurance time E. Note that the rendezvous time needed by the drone to land on the truck could be added on both the truck and the drone time when a rendezvous happens; however, since we do not include rendezvous times in the time variables computation, to obtain a simpler model and to better decompose the completion time in the objective function, we thus need to add that time when considering the duration of a sortie, being that σ^R is performed while flying.

Variable constraints

$$x^L_{ij} \in \{0,1\} \quad i \in V, j \in V_C \tag{37}$$

$$x^R_{ij} \in \{0,1\} \quad i \in V_C, j \in V \tag{38}$$

Constraints (16)–(18), (20), (37), and (38). define the possible values of the variables.

Polynomial-size families of valid inequalities To help the convergence of Formulation Two, we use the valid inequalities (21), (25), (26) and the following ones:

$$t_d \geq \sum_{i \in V} \sum_{j \in V_C} \tau^D_{ij} x^L_{ij} + \sum_{i \in V_C} \sum_{j \in V} \tau^D_{ij} x^R_{ij} \tag{39}$$

$$x^L_{ij} + x^R_{ji} \leq 1 \quad i \in V, j \in V_C \tag{40}$$

$$x^L_{ij} + x^L_{ji} \leq 1 \quad i \in V_C, j \in V_C \tag{41}$$

$$x^R_{ij} + x^R_{ji} \leq 1 \quad i \in V_C, j \in V_C \tag{42}$$

$$x^L_{ij} + x^R_{ij} \leq 1 \quad i \in V_C, j \in V_C \tag{43}$$

$$x^L_{ij} + x^R_{jk} \leq 1 \quad i \in V, j \in V_C, k \in V, i \neq k : \tau^D_{ij} + \tau^D_{jk} + \sigma^R > E \tag{44}$$

Together with (21), inequalities (39) provides a lower bound on the time of arrival at the final depot. Inequalities (40)–(44) avoid infeasible sorties and subtours of size two. In particular, constraints (40) avoid sorties that start and return at the same node. Note that subtour elimination is guaranteed by the time variables and the corresponding constraints; however these inequalities can strengthen the formulation. It is worth noting that if a subset of nodes is located at the same position and thus the time to travel between each pair (i, j) of those nodes is 0 ($\tau^T_{ij} = 0$), then the subtour elimination must be imposed explicitly among those nodes. All of the described polynomial-size families of inequalities have been tested to demonstrate that they improve the formulation's average solving times (see, e.g., Section 7).

4.3 Formulation Three

The formulation described in this section, Formulation Three, reduces the number of needed binary variables and constraints with respect to Formulation Two. It is inspired by those proposed in Dell'Amico et al. [20] for the FSTSP (compared to which we reduced the number of linear variables and big-M constraints in the drone component of the model) and in Buzzega and Novellani [12] for the TSPL, and makes use of the already defined variables x^T_{ij}, x^L_{ij}, x^R_{ij}, y_{ib}, t_i and the following new ones:

– z_i, a binary variable that allows to define if a sortie can be performed and takes value 1 if the drone is on the truck or is launched or retrieved at node $i \in V$, 0 otherwise. z_0 and z_d can be set to 1 to help the computation of the other z variables.

Regarding the part of the model concerning the drone, with respect to Dell'Amico et al. [20] that inspires it, we avoid using a linear variable to specify how long the truck must wait for the drone and the corresponding big-M constraints. Moreover, by using constraints (35), rather than two sets of time constraints for launch and rendezvous arcs, we also reduce the number of big-M time constraints that are used.

In the following, we only highlight the model's previously unreported components and briefly mention the remaining parts.

Variables fixing We fix variables x^T, x^L, x^R, y and t as in the previous formulations.

Objective function Formulation Three inherits (27) from Formulation Two and the relative inequalities (21) and (39) are also still valid.

Routing constraints Formulation Three inherits (2), (28), and (29) from the previous formulations.

Time constraints Formulation Three inherits constraints (13) and (35) from the previous formulations. These constraints are needed to represent the travel times and to avoid subtours.

Endurance constraints Constraints (36) are inherited from Formulation Two.

Sortie constraints

$$z_i \leq \sum_{j \in V} x^T_{ji} \quad i \in V \tag{45}$$

$$\sum_{j \in V_C} x^L_{ij} \leq z_i \quad i \in V \tag{46}$$

$$z_j \leq z_i - x^T_{ij} + \sum_{k \in V_C} (x^R_{kj} - x^L_{ik}) + 1 \quad i \in V, j \in V, i \neq j \tag{47}$$

Constraints (30) are inherited from Formulation Two and included in this formulation as well. Constraints (45) impose that variable z_i is 0 if no truck arc is entering node i. Constraints (46) impose that launches can happen at node i on the truck only when $z_i = 1$. Constraints (47) are used to update variable z along the truck route in order to avoid the so-called crossing sorties, which are those infeasibilties that occur when the drone is launched before it is retrieved from a previous launch. If the arc (i, j) is not traveled by the truck, then the variable $x^T_{ij} = 0$ and no binding constraint is imposed to z variables. If the arc (i, j) is traveled by the truck, then the constraint becomes $z_j \leq z_i + \sum_{k \in V_C}(x^R_{kj} - x^L_{ik})$. In case no launch is performed at node i or return at node j, then the variable z_j is imposed to 0 if $z_i = 0$ (the drone is not on the truck at node i and is not collected at node j), while z_j is free otherwise, and thus it can be 1 (a launch at node j can occur). If the drone is launched at node i, then variable z_i must be 1, and z_j must be 0 if no return happens at node j, but it can take value 1 if the drone is returned to the truck at node j. In this formulation, constraints (40) are necessary to avoid loops, i.e. sorties that start and return at the same node.

Locker constraints Locker constraints (9)–(12) are inherited from the previous formulations.

Variable constraints

$$z_i \in \{0, 1\} \quad i \in V \tag{48}$$

Variables x^T, y, x^L, x^R, and z are defined binary in (16), (18), (37), (38), and (48), respectively. Variables t are set as non-negative variables in (20).

Polynomial-size families of valid inequalities Also for Formulation Three, we introduce some polynomial-size families of valid inequalities to strengthen the model. In particular, we include (21), (25), (39)–(44) and the following new ones.

$$\sum_{j \in V_C} x^R_{ji} \leq z_i \quad i \in V \tag{49}$$

$$\sum_{j \in V} x^L_{ji} \leq 1 - z_i \quad i \in V_C \tag{50}$$

$$z_i \leq \sum_{j \in V} x^T_{ij} \quad i \in V \tag{51}$$

According to constraints (49), a rendezvous can only occur at node $i \in V$ if $z_i = 1$. The constraints (50) state that a drone can only serve a node $i \in V_C$ if $z_i = 0$. Inequalities (51) force $z_i = 0$ if the corresponding node i is not visited by the truck. All of the described polynomial-size families of inequalities improved the formulation's average solving times (see, e.g., Section 7).

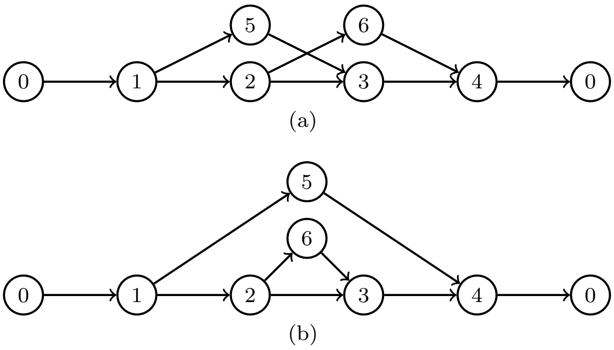

Fig. 2: Examples of infeasible solutions caused by crossing sorties.

4.4 Formulation Four

The model described in this section is derived from Formulation Three, but without using the z variables and the related constraints to avoid solutions with infeasible crossing sorties (see, for example, Figure 2), which are instead separated dynamically. Note that this idea is based on the crossing sorties elimination constraints proposed by Dell'Amico et al. [20] where the authors separate the infeasible solutions dynamically by proposing a procedure that is based on a depth-first exploration of integer and fractional solutions (see Section 6). Before explaining the crossing sorties elimination constraint, we highlight that constraints (31) and (32) are needed to avoid other infeasibilities, such as launches of the drone from nodes served by the drone itself, and are thus included in Formulation Four.

Crossing sorties elimination constraints To avoid crossing sorties, we need to impose the inequalities (52), that we report in its tournament version hereafter. Let $i \in V$ and $l \in V_C$ be the starting and ending vertices of the truck path P from i to l, i.e., $P = \{v(1), v(2), \dots, v(q)\}$ with $v(1) = i, v(q) = l$, $i \neq l$. Along the path P, we assume that two sortie launches defined by $x^L_{ij} > 0$ and $x^L_{lm} > 0$ occur and that there is a node $k \notin P$ such that $x^R_{jk} > 0$. In this case, the second sortie starts before the first sortie is terminated, and the following "tournament" crossing sorties elimination is violated:

$$\sum_{r=1}^{|P|-1} \sum_{s=r+1}^{|P|} x^T_{v(r)v(s)} + x^L_{v(1),j} + x^L_{v(|P|),m} + \sum_{\substack{k \in V \\ k \notin P}} x^R_{jk} \leq |P| + 1 \quad P \in \mathcal{P}, j, m \in V_C, j, m \notin P \tag{52}$$

where $\mathcal{P}$ defines the set of all the paths with the described characteristics.

The complete Formulation Four is thus made by (2), (9)–(13), (16), (18), (20), (21), (25), (27)– (32), (35)–(44), and (52). All the fixing on variables inherited from Formulation Three are applied to Formulation Four as well.

5 Variants

5.1 Unbounded endurance

If a drone battery limit is not required, one could set $E = \infty$ and use the proposed formulations as reported. However, in order to obtain better performing algorithms, it is preferable to remove all the constraints and variable settings due to the drone's endurance. For Formulation Two, Formulation Three, and Formulation Four, we must remove constraints (36) and avoid using inequalities (44). Moreover, the following variable settings must be avoided: $x^L_{ij} = 0$ if $\tau^D_{ij} > E, i \in V, j \in V_C$ and $x^R_{ij} = 0$ if $\tau^D_{ij} + \sigma^R > E, i \in V_C, j \in V$. To update Formulation One to solve the new variant, one must remove constraints (15) and avoid using inequalities (23) and (24).

5.2 UGV or UAV with no hovering needed: the TSPDL 'Wait'

In some applications, the UAV is allowed to wait on the ground and is not required to hover and consume battery while waiting. In others, the drone is a UGV, and thus it can wait without consuming the battery. In such cases, that we call the TSPDL 'Wait', one can only avoid the sorties whose length exceeds the endurance without considering the time of the truck's path traveled in the meantime. To do so, for Formulation Two, Formulation Three, and Formulation Four, one must remove constraints (36) but impose (44). The following variable fixing is also applied: $x^L_{ij} = 0$ if $\tau^D_{ij} > E, i \in V, j \in V_C$ and $x^R_{ij} = 0$ if $\tau^D_{ij} + \sigma^R > E, i \in V_C, j \in V$. For Formulation One, one must remove (15) but impose (24), whereas (23) can be used as a valid inequality.

6 Branch-and-cut implementation

Formulation Four requires a B&C implementation since it includes constraints (52), that are exponentially many. To separate them, we use the separation method proposed by Dell'Amico et al. [23] and Dell'Amico et al. [18] for similar constraints: at a given node of the branching tree, we consider the residual graph $G' = (N, A')$ obtained from G by selecting the only arcs associated with a non-zero variable (x^T, x^L, x^R) in the linear relaxation of the model and associate to these arcs the value of the corresponding variable solution in the linear relaxation. For the crossing sorties elimination constraints, we explore the graph starting from depot vertex 0 in a highest first fashion, which means that we build a truck path from the depot by following first the arc leaving the previous vertex with the highest value associated with it (namely, the value of x^T corresponding to that arc). At each new vertex in the path, we check if a crossing sortie is happening, that is, that at least two arcs associated with a launch (and hence to variables x^L) have left one vertex of the path before a rendezvous arc (associated with a variable x^L) has returned to the truck path. When a crossing sortie is found, if any, and the corresponding constraint (31) is violated, then a new cut is identified. If no crossing sorties are found and the path being built returns to the depot (node d) or if there are no more arcs to explore, we backtrack to the previous node with exiting arcs and keep exploring the graph by selecting the arc with the next best x^T value. If no crossing sortie is found after exploring the entire residual graph, then the current solution is feasible. After preliminary computational tests, we observed that Formulation

Four benefited from separating these constraints only for integer solutions. The overall cut separation procedure in this case has a time complexity of $O(|A|)$. We use Gurobi 9.5.2's B&C framework, which solves the linear relaxation of a MILP model at each node of the enumeration tree and then invokes user-developed separation procedures to possibly add cuts. As a branching rule, we chose strong branching, a well-known variable selection rule that involves determining which of the candidate variables provides the best improvement to the objective function before branching on them in order to result in a small search tree (see, e.g, Yang et al. [86]). Using a simple greedy heuristic, we obtain an initial solution at the root node of the B&C procedure. The integrality tolerance was set to 10e-10.

7 Computational Results

In this section, we show the computational results obtained by running the proposed methods on a single thread of an Intel(R) Core(TM) i7-10700 CPU @ 2.90 GHz with 32.00 GB of RAM under Windows 11. All of our methods were implemented in C++ and the MILP solver used was Gurobi 9.5.2.

7.1 Benchmark instances

In the following, we describe the instances used. Note that the instances that include newly generated information can be found at the link: `https://github.com/stenov/TSPDL/tree/main`.

7.1.1 Instance set MC10_1

We first tested our formulations by using a well-known set of benchmark instances proposed by Murray and Chu's [57] for the FSTSP, which we adapted to include lockers. They proposed 36 randomly generated benchmark instances with 10 customers and the endurance E set to either 20 or 40 minutes, giving 72 instances. The customers are randomly distributed across an eight-mile square region, while the depot is located in four different positions: 'a' indicates that the depot is near the center of gravity of the customers (instances 'v1', 'v5', and 'v9'); while 'b' (instances 'v2', 'v6', and 'v10'), 'c' (instances 'v3', 'v7', and 'v11'), and 'd' (instances 'v4', 'v8', and 'v12') have the following (x, y) coordinates, respectively: (4.0,2.7), (4.0,0.0), and (4,-2.7). The other nodes are positioned in the same location in instance sets '37', '40', or '43'. The drone travel times are based on Euclidean distances and the speed is selected between 15 miles/h (instances 'v1', ..., 'v4'), 25 miles/h (instances 'v5', ..., 'v8'), and 35 miles/h (instances 'v9', ..., 'v12'); while the truck travel times are based on Manhattan distances and the truck speed was assumed to be 25 miles/h.

To accommodate a locker in those instances, we considered the last customer node to be the locker, with a capacity of $\nu_b = 5$, ending up with nine customers and one locker. We set $\alpha_1 = \alpha_2 = 1$, $\sigma^L = \sigma^R = 1$, and $\gamma_i = 1, i \in V_C$. The distance between the locker and the customer determines the cost of serving a customer from that locker: $c_{ib} = \tau^T_{ib}$.

All customers can be served by the locker, so we set $\delta_{ib} = 1$ for all $i \in V_C, b \in V_L$. Customers who could not be served by the drone in the original instances cannot be served by the drone in our instances, as reflected by the value of $\beta_i, i \in V_C$. In particular, in instances '37' all but one customer can be served with the drone, while instances '40' and '43' count two customers that cannot be served by the drone.

We refer to this set of instances as set *MC10_1*.

7.1.2 Instance set MC10_2

We used two additional instance sets to test the best of the proposed formulations. The first set, *MC10_2*, is derived from set *MC10_1*, to which we added a second locker, resulting in instances with nine customers and two lockers. The second locker's coordinates were generated randomly between 0 and the maximum 'x' and 'y' coordinates among the other nodes for each instance subset, namely '37', '40', or '43'. The distance calculation, as well as all of the other parameters, are inherited from set *MC10_1*.

7.1.3 Instance set MC20

Similarly to what we did previously, we selected 20 instances from the set of randomly generated instances with 20 customers proposed by Murray and Chu's [57] for the parallel drone scheduling TSP, which we adapted to include lockers.

There are a total of 20 customers in these instances. We assumed that the last customer node was a locker with a capacity of $\nu_b = 5$, and we randomly generated the coordinates of the other two nodes between 0 and the maximum values of the other nodes' coordinates, yielding instances with 19 customers and up to three lockers. The endurance E was set to 20 or 40 minutes, resulting in 40 instances. The drone travel times are calculated using Euclidean distances and a speed of 25 miles per hour, whereas the truck travel times are calculated using Manhattan distances and a speed of 25 miles per hour.

We set $\alpha_1 = \alpha_2 = 1$, $\sigma^L = \sigma^R = 1$, and $\gamma_i = 1, i \in V_C$. The cost of serving a customer from that locker is determined by the distance between the locker and the customer: $c_{ib} = \tau^T_{ib}$. All customers can be served by the locker, so we set $\delta_{ib} = 1$ for all $i \in V_C, b \in V_L$. Customers who could not be served by the drone in the original instances (between 80% and 90% of customers) cannot be served by the drone in our instances, as reflected by the value of $\beta_i, i \in V_C$.

We refer to this set of instances as set *MC20*.

7.2 Results analysis

In this section, we compare the performance of the mathematical formulations described in Section 4 and analyze the results obtained for the TSPDL and its variants considered in Section 5. Moreover, we test the best-performing of our methods with larger instances. Lastly, we provide some useful managerial insights.

7.2.1 Comparison of the formulations

We show a comparison, based on instance set *MC10_1*, of the proposed formulations when solving the TSPDL and the other variants. For the sake of conciseness, we call Formulation One, Formulation Two, Formulation Three, and Formulation Four as F1, F2, F3, and F4, in the following. In Table 2, we show the running times (in seconds) for the four presented formulations on the different variants of the TSPDL solved, organized by endurance, drone speed, depot location, and graph type. Notice that when $E = \infty$ the solutions of the TSPDL and the TSPDL 'Wait' variants coincide. The online appendix [8] contains detailed results for each problem, instance, and formulation.

Every formulation could solve all instances to optimality, but according to the reported values, F4 was almost always the fastest formulation on average. On average, problems with endurance $E = 20$ are easier to solve than those with endurance $E = 40$, which could be due to fewer feasible sorties and thus a more restricted solution space. On the contrary, because fewer constraints are imposed, instances with $E = \infty$ may be easier to solve than others. The TSPDL 'Wait' could be solved in a faster way than the TSPDL. As shown in the online appendix [8], all formulations have very similar linear relaxation values, with F1 having slightly better values on average than the other formulations; however, the lower bounds at the root node are improved further for the other formulations once the presolve phase is included, with F3 and F4 displaying the best bounds.

The instances in which the speed of the drone is the same as that of the truck are normally the hardest to solve. The cases with the depot in 'c' are obviously the simplest to handle when taking the depot location into account. However, the difference on the graphs is what matters most when determining how difficult a set of instances is; in fact, instances of type '40' can always be handled in less time than the other two sets.

Table 2: Results on the instance set *MC10_1* grouped by drone speed, depot location, and graph type.

	speed	sec F1	sec F2	sec F3	sec F4	network	sec F1	sec F2	sec F3	sec F4	depot	sec F1	sec F2	sec F3	sec F4
E=20	15	43.23	11.09	6.18	**3.15**	37	68.15	197.76	88.42	**60.07**	a	28.83	63.42	41.14	**24.01**
	25	125.35	167.53	97.76	**57.72**	40	33.66	19.06	10.02	**6.59**	b	54.91	57.77	51.65	**29.49**
	35	44.63	121.88	44.78	**29.69**	43	111.41	83.68	50.28	**23.89**	c	43.19	61.53	33.51	**25.35**
											d	157.35	217.95	72.00	**41.89**
avg.												71.07	100.17	49.57	**30.18**
E=40	15	157.70	109.68	86.13	**38.26**	37	**56.82**	159.74	149.21	67.54	a	39.50	50.56	39.22	**28.50**
	25	**31.49**	131.29	100.41	65.80	40	78.66	38.11	14.58	**14.11**	b	34.44	57.92	38.16	**27.34**
	35	**22.71**	63.24	45.67	27.46	43	76.42	106.36	68.41	**49.88**	c	55.35	89.99	67.96	**38.29**
											d	153.24	207.14	164.28	**81.24**
avg.												70.63	101.40	77.40	**43.84**
E=∞	15	157.49	86.99	41.23	**24.24**	37	**28.18**	127.78	62.88	41.96	a	16.78	34.35	23.17	**13.62**
	25	**15.07**	88.42	51.95	33.22	40	94.89	30.71	11.52	**9.16**	b	23.56	39.03	24.60	**16.95**
	35	**9.19**	47.69	18.78	21.18	43	58.67	64.61	37.54	**27.52**	c	28.90	44.48	38.07	**18.49**
											d	173.09	179.59	63.43	**55.79**
avg.												60.58	74.37	37.32	**26.21**
E=20 Wait	15	6.23	4.87	2.75	**1.90**	37	**12.78**	76.56	43.27	30.26	a	**10.41**	45.15	32.86	16.59
	25	**20.93**	83.73	54.40	36.03	40	7.67	7.66	3.30	**2.48**	b	**12.96**	36.14	22.66	16.28
	35	**9.66**	48.38	19.55	14.88	43	**16.38**	52.77	30.12	20.06	c	**8.83**	46.74	22.46	15.27
											d	**16.89**	54.62	24.28	22.28
avg.												**12.27**	45.66	25.57	17.60
E=40 Wait	15	112.39	79.20	37.12	**24.49**	37	**34.04**	115.83	53.98	42.41	a	**19.37**	33.83	22.94	13.04
	25	**17.22**	91.03	48.51	35.47	40	41.40	36.12	10.48	**9.93**	b	**19.83**	46.44	23.22	20.32
	35	**11.66**	47.51	18.64	21.32	43	65.83	65.79	39.82	**28.94**	c	31.11	50.22	35.26	**17.70**
											d	118.05	159.83	57.61	**57.30**
avg.												47.09	72.58	34.76	**27.09**

7.2.2 Tests on larger instances and multiple lockers

In this section, we evaluate the impact of a larger number of available lockers and customers by testing the formulation that produced the best results, namely F4, on instance sets *MC10_2* and *MC20*.

In Table 3, we show the computing time in seconds needed by F4 to retrieve the proven optimal solution when solving the instances of set *MC10_2*, which considers nine customers and two lockers. The results for the several variants are grouped by endurance, drone speed, network type, and depot location. It can be noted that the computing times follow a similar behavior to that of the case with one locker, namely on instances *MC10_1*. However, as expected, the solving times are slightly longer when there are two lockers.

Table 3: Results of F4 on the instance set *MC10_2* grouped by drone speed, depot location, and graph type.

	speed	sec F4	network	sec F4	depot	sec F4
E=20	15	5.58	37	85.66	a	28.38
	25	129.73	40	8.35	b	49.96
	35	57.44	43	98.75	c	61.27
					d	117.40
avg.						64.25
E=40	15	93.51	37	204.23	a	42.02
	25	134.54	40	24.48	b	42.70
	35	83.31	43	82.65	c	86.88
					d	243.55
avg.						103.79
E=∞	15	34.15	37	50.99	a	20.22
	25	51.85	40	12.60	b	23.73
	35	24.36	43	46.77	c	26.33
					d	76.86
avg.						36.79
E=20 Wait	15	2.92	37	52.27	a	16.24
	25	65.44	40	4.92	b	35.85
	35	28.76	43	39.93	c	43.67
					d	33.73
avg.						32.37
E=40 Wait	15	49.67	37	61.69	a	20.00
	25	45.27	40	13.67	b	26.62
	35	24.29	43	43.86	c	24.77
					d	87.58
avg.						39.74

The results of F4 on the instance set *MC20* with one, two, or three available lockers and $E = 20$ or $E = 40$ are shown in Table 4. We display the number of available lockers (m) as well as the percentage average gap calculated as $\%gap = 100 \cdot (UB - LB)/UB$, where UB and LB are the upper and lower bounds of the problem at the end of the algorithm, respectively. Note that for each row, the average percentage gap is calculated across all 20 instances. Furthermore, we show the number of seconds required to obtain the proven optimal solution or the time limit of one hour if the optimal solution cannot be found, hence, the value under column *sec F4* is the average over all 20 instances of each row, including the time limit value for those instances that could not be solved to the proven optimality.

With $E = 20$, 23 of the 60 TSPLD instances could be solved to optimality, with an average percentage gap of 6.57. The number of solved instances increases to 28 in the TSPDL 'Wait' case, with an average percentage gap of 5.70. For the instances with $E = 40$ and $E = \infty$, however, no optima could be found within the time limit.

As a result, we can conclude that an increase in the number of customers and a larger endurance make the problem harder to solve, whereas an increase in the number of available lockers is less important. Furthermore, one can state that we have found the limit of formulation F4.

7.2.3 Analysis of the results and managerial insights

In this section, we discuss the results of the TSPDL and the studied variants with respect to the TSP, the *TSP with lockers* (TSPL), and the TSP-D (that we recall is the TSP with drone).

Table 4: Results of F4 on instance set *MC20* with 1, 2, or 3 available lockers.

	m	TSPDL %gap	sec F4	#opt	TSPDL 'Wait' %gap	sec F4	#opt
E=20	1	6.21	2338.72	8/20	5.23	2151.09	9/20
	2	6.21	2423.61	8/20	5.19	2189.06	10/20
	3	7.30	2692.66	7/20	6.68	2392.50	9/20
avg./tot.		6.57	2484.99	23/60	5.70	2244.21	28/60
E=40	1	16.37	3600.00	0/20	15.70	3600.00	0/20
	2	16.57	3600.00	0/20	15.99	3600.00	0/20
	3	17.12	3600.00	0/20	16.54	3600.00	0/20
avg./tot.		16.69	3600.00	0.00	16.08	3600.00	0/20
E=∞	1	14.21	3600.00	0/20			
	2	17.12	3600.00	0/20			
	3	15.54	3600.00	0/20			
avg./tot.		15.62	3600.00	0/20			

All the data is represented in boxplots with the following characteristics. The ends of the boxes are the 25th and 75th percentiles of the data distribution. The horizontal line in the box shows the median value of the distribution, and the cross represents the average. The inter quartile range (IQR) is obtained by subtracting the 25th percentile from the 75th. The lower outlier limit (the lower whisker) is calculated as $25\% - 1.5 \cdot IQR$, while the upper outlier limit (the upper whisker) is calculated as $75\% + 1.5 \cdot IQR$. The diamonds represent outlier values.

Figure 3 depicts the distribution of the $\%GapTSP = 100 \cdot (sol_{TSP} - sol_i)/sol_{TSP}$ with respect to three different drone endurance values. sol_{TSP} represents the optimal solution value of the TSP on the graph (a solution that does not visits the locker), whereas sol_i represents the value of the optimal solutions of the other problems solved, namely the TSPL (in which only truck and lockers are used for serving customers), the TSP-D (in which only truck and drone are available), the TSPDL, the TSPDL 'Wait' (called TSPDL-Wait in the figure), and the cases where a second lockers is considered (2L). Note that the TSPL and the TSPL-2L do not consider endurance and thus they are listed separately. This analysis is based on instance sets *MC10_1* and *MC10_2*. The introduction of lockers, with respect to the solution of the TSP, improves the solution value by up to 7%, which is in line with the results of Buzzega and Novellani [12], that show an average percentage improvement from 3.5% to 7.2% for the TSPL when no time window is considered. The introduction of a second locker improves the solution value with up to 10% with respect to the TSP one. The introduction of the drone allows the costs to decrease by 10% on average with a limited endurance value, and by more than 15% when the endurance is higher. The joint use of lockers and of a drone improves the value of the solution even more, especially when the endurance is limited. In the variant in which the drone consumes the battery only when traveling (namely TSPDL 'Wait'), the solution values can improve mainly when the endurance is restrictive, as expected. Because waiting is irrelevant when the endurance is unlimited, the optimal solutions for the TSPDL with $E = \infty$ and the TSPDL 'Wait' with $E = \infty$ are the same, so we report the same results in both cases. Finally, allowing the use of a drone and two lockers provides greater benefits, even more so in the 'Wait' case.

According to three values of endurance, three values of drone speed, and four depot locations, respectively, the Figures 4, 5, and 6 show in boxplots the number of customers served by the drone or the lockers in the solutions of the TSPL, the TSP-D, the TSPDL, and the variants of the TSPDL. The number of customers served via drone increases with the battery endurance and decreases slightly when the lockers are introduced. On the other hand, the customers served by lockers decrease more visibly when the drone is introduced, and even more visibly with large values of endurance. As expected, the number of sorties increases largely when the drone speed is increased, and thus the number of customers served via locker decreases. The number of sorties appears to be more sensitive to the depot's location than the number of customers served by lockers. With the addition of a second locker, the number of customers served by lockers increases slightly at the expense of sorties.

We now analyze what happens when considering a fixed cost for each customer served via locker, instead of a cost depending on the distance from the locker. In particular, we analyze the results of instances *MC10_1* with the following values: 0, 5, 10, 15, and 20.

In Figure 7, we show, in boxplots, the percentage gap between the solution of the TSP and that of the TSPDL with the different fixed cost values. Moreover, we show the difference between the TSPDL where the cost of serving a customer from a locker depends on the distance between the customer and the locker and the TSPDL with fixed costs. When the use of lockers is free of charge, the traveling costs can decrease by more than 50%, on average. A value that increases by a large amount if compared to the improvements given by the only use of the drone, which is roughly between 10 and 16% (see Figure 3). As the fixed costs rise, the solutions become more similar to those of the TSP-D, and the endurance effect becomes more apparent. Let us consider the difference between the cost of the TSPDL with fixed costs and the TSPDL with a cost that varies with the distance of the locker. We note that the solution values of the problems with a fixed cost that equals 10 have a very similar pattern with respect to the solution values of the TSPDL with variable costs. We also report that the average of the entries of the truck distance matrix is about 10 units in our instances. Thus, one could use this number as the maximum limit on the discount to apply to the deliveries from the lockers, being that afterwards the improvement is mostly given by the use of the drone. Note, moreover, that the average cost per customer served by locker in the TSPDL with costs linked to the distances is between 8 and 10 units in all cases. In Figure 8, we see that the number of lockers decreases with the increase of the fixed cost, as expected, and that this trend is quite rapid when the cost goes from 0 to 10, after the solutions are mostly similar to those of the TSPD. On the other hand, the number of customers served with the drone increased with the decrease of customers served via lockers.

This analysis suggests that the joint use of lockers and of a drone allows a remarkable decrease in travel times and costs and should be pursued by selecting an incentive for customers to use the lockers that is not too high.

Eventually, we analyze the influence of parameters α_1 and α_2 on the solution structure of instances *MC10_1*.

Figure 9 depicts the number of customers served by drones and lockers in relation to the value of α_1, which may be 0, 0.2, 0.4, 0.6, or 0.8, having set $\alpha_2 = \alpha_1 - 1$. Even as α_1 increases, several sorties are still performed, whereas when its cost coefficient is high, logically, very few customers are served at lockers, making the use of lockers more sensitive to the selection of these parameters. As a result, practitioners should once again incentivize lockers by selecting lower α_2 values.

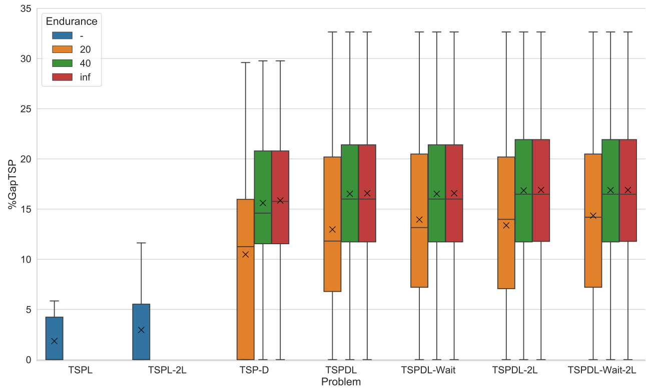

Fig. 3: Boxplots displaying the distributions of the percentage gap between the TSP solution and those of the TSPL, the TSP-D, the TSPDL, and 'Wait' variants with one or two lockers, grouped by endurance.

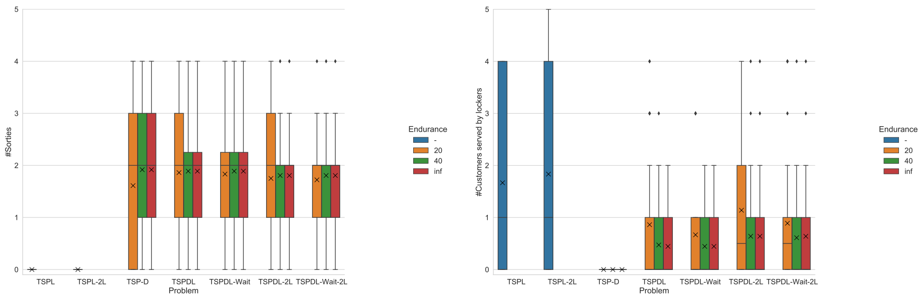

Fig. 4: Boxplots displaying the number of customers served via drone or lockers in the solution of the TSPL, the TSP-D, the TSPDL, and 'Wait' variants with one or two lockers, grouped by endurance.

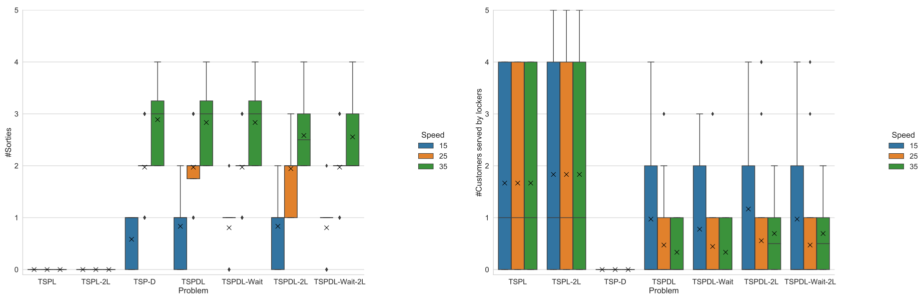

Fig. 5: Boxplots displaying the number of customers served via drone or lockers in the solution of the TSPL, the TSP-D, the TSPDL, and 'Wait' variants with one or two lockers, grouped by drone speed.

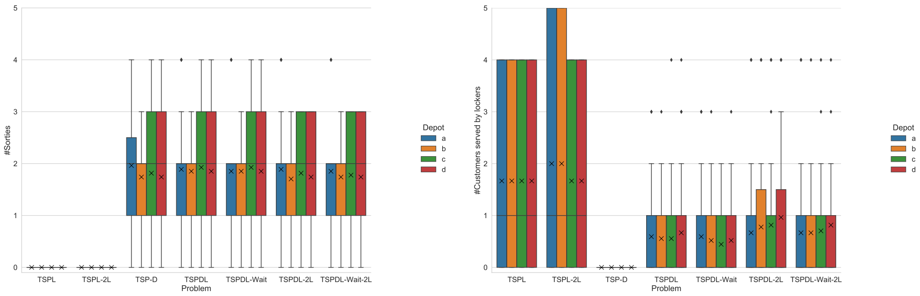

Fig. 6: Boxplots displaying the number of customers served via drone or lockers in the solution of the TSPL, the TSP-D, the TSPDL, and 'Wait' variants with one or two lockers, grouped by depot location.

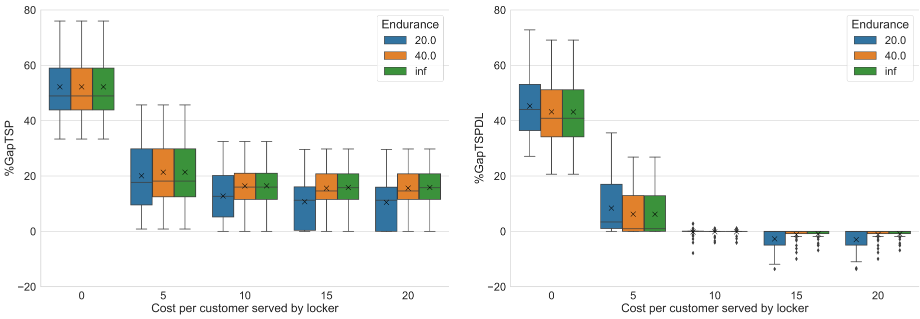

Fig. 7: Boxplots displaying the percentage gap between the solutions of the TSP and the TSPDL (with variable costs) with respect to those of the TSPDL with a fixed cost per customer served via locker with a value of 0, 5, 10, 15, 20, grouped by endurance.

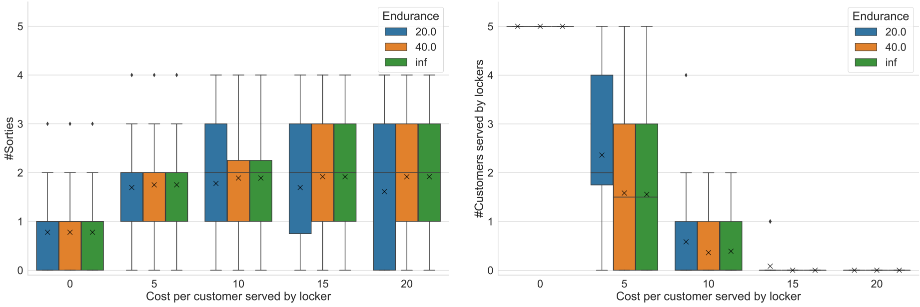

Fig. 8: Boxplots displaying the number of customers served via drone or lockers in the solutions of the TSPDL with a fixed cost per customer served via locker with a value of 0, 5, 10, 15, 20, grouped by endurance.

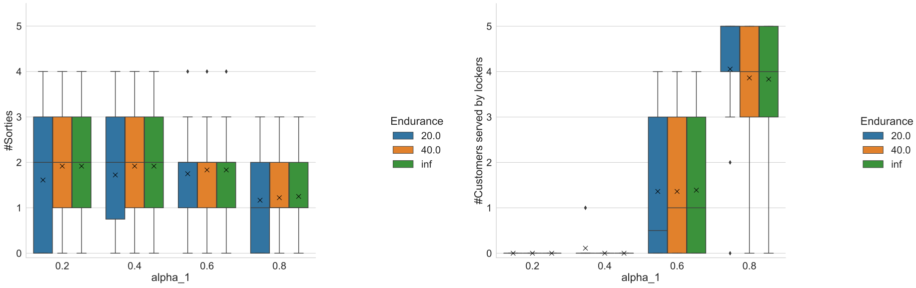

Fig. 9: Boxplots displaying the number of customers served via drone or lockers in the solutions of the TSPDL with α_1 having value 0.2, 0.4, 0.6, 0.8, and $\alpha_2 = 1 - \alpha_1$, grouped by endurance.

8 Conclusion

In this paper, we propose for the first time the traveling salesman problem with drone and lockers, a problem that considers the use of a drone (aerial or terrestrial) combined with a truck to efficiently deliver parcels either at a customer's home or at lockers.

To solve the problem and some of its variants, we propose four mixed integer programming formulations, one of which needs a branch-and-cut implementation. Several polynomial-size families of additional valid inequalities are also included in the models in order to speed up the solution time. We show experimentally that one of the proposed formulations is the fastest of all four on average.

The introduction of drones and lockers in last-mile parcel delivery was already shown to be profitable when done separately. Here we show that their combination can provide further improvements in costs. In particular, if the delivery company incurs a penalty when delivering to lockers as an incentive to the customers, its value should be limited to be effective. On the other hand, most of the time, the service at lockers is already convenient for the customers without an incentive because it allows them to have a more flexible delivery schedule, and thus the introduction of lockers can be even more profitable for the delivery companies.

Future research could concentrate on improving the newly proposed formulations by defining new valid inequalities to include in a branch-and-cut manner. Moreover, one could focus on similar problems, including time windows, pickup-and-delivery, lockers modeled with cells of different capacities, parcels with different sizes and weights, single or multiple loop sorties, the possibility to launch the drone before the service at a node, etc. A possible future line of research could focus on the development of dedicated metaheuristics that could solve large instances, because the proposed exact methods could only solve small instances to optimality. Larger instances could be solved optimally by developing dedicated branch-and-price algorithms.

References

1. Agatz, N., Bouman, P., Schmidt, M.: Optimization approaches for the traveling salesman problem with drone. Transportation Science **52**(4), 965–981 (2018)

2. Alibaba: Alibaba's drones deliver packages to islands. http://www.chinadaily.com.cn/business/2017-11/07/content_34230012.htm (2022). http://www.chinadaily.com.cn/business/2017-11/07/content_34230012.htm (Accessed: 2022-08-14)
3. Amazon: Amazon hub (2022). https://www.amazon.com/ulp/view (Accessed: 2022-10-24)
4. Amazon: Amazon primeair (2022). https://www.aboutamazon.com/news/transportation/amazon-prime-air-prepares-for-drone-deliveries (Accessed: 2022-08-14)
5. Aphabet: Wing - transforming the way goods are transported. https://x.company/projects/wing/ (2022). https://x.company/projects/wing/ (Accessed: 2022-08-14)
6. Boccia, M., Masone, A., Sforza, A., Sterle, C.: A column-and-row generation approach for the flying sidekick travelling salesman problem. Transportation Research Part C: Emerging Technologies **124**, 102,913 (2021)
7. Boccia, M., Masone, A., Sforza, A., Sterle, C.: An exact approach for a variant of the fs-tsp. Transportation Research Procedia **52**, 51–58 (2021)
8. Boschetti, M.A., Novellani, S.: Last-mile delivery with drone and lockers - online appendix (2023)
9. Bouman, P., Agatz, N., Schmidt, M.: Dynamic programming approaches for the traveling salesman problem with drone. Networks **72**(4), 528–542 (2018)
10. Boysen, N., Fedtke, S., Schwerdfeger, S.: Last-mile delivery concepts: a survey from an operational research perspective. Or Spectrum **43**(1), 1–58 (2021)
11. Bug, P., Gordon, N., Staudenmaier, A.S.: Fashion Business Case Study on the German Click & Collect Situation, pp. 139–163. Springer Singapore, Singapore (2018). DOI 10.1007/978-981-10-7007-5_9
12. Buzzega, G., Novellani, S.: Last mile deliveries with lockers: formulations and algorithms. Soft Computing pp. 1–19 (2022)
13. Cavani, S., Iori, M., Roberti, R.: Exact methods for the traveling salesman problem with multiple drones. Transportation Research Part C: Emerging Technologies **130**, 103,280 (2021)
14. Chang, Y.S., Lee, H.J.: Optimal delivery routing with wider drone-delivery areas along a shorter truck-route. Expert Systems with Applications **104**, 307–317 (2018)
15. Chung, S.H., Sah, B., Lee, J.: Optimization for drone and drone-truck combined operations: A review of the state of the art and future directions. Computers & Operations Research **123**, 105,004 (2020)
16. Clarke, G., Wright, J.W.: Scheduling of vehicles from a central depot to a number of delivery points. Operations research **12**(4), 568–581 (1964)
17. Daknama, R., Kraus, E.: Vehicle routing with drones. arXiv preprint arXiv:1705.06431 (2017)
18. Dell'Amico, M., Montemanni, R., Novellani, S.: Modeling the flying sidekick traveling salesman problem with multiple drones. Networks **78**(3), 303–327 (2021)
19. Dell'Amico, M., Montemanni, R., Novellani, S.: A random restart local search matheuristic for the flying sidekick traveling salesman problem. In: 2021 The 8th International Conference on Industrial Engineering and Applications (Europe), pp. 205–209 (2021)
20. Dell'Amico, M., Montemanni, R., Novellani, S.: Exact models for the flying sidekick traveling salesman problem. International Transactions in Operational Research **29**(3), 1360–1393 (2022)
21. Dell'Amico, M., Montemanni, R., Novellani, S.: Pickup and delivery with lockers. Transportation Research Part C: Emerging Technologies **148**, 104,022 (2023)
22. Dell'Amico, M., Montemanni, R., Novellani, S.: Algorithms based on branch and bound for the flying sidekick traveling salesman problem. Omega **104**, 102,493 (2021)
23. Dell'Amico, M., Montemanni, R., Novellani, S.: Drone-assisted deliveries: New formulations for the flying sidekick traveling salesman problem. Optimization Letters **15**(5), 1617–1648 (2021)
24. DHL: Parcelcopter: Dhl's drone. https://www.dhl.com/discover/en-my/business/business-ethics/parcelcopter-drone-technology (2022). https://www.dhl.com/discover/en-my/business/business-ethics/parcelcopter-drone-technology (Accessed: 2022-08-14)
25. Di Puglia Pugliese, L., Macrina, G., Guerriero, F.: Trucks and drones cooperation in the last-mile delivery process. Networks **78**(4), 371–399 (2021)
26. Dumez, D., Lehuédé, F., Péton, O.: A large neighborhood search approach to the vehicle routing problem with delivery options. Transportation Research Part B: Methodological **144**, 103–132 (2021)

27. Enthoven, D.L., Jargalsaikhan, B., Roodbergen, K.J., uit het Broek, M.A., Schrotenboer, A.H.: The two-echelon vehicle routing problem with covering options: City logistics with cargo bikes and parcel lockers. Computers & Operations Research **118**, 104,919 (2020)
28. Foster, B.A., Ryan, D.M.: An integer programming approach to the vehicle scheduling problem. Journal of the Operational Research Society **27**(2), 367–384 (1976)
29. de Freitas, J.C., Penna, P.H.V.: A randomized variable neighborhood descent heuristic to solve the flying sidekick traveling salesman problem. Electronic Notes in Discrete Mathematics **66**, 95–102 (2018)
30. de Freitas, J.C., Penna, P.H.V.: A variable neighborhood search for flying sidekick traveling salesman problem. Int. Trans. Oper. Res **27**, 267–290 (2020)
31. Grabenschweiger, J., Doerner, K.F., Hartl, R.F., Savelsbergh, M.W.P.: The vehicle routing problem with heterogeneous locker boxes. Central European Journal of Operations Research **29**, 113–142 (2021)
32. Ha, Q.M., Deville, Y., Pham, Q.D., Hà, M.H.: Heuristic methods for the traveling salesman problem with drone. Comput. Sci (2015)
33. Ha, Q.M., Deville, Y., Pham, Q.D., Hà, M.H.: On the min-cost traveling salesman problem with drone. Transportation Research Part C: Emerging Technologies **86**, 597–621 (2018)
34. Ha, Q.M., Deville, Y., Pham, Q.D., Hà, M.H.: A hybrid genetic algorithm for the traveling salesman problem with drone. Journal of Heuristics **26**, 219–247 (2020)
35. Hu, M., Liu, W., Peng, K., Ma, X., Cheng, W., Liu, J., Li, B.: Joint routing and scheduling for vehicle-assisted multi-drone surveillance. IEEE Internet of Things Journal (2018)
36. JD.com: Jd.com's drone delivery program takes flight in rural china. `http://corporate.jd.com/whatIsNewDetail?contentCode=6IhXLeeSAFLjLLlyuZatDA` (2022). `http://corporate.jd.com/whatIsNewDetail?contentCode=6IhXLeeSAFLjLLlyuZatDA` (Accessed: 2022-08-14)
37. Jeong, H.Y., Lee, S.: Optimization of vehicle-carrier routing: Mathematical model and comparison with related routing models. Procedia Manufacturing **39**, 307–313 (2019)
38. Jiang, L., Dhiaf, M., Dong, J., Liang, C., Zhao, S.: A traveling salesman problem with time windows for the last mile delivery in online shopping. International Journal of Production Research pp. 1–12 (2019)
39. JLL: How can parcel lockers improve last-mile delivery?, `https://www.jll.fi/fi/trendit-ja-tutkimukset/kaupungit/how-can-parcel-lockers-improve-last-mile-delivery` (2022). (Accessed: 2022-08-14)
40. Joerss, M., Schröder, J., Neuhaus, F., Klink, C., Mann, F.: Parcel delivery - the future of last mile. Travel, Transport and Logistics (2016)
41. Karak, A., Abdelghany, K.: The hybrid vehicle-drone routing problem for pick-up and delivery services. Transportation Research Part C: Emerging Technologies **102**, 427–449 (2019)
42. Kerr, I., Różycki, M.: In-store parcel lockers get up and running, `https://www.parcelandpostaltechnologyinternational.com/analysis/in-store-parcel-lockers-get-up-and-running.html` (2022). (Accessed: 2022-03-22)
43. Khoufi, I., Laouiti, A., Adjih, C.: A survey of recent extended variants of the traveling salesman and vehicle routing problems for unmanned aerial vehicles. Drones **3**(3), 66 (2019)
44. Kim, W., Wang, X.C.: The adoption of alternative delivery locations in new york city: Who and how far? Transportation Research Part A: Policy and Practice **158**, 127–140 (2022). DOI https://doi.org/10.1016/j.tra.2022.02.006
45. Kitjacharoenchai, P., Ventresca, M., Moshref-Javadi, M., Lee, S., Tanchoco, J.M., Brunese, P.A.: Multiple traveling salesman problem with drones: Mathematical model and heuristic approach. Computers & Industrial Engineering **129**, 14–30 (2019)
46. Li, H., Chen, J., Wang, F., Bai, M.: Ground-vehicle and unmanned-aerial-vehicle routing problems from two-echelon scheme perspective: A review. European Journal of Operational Research **294**(3), 1078–1095 (2021)
47. Li, J., Ensafian, H., Bell, M.G., Geers, D.G.: Deploying autonomous mobile lockers in a two-echelon parcel operation. Transportation Research Part C: Emerging Technologies **128**, 103,155 (2021)
48. Lin, Y., Wang, Y., Lee, L.H., Chew, E.P.: Profit-maximizing parcel locker location problem under threshold luce model. Transportation Research Part E: Logistics and Transportation Review **157**, 102,541 (2022). DOI https://doi.org/10.1016/j.tre.2021.102541

49. Logmore: Last mile and parcel lockers as logistics customer satisfaction factors, https://www.logmore.com/post/last-mile-customer-experience (2022). (Accessed: 2022-08-14)
50. Luo, Z., Poon, M., Zhang, Z., Liu, Z., Lim, A.: The multi-visit traveling salesman problem with multi-drones. Transportation Research Part C: Emerging Technologies **128**, 103,172 (2021)
51. Macrina, G., Pugliese, L.D.P., Guerriero, F., Laporte, G.: Drone-aided routing: A literature review. Transportation Research Part C: Emerging Technologies **120**, 102,762 (2020)
52. Mancini, S., Gansterer, M.: Vehicle routing with private and shared delivery locations. Computers & Operations Research **113**, 105,361 (2021)
53. Mitrea, I.A., Zenezini, G., De Marco, A., Ottaviani, F.M., Delmastro, T., Botta, C.: Estimating e-consumers' attitude towards parcel locker usage. In: 2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC), pp. 1731–1736. IEEE (2020)
54. Morganti, E., Seidel, S., Blanquart, C., Dablanc, L., Lenz, B.: The impact of e-commerce on final deliveries: alternative parcel delivery services in france and germany. Transportation Research Procedia **4**, 178–190 (2014)
55. Moshref-Javadi, M., Hemmati, A., Winkenbach, M.: A truck and drones model for last-mile delivery: A mathematical model and heuristic approach. Applied Mathematical Modelling **80**, 290–318 (2020)
56. Moshref-Javadi, M., Lee, S., Winkenbach, M.: Design and evaluation of a multi-trip delivery model with truck and drones. Transportation Research Part E: Logistics and Transportation Review **136**, 101,887 (2020)
57. Murray, C.C., Chu, A.G.: The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery. Transportation Research Part C: Emerging Technologies **54**, 86–109 (2015)
58. Murray, C.C., Raj, R.: The multiple flying sidekicks traveling salesman problem: Parcel delivery with multiple drones. Transportation Research Part C: Emerging Technologies **110**, 368–398 (2020)
59. Oliveira, W.J.P., dos Santos, A.G.: Last mile delivery with lockers: Formulation and heuristic. In: ICEIS (1), pp. 460–467 (2020)
60. Orenstein, I., Raviv, T., Sadan, E.: Flexible parcel delivery to automated parcel lockers: models, solution methods and analysis. EURO Journal on Transportation and Logistics **8**(5), 683–711 (2019)
61. Otto, A., Agatz, N., Campbell, J., Golden, B., Pesch, E.: Optimization approaches for civil applications of unmanned aerial vehicles (uavs) or aerial drones: A survey. Networks **72**(4), 411–458 (2018)
62. Peng, K., Du, J., Lu, F., Sun, Q., Dong, Y., Zhou, P., Hu, M.: A hybrid genetic algorithm on routing and scheduling for vehicle-assisted multi-drone parcel delivery. IEEE Access **7**, 49,191–49,200 (2019)
63. Poikonen, S., Campbell, J.F.: Future directions in drone routing research. Networks **77**(1), 116–126 (2021)
64. Poikonen, S., Golden, B.: Multi-visit drone routing problem. Computers & Operations Research **113**, 104,802 (2020)
65. Poikonen, S., Golden, B., Wasil, E.A.: A branch-and-bound approach to the traveling salesman problem with a drone. INFORMS Journal on Computing **31**(2), 335–346 (2019)
66. Poikonen, S., Wang, X., Golden, B.: The vehicle routing problem with drones: Extended models and connections. Networks **70**(1), 34–43 (2017)
67. Rabe, M., Chicaiza-Vaca, J., González-Feliu, J.: System behavior for automated parcel lockers as an urban logistics delivery scheme
68. Risberg, A.: A systematic literature review on e-commerce logistics: towards an e-commerce and omni-channel decision framework. The International Review of Retail, Distribution and Consumer Research pp. 1–25 (2022)
69. Roberti, R., Ruthmair, M.: Exact methods for the traveling salesman problem with drone. Transportation Science **55**(2), 315–335 (2021)
70. Rohmer, S., Gendron, B.: A guide to parcel lockers in last mile distribution–highlighting challenges and opportunities from an or perspective. Technical Report (2020)
71. Rojas Viloria, D., Solano-Charris, E.L., Muñoz-Villamizar, A., Montoya-Torres, J.R.: Unmanned aerial vehicles/drones in vehicle routing problems: a literature review. International Transactions in Operational Research **28**(4), 1626–1657 (2021)

72. Sacramento, D., Pisinger, D., Ropke, S.: An adaptive large neighborhood search metaheuristic for the vehicle routing problem with drones. Transportation Research Part C: Emerging Technologies **102**, 289–315 (2019)
73. Salama, M., Srinivas, S.: Joint optimization of customer location clustering and drone-based routing for last-mile deliveries. Transportation Research Part C: Emerging Technologies **114**, 620–642 (2020)
74. Schermer, D., Moeini, M., Wendt, O.: A matheuristic for the vehicle routing problem with drones and its variants. Transportation Research Part C: Emerging Technologies **106**, 166–204 (2019)
75. Schermer, D., Moeini, M., Wendt, O.: A branch-and-cut approach and alternative formulations for the traveling salesman problem with drone. Networks **76**(2), 164–186 (2020)
76. Seifried, K.: The traveling salesman problem with one truck and multiple drones. Available at SSRN 3389306 (2019)
77. Sitek, P., Wikarek, J.: Capacitated vehicle routing problem with pick-up and alternative delivery (cvrppad): model and implementation using hybrid approach. Annals of Operations Research **273**(1-2), 257–277 (2019)
78. Sitek, P., Wikarek, J., Rutczyńska-Wdowiak, K., Bocewicz, G., Banaszak, Z.: Optimization of capacitated vehicle routing problem with alternative delivery, pick-up and time windows: A modified hybrid approach. Neurocomputing (2020)
79. Statista: Retail e-commerce sales worldwide from 2014 to 2026 (in billion u.s. dollars), `https://www.statista.com/statistics/379046/worldwide-retail-e-commerce-sales/` (2023). (Accessed: 2023-09-06)
80. Symonds, D.: Dhl expands parcel locker network in germany, `https://www.parcelandpostaltechnologyinternational.com/news/lockers-pudo/dhl-expands-parcel-locker-network-in-germany.html` (2022). (Accessed: 2022-03-22)
81. Tilk, C., Olkis, K., Irnich, S.: The Last-mile Vehicle Routing Problem with Delivery Options. OR Spectrum (2021)
82. Tu, P.A., Dat, N.T., Dung, P.Q.: Traveling salesman problem with multiple drones. In: Proceedings of the Ninth International Symposium on Information and Communication Technology, pp. 46–53. ACM (2018)
83. Vakulenko, Y., Hellström, D., Hjort, K.: What's in the parcel locker? exploring customer value in e-commerce last mile delivery. Journal of Business Research **88**, 421–427 (2018). DOI https://doi.org/10.1016/j.jbusres.2017.11.033. URL `https://www.sciencedirect.com/science/article/pii/S0148296317304794`
84. Wang, X., Poikonen, S., Golden, B.: The vehicle routing problem with drones: Several worst-case results. Optimization Letters **11**(4), 679–697 (2017)
85. Wang, Z., Sheu, J.B.: Vehicle routing problem with drones. Transportation research part B: methodological **122**, 350–364 (2019)
86. Yang, Y., Boland, N., Dilkina, B., Savelsbergh, M.: Learning generalized strong branching for set covering, set packing, and 0–1 knapsack problems. European Journal of Operational Research **301**(3), 828–840 (2022)
87. Yu, V.F., Susanto, H., Yeh, Y.H., Lin, S.W., Huang, Y.T.: The vehicle routing problem with simultaneous pickup and delivery and parcel lockers. Mathematics **10**(6), 920 (2022)
88. Yurek, E.E., Ozmutlu, H.C.: A decomposition-based iterative optimization algorithm for traveling salesman problem with drone. Transportation Research Part C: Emerging Technologies **91**, 249–262 (2018)
89. Zhou, F., He, Y., Zhou, L.: Last mile delivery with stochastic travel times considering dual services. IEEE Access **7**, 159,013–159,021 (2019)
90. Zhou, L., Baldacci, R., Vigo, D., Wang, X.: A multi-depot two-echelon vehicle routing problem with delivery options arising in the last mile distribution. European Journal of Operational Research **265**(2), 765–778 (2018)