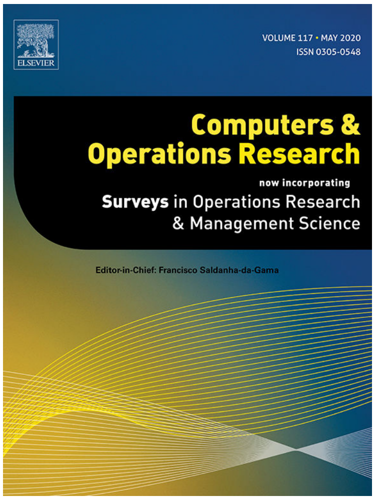


The Traveling Salesman Problem with Release Dates and Drone Resupply

Juan C. Pina-Pardo, Daniel F. Silva, Alice E. Smith

PII: S0305-0548(20)30287-2
DOI: https://doi.org/10.1016/j.cor.2020.105170
Reference: CAOR 105170

To appear in: *Computers and Operations Research*

Received Date: 24 August 2020
Accepted Date: 4 December 2020

Please cite this article as: J.C. Pina-Pardo, D.F. Silva, A.E. Smith, The Traveling Salesman Problem with Release Dates and Drone Resupply, *Computers and Operations Research* (2020), doi: https://doi.org/10.1016/j.cor.2020.105170

The Traveling Salesman Problem with Release Dates and Drone Resupply

Juan C. Pina-Pardo[a], **Daniel F. Silva**[b]*, **Alice E. Smith**[b]

[a]School of Industrial Engineering, Pontificia Universidad Catolica de Valparaiso,
Avenida Brasil 2241, Valparaiso 2362807, Chile

[b]Department of Industrial and Systems Engineering, Auburn University,
3301 Shelby Center, Auburn, AL 36849, USA

Abstract

This paper introduces the Traveling Salesman Problem with Release Dates and Drone Resupply, which consists of finding a minimum time route for a single truck that can receive newly available orders en route via a drone sent from the depot. We assume that each order's release date is known at the time of delivery planning. This context is common for many applications, notably last-mile logistics. We develop a Mixed-Integer Linear Program and a solution approach for larger instances based on decomposing the problem into the truck-routing and the drone-resupply decisions. Numerical experiments show that using drones for resupply can reduce the total delivery time by up to 20%. Additionally, experiments show that the decomposition can rapidly obtain high-quality solutions. For instances of 10 and 15 customers, the decomposition solved the majority to optimality, with a trivial gap. For larger instances, this approach provided lower delivery times than a traditional parcel delivery system using a truck only. Investigations on the effects of drone speed, drone capacity, depot location, constraint addition, and allowing the truck to return to the depot are studied. We consider instances up to 50 customers.

Keywords traveling salesman; drone resupply; urban logistics; release dates; decomposition; matheuristics.

*Corresponding author. E-mail address: silva@auburn.edu

1 Introduction

According to the World Economic Forum (2020), the demand for last-mile delivery services will grow nearly 80% globally by 2030. Although encouraging for retailers, this projection presents a challenge for companies to deliver products efficiently and on time, as local delivery is the most expensive part of the overall supply chain (Capgemini Research Institute, 2019).

One trend accompanying this growth has been the increased expectation by customers of fast delivery (World Economic Forum, 2020). This includes next-day or 2-day shipping, which is now offered by many retailers, often at no additional cost. These shipments are usually handled in two legs: a long-haul leg and a last-mile leg. Last-mile logistics operators normally deliver goods on the same day that they receive them at their local facilities, creating the challenge of planning delivery routes while simultaneously receiving new requests. Once delivery vehicles have started their routes, these operators commonly employ two strategies to manage the arrival of new orders: dispatch a new vehicle to deliver these orders or instruct an en-route vehicle to return to the depot to collect them. Both options incur additional worker time and vehicle travel.

This paper studies an alternative for this problem: using a drone to send newly available orders to delivery vehicles while they are en route, which allows them to continue their distribution without the need to return to the depot periodically. We investigate the benefits of this alternative considering a single delivery truck. To enable a fair comparison between the *traditional strategy* – where the truck must return to the depot to pick up newly available orders – and the *drone-resupply strategy*, we consider that the order information (i.e., delivery locations and order release times) is known. This is valid for those cases when last-mile delivery operators know with good accuracy when their suppliers will deliver the orders to their facilities (Archetti & Bertazzi, 2020; Mor & Speranza, 2020). A second important application area is the planning of deliveries of products ordered in advance but prepared the same day that they are shipped. In this case, the planner also knows what time each order will be ready to ship, but these times are distributed throughout the day. Examples of this situation include compounding pharmacies and bakeries, where the freshness of the product and timely delivery are both critical concerns.

The objective function that we use is the minimization of the completion time of the delivery process, which is the time when the driver returns to the depot after delivering all orders. This metric is chosen to be consistent with standard employment practices, which involve fixed hourly rates and fixed work schedules. For example, in the United States, UPS drivers are members of the Teamsters union and require the company to give them the same 8-hour schedule for five consecutive days each week (International Brotherhood of Teamsters, 2018). This means that drivers receive the same payment regardless of whether they are driving, waiting, or working at the depot. This choice is also consistent with the literature, as the minimization of completion time is the most common objective function used in routing problems with release dates (see, e.g., Archetti et al. 2018; Li et al., 2019), as well as amongst papers addressing drone delivery (Chung et al., 2020).

Considering a single resupply drone, we formally present the problem of finding the optimal delivery route that the truck should follow, as well as the timing and locations where the drone will resupply the truck with new orders. We term this problem *The Traveling Salesman Problem with Release Dates and Drone Resupply* (TSPRD-DR). The term *release dates* was coined by Cattaruzza et al. (2016) to refer to the moment when the orders of each customer become available for dispatch at the depot. We consider planning orders to be delivered in a single day, hence our release dates refer to specific times within the day, but we use the release dates terminology to be consistent with the literature.

Note that if we assume that all orders are ready for dispatch at the beginning of the planning horizon and there is no time limit to deliver them, then, due to the triangle inequality (which we assume holds for the data), the optimal solution is the Traveling Salesman Problem (TSP) optimal tour of the truck, without needing to use the drone. Therefore, the TSPRD-DR is NP-Hard since it has the TSP as a special case. This becomes apparent as we attempt to use our MILP model to solve larger instances of the TSPRD-DR (20 customers or more), as it becomes impossible to solve the MILP model in a reasonable time using state-of-the-art commercial solvers. Hence, we also propose a decomposition approach, where we find the truck route first and assign the drone-resupply strategy second. Computational experiments show that for larger size instances this approach works better than simply running the MILP solver for a few hours.

The main contributions of this paper are:

- The use of drones to send newly available orders to delivery vehicles is introduced in a setting where orders' release dates are known in advance. This is a realistic scenario for many businesses, such as last-mile logistics operators, and is also a more practical application of drones than when the drone is used as the end delivery mechanism at the customer location. A MILP model to minimize the total delivery time is proposed for this new TSPRD-DR.

- An extension of the TSPRD-DR formulation that combines both traditional and the drone-resupply strategies is also developed. Here, newly released orders can either be collected by the truck at the depot or resupplied via a drone.

- The benefit of the drone-resupply strategy is compared with the traditional strategy of truck returns to the depot. Numerical experiments show that using drones to resupply trucks can reduce the total delivery times by up to 20%.

- A decomposition approach that can quickly obtain high-quality solutions is developed. For instances of 10 and 15 customers, the decomposition approach solved half of the instances to optimality. Additionally, it provided lower delivery times than the traditional strategy of having the truck return to the depot to pick up the orders, when considering larger instances of 20 to 30 customers.

- The utilization of the drone – the number of flights, the use of its capacity, and the number of orders resupplied – is also studied over instances of up to 50 customers. We provide an analysis of wait times of the truck as well.

The remainder of this paper is organized as follows. Section 2 provides a brief overview of the related literature. In Section 3 we describe the TSPRD-DR in detail and present the MILP formulation for the problem. Section 4 presents our decomposition approach to route first and resupply second. Section 5 collects the results and analysis of extensive numerical experiments. Finally, Section 6 concludes the paper. Some additional computational results are given in the appendices.

2 Literature Review

In this section, we begin by describing the literature associated with routing problems with release dates. We then focus on explaining the main papers related to last-mile logistics with drone delivery. Finally, we present the only two works, so far, that have addressed the use of drones for resupply, highlighting the differences and contributions of our work.

2.1 Routing Problems with Release Dates

In classical routing problems, such as the TSP or the Vehicle Routing Problem (VRP), one of the main assumptions is that all packages are available for dispatch at the beginning of the planning horizon. Cattaruzza et al. (2016) first introduced the concept of *release dates* (i.e., the moment when orders become available for dispatch at the depot) to address this assumption. In this work, the authors extended the VRP with multiple use of vehicles to consider different order release dates. Without proposing a mathematical formulation, Cattaruzza et al. (2016) described a genetic algorithm with an embedded local search procedure. They also presented a process to create a set of instances used for their computational experiments.

For a single vehicle that may perform multiple routes, Archetti et al. (2015) introduced the *TSP with Release Dates* (TSPRD) to consider release dates. These authors described the complexity of the TSPRD when the graph is either a star (i.e., the depot is at the center of a distribution area) or a line (i.e., the depot and customers are located on the same road). Archetti et al. (2018) were the first to propose a MILP formulation for the TSPRD. Their objective was to minimize the completion time, which is the sum of travel times and waiting times of the vehicle at the depot before starting new routes. They were able to find optimal solutions when up to 20 customers were considered. Additionally, that paper presented two variants of a heuristic approach based on iterated local search for solving large instances, and compared the results of each heuristic with the optimal solutions reached by the mathematical formulation.

Other authors have also considered a deadline to deliver the orders. In his doctoral dissertation, Shelbourne (2016) extended the capacitated VRP to take into account release and due dates for each order. He proposed MILP formulations for determining a set of routes that minimize a convex combination of the total travel costs and the total weighted tardiness, and found optimal solutions for instances with 20 customers. Later, Shelbourne et al. (2017) presented a path-relinking algorithm for the problem studied by Shelbourne (2016) with a penalty function based on the exceeded capacity of vehicles. Finally, Reyes et al. (2018) extended the work of Archetti et al. (2015) to study the complexity of routing problems with release dates and deadlines, where customers are located along the same road.

With same-day delivery (SDD) services, routing problems with release dates arise. Klapp et al. (2016) studied an SDD dispatch problem considering a single truck, where customers and the depot are located on the same road. The authors defined the problem as a Markov Decision Process (MDP) and their computational experiments were focused on studying the trade-off between a truck waiting and dispatching and between longer and shorter vehicle routes. To extend their previous work to a normal network graph, Klapp et al. (2018) proposed a MILP formulation for the deterministic case (i.e., release dates are known at the beginning of the horizon) to determine a set of mutually disjoint tours that a single truck needs to perform to minimize the costs of transportation and customers not served. In this model, the dispatch decisions are only made during discrete moments of the day, which they term *waves*. They also defined a stochastic version of the problem as an MDP and discussed results reached by three heuristics for randomly generated instances.

2.2 Routing Problems with Drone Delivery

Since companies such as Amazon and Google announced their intention to incorporate drones in the parcel delivery process, the interest in delivery logistics using drones has grown considerably (Otto et al., 2018; Chung et al., 2020). Murray and Chu (2015) were the first to propose an optimization model for a truck-and-drone delivery system, in the problem that they termed the *Flying Sidekick Traveling Salesman Problem* (FSTSP). The FSTSP is an extension of the TSP, where a single truck is equipped with a drone that can be launched en route to deliver some parcels. Due to its limited flight endurance, the drone must return to the truck before its energy runs out.

Wang et al. (2017) presented an extension of the FSTSP that considers the use of multiple vehicles equipped with multiple drones. They performed a worst-case analysis of cost reduction compared with the traditional VRP. In Poikonen et al. (2017), the same authors presented an extension of the VRPD that includes physical restrictions such as drone flight endurance and the calculation of distances for truck routes. Schermer et al. (2018) proposed two heuristics to solve the VRPD, evaluating performance with modified TSP instances.

Recently, Kitjacharoenchai et al. (2019) studied an extension of the VRP where drones are allowed to make deliveries directly from the depot and also depart and return to different trucks. They proposed an insertion heuristic that follows the same two-phase idea exploited by Murray and Chu (2015), generating truck tours first and then assigning customers to drones. There is also the extension introduced by Schermer et al. (2019a), called the *VRPD and en route operations*, where drones can be launched and retrieved at vertices and some points of the arcs of the network. Poikonen and Golden (2020) presented the *k-multi-visit drone routing problem* considering a tandem between a truck and k-drones, where there is a drone energy drain function that takes into account the package weight that is transported. Computational experiments showed that the total completion time of the service is highly sensitive to drone speed. Finally, Schermer et al. (2019b) proposed a MILP and a matheuristic approach for the VRPD, where the allocation of customers and routes to each vehicle are decisions made separately. Experiments showed that their solution approach can obtain high-quality solutions in reasonable computational time.

2.3 Routing Problems with Drone Resupply

To conclude the literature review, we highlight the use of drones for resupply has been studied only twice. The first to propose using drones to resupply the delivery vehicle were

Dayarian et al. (2020). They focused on the case where previously unknown orders arrive dynamically throughout the day and the operator must decide how and when to dispatch them to customers. Since no late deliveries are accepted, their objective is to maximize the total number of orders delivered, using a single truck and a single resupply drone, where the former can perform only one trip during the day. They presented two resupply strategies, one where the truck receives new packages from the drone only at the end of the truck delivery tour (called *restricted resupply*), and the second where the truck meets the drone at some location to be resupplied (called *flexible resupply*). The heuristics developed for each strategy (no exact method was given in the paper) solve optimization models several times a day, as new information becomes available. Considering a drone load capacity of 50 orders, they showed that using drones for resupplying can significantly increase the number of customers served compared to a truck-only delivery system. Their numerical experiments also suggest that the flexible-resupply strategy provides better results than the restricted-resupply strategy.

Drone resupply also appears in a recent master's thesis from McCunney and Van Cauwenberghe (2019), who conducted simulation tests to study the impact of using drones to transport new orders to transshipment points to be picked up by dispatch vehicles. Compared with a traditional parcel delivery system, they showed using a single drone, with a capacity of five orders, and one transshipment point reduces both the delivery time and the total distance traveled by the trucks.

Our research is different from the two works mentioned above in that we assume that the orders and their destinations are known at the beginning of the day, but are not yet (necessarily) ready to be shipped. From an application perspective, the model we propose can be used to plan same-day orders in a context where the orders are known at the beginning of the day, but are not yet ready. These can include make-to-order operations, where orders become ready as they are completed, or last-mile-logistics operations, where orders become ready for the final segment of delivery when they arrive from the previous segment. Unlike Dayarian et al. (2020) and McCunney and Van Cauwenberghe (2019), we present a mathematical formulation of our problem and solve it to optimality (for smaller instances). Additionally, we extend this model to allow the truck to perform several delivery routes during the day, which implies that newly released orders can either be shipped by drone or collected directly by the truck at the depot. We also evaluate the performance of our decomposition approach relative to the optimal solution.

3 Problem Description and Mathematical Formulation

In this section, we describe a MILP formulation for the TSPRD-DR. This formulation finds a minimum time route for a single truck, which can be resupplied en route with new orders by means of a drone. We assume that every customer can order at most once during the day, and that the release dates of orders are known at the beginning of the planning horizon. Orders can then be loaded onto the truck either at the depot or via the drone while en route. In addition, we assume that the drone can rendezvous with the truck only at customer locations, and we consider an unloading and drone launching time when this meeting takes place. The truck does not return to the depot during the day and the drone must return to the depot before it can be sent out to resupply the truck again. We assume that the drone has a given capacity and flight endurance, whereas the truck has no such limits.

3.1 An Illustrative Example

To show the complexity of the decisions, in Figure 1 we present an instance of the problem where the depot and four customers are located on the same road. The rectangle represents the depot and the circles are customers. The release dates in minutes, denoted by w_i for $i = 1, ..., 4$, are written above each node (e.g. the order of customer 3 is ready to dispatch at the depot at 35 minutes). The numbers below each customer represent the distance from the depot.

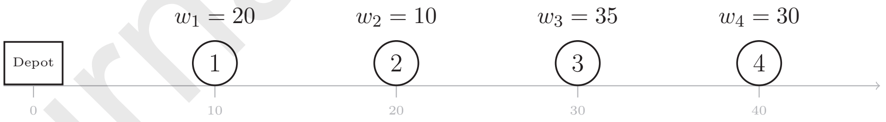


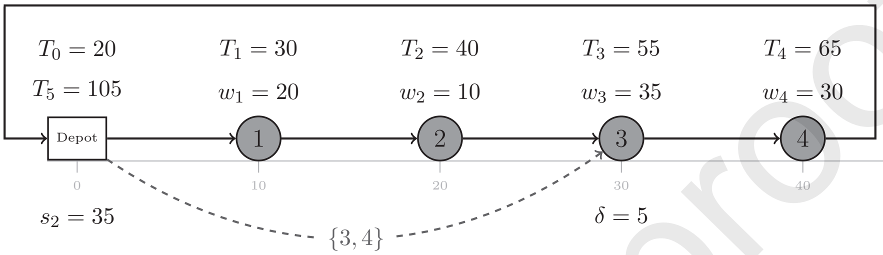
Figure 1: Example of an instance of the problem.

Figure 2 depicts a feasible solution for this example. The continuous lines represent the truck trip, whereas the dashed line represents the drone tour. We consider a truck speed of 1 distance unit per minute and the drone speed is set to twice the truck speed. When Order 1 becomes ready for dispatch at $w_1 = 20$, the truck leaves the depot at $T_0 = 20$ with this order and Order 2 loaded onto it (note that Order 2 has a release time of $w_2 = 10$). During its journey, Orders 3 and 4 become available at the depot at times $w_3 = 35$ and $w_4 = 30$, respectively. The drone departs from the depot to customer 3 at minute $s_3 = 35$, carrying Orders 3 and 4 (as can be seen on the dashed line label). The drone arrives at the same time (minute 50) as the truck at the location of Customer 3. After receiving the drone and unloading the new orders and launching the drone back to the depot, which takes $\delta = 5$ minutes, the truck moves to Customer 4 and then returns to the depot at $T_5 = 105$.

Note that if the drone is not used, the truck must wait until the last order becomes ready to dispatch at the depot to start the route (i.e., at minute 35). In this case, the truck returns to the depot at minute 115. Alternatively, the truck could perform multiple trips, but that would take even more time.

drone toi 3 cung thoi gian voi truck (50), truck xuat phat tai thoi diem T3

Truck xuat phat ti thoi diem T0 = time hang diem w1 san sang, (mang duoc w1, w2)



drone xuat phat tu depot tai thoi diem w4 ready, mang w3,w4 cho truck

Figure 2: Feasible solution for the example.

3.2 Mathematical Formulation

Let $N = \{1, 2, ..., n\}$ be the set of customers and $D \subset N$ be the subset that can be visited by the drone according to its endurance. We represent the depot with the subindices 0 and $n+1$ and we define $N_0 = N \cup \{0\}$, $N_{n+1} = N \cup \{n+1\}$, $D_0 = D \cup \{0\}$, $D_{n+1} = D \cup \{n+1\}$, and $N_{0,n+1} = N \cup \{0, n+1\}$ as sets of nodes that include the depot. The problem is defined on the directed graph $G = (N_{0,n+1}, E)$, where $N_{0,n+1}$ represents the set of nodes and E is the set of edges. The parameters and decision variables of the model are listed below.

3.2.1 Parameters

- w_i: Release date of the order of customer i.
- t_{ij}: Truck travel time associated with the edge (i, j).
- d_j: Drone flying time between the depot and the node j.
- δ: Time for receiving and unloading orders from the drone and launching the drone back to the depot.
- a_i: Drone load capacity units required to transport the order of customer i.

- A: Drone maximum load capacity.
- M: A sufficiently large constant.

3.2.2 Decision Variables

- x_{ij}: 1, if node j is visited immediately after node i by the truck. 0, otherwise.
- r_{ij}: 1, if node j is visited (not necessarily immediately) after node i by the drone. 0, otherwise.
- u_i: 1, if the drone flies to node i to resupply the truck with new orders. 0, otherwise.
- y_{ij}: 1, if the order of customer j is loaded onto the truck at node i. 0, otherwise.
- T_i: Time when the truck departs node i.
- s_i: Time when the drone is launched for node i.
- ϵ_i: Elapsed time between arrival and departure of the truck at node i.

3.2.3 MILP Formulation

The MILP formulation is described in (1)–(30). We interpret $y_{ij} = 1$ for some $i \in N$ as the decision to dispatch the drone with the order of customer j to node i, whereas y_{0j} represents if the order of customer j is loaded onto the truck at the depot. Additionally, we interpret $u_0 = 1$ as the decision to use the drone to resupply the truck with new orders at least once, and $r_{0j} = 1$ represents j being the first customer visited by the drone.

$$\text{minimize } T_{n+1} \tag{1}$$

Subject to:

- Truck routing constraints:

truck xuat phat 1 lan

$$\sum_{j \in N} x_{0j} = 1 \tag{2}$$

truck ve depot 1 lan

$$\sum_{i \in N} x_{i,n+1} = 1 \tag{3}$$

$$\sum_{i \in N_0 \setminus \{j\}} x_{ij} = 1 \qquad \forall j \in N \tag{4}$$

den tham moi customer 1 lan

$$\sum_{j \in N_{n+1} \setminus \{i\}} x_{ij} = 1 \qquad \forall i \in N \tag{5}$$

xuat phat tu cac node 1 lan

- Drone routing constraints:

$$u_i \leq u_0 \qquad \forall i \in D \tag{6}$$

$$\sum_{j \in D} r_{0j} = u_0 \tag{7}$$

$$\sum_{i \in D} r_{i,n+1} = u_0 \tag{8}$$

constrain dam bao drone bay theo lo trinh da dinh truoc (lo trinh dua theo cac bien)

$$\sum_{i \in D_0 \setminus \{j\}} r_{ij} = u_j \qquad \forall j \in D \tag{9}$$

$$\sum_{j \in D_{n+1} \setminus \{i\}} r_{ij} = u_i \qquad \forall i \in D \tag{10}$$

- Drone capacity constraint:

$$\sum_{j \in N} y_{ij} a_j \leq A \cdot u_i \qquad \forall i \in D \tag{11}$$

- Constraint on loading the orders onto the truck:

$$\sum_{i \in D_0} y_{ij} = 1 \qquad \forall j \in N \tag{12}$$

- Synchronization and timing constraints:

$$T_j \geq w_j + \min\{d_j, t_{0j}\} \qquad \forall j \in N \tag{13}$$

$$T_j \geq T_i - M(1 - y_{ij}) \qquad \forall i \in D_0, j \in N \setminus \{i\} \tag{14}$$

$$T_j \geq T_i + t_{ij} - M(1 - x_{ij}) \qquad \forall i \in N_0, j \in N_{n+1} \setminus (D \cup \{i\}) \tag{15}$$

$$T_j \geq T_i + t_{ij} + \delta \cdot u_j - M(1 - x_{ij}) \qquad \forall i \in N_0, j \in D \setminus \{i\} \tag{16}$$

$$T_j \geq s_j + d_j + \delta - M(1 - u_j) \qquad \forall j \in D \tag{17}$$

$$s_j \geq T_i + d_i - M(1 - r_{ij}) \qquad \forall i \in D, j \in D \setminus \{i\} \tag{18}$$

- Constraints on the start time of the truck route and drone flights:

$$s_j \geq w_i \cdot y_{ji} \qquad \forall i \in N, j \in D \tag{19}$$

$$T_0 \geq w_j \cdot y_{0j} \qquad \forall j \in N \tag{20}$$

- Computation of wait times and lower bounds for the total delivery time:

$$\begin{aligned}
\epsilon_j &\geq T_j - (T_i + t_{ij}) - M(1 - x_{ij}) && \forall j \in D, i \in N_0 \backslash \{j\} && (21)\\
\epsilon_j &\leq M \cdot u_j && \forall j \in D && (22)\\
T_{n+1} &\geq T_i + t_{i,n+1} && \forall i \in N && (23)\\
T_{n+1} &\geq T_0 + \sum_{(i,j)\in E} t_{ij}x_{ij} + \sum_{i\in D} \epsilon_i && && (24)
\end{aligned}$$

- Non-negativity and integrality constraints:

$$\begin{aligned}
x_{ij} &\in \{0,1\} && \forall (i,j) \in E && (25)\\
r_{ij} &\in \{0,1\} && \forall i \in D_0, j \in D_{n+1} \backslash \{i\} && (26)\\
u_i &\in \{0,1\} && \forall i \in D_0 && (27)\\
y_{ij} &\in \{0,1\} && \forall i \in D_0, j \in N && (28)\\
T_i &\geq 0 && \forall i \in N_{0,n+1} && (29)\\
s_i, \epsilon_i &\geq 0 && \forall i \in D && (30)
\end{aligned}$$

The objective function (1) minimizes the total delivery time to serve all customers. Constraints (2)–(5) are the classic flow balance constraints used in VRPs. Constraints (6)–(8) ensure that the drone leaves and returns to the depot if it is used to resupply the truck with new orders at least once, while (9) and (10) force the drone to fly to the location that is assigned to it. The load capacity of the drone is restricted in (11). Constraint (12) ensures that the order of each customer is loaded onto the truck at exactly one node, either at the depot or by means of the drone at a customer's location. Constraint (13) establish a lower bound to the departure time of the truck at each node. Constraint (14) forces an order to be loaded before or when its associated customer is visited (i.e., if the order of customer j is loaded onto the truck at node i, then the departure time at i must be less than the departure time at j). Inequalities (15) and (16) are for subtour elimination, used with (17) to update the departure time of the truck at each node. That is, for a resupply at node j, the truck departure time from that location will be given by the time at which the last vehicle arrives at j plus the time for the driver to receive the drone, unload its orders, and send it back to the depot (denoted by δ). Constraints (18) and (19) ensure that the drone can start a new flight to node j as long as it is available at the depot (after returning from a previous resupply mission at node i) and all orders that it will carry to resupply the truck at node j are ready for dispatch. In a similar way, constraint (20) ensures that the truck can only start the delivery trip once all orders to be loaded onto it are available for dispatch at the depot. In constraint (21), the elapsed time between arrival and departure of the truck at each node is computed. Constraint (22) forces these variables to zero in case there is no drone resupply at the corresponding location. Constraints (23) and (24) are introduced to strengthen the formulation: they ensure that the truck returns to the depot only after complete the distribution. (The effectiveness of these constraints is discussed at the beginning of Section 5.2.) Finally, non-negativity and integrality constraints are included in (25)–(30).

An extension of this formulation which allows the truck to return more than once to the depot is described in Appendix A. In this model, newly released orders can be either picked up by the truck at the depot or resupplied by the drone. The results of that MILP model are discussed at the end of Section 5.2.

4 Decomposition Approach

Because of the impracticality of using an exact approach for larger sized problems, in this section, we present a solution approach based on decomposing the problem into the two main decisions involved: the *truck-routing* and the *drone-resupply* decisions, which are now made in consecutive stages. The former corresponds to the decision of determining the sequence in which the truck will visit the customers, whereas the latter corresponds to the decision of defining the locations where each order will be loaded onto the truck.

4.1 Routing Decisions

To find a good truck route at the first stage of the decomposition, we consider the hypothetical situation where there exist as many drones as customers. Under this assumption, the problem is transformed into a TSP with Time Windows (TSP-TW) where the time when each customer's time window starts is given by its order's release date (time) plus the drone flying time from the depot. To solve this problem, we used the following MILP formulation, which is a relaxation the MILP model (1)–(30).

$$\text{minimize} \quad T_{n+1} \tag{31}$$

Subject to (2)–(5), (13), (23), (25), (29),

$$T_j \geq T_i + t_{ij} - M(1 - x_{ij}) \qquad \forall (i,j) \in E \tag{32}$$

$$\epsilon_j \geq T_j - (T_i + t_{ij}) - M(1 - x_{ij}) \qquad \forall j \in N \tag{33}$$

$$T_{n+1} \geq T_0 + \sum_{(i,j)\in E} t_{ij}x_{ij} + \sum_{i\in N} \epsilon_i \tag{34}$$

$$\epsilon_i \geq 0 \qquad \forall i \in N \tag{35}$$

Note that the only difference between constraints (32)–(35) and constraints (15), (16), (21), (24) and (30) in the original model is that this model also assumes that drones can reach any node, so the variable ϵ_i is now defined for all $i \in N$. Thus, every solution of the TSP-TW provides a lower bound (which may be feasible or not) for the MILP model (1)–(30). Nevertheless, as we will explain in the next section, the second stage of the decomposition can start with this solution and modify it to be feasible for the original problem.

4.2 Drone-Resupply Decisions

Let us now introduce a MILP formulation to find the best locations along the truck route to load each order onto the truck. We represent a truck route by the sequence $\pi = (\pi_0, \pi_1, \ldots, \pi_{|N|})$, where $\pi_i \in N$ represents a customer, for $i \in \{1, ..., |N|\}$, and π_0 corresponds to the depot. We assume that after visiting customer $\pi_{|N|}$, the truck returns to the depot.

To avoid confusion between customers versus their positions in π, let $I = \{1, ..., |N|\}$ be the set of indices (locations) of sequence π. Additionally, let $\mathcal{D} \subset I$ be the set of indices (locations) that could be visited by the drone according to its endurance. Considering these sets, the new definition of the parameters and decision variables of the model are listed below.

4.2.1 Parameters

- w_{π_i}: Release date of the order of customer visited at position i.
- t_{π_{i-1},π_i}: Truck travel time between customers visited at positions $i-1$ and i.
- d_{π_i}: Drone flying time between the depot and the node visited at position i.

4.2.2 Decision Variables

- u_{π_i}: 1, if the drone flies to the node visited at position $i \in \mathcal{D}$ to resupply the truck with new orders. 0, otherwise.
- T_{π_i}: Time when the truck departs the node visited at position $i \in I \cup \{0\}$.

- y_{π_i,π_j}: 1, if the order of the customer visited at position $j \in I$ is loaded onto the truck in position $i \in \mathcal{D} \cup \{0\}$. 0, otherwise.

4.2.3 MILP Formulation

To minimize the total delivery time, the MILP model to determine the best locations to load each order onto the truck is presented as follows.

$$\text{minimize } T_{\pi_{|I|}} + t_{\pi_{|I|},n+1} \tag{36}$$

Subject to:

$$\begin{aligned}
\sum_{i \in \mathcal{D}\cup\{0\}, i \le j} y_{\pi_i,\pi_j} &= 1 && \forall j \in I && (37)\\
\sum_{j=i}^{|I|} y_{\pi_i,\pi_j} \cdot a_{\pi_j} &\le A \cdot u_{\pi_i} && \forall i \in \mathcal{D} && (38)\\
T_{\pi_0} &\ge w_{\pi_j} \cdot y_{\pi_0,\pi_j} && \forall j \in I && (39)\\
T_{\pi_i} &\ge T_{\pi_{i-1}} + t_{\pi_{i-1},\pi_i} && \forall i \in I \backslash \mathcal{D} && (40)\\
T_{\pi_i} &\ge T_{\pi_{i-1}} + t_{\pi_{i-1},\pi_i} + \delta \cdot u_{\pi_i} && \forall i \in \mathcal{D} && (41)\\
T_{\pi_i} &\ge (w_{\pi_j} + d_{\pi_i} + \delta) \cdot y_{\pi_i,\pi_j} && \forall i \in \mathcal{D}, i \le j \le |I| && (42)\\
T_{\pi_i} &\ge T_{\pi_k} + d_{\pi_k} + d_{\pi_i} + \delta - M(2 - u_{\pi_i} - u_{\pi_k}) && \forall i \in \mathcal{D}\backslash\{1\}, k \in \mathcal{D}, k < i && (43)\\
u_{\pi_i} &\in \{0,1\} && \forall i \in \mathcal{D} && (44)\\
y_{\pi_i,\pi_j} &\in \{0,1\} && \forall i \in \mathcal{D}\cup\{0\}, j \in I, i \le j \le |I| && (45)\\
T_{\pi_i} &\ge 0 && \forall i \in I \cup \{0\} && (46)
\end{aligned}$$

The objective function (36) minimizes the time when the truck returns to the depot (denoted by $n + 1$) after delivering all orders. Constraint (37) states that every order must be loaded onto the truck at exactly one location. The load capacity of the drone is restricted in (38). Constraint (39) ensures that the truck can only start its tour once all orders loaded onto it at the depot are available for dispatch. Constraints (40)–(43) update the time when the truck departs each location $i \in I$. Finally, non-negativity and integrality constraints are included in (44)–(46).

Note that the variables y_{π_i,π_j} are only defined for $i \in \mathcal{D} \cup \{0\}, i \le j \le |I|$, which means that each customer order must be loaded before or when this customer is visited. The reduction of the feasible solution space given by this definition is a clear advantage (computationally) compared to the MILP model presented in Section 3, where the same variables are defined without knowing in advance the route that the truck will perform.

5 Computational Experiments

In this section, we evaluate the performance of the MILP model and the decomposition approach. We also present sensitivity analysis to understand the behavior of the drone-resupply delivery system for different problem settings. The model and the decomposition were encoded in Java on a windows-based workstation Intel(R) Xeon(R) Gold 5118 CPU @2.30 GHz (12 cores) with 64 GB RAM, using CPLEX 12.8 for solving the MILP models.

We begin this section by describing our parameter choices. We then present the computational performance of the MILP model and the decomposition approach on the benchmark instances created by Archetti et al. (2018). Using these instances, we show the benefits of drone-resupply compared to the traditional method studied by those authors (where the truck must return to the depot to collect newly available orders). For this comparison, we consider the distribution of the customers, drone capacity, and the spread of the orders' release dates over time. Additionally, we assess the impact of permitting both strategies (drones and return to the depot) for resupply. Finally, we study the impact of the drone speed and depot location on the total delivery times, using randomly generated instances from 10 to 50 customers.

5.1 Parameter Definition

We employ different distance metrics and travel speeds for each vehicle. The truck distances are computed using the ℓ_1-norm (Manhattan distance) with a constant truck speed of 30 km/hr. The drone distances use the Euclidean norm, and we consider two constant drone speeds of 45 and 60 km/hr.

To establish the flight endurance and load capacity of the drone, we considered the Avidrone 210TL drone manufactured by Avidrone Aerospace (Avidrone Aerospace, 2019). This drone can carry up to 25 kilograms and has an endurance of 90 minutes, which makes it appropriate for our application. Therefore, we set the flight endurance of the drone to 90 minutes. Alternatively, an equal assumption is an endurance of 45 minutes and the driver swaps the drone's battery after receiving the drone and unloading the orders. We consider drone load capacities (A) of 2, 4, and 8 orders. These capacities are conservative estimates considering that the majority of orders delivered by Amazon weigh less than 2.3 kilograms (Poikonen and Golden, 2020). We assume that all orders are interchangeable according to size and weight ($a_i = 1, \forall i \in N$). The time to retrieve the drone, unload orders from the drone, swap the battery if needed, and launch the drone back to the depot is $\delta = 5$ minutes.

All MILP models described earlier require that M be a number large enough not to lead to infeasibility, but small enough not to lead to poor linear relaxations. Since a solution of the TSPRD-DR consists of waiting for all orders to become available for dispatch and then performing the TSP optimal tour of the truck, we established $M = \max_{i \in N}\{w_i\} + z_{TSP}$, where z_{TSP} corresponds to the TSP optimal value. This means that M is an upper bound for the optimal solution. However, when using this value, preliminary computational experiments showed that CPLEX was not always able to quickly find an initial feasible solution to start the branching processes of MILP models (1)–(30) and (31)–(35). Therefore, we initialized CPLEX with the solution where the truck waits until all orders are ready and then performs a TSP optimal tour.

5.2 MILP Results

To both study the computational performance of the MILP model (1)–(30) and compare its results with the MILP model of Archetti et al. (2018), we use the instances generated by those authors. The instances are based on Solomon's instances C101, C201, R101, and RC101 (Solomon, 1987). The C instances have clustered customers, the R instances have randomly distributed customers, and the RC instances have randomly distributed clusters of customers. For a given number of customers (n), the authors took the first $n+1$ nodes from each Solomon instance, assuming that the first node corresponds to the depot. We also used the first listed node as the depot. After computing the TSP optimal value (z_{TSP}) for the instance selected (without release dates), they generated a release date for each order uniformly at random in the interval $[0, \beta \cdot z_{TSP}]$, where β is a factor that defines the spread of the release dates over time. The authors created an instance for each combination of $\beta \in \{0.5, 1.0, ..., 2.5, 3.0\}$ and Solomon's instance used, for a total of 24 instances for each value of n.

We analyzed the performance of the MILP model considering the following metrics:

- Optimality GAP and the CPU time (in seconds) given by CPLEX.
- Objective value (total delivery time) in minutes.
- Number of drone trips (n_{drone}) performed.
- Total number of orders resupplied (o_r) by the drone.

We begin by describing the results obtained by CPLEX for the instances of 10 and 15 customers, considering drone speed of 60 km/hr, load capacity of $A = 4$, and maximum run-time of 3 hours. (The effectiveness of constraints (21)–(24) on the run-times and gaps reached by the MILP model (1)–(30) is presented in Table 9 of Appendix C. As can be observed, removing these constraints seriously degrades computational performance.) Table 1 shows that for instances of 10 customers CPLEX can find optimal solutions in less than a minute. However, for 15 customers, CPU times increase significantly as the spread of release dates increases. With $\beta = 3.0$, the average GAP reached by CPLEX in 3 hours is 5.0% across 15-customer instances.

The total number of orders resupplied by the drone (o_r) – presented in Table 1 – shows that the vast majority are transported by the drone. Even for problem instances with a relatively low spread (β) of release dates, the drone is useful, though often with fewer trips (on C101 and RC101 instances). The average number of orders resupplied by the drone is 8.3 and 13 for instances of 10 and 15 customers, respectively, demonstrating the utility of drone resupply. From a managerial perspective, this type of data analysis can help understand how often the drone will be used and the typical proportion of packages that the drone will handle.

Table 1: Results of the MILP model (1)–(30) by CPLEX within a maximum run-time of 3 hours for instances of $n = 10$ and $n = 15$ customers with a drone load capacity of 4 orders.

Instance	β	$n = 10$					$n = 15$				
		CPU time	Ob. value	GAP	n_{drone}	o_r	CPU time	Ob. value	GAP	n_{drone}	o_r
C101	0.5	1.5	139.0	0.0%	1	4	64.4	272.0	0.0%	2	8
	1.0	8.9	149.0	0.0%	1	4	220.9	284.7	0.0%	4	14
	1.5	19.8	158.1	0.0%	2	7	452.1	301.1	0.0%	4	13
	2.0	16.5	167.6	0.0%	3	9	9181.5	339.6	0.0%	4	14
	2.5	10.7	187.1	0.0%	3	9	7515.5	372.6	0.0%	4	13
	3.0	31.9	210.1	0.0%	3	9	10800.2	410.1	2.2%	3	8
C201	0.5	0.9	359.0	0.0%	3	9	28.7	446.7	0.0%	4	14
	1.0	0.7	376.5	0.0%	3	9	30.0	464.9	0.0%	4	13
	1.5	1.0	397.1	0.0%	3	9	31.2	493.9	0.0%	4	13
	2.0	5.0	422.3	0.0%	3	9	182.2	533.6	0.0%	5	14
	2.5	3.8	465.9	0.0%	5	9	2312.4	596.3	0.0%	4	13
	3.0	6.0	505.3	0.0%	5	9	10800.1	658.9	7.1%	5	14
R101	0.5	0.2	431.0	0.0%	3	9	1.9	603.0	0.0%	4	14
	1.0	0.1	450.0	0.0%	3	9	8.8	629.4	0.0%	5	14
	1.5	0.3	467.0	0.0%	3	9	11.5	656.0	0.0%	4	14
	2.0	0.5	534.0	0.0%	3	9	54.0	700.2	0.0%	5	14
	2.5	2.1	602.2	0.0%	4	9	373.5	771.4	0.0%	5	13
	3.0	6.2	664.2	0.0%	3	9	991.5	827.4	0.0%	7	13
RC101	0.5	6.9	359.5	0.0%	3	9	5458.4	413.6	0.0%	3	12
	1.0	2.8	371.1	0.0%	3	9	2617.2	420.0	0.0%	4	14
	1.5	4.3	397.3	0.0%	3	9	2207.8	455.3	0.0%	3	12
	2.0	13.7	447.3	0.0%	3	7	10800.2	504.1	4.2%	4	14
	2.5	30.7	497.3	0.0%	3	9	10800.2	558.3	10.1%	4	14
	3.0	13.5	546.3	0.0%	3	6	10800.2	613.3	10.8%	4	14
Avg.		**7.8**	**387.7**	**0.0%**	**3.0**	**8.3**	**3572.7**	**513.6**	**1.4%**	**4.1**	**13.0**

An illustration of an optimal TSPRD-DR solution of instance R101 with 10 customers and $\beta = 1.5$ is shown in Figure 3. Figure 3a depicts the first drone resupply operation. The continuous lines represent the truck route, and their labels show the truck travel times associated with each edge. The dashed line represents the drone flight, and its label shows that Orders 2, 3, 4, and 10 are transported on the drone to the truck at Node 2. The truck departs Node 1 at $T_1 = 40$ and arrives at Node 2 at minute 116. The drone is launched from the depot at $s_2 = 98$ to Node 2, arriving at the same time as the truck. The light gray nodes represent orders available at the time of drone launch from the depot; the dark gray nodes are orders that have already been delivered, and the white nodes are orders that are not yet ready for dispatch. Since $\delta = 5$ minutes, both vehicles depart Node 2 at $T_2 = 121$. Figure 3b shows an optimal TSPRD-DR solution with an objective value of $T_{n+1} = 467$ for this instance.

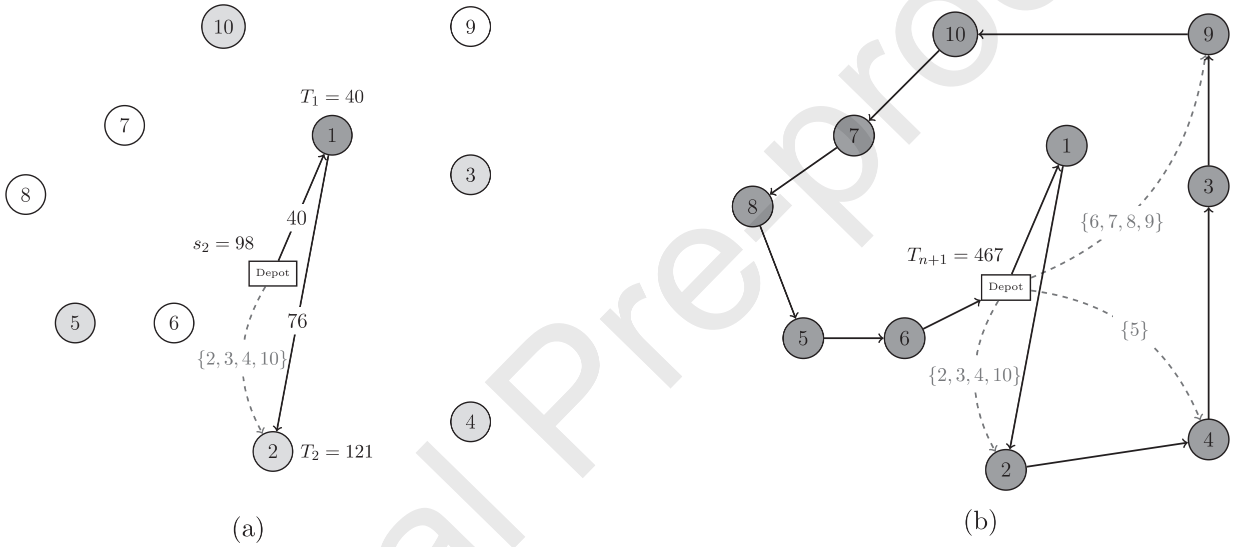


Figure 3: Instance R101 of 10 customers with $\beta = 1.5$. (a) First drone resupply operation. (b) Optimal TSPRD-DR solution.

5.2.1 Benefits of Using Drones to Resupply Vehicles

To show the benefits of the drone-resupply strategy, we compare our results with the traditional delivery system proposed by Archetti et al. (2018), where the truck returns to the depot to pick up newly released orders (the results obtained by CPLEX using the MILP model of these authors are in Table 7 and 8 in Appendix B). Considering drone load capacities (A) of 2 and 4 with drone speed of 60 km/hr, Figure 4 shows that the drone resupply approach always results in lower total delivery times than the truck only system for instances of 10 customers (the results of the MILP model (1)–(30) with $A = 2$ are in Table 10 in Appendix C). For both drone capacities, the minimum time savings occurs for instance C101 with $\beta = 0.5$ of 3 and 8 minutes for $A = 2$ and $A = 4$, respectively. The maximum time savings is for RC101 with $\beta = 2.5$, where our approach reduces the total delivery times by 115 and 120 minutes for capacities of $A = 2$ and $A = 4$, respectively. The improvement is hardly surprising since the drone can travel while the truck is also traveling, and the drone moves faster. This combination of activities in parallel and the greater speed of the drone enable drone resupply to be a better option for all instances studied.

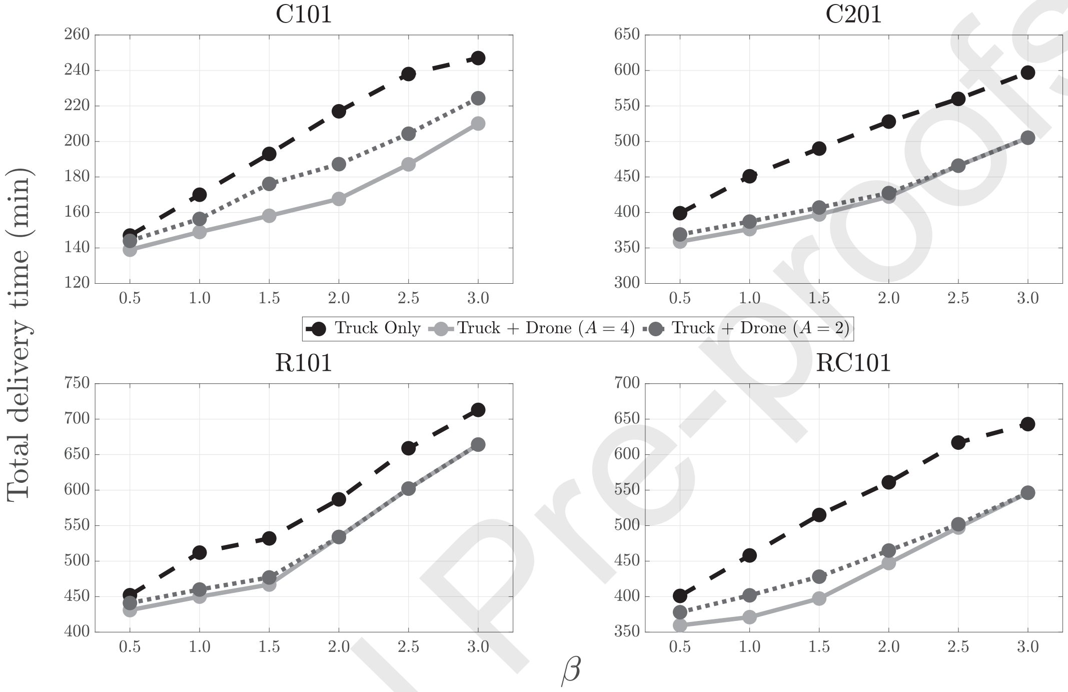


Figure 4: Total delivery times of the truck only delivery system proposed by Archetti et al. (2018) and the truck-and-drone delivery system (this paper), using drone speed of 60 km/hr and load capacities of $A = 2$ and $A = 4$ orders, for instances of 10 customers.

To further consider the effect of drone load capacity, Figure 5 shows the distribution of the percentage of time savings for capacities of 2, 4, and 8 orders of the drone resupply system relative to a truck only strategy (the results of the MILP model (1)–(30) with $A = 2$ and $A = 8$ are in Table 11 in Appendix C). The results show that increasing the drone capacity from 2 to 4 significantly increases the median percent time saved for the C101 and RC101 instances. However, doubling the drone load capacity from 4 to 8 only slightly reduces the total delivery times (as can be seen in Table 11 and the time reduction only occurs when $\beta \leq 1.5$). The maximum percent time saved is 20.86% and occurs for instance RC101 with $\beta = 2.0$. For 15 customers, it seems that a drone capacity of 4 is sufficient for most instances.

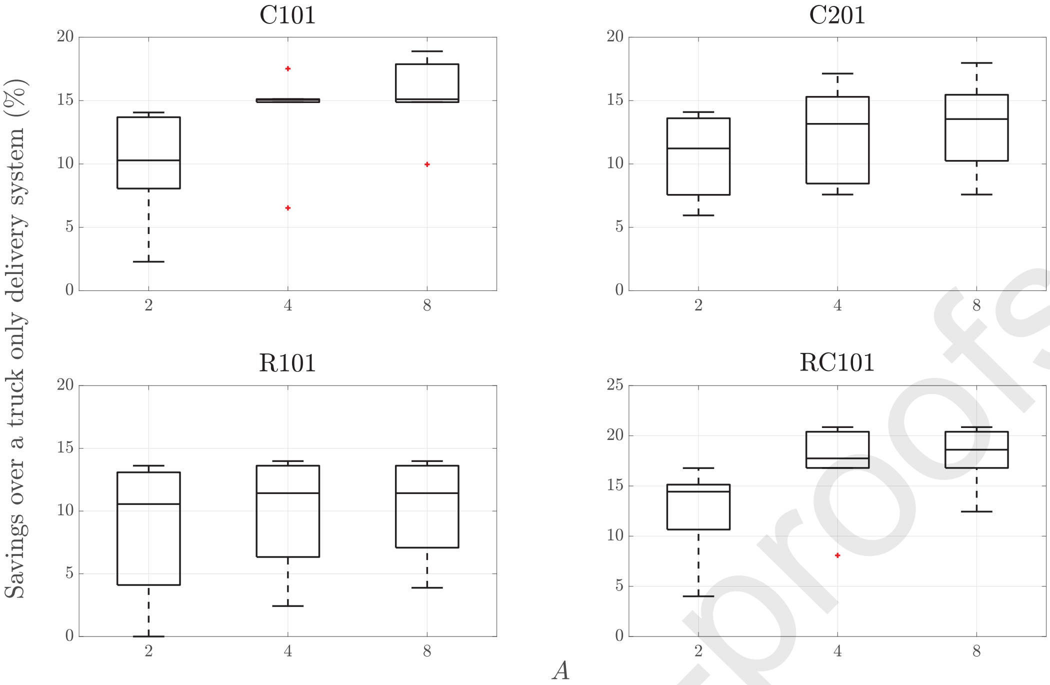


Figure 5: Percent time saved of the drone delivery system with a drone speed of 60 km/hr compared to a truck only delivery system (Archetti et al. (2018) MILP model) for instances of 15 customers with drone capacity on the x axis.

5.2.2 Allowing Truck Returns to the Depot with Drone-Resupply

Now, we consider the impact of allowing the truck to return to the depot during the day, which we term *The Multi-trip TSPRD-DR* (MT-TSPRD-DR). This version considers both traditional (depot returns) and drone-resupply strategies, so new orders can either be resupplied by the drone or picked up by truck at the depot. The mathematical formulation of this problem is an extension of the MILP model (1)–(30) and is detailed in Appendix A. The detailed numerical results of this model can be found in Appendix A.3, here we provide a summary.

Considering 10 customers, the MT-TSPRD-DR improves the delivery times only for instance R101 with $\beta = 1.5$. The reduction is 10 and 5 minutes for drone capacities of $A = 2$ and $A = 4$, respectively (the difference from the solution shown in Figure 3 is that the truck serves Customer 1 on the first trip, then returns to the depot to collect Orders 2, 3, 4 and 5, and then starts the second and final trip). When considering 15 customers and a maximum run-time of 3 hours, the MT-TSPRD-DR reduces total delivery times for only one instance (C201 with $\beta = 3.0$) with a drone capacity of 4. When considering a capacity of 2, although the multi-trip model could improve objective values for 5 instances (the maximum reduction was 11 minutes in instance R101 with $\beta = 0.5$), for two instances (RC101 with $\beta = 1.0$ and $\beta = 3.0$) it had worse results in 3 hours than the original TSPRD-DR MILP model (1)–(30). Although the multi-trip MILP model should theoretically achieve the same or better results than the single-trip model (1)–(30) (since every feasible solution to the TSPRD-DR is a feasible solution to the MT-TSPRD-DR), for instances of more than 10 customers, we cannot solve either model to optimality in computationally reasonable times and therefore we cannot make definitive comparisons. Future research could include strengthening the mathematical formulations of both models using valid inequalities so that realistically sized instances can be solved and compared.

5.3 Decomposition Results

In this section, we describe the results from the decomposition approach for the instances created by Archetti et al. (2018) and for large-size randomly generated instances. Recall that the decomposition basically relaxes the problem in the first stage (eliminates the single drone constraint by assuming there are as many drones as customers) to obtain a route, then the second stage finds the best locations to resupply the truck from a single drone resulting in a feasible solution to the original problem. In the first stage, to obtain the TSP-TW solution using CPLEX, we established a maximum run-time of 5 minutes. (Preliminary experiments showed that increasing the run-time does not necessarily lead to better results in the second stage.) We then use this solution to execute the second stage, which specifies the best locations to load each order onto the truck using the drone.

5.3.1 Decomposition Results for Instances of Archetti et al. (2018)

Considering a drone speed of 60 km/hr and a load capacity of 4 orders, the average results of the decomposition for the instances created by Archetti et al. (2018) are shown in Table 2. After running the first stage, the average CPU times of the second stage show that CPLEX can find optimal solutions for the second-stage problem in less than one second. The objective value increases in this stage since the TSP-TW solutions are not necessarily feasible for the original problem (see Section 4.1). The average number of orders resupplied (o_r) exhibits the same behavior described in the previous subsection: the drone resupplies the majority of orders. Computational time is very modest even for 20 customers and all GAP values reach 0% for the second-stage MILP. Note that the gaps listed in Table 2 refer to the MILP at each stage of the decomposition approach.

Table 2: Average results of the decomposition approach over the six values of β, considering a drone capacity of 4 orders.

n	Instance	**First stage: TSP-TW tour**			**Second stage: Drone resupply**				
		CPU time	Ob. value	GAP	CPU time	Ob. value	GAP	n_{drone}	o_r
10	C101	0.05	154.1	0.0%	0.05	175.6	0.0%	2.3	7.3
	C201	0.17	407.7	0.0%	0.04	424.9	0.0%	3.7	9.0
	R101	0.14	513.9	0.0%	0.04	524.7	0.0%	2.8	8.0
	RC101	0.88	422.7	0.0%	0.05	439.5	0.0%	2.7	7.8
15	C101	87.28	310.7	0.2%	0.09	336.9	0.0%	3.3	11.5
	C201	59.80	514.2	1.3%	0.07	534.8	0.0%	4.5	13.7
	R101	10.92	680.6	0.0%	0.07	704.6	0.0%	4.8	13.8
	RC101	160.64	477.1	4.9%	0.08	503.4	0.0%	3.0	11.3
20	C101	203.19	372.1	2.3%	0.14	409.3	0.0%	4.2	14.8
	C201	106.92	539.5	3.7%	0.12	561.4	0.0%	5.0	18.2
	R101	115.52	789.1	4.3%	0.12	809.1	0.0%	5.8	17.7
	RC101	300.67	680.4	22.0%	0.14	758.0	0.0%	4.0	13.3

To have a sense of the quality of the solutions found by the decomposition approach, we consider the following metrics:

- GAP of the decomposition: computed considering the best bound of the MILP model (1)–(30) reached by CPLEX within a maximum run-time of 3 hours.
- The number of optimal solutions (s_{opt}) reached by the MILP model (1)–(30) and the decomposition.
- The number of solutions (s_{bet}) where the decomposition reached a better objective value than the MILP model (1)–(30).

Table 3 shows the average results found by the decomposition for six β values. Considering $n = 10$ customers, the decomposition finds 13 of 24 optimal solutions, with an average GAP of 1.3%. For instances of 15 customers, the average GAP by the decomposition is 2.6%, with a total of 12 optimal solutions. When considering 20 customers, the decomposition gives 7 of the 8 optimal solutions obtained by the MILP model (1)–(30) (the results obtained by this MILP model can be found in Table 12 Appendix C). Additionally, in four instances the decomposition reaches a better value than the MILP under a 3-hour time limit. These results are more striking when considering that the maximum run-time established for the MILP is 3 hours, while for the decomposition it is 5 minutes. Finally, as before, the solutions from decomposition always have lower average total delivery times than the truck only delivery system proposed by Archetti et al. (2018). This is also true for the larger 25 and 30 customers instances, as can be seen in Figure 7 in Appendix D.

Table 3: Average results of the decomposition over six values of β, considering a drone load capacity of 4 orders.

n	Instance	**Truck only MILP**	**Truck + Drone MILP**			**Decomposition**			
		Ob. value	Ob. value	GAP	s_{opt}	Ob. value	GAP	s_{opt}	s_{bet}
10	C101	202.0	168.5	0.0%	6	175.6	3.6%	2	0
	C201	504.2	421.0	0.0%	6	424.9	0.9%	2	0
	R101	575.8	524.7	0.0%	6	524.7	0.0%	6	0
	RC101	532.5	436.5	0.0%	6	439.5	0.8%	3	0
	Avg.	**453.6**	**387.7**	**0.0%**		**391.2**	**1.3%**		
15	C101	385.3	330.0	0.4%	5	336.9	2.0%	3	0
	C201	607.8	532.4	1.2%	5	534.8	1.6%	3	0
	R101	779.3	697.9	0.0%	6	704.6	0.9%	4	0
	RC101	597.3	494.1	4.2%	3	503.4	6.0%	2	0
	Avg.	**592.5**	**513.6**	**1.4%**		**519.9**	**2.6%**		
20	C101	443.8	402.2	9.2%	1	409.3	10.8%	0	0
	C201	632.8	560.3	4.5%	4	561.4	4.6%	4	0
	R101	896.2	814.7	7.1%	3	809.1	6.5%	3	1
	RC101	803.8	731.4	27.1%	0	758.0	29.2%	0	3
	Avg.	**694.2**	**627.2**	**12.0%**		**634.5**	**12.8%**		

5.3.2 Decomposition Results on Randomly Generated Instances

So far, we have evaluated the computational performance of the MILP model and the decomposition approach using the benchmark instances created by Archetti et al. (2018). The results have shown that using drones for resupplying improves delivery times for different time-space distribution of customer orders and different drone load capacities. We have also validated that decomposing the problem into the truck-routing and the drone-resupply decisions can effectively solve small and medium-sized instances. However, as can be seen in Table 3, the total delivery times are sometimes quite long (more than 12 hours). This is partly explained by to the release dates (that is, times) of some orders (e.g., in the instance R101 of 20 customers with $\beta = 3$, there are orders released after minute 600). But more importantly, the distribution area considered is unrealistically large: from 32×32 km^2 for the C101 instances, to 59×59 km^2 for the RC101 instances.

Therefore, to continue studying the effectiveness of the decomposition approach and analyze the impact of other parameters (drone speed and depot location), we created instances from 10 to 50 customers (10 instances per each number of customers) distributed uniformly at random over an area of $16 \times 16 = 256$ km^2. This area is a similar size to Baltimore, MD (239 km^2) or Arlington, TX (258 km^2). Additionally, we assumed that the orders' release dates follow a uniform distribution between 8:00 AM and 4:00 PM. A similar distribution area and time interval were used in Dayarian et al. (2020) in their paper on drone resupply, as well as in Ulmer et al. (2019) in their paper on preemptive depot return strategies for SDD.

We consider two locations for the depot: at the center (8,8) and outside (20,8) of the distribution area. We test drone speeds of 60 and 45 km/hr (2 and 1.5 times the truck speed, respectively). A summary of the results is presented in Table 4, considering a drone load capacity of 4 orders. Again, in the first stage, we set a maximum run-time of 5 minutes. (In preliminary experiments, we also tested with 10 minutes and corroborated that this does not necessarily produce better results after running the second stage.) Once the truck route is obtained, we run the second stage considering the two drone speeds. The solution times of this stage show, again, that it finds optimal solutions in less than a minute.

Table 4: Average results from by the decomposition over randomly generated instances from 10 to 50 customers (n), considering two depot locations (layout), two drone speeds, and $A = 4$.

n	Layout	Drone speed	First stage: TSP-TW			Second stage: Drone resupply				
			CPU time	Ob. Value	GAP	CPU time	Ob. Value	n_{drone}	o_r	Wait time
10	Outside	60	0.57	499.0	0.0%	0.04	508.9	4.2	6.1	3.6
		45	0.57	499.0	0.0%	0.04	514.5	3.6	5.8	3.7
	Center	60	1.02	475.7	0.0%	0.04	487.1	4.2	5.5	3.0
		45	1.02	475.7	0.0%	0.04	489.0	4.0	5.6	2.9
20	Outside	60	145.20	519.7	1.2%	0.12	540.9	6.1	14.1	4.1
		45	145.20	519.7	1.2%	0.11	547.2	5.8	13.2	6.4
	Center	60	94.22	498.9	1.2%	0.11	521.1	5.7	12.7	2.7
		45	94.22	498.9	1.2%	0.11	523.6	5.3	11.7	4.4
30	Outside	60	271.50	524.0	3.0%	0.43	564.1	7.5	23.4	4.1
		45	271.50	524.0	3.0%	0.39	570.8	7.3	23.1	3.9
	Center	60	272.86	505.5	4.5%	0.43	548.3	7.8	22.9	7.2
		45	272.86	505.5	4.5%	0.44	551.3	7.6	22.0	6.0
40	Outside	60	276.65	542.5	4.8%	0.84	612.7	7.8	28.4	5.8
		45	276.65	542.5	4.8%	0.86	618.2	8.1	28.9	6.9
	Center	60	301.23	537.3	9.6%	0.97	586.3	8.2	28.5	5.1
		45	301.23	537.3	9.6%	0.85	588.4	7.9	27.2	4.4
50	Outside	60	303.75	555.0	6.6%	3.27	617.4	9.5	35.0	4.7
		45	303.75	555.0	6.6%	3.28	624.7	8.9	33.7	4.8
	Center	60	300.48	535.8	8.5%	23.86	624.5	10.1	37.3	7.0
		45	300.48	535.8	8.5%	27.15	627.7	10.0	36.9	6.6

As expected, the total delivery times increase as the drone speed decreases. Nevertheless, the increases are marginal and remain relatively stable as the number of customers increases: between 5.6 to 7.3 minutes on average when the depot is located outside the area and between 1.9 to 3.1 minutes when it is at the center. This shows the operational efficiency of carrying out activities in parallel, that is, the drone flies to resupply the truck while the latter is making deliveries. In instances of up to 40 customers, the total delivery times increase by 16 to 30 minutes by moving the depot from the center to outside of the area. This is also expected, since the truck's travel time to get from the center (8,8) of the area to the outside location (20,8) is 24 minutes. However, this behavior is not observed for 50 customers, which may be explained by the fact that we are using a heuristic approach to solve these instances, and, likely, order density is so high at this point that the location of the depot becomes less relevant. Finally, to better understand the proposed drone-resupply delivery system, the last column of Table 4 exhibits a new metric: the average time (in minutes) the truck waits per drone flight. Considering all depot locations and drone speeds, this average time increases from 3.3 to 5.8 minutes for instances from 10 to 50 customers, respectively. Additionally, it is observed that there is no relationship between the average wait time per drone flight and drone speed. These times might be expected to increase as the depot is farther from the customers but this does not occur (e.g., in instances of 30 and 50 customers considering a drone speed of 60 km/hr).

Finally, considering a drone speed of 60 km/hr, Figure 6 shows the utilization of the drone capacity, as well as the percentages of drone flights and orders resupplied with respect to the number of customers. As can be seen, these metrics do not vary significantly between the two different depot locations. Figure 6a shows that the idle capacity of the drone decreases as the number of customers increases: while the drone flies with 1.5 parcels on average for 10 customers, it resupplies 3.7 parcels per flight when considering 50 customers. In Figure 6b it can be observed that the percentage of flights performed by the drone (relative to the number of customers) decreases as the size of the instance grows. The percentage of orders resupplied by the drone (Figure 6c) reaffirms that the vast majority of orders are resupplied by the drone (which coincides with Table 1).

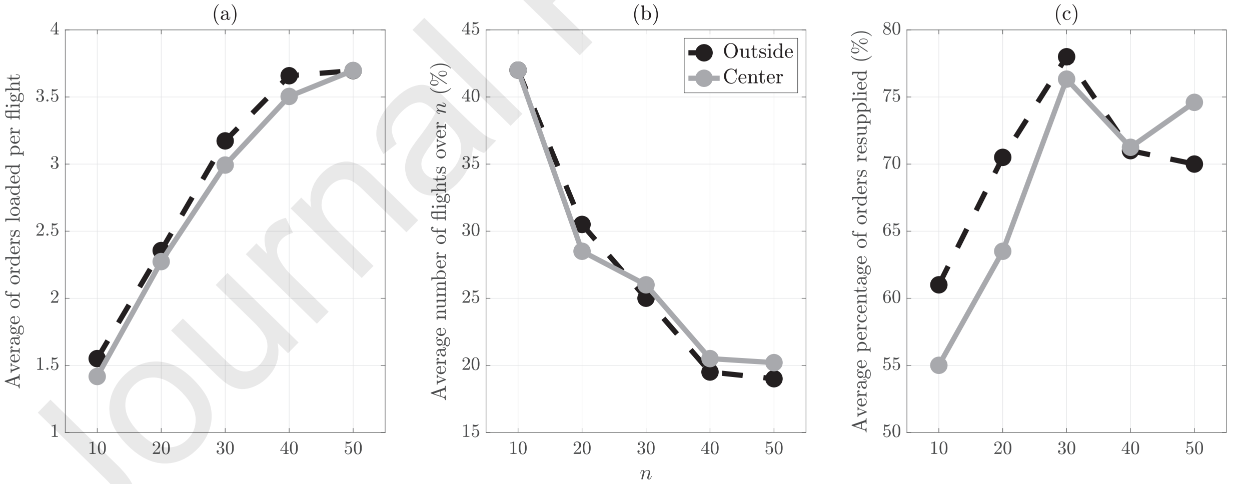


Figure 6: Drone utilization for each size of the instances, considering a load capacity of 4 orders. (a) Average of orders loaded per flight. (b) Average number of drone flights over n. (c) Average percentage of orders resupplied.

6 Conclusions

In this paper, we introduced the Traveling Salesman Problem with Release Dates and Drone Resupply (TSPRD-DR). This problem consists of determining a delivery route for a truck that can receive new orders en route by sending a drone from the depot for resupply. This is a common last-leg distribution situation, where orders become available throughout the day and the time when the orders will be ready is known. What is so appealing about this scenario is that it leverages the cost and environmental savings that drones can bring but at the same time eliminates the serious issues of drones interacting directly with customers. We postulate that this version of truck-and-drone logistics is more pragmatic and achievable, at least in the short term.

We developed and solved a MILP formulation to minimize the time to deliver all orders. The performance of this model was evaluated using the instances generated by Archetti et al. (2018), which considered different patterns of customer distribution. Computational experiments showed that the MILP model can only obtain optimal solutions for instances up to 15 customers. Compared to the traditional parcel delivery system proposed by Archetti et al. (2018), experiments showed that using drones for resupplying can reduce the total delivery time by up to 20% since the truck and the drone work in parallel and the drone is quicker than the truck. We did some auxiliary work considering allowing the truck to return to the depot and also to assess the effectiveness of our constraints on the computational performance of the MILP model. These are detailed in the appendices. Future research could involve developing a decomposition heuristic for the version of the problem that allows the truck to return to the depot so that larger instances could be assessed. Because of the change of structure for this problem version, it would require a significant effort rather than a trivial modification.

To solve problems of larger size, we developed an effective decomposition approach where the truck-routing and the drone-resupply decisions are made in consecutive stages. Over the instances of 10 customers, experiments showed that this approach obtained 13 of 24 optimal solutions in less than one second on average. For instances of 15 customers, 12 of 19 optimal solutions were found in less than 2 minutes on average while 7 of 8 optimal solutions were found for the 20 customer problems. Experiments on instances up to 30 customers showed that the decomposition provides lower delivery times than the truck only delivery system proposed by Archetti et al. (2018). We also considered the effect of drone capacity and the impact of different ranges of time of order availability (parameter β). Finally, the effectiveness of the decomposition and the impact of the drone speed and depot location were studied in randomly generated instances from 10 to 50 customers.

Future work includes considering a fleet of vehicles and drones, where the latter can be associated with either single or multiple vehicles. The proposed delivery system can also be extended considering multiple depots. Given its resolution speed, the MILP model used for the second stage of the decomposition approach could be nested in a local search strategy. Furthermore, the MILP models of the TSPRD-DR and MT-TSPRD-DR could be strengthened to investigate under what conditions allowing the truck to return to the depot to pick up new orders can lead to better results. Finally, a stochastic version of the problem, where orders arrive randomly throughout the day, at the same time as other deliveries are taking place, will be a challenge to address.

7 Acknowledgement

We gratefully acknowledge the financial support provided by Toyota Material Handling North America through their University Research Program.

A The Multi-trip TSPRD-DR

In this appendix, we present an extension of the MILP model (1)–(30) to allow the truck to perform several delivery routes during the day, that is, the truck can return to the depot during the day. Similarly to the approach used for modeling multiple drone flights, we employ variables that indicate whether the truck travels directly from one customer to another or if it visits the depot in between. This means that the formulation does not need a trip index, which is difficult to bound in practice (see Archetti et al., 2018).

Let $\hat{t}_{ij} = t_{i,n+1} + t_{0j}$ be the truck travel time associated with the edges $(i, n+1)$ and $(0, j)$. The *new* decision variables are listed below.

A.1 New Decision Variables

- $\hat{x}_{ij}$: 1, if the truck ends a route at node i and initiates a new route visiting node j. 0, otherwise.
- γ_{ij}: 1, if the order of customer i is loaded onto the truck at the depot before initiates a new route visiting node j. 0, otherwise.

A.2 MILP Formulation for the Multi-Trip TSPRD-DR

The MILP formulation is presented below.

$$\text{minimize } T_{n+1} \tag{47}$$

Subject to (2), (3), (6)–(11), (13)–(23), (25)–(30), and (48)–(59). The last group of constraints is described as follows.

- Modification of constraints (4) and (5):

$$\sum_{j\in N\setminus\{i\}} (x_{ij} + \hat{x}_{ij}) + x_{i,n+1} = 1 \qquad \forall i \in N \tag{48}$$

$$\sum_{i\in N\setminus\{j\}} (x_{ij} + \hat{x}_{ij}) + x_{0j} = 1 \qquad \forall j \in N \tag{49}$$

- Modification of constraint (12):

$$\sum_{i \in D_0} y_{ij} + \sum_{k \in N} \gamma_{jk} = 1 \qquad \forall j \in N \tag{50}$$

- New constraint related to (14):

$$T_j \geq T_i - M(1 - \gamma_{ji}) \qquad \forall i \in N, j \in N \backslash \{i\} \tag{51}$$

- New constraints related to (15) and (16):

$$T_j \geq T_i + \hat{t}_{ij} - M(1 - \hat{x}_{ij}) \qquad \forall i \in N, j \in N \backslash (D \cup \{i\}) \tag{52}$$

$$T_j \geq T_i + \hat{t}_{ij} + \delta \cdot u_j - M(1 - \hat{x}_{ij}) \qquad \forall i \in N, j \in D \backslash \{i\} \tag{53}$$

- New constraints related to (20):

$$T_j \geq (w_i + t_{0j}) \cdot \gamma_{ij} \qquad \forall i \in N, j \in N \backslash D \tag{54}$$

$$T_j \geq (w_i + t_{0j}) \cdot \gamma_{ij} + \delta \cdot u_j \qquad \forall i \in N, j \in D \tag{55}$$

- Modification of constraint (24):

$$T_{n+1} \geq T_0 + \sum_{(i,j) \in E} t_{ij} x_{ij} + \sum_{i \in N} \sum_{j \in N \backslash \{i\}} \hat{t}_{ij} \hat{x}_{ij} + \sum_{i \in D} \epsilon_i \tag{56}$$

- New constraint to link up the new variables:

$$\gamma_{ij} \leq \sum_{k \in N \backslash \{i,j\}} \hat{x}_{kj} \qquad \forall i, j \in N \tag{57}$$

- New integrality constraints:

$$\hat{x}_{ij} \in \{0, 1\} \qquad \forall i \in N, j \in N \backslash \{i\} \tag{58}$$

$$\gamma_{ij} \in \{0, 1\} \qquad \forall i, j \in N \tag{59}$$

A.3 MT-TSPRD-DR MILP Results

The following tables present the results of the above MILP formulation considering a maximum run-time of 3 hours. The column n_{truck} shows the number of routes performed by the truck, while Δ is the percentage of time savings relative to the single-trip MILP model (1)–(30).

Table 5: Optimal results obtained by CPLEX using the MT-TSPRD-DR MILP formulation, using the instances of Archetti et al. (2018) of 10 customers.

Instance	β	$A = 2$						$A = 4$					
		CPU time	Ob. Value	n_{drone}	o_r	n_{truck}	Δ	CPU time	Ob. Value	n_{drone}	o_r	n_{truck}	Δ
C101	0.5	5.3	144.0	2	4	1	0.0%	1.9	139.0	1	4	1	0.0%
	1.0	15.0	156.4	2	4	1	0.0%	9.8	149.0	1	4	1	0.0%
	1.5	759.5	176.1	2	4	1	0.0%	23.9	158.1	2	7	1	0.0%
	2.0	302.0	187.2	2	4	1	0.0%	21.1	167.6	3	9	1	0.0%
	2.5	462.8	204.4	3	6	1	0.0%	24.1	187.1	2	8	1	0.0%
	3.0	850.4	224.4	3	6	1	0.0%	43.4	210.1	1	4	1	0.0%
C201	0.5	1.5	369.0	5	9	1	0.0%	1.4	359.0	3	9	1	0.0%
	1.0	2.9	387.1	5	9	1	0.0%	1.4	376.5	3	9	1	0.0%
	1.5	3.8	407.0	4	8	1	0.0%	2.0	397.1	3	9	1	0.0%
	2.0	4.6	427.3	5	9	1	0.0%	4.6	422.3	3	9	1	0.0%
	2.5	9.9	465.9	5	9	1	0.0%	7.7	465.9	4	9	1	0.0%
	3.0	14.5	505.3	5	9	1	0.0%	13.8	505.3	4	9	1	0.0%
R101	0.5	0.4	441.0	5	9	1	0.0%	0.3	431.0	3	9	1	0.0%
	1.0	0.5	460.0	5	9	1	0.0%	0.3	450.0	3	9	1	0.0%
	1.5	0.7	467.0	3	5	2	**2.1%**	0.5	462.0	2	5	2	**1.1%**
	2.0	1.2	534.0	5	8	1	0.0%	1.3	534.0	3	7	1	0.0%
	2.5	8.3	602.2	5	7	2	0.0%	4.7	602.2	3	8	1	0.0%
	3.0	14.6	664.2	4	6	4	0.0%	14.3	664.2	4	7	2	0.0%
RC101	0.5	14.4	378.0	3	6	1	0.0%	11.2	359.5	3	9	1	0.0%
	1.0	8.3	401.8	3	5	1	0.0%	2.8	371.1	3	9	1	0.0%
	1.5	50.5	428.1	3	6	1	0.0%	5.8	397.3	3	9	1	0.0%
	2.0	45.0	464.9	3	5	1	0.0%	20.7	447.3	3	9	1	0.0%
	2.5	37.3	501.9	3	5	1	0.0%	28.0	497.3	2	5	1	0.0%
	3.0	78.4	546.3	3	4	1	0.0%	26.6	546.3	4	9	1	0.0%

Table 6: Results obtained by CPLEX using the MT-TSPRD-DR MILP formulation, using the instances of Archetti et al. (2018) of 15 customers. A negative value of Δ means that the MT-TSPRD-DR MILP model reached a worse solution than the MILP model (1)–(30) in 3 hours.

Instance	β	$A = 2$							$A = 4$						
		CPU time	Ob. Value	GAP	n_{drone}	o_r	n_{truck}	Δ	CPU time	Ob. Value	GAP	n_{drone}	o_r	n_{truck}	Δ
C101	0.5	7.0	284.3	0.0%	4	8	1	0.0%	92.3	272.0	0.0%	2	8	1	0.0%
	1.0	10801.1	308.0	2.0%	4	8	1	0.0%	371.3	284.7	0.0%	4	14	1	0.0%
	1.5	10809.4	333.0	13.2%	4	8	1	0.0%	651.5	301.1	0.0%	4	14	1	0.0%
	2.0	10808.5	351.9	11.0%	5	10	1	0.0%	10801.6	339.6	6.4%	4	12	1	0.0%
	2.5	10808.0	378.9	5.8%	5	10	1	0.0%	10806.2	372.6	4.2%	3	10	1	0.0%
	3.0	10807.0	415.1	3.4%	5	10	1	0.0%	10813.8	410.1	2.2%	4	13	1	0.0%
C201	0.5	79.8	459.0	0.0%	4	8	1	0.0%	22.6	446.7	0.0%	4	14	1	0.0%
	1.0	138.6	486.0	0.0%	5	10	1	0.0%	28.2	464.9	0.0%	4	13	1	0.0%
	1.5	199.6	512.0	0.0%	5	10	1	0.0%	85.3	493.9	0.0%	4	13	1	0.0%
	2.0	1517.5	544.3	0.0%	6	12	1	0.0%	1116.9	533.6	0.0%	5	14	1	0.0%
	2.5	5794.3	596.3	0.0%	7	14	1	0.0%	7122.3	596.3	0.0%	4	14	1	0.0%
	3.0	10805.6	657.9	12.7%	5	8	2	**0.2%**	10806.8	653.9	12.2%	5	11	2	**0.8%**
R101	0.5	7.8	607.0	0.0%	1	2	2	**1.8%**	3.6	603.0	0.0%	4	14	1	0.0%
	1.0	16.3	637.0	0.0%	5	8	2	**1.2%**	18.7	629.4	0.0%	5	14	1	0.0%
	1.5	101.5	665.0	0.0%	5	10	2	**0.4%**	22.8	656.0	0.0%	5	14	1	0.0%
	2.0	140.9	703.4	0.0%	6	12	2	**0.7%**	103.0	700.2	0.0%	4	13	1	0.0%
	2.5	10802.2	771.4	4.3%	4	8	2	0.0%	1646.4	771.4	0.0%	5	9	2	0.0%
	3.0	10801.0	827.4	10.1%	7	13	1	0.0%	10800.4	827.4	9.5%	5	9	2	0.0%
RC101	0.5	10805.1	432.0	4.0%	4	8	1	0.0%	4136.7	413.6	0.0%	3	12	1	0.0%
	1.0	10805.8	479.0	16.6%	3	6	1	**-3.9%**	3019.4	420.0	0.0%	4	14	1	0.0%
	1.5	10804.3	491.9	16.9%	4	8	1	0.0%	10800.9	455.3	1.4%	3	12	1	0.0%
	2.0	10800.8	530.1	16.2%	4	7	1	0.0%	10800.6	504.1	18.4%	4	15	1	0.0%
	2.5	10800.8	571.3	22.4%	4	7	1	0.0%	10800.6	558.3	12.3%	4	13	1	0.0%
	3.0	10809.5	628.0	18.1%	5	8	2	**-0.3%**	10800.7	613.3	17.3%	4	13	1	0.0%

B Results of the MILP of Archetti et al. (2018) for the Truck Only Model

Table 7: Optimal results obtained by CPLEX using the Archetti et al. (2018) MILP formulation for instances of 10, 15, and 20 customers.

Instance	β	$n = 10$		$n = 15$		$n = 20$	
		CPU time (s)	Ob. Value (min)	CPU time (s)	Ob. Value (min)	CPU time (s)	Ob. Value (min)
C101	0.5	0.2	147	0.5	291	2.3	337
	1.0	0.5	170	0.8	335	10.1	382
	1.5	0.4	193	10.2	365	129.6	416
	2.0	0.3	217	11.5	399	164.3	457
	2.5	1.0	238	14.1	439	507.0	511
	3.0	0.7	247	31.3	483	584.0	560
C201	0.5	0.4	399	1.6	488	4.3	509
	1.0	0.3	451	6.0	544	47.1	567
	1.5	1.0	490	5.4	596	69.8	612
	2.0	1.4	528	14.5	630	148.6	659
	2.5	1.3	560	30.6	676	175.0	699
	3.0	1.2	597	44.7	713	401.6	751
R101	0.5	0.3	452	0.9	618	2.1	724
	1.0	0.5	512	2.6	672	24.1	764
	1.5	0.5	532	4.6	727	72.2	848
	2.0	0.8	587	31.7	814	211.0	936
	2.5	1.2	659	42.8	893	231.1	1020
	3.0	1.1	713	42.1	952	382.3	1085
RC101	0.5	0.2	401	0.9	450	2.7	634
	1.0	0.3	458	1.2	516	7.7	719
	1.5	0.3	515	1.4	572	28.5	776
	2.0	0.4	561	9.4	637	89.1	839
	2.5	0.7	617	27.7	671	192.3	905
	3.0	0.8	643	35.8	738	264.4	950
Avg.		**0.7**	**453.6**	**15.5**	**592.5**	**156.3**	**694.2**

Table 8: Results obtained by CPLEX using the Archetti et al. (2018) MILP formulation for instances of 25 and 30 customers with a maximum run-time of 3 hours.

Instance	β	$n = 25$			$n = 30$		
		CPU time (s)	GAP	Ob. Value (min)	CPU time (s)	GAP	Ob. Value (min)
C101	0.5	7.4	0.0%	362	9.5	0.0%	386
	1.0	174.1	0.0%	416	357.6	0.0%	432
	1.5	399.1	0.0%	453	1253.0	0.0%	481
	2.0	2238.0	0.0%	516	2620.5	0.0%	536
	2.5	1417.9	0.0%	574	3320.9	0.0%	582
	3.0	10808.1	11.2%	640	10801.7	7.2%	652
C201	0.5	6.3	0.0%	558	16.6	0.0%	618
	1.0	261.6	0.0%	636	662.4	0.0%	696
	1.5	804.1	0.0%	675	658.0	0.0%	735
	2.0	1000.3	0.0%	725	2201.2	0.0%	785
	2.5	2295.8	0.0%	802	2618.2	0.0%	858
	3.0	10816.5	8.8%	894	10810.7	16.9%	978
R101	0.5	11.8	0.0%	845	87.6	0.0%	884
	1.0	261.8	0.0%	940	1756.0	0.0%	980
	1.5	1426.1	0.0%	1042	10386.5	0.0%	1077
	2.0	10825.9	2.8%	1135	10820.8	10.5%	1172
	2.5	5316.7	0.0%	1240	10818.6	21.2%	1280
	3.0	10815.2	9.0%	1361	10801.0	24.7%	1507
RC101	0.5	6.0	0.0%	687	12.6	0.0%	907
	1.0	97.8	0.0%	806	209.8	0.0%	1026
	1.5	414.8	0.0%	921	874.3	0.0%	1141
	2.0	1452.1	0.0%	1040	2398.5	0.0%	1252
	2.5	4229.3	0.0%	1139	10801.7	6.9%	1375
	3.0	10823.0	7.7%	1250	10774.1	0.0%	1454
Avg.		**3162.9**	**1.6%**	**819.0**	**4378.0**	**3.6%**	**908.1**

C Supplementary Results of the MILP model (1)–(30) for the TSPRD-DR

Table 9: Results of the MILP model over the instances of Archetti et al. (2018) of 10 customers, according to the constraints considered.

Instance	β	MILP (1)–(30)		MILP without (21), (22), and (24)		MILP without (23)		MILP without (21)–(24)	
		CPU time	GAP	CPU time	GAP	CPU time	GAP	CPU time	GAP
C101	0.5	1.5	0.0%	160.8	0.0%	1.4	0.0%	762.9	0.0%
	1.0	8.9	0.0%	272.3	0.0%	9.7	0.0%	1169.1	0.0%
	1.5	19.8	0.0%	107.1	0.0%	23.0	0.0%	1592.0	0.0%
	2.0	16.5	0.0%	18.4	0.0%	43.6	0.0%	3600.3	6.3%
	2.5	10.7	0.0%	14.3	0.0%	107.1	0.0%	3600.1	3.7%
	3.0	31.9	0.0%	28.3	0.0%	214.7	0.0%	1862.3	0.0%
	Avg.	**14.9**	**0.0%**	**100.2**	**0.0%**	**66.6**	**0.0%**	**2097.8**	**1.7%**
C201	0.5	0.9	0.0%	59.4	0.0%	1.4	0.0%	629.4	0.0%
	1.0	0.7	0.0%	50.2	0.0%	1.1	0.0%	810.3	0.0%
	1.5	1.0	0.0%	42.7	0.0%	1.9	0.0%	929.6	0.0%
	2.0	5.0	0.0%	28.5	0.0%	2.4	0.0%	883.9	0.0%
	2.5	3.8	0.0%	12.8	0.0%	4.8	0.0%	1095.3	0.0%
	3.0	6.0	0.0%	13.8	0.0%	9.2	0.0%	2321.8	0.0%
	Avg.	**2.9**	**0.0%**	**34.6**	**0.0%**	**3.5**	**0.0%**	**1111.7**	**0.0%**
R101	0.5	0.2	0.0%	50.2	0.0%	0.2	0.0%	509.6	0.0%
	1.0	0.1	0.0%	41.3	0.0%	0.3	0.0%	374.8	0.0%
	1.5	0.3	0.0%	14.6	0.0%	0.3	0.0%	421.9	0.0%
	2.0	0.5	0.0%	21.5	0.0%	1.0	0.0%	928.1	0.0%
	2.5	2.1	0.0%	21.2	0.0%	3.7	0.0%	522.8	0.0%
	3.0	6.2	0.0%	22.2	0.0%	8.5	0.0%	497.0	0.0%
	Avg.	**1.6**	**0.0%**	**28.5**	**0.0%**	**2.3**	**0.0%**	**542.4**	**0.0%**
RC101	0.5	6.9	0.0%	41.8	0.0%	4.9	0.0%	410.8	0.0%
	1.0	2.8	0.0%	47.2	0.0%	2.1	0.0%	659.3	0.0%
	1.5	4.3	0.0%	18.7	0.0%	5.2	0.0%	471.9	0.0%
	2.0	13.7	0.0%	60.6	0.0%	11.2	0.0%	714.8	0.0%
	2.5	30.7	0.0%	89.8	0.0%	22.3	0.0%	797.4	0.0%
	3.0	13.5	0.0%	54.8	0.0%	41.6	0.0%	3600.2	54.2%
	Avg.	**12.0**	**0.0%**	**52.1**	**0.0%**	**14.6**	**0.0%**	**1109.1**	**9.0%**

Table 10: Optimal results of the MILP model (1)–(30) obtained by CPLEX over instances of 10 customers, considering two load capacities of $A = 2$ and $A = 4$ orders.

Instance	β	$A = 2$				$A = 4$			
		CPU time	Ob. value	n_{drone}	o_r	CPU time	Ob. value	n_{drone}	o_r
C101	0.5	4.5	144.0	2	4	1.5	139.0	1	4
	1.0	23.2	156.4	2	4	8.9	149.0	1	4
	1.5	383.0	176.1	2	4	19.8	158.1	2	7
	2.0	133.6	187.2	2	4	16.5	167.6	3	9
	2.5	315.4	204.4	3	5	10.7	187.1	3	9
	3.0	341.3	224.4	2	4	31.9	210.1	3	9
C201	0.5	1.4	369.0	5	9	0.9	359.0	3	9
	1.0	1.2	387.1	5	9	0.7	376.5	3	9
	1.5	2.0	407.0	4	8	1.0	397.1	3	9
	2.0	2.2	427.3	5	9	5.0	422.3	3	9
	2.5	5.1	465.9	5	9	3.8	465.9	5	9
	3.0	8.9	505.3	5	9	6.0	505.3	5	9
R101	0.5	0.4	441.0	5	9	0.2	431.0	3	9
	1.0	0.3	460.0	5	9	0.1	450.0	3	9
	1.5	0.7	477.0	5	9	0.3	467.0	3	9
	2.0	0.7	534.0	5	9	0.5	534.0	3	9
	2.5	4.3	602.2	5	7	2.1	602.2	4	9
	3.0	7.8	664.2	5	9	6.2	664.2	3	9
RC101	0.5	9.3	378.0	3	6	6.9	359.5	3	9
	1.0	10.4	401.8	3	6	2.8	371.1	3	9
	1.5	34.2	428.1	3	6	4.3	397.3	3	9
	2.0	55.9	464.9	3	5	13.7	447.3	3	7
	2.5	60.6	501.9	3	5	30.7	497.3	3	9
	3.0	17.6	546.3	3	5	13.5	546.3	3	6

Table 11: Results of the MILP model (1)–(30) obtained by CPLEX with a maximum run-time of 3 hours over instances of 15 customers, considering drone load capacity (A) of 2, 4, and 8 orders.

Instance	β	$A = 2$					$A = 4$					$A = 8$				
		CPU time	Ob. value	GAP	n_{drone}	o_r	CPU time	Ob. value	GAP	n_{drone}	o_r	CPU time	Ob. value	GAP	n_{drone}	o_r
C101	0.5	73.3	284.3	0.0%	4	8	64.4	272.0	0.0%	2	8	10.2	262.0	0.0%	2	14
	1.0	1905.9	308.0	0.0%	4	8	220.9	284.7	0.0%	4	14	19.8	275.1	0.0%	2	14
	1.5	10800.2	333.0	6.1%	4	8	452.1	301.1	0.0%	4	13	214.7	296.1	0.0%	3	14
	2.0	10800.2	351.9	5.9%	5	10	9181.5	339.6	0.0%	4	14	5334.7	339.6	0.0%	3	14
	2.5	10800.3	378.9	3.3%	4	8	7515.5	372.6	0.0%	4	13	10800.3	372.6	1.0%	3	12
	3.0	10800.3	415.1	3.4%	5	8	10800.2	410.1	2.2%	3	8	10800.2	410.1	2.2%	2	7
C201	0.5	41.0	459.0	0.0%	4	8	28.7	446.7	0.0%	4	14	9.6	438.0	0.0%	2	14
	1.0	77.9	486.0	0.0%	5	10	30.0	464.9	0.0%	4	13	14.6	459.9	0.0%	3	13
	1.5	122.5	512.0	0.0%	5	10	31.2	493.9	0.0%	4	13	19.5	488.9	0.0%	3	13
	2.0	318.5	544.3	0.0%	6	12	182.2	533.6	0.0%	5	14	174.4	533.6	0.0%	5	14
	2.5	2227.3	596.3	0.0%	7	14	2312.4	596.3	0.0%	4	13	2536.8	596.3	0.0%	4	13
	3.0	10800.2	659.1	7.4%	6	12	10800.1	658.9	7.1%	5	14	10800.3	658.9	8.6%	7	15
R101	0.5	7.3	618.0	0.0%	7	14	1.9	603.0	0.0%	4	14	1.4	594.0	0.0%	2	14
	1.0	8.2	644.4	0.0%	8	14	8.8	629.4	0.0%	5	14	6.3	624.4	0.0%	4	14
	1.5	18.0	667.8	0.0%	7	14	11.5	656.0	0.0%	4	14	13.4	656.0	0.0%	4	14
	2.0	57.4	708.4	0.0%	8	14	54.0	700.2	0.0%	5	14	49.2	700.2	0.0%	4	14
	2.5	320.6	771.4	0.0%	6	12	373.5	771.4	0.0%	5	13	397.4	771.4	0.0%	6	13
	3.0	1101.4	827.4	0.0%	8	15	991.5	827.4	0.0%	7	13	815.5	827.4	0.0%	6	14
RC101	0.5	4829.3	432.0	0.0%	4	8	5458.4	413.6	0.0%	3	12	1115.3	394.0	0.0%	2	14
	1.0	10805.8	461.0	12.9%	4	7	2617.2	420.0	0.0%	4	14	1295.8	411.1	0.0%	3	14
	1.5	10800.3	491.9	14.4%	4	8	2207.8	455.3	0.0%	3	12	1986.0	455.3	0.0%	3	14
	2.0	10800.2	530.1	11.7%	4	7	10800.2	504.1	4.2%	4	14	10800.3	504.1	3.4%	4	14
	2.5	10800.2	571.3	12.2%	4	7	10800.2	558.3	10.1%	4	14	10800.2	558.3	7.9%	4	15
	3.0	10800.2	626.3	19.0%	4	7	10800.2	613.3	10.8%	4	14	10800.2	613.3	9.8%	3	13

Table 12: Results of the MILP model (1)–(30) obtained by CPLEX with a maximum run-time of 3 hours over instances of 20 customers, considering drone load capacity (A) of 2, 4, and 8 orders.

Instance	β	$A = 2$					$A = 4$					$A = 8$				
		CPU time	Ob. value	GAP	n_{drone}	o_r	CPU time	Ob. value	GAP	n_{drone}	o_r	CPU time	Ob. value	GAP	n_{drone}	o_r
C101	0.5	10807.4	334.0	6.7%	1	2	5262.2	317.0	0.0%	3	11	907.3	307.0	0.0%	3	19
	1.0	10800.5	371.4	9.0%	4	7	10803.6	346.4	4.5%	4	16	7895.2	336.1	0.0%	3	18
	1.5	10800.5	411.2	20.1%	3	6	10808.7	389.3	17.8%	4	14	10800.4	384.3	12.3%	3	15
	2.0	10810.3	448.1	21.7%	5	9	10802.7	419.8	16.4%	5	16	10806.1	409.3	14.2%	4	18
	2.5	10807.5	490.2	17.4%	6	11	10805.9	453.9	10.8%	5	15	10806.3	455.0	11.0%	4	17
	3.0	10801.0	520.5	12.0%	7	12	10820.1	486.9	5.9%	5	14	10810.0	486.9	5.9%	5	16
C201	0.5	694.3	494.2	0.0%	7	14	354.6	467.9	0.0%	5	19	61.4	457.9	0.0%	3	19
	1.0	9285.8	514.2	0.0%	7	14	477.7	490.4	0.0%	5	19	201.3	480.4	0.0%	3	19
	1.5	10812.5	550.5	3.4%	6	12	689.0	522.7	0.0%	5	19	242.6	514.9	0.0%	3	20
	2.0	10801.8	577.3	7.2%	7	14	6367.6	556.9	0.0%	5	19	10801.4	556.9	1.8%	5	19
	2.5	10800.7	643.3	16.9%	8	16	10800.5	633.3	14.3%	6	19	10800.5	633.3	14.0%	5	19
	3.0	10803.5	719.1	15.9%	6	11	10800.4	690.9	12.4%	5	17	10802.5	690.9	12.4%	5	19
R101	0.5	161.3	717.7	0.0%	10	19	38.1	692.7	0.0%	5	19	15.5	682.7	0.0%	3	19
	1.0	917.0	750.4	0.0%	10	19	214.1	730.4	0.0%	6	19	121.2	723.0	0.0%	3	19
	1.5	1103.8	772.0	0.0%	10	19	843.5	752.0	0.0%	6	19	351.3	747.0	0.0%	4	19
	2.0	10811.3	862.4	11.8%	7	14	10807.5	857.4	12.7%	5	16	10803.9	810.4	8.3%	5	18
	2.5	10803.9	900.4	12.5%	8	16	10804.0	895.4	12.1%	6	17	10802.9	895.4	15.0%	6	19
	3.0	10804.7	985.4	18.7%	6	11	10800.7	960.4	17.8%	7	19	10800.3	980.4	19.5%	5	19
RC101	0.5	10819.2	624.0	32.3%	1	2	10818.5	605.7	31.7%	3	12	10816.2	568.9	28.9%	3	19
	1.0	10814.6	686.0	31.5%	4	7	10815.5	628.9	26.9%	3	12	10806.0	604.1	22.4%	3	17
	1.5	10810.4	739.8	31.9%	4	8	10802.7	682.9	21.2%	3	11	10803.1	681.3	23.5%	2	11
	2.0	10806.9	797.2	30.9%	5	10	10803.2	745.3	28.1%	5	18	10804.1	742.1	22.0%	3	17
	2.5	10800.5	844.2	30.9%	5	10	10800.6	818.3	28.6%	5	19	10803.1	818.3	26.9%	3	16
	3.0	10803.9	917.1	26.7%	5	10	10802.6	907.5	25.9%	5	19	10800.5	854.5	21.3%	4	19

D Time Savings of the Decomposition on Larger Instances

Figure 7 shows the percentage of time savings of the decomposition solutions over the results achieved by the MILP model of Archetti et al. (2018) (see Table 8 of the Appendix). As can be seen, the percentage of time savings was always positive, which means that the decomposition provided lower delivery times than the truck only model.

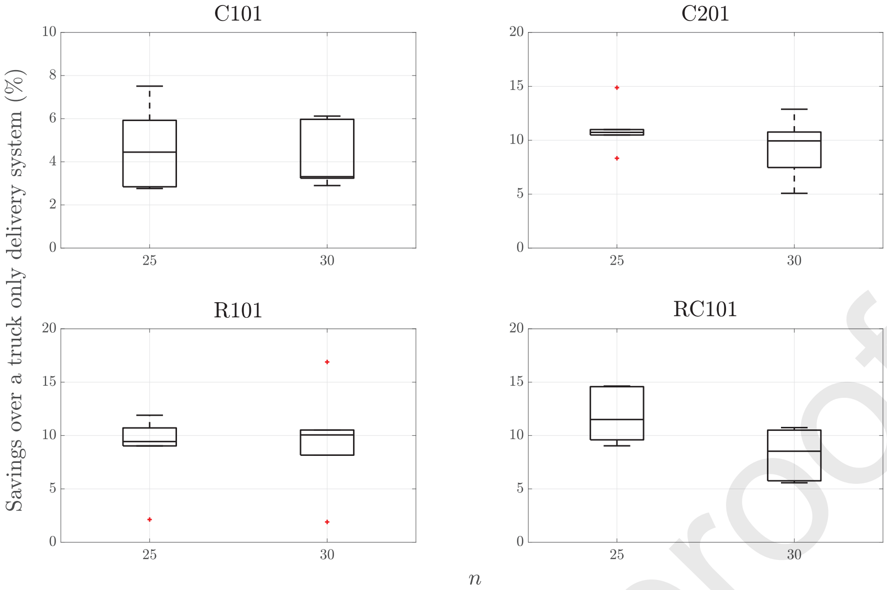


Figure 7: Percentage of time savings of the decomposition approach over a truck only delivery system (Archetti et al. (2018) MILP model) using instances of 25 and 30 customers and a drone load capacity of 4 orders.

References

Archetti, C., & Bertazzi, L. (2020). Recent challenges in routing and inventory routing: e-commerce and last-mile delivery. *Networks.*

Archetti, C., Feillet, D., Mor, A., & Speranza, M. G. (2018). An iterated local search for the traveling salesman problem with release dates and completion time minimization. *Computers & Operations Research*, 98, 24-37.

Archetti, C., Feillet, D., & Speranza, M. G. (2015). Complexity of routing problems with release dates. *European Journal of Operational Research*, 247(3), 797-803.

Avidrone Aerospace (2019). Avidrone Aerospace. Retrieved October 2020. Published online on URL: https://www.avidroneaerospace.com

Cattaruzza, D., Absi, N., & Feillet, D. (2016). The multi-trip vehicle routing problem with time windows and release dates. *Transportation Science*, 50(2), 676-693.

Capgemini Research Institute (2019). The last-mile delivery challenge: Giving retail and consumer product customers a superior delivery experience without impacting profitability. Published online on: https://bit.ly/31kW0jm

Chung, S.H., Sah, B., & Lee, J (2020). Optimization for drone and drone-truck combined operations: A review of the state of the art and future directions. *Computers & Operations Research*, 105004.

Dayarian, I., Savelsbergh, M., & Clarke, J. P. (2020). Same-day delivery with drone resupply. *Transportation Science*, 54(1), 229-249.

International Brotherhood of Teamsters (2018). National Master United Parcel Service Agreement. Published online on URL: https://bit.ly/3oefjVe

Kitjacharoenchai, P., Ventresca, M., Moshref-Javadi, M., Lee, S., Tanchoco, J. M., & Brunese, P. A. (2019). Multiple traveling salesman problem with drones: Mathematical model and heuristic approach. *Computers & Industrial Engineering*, 129, 14-30.

Klapp, M. A., Erera, A. L., & Toriello, A. (2018). The dynamic dispatch waves problem for same-day delivery. *European Journal of Operational Research*, 271(2), 519-534.

Klapp, M. A., Erera, A. L., & Toriello, A. (2016). The one-dimensional dynamic dispatch waves problem. *Transportation Science*, 52(2), 402-415.

Li, W., Wu, Y., Kumar, P. R., & Li, K. (2019). Multi-trip vehicle routing problem with order release time. *Engineering Optimization*, 1-16.

McCunney, B., & Van Cauwenberghe, K. (2019). Simulation test bed for drone-supported logistics systems. Master dissertation, Massachusetts Institute of Technology.

Mor, A., & Speranza, M. G. (2020). Vehicle routing problems over time: a survey. *4OR*, 1-21.

Murray, C., & Chu, A. (2015). The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery. *Transportation Research*, 54, 86-109.

Otto, A., Agatz, N., Campbell, J., Golden, B., & Pesch, E. (2018). Optimization approaches for civil applications of unmanned aerial vehicles (UAVs) or aerial drones: A survey. *Networks*, 72(4), 411-458.

Poikonen, S., & Golden, B. (2020). Multi-visit drone routing problem. *Computers & Operations Research*, 113, 104802.

Poikonen, S., Wang, X., & Golden, B. (2017). The vehicle routing problem with drones: Extended models and connections. *Networks*, 70(1), 34-43.

Reyes, D., Erera, A. L., & Savelsbergh, M. W. (2018). Complexity of routing problems with release dates and deadlines. *European Journal of Operational Research*, 266(1), 29-34.

Schermer, D., Moeini, M., & Wendt, O. (2018). Algorithms for solving the vehicle routing problem with drones. In *Asian Conference on Intelligent Information and Database Systems. Springer*, 352-361.

Schermer, D., Moeini, M., & Wendt, O. (2019a). A hybrid VNS/tabu search algorithm for solving the vehicle routing problem with drones and en route operations. *Computers & Operations Research*, 109, 134-158.

Schermer, D., Moeini, M., & Wendt, O. (2019b). A matheuristic for the vehicle routing problem with drones and its variants. *Transportation Research Part C: Emerging Technologies*, 106, 166-204.

Shelbourne, B., Battarra, M., Potts, C. (2017). The vehicle routing problem with release and due dates. *INFORMS Journal on Computing*, 29(4), 705-723.

Solomon, M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, 35, 254–265.

Ulmer, M. W., Thomas, B. W., & Mattfeld, D. C. (2019). Preemptive depot returns for dynamic same-day delivery. *EURO Journal on Transportation and Logistics*, 8(4), 327-361.

Wang, X., Poikonen, S., & Golden, B. (2017). The vehicle routing problem with drones: several worst-case results. *Optimization Letters*, 11(4), 679-697.

World Economic Forum (2020). The future of the last-mile ecosystem. Published online on URL: https://www.weforum.org/reports/the-future-of-the-last-mile-ecosystem