Transportation Research Part C 110 (2020) 368–398

Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

# The multiple flying sidekicks traveling salesman problem: Parcel delivery with multiple drones

Chase C. Murray*, Ritwik Raj

*Department of Industrial & Systems Engineering, University at Buffalo, 309 Bell Hall, Buffalo, NY 14260, USA*

ARTICLE INFO

*Keywords:*
Unmanned aerial vehicle
Drone
Vehicle routing problem
Traveling salesman problem
Logistics
Integer programming
Heuristics

ABSTRACT

This paper considers a last-mile delivery system in which a delivery truck operates in coordination with a fleet of unmanned aerial vehicles (UAVs, or drones). Deploying UAVs from the truck enables customers located further from the depot to receive drone-based deliveries. The problem is first formulated as a mixed integer linear program (MILP). However, owing to the computational complexity of this problem, only trivially-sized problems may be solved directly via the MILP. Thus, a heuristic solution approach that consists of solving a sequence of three subproblems is proposed. Extensive numerical testing demonstrates that this approach effectively solves problems of practical size within reasonable runtimes. Additional analysis quantifies the potential time savings associated with employing multiple UAVs. The analysis also reveals that additional UAVs may have diminishing marginal returns. An analysis of five different endurance models demonstrates the effects of these models on UAV assignments. The model and heuristic also support anticipated future systems that feature automation for UAV launch and retrieval.

## 1. Introduction

This paper introduces the *multiple flying sidekicks traveling salesman problem* (mFSTSP), in which a delivery truck and a heterogeneous fleet of unmanned aerial vehicles (UAVs, commonly called drones) coordinate to deliver small parcels to geographically distributed customers. Each UAV may be launched from the truck to deliver a single customer package, and then rendezvous (return) to the truck to be loaded with a new parcel or transported to a new launch location. The objective of the problem is to leverage the delivery truck and the fleet of UAVs to complete the delivery process and return to the depot in the minimum amount of time.

The problem of pairing UAVs with traditional delivery trucks was first introduced by Murray and Chu (2015). The paper provided a mathematical programming formulation and a simple heuristic for the problem of coordinating a single traditional delivery truck with a single UAV, dubbed the *flying sidekick traveling salesman problem* (FSTSP). Nearly identical problems have been described by other researchers under the name *TSP with drone* (e.g., Agatz et al., 2018; Bouman et al., 2018; Ha et al., 2018).

A number of industry implementations have occurred since the original FSTSP paper appeared. For example, Amazon made its first delivery worldwide (Wells and Stevens, 2016) and its first U.S. delivery a few months later (Rubin, 2017). However, drone-maker Flirtey beat Amazon to several milestones, including the first Federal Aviation Administration (FAA) approved U.S. drone delivery (Vanian, 2016). Logistics solution provider UPS also entered the drone delivery race, teaming with UAV manufacturers Zipline to deliver blood for lifesaving transfusions in Rwanda (Tilley, 2016), CyPhy Works to deliver medical supplies in the U.S. (Carey, 2016), and electric truck and UAV manufacturer Workhorse to demonstrate a truck/drone tandem (Peterson and Dektas,

* Corresponding author.
*E-mail addresses:* cmurray3@buffalo.edu (C.C. Murray), r28@buffalo.edu (R. Raj).

https://doi.org/10.1016/j.trc.2019.11.003
Received 20 February 2019; Received in revised form 29 October 2019; Accepted 2 November 2019
Available online 13 December 2019

2017). Mercedes-Benz also revealed a concept for a drone delivery van that automatically loads UAVs with parcels without the need for a driver (Etherington, 2017). The 2016 Material Handling Industry (MHI) Annual Industry Report (The, 2016) notes that 59% of its survey respondents believe that emerging technologies like drones are already having an impact on supply chains. The report also claims that adoption rates for technologies like drones are expected to grow to 50% over the next decade.

The present paper extends the original FSTSP, as well as other related studies on truck/UAV routing, in several key respects. It features a more comprehensive treatment of the operating conditions, to better reflect the realities associated with the complex nature of this coordinated vehicle routing problem. Specifically, it contributes to the existing literature in the following ways. First, the mFSTSP considers an arbitrary number of *heterogeneous* UAVs that may be deployed from the depot or from the delivery truck. These UAVs may have different travel speeds, payload capacities, service times, and flight endurance limitations. Accounting for these differences accommodates service providers who may expand their fleet with a variety of drones over time. For example, Amazon's UAV designs have evolved from eight-rotor octocopters, to a multi-rotor/fixed-wing hybrid, to the latest four-rotor quadcopter in just a few years (Amazon, 2019). Given the rapidly-changing technology, it is reasonable to assume that companies will augment their fleet over time with improved drones. Thus, as companies change the composition of their fleets (either via expansion or via replacement of drones no longer fit for service), it is expected that companies will increasingly operate heterogeneous fleets. Similarly, model variations for evolving automated UAV launch and recovery systems (at the depot or within the truck) are also provided.

Second, because the delivery truck is typically too small to safely accommodate multiple drones landing or launching simultaneously, the mFSTSP explicitly queues the aircraft in both the launch and retrieval phases. This additional scheduling problem adds complexity to the problem, but more accurately reflects the limitations associated with deploying multiple drones from a relatively small space. Consideration of queuing of these activities is important because they may comprise a significant portion of the total operational time; disregarding this aspect may result in solutions leading to UAVs running out of power before being safely retrieved.

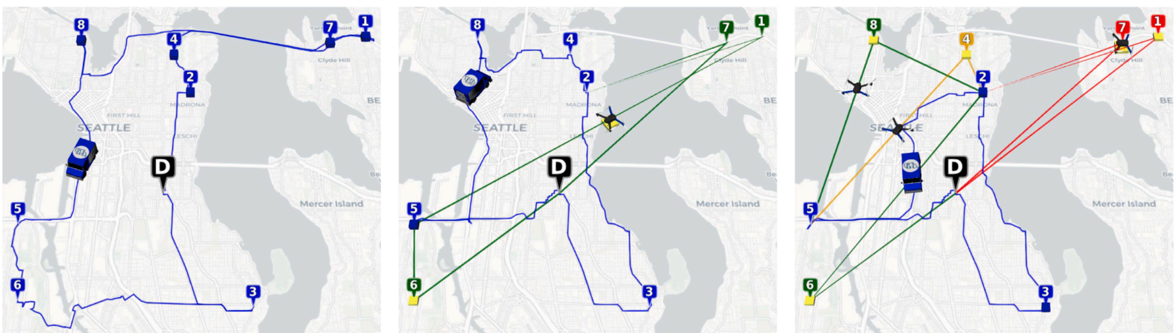
Finally, this paper provides a comparative analysis of five endurance models, one of which (a non-linear model that determines energy consumption as a function of velocity and parcel weight) has not been applied to truck/UAV routing problems. The analysis highlights the potential risks associated with constructing schedules based on overly-simplified endurance models. As with the queue scheduling, the primary risk of using an optimistic endurance model is solutions resulting in UAVs that fail to return to their recovery locations.

A small eight-customer example is provided to highlight the time savings that may be afforded by the mFSTSP. Fig. 1 shows a comparison of vehicle routes for customers located in the Seattle area. The route generated by solving a standard traveling salesman problem (TSP) demonstrates the long travel distance that must be covered if a single truck makes all eight deliveries. With the addition of one UAV, the truck visits only five of the customers and avoids the eastern half of the region. When three UAVs are employed, the truck needs to visit only three customers. The Gantt chart in Fig. 2 highlights the coordination required. In particular, as more UAVs are utilized, the truck driver must spend more time launching and retrieving the drones. Additionally, the drones may spend more time waiting for the truck to arrive at the recovery location. Table 1 reveals the significant time savings for even trivially-sized problems.

The remainder of this paper is organized as follows. An overview of related academic literature is provided in Section 2. A formal problem definition and mixed integer linear programming (MILP) formulation are provided in Section 3. A three-phased heuristic solution approach is proposed in Section 4, followed by a numerical analysis in Section 5 to highlight the benefits, and limitations, of deploying multiple drones from the delivery truck. Additional analysis explores the impacts of the region size, potential automation improvements, and implications of different endurance models. Finally, conclusions and future research directions are provided in Section 6.

## 2. Related literature

This review focuses on problems involving the coordinated use of trucks and UAVs for parcel delivery. While beyond this scope, we note another class of problems that consider the use of only UAVs to make deliveries (i.e., without a truck). Variants of this



(a) TSP Solution (no UAVs) (b) mFSTSP with 1 UAV (c) mFSTSP with 3 UAVs

**Fig. 1.** A comparison of routes for a small eight-customer problem in the Seattle region.

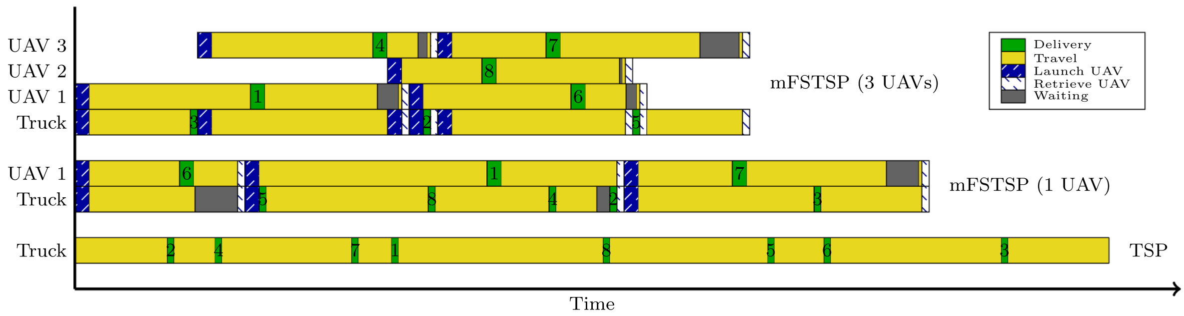



**Fig. 2.** Detailed vehicle timing for the small Seattle example, using either a single truck (TSP), a truck with one UAV, or a truck with three UAVs. Numbers on the bars identify each of the eight customers.

**Table 1**
Comparison of time savings for the small Seattle example.

| | Makespan [hr:min:sec] | Time Savings [hr:min:sec] | **Improvement over TSP** |
|---|---|---|---|
| 3 UAVs | 0:47:00 | 0:25:01 | 34.7% |
| 1 UAV | 0:59:29 | 0:12:32 | 17.4% |
| TSP | 1:12:01 | – | – |

problem include variable UAV battery energy (Dorling et al., 2017; Venkatachalam et al., 2017; Cheng et al., 2018), multi-objective (San et al., 2016), multiple UAV replenishment locations (Song et al., 2018), and coordinated UAVs (Oh et al., 2018). Additionally, numerous works have considered the benefits of UAVs in a variety of non-military applications. Recent examples include UAV logistics infrastructure (Shavarani et al., 2018; Hong et al., 2018; Kim and Awwad, 2017; Chauhan et al., 2019), healthcare (Scott and Scott, 2018; Kim et al., 2017), and disaster response (Rabta et al., 2018; Chowdhury, 2018; Zhong et al., 2018). A survey of the literature on UAVs for civil applications is provided by Otto et al. (2018).

The problem of combining a drone with a traditional delivery truck for parcel delivery was first formally defined by Murray and Chu (2015). That paper introduced an MILP formulation for the FSTSP, and also defined the parallel drone scheduling TSP (PDSTSP), where multiple drones are launched from the depot to serve nearby customers, independent of the truck delivery. Greedy construction heuristics for both problems were provided.

Numerous studies have since explored variations of the single-truck single-drone problem, often called the *TSP with drone* (TSP-D). For example, Agatz et al. (2018) provided a new MILP model for the TSP-D, as well as several route first-cluster second heuristics. An improved formulation for the FSTSP was presented by Dell'Amico et al. (2019a). Ponza (2016), Ha et al. (2018), Freitas and Penna (2018), Daknama and Kraus (2017), and Schermer et al. (2018) explored neighborhood search based heuristics, while Bouman et al. (2018) and Tang et al. (2019) present dynamic programming and constraint programming approaches, respectively, for obtaining optimal solutions. A multi-objective variant of the TSP-D, with a non-dominated sorting genetic algorithm, was proposed by Wang et al. (2019b). Poikonen et al. (2019) provide four branch-and-bound-based heuristics for the TSP-D. Jeong et al. (2019) modified the FSTSP to consider variable UAV energy consumption and restricted flying areas. Dukkanci et al. (2019) consider a variation of the FSTSP that minimizes the operational cost and calculate UAV energy consumption as a function of speed.

In the case of single-truck multi-UAV problems, Ferrandez et al. (2016) and Chang and Lee (2018) consider a system in which the truck deploys multiple drones from distributed launch sites along the truck's route. The drones return to the truck before the truck departs for its next destination. Clustering heuristics have been developed, such that the truck is routed to each cluster and nearby customers are served via UAV. Conversely, similar to the mFSTSP, Yoon (2018) consider a single truck that may launch multiple UAVs, with the UAVs returning to the truck at a different location. An MILP formulation is provided, which is tested on instances with up to 10 customers. Tu et al. (2018) propose an adaptive large neighborhood search heuristic for a similar problem, the TSP with multiple drones (TSP-mD).

The use of multiple trucks and multiple UAVs is considered by Kitjacharoenchai et al. (2019), which is an extension of the FSTSP, but without endurance limitations and launch/delivery time considerations. While UAVs can be launched from and retrieved at different trucks, only one UAV can be launched or retrieved at any customer location. They present an MILP formulation and propose an insertion based heuristic to solve problems with up to 50 customers. Sacramento et al. (2019) extend the FSTSP with multiple trucks, each carrying a single UAV. A solution approach based on adaptive large neighborhood search is provided. Wang and Sheu (2019) consider a problem in which multiple UAVs may be launched from multiple trucks at customer locations, where each UAV can serve multiple customers on a sortie. UAVs must be retrieved at separate docking locations. A branch-and-price algorithm is demonstrated on problems with up to 15 customers. Schermer et al. (2019) address a multi-truck problem in which each truck may have multiple UAVs. A matheuristic is proposed for larger-scale instances. Additionally, Schermer et al. (2019) consider a multi-truck, multi-UAV problem where UAV launches and retrievals may occur at non-customer locations, termed *en route operations*. An MILP formulation for this problem is provided, along with a variable neighborhood search heuristic.

Work related to the PDSTSP includes new heuristics proposed by Mbiadou Saleu et al. (2018) and Dell'Amico et al. (2019b). A multi-truck variant of the PDSTSP, solved via constraint programming, is provided by Ham (2018) in which UAVs can perform both delivery and pickup activities. Kim and Moon (2019) consider a variation of the PDSTSP in which UAVs can be deployed from the depot and several drone stations. Another variant, proposed by Wang et al. (2019a), considers a fleet of trucks, each carrying a UAV, operating simultaneously with additional independent UAVs that are launched from the depot.

Another class of truck/UAV problems assume that only the UAVs may make deliveries, such as the *multi-visit drone routing problem* (MVDRP) proposed by Poikonen (2018) and the truck/drone tandems considered by Mathew et al. (2015), Bin Othman et al. (2017), Peng et al. (2019), Wikarek et al. (2019). Although not a parcel delivery application, Luo et al. (2017) proposed a two-echelon ground vehicle and UAV cooperative routing problem in which a truck carries one UAV which is responsible for visiting one or more surveillance targets before returning to the truck.

Several theoretical studies have shown the benefits of using a combined truck-UAV delivery system. Wang et al. (2016) introduced the vehicle routing problem with drones (VRPD), and determined bounds on the ratio of VRPD time savings versus traditional routing problems (e.g., VRP and TSP). The analysis considered particular cases where trucks and drones follow the same distance metric and drone battery life is unlimited. Poikonen et al. (2017) relaxed these assumptions to develop bounds on similar ratios, but with differing distance metrics for trucks and drones and limited drone endurance. Carlsson and Song (2017) consider a continuous approximation model to replace computationally difficult combinatorial approaches. Their *horsefly routing problem* consists of one truck and one UAV. Unlike other models, the UAV launch/retrieval locations are not restricted to customer nodes. Campbell and Sweeney (2017) and Li et al. (2018) also used continuous approximation methods, and developed cost models to study the economic impacts. Results by Campbell and Sweeney (2017) suggest that substantial cost savings can be achieved using the combined truck-drone delivery system with multiple drones per truck, and highlight the benefits associated with automated loading and reduced delivery service times. Boysen et al. (2018) study the complexity of problems involving a given set of UAV customers and a fixed truck route.

In contrast to the above studies, Ulmer and Thomas (2018) introduce a problem in which customer orders arrive dynamically. For each incoming order, the firm must decide whether a truck or UAV will make the delivery (if at all). There is no interaction between trucks and UAVs.

## 3. Problem definition and mathematical programming formulation

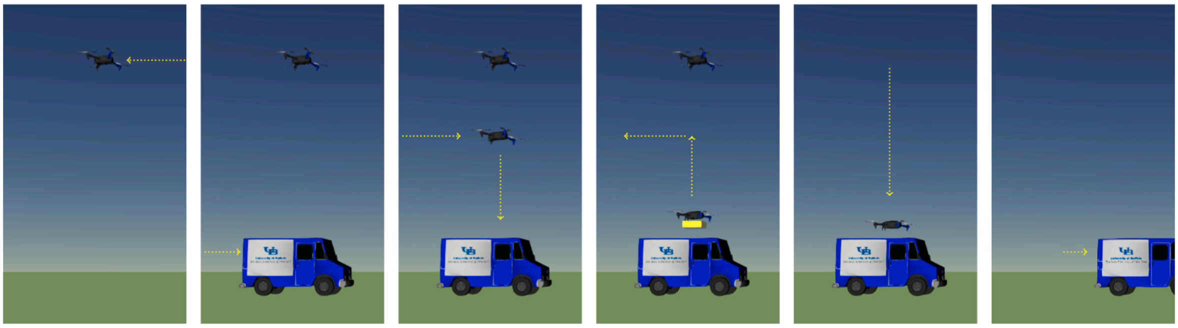
This section provides a formal definition of the mFSTSP, as well as an MILP formulation of the problem. A summary of the parameter notation is described in Table 2.

Let $C$ represent the set of all customer parcels, such that $C = \{1, 2, ..., c\}$. Each customer must receive exactly one delivery by either the single delivery truck or by one of the heterogeneous UAVs that are denoted by the set $V$. A particular customer $i \in C$ is said to be "droneable" by UAV $v \in V$, and thus belongs to the set $\widehat{C}_v$, if that customer's parcel is eligible to be delivered by $v$. This categorization may be a function of several factors, including the parcel's weight or size, whether a customer signature is required, whether the parcel contains hazardous material that should not be flown, or whether the customer's location is conducive to accommodating a drone (e.g., apartments or heavily-wooded areas may be inaccessible to a UAV).

Each UAV is capable of carrying one droneable parcel at a time, although the weight or volume capacity of each UAV may differ. UAVs may be launched from the depot, or from the truck. While a UAV can be launched multiple times, it cannot be launched from the same location more than once. This assumption, consistent with the definition of the original FSTSP, is made primarily for algorithmic convenience. A UAV can be retrieved at the depot, or by the truck at a customer location, but it cannot be retrieved at the same customer location from which it was launched. When a UAV returns to the truck, it may be loaded onto the truck or it can be launched from the truck (at this location) with a new package. The truck can also make stops between UAV launch and retrieval locations to serve other customers while UAVs are airborne. It is assumed that the truck can transport all of the available UAVs at once, although the truck may only launch or retrieve one UAV at a time.

**Table 2**
Parameter notation.

| | |
|---|---|
| $V$ | Set of UAVs. |
| $C$ | Set of customers; $C = \{1, 2, ..., c\}$. |
| $\widehat{C}_v$ | Set of customers that may be served by UAV $v \in V$; $\widehat{C}_v \subseteq C$ for all $v \in V$. |
| $N$ | Set of all nodes; $N = \{0, 1, ..., c+1\}$. |
| $N_0$ | Set of nodes from which a vehicle may depart; $N_0 = \{0, 1, ..., c\}$. |
| $N_+$ | Set of nodes to which a vehicle may visit; $N_+ = \{1, 2, ..., c+1\}$. |
| $\tau_{ij}$ | Truck's travel time from node $i \in N_0$ to node $j \in N_+$. |
| $\tau'_{vij}$ | Travel time for UAV $v \in V$ from node $i \in N_0$ to node $j \in N_+$. |
| $s^L_{v,i}$ | Launch time for UAV $v \in V$ from node $i \in N_0$. |
| $s^R_{v,k}$ | Recovery time for UAV $v \in V$ at node $k \in N_+$. |
| $\sigma_k$ | Service time by truck at node $k \in N_+$, where $\sigma_{c+1} \equiv 0$. |
| $\sigma'_{vk}$ | Service time by UAV $v \in V$ at node $k \in N_+$, where $\sigma'_{v,c+1} \equiv 0$. |
| $P$ | A set of tuples of the form $\langle v, i, j, k \rangle$, specifying all possible three-node sorties that may be flown by UAV $v \in V$. |
| $e_{vijk}$ | Endurance, in units of time, for UAV $v \in V$ traveling from nodes $i \in N_0$ to $j \in \widehat{C}_v$ to $k \in N_+$. |



(a) UAV 1 arrives (b) Truck arrives (c) UAV 2 arrives and is retrieved (d) UAV 2 is relaunched (e) UAV 1 is retrieved (f) Driver serves customer and truck departs

**Fig. 3.** A notional example depicting the scheduling activities required at truck customer locations.

For now, assume that the truck must be present at the depot when the UAVs are launched or retrieved; in Section 3.7 we address a variant of the problem for cases where the depot features automation or is sufficiently staffed to prepare and receive UAVs without the driver. We also initially assume that the driver must participate in the launch/recovery process when en route (away from the depot). However, Section 3.8 describes model modifications that leverage UAV-handling automation within the truck.

Because the truck may launch and retrieve (and re-launch) multiple UAVs at a particular customer location, it is important to coordinate these activities carefully to avoid mid-air collisions. Thus, the driver's task of dropping off a customer's parcel must also be scheduled with the UAV launch and recovery activities. Fig. 3 shows an example of flow of the scheduled activities.

To characterize the underlying network structure of the mFSTSP, we define the set of all nodes in the network to be $N = \{0, 1, \ldots, c + 1\}$, where nodes 0 and $c + 1$ represent the depot from which all vehicles must originate and return. This convention accommodates the case in which the origin depot (0) and destination depot ($c + 1$) have different physical locations. The truck may only depart from node 0, and must return to node $c + 1$. The set of nodes from which a vehicle may depart is represented by $N_0 = \{0, 1, \ldots, c\}$, while $N_+ = \{1, 2, \ldots, c + 1\}$ describes the set of all nodes to which a vehicle may visit.

The truck's travel time along the road network from node $i \in N_0$ to $j \in N_+$ is given by $\tau_{ij}$. Similarly, $\tau'_{vij}$ represents the time required for UAV $v \in V$ to fly from node $i \in N_0$ to node $j \in N_+$.

When UAV $v \in V$ is launched from node $i \in N_0$, it requires $s^L_{v,i}$ units of time. This launch time is indexed on $v$ to incorporate differences in UAVs (some of which may be better designed to load parcels or swap batteries). The launch time is also indexed on the launch location; launching from the depot may require a different amount of time (e.g., perhaps the drones are already loaded with their first parcel and already have a fresh battery, or perhaps there's automation within the depot). The recovery time, $s^R_{v,k}$, is similarly defined for UAV $v \in V$ retrieved at node $k \in N_+$.

Truck deliveries require $\sigma_k$ units of time for service at node $k \in N_+$, where $\sigma_{c+1} \equiv 0$ as there is no delivery at the depot. Similarly, UAV $v \in V$ requires $\sigma'_{vk}$ units of time to perform the delivery service at node $k \in N_+$, where $\sigma'_{v,c+1} \equiv 0$. This service time is indexed on the UAV to reflect differences in delivery mechanisms among UAVs in the fleet. For example, some UAVs deliver goods via a tether while the drone remains airborne (c.f., Google's "egg" (Mogg, 2015) and Flirtey's pizza delivery UAV (Boyle, 2016)), others require the UAV to land to release the package (c.f., DHL's 'parcelcopter' (Adams, 2016) and UPS/Workhorse truck/UAV tandem (Adams, 2017)), while other designs drop goods via parachute (c.f., Amazon's patent for a shipping label with built-in parachute (Mogg, 2017) and Zipline's blood deliveries (Toor, 2016)).

### 3.1. UAV endurance

Each UAV $v \in V$ has a unique endurance, represented as $e_{vijk}$ and measured in units of time, for which it may remain operational as it travels from node $i \in N_0$ (launch) to $j \in \hat{C}_v$ (delivery), and then to $k \in N_+$ (rendezvous). The incorporation of the UAV's endurance is critical, as UAV operations are hampered by limited battery capacity.

To identify potential valid UAV sorties (i.e., the sequence of a launch, customer delivery, and rendezvous), $P$ is defined to be a set of four-tuples of the form $\langle v, i, j, k \rangle$ for $v \in V$, $i \in N_0$, $j \in \hat{C}_v$, and $k \in N_+$. This set has the following properties:

- The launch node, $i$, must not be the ending depot node (i.e., $i$ is restricted to $N_0$);
- The delivery node, $j$, must be an eligible customer for UAV $v$ (i.e., $j \in \{\hat{C}_v: j \neq i\}$);
- The rendezvous point, $k$, may be either a customer or the ending depot (but it must not be either node $i$ or $j$); and
- The UAV's travel time from $i \to j \to k$ must not exceed the endurance of the UAV (i.e., $\tau'_{vij} + \sigma'_{vj} + \tau'_{vjk} \leqslant e_{vijk}$ for $k \in \{N_+: k \neq j, k \neq i\}$).

**Table 3**
Decision variables.

| | |
|---|---|
| $x_{ij} \in \{0, 1\}$ | $x_{ij} = 1$ if the truck travels from node $i \in N_0$ immediately to node $j \in \{N_+ : j \neq i\}$. |
| $p_{ij} \in \{0, 1\}$ | $p_{ij} = 1$ if node $i \in N_0$ appears in the truck's route before node $j \in \{C : j \neq i\}$. $p_{0j} \equiv 1$ for all $j \in C$. |
| $y_{vijk} \in \{0, 1\}$ | $y_{vijk} = 1$ if UAV $v \in V$ travels from node $i \in N_0$ to customer $j \in \{\widehat{C}_v : j \neq i\}$, re-joining the truck at node $k \in \{N_+ : \langle v, i, j, k\rangle \in P\}$. |
| $\check{t}_i \geqslant 0$ | Truck's arrival time to node $i \in N$, where $\check{t}_0 \equiv 0$. |
| $\bar{t}_i \geqslant 0$ | Truck's service time completion at node $i \in N_+$, where $\bar{t}_0 \equiv 0$. |
| $\hat{t}_i \geqslant 0$ | Truck's completion time at node $i \in N$ (e.g., the earliest departure time from this node if $i \in N_0$). If the truck is not required to be at the depot when UAVs launch, then $\hat{t}_0 \equiv 0$ |
| $\check{t}'_{vi} \geqslant 0$ | Arrival time for UAV $v \in V$ to node $i \in N$. |
| $\hat{t}'_{vi} \geqslant 0$ | Completion time for UAV $v \in V$ at node $i \in N$. |
| $z^R_{v1,v2,k} \in \{0, 1\}$ | $z^R_{v1,v2,k} = 1$ if $v_1 \in V$ and $v_2 \in \{V : v_2 \neq v_1\}$ are recovered at node $k \in N_+$, such that $v_1$ is recovered before $v_2$. |
| $z^R_{0,v,k} \in \{0, 1\}$ | $z^R_{0,v,k} = 1$ if the truck completes its service activities at node $k \in N_+$ before UAV $v \in V$ is retrieved at node $k$. |
| $z^R_{v,0,k} \in \{0, 1\}$ | $z^R_{v,0,k} = 1$ if UAV $v \in V$ is retrieved at node $k \in N_+$ before the truck completes its service activities at node $k$. We define $z^R_{v,0,c+1} \equiv 0$ for all $v \in V$ (since the truck has no service activities at the depot node, the order does not matter). |
| $z^L_{v1,v2,i} \in \{0, 1\}$ | $z^L_{v1,v2,i} = 1$ if UAV $v_1 \in V$ is launched from node $i \in N_0$ before $v_2 \in \{V : v_2 \neq v_1\}$ is launched from $i$. |
| $z^L_{0,v,i} \in \{0, 1\}$ | $z^L_{0,v,i} = 1$ if the truck completes its service activities at node $i \in N_0$ before UAV $v \in V$ is launched from $i$. |
| $z^L_{v,0,i} \in \{0, 1\}$ | $z^L_{v,0,i} = 1$ if UAV $v \in V$ is launched from node $i \in N_0$ before the truck completes its service activities at node $i$. If the truck is not required to be present when UAVs launch from the depot, we may define $z^L_{v,0,0} = 0$ for all $v \in V$ (since the truck has no service activities at the depot node, the order does not matter). |
| $z'_{v1,v2,i} \in \{0, 1\}$ | $z'_{v1,v2,i} = 1$ if UAV $v_1 \in V$ launches from node $i \in C$ before UAV $v_2 \in \{V : v_2 \neq v_1\}$ lands at $i$. |
| $z''_{v1,v2,i} \in \{0, 1\}$ | $z''_{v1,v2,i} = 1$ if UAV $v_1 \in V$ lands at node $i \in C$ before UAV $v_2 \in \{V : v_2 \neq v_1\}$ launches from $i$. |
| $1 \leqslant u_i \leqslant c + 2$ | Truck subtour elimination variables, defined for all $i \in N_+$, which indicate the relative ordering of visits to node $i$. |

### 3.2. Objective and decision variables

The objective of the mFSTSP is to minimize the time required to deliver all parcels and return to the depot (i.e., to minimize the makespan). This is accomplished via determination of decision variable values across six main classes, a summary of which is provided in Table 3. First, binary decision variable $x_{ij} = 1$ if the truck travels from node $i \in N_0$ immediately to node $j \in \{N_+ : j \neq i\}$. This decision variable determines the route of the delivery truck. Similarly, in the second class, binary decision variable $p_{ij} = 1$ if the truck visits node $i \in N_0$ at some time prior to visiting node $j \in \{C : j \neq i\}$. We define $p_{0j} \equiv 1$ for all $j \in C$ to indicate that the truck must leave the depot (node 0). This decision variable is employed to ensure that a UAV's launch and recovery nodes are consistent with the truck's route (i.e., if a UAV is launched from a truck, it cannot return to the truck at a location that was earlier in the truck's route).

In the third class, binary decision variable $y_{vijk} = 1$ if UAV $v \in V$ travels from node $i \in N_0$ to customer $j \in \{\widehat{C}_v : j \neq i\}$, re-joining the truck at node $k \in \{N_+ : \langle v, i, j, k\rangle \in P\}$. This decision variable identifies UAV sorties.

The fourth class involves five continuous decision variables to determine the time at which key events for the truck and UAVs occur. Specifically, $\check{t}_i \geqslant 0$ captures the truck's arrival time to node $i \in N$, where $\check{t}_0 \equiv 0$ to indicate that the truck is available to begin operations at time zero. The truck's service time completion at node $i \in N_+$ is given by $\bar{t}_i \geqslant 0$, where $\bar{t}_0 \equiv 0$ to reflect the fact that there is no customer associated with the depot. This decision variable indicates the time at which customer $i$'s parcel has been delivered. Next, $\hat{t}_i \geqslant 0$ identifies the truck's completion time at node $i \in N$ (e.g., the earliest departure time from this node if $i \in N_0$). In the problem variant where the truck is not required to be at the depot when UAVs launch, then $\hat{t}_0 \equiv 0$. Similarly, timing for the UAVs is determined by $\check{t}'_{vi} \geqslant 0$, which denotes the arrival time for UAV $v \in V$ to node $i \in N$, and $\hat{t}'_{vi} \geqslant 0$, which identifies the completion time for UAV $v \in V$ at node $i \in N$.

Next, numerous binary decision variables (all identified by the letter $z$ with sub- and super-scripts) are employed to determine the coordination between the driver and each UAV, and to establish the sequencing of UAV launches and retrievals at each node. Details on each of these variables are provided in Table 3.

Finally, $1 \leqslant u_i \leqslant c + 2$ are standard truck subtour elimination variables, defined for all $i \in N_+$, that indicate the relative ordering of visits to node $i$.

Details of the MILP formulation are provided in the remainder of this section. Due to the length of the model, constraints are grouped according to functionality.

### 3.3. Core model components from the FSTSP

The mFSTSP leverages core components of the FSTSP model provided by Murray and Chu (2015), with modifications to accommodate multiple UAVs. This model employs several "big-M" constraints, where the value of $M$ represents a sufficiently large number. One valid value of $M$ is the length of a TSP tour such that the truck visits each customer, plus the sum of truck service times over all customers.

The objective function and the general constraints related to guaranteeing customer deliveries and eliminating truck subtours are as follows:

$$\text{Min} \quad \hat{t}_{c+1} \tag{1}$$

$$\text{s. t.} \quad \sum_{\substack{i \in N_0 \\ i \neq j}} x_{ij} + \sum_{v \in V} \sum_{\substack{i \in N_0 \\ i \neq j}} \sum_{\substack{k \in N_+ \\ \langle v,i,j,k \rangle \in P}} y_{vijk} = 1 \quad \forall \ j \in C, \tag{2}$$

$$\sum_{j \in N_+} x_{0j} = 1, \tag{3}$$

$$\sum_{i \in N_0} x_{i,c+1} = 1, \tag{4}$$

$$\sum_{\substack{i \in N_0 \\ i \neq j}} x_{ij} = \sum_{\substack{k \in N_+ \\ k \neq j}} x_{jk} \quad \forall \ j \in C, \tag{5}$$

$$\sum_{\substack{j \in C \\ j \neq i}} \sum_{\substack{k \in N_+ \\ \langle v,i,j,k \rangle \in P}} y_{vijk} \leqslant 1 \quad \forall \ i \in N_0, \ v \in V, \tag{6}$$

$$\sum_{\substack{i \in N_0 \\ i \neq k}} \sum_{\substack{j \in C \\ \langle v,i,j,k \rangle \in P}} y_{vijk} \leqslant 1 \quad \forall \ k \in N_+, \ v \in V, \tag{7}$$

$$2y_{vijk} \leqslant \sum_{\substack{h \in N_0 \\ h \neq i}} x_{hi} + \sum_{\substack{l \in C \\ l \neq k}} x_{lk} \quad \forall \ v \in V, \ i \in C, \ j \in \{C: j \neq i\}, \ k \in \{N_+: \langle v, i, j, k \rangle \in P\}, \tag{8}$$

$$y_{v0jk} \leqslant \sum_{\substack{h \in N_0 \\ h \neq k}} x_{hk} \quad \forall \ v \in V, \ j \in C, \ k \in \{N_+: \langle v, 0, j, k \rangle \in P\}, \tag{9}$$

$$u_k - u_i \geqslant 1 - (c+2)\left(1 - \sum_{\substack{j \in C \\ \langle v,i,j,k \rangle \in P}} y_{vijk}\right) \quad \forall \ i \in C, \ k \in \{N_+: k \neq i\}, \ v \in V, \tag{10}$$

$$u_i - u_j + 1 \leqslant (c+2)(1 - x_{ij}) \quad \forall \ i \in C, \ j \in \{N_+: j \neq i\}, \tag{11}$$

$$u_i - u_j \geqslant 1 - (c+2)p_{ij} \quad \forall \ i \in C, \ j \in \{C: j \neq i\}, \tag{12}$$

$$u_i - u_j \leqslant -1 + (c+2)(1 - p_{ij}) \quad \forall \ i \in C, \ j \in \{C: j \neq i\}, \tag{13}$$

$$p_{ij} + p_{ji} = 1 \quad \forall \ i \in C, \ j \in \{C: j \neq i\}. \tag{14}$$

The objective function (1) seeks to minimize the latest time at which either the truck or a UAV return to the depot. Although $\hat{t}_{c+1}$ is explicitly defined for only the truck's return time to the depot, constraints in Sections 3.4 and 3.5 serve to link the UAVs' and truck's return time to the depot. Thus, the objective function is equivalent to $\min\{\max_{v \in V}\{\hat{t}_{c+1}, \hat{t}'_{v,c+1}\}\}$.

Constraint (2) requires each customer to be visited exactly once. Constraint (3) ensures that the truck departs from the depot exactly once, while Constraint (4) requires the truck to return to the depot exactly once.

Constraint (5) provides flow balance for the truck, which must depart from each node that it visits (except the ending depot node), while Constraint (6) states that each UAV may launch at most once from any particular node, including the depot. Similarly, Constraint (7) indicates that each UAV may rendezvous at any particular node (including customers and the ending depot) at most once.

If a UAV is launched from *customer* $i$ and is collected by the truck at node $k$, then Constraint (8) states that the truck must be assigned to both nodes $i$ and $k$. Furthermore, Constraint (9) ensures that if a UAV launches from *the starting depot* 0 and is collected at node $k$, then the truck must be assigned to node $k$. Similarly, Constraint (10) ensures that the truck must visit $i$ before $k$ if a UAV launches from customer $i$ and is collected at node $k$. Subtour elimination constraints for the truck are provided by (11).

Constraints (12)–(14) determine the proper values of $p_{ij}$. Because $u_i$ and $p_{ij}$ describe the ordering of nodes visited by the *truck* only, values of these decision variables are inconsequential for any $i$ and $j$ that are visited only by a UAV.

### 3.4. UAV timing constraints

The following constraints establish the times at which each UAV launches from either the depot or the truck, arrives at a customer location, and returns to either the truck or the depot. These constraints also address UAV flight endurance limitations.

$$\hat{t}'_{vl} \geqslant \check{t}'_{vk} - M\left(3 - \sum_{\substack{j \in C \\ \langle v,i,j,k \rangle \in P \\ j \neq l}} y_{vijk} - \sum_{\substack{m \in C \\ m \neq i \\ m \neq k \\ m \neq l}} \sum_{\substack{n \in N_+ \\ \langle v,l,m,n \rangle \in P \\ n \neq i \\ n \neq k}} y_{vlmn} - p_{il}\right)$$
$$\forall \ i \in N_0, \ k \in \{N_+ : k \neq i\}, \ l \in \{C : l \neq i, l \neq k\}, \ v \in V \quad (15)$$

$$\hat{t}'_{vi} \geqslant \check{t}'_{vi} + s^L_{v,i} - M\left(1 - \sum_{\substack{j \in C \\ j \neq i}} \sum_{\substack{k \in N_+ \\ \langle v,i,j,k \rangle \in P}} y_{vijk}\right) \quad \forall \ v \in V, \ i \in N_0, \quad (16)$$

$$\hat{t}'_{vi} \geqslant \check{t}_i + s^L_{v,i} - M(1 - z^L_{v0i}) \quad \forall \ v \in V, \ i \in N_0, \quad (17)$$

$$\hat{t}'_{vi} \geqslant \bar{t}_i + s^L_{v,i} - M(1 - z^L_{0vi}) \quad \forall \ v \in V, \ i \in N_0, \quad (18)$$

$$\hat{t}'_{vi} \geqslant \hat{t}'_{v_2,i} + s^L_{v,i} - M(1 - z^L_{v_2,v,i}) \quad \forall \ v \in V, \ v_2 \in \{V : v_2 \neq v\}, \ i \in N_0, \quad (19)$$

$$\hat{t}'_{v_2,i} \geqslant \check{t}'_{vi} + s^L_{v_2,i} - M(1 - z''_{v,v_2,i}) \quad \forall \ v \in V, \ v_2 \in \{V : v_2 \neq v\}, \ i \in C, \quad (20)$$

$$\check{t}'_{vj} \geqslant \hat{t}'_{vi} + \tau'_{vij} - M\left(1 - \sum_{\substack{k \in N_+ \\ \langle v,i,j,k \rangle \in P}} y_{vijk}\right) \quad \forall \ v \in V, \ j \in C, \ i \in \{N_0 : i \neq j\}, \quad (21)$$

$$\check{t}'_{vj} \leqslant \hat{t}'_{vi} + \tau'_{vij} + M\left(1 - \sum_{\substack{k \in N_+ \\ \langle v,i,j,k \rangle \in P}} y_{vijk}\right) \quad \forall \ v \in V, \ j \in C, \ i \in \{N_0 : i \neq j\}, \quad (22)$$

$$\hat{t}'_{vj} \geqslant \check{t}'_{vj} + \sigma'_{vj} \sum_{\substack{i \in N_0 \\ i \neq j}} \sum_{\substack{k \in N_+ \\ \langle v,i,j,k \rangle \in P}} y_{vijk} \quad \forall \ v \in V, \ j \in C, \quad (23)$$

$$\hat{t}'_{vj} \leqslant \check{t}'_{vj} + \sigma'_{vj} + M\left(1 - \sum_{\substack{i \in N_0 \\ i \neq j}} \sum_{\substack{k \in N_+ \\ \langle v,i,j,k \rangle \in P}} y_{vijk}\right) \quad \forall \ v \in V, \ j \in C, \quad (24)$$

$$\check{t}'_{vk} \geqslant \check{t}_k + s^R_{v,k} - M(1 - z^R_{v0k}) \quad \forall \ v \in V, \ k \in N_+, \quad (25)$$

$$\check{t}'_{vk} \geqslant \bar{t}_k + s^R_{v,k} - M(1 - z^R_{0vk}) \quad \forall \ v \in V, \ k \in N_+, \quad (26)$$

$$\check{t}'_{vk} \geqslant \check{t}'_{v_2,k} + s^R_{v,k} - M(1 - z^R_{v_2,v,k}) \quad \forall \ v \in V, \ v_2 \in \{V : v_2 \neq v\}, \ k \in N_+, \quad (27)$$

$$\check{t}'_{vk} \geqslant \hat{t}'_{v_2,k} + s^R_{v,k} - M(1 - z'_{v_2,v,k}) \quad \forall \ v \in V, \ v_2 \in \{V : v_2 \neq v\}, \ k \in C, \quad (28)$$

$$\check{t}'_{vk} \geqslant \hat{t}'_{vj} + \tau'_{vjk} + s^R_{v,k} - M\left(1 - \sum_{\substack{i \in N_0 \\ \langle v,i,j,k \rangle \in P}} y_{vijk}\right) \quad \forall \ v \in V, \ k \in N_+, \ j \in \{C : j \neq k\}, \quad (29)$$

$$(\check{t}'_{vk} - S^R_{v,k}) - \hat{t}'_{vi} \leqslant e_{vijk} + M(1 - y_{vijk})$$
$$\forall \ v \in V, \ i \in N_0, \ j \in \{\hat{C}_v : j \neq i\}, \ k \in \{N_+ : \langle v, i, j, k \rangle \in P\}. \quad (30)$$

Constraint (15) prohibits individual UAV sorties from overlapping. For example, suppose that UAV $v$ launches from $i$ and returns to $k$. Further, suppose that the UAV later launches from $l$ (thus, $p_{il} = 1$). This constraint prevents the launch time from $l$, $\hat{t}'_{vl}$, from preceding the return time to $k$, $\check{t}'_{vk}$. If the UAV does not return to $k$, the UAV does not launch from $l$, or $i$ does not precede $l$, then this constraint will not be binding. This constraint requires the definition of $p_{0l} = 1$ for all $l \in C$.

Constraints (16)–(20) address launching of UAVs. The launch service (preparation) time, $s^L_{v,i}$, is included in these constraints to be consistent with the definition of $\hat{t}'_{vi}$ (the time at which UAV $v$ is launched from node $i$). Constraint (16) states that $v$ cannot launch from $i$ (to $j$ and $k$) until after $v$ has arrived at node $i$; if $v$ were transported on the truck to node $i$, $\check{t}'_{vi}$ would be a meaningless value. Per Constraint (17), UAV $v$ cannot launch from $i$ until after the truck has arrived to node $i$ if the truck customer is served after $v$ is launched (i.e., if $z^L_{v0i} = 1$). Conversely, Constraint (18) states that UAV $v$ cannot launch from $i$ until after the truck has served this customer if the truck serves the customer before $v$ is launched (i.e., if $z^L_{0vi} = 1$). In Constraint (19), UAV $v$ cannot launch from $i$ until

UAV $v_2$ has launched if $v_2$ launches before $v$ (i.e., if $z^L_{v_2,v,i} = 1$), while Constraint (20) ensures that $v_2$ does not launch until after $v$ has landed, if $v$ lands before $v_2$ launches (i.e., if $z^L_{v,v_2,i} = 1$). Note that, in Constraint (20), node $i$ is restricted to being a customer node since no other node type permits both launching and landing.

Constraints (21) and (22) govern the arrival timing for a UAV serving some customer $j$; Constraints (23) and (24) perform the same function for the departure timing. These four constraints ensure that a UAV will travel directly to the customer location, and will depart immediately after completing service. Any required loitering while waiting to be retrieved by the truck will occur at the truck's location. Constraints (25)–(29) address the landing of UAVs at retrieval locations (i.e., not at a drone delivery customer). The recovery service time (e.g., $s^R_{v,k}$) is included to be consistent with the definition of $\check{t}'_{vk}$ (the time at which $v$ is deemed to have arrived at node $k$). In particular, Constraint (25) states that $v$ can land at node $k$ as soon as the truck has arrived if the truck serves customer $k$ after retrieving $v$ (i.e., if $z^R_{v0k} = 1$). However, if the truck serves customer $k$ first (i.e., if $z^R_{0vk} = 1$), then UAV $v$ cannot land at $k$ until the truck completed this service.

If UAV $v_2$ is recovered at node $k$ before UAV $v$ (i.e., if $z^R_{v2,v,k} = 1$), Constraint (27) ensures that the arrival time for $v$ is after the arrival time for $v_2$. Similarly, if $v_2$ is launched from node $k$ before $v$ is recovered (i.e., if $z'_{v_2,v,k} = 1$), Constraint (28) requires the arrival time for $v$ to be after the launch time for $v_2$. In this constraint, $k$ is restricted to the set of customers because UAVs cannot launch and land from any other nodes. In Constraint (29), UAV $v$ cannot land at $k$ until it has launched from customer $j$ and travels from $j$ to $k$.

UAV endurance limitations are addressed by Constraint (30). If $v$ travels from $i$ to $j$ to $k$, then the difference between the arrival time at $k$ (less the recovery time, which is incorporated in $\check{t}'_{vk}$) and the departure time from $i$ must not exceed the endurance limit.

### 3.5. Truck timing constraints

The following constraints govern the arrival, service, and departure activities for the truck.

$$\check{t}_j \geqslant \hat{t}_i + \tau_{ij} - M(1 - x_{ij}) \quad \forall \;\; i \in N_0,\;\; j \in \{N_+ : j \neq i\}, \tag{31}$$

$$\bar{t}_k \geqslant \check{t}_k + \sigma_k \sum_{\substack{j \in N_0 \\ j \neq k}} x_{jk} \quad \forall \;\; k \in N_+, \tag{32}$$

$$\bar{t}_k \geqslant \check{t}'_{vk} + \sigma_k - M(1 - z^R_{v0k}) \quad \forall \;\; k \in N_+,\;\; v \in V, \tag{33}$$

$$\bar{t}_k \geqslant \hat{t}'_{vk} + \sigma_k - M(1 - z^L_{v0k}) \quad \forall \;\; k \in C,\;\; v \in V, \tag{34}$$

$$\hat{t}_k \geqslant \bar{t}_k \quad \forall \;\; k \in N_+, \tag{35}$$

$$\hat{t}_k \geqslant \check{t}'_{vk} - M\left(1 - \sum_{\substack{i \in N_0 \\ i \neq k}} \sum_{\substack{j \in C \\ \langle v,i,j,k \rangle \in P}} y_{vijk}\right) \quad \forall \;\; k \in N_+,\;\; v \in V, \tag{36}$$

$$\hat{t}_k \geqslant \hat{t}'_{vk} - M\left(1 - \sum_{\substack{l \in C \\ l \neq k}} \sum_{\substack{m \in N_+ \\ \langle v,k,l,m \rangle \in P}} y_{vklm}\right) \quad \forall \;\; k \in N_0,\;\; v \in V. \tag{37}$$

The truck's travel time is incorporated in Constraint (31), which states that the truck cannot arrive at $j$ until after it has left $i$ and traveled from $i$ to $j$.

Constraints (32)–(34) establish the truck's service time completion at a customer. Note that there is no customer service time at the depot (i.e., $\sigma_{c+1} \equiv 0$). Thus, the service completion time for the truck when visiting the depot does not have a service time component, although it does also depend on the arrival of any UAVs to the depot in the event that we require the truck to be present when UAVs arrive. In Constraint (32), the truck's service time completion at node $k$ must not be prior to arriving at the node and finishing service. If the truck does not serve $k$, then $\check{t}_k$ will be a meaningless value (probably zero). Constraint (33) states that the truck cannot complete service to customer $k$ until UAV $v$ has arrived, if $v$ is recovered at node $k$ before the truck service begins (i.e., if $z^R_{v0k} = 1$). Similarly, the truck cannot complete its service of customer $k$ until UAV $v$ has launched, if $v$ is launched from $k$ before the truck begins service (i.e., if $z^L_{vok} = 1$), as in Constraint (34). Note that $k \in C$ (rather than in $N_+$) because UAVs cannot be launched from node $c + 1$.

Constraints (35)–(37) establish the truck's earliest departure from a node. Constraint (35) prevents the truck from departing a node until it has completed serving the customer. If the truck does not serve $k$, then $\check{t}_k$ will be a meaningless value. In Constraint (36), if a UAV is retrieved at node $k$, then the truck cannot depart until that UAV has arrived. Similarly, the truck cannot depart from a node until after all UAVs have launched from that node, as per Constraint (37).

### 3.6. Sequencing of retrievals, launches, and truck service

In this section, constraints are provided to establish proper values of the binary decision variables used to sequence the activities at each node. We begin with constraints for setting the $z^R_{\cdot,\cdot,\cdot}$ values:

$$z^R_{0vk} + z^R_{v0k} = \sum_{\substack{i \in N_0 \\ i \neq k}} \sum_{\substack{j \in C \\ \langle v,i,j,k \rangle \in P}} y_{vijk} \quad \forall \ v \in V, \ k \in N_+, \tag{38}$$

$$z^R_{v,v_2,k} \leqslant \sum_{\substack{i \in N_0 \\ i \neq k}} \sum_{\substack{j \in C \\ \langle v,i,j,k \rangle \in P}} y_{vijk} \quad \forall \ v \in V, \ v_2 \in \{V: v_2 \neq v\}, \ k \in N_+, \tag{39}$$

$$z^R_{v,v_2,k} \leqslant \sum_{\substack{i \in N_0 \\ i \neq k}} \sum_{\substack{j \in C \\ \langle v_2,i,j,k \rangle \in P}} y_{v_2ijk} \quad \forall \ v \in V, \ v_2 \in \{V: v_2 \neq v\}, \ k \in N_+, \tag{40}$$

$$z^R_{v,v_2,k} + z^R_{v_2,v,k} \leqslant 1 \quad \forall \ v \in V, \ v_2 \in \{V: v_2 \neq v\}, \ k \in N_+, \tag{41}$$

$$z^R_{v,v_2,k} + z^R_{v_2,v,k} + 1 \geqslant \sum_{\substack{i \in N_0 \\ i \neq k}} \sum_{\substack{j \in C \\ \langle v,i,j,k \rangle \in P}} y_{vijk} + \sum_{\substack{i \in N_0 \\ i \neq k}} \sum_{\substack{j \in C \\ \langle v_2,i,j,k \rangle \in P}} y_{v_2ijk} \quad \forall \ v \in V, \ v_2 \in \{V: v_2 \neq v\}, \ k \in N_+. \tag{42}$$

In Constraint (38), if $v$ is retrieved at node $k$, then the truck must serve $k$ either before or after $v$ arrives. Conversely, $z^R_{v,v_2,k}$ cannot equal one if neither $v$ nor $v_2$ are retrieved at node $k$, as per Constraints (39) and (40). Constraint (41) states that either $v$ is retrieved before $v_2$, $v_2$ is retrieved before $v$, or at least one of these UAVs is not retrieved at $k$. Finally, if $v$ and $v_2$ are both retrieved at $k$, then either $v$ is retrieved before $v_2$, or $v_2$ is retrieved before $v$, as in Constraint (42).

Constraints (43)–(47), below, are the *launch* analogues to Constraints (38)–(42). These constraints are used to set the $z^L_{\cdot,\cdot,\cdot}$ decision variable values:

$$z^L_{0vi} + z^L_{v0i} = \sum_{\substack{j \in C \\ j \neq i}} \sum_{\substack{k \in N_+ \\ \langle v,i,j,k \rangle \in P}} y_{vijk} \quad \forall \ v \in V, \ i \in N_0, \tag{43}$$

$$z^L_{v,v_2,i} \leqslant \sum_{\substack{j \in C \\ j \neq i}} \sum_{\substack{k \in N_+ \\ \langle v,i,j,k \rangle \in P}} y_{vijk} \quad \forall \ v \in V, \ v_2 \in \{V: v_2 \neq v\}, \ i \in N_0, \tag{44}$$

$$z^L_{v,v_2,i} \leqslant \sum_{\substack{j \in C \\ j \neq i}} \sum_{\substack{k \in N_+ \\ \langle v_2,i,j,k \rangle \in P}} y_{v_2ijk} \quad \forall \ v \in V, \ v_2 \in \{V: v_2 \neq v\}, \ i \in N_0, \tag{45}$$

$$z^L_{v,v_2,i} + z^L_{v_2,v,i} \leqslant 1 \quad \forall \ v \in V, \ v_2 \in \{V: v_2 \neq v\}, \ i \in N_0, \tag{46}$$

$$z^L_{v,v_2,i} + z^L_{v_2,v,i} + 1 \geqslant \sum_{\substack{j \in C \\ j \neq i}} \sum_{\substack{k \in N_+ \\ \langle v,i,j,k \rangle \in P}} y_{vijk} + \sum_{\substack{j \in C \\ j \neq i}} \sum_{\substack{k \in N_+ \\ \langle v_2,i,j,k \rangle \in P}} y_{v_2ijk} \quad \forall \ v \in V, \ v_2 \in \{V: v_2 \neq v\}, \ i \in N_0. \tag{47}$$

Next, constraints are required for nodes at which one UAV launches and another UAV lands. Binary decision variable $z'_{v_1,v_2,i} = 1$ if UAV $v_1 \in V$ launches from node $i \in C$ before UAV $v_2 \in \{V: v_2 \neq v_1\}$ lands at $i$, while $z''_{v_1,v_2,i} = 1$ if UAV $v_1 \in V$ lands at node $i \in C$ before UAV $v_2 \in \{V: v_2 \neq v_1\}$ launches from $i$.

$$z'_{v_2,v,k} \leqslant \sum_{\substack{l \in C \\ l \neq k}} \sum_{\substack{m \in N_+ \\ \langle v_2,k,l,m \rangle \in P}} y_{v_2,k,l,m} \quad \forall \ v_2 \in V, \ v \in \{V: v \neq v_2\}, \ k \in C, \tag{48}$$

$$z''_{v_2,v,k} \leqslant \sum_{\substack{l \in C \\ l \neq k}} \sum_{\substack{m \in N_+ \\ \langle v,k,l,m \rangle \in P}} y_{v,k,l,m} \quad \forall \ v \in V, \ v_2 \in \{V: v_2 \neq v\}, \ k \in C, \tag{49}$$

$$z'_{v_2,v,k} \leqslant \sum_{\substack{i \in N_0 \\ i \neq k}} \sum_{\substack{j \in C \\ \langle v,i,j,k \rangle \in P}} y_{vijk} \quad \forall \ v \in V, \ v_2 \in \{V: v_2 \neq v\}, \ k \in C, \tag{50}$$

$$z''_{v_2,v,k} \leqslant \sum_{\substack{i \in N_0 \\ i \neq k}} \sum_{\substack{j \in C \\ \langle v_2,i,j,k \rangle \in P}} y_{v_2,i,j,k} \quad \forall \ v_2 \in V, \ v \in \{V: v \neq v_2\}, \ k \in C, \tag{51}$$

$$z'_{v_2,v,k} + z''_{v,v_2,k} + 1 \geqslant \sum_{\substack{i \in N_0 \\ i \neq k}} \sum_{\substack{j \in C \\ \langle v,i,j,k \rangle \in P}} y_{vijk} + \sum_{\substack{l \in C \\ l \neq k}} \sum_{\substack{m \in N_+ \\ \langle v_2,k,l,m \rangle \in P}} y_{v_2klm} \quad \forall \ v \in V, \ v_2 \in \{V: v_2 \neq v\}, \ k \in C, \tag{52}$$

$$z'_{v_2,v,k} + z''_{v,v_2,k} \leqslant 1 \quad \forall \ v \in V, \ v_2 \in \{V: v_2 \neq v\}, \ k \in C, \tag{53}$$

$$z'_{v_2,v,k} + z'_{v,v_2,k} \leqslant 1 \quad \forall \ v \in V, \ v_2 \in \{V: v_2 \neq v\}, \ k \in C, \tag{54}$$

$$z''_{v_2,v,k} + z''_{v,v_2,k} \leqslant 1 \quad \forall \ v \in V, \ v_2 \in \{V: v_2 \neq v\}, \ k \in C. \tag{55}$$

Constraint (48) states that $z'_{v_2,v,k} = 0$ if $v_2$ does not launch from $k$, while Constraint (49) sets $z''_{v_2,v,k} = 0$ if $v$ does not launch from $k$.

Similarly, Constraint (50) requires $z'_{v_2,v,k} = 0$ if $v$ is not retrieved at $k$ and Constraint (51) sets $z''_{v_2,v,k} = 0$ if $v_2$ is not retrieved at $k$. If $v$ is retrieved at customer $k$ and $v_2$ is launched from customer $k$, then either $v_2$ is launched before $v$ lands, or $v$ lands before $v_2$ launches, per Constraint (52).

Constraint (53) states that it is impossible for $v_2$ to be launched before $v$ lands and for $v$ to land before $v_2$ launches. Similarly, Constraint (54) indicates that $v_2$ cannot launch before $v$ lands, if $v$ launches before $v_2$ lands; while Constraint (55) states that it is impossible for $v_2$ to land before $v$ launches and for $v$ to land before $v_2$ launches.

### 3.7. Variant 1: truck not required at depot

The model above assumes that the truck must be present at the depot when UAVs are launched, and must also be at the depot when UAVs return. This reflects the case that the driver is responsible for manually performing these activities. However, the model may be relaxed to allow the UAVs to launch from, and return to, the depot independent of the driver.

We begin by modifying the definitions of two decision variables. First, we define $\hat{t}_0 \equiv 0$ to allow the truck to immediately depart from the depot (without waiting for UAVs to be launched). Second, we define $z^L_{v,0,0} \equiv 0$ for all $v \in V$; since the truck has no service activities at the depot node, it does not need to be considered in the UAV launch sequencing.

Next, Constraints (25), (26) and (33) need not be satisfied at depot node $c + 1$, since $v$ can land at the depot independent of the truck's arrival and service time completion. Thus, those constraints should be replaced by

$$\check{t}'_{vk} \geqslant \check{t}_k + s^R_{v,k} - M(1 - z^R_{v0k}) \quad \forall \ v \in V, \ k \in C, \tag{56}$$

$$\check{t}'_{vk} \geqslant \bar{t}_k + s^R_{v,k} - M(1 - z^R_{0vk}) \quad \forall \ v \in V, \ k \in C, \tag{57}$$

$$\bar{t}_k \geqslant \check{t}'_{vk} + \sigma_k - M(1 - z^R_{v0k}) \quad \forall \ v \in V, \ k \in C. \tag{58}$$

If the objective of the model is to minimize the time at which the last vehicle returns to the depot, then Constraint (36) remains as is. Otherwise, if the objective is to minimize the time at which the truck returns to the depot, then (36) should be modified to exclude depot node $k = c + 1$.

Finally, Constraint (37) should be relaxed for node $k = 0$ since the departure of UAV $v$ from the depot is now independent of the truck's departure. Thus, (37) should be replaced by

$$\hat{t}_k \geqslant \hat{t}'_{vk} - M\left(1 - \sum_{\substack{l \in C \\ l \neq k}} \sum_{\substack{m \in N_+ \\ \langle v,k,l,m\rangle \in P}} y_{vklm}\right) \quad \forall \ k \in C, \ v \in V. \tag{59}$$

### 3.8. Variant 2: automated launch and recovery systems

The default mFSTSP model assumes that the truck driver must be engaged in the UAV launch and recovery process at customer locations. However, concept vehicles have been proposed (c.f., Etherington, 2017) that automate these activities. The obvious benefit of such automation is that the truck driver can make a delivery at a customer location independent of the UAV launches and retrievals. Note, however, that the *truck* still has to be present for the launch and recovery at a customer location; the difference is that the *driver* is not required.

To accommodate automated launch and recovery systems, the following modifications to the baseline mFSTSP model are required. First, the UAV launch timing is no longer a function of the driver's service at the customer. Thus, Constraint (17) should be replaced by

$$\hat{t}'_{vi} \geqslant \check{t}_i + s^L_{v,i} - M\left(1 - \sum_{j \in \{C: j \neq i\}} \sum_{k \in \{N_+: \langle v,i,j,k\rangle \in P\}} y_{v,i,j,k}\right) \quad \forall \ v \in V, \ i \in N_0, \tag{60}$$

and Constraint (18) should be removed. Similarly, the UAV recovery timing constraints should be modified such that Constraint (25) is replaced by

$$\check{t}'_{vk} \geqslant \check{t}_k + s^R_{v,k} - M\left(1 - \sum_{i \in N_0} \sum_{j \in \{C: \langle v,i,j,k\rangle \in P\}} y_{v,i,j,k}\right) \quad \forall \ v \in V, \ k \in N_+, \tag{61}$$

and Constraint (26) is removed.

The truck service constraints in (33) and (34) should be removed, as the start of the truck driver's service at a customer is no longer dependent upon the UAV arrivals or departures. Similarly, Constraints (38) and (43), which require sequencing for driver service with recovery and launch operations, respectively, should be removed.

Finally, the decision variables $z^L_{v,0,i}$, $z^L_{0,v,i}$, $z^R_{v,0,k}$, and $z^R_{0,v,k}$ are no longer required. Although the UAVs still require queueing, driver service at a customer may start immediately (i.e., the UAV queueing is now independent of the driver's service at a truck customer).

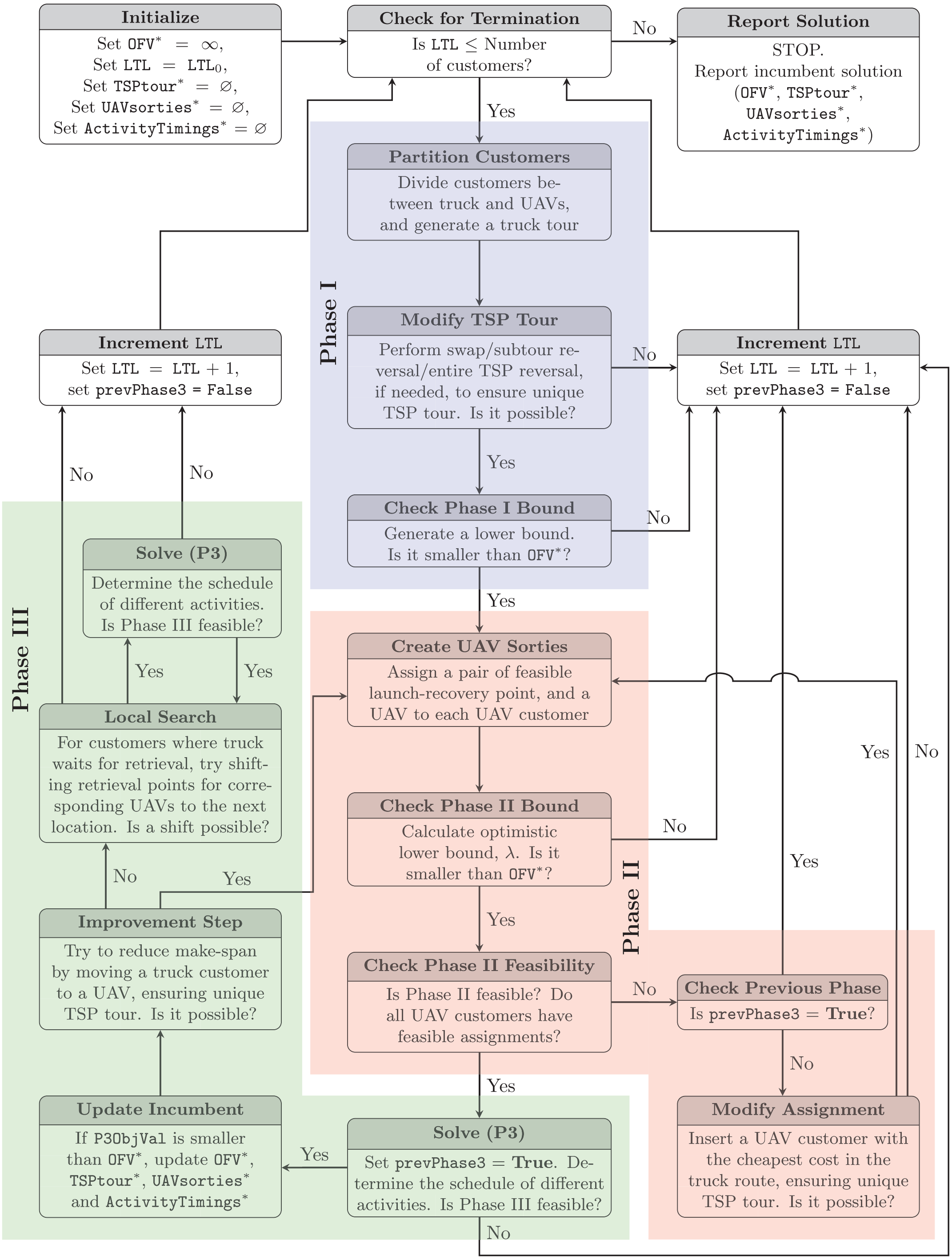



**Fig. 4.** Heuristic flowchart.

## 4. A three-phased heuristic solution approach

Due to the NP-hard nature of the mFSTSP, heuristic approaches are required for problems of practical size. A three-phased iterative heuristic, depicted in Fig. 4, is proposed.

In Phase I, customers are partitioned into two sets – those that will be served via truck and those served via UAVs. The minimum number of customers in the truck set is given by an input parameter called the *lower truck limit* (LTL). The LTL is initialized to

$$\mathrm{LTL}_0 = \left\lceil \frac{|C| - |V|}{|V| + 1} \right\rceil,$$

where $\mathrm{LTL}_0$ represents the minimum number of truck customers required for a feasible solution. For example, consider a 50-customer problem. If only 1 UAV is available, $\mathrm{LTL}_0 = 25$, indicating that at least 25 customers must be assigned to the truck route. If 4 UAVs are available, then at least 10 customers must be assigned to the truck. The value of the `LTL` is increased over the course of the iterative procedure. In addition to partitioning the customer base into truck- and UAV-assigned customers, Phase I also produces a unique TSP-like truck tour.

In Phase II, sorties for the UAV-assigned customers (as determined in Phase I) are generated. These sorties define the launch and recovery locations associated with each UAV customer, as well as the UAV assigned to each sortie. At the conclusion of Phase II, all truck and UAV routes are identified, but the timing of the activities is determined in Phase III.

In Phase III, an MILP is solved to determine the exact timing of the launch, recovery, and service activities for the truck and the UAVs. Phase III also determines the queueing sequences for the UAVs.

After Phase III is completed, a local search procedure is executed to refine the solution. The value of `LTL` is then incremented to add diversity to the search space, and the procedure returns to Phase I. The iterative procedure is repeated until the `LTL` equals the number of customers (i.e., until the problem becomes simply solving a TSP tour to visit all customers via truck). Details of each phase are described in the remainder of this section.

### 4.1. Phase I – initial customer assignments

The goal of Phase I is to establish a unique truck tour containing at least `LTL` customers, which is analogous to finding the $x_{ij}$ decision variables in the mFSTSP formulation. Any customer not on the truck's route will be allocated (if possible) in Phase II to the UAVs. Pseudocode for Phase I is provided in Algorithms 1–3.

This phase begins by creating a TSP tour of only those customers that are not UAV-eligible (lines 1–4 of Algorithm 1). The **getTSP()** function solves an MIP using the "lazy constraints" method detailed in Gurobi Optimization (2018).

Next, customers are added or removed from the truck tour (lines 6–27 of Algorithm 1) according to a `savings` metric, with the aim of reducing the makespan. The `savings` metric captures trade-offs between truck travel times, truck service time, and UAV launch and recovery times. This process is repeated until either no further improvements are found, or the `savings` metric suggests a cycle (i.e., infinite loop) of moving the same customers between the truck and UAVs.

The third step (lines 28–66 of Algorithm 2) ensures that at least `LTL` customers are served. This step also attempts to evaluate the feasibility of the UAV assignments that will be made in Phase II. Feasibility is first determined by identifying UAV customers $j$ for any UAV $v$ that do not have a corresponding valid sortie $\langle v, i, j, k\rangle \in P$ (line 31) for customers $i$ and $k$ that are currently assigned to the truck's route.

Feasibility is further assessed via the **checkP2Feasibility()** function, which solves the following objective-free integer linear program:

$$\sum\nolimits_{j \in H_i} r_{ij} \leqslant \left|V\right| \quad \forall \ i \in \mathrm{TruckCustomers} \cup \{0\}, \tag{62}$$

$$\sum\nolimits_{i \in G_j} r_{ij} = 1 \quad \forall \ j \in \mathrm{UAVCustomers}, \tag{63}$$

$$r_{ij} \in \{0, 1\} \quad \forall \ i \in \mathrm{TruckCustomers} \cup \{0\}, \ j \in \mathrm{UAVCustomers}. \tag{64}$$

The set $G_j$ contains all customers $i \in \mathrm{TruckCustomers} \cup \{0\}$ that can act as a UAV launch point for customer $j \in \mathrm{UAVCustomers}$, such that the retrieval point $k \in N_+$ is immediately after $i$ in the `TSPtour` and $\langle v, i, j, k\rangle \in P$. Similarly, $H_i$ is the set of customers $j \in \mathrm{UAVCustomers}$ that can be served by launching from customer $i \in \mathrm{TruckCustomers} \cup \{0\}$, such that the retrieval point $k \in N_+$ is immediately before $i$ in the `TSPtour` and $\langle v, i, j, k\rangle \in P$. Binary decision variable $r_{ij}$ equals 1 if $j \in \mathrm{UAVCustomers}$ is assigned to $i \in \mathrm{TruckCustomers} \cup \{0\}$ as the launch point. Constraint (62) ensures that only a maximum of $|V|$ UAVs can be launched from any launch location, while Constraint (63) requires each UAV customer to have exactly one launch point. If a feasible solution to these constraints exists, the function **checkP2Feasibility()** returns an empty set; otherwise, the function returns a set of UAV customers that do not have feasible assignments.

If any infeasible UAV customers are identified, or if the number of truck customers is less than the `LTL`, one customer at a time will be added to the truck's tour. $\mathrm{cost}_{jk}$ represents the change in the truck's makespan that would result from inserting UAV customer $j$ immediately before customer $k$ in the truck tour. $\mathrm{cover}_{jk}$ is the set of customers in `infeasCust` whose assignment would become feasible by inserting $j$ in the truck tour immediately before $k$.

In the event that inserting $j$ into the truck's route leads to a reduction in the makespan (i.e., a negative cost), the $\mathrm{score}_{jk}$ metric is calculated by multiplying $\mathrm{cost}_{jk}$ by the number of UAV customers supported by the insertion. This indicates that the benefit is shared among numerous UAV customers. Conversely, if the makespan increases by inserting $j$, $\mathrm{score}_{jk}$ is calculated as the cost per infeasible UAV customer that is being supported. Thus, if two insertions have the same cost, the one that may eliminate a larger number of infeasibilities would be preferable. In lines 52 and 53, the UAV customers $j_1^*$ and $j_2^*$ with minimum $\mathrm{cost}_{jk}$ and $\mathrm{score}_{jk}$, respectively, are identified. If there are no infeasible UAV customers but the number of truck customers is less than `LTL`, we choose to move the customer with the cheapest cost to the truck (and re-solve the TSP). However, if infeasible UAV customers have been identified and a sufficient number of truck customers have already been added, we choose to insert the customer with the cheapest score immediately before customer $k$. Otherwise (i.e., if there are infeasible UAV customers and the length of the truck route is less than `LTL`), the customer with the cheapest score is added to the truck route and a new TSP tour is generated. This process of moving customers to the truck route is continued until there are no more UAV customers with infeasible assignments and there are at least `LTL` truck customers.

Phase I continues in Line 67 of Algorithm 3, where the truck tour is perturbed if it has been previously evaluated. If such a modification is necessary, the procedure attempts to (1) swap a truck customer and a UAV customer, (2) perform a subtour reversal (i.e., $i \to j \to k \to l$ becomes $i \to k \to j \to l$), and (3) reverse the entire TSP tour. The modification that produces a unique truck route with the minimum associated cost is selected. If no unique tour is found, the value of `LTL` is increased by one and the procedure returns to Phase I. Otherwise, a lower bound is generated by using the TSP duration (including truck customer service times) and adding launch and retrieval times for the UAV customers. If the lower bound exceeds the current incumbent (OFV*), the `LTL` is updated and the procedure repeats Phase I. If the bound is less than the current incumbent, the procedure continues to Phase II.

**Algorithm 1.** Pseudocode for Phase I – Part 1 of 3

1: # Initialize:
2: TruckCust = all $j \in \{C: j \notin \hat{C}_v \ \forall\, v \in V\}$
3: UAVCust = $C \setminus$TruckCust
4: TSPtour = **getTSP**(TruckCust)
5:
6: # Reduce TSP tour cost by adding/removing truck customers:
7: **while** (improvements are possible and no cycles occur) **do**
8:   # Try to move customers from UAV to truck:
9:   **for** $j \in$ UAVCust **do**
10:     savings = $\max_{v \in V, \langle i,k \rangle \in \text{TSPtour}}(\tau_{ik} + s^L_{vj} + s^R_{vj} - \tau_{ij} - \tau_{jk} - \sigma_j)$
11:     **if** (savings $> 0$) **then**
12:       TruckCust $\leftarrow$ TruckCust $\cup \{j\}$
13:     **end if**
14:   **end for**
15:   UAVCust $\leftarrow$ UAVCust$\setminus$TruckCust
16:   TSPtour $\leftarrow$ **getTSP**(TruckCust)
17:
18:   # Try to move customers from truck to UAV:
19:   **for** $j \in$ {TruckCust: $j \in \hat{C}_v$ for any $v \in V$} **do**
20:     savings $\leftarrow \max_{\langle v,i,j,k \rangle \in P}(\tau_{ij} + \tau_{jk} + \sigma_j - \tau_{ik} - s^L_{vj} - s^R_{vj})$
21:     **if** (savings $> 0$) **then**
22:       UAVCust $\leftarrow$ UAVCust $\cup \{j\}$
23:     **end if**
24:   **end for**
25:   TruckCust $\leftarrow$ TruckCust$\setminus$UAVCust
26:   TSPtour $\leftarrow$ **getTSP**(TruckCust)
27: **end while**

**Algorithm 2.** Pseudocode for Phase I – Part 2 of 3

28: # Move customers to truck for feasibility (and to satisfy `LTL` requirement):
29: feasible = **False**     # Assume infeasible by default
30: **while** (not feasible) **do**
31:   infeasCust = all $j \in$ {UAVCust: $\langle v, i, j, k \rangle \notin P \quad \forall\ v \in V,\ i \in$ TruckCust, $k \in$ TruckCust}
32:   **If** (infeasCust = =$\emptyset$) **then**
33:     infeasCust $\leftarrow$ **checkP2Feasibility**(UAVCustomers, TSPtour)
34:   **end if**
35:   **If** ((infeasCust = =$\emptyset$)**and**(**len**(TruckCust) $\geqslant$ LTL))
36:     feasible $\leftarrow$ **True**
37:   **else**
38:     **for** $j \in$ UAVCust **do**
39:       **for** $k \in$ TruckCust $\cup \{c + 1\}$ **do**
40:         $\text{cost}_{jk} = \min_{\langle i,k \rangle \in \text{TSPtour}, \langle v,i,j,k \rangle \in P}\left(\tau_{ij} + \tau_{jk} + \sigma_j - \tau_{ik} - s^L_{vj} - s^R_{vj}\right)$
41:         $\text{cover}_{jk}$ = all $l \in$ {infeasCust: ($\langle v, m, l, j \rangle$ **or** $\langle v, j, l, m \rangle$) $\in P \quad \forall\ v \in V,\ m \in$ TruckCust}
42:         **If** ($j \in$ infeasCust) **then**
43:           $\text{cover}_{jk} \leftarrow \text{cover}_{jk} \cup \{j\}$
44:         **end if**
45:         **if** ($\text{cost}_{jk} < 0$) **then**
46:           $\text{score}_{jk} = \text{cost}_{jk}$ ***len**($\text{cover}_{jk}$)

        **else**
            $\text{score}_{jk} = \text{cost}_{jk} / \textbf{len}(\text{cover}_{jk})$
        **end if**
    **end for**
**end for**
$\langle j_1^*, k_1^* \rangle = \text{argmin}_{j \in \text{UAVCust}, k \in \text{TruckCust} \cup \{c+1\}} (\text{cost}_{jk})$
$\langle j_2^*, k_2^* \rangle = \text{argmin}_{j \in \text{UAVCust}, k \in \text{TruckCust} \cup \{c+1\}} (\text{score}_{jk})$
**if** ((infeasCust $= =\emptyset$)**and**(**len**(TruckCust) $<$ LTL)) **then**
    TruckCust $\leftarrow$ TruckCust $\cup \{j_1^*\}$
    TSPtour $\leftarrow$ **getTSP**(TruckCust)
**else if** ((infeasCust $\neq \emptyset$)**and**(**len**(TruckCust) $\geqslant$ LTL)) **then**
    TruckCust $\leftarrow$ TruckCust $\cup \{j_2^*\}$
    TSPtour $\leftarrow$ insert $j_2^*$ in current TSPtour immediately before $k_2^*$
**else**
    TruckCustomers $\leftarrow$ TruckCustomers $\cup \{j_2^*\}$
    TSPtour $\leftarrow$ **getTSP**(TruckCustomers)
**end if**
UAVCust $\leftarrow$ UAVCust$\setminus$TruckCust
**end if**
**end while**

**Algorithm 3.** Pseudocode for Phase I – Part 3 of 3

Modify TSP tour if not unique. If no unique tour is found, increase `LTL` and repeat Phase I.

\# Evaluate lower bound:
lowBound = TSPcost + $\min_{v \in V} \left\{ \sum_{j \in \text{UAVCust}} (s_{vj}^L + s_{vj}^R) \right\}$
**if** (lowBound > OFV*) **then**
    `increaseLTL() # See function below`
**else**
    Continue to Phase II.
**end if**

**procedure** `increaseLTL()`
    LTL $\leftarrow$ LTL + 1
    `prevPhase3` = **False**
    **If** (LTL $\leqslant |C|$) **then**
        `Return to Phase I with new LTL value.`
    **else**
        `Stop. Report` OFV*, TSPtour*, UAVsorties*, ActivityTimings*.
    **end if**
**end procedure**

### 4.2. Phase II – Create UAV sorties

The goal of Phase II is to determine the individual sorties $\langle v, i, j, k \rangle$ for each UAV $v$, where $i$ is the launch location, $j$ is the customer that is being served by the UAV, and $k$ is the retrieval location. This is analogous to determining the $y_{vijk}$ decision variables in the mFSTSP formulation. Pseudocode for Phase II is provided in Algorithms 4 and 5.

Inputs to this phase include the set of customers to be assigned to UAVs (`UAVcust`) and the truck route found in Phase I (`TSPtour`). Additionally, the earliest time at which the truck can arrive at each location on the route, denoted as $t_j$ for all $j \in$ TSPtour, is calculated. While $t_j$ incorporates truck customer service times, UAV launch and retrieval times are ignored. Thus, $t_j$ is sum of the arrival time at $j$ according to `TSPtour` and the service times at all previous truck customer locations.

This phase begins in lines 3–6 by initializing the lists of UAV sorties (`UAVsorties`), customers with no feasible assignments (`infeasCust`), UAVs available at each launch location (availUAVs$_j$), and unassigned UAV customers (`UnasgnCust`).

Next, in lines 8–12, the number of potential UAV sorties (numOptions$_j$) is determined for each UAV customer $j$. We define $P' \subseteq P$ such that $\langle v, i, j, k \rangle \in P'$ represents only valid $i, j, k$ combinations where $i$ and $k$ are consecutive stops on the truck's tour. Thus, in Phase II, there are no truck customers between the launch and recovery points in the UAV sorties. A local search procedure is applied in Phase III to relax this restriction (i.e., to allow the truck to make multiple stops while a UAV is en route).

In lines 14–32, UAV sorties are generated. Customers with the minimum numOptions are prioritized, as they possess the fewest number of feasible candidate sorties. For each chosen UAV customer, $j$, the assignment $\langle v, i, j, k \rangle \in P'$ is selected based on its impact on the waiting time for both the truck and the UAV. Variable `w` (line 20) captures the time difference between the UAV's activities (travel from launch point $i$ to customer $j$, serve customer $j$, and travel to retrieval point $k$) and the truck's activities (travel from launch point $i$ to retrieval point $k$). A positive value of `w` indicates that the truck must wait for the UAV. If

(WaitTime $\geqslant 0$) and (w < WaitTime), the current truck assignment incurred a waiting time that can be reduced by assigning this UAV sortie. Conversely, if WaitTime < w < 0, this UAV assignment would lead to a reduction in the time that a UAV would wait for the truck. Thus, the aim is to find an assignment that results in zero truck waiting and minimum UAV waiting, or minimum truck waiting (if zero truck waiting is not possible). If no feasible assignment is found (line 26), the customer is added to the `infeasCust` list; otherwise (line 28), the assignment is stored in the `UAVsorties` list and the corresponding UAV becomes unavailable at the corresponding launch location $i^*$.

If any UAV customers remain without an assigned sortie, it is necessary to move a UAV customer to the truck route. In lines 34–37 of Algorithm 5 we calculate `cheapestInsCost`, which is the minimum cost associated with inserting a UAV customer into the current TSP tour which might make Phase II feasible. The following costs comprise `cheapestInsCost`, and are associated with the insertion of UAV customer $i$ into position $p$ of the truck route:

$c_{i,p}^{\text{ins}}$ = Additional time associated with inserting customer i ∈`UAVCust` into position p of the truck's route.
$\tau_{h,i} + \sigma_i + \tau_{i,k} - \tau_{h,k}$,
where node $h$ is at position $p-1$ and node $k$ is at position $p$ in `TSPtour`.

$c_{i,p,j}^{\text{wait,launch}}$ = Truck waiting time if a UAV were launched from customer i ∈`UAVCust` inserted into position p ∈{`TSPtour` positons} to serve customer j ∈`infeasCust` \i.
$\min_{v \in V,\ k \in \{\text{TSPtourpositions after } p\}} \{\tau'_{v,i,j} + \sigma'_{v,j} + \tau'_{v,j,k} - (t_k - t_i - \sigma_i)\}$,

$c_{i,p,j}^{\text{wait,retrieve}}$ = Truck waiting time if a UAV were retrieved at customer i ∈`UAVCust` inserted into position p ∈{`TSPtour` positons} after serving customer j ∈`infeasCust` \i.
$\min_{v \in V,\ k \in \{\text{TSPtour positions before } p\}} \{\tau'_{v,k,j} + \sigma'_{v,j} + \tau'_{v,j,i} - (t_i - t_k - \sigma_k)\}$,

$c_{i,p,j}^{\text{wait}}$ = Minimum time the truck would spend waiting for a UAV, if the UAV is launched or retrieved at customer $i \in$ UAVCust inserted into position p ∈{`TSPtour` positions} to serve customer j ∈`infeasCust` \i.
$\min\{c_{i,p,j}^{\text{wait,launch}}, c_{i,p,j}^{\text{wait,retrieve}}\}$.

$c_{i,p,j}^{\text{fail}}$ = Cost of inserting $j \in$ infeasCust into the truck's route, if inserting customer i ∈`UAVCust` \j into position p ∈{`TSPtour` positions} cannot make the assignment of customer $j$ feasible.
$\min_{\langle i,k \rangle \in \text{TSPtour}} \tau_{i,j} + \sigma_j + \tau_{j,k} - \tau_{i,k}$.

Thus, we may calculate

$$\text{cheapestInsCost} = \min_{i \in \text{UAVCust},\ p \in \{\text{TSPtourpositions}\}} \left[ c_{i,p}^{\text{ins}} + \sum_{j \in \text{infeasCust}} (c_{i,p,j}^{\text{wait}} + c_{i,p,j}^{\text{fail}}) \right]. \tag{65}$$

The indices resulting in `cheapestInsCost` are saved as $i_{\min}$ and $p_{\min}$.

The lower bound, $\lambda$ (line 40), reflects the truck tour duration (including truck service time), launch and retrieval times for UAV customers, and `cheapestInsCost` (in the event that there are UAV customers with no feasible assignments). If the lower bound is not promising (line 42), increase the `LTL` and return to Phase I.

If no infeasible UAV assignments have been identified (line 44), the procedure continues to Phase III. Otherwise, if Phase II infeasibility occurred as a result of trying to improve upon the Phase III solution (line 47), stop the trial for improvement and return to Phase I with a new (or the same) `LTL`. In lines 50–57, insert customer $i_{min}$ into position $p_{min}$ in the truck's route and update the truck tour. If this tour is unique, repeat Phase II. Otherwise, return to Phase I with a new (or the same) `LTL` value.

**Algorithm 4.** Pseudocode for Phase II – Part 1 of 2

1: Inputs: TSPtour, TruckCust, UAVCust, Truck Arrival Times ($t_j$), prevPhase3
2:
3: UAVsorties = ∅
4: infeasCust = ∅
5: availUAVs$_j = V \quad \forall\ j \in$ TruckCust
6: `UnasgnCust = UAVCust`
7:
8: # Determine number of potential sorties for each UAV customer:
9: numOptions$_j = 0 \quad \forall\ j \in$ UAVCust
10: **for** ($\langle v, i, j, k \rangle \in P'$) **do**
11:   numOptions$_j \leftarrow$ numOptions$_j + 1$
12: **end for**
13:
14: # Create UAV sorties:
15: **while** (UnasgnCust $\neq \emptyset$) **do**
16:   $j = \text{argmin}_{j \in \text{UnasgnCust}} \text{numOptions}_j$

UnasgnCust ← UnasgnCust$\setminus\{j\}$
  WaitTime = $\infty$
  **for** ($\langle v, i, j', k\rangle \in \{P': j' = j$ and $v \in$ availUAVs$_j\}$) **do**
    w = $(\tau'_{v,i,j} + \sigma'_{vj} + \tau'_{v,j,k}) - (t_k - t_i)$
    **if** ([(WaitTime $\geqslant$ 0)and(w < WaitTime)] or (WaitTime < w < 0)) **then**
      WaitTime ← w
      $\langle v^*, i^*, j^*, k^*\rangle \leftarrow \langle v, i, j, k\rangle$
    **end if**
  **end for**
  **if** (WaitTime = =$\infty$) **then**
    infeasCust ← infeasCust $\cup \{j\}$
  **else**
    UAVsorties ← UAVsorties $\cup \{\langle v^*, i^*, j^*, k^*\rangle\}$
    availUAVs$_{i^*}$ ← availUAVs$_{i^*}\setminus\{v^*\}$
  **end if**
**end while**

**Algorithm 5.** Pseudocode for Phase II – Part 2 of 2

CheapestInsCost = 0
**If** (infeasCust $\neq \emptyset$) **then**
  Calculate CheapestInsCost per Eq. (65).
  Store $i_{\min}$ and $p_{\min}$.
**end if**

\# Calculare lower bound:
$\lambda$ = TSPcost + $\sum_{\langle v,i,j,k\rangle \in \text{UAVsorties}} (s^L_{v,i} + s^R_{v,k})$ + CheapestInsCost

**if** ($\lambda \geqslant$ OFV*)) **then**
  `increaseLTL() # See function in Algorithm 3`
**else if** (infeasCust = =$\emptyset$) **then**
  Procede to Phase III
**else**
  **if** (`prevPhase3` == **True**) **then**
    `increaseLTL() # See function below`
  **else**
    \# Modify Assignment (Insert customer with cheapest cost into truck route):
    TSPtour ← TSPtour with customer $i_{\min}$ inserted into position $p_{\min}$
    **if** (TSPtour $\notin$ prevTSPtours) **then**
      prevTSPtours ← prevTSPtours $\cup$ TSPtour
      Repeat Phase II with updated `TSPtour`
    **else**
      `increaseLTL() # See function in Algorithm 3`
    **end if**
  **end if**
**end if**

### 4.3. Phase III – timing

The role of Phase III is to determine the activity start times for the truck and UAVs (analogous to finding the various versions of the "$t$" decision variables in the mFSTSP formulation), as well as the queueing of the launch and retrieval operations (analogous to the "$z$" decision variables in the mFSTSP). Phase III makes use of the truck tour found in Phase I and the UAV sorties identified in Phase II. Pseudocode for Phase III is outlined in Algorithm 6.

**Algorithm 6.** Pseudocode for Phase III

Inputs: `TruckCust`, `UAVCust`, `TSPtour`, `UAVsorties`, $\lambda$

`P3ObjVal, ActivityTimings ← Solve formulation (P3) using an MILP solver`
`prevPhase3 =` **True**

**if** ((Formulation (P3) is infeasible) **or** ($\exists\ j \in \bar{C}$ such that $\ddot{z}_j$ = =1)) **then**
  `increaseLTL() # See function in Algorithm 3`
**else**
  **if** (`P3ObjVal` <OFV*) **then**

```
    # Save new-found incumbent solution
    ⟨OFV*, ActivityTimings*⟩ ← ⟨P3ObjVal, ActivityTimings⟩
    ⟨TSPtour*, UAVsorties*⟩ ← ⟨TSPtour, UAVsorties⟩
  end if

  # Improve by moving a customer from truck to a UAV:
  candCust = ∅        # Empty list of candidates
  for j ∈ TruckCust do
    if ((Launch and retrieval points exist) and (TSPtour\{j} ∉ prevTSPtours)) then
      candCust ← candCust ∪ {j}
    end if
  end for
  j* ← Customer with maximum potential reduction in makespan among candCust
  bestSaving ← Potential reduction in makespan if j* is moved to a UAV
  if (bestSaving > 0) then
    Return to Phase II with j* removed from TruckCust and added to UAVCust
  else
    # Local Search on UAV sorties:
    newUAVsorties = ∅
    while True do
      for k ∈ TruckCust do
        if (Truck has to wait at k for a UAV retrieval) then
          Assign the retrieval at next truck customer, if feasible assignment possible
          newUAVsorties ← Updated UAVsorties with the new retrieval assignment
        end if
      end for
      if (newUAVsorties ≠ UAVsorties) then
        Solve formulation (P3) with newUAVsorties
        if ((Formulation (P3) is infeasible) or (∃ j ∈ C̄ such that z̈_j = =1)) then
          Break while loop; increaseLTL()
        end if
      else
        Break while loop; increaseLTL()
      end if
    end while
  end if
end if
```

Phase III begins by solving an MILP, denoted as formulation (P3). The solution to (P3) provides the detailed timing of truck and UAV activities, including sequencing of launch, recovery, and service activities performed at the truck. In addition to several previously-defined decision variables, (P3) requires a new decision variable, $\ddot{z}_j \in \{0, 1\}$, defined for all $j \in \overline{C}$, where $\ddot{z}_j = 1$ if customer $j$ was assigned in Phases I and II to be served by a UAV but cannot do so. New parameters used in formulation (P3), most of which are derived from inputs `TSPtour` and `UAVsorties`, are described in Table 4. Formulation (P3) as presented below assumes that the truck is required at the depot to launch and retrieve UAVs; this assumption is relaxed in Section 4.4.

$$\left(\text{P3}\right) \quad \text{Min} \quad \hat{\dot{t}}_{c+1} + \sum_{j \in \overline{C}} \left(\lambda + (10)\delta_j\right) \ddot{z}_j \tag{66}$$

$$\text{s. t.} \quad \hat{t}'_{vi} \geqslant \dot{t}'_{vi} + s^L_{vi}(1 - \ddot{z}_{A(v,i)}) \quad \forall \ i \in N'_0, \ v \in \Delta_i, \tag{67}$$

**Table 4**
Additional parameters required by formulation (P3).

| | |
|---|---|
| $N'_0$ | Subset of nodes from which the truck can depart (excluding the customers not assigned to the truck). |
| $N'_+$ | Subset of nodes to which the truck may visit. |
| $\overline{C}$ | Subset of $C$, represents only customers not assigned to the truck. |
| $\Delta_i$ | Set of all $v \in V$ such that $v$ launches from node $i$ (i.e., where $\langle v, i, j, k\rangle \in$ UAVsorties in Phase II for some $j \in \overline{C}$ and $k \in N'_+$). |
| $\nabla_k$ | Set of all $v \in V$ such that $v$ lands at node $k$ (i.e., where $\langle v, i, j, k\rangle \in$ UAVsorties in Phase II for some $i \in N'_0$ and $j \in \overline{C}$). |
| $A(v, i)$ | The customer who is supposed to be visited by UAV $v \in \Delta_i$ after launching from node $i$. $A(v, i) = j$ where $\langle v, i, j, k\rangle \in$ UAVsorties. |
| $B(v, k)$ | The customer who is supposed to be visited by UAV $v \in \nabla_k$ before landing at node $k$. $B(v, k) = j$ where $\langle v, i, j, k\rangle \in$ UAVsorties. |
| $\lambda$ | Optimistic lower bound from Phase II. |
| $x$ | Set of all the pairs $\langle i, j\rangle$ such that the truck visits $j$ immediately after $i$. |
| $\delta_j$ | Same definition as in Section 4.2. |
| $M$ | Duration of the TSP tour including return time to depot and truck service times. |

$$\dot{\hat{t}}'_{vi} \geqslant \dot{\check{t}}_i + s^L_{vi} - M(1 - \dot{z}^L_{v,0,i}) \quad \forall \;\; i \in N'_0,\;\; v \in \triangle_i, \tag{68}$$

$$\dot{\hat{t}}'_{vi} \geqslant \dot{\bar{t}}_i + s^L_{vi} - M(1 - \dot{z}^L_{0,v,i}) \quad \forall \;\; i \in N'_0,\;\; v \in \triangle_i, \tag{69}$$

$$\dot{\hat{t}}'_{vi} \geqslant \dot{\hat{t}}'_{v_2,i} + s^L_{vi} - M(1 - \dot{z}^L_{v_2,v,i}) \quad \forall \;\; i \in N'_0,\;\; v \in \triangle_i,\;\; v_2 \in \{\triangle_i\colon v_2 \neq v\}, \tag{70}$$

$$\dot{\hat{t}}'_{v_2,i} \geqslant \dot{\check{t}}'_{vi} + s^L_{v_2,i} - M(1 - \dot{z}''_{v,v_2,i}) \\ \forall \;\; i \in \{N'_0\colon i \neq 0\},\;\; v \in \triangledown_i,\;\; v_2 \in \{\triangle_i\colon v_2 \neq v\}, \tag{71}$$

$$\dot{\check{t}}'_{vj} \geqslant \dot{\hat{t}}'_{vi} + \tau'_{vij}(1 - \ddot{z}_j) \quad \forall \;\; \langle v, i, j, k\rangle \in y, \tag{72}$$

$$\dot{\check{t}}'_{vj} \leqslant \dot{\hat{t}}'_{vi} + \tau'_{vij} + M\ddot{z}_j \quad \forall \;\; \langle v, i, j, k\rangle \in y, \tag{73}$$

$$\dot{\hat{t}}'_{vj} \geqslant \dot{\check{t}}'_{vj} + \sigma'_{vj}(1 - \ddot{z}_j) \quad \forall \;\; \langle v, i, j, k\rangle \in y, \tag{74}$$

$$\dot{\hat{t}}'_{vj} \leqslant \dot{\check{t}}'_{vj} + \sigma'_{vj} + M\ddot{z}_j \quad \forall \;\; \langle v, i, j, k\rangle \in y, \tag{75}$$

$$\dot{\check{t}}'_{vk} \geqslant \dot{\check{t}}_k + s^R_{v,k} - M\ddot{z}_{B(v,k)} \quad \forall \;\; k \in N'_+,\;\; v \in \triangledown_k, \tag{76}$$

$$\dot{\check{t}}'_{vk} \geqslant \dot{\bar{t}}_k + s^R_{v,k} - M(1 - \dot{z}^R_{0vk}) \quad \forall \;\; k \in N'_+,\;\; v \in \triangledown_k, \tag{77}$$

$$\dot{\check{t}}'_{vk} \geqslant \dot{\check{t}}'_{v_2,k} + s^R_{v,k} - M(1 - \dot{z}^R_{v_2,v,k}) \quad \forall \;\; k \in N'_+,\;\; v \in \triangledown_k,\;\; v_2 \in \{\triangledown_k\colon v_2 \neq v\}, \tag{78}$$

$$\dot{\check{t}}'_{vk} \geqslant \dot{\hat{t}}'_{v_2,k} + s^R_{v,k} - M(1 - \dot{z}'_{v_2,v,k}) \\ \forall \;\; k \in \{N'_+\colon k \neq c + 1\},\;\; v \in \triangledown_k,\;\; v_2 \in \{\triangle_k\colon v_2 \neq v\}, \tag{79}$$

$$\dot{\check{t}}'_{vk} \geqslant \dot{\hat{t}}'_{vj} + (\tau'_{vjk} + s^R_{vk})(1 - \ddot{z}_j) \quad \forall \;\; \langle v, i, j, k\rangle \in y, \tag{80}$$

$$\dot{\check{t}}'_{vk} - s^R_{vk} - \dot{\hat{t}}'_{vi} \leqslant e_{vijk} \quad \forall \;\; \langle v, i, j, k\rangle \in y, \tag{81}$$

$$\dot{\check{t}}_j \geqslant \dot{\hat{t}}_i + \tau_{ij} \quad \forall \;\; \langle i, j\rangle \in x, \tag{82}$$

$$\dot{\bar{t}}_k \geqslant \dot{\check{t}}_k + \sigma_k \quad \forall \;\; k \in \{N'_+\colon k \neq c + 1\}, \tag{83}$$

$$\dot{\bar{t}}_{c+1} \geqslant \dot{\check{t}}_{c+1}, \tag{84}$$

$$\dot{\bar{t}}_k \geqslant \dot{\check{t}}'_{vk} + \sigma_k - M(1 - \dot{z}^R_{v,0,k}) \quad \forall \;\; k \in N'_+,\;\; v \in \triangledown_k, \tag{85}$$

$$\dot{\bar{t}}_k \geqslant \dot{\hat{t}}'_{vk} + \sigma_k - M(1 - \dot{z}^L_{v,0,k}) \quad \forall \;\; k \in \{N'_+\colon k \neq c + 1\},\;\; v \in \triangle_k, \tag{86}$$

$$\dot{\hat{t}}_k \geqslant \dot{\bar{t}}_k \quad \forall \;\; k \in N'_+, \tag{87}$$

$$\dot{\hat{t}}_k \geqslant \dot{\check{t}}'_{vk} \quad \forall \;\; k \in N'_+,\;\; v \in \triangledown_k, \tag{88}$$

$$\dot{\hat{t}}_k \geqslant \dot{\hat{t}}'_{vk} \quad \forall \;\; k \in \{N'_+\colon k \neq c + 1\},\;\; v \in \triangle_k, \tag{89}$$

$$\dot{\hat{t}}_0 \geqslant \dot{\hat{t}}'_{v0} \quad \forall \;\; v \in \triangle_0, \tag{90}$$

$$\dot{z}^R_{0,v,k} + \dot{z}^R_{v,0,k} + \ddot{z}_{B(v,k)} = 1 \quad \forall \;\; k \in N'_+,\;\; v \in \triangledown_k, \tag{91}$$

$$\dot{z}^R_{v,v_2,k} + \dot{z}^R_{v_2,v,k} \leqslant 1 \quad \forall \;\; k \in N'_+,\;\; v \in \triangledown_k,\;\; v_2 \in \{\triangledown_k\colon v_2 \neq v\}, \tag{92}$$

$$\dot{z}^R_{v,v_2,k} + \dot{z}^R_{v_2,v,k} + \ddot{z}_{B(v,k)} + \ddot{z}_{B(v_2,k)} \geqslant 1 \\ \forall \;\; k \in N'_+,\;\; v \in \triangledown_k,\;\; v_2 \in \{\triangledown_k\colon v_2 \neq v\}, \tag{93}$$

$$\dot{z}^L_{0,v,i} + \dot{z}^L_{v,0,i} + \ddot{z}_{A(v,i)} = 1 \quad \forall \;\; i \in N'_0,\;\; v \in \triangle_i, \tag{94}$$

$$\dot{z}^{L}_{v,v_2,i} + \dot{z}^{L}_{v_2,v,i} \leqslant 1 \quad \forall \; i \in N'_0, \; v \in \Delta_i, \; v_2 \in \{\Delta_i : v_2 \neq v\}, \tag{95}$$

$$\dot{z}^{L}_{v,v_2,i} + \dot{z}^{L}_{v_2,v,i} + \ddot{z}_{A(v,i)} + \ddot{z}_{A(v_2,i)} \geqslant 1 \quad \forall \; i \in N'_0, \; v \in \Delta_i, \; v_2 \in \{\Delta_i : v_2 \neq v\}, \tag{96}$$

$$\dot{z}'_{v_2,v,k} + \dot{z}''_{v,v_2,k} + \ddot{z}_{B(v,k)} + \ddot{z}_{A(v_2,k)} \geqslant 1 \\ \forall \; k \in \{N'_0 : k \neq 0\}, \; v \in \nabla_k, \; v_2 \in \{\Delta_k : v_2 \neq v\}, \tag{97}$$

$$\dot{z}'_{v_2,v,k} + \dot{z}''_{v,v_2,k} \leqslant 1 \quad \forall \; k \in \{N'_0 : k \neq 0\}, \; v \in \nabla_k, \; v_2 \in \{\Delta_k : v_2 \neq v\}, \tag{98}$$

$$\dot{\hat{t}}_k \geqslant \dot{\check{t}}_k + \sum_{v \in \Delta_k} s^{L}_{vk} (1 - \ddot{z}_{A(v,k)}) + \sum_{v \in \nabla_k} s^{R}_{vk} (1 - \ddot{z}_{B(v,k)}) \quad \forall \; k \in N'_+, \tag{99}$$

$$\dot{z}^{R}_{v,0,c+1} = 0 \quad \forall \; v \in V, \tag{100}$$

$$\dot{\check{t}}_i, \; \dot{\bar{t}}_i, \; \dot{\hat{t}}_i \geqslant 0 \quad \forall \; i \in N, \tag{101}$$

$$\dot{\check{t}}'_{vi}, \; \dot{\hat{t}}'_{vi} \geqslant 0 \quad \forall \; v \in V, \; i \in N, \tag{102}$$

$$\dot{z}^{L}_{v_1,v_2,i} \in \{0, 1\} \quad \forall \; i \in N'_0, \; v_1 \in \{0 \cup \Delta_i\}, \; v_2 \in \{0 \cup \Delta_i : v_2 \neq v_1\}, \tag{103}$$

$$\dot{z}^{R}_{v_1,v_2,k} \in \{0, 1\} \quad \forall \; k \in N'_+, \; v_1 \in \{0 \cup \nabla_k\}, \; v_2 \in \{0 \cup \nabla_k : v_2 \neq v_1\}, \tag{104}$$

$$\dot{z}'_{v_1,v_2,i} \in \{0, 1\} \quad \forall \; i \in N'_+, \; v_1 \in \Delta_i, \; v_2 \in \nabla_i, \tag{105}$$

$$\dot{z}''_{v_1,v_2,i} \in \{0, 1\} \quad \forall \; i \in N'_+, \; v_1 \in \nabla_i, \; v_2 \in \Delta_i, \tag{106}$$

$$\ddot{z}_j \in \{0, 1\} \quad \forall \; j \in \overline{C}. \tag{107}$$

The objective function (66) seeks to minimize the makespan, but also includes a penalty for moving UAV customers to the truck's route. While Phase II allowed such moves without penalty, these moves are discouraged in Phase III in an effort to minimize disruptions to the previously-identified assignments. Constraints (67)–(98) are modified from the original mFSTSP formulation, and reflect the fact that some decisions have been made in the previous phases; Table 5 summarizes these relationships. Constraint (99) is an optional strengthening constraint, which provides a lower limit for the truck's departure from each visited node to be equal to its arrival time plus any applicable UAV launch and recovery times. Constraint (100) states that since the truck has no service activities at the depot node, the order of activity does not matter. Finally, Constraints (101)–(107) are decision variable definitions.

Infeasibility in Phase III may be identified in two cases. First, formulation (P3) itself may be infeasible. Second (P3) may be feasible but any of the $\ddot{z}_j$ variables turn out to be 1, meaning no feasible assignment was found for customer *j*. If Phase III is infeasible, the heuristic returns to Phase I with an updated `LTL` (lines 6–7).

If Phase III is feasible and the objective function value of (P3) is smaller than OFV*, a new incumbent has been found. The corresponding truck tour (`TSPtour`), UAV sorties (`UAVsorties`), and the decision variable values corresponding to timings and coordination (`ActivityTimings`) from Phase III are stored (lines 9–13). The procedure then continues to an improvement stage.

The first step of the improvement stage is to consider moving a customer from the truck to a UAV. A truck customer that has feasible launch and recovery points in the current truck tour is a candidate to be moved to a UAV assignment, provided that the resulting truck tour – after removing that customer – is unique (lines 16–21). The candidate truck customer with the most favorable savings metric is selected. If the best savings is positive (i.e., it would reduce the makespan), the candidate customer is removed from the truck tour and the heuristic returns to Phase II. Otherwise, if the best savings is negative, the improvement stage is terminated and a final local search procedure begins.

The last local search procedure uses the most recent solution obtained after the improvement stage, and identifies customer locations where the truck waited for a UAV retrieval. For such locations, an attempt is made to shift the retrieval point of that UAV to the next truck customer location, maintaining assignment feasibility. If no shift in any of the retrieval points is possible, the procedure is terminated, and the heuristic returns to Phase I with an updated `LTL`. If the shift is possible for at least one retrieval point, a new set of UAV sorties is obtained by making all possible shifts. The new set of UAV sorties is then provided to formulation (P3) to generate a

**Table 5**
Comparison of (P3) constraints with the base mFSTSP formulation.

| Constraint Type | (P3) Constraints | mFSTSP Constraints |
|---|---|---|
| UAV Timing | (67)–(81) | (16)–(30) |
| Truck Timing | (82)–(90) | (31)–(37) |
| UAV Recovery | (91)–(93) | (38), (41) and (42) |
| UAV Launch | (94)–(96) | (43), (46) and (47) |
| UAV Launch/Recovery Sequencing | (97) and (98) | (52) and (53) |

new solution. The local search procedure continues until (P3) becomes infeasible, at which point the heuristic returns to Phase I with an updated LTL.

### 4.4. Modifying the heuristic for variants

#### 4.4.1. Variant 1 – Truck not required at depot

As described in Section 3.7, the problem can be relaxed to allow drones to launch from, and return to, the depot independently. The only required heuristic changes are for formulation (P3) in Phase III. First, $\hat{\dot{t}}_0 = \dot{\check{t}}_0 = \dot{t}_0 = 0$, since the truck can immediately depart from the depot. Second, $\dot{z}^L_{v,0,0} = 0$ for all $v \in V$, because the truck does not have any service activity at the depot node. Also, Constraints (76), (77), and (99) should be modified such that they are no longer applicable at the depot (i.e., $k \in \{N'_+ : k \neq c + 1\}$). Moreover, if the objective is to minimize the time at which the last vehicle returns to the depot, Constraint (88) remains as is. Otherwise, if the objective is to minimize the time at which the truck returns to the depot, (88) should be such that depot node $k = c + 1$ is excluded from the constraint. Finally, Constraint (90) should be removed, since the truck is not required for launch/retrieval of UAVs at depot, and it can leave before the UAVs have been launched from the depot.

#### 4.4.2. Variant 2 – automated launch and recovery systems

The problem can also be modified to allow the driver to make deliveries while UAVs are being launched/collected (as described in Section 3.8). Again, only formulation (P3) needs to be modified to account for this variant. First, the $\dot{z}^L_{v,0,i}$, $\dot{z}^L_{0,v,i}$, $\dot{z}^R_{v,0,k}$, and $\dot{z}^R_{0,v,k}$ decision variables are no longer required. Because parcel delivery by the driver and launch/retrieval of UAVs can take place in parallel, their relative ordering is unimportant. For the same reason, Constraints (68), (69), (77), (85), (86), (91), and (94) should be removed from formulation (P3). Finally, the constraint

$$\hat{t}'_{vi} \geqslant \dot{\check{t}}_i + s^L_{vi} - M(\ddot{z}_{A(v,i)}) \quad \forall \ i \in N'_0, \ v \in \Delta_i \tag{108}$$

should be added as a replacement for Constraint (68). This ensures that, if customer $j$ has a feasible UAV assignment, then the UAV may be launched from customer $i$ only after the truck has arrived at customer $i$ and has performed the launch service activity.

## 5. Numerical analysis

This section provides a quantitative analysis to gain insights from provably optimal solutions and to assess heuristic performance. We begin with a description of the test instances, including a discussion of five potential approaches to determine UAV endurance.

Next, an analysis of optimal solutions for small-scale problems explores the behavior of the system as more UAVs are added. The analysis on small-scale problems also investigates the impacts of the size of the region and demonstrates. The benefits of an automated depot (where the truck is not required to launch or recover UAVs at the depot) and an automated delivery truck (where the driver is not required for UAV launch or recovery operations at customer locations) are explored. Also provided is an analysis on the impacts of using different endurance models.

Finally, an analysis is conducted to assess the performance of the heuristic on problems of realistic size. For small-scale problems, the heuristic is compared against optimal solutions generated via Gurobi. For larger-scale problems, the behavior of the heuristic is analyzed to demonstrate that it produces expected results and that it scales appropriately.

All computational work was conducted on a Dell desktop PC with an 8-core Intel i7-6700 processor and 16 GB RAM running Ubuntu Linux 14.04 in 64-bit mode. Where applicable, MILP models were solved via Gurobi version 7.0.1, a popular solver software package. Heuristics were coded in Python version 2.7.6. All test problems, and source code for running both the MILP model and the heuristic, are available at https://github.com/optimatorlab/mFSTSP.

### 5.1. Test problem development

As the mFSTSP is a new problem, there are no existing benchmark problems (and, therefore, no existing solution approaches against which to compare). Thus, a suite of test problems has been generated. These test problems are comprised of 100 unique customer collections, such that 20 problems were created for each of 5 levels of customer numbers (8, 10, 25, 50, and 100 customers). Within each of the 20-problem subgroups, 10 were generated on the map of Buffalo (small city), and the other 10 on the map of Seattle (large city). Of the 10 instances (for both Buffalo and Seattle), 5 have a depot located in the middle of the customer locations, while 5 have a depot at the periphery. Each problem instance was solved with 1, 2, 3 and 4 UAVs, and also with four different types of UAVs: (i) low speed and low range; (ii) high speed and low range; (iii) low speed and high range; and (iv) high speed and high range. Thus, 320 problem instances were solved for each of the 5 levels of number of customers, resulting in 1600 test instances.

Truck travel times were generated via pgRouting (pgRouting, 2017), a PostgreSQL extension, with data from OpenStreetMap (OpenStreetMap, 2017). UAV travel times were calculated assuming Euclidean distances, with speed and altitude parameters shown in Table 6. The speed parameters are comparable to the specifications for the DJI Phantom 4 UAV (DJI, 2016). The cruise altitude of 50 meters is selected in accordance with the FAA regulation that the maximum allowable altitude is 400 feet (122 meters) above ground level (Dorr, 2018).

UAV launch times ($s^L_{v,i}$) of 60 s and recovery times ($s^R_{v,k}$) of 30 s are assumed. Truck service times are assumed to be 30 s, while UAV service times are 60 s. UAV payload capacities were chosen as 2.27 kg (5 lbs). Averaged over all the test problems, 85% of the

**Table 6**
Speed Parameters for UAVs.

| Speed Class | Takeoff Speed [m/s] ([mph]) | Cruise Speed [m/s] ([mph]) | Landing Speed [m/s] ([mph]) | Cruise Altitude [meters] |
|---|---|---|---|---|
| Low Speed | 7.8 (17.5) | 15.6 (35.0) | 3.9 (8.75) | 50.0 |
| High Speed | 15.6 (35.0) | 31.3 (70.0) | 7.8 (17.50) | 50.0 |

packages were light enough to be UAV-transported. For the purpose of the analysis, the UAVs in a particular problem instance are considered to be identical (although the model does not require this). Thus, each customer can either be served by any of the UAVs or none of them.

#### 5.1.1. UAV endurance and range specification

We consider five different approaches to specifying UAV endurance, as described by the $e_{vijk}$ parameter:

1. A **non-linear model** provided by Liu et al. (2017), in which energy consumption depends both on the parcel weight and the UAV speed. This model has not been previously applied in the truck-UAV routing literature.
2. A **linear model**, similar to Dorling et al. (2017) but with different parameter values, in which energy consumption only depends on the parcel weight.
3. **Fixed time-based endurance**, where all UAVs are considered to have the same maximum flight duration, irrespective of speed, distance, or parcel weight. In this case, $e_{vijk}$ is constant for all $\langle v, i, j, k\rangle$ combinations. This is the most common endurance model in the combined truck-UAV delivery literature.
4. **Unlimited endurance**, where UAV battery energy restrictions are ignored. In this case, also pervasive in the truck-UAV literature, $e_{vijk} = \infty$ for all $\langle v, i, j, k\rangle$ combinations.
5. **Fixed distance-based endurance**, where UAV flights are limited solely as a function of travel distance. As there is no limit on the flight duration, unlimited waiting is permissible at the retrieval location provided that the destination is within the distance limit.

To estimate endurance for the non-linear and linear models (the first two cases), the UAV flight profile is partitioned into eight distinct phases. Suppose UAV $v$ launches from location $i$, delivers the parcel to customer $j$, and rejoins the truck at location $k$. This requires a takeoff time from location $i$ (denoted as $\tau'^{\mathrm{t}}_{vi}$), cruise time from location $i$ to $j$ ($\tau'^{\mathrm{c}}_{vij}$), landing time at location $j$ ($\tau'^{\mathrm{l}}_{vj}$), service time at location $j$ ($\sigma'_{vj}$), takeoff time from $j$ ($\tau'^{\mathrm{t}}_{vj}$), cruise time from $j$ to $k$ ($\tau'^{\mathrm{c}}_{vjk}$), and a landing time at location $k$ ($\tau'^{\mathrm{l}}_{vk}$). We denote this minimum travel time by:

$$T^{\min}_{vijk} = \tau'^{\mathrm{t}}_{vi} + \tau'^{\mathrm{c}}_{vij} + \tau'^{\mathrm{l}}_{vj} + \sigma'_{vj} + \tau'^{\mathrm{t}}_{vj} + \tau'^{\mathrm{c}}_{vjk} + \tau'^{\mathrm{l}}_{vk}.$$

Note that $T^{\min}_{vijk}$ does not include any time that the UAV might spend waiting for the truck to arrive at node $k$, as such waiting time is not known prior to solving the problem. This waiting time, the eighth flight phase, is addressed below.

During each phase of flight, a UAV may carry a different payload weight, may travel at a different speed, and may consume a different amount of power. In particular, during the leg from node $i$ to $j$, the UAV carries a parcel of weight $w_j$ kg, but it travels empty from node $j$ to $k$. Suppose that UAV $v$'s travel speeds for the takeoff, cruise, and landing phases are given by $v_v^t$, $v_v^c$, and $v_v^l$, respectively. The power consumption in the two energy-based models is described next.

#### 5.1.2. Endurance in the non-linear model

The equations for power consumption in the non-linear model are taken from Liu et al. (2017) – with some notational changes to distinguish among the different flight phases – are as follows:

- Power consumed during vertical takeoff or landing:

$$P^{\mathrm{tl}}\left(w, V_{\mathrm{vert}}\right) = k_1\left(W + w\right)g\left[\frac{V_{\mathrm{vert}}}{2} + \sqrt{\left(\frac{V_{\mathrm{vert}}}{2}\right)^2 + \frac{(W+w)g}{k_2^2}}\right] + c_2((W+w)g)^{3/2};$$

- Power consumed during horizontal cruise:

$$P^{\mathrm{c}}(w, V_{\mathrm{air}}) = (c_1 + c_2)[((W+w)g - c_5(V_{\mathrm{air}}\cos\alpha)^2)^2 + (c_4 V_{\mathrm{air}}^2)^2]^{3/4} + c_4 V_{\mathrm{air}}^3;$$

- Power consumed while waiting to land (i.e., hovering):

$$P^{\mathrm{h}}(w) = (c_1 + c_2)((W+w)g)^{3/2};$$

where $k_1$, $k_2$, $c_1$, $c_2$, $c_4$, and $c_5$ are model coefficients whose values were estimated by Liu et al. (2017), and are provided in Table 7. Additionally, $W$ is the UAV frame weight (considered as 1.5 kg), $w$ is the parcel weight, $g$ is the gravitational constant (9.8 m/s$^2$), $\alpha$

**Table 7**
Coefficient values for the non-linear endurance model as in Liu et al. (2017).

| Coefficient | Value |
|---|---|
| $k_1$ | 0.8554 [Unitless] |
| $k_2$ | 0.3051 $\sqrt{\text{kg/m}}$ |
| $c_1$ | 2.8037 $\sqrt{\text{m/kg}}$ |
| $c_2$ | 0.3177 $\sqrt{\text{m/kg}}$ |
| $c_4$ | 0.0296 kg/m |
| $c_5$ | 0.0279 Ns/m |

is the forward tilt of the UAV while flying horizontally (e.g., the "angle of attack", assumed to be 10 degrees), $V_{\text{vert}}$ is the vertical UAV speed (during ascent or descent), and $V_{\text{air}}$ is the horizontal UAV speed (during cruise). It is assumed that power consumption during the service is negligible. Thus, the minimum energy required for UAV $v$ to complete the visit from $i$ to $j$ to $k$ is expressed as:

$$E_{vijk}^{\min} = \tau_{vi}'^{\text{t}} P^{\text{tl}}(w_j, \nu_v^t) + \tau_{vij}'^{\text{c}} P^{\text{c}}(w_j, \nu_v^c) + \tau_{vj}'^{\text{l}} P^{\text{tl}}(w_j, \nu_v^l) + \tau_{vj}'^{\text{t}} P^{\text{tl}}(0, \nu_v^t) + \tau_{vjk}'^{\text{c}} P^{\text{c}}(0, \nu_v^c) + \tau_{vk}'^{\text{l}} P^{\text{tl}}(0, \nu_v^l).$$

We denote the energy of the battery loaded on UAV $v$ by $E_v^{\text{avail}}$. Thus, the UAV requires $E_{vijk}^{\min} \leqslant E_v^{\text{avail}}$ to safely fly from $i$ to $j$ to $k$. If this requirement is satisfied, the endurance $e_{vijk}$ (in seconds) is given by

$$e_{vijk} = T_{vijk}^{\min} + \frac{E_v^{\text{avail}} - E_{vijk}^{\min}}{P^{\text{h}}(0)},$$

where the second term in the expression captures the maximum length of time that the UAV may hover above node $k$ while waiting for the truck to arrive.

#### 5.1.3. Endurance in the linear model

The energy consumption in the linear model is a function of parcel weight (but not velocity), similar to the model by Dorling et al. (2017). Power consumed during different flight phases (takeoff, cruise, landing, or hovering) is considered the same, and is given by

$$P^{\text{tclh}}(w) = \beta w + \gamma,$$

where $\beta$ is the power consumed per unit parcel weight, $\gamma$ is the power consumed to keep an empty UAV moving, and $w$ is the parcel weight. Note that the functional form used here is the same as in the model by Dorling et al. (2017), the values of $\beta$ and $\gamma$ are different. The method of their value estimation is explained in Section 5.3. The minimum energy required for UAV $v$ to complete the visit from $i$ to $j$ (carrying a parcel of weight $w_j$) and from $j$ to $k$ (without a parcel, which was delivered to $j$) is:

$$E_{vijk}^{\min} = (\tau_{vi}'^{\text{t}} + \tau_{vij}'^{\text{c}} + \tau_{vj}'^{\text{l}}) P^{\text{tclh}}(w_j) + (\tau_{vj}'^{\text{t}} + \tau_{vjk}'^{\text{c}} + \tau_{vk}'^{\text{l}}) P^{\text{tclh}}(0).$$

If this requirement is less than the energy available ($E_v^{\text{avail}}$) then endurance $e_{vijk}$ is given by

$$e_{vijk} = T_{vijk}^{\min} + \frac{E_v^{\text{avail}} - E_{vijk}^{\min}}{P^{\text{tclh}}(0)}.$$

#### 5.1.4. Range specification

In the numerical analysis, a low-range UAV is defined to have sufficient battery power to carry a 2.27 kg (5 lb) parcel up to 3 miles, with an empty return to the launch location (i.e., 6 miles round-trip). Note that a UAV carrying less than its full capacity (2.27 kg) would have a range longer than 3 miles. For a high-range UAV, the battery is sufficient to carry a 2.27 kg parcel up to 6 miles, with an empty return. Table 8 provides the battery capacities for low- and high-range UAVs.

### 5.2. Analysis of optimal solutions

We first examine properties of optimal mFSTSP solutions, as generated by the MILP model and solved via Gurobi. Due to the

**Table 8**
Battery capacity, represented by $E_v^{\text{avail}}$.

| | $E_v^{\text{avail}}$ | |
|---|---|---|
| **UAV Range** | **Low Speed [kJ]** | **High Speed [kJ]** |
| Low | 291.1 | 457.5 |
| High | 563.0 | 904.0 |

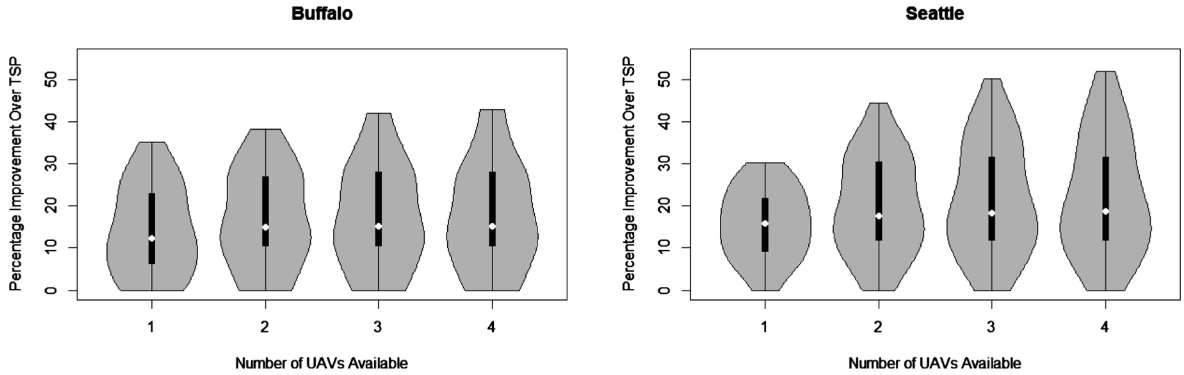



**Fig. 5.** Impact of increasing the number of UAVs for eight-customer problems in a small (Buffalo) and a large (Seattle) city.

computational complexity of the problem, provably optimal solutions may only be obtained for small-scale problems. Therefore, the analysis has been conducted for 8-customer problems only.

A cutoff time of 1-h was given to Gurobi for each of the 320 8-customer problems; 212 of which (about 66%) solved to optimality within the time limit. Hence, the optimal solution analysis below is performed on these 212 problems.

Fig. 5 provides a summary of the percentage improvement in makespan if UAVs are used in synchronization with the truck, as compared to the truck-only case. Increasing the number of UAVs does serve to reduce the makespan. However, due to the limited number of customers, the impact of additional UAVs is minimal. By contrast, more significant improvement is observed in the large-scale problems discussed in Section 5.4. To help put the percentage improvements in perspective, Table 9 provides average delivery times.

As shown in Fig. 6, the system rarely serves all "droneable" customers via UAVs. This is a function of the proximity among customers, the service times for preparing and retrieving the UAVs, and the flight endurance.

An analysis was also conducted to study the impacts of an automated depot and an automated delivery truck. Table 10 summarizes five cases involving combinations of these automation technologies, with two different metrics. After modifying the constraints corresponding to these different variants (as described in Sections 3.7 and 3.8), the same set of 212 problems (for which we obtained optimal solutions) were run using the IP model to study the improvements in makespan over a baseline system where (1) the truck is not automated for launch/recovery by itself, and (2) the truck is required at the depot for launch/recovery.

The results of this analysis, summarized in Fig. 7, indicate a 4.4% average savings of the makespan if the truck is automated to launch and recover UAVs by itself (Case 1). The savings increases to an average of 7.0% if, instead, the truck is not required to launch and recover UAVs at the depot (Case 2a). However, if the objective is to minimize the time at which the truck returns to depot (i.e., the time at which the UAVs return is ignored), the savings further improves to 8.9% (Case 2b). When both Cases 1 and 2a are combined – so the truck is automated and is also not required at the depot – the average improvement over the default case becomes 10.2% (Case 3a). Finally, Case 3b provides the maximum average improvement of 12.8%. This analysis indicates that automation at the depot may be more beneficial than automating the truck. To put these savings into perspective, the average makespan for the default setting is 49 min. The average makespans (in min:sec) for the other cases are 47:32 (Case 1), 46:07 (Case 2a), 45:20 (Case 2b), 44:58 (Case 3a) and 43:58 (Case 3b). It should be noted, however, that the conclusions here are drawn from only small 8-customer instances in which the time required by the truck for launch/delivery at the depot occupies a relatively large fraction of the total makespan. System behavior for larger problems is unknown due to the lack of provably optimal solutions.

### 5.3. Comparison of endurance models

We next examine the impacts of using the five different endurance models introduced in Section 5.1.1. The parameters for the

**Table 9**
Comparison of average objective function values between the TSP and mFSTSP with differing numbers of UAVs, for 8-customer problem instances.

| | | Avg. Objective Function Value | | | |
|---|---|---|---|---|---|
| | | Buffalo | | Seattle | |
| | | [hr:min:sec] | % Savings | [hr:min:sec] | % Savings |
| **TSP** | | 0:23:49 | – | 1:16:19 | – |
| **mFSTSP UAVs** | 1 | 0:20:22 | 14.30 | 1:04:24 | 15.75 |
| | 2 | 0:19:44 | 16.99 | 1:01:10 | 20.14 |
| | 3 | 0:19:36 | 17.51 | 1:00:02 | 21.61 |
| | 4 | 0:19:33 | 17.75 | 0:59:37 | 22.19 |

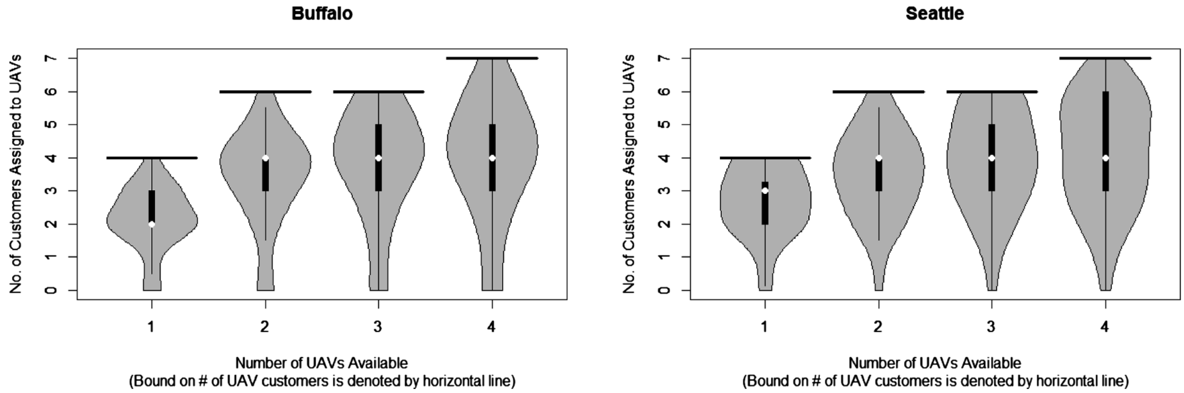



**Fig. 6.** A comparison of the number of customers assigned to UAVs.

**Table 10**
A summary of the five cases studied for future automation enhancements.

| | Metric | Condition | |
|---|---|---|---|
| | Time at which the ... | *Driver* not req'd at *customer locations* for launch/recovery | *Truck* not req'd at *depot* for launch/recovery |
| Case 1 | last vehicle returns to depot. | ✓ | |
| Case 2a | last vehicle returns to depot. | | ✓ |
| Case 2b | truck returns to depot. | | ✓ |
| Case 3a | last vehicle returns to depot. | ✓ | ✓ |
| Case 3b | truck returns to depot. | ✓ | ✓ |

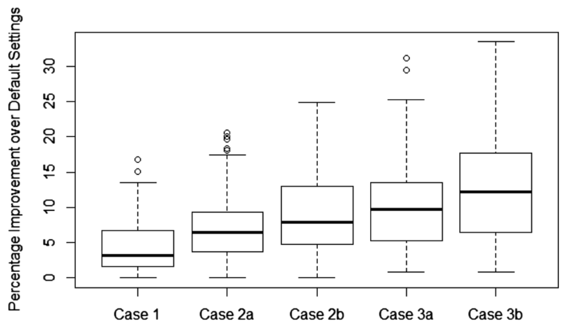


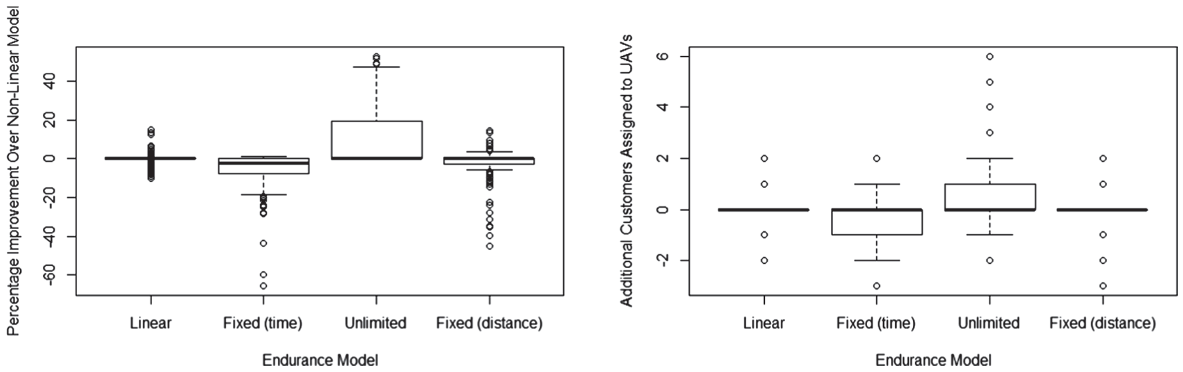

**Fig. 7.** Comparison of percentage improvement for five different cases over a default system in which the truck is required to launch and retrieve UAVs at the depot, and the driver is required to participate in the launch and retrieval process at each customer location.

non-linear model are specified in Table 7. The parameter values for the other four models are chosen here such that the resulting endurance values (in seconds) are comparable to the non-linear model in an 'average' sense. For example, battery capacity for the linear model is considered the same as in Table 8. Additionally, the coefficients for this model are estimated such that it linearly approximates the power consumed by the non-linear model. A complete specification of the other four models is provided in Table 11.

The same set of 212 problems, for which optimal solutions were obtained, were run using the different endurance models. Results

**Table 11**
Specifications for endurance models

| UAV Type Speed/Range | Linear $\beta$ [W/kg] | $\gamma$ [W] | Fixed (time) [s] | Unlimited [s] | Fixed (distance) [miles] |
|---|---|---|---|---|---|
| Low/Low | 210.8 | 181.2 | 700 | ∞ | 6 |
| Low/High | 210.8 | 181.2 | 1400 | ∞ | 12 |
| High/Low | 24.2 | 1392.0 | 350 | ∞ | 6 |
| High/High | 24.2 | 1392.0 | 700 | ∞ | 12 |




(a) Percentage makespan improvement versus non-linear endurance model (b) Additional customers assigned to UAVs versus non-linear endurance model

**Fig. 8.** Impact of using different endurance models for eight-customer problems.

are compared against the baseline case of non-linear endurance, as presented in Fig. 8. As expected, different endurance models produce different objective function values (makespans) for the same problem instance, since the feasible set of tuples $P$ is affected. Specifically, the percentage of problems with objective function values that differ from the default case is: 41% for linear; 62% for fixed-time endurance, 52% for unlimited endurance, and 50% for the fixed-distance endurance models.

The largest improvement in the objective function value over the non-linear model is shown by the unlimited endurance model, as in Fig. 8a. This is intuitive, since all $\langle v, i, j, k\rangle$ combinations are feasible if endurance is ignored. This is followed by fixed endurance models (both time and distance), where both positive and negative improvements in objective function values are observed. Note that the fixed-distance model tends to overestimate the endurance when compared to the fixed-time model, as the fixed-distance model ignores the endurance impacts when drones wait for the truck (waiting does not involve any additional travel distance). Thus, the fixed-time model is the more restrictive of the two. The least difference between the objectives corresponds to the linear model, since it is the closest approximation to the baseline non-linear endurance model. Fig. 8b provides the distribution of additional customers assigned to UAVs, as compared to the non-linear model. The greater improvement in makespan for the unlimited endurance model is attributed to the fact that more customers were assigned to UAVs in that case. Conversely, the poor performance of the fixed endurance (time) model in terms of objective function value is due to fewer customers being assigned to UAVs. This shows that using a fixed endurance model, instead of an energy consumption-based model, may result in under-utilization of resources.

Finally, we consider the risks of using an “incorrect” endurance model. For this analysis, non-linear endurance is assumed to be the true energy consumption model. Solutions obtained from the other endurance models were input to the MILP model with non-linear endurance to determine if feasibility was maintained. This test was conducted on the same set of 212 problems for which optimal solutions were obtained. The percentage of problems for which the MILP model with non-linear endurance was feasible was: 84% for the linear model, 94% for the fixed-time endurance, 42% for unlimited endurance, and 58% for fixed-distance endurance. Thus, if the solutions were generated by assuming fixed-distance endurance, then 42% of problems would be infeasible if the UAV energy consumption was actually non-linear with speed and payload weight. Similarly, while only 6% of the solutions determined by the fixed-time endurance model were infeasible when assessed with the non-linear model, 58% of the unlimited endurance model were infeasible. In such cases, there is a danger of UAVs running out of battery power while en route, if the routes were generated via an incorrect energy consumption model.

### 5.4. Heuristic performance

The effectiveness of the heuristic was assessed on 320 instances for each of the 8-, 10-, 25-, 50-, and 100-customer problems. A comparison of heuristic solutions to optimal solutions is only provided for 8-customer problems, due to the inability of Gurobi to generate optimal solutions (or, frequently, even feasible solutions) for larger-scale problems in a reasonable time. In the mFSTSP heuristic, a 5-s cutoff time was imposed when solving formulation (P3) in Phase III.

Table 12 provides a comparison of runtimes and gaps between Gurobi and the proposed heuristic. Recall that Gurobi was given a 1-h (3600-s) cutoff time, such that 108 of the 320 8-customer test instances were unable to solve to optimality within this time limit. The average runtime for Gurobi to solve the 8-customer problems is 1428.3 s (about 24 min), while it is 0.10 s for the heuristic. One interesting observation is that the Gurobi runtimes for Seattle problems were nearly 78% less than for the Buffalo problems. One possible explanation is that the presence of multiple waterways in Seattle reduced the number of similarly-attractive truck tours. To put the optimality gaps into perspective, the 5% average gap associated with the heuristic indicates that this solution approach produces solutions that result in only 147 s of additional delivery time beyond what was found by the complete MILP formulation solved via Gurobi. Furthermore, the relatively long launch and recovery times relative to the makespan provide little margin for error for these small-scale problems (e.g., the 60-s launch time is approximately 2% of the average optimal makespan of 2943 s).

To assess the solution quality of larger-scale problems, for which the MILP formulation becomes computationally intractable, a comparison with TSP solutions is provided. Table 13 summarizes the runtimes and percentage improvement over TSP solutions for

**Table 12**
Summary statistics for 8-customer problem instances.

| | | Avg. Runtime [s] | | % Gap to MILP 212 Optimal Solns | | | % Gap to MILP All 320 Solns | | |
|---|---|---|---|---|---|---|---|---|---|
| | | MILP | Heuristic | Avg. | Min | Max | Avg. | Min | Max |
| **UAVs** | 1 | 51.4 | 0.05 | 5.0 | 0.0 | 21.1 | 5.0 | 0.0 | 21.1 |
| | 2 | 1075.5 | 0.10 | 4.9 | 0.0 | 44.2 | 4.6 | 0.0 | 44.2 |
| | 3 | 2155.6 | 0.11 | 4.7 | 0.0 | 32.7 | 3.5 | 0.0 | 32.7 |
| | 4 | 2430.5 | 0.14 | 5.4 | 0.0 | 32.3 | 2.8 | 0.0 | 32.3 |
| **City** | Buffalo | 2349.1 | 0.11 | 5.1 | 0.0 | 18.9 | 3.3 | 0.0 | 18.9 |
| | Seattle | 507.4 | 0.08 | 4.9 | 0.0 | 44.2 | 4.7 | 0.0 | 44.2 |
| | **Overall** | **1428.3** | **0.10** | **5.0** | **0.0** | **44.2** | **4.0** | **0.0** | **44.2** |

**Table 13**
Heuristic solution statistics.

| Customers | UAVs | Avg. Runtime [s] | Avg. TSP Obj. [hr:min:sec] | Avg. mFSTSP Obj. [hr:min:sec] | Avg. Improv. over TSP [%] |
|---|---|---|---|---|---|
| 10 | 1 | 0.1 | 0:55:48 | 0:49:09 | 10.2 |
| | 2 | 0.1 | 0:55:48 | 0:46:56 | 13.7 |
| | 3 | 0.1 | 0:55:48 | 0:45:54 | 15.3 |
| | 4 | 0.2 | 0:55:48 | 0:45:46 | 15.7 |
| 25 | 1 | 1.0 | 2:34:32 | 2:08:06 | 16.9 |
| | 2 | 2.0 | 2:34:32 | 1:58:47 | 22.7 |
| | 3 | 3.4 | 2:34:32 | 1:54:56 | 25.3 |
| | 4 | 3.7 | 2:34:32 | 1:53:11 | 26.6 |
| 50 | 1 | 10.7 | 3:40:03 | 3:09:20 | 13.9 |
| | 2 | 26.7 | 3:40:03 | 2:53:16 | 21.3 |
| | 3 | 55.5 | 3:40:03 | 2:44:55 | 25.2 |
| | 4 | 65.4 | 3:40:03 | 2:41:25 | 26.8 |
| 100 | 1 | 175.5 | 5:31:49 | 4:51:43 | 11.7 |
| | 2 | 434.3 | 5:31:49 | 4:28:45 | 18.6 |
| | 3 | 709.5 | 5:31:49 | 4:14:36 | 22.8 |
| | 4 | 1009.7 | 5:31:49 | 4:08:31 | 24.6 |

these problems. As expected, the runtime increases with problem size, with most problems up to 50 customers solving in a minute or less. While the 100-customer problems have longer runtimes, the largest problems are still solved in approximately 17 min.

Also of note, the average improvements over the TSP for 10-, 25-, 50-, and 100-customer problems are 14%, 23%, 22% and 19% respectively. We suggest two possible explanations for this non-increasing behavior. First, it is possible that the heuristic is not providing near-optimal solutions for the 50- and 100-customer problems. Unfortunately, there is no existing method for assessing an optimality gap for the mFSTSP. However, there is evidence that the relatively time-expensive launch time (60 s), recovery time (30 s), and customer service time (30 s longer for a drone delivery than a driver delivery) make it less beneficial to deploy the UAVs when the customer density increases (recall that the service region remains unchanged as more customers are added).

These factors also contribute to the diminishing rates of improvement as the number of UAVs is increased, as shown in Fig. 9. However, the results do demonstrate that additional UAVs lead to a reduction in makespan, indicating that the heuristic's behavior is consistent with expectations.

### 5.5. Analysis of queue scheduling conflicts

We conclude the numerical study with an analysis of queue scheduling conflicts exhibited in the test problems. Table 14 summarizes the frequency of conflicts for a variety of problem sizes (based on a total of 960 test instances). The last column describes the average number of truck customers where two or more UAV activities occurred (e.g., multiple launches, multiple retrievals, or a mix of launches and retrievals). Even in cases with just two UAVs, conflict at truck customer locations occurs between 28% and 50% of the time. As expected, conflict occurrences generally tend to increase as more UAVs are available.

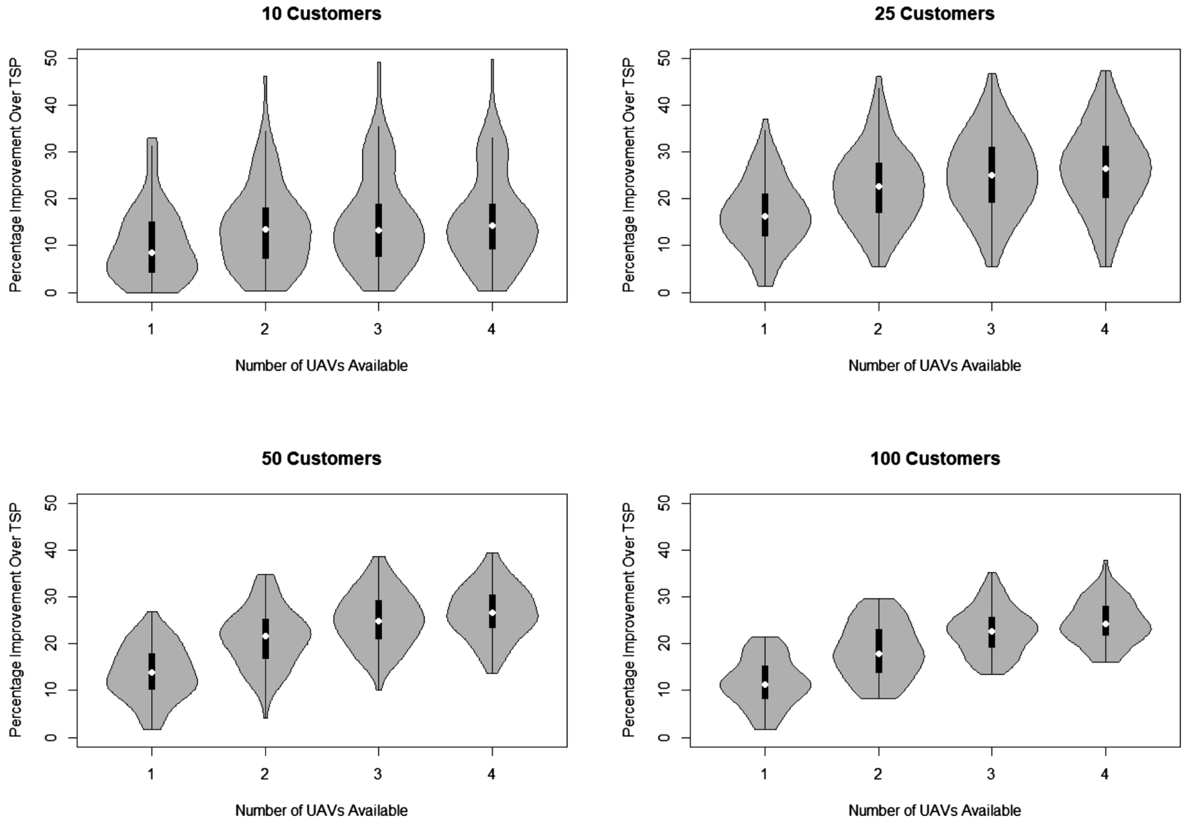



**Fig. 9.** Impacts of additional UAVs for larger-scale problems.

**Table 14**
UAV launch/retrieval conflicts.

| Customers | Number of UAVs | Average number of truck customers | Average number of truck customers where conflict occurs |
|---|---|---|---|
| 10 | 2 | 6.79 | 1.90 **(27.98%)** |
| | 3 | 6.49 | 2.10 **(32.36%)** |
| | 4 | 6.39 | 1.89 **(29.58%)** |
| 25 | 2 | 13.34 | 6.66 **(49.93%)** |
| | 3 | 11.60 | 6.53 **(56.29%)** |
| | 4 | 10.69 | 6.22 **(58.19%)** |
| 50 | 2 | 27.09 | 12.94 **(47.77%)** |
| | 3 | 22.38 | 12.85 **(57.42%)** |
| | 4 | 20.24 | 12.71 **(62.80%)** |
| 100 | 2 | 56.15 | 23.30 **(41.50%)** |
| | 3 | 45.20 | 24.36 **(53.89%)** |
| | 4 | 40.54 | 23.10 **(56.98%)** |

## 6. Conclusions and future research

This paper introduced the mFSTSP, whereby customer parcels may be delivered via multiple heterogeneous UAVs and a single delivery truck. These UAVs may be deployed from, and recovered by, the truck to extend the drones' effective range. The flight endurance for each drone is modeled as a function of the drone's battery size, payload, travel distance, and flight phases (e.g., takeoff, cruising, and landing). To reflect the realities of deploying multiple aircraft from a single truck, detailed queue scheduling for UAV arrivals and departures is incorporated within the proposed MILP formulation. A three-phased heuristic solution approach provides high-quality solutions with reasonable runtimes for problems of realistic size.

From the analyses conducted in this paper, we obtain some critical insights into the truck-UAV system design. Although it is generally faster to reach a customer with a UAV rather than truck, it is rarely beneficial to serve all UAV-eligible customers via drone. Adding more UAVs to an existing fleet tends to have diminishing marginal makespan improvements, with UAVs offering a greater benefit in instances involving a larger number of customers. Furthermore, we observed that problem instances involving densely-distributed customers tend to benefit the most. The analyses reveal that automation within both the truck and the depot result in time savings, with depot automation providing the greatest savings. Finally, the analysis also demonstrated the influence of a chosen endurance model on UAV assignments. Of particular importance is the danger of operational failure if a schedule is constructed using constant or unlimited endurance. Such endurance models do not account for speed and parcel weight in UAV energy consumption, and may lead to solutions that are not implementable in practice.

There are myriad opportunities for future research in this area. For example, the mFSTSP can be extended to allow UAVs to re-launched multiple times from the same location. The UAV queue scheduling presented in this paper could also be applied to problems in which large numbers of UAVs are deployed from a depot (as in the PDSTSP). Furthermore, this study considers a last-mile delivery service whereby all parcels are available at the depot at the beginning of the delivery period; no additional orders arrive after the vehicle assignments have been made. However, UAVs also promise benefits to companies that seek to fulfill incoming orders directly from distribution centers (an obvious example of which is Amazon). In such a case, orders for on-hand products arrive throughout the day, resulting in a dynamic routing problem.

Of course there also remain opportunities to develop more efficient heuristic approaches, and to continue adding realism to the models by eliminating unrealistic simplifying assumptions. New theoretical bounds that capture such features as heterogeneous UAVs and queue scheduling would be valuable in assessing heuristic performance for large-scale problems. Further studies that provide managerial insights would also be of great benefit. For example, although the mFSTSP model presented here considers UAVs of differing capacities, there was no analysis regarding the impact of these capacities. Additional studies may consider the impacts of customer density, and could further analyze the economic impact of these novel delivery systems.

## Acknowledgements

The authors are grateful to Editor-in-Chief Yafeng Yin and the two anonymous reviewers whose helpful suggestions have improved the quality of this research. The authors also wish to acknowledge Amanda Chu and Eddie Rivera for their contributions to the preliminary modeling efforts.

## Appendix A. Supplementary material

Supplementary data associated with this article can be found, in the online version, at https://doi.org/10.1016/j.trc.2019.11.003.

## References


E. Adams.DHL's tilt-rotor 'Parcelcopter' is both awesome and actually useful, 2016. URL: <https://www.wired.com/2016/05/dhls-new-drone-can-ship-packages-around-alps/>.

Adams, E., 2017. UPS tests drone delivery, launching UAV from top of truck, 2017. URL <http://www.techrepublic.com/article/ups-tests-drone-delivery-launching-uav-from-top-of-truck/>.

Agatz, N., Bouman, P., Schmidt, M., 2018. Optimization approaches for the traveling salesman problem with drone. Transport. Sci.

Amazon, 2019. <https://www.amazon.com/Amazon-Prime-Air/b?node=8037720011>.

Bin Othman, M.S., Shurbevski, A., Karuno, Y., Nagamochi, H., 2017. Routing of carrier-vehicle systems with dedicated last-stretch delivery vehicle and fixed carrier route. J. Inform. Process. 25, 655–666.

Bouman, P., Agatz, N., Schmidt, M., 2018. Dynamic programming approaches for the traveling salesman problem with drone. Networks 72 (4), 528–542.

Boyle, A., 2016. Flying saucer? No, it's the first drone pizza delivery, from Domino's and Flirtey, 2016. URL <https://www.geekwire.com/2016/drone-delivery-pizza-dominos-flirtey/>.

Boysen, N., Briskorn, D., Fedtke, S., Schwerdfeger, S., 2018. Drone delivery from trucks: drone scheduling for given truck routes. Networks 72 (4), 506–527.

Campbell, J.F., Sweeney II D.C., Zhang, J., 2017. Strategic design for delivery with trucks and drones. URL <https://www.researchgate.net/profile/James_Campbell3/publication/316169890_Strategic_Design_for_Delivery_with_Trucks_and_Drones/links/58f4efb7a6fdcc11e569fb5d/Strategic-Design-for-Delivery-with-Trucks-and-Drones.pdf>.

Carey, N., 2016. UPS launches U.S. drone test flights for urgent medical deliveries, 2016. URL http://www.reuters.com/article/united-parcel-drones-delivery-idUSL2N1BY1UY.

Carlsson, J.G., Song, S., 2017. Coordinated logistics with a truck and a drone. Manage. Sci.

Chang, Y.S., Lee, H.J., 2018. Optimal delivery routing with wider drone-delivery areas along a shorter truck-route. Exp. Syst. Appl. 104, 307–317.

Chauhan, D., Unnikrishnan, A., Figliozzi, M., 2019. Maximum coverage capacitated facility location problem with range constrained drones. Transport. Res. Part C: Emerg. Technol. 99, 1–18.

Cheng, C., Adulyasak, Y., Rousseau, L.M., 2018. Formulations and exact algorithms for drone routing problem. Technical report, 2018. URL <https://www.cirrelt.ca/DocumentsTravail/CIRRELT-2018-31.pdf>.

Chowdhury, S., 2018. Drone routing and optimization for post-disaster inspection. PhD thesis. Mississippi State University.

Daknama, R., Kraus, E., 2017. Vehicle routing with drones. arXiv preprint arXiv:1705.06431, 2017. https://arxiv.org/pdf/1705.06431.pdf.

Dell'Amico, M., Montemanni, R., Novellani, S., 2019a Drone-assisted deliveries: New formulations for the flying sidekick traveling salesman problem. arXiv preprint arXiv:1905.13463, 2019a.

Dell'Amico, M., Montemanni, R., Novellani, S., 2019b. Matheuristic algorithms for the parallel drone scheduling traveling salesman problem. arXiv preprint arXiv:1906.02962, 2019b.

DJI, 2016. <https://www.dji.com/phantom-4/info>.

Dorling, K., Heinrichs, J., Messier, G.G., Magierowski, S., 2017. Vehicle routing problems for drone delivery. IEEE Trans. Syst., Man, Cybernet.: Syst. 47 (1), 70–85.

Dorr, L., 2018. Federal Aviation Administration, 2018. URL <https://www.faa.gov/news/fact_sheets/news_story.cfm?newsId=22615>.

Dukkanci, O., Kara, B.Y., Bektas, T., 2019. The drone delivery problem. Available at SSRN, January 2019. <https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3314556>.

Etherington, D., 2017. Check out the real world version of the Mercedes-Benz drone delivery van, 2017. <https://techcrunch.com/2017/01/10/check-out-the-real-

world-version-of-the-mercedes-benz-drone-delivery-van/>.
Ferrandez, S.M., Harbison, T., Weber, T., Sturges, R.R., Rich, R., 2016. Optimization of a truck-drone in tandem delivery network using k-means and genetic algorithm. J. Industr. Eng. Manage./ 9 (2), 374.
Freitas, J.C., Penna, P.H.V., 2018. A variable neighborhood search for flying sidekick traveling salesman problem. arXiv preprint arXiv:1804.03954, 2018. https://arxiv.org/pdf/1804.03954.pdf.
Gurobi Optimization. tsp.py, 2018. <https://www.gurobi.com/documentation/8.1/examples/tsp_py.html>.
Ha, Q.M., Deville, Y., Pham, Q.D., Hà, M.H., 2018. On the min-cost traveling salesman problem with drone. Transport. Res. Part C: Emerg. Technol. 86, 597–621.
Ham, A.M., 2018. Integrated scheduling of m-truck, m-drone, and m-depot constrained by time-window, drop-pickup, and m-visit using constraint programming. Transport. Res. Part C: Emerg. Technol. 91, 1–14.
Hong, I., Kuby, M., Murray, A.T., 2018. A range-restricted recharging station coverage model for drone delivery service planning. Transport. Res. Part C: Emerg. Technol. 90, 198–212.
Jeong, H.Y., Lee, S., Song, B.D., 2019. Truck-drone hybrid delivery routing: payload-energy dependency and no-fly zones. Int. J. Prod. Econ.
Kim, K., Awwad, M.A., 2017. Modeling effective deployment of airborne fulfilment centers. In: Proceedings of the American Society for Engineering Management 2017 International Annual Conference.
Kim, S., Moon, I., 2019. Traveling salesman problem with a drone station. IEEE Trans. Syst., Man, Cybernet.: Syst. 49 (1), 42–52.
Kim, S.J., Lim, G.J., Cho, J., Côté, M.J., 2017. Drone-aided healthcare services for patients with chronic diseases in rural areas. J. Intell. Robot. Syst. 88 (1), 163–180.
Kitjacharoenchai, P., Ventresca, M., Moshref-Javadi, M., Lee, S., Tanchoco, J.M.A., Brunese, P.A., 2019. Multiple traveling salesman problem with drones: mathematical model and heuristic approach. Comput. Industr. Eng.
Li, Y., Zhang, G., Pang, Z., Li, L. Continuum approximation models for joint delivery systems using trucks and drones. Enterprise Inform. Syst., pp 1–30, 2018.
Liu, Z., Sengupta, R., Kurzhanskiy, A., 2017. A power consumption model for multi-rotor small unmanned aircraft systems. In: 2017 International Conference on Unmanned Aircraft Systems (ICUAS). IEEE, pp. 310–315.
Luo, Z., Liu, Z., Shi, J., 2017. A two-echelon cooperated routing problem for a ground vehicle and its carried unmanned aerial vehicle. Sensors 17 (5), 1144.
Mathew, N., Smith, S.L., Waslander, S.L., 2015. Planning paths for package delivery in heterogeneous multirobot teams. IEEE Trans. Autom. Sci. Eng. 12 (4), 1298–1308.
MHI, 2016. The 2016 MHI Annual Industry Report – Accelerating change: How innovation is driving digital, always-on supply chains. Technical report, Material Handling Institute, 2016.
Mogg, T., 2015. Watch Google's new Project Wing drone deliver a package, 2015. URL <https://www.digitaltrends.com/cool-tech/watch-googles-new-project-wing-drone-deliver-a-package/>.
Mogg, T., 2017. Amazon patents shipping label with built-in parachute for drone deliveries, 2017. URL <http://money.cnn.com/2017/05/31/technology/amazon-parachute-shipping-label/index.html>.
Murray, C.C., Chu, A.G., 2015. The flying sidekick traveling salesman problem: optimization of drone-assisted parcel delivery. Transport. Res. Part C: Emerg. Technol. 54, 86–109.
Oh, G., Kim, Y., Ahn, J., Choi, H.L., 2018. Task allocation of multiple UAVs for cooperative parcel delivery. In: Advances in Aerospace Guidance, Navigation and Control. Springer, pp. 443–454.
OpenStreetMap, 2017. <http://openstreetmap.org>.
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.
Peng, K., Du, J., Lu, F., Sun, Q., Dong, Y., Zhou, P., Hu, M., 2019. A hybrid genetic algorithm on routing and scheduling for vehicle-assisted multi-drone parcel delivery. IEEE Access 7, 49191–49200.
Peterson, K., Dektas, M., 2017. UPS tests residential delivery via drone launched from atop package car, 2017. URL <https://www.pressroom.ups.com/pressroom/ContentDetailsViewer.page?ConceptType=PressReleases&id=1487687844847-162>.
pgRouting, 2017. <http://pgrouting.org>.
Poikonen, S., 2018. Multi-visit drone routing problem. Working Paper, 2018. URL <http://www.stefan-poikonen.net/MVDRP_September_2018.pdf>.
Poikonen, S., Wang, X., Golden, B., 2017. The vehicle routing problem with drones: extended models and connections. Networks.
Poikonen, S., Golden, B., Wasil, E.A., 2019. A branch-and-bound approach to the traveling salesman problem with a drone. INFORMS J. Comput. 31 (2), 335–346.
Ponza, A., 2016. Optimization of drone-assisted parcel delivery. Master's thesis, Università Degli Studi Di Padova, 2016.
Rabta, B., Wankmüller, C., Reiner, G., 2018. A drone fleet model for last-mile distribution in disaster relief operations. Int. J. Disaster Risk Reduct. 28, 107–112.
Rubin, B.F., 2017. Amazon Prime Air drone completes its first U.S. public delivery, 2017. URL <https://www.cnet.com/news/amazon-prime-air-first-drone-delivery-us-public-sunscreen-mars/>.
Sacramento, D., Pisinger, D., Ropke, S., 2019. An adaptive large neighborhood search metaheuristic for the vehicle routing problem with drones. Transport. Res. Part C: Emerg. Technol. 102, 289–315.
Mbiadou Saleu, R.G., Deroussi, L., Feillet, D., Grangeon, N., Quilliot, A., 2018. An iterative two-step heuristic for the parallel drone scheduling traveling salesman problem. Networks 72 (4), 459–474.
San, K.T., Lee, E.Y., Chang, Y.S., 2016. The delivery assignment solution for swarms of UAVs dealing with multi-dimensional chromosome representation of genetic algorithm. In: Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), IEEE Annual, IEEE, 2016, pp. 1–7.
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, pp. 352–361.
Schermer, D., Moeini, M., Wendt, O., 2019. A hybrid VNS/tabu search algorithm for solving the vehicle routing problem with drones and en route operations. Comput. Oper. Res. 109, 134–158.
Schermer, D., Moeini, M., Wendt, O., 2019. A matheuristic for the vehicle routing problem with drones and its variants. Transport. Res. Part C: Emerg. Technol. 106, 166–204.
Scott, J.E., Scott, C.H., 2018. Models for drone delivery of medications and other healthcare items. Int. J. Healthcare Inform. Syst. Inform. (IJHISI) 13 (3), 20–34.
Shavarani, S.M., Nejad, M.G., Rismanchian, F., Izbirak, G., 2018. Application of hierarchical facility location problem for optimization of a drone delivery system: a case study of Amazon Prime Air in the city of San Francisco. Int. J. Adv. Manuf. Technol. 95 (9–12), 3141–3153.
Song, B.D., Park, K., Kim, J., 2018. Persistent UAV delivery logistics: MILP formulation and efficient heuristic. Comput. Industr. Eng.
Tang, Z., van Hoeve, W.J., Shaw, P., 2019. A study on the traveling salesman problem with a drone. In: International Conference on Integration of Constraint Programming, Artificial Intelligence, and Operations Research. Springer, pp. 557–564.
Tilley, A., 2016. UPS experiments with drone delivery in partnership with Zipline, 2016. URL <https://www.forbes.com/sites/aarontilley/2016/05/09/ups-experiments-with-drone-delivery-in-partnership-with-zipline/#58f1c1d145a4>.
Toor, A., 2016. Drones will begin delivering blood and medicine in the U.S., 2016. URL <https://www.theverge.com/2016/8/2/12350274/zipline-drone-delivery-us-launch-blood-medicine>.
Tu, P.A., Dat, N.T., Dung, P.Q., 2018. Traveling salesman problem with multiple drones. In: Proceedings of the Ninth International Symposium on Information and Communication Technology, ACM, pp. 46–53.
Ulmer, M.W., Thomas, B.W., 2018. Same-day delivery with heterogeneous fleets of drones and vehicles. Networks 72 (4), 475–505.
Vanian, J., 2016. This drone startup just achieved a milestone in doorstep delivery, 2016. URL <http://fortune.com/2016/03/25/flirtey-drone-legal-delivery-urban/>.
Venkatachalam, S., Sundar, K., Rathinam, S., 2017. Two-stage stochastic programming model for routing multiple drones with fuel constraints. arXiv preprint arXiv:1711.04936, 2017. <https://arxiv.org/pdf/1711.04936.pdf>.
Wang, D., Hu, P., Du, J., Zhou, P., Deng, T., Hu, M., 2019a. Routing and scheduling for hybrid truck-drone collaborative parcel delivery with independent and truck-

carried drones. IEEE Internet Things J.
Wang, K., Yuan, B., Zhao, M., Lu, Y., 2019b. Cooperative route planning for the drone and truck in delivery services: a bi-objective optimisation approach. J. Oper. Res. Soc., pp 1–18, 2019b.
Wang, X., Poikonen, S., Golden, B., 2016. The vehicle routing problem with drones: several worst-case results. Optim. Lett. 4 (11), 679–697.
Wang, Z., Sheu, J.B., 2019. Vehicle routing problem with drones. Transport. Res. Part B: Method. 122, 350–364.
Wells, G., Stevens, L., 2016. Amazon conducts first commercial drone delivery, 2016. URL <https://www.wsj.com/articles/amazon-conducts-first-commercial-drone-delivery-1481725956>.
Wikarek, J., Sitek, P., Zawarczyński, Ł., 2019. An integer programming model for the capacitated vehicle routing problem with drones. In: International Conference on Computational Collective Intelligence. Springer, pp. 511–520.
Yoon, J.J., 2018. The traveling salesman problem with multiple drones: An optimization model for last-mile delivery. Master's thesis. Institute of Technology, Massachusetts.
Zhong, L., Garlichs, K., Yamada, S., Takano, K., Ji, Y., 2018. Mission planning for UAV-based opportunistic disaster recovery networks. In: Consumer Communications & Networking Conference (CCNC), 2018 15th IEEE Annual. IEEE, pp. 1–6.