The Drone-Assisted Traveling Salesman Problem with Robot Stations

Daniel Schermer   Mahdi Moeini   Oliver Wendt
Chair of Business Information Systems and Operations Research,
Technische Universität Kaiserslautern, Germany
{daniel.schermer, mahdi.moeini, wendt}@wiwi.uni-kl.de

Abstract

In this paper, we study the Drone-Assisted Traveling Salesman Problem with Robot Stations (TSP-D-RS). Specifically, we assume that there is a single truck that is equipped with a drone, and one or more potential sites of stations that might accommodate some robots. The TSP-D-RS asks for a valid route of the truck as well as feasible utilization of the drone and robots, such that all customers are served and minimal delivery time (makespan) or cost is accomplished. We provide a Mixed Integer Linear Programming formulation of the problem and perform a detailed numerical study. Through our numerical results, it is revealed that our formulation can be effectively addressed by a state-of-the-art solver. In addition, we demonstrate that optimizing the makespan coincides with reduced costs. In contrast, optimizing the operational costs might increase the makespan significantly. Furthermore, depending on the objective function, the operational utilization of the vehicles differs.

1. Introduction

Drones are on the brink of achieving market maturity as a commercial technology for civil applications in many public and private sectors. In particular, drones have already been successfully applied for monitoring tasks in agriculture, energy, or infrastructure, and for the delivery of packages (see [9] and references therein). Besides that, to reduce the negative impact of traffic in urban areas, there is a rising interest in using autonomous delivery robots in last-mile delivery (see, e.g., [2, 7, 16]). Due to the complementary nature of trucks, drones, and robots such as, e.g., their different speeds and carrying capacities, in a combined approach, benefits might emerge in terms of improved delivery times or reduced operational costs.

As a result, in this paper, we introduce the *Drone-Assisted Traveling Salesman Problem with Robot*

Stations (TSP-D-RS)[1] (see also Figure 1). To be more specific, we study the interoperability of a drone-assisted truck (refer to [1, 8]) with the possibility of using remote robot stations (refer to [6, 15]) in an integrated model. In this problem, we assume that a truck with sufficient capacity, which carries a drone, is located at a central depot. Moreover, we expect that a set of customer locations, each of them with an equal type of demand, as well as the potential sites of robot stations are known in advance. A subset of stations, which might accommodate a given number of autonomous robots, can be visited by the truck. The TSP-D-RS asks for a valid route of the truck, feasible operations of the drone, and admissible use of the robots, such that all customers are served and minimal delivery time (*makespan*) or cost is accomplished. Notably, this problem is concerned with various aspects of *synchronization* that determine the interplay between the vehicles [3]. In other words, the interactions between the truck, drone, and robots are governed by several operational constraints.

The remainder of this paper is organized as follows. We begin with a brief literature review in Section 2. Afterwards, in Section 3, we clarify the assumptions of the TSP-D-RS and provide a *Mixed Integer Linear Programming* (MILP) formulation of the problem. Computational experiments and their numerical results are then presented in Section 4. Finally, in Section 5, we draw concluding remarks on our work and derive some possible implications for future research.

2. Related Literature

In [8] two novel problems were introduced, where a truck is assisted by a single drone in the context of last-mile delivery. The first one is called the *Flying Sidekick Traveling Salesman Problem* (FSTSP). Here, if the depot is remotely located from the demand centers,

---

[1]In this work, we make no strong assumptions about the technical nature of *drones* and *robots*. However, for the sake of readability, in our problem, we use the term *drone* to refer to the (unmanned aerial) vehicle that travels along with the truck and the term *robot* to refer to one of the (unmanned ground) vehicles that are assigned to a station.
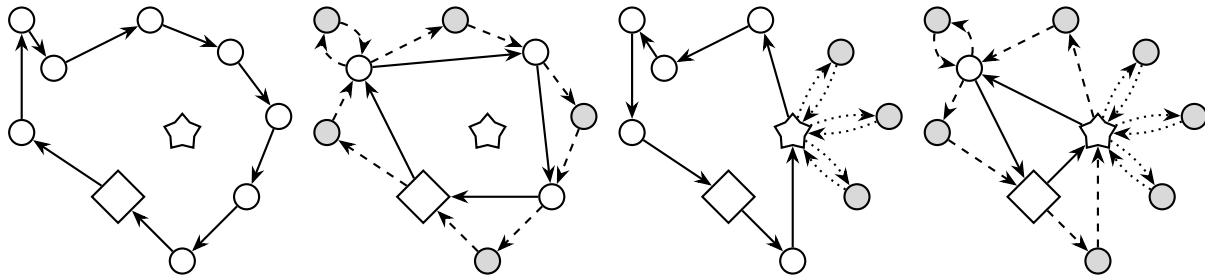
HICSS

Figure 1. An illustration of TSP, TSP-D, TSP-DS, and TSP-D-RS solutions (from left to right): the paths of the truck, drones, and robots are indicated by solid, dashed, and dotted lines, respectively. In each figure, the depot and drone station (micro depot) are indicated by the square and star shapes; however, the robots may only be utilized in the TSP-DS and TSP-D-RS.

it might be beneficial to have the drone working in close collaboration with the truck. Therefore, for the FSTSP, we assume that the drone is taken along by the truck and might be launched at some locations to initiate a multi-leg flight. In detail, the first leg consists of the loaded drone departing from the truck to perform an unattended delivery to a customer. Subsequently, as the drone needs to be resupplied, the second (empty) leg consists of the drone autonomously returning to the truck. The objective is to serve all customers and achieve minimal delivery time. Noteworthy, in the FSTSP, the drone cannot be recovered at the same location from which it was launched, i.e., round trips are forbidden [8]. In contrast, in a closely related problem, named the *Traveling Salesman Problem with Drone* (TSP-D) [1], such a move is permitted (see Figure 1). In recent years, a rapidly increasing number of research papers have been published, which follow the general idea of the FSTSP and TSP-D (see, e.g., [10, 11, 14, 16, 17]). These works address variants, provide theoretical insights, or develop advanced computational methods for solving problems involving a truck-drone tandem.

In contrast to the FSTSP, if the central depot is located within a dense customer cluster, the *Parallel Drone Scheduling Traveling Salesman Problem* (PDSTSP) provides a different perspective [8]. In this case, the truck and drone perform their actions more independently from one another. Indeed, while the truck follows a predefined tour to serve remote locations, the drone might be used to fulfill requests that are located in the area surrounding the depot. While the truck serves the customers that might be inaccessible by the drone, the latter continuously performs round trips: starting from the depot, a customer is served, and the drone resupplied afterwards. Hence, a much smaller degree of synchronization is required in the PDSTSP [8].

In practice, it seems unlikely that the central depot is located in close proximity to demand centers — in order to overcome this issue, the *Traveling Salesman*

Problem with a Drone Station (TSP-DS) was introduced as a generalization of the PDSTSP [6] (see also Figure 1). More precisely, apart from the central depot, where the truck is initially located, it is assumed that there is a single *drone station* (or *micro depot*), which provides shelter to one or more drones. In the TSP-DS, as soon as the drone station has been supplied by the truck, the drones can start their operation in a manner that mirrors the PDSTSP. In this problem, as in the FSTSP and PDSTSP, the objective is to serve all customers with minimal makespan, by effectively utilizing the drone station. Specifically, when the number of drones that are located at the station and their relative velocity are sufficiently large, it is possible to determine from the start which customers should be assigned to the truck or drone, respectively. Moreover, under these special assumptions, it is possible to decompose the problem into two independent subproblems [6]: a *Traveling Salesman Problem* to determine the route of the truck and a *Parallel Machine Scheduling Problem* to create a schedule for the drones that are located at the station.

Compared to existing large-scale central depots, drone stations might be a low-cost infrastructure of much smaller scale. Above all, they need to provide shelter along with the infrastructure that is needed for automated storage and retrieval of parcels as well as recharging of drones. Consequently, it is conceivable to incorporate the decision of locating (or selecting) a drone station from several potential sites into the problem at hand. Indeed, this was addressed in the *Traveling Salesman Drone Station Location Problem* (TSDSLP) that treats the routing of the truck, facility location of the drone stations, and scheduling of the drones in an integrated model [15]. In general, works that follow the concept of the FSTSP conclude that a drone must have a sufficiently high velocity to provide significant improvements w.r.t. the makespan (see, e.g. [1, 13]). In contrast, [6, 15] can show that drone stations are also effective even when the speed of the drones is

low compared to that of the truck.

To the best of our knowledge, at this point, there is no work that combines the concepts of the FSTSP (or TSP-D) with the general idea of using drone stations. However, in an effort to further reduce delivery times or costs, it might be beneficial to consider an integrated delivery system that merges a (costly high-speed aerial) drone that travels along with the truck with some (cheaper ground-operated and slow-moving) robots that are coordinated directly from stations.

3. Problem Definition

In this section, we formally introduce the TSP-D-RS, which integrates the *TSP with a drone* (refer to [1, 8]) and the *TSP with robot stations* (refer to [6, 15]). To be more precise, the TSP-D-RS combines the concepts of a drone that travels along with a truck with the possibility of the truck driver utilizing stations that house unmanned autonomous units, e.g., robots, which might be used for performing unattended deliveries.

For the description of the problem, suppose that the truck-drone tandem is initially located at a central depot. Furthermore, we accept the presence of one or more robot stations that provide accommodation to a fixed number of robots. In the TSP-D-RS, the objective is to serve a given set of customers by utilizing the truck, drone, and robots in such a way that all vehicles have returned to their initial location as quickly as possible, i.e., with minimal *makespan*. Alternatively, we consider an objective that is based purely on the operational cost. We accept the following assumptions regarding the nature of the drone and robots [1, 6, 8, 15]:

- When the drone is launched from the truck, it can serve exactly one customer and afterwards it needs to return to the truck. Similarly, we ask that a robot can deliver exactly one parcel to a customer. Then, it also needs to return to the station from which it was launched for a resupply.

- We assume that the drone and robots have a limited endurance of $\mathcal{E}_d$ and $\mathcal{E}_r$ *time units* per operation, respectively. Moreover, we ask that the batteries of the drone and robots are recharged (or swapped) instantaneously after each operation.

- While the truck is restricted to follow the road network, the drone and robots might be able to use different routes with different speeds. Hence, the times required to travel between two locations can differ for the truck, drone, and robots.

- In this problem, we assume that at most $C \in \mathbb{Z}_{\geq 0}$ robot stations can be utilized by the truck (which might be a managerial decision).

Let us present the notation that we are going to use throughout the paper. Assume that a complete graph $\mathcal{G} = (V, E)$ is given, where V is the set of vertices and E is the set of edges. We identify several pairwise disjoint subsets in V:

- $\{0, n+1\} \subset V$ marks the depot at the start and end of the tour, respectively.

- $V_N = \{1, \ldots, n\} \subset V$ is the set of all customers. Moreover, $V_D \subseteq V_N$ is the subset of customers that may be served by either the truck, drone, or robot. But, $V_N \setminus V_D$ is the subset of customers that may be served by truck only.

- $V_S = \{s_1, \ldots, s_m\} \subset V$ is the set of potential robot station sites that are present in the network.

Thus, $V = \{0\} \cup V_N \cup V_S \cup \{n+1\}$, where $0 \equiv n+1$. To simplify the notation, we introduce two further sets named $V_L = V \setminus \{n+1\}$ and $V_R = V \setminus \{0\}$ that mark the locations from which a drone may be *launched* and *retrieved*, respectively. A *drone operation* (or *sortie*) is characterized by a triple $(i, w, j) \in P$, where P is the set of all feasible operations and the following conditions must be fulfilled [1, 8]:

- The location i, from which the drone is launched, must be contained in the set V_L.

- The customer $w \in V_D$, that is served by the drone, must be different from i.

- At any retrieval location j, $w \neq j$ must hold. In addition, if $i = j$, we call it a *cyclic* operation or a *round trip*. Otherwise, if $i \neq j \in V_R$, we call it an *acyclic* or *multi-leg* operation.

By the parameters d_{ij}, $\overline{d}_{ij}$, and $\tilde{d}_{ij}$, for all $i, j \in V$, we define the distance required to travel from vertex i to vertex j by truck, drone, and robot, respectively. In a similar fashion, we use t_{ij}, $\overline{t}_{ij}$, and $\tilde{t}_{ij}$ to define the times required to travel from vertex i to j depending on the mode of transportation. Furthermore, we assume that $\overline{t}_l$ (respectively, $\overline{t}_r$) time units are required to launch (respectively, recover) the drone [8].

We assume that each robot station accommodates a limited and identical number of robots $K := \{1, \ldots, k\}$, where $|K| \in \mathbb{Z}_{\geq 0}$. A *robot delivery* is characterized by a triple (k, s, w) as follows: the robot $k \in K$ starts from a station $s \in V_S$, fulfills a request at $w \in V_D$, and returns to the same station from which it was launched [6, 15].

3.1. Minimal Makespan TSP-D-RS

To formulate the TSP-D-RS, we use the decision variables according to Table 1. Using this notation, we

Table 1. Decision variables used for the TSP-D-RS formulation.

$\tau \in \mathbb{R}_{\geq 0}$	defines the makespan.	$a_i \in \mathbb{R}_{\geq 0}$ $\forall i \in V$	indicates the arrival time of the truck at i.
$x_{ij} \in \{0,1\}$ $\forall i \in V_L, j \in V_R, i \neq j$	is equal to 1, iff arc (i,j) is used by the truck.	$s_i \in \mathbb{R}_{\geq 0}$ $\forall i \in V$	states the earliest possible departure time of the truck from i.
$y_{iw} \in \{0,1\}$ $\forall i \in V_L, w \in V_D, i \neq w$	is equal to 1, iff the drone serves w from i, i.e., arc (i,w) is the start of a multi-leg.	$\overline{r}_i \in \mathbb{R}_{\geq 0}$ $\forall i \in V$	indicates the earliest possible release time of the drone at i.
$y'_{wj} \in \{0,1\}$ $\forall w \in V_D, j \in V_R, w \neq j$	is equal to 1, iff the drone returns from w to j, i.e., arc (w,j) is the end of a multi-leg.	$\overline{s}_i \in \mathbb{R}_{\geq 0}$ $\forall i \in V$	states the earliest possible departure time of the drone from i.
$\hat{y}_{iw} \in \{0,1\}$ $\forall i \in V_L, w \in V_D, i \neq w$	is equal to 1, iff the drone performs a round trip (i,w,i).	$y_i \in \{0,1\}$ $\forall i \in V$	is equal to 1, iff the drone is available for an operation at i.
$z^k_{sw} \in \{0,1\}$ $\forall k \in K, s \in V_S, w \in V_D$	is equal to 1, iff robot k serves customer w from station s.		

introduce the following MILP formulation (1) - (27):

$$\min \quad \tau \tag{1}$$

$$\text{subject to} \quad s_{n+1} \leq \tau \tag{2}$$

$$a_s + \sum_{w \in V_D} (\tilde{t}_{sw} + \tilde{t}_{ws}) z^k_{sw} \leq \tau : \forall s \in V_S, k \in K \tag{3}$$

In the TSP-D-RS, according to (1), the objective is to minimize the makespan τ. The makespan has multiple lower bounds which are given by the time at which the truck-drone tandem has returned to the depot as well as the time at which each robot station has concluded its operation. These bounds are defined in constraints (2) and (3) for all vehicle types.

$$\sum_{j \in V_R} x_{0j} = \sum_{i \in V_L} x_{i,n+1} = 1 \tag{4}$$

$$\sum_{\substack{i \in V_L, \\ i \neq h}} x_{ih} = \sum_{\substack{j \in V_R, \\ h \neq j}} x_{hj} : \forall h \in V_N \cup V_S \tag{5}$$

The flow of the truck is defined through constraints (4) and (5). More precisely, according to constraints (4), the truck should commence and conclude its tour exactly once. Moreover, for each vertex h, that is visited by the truck in between, the flow must be conserved according to constraints (5).

$$\sum_{\substack{i \in V_L, \\ i \neq w}} x_{iw} = 1 : \forall w \in V_N \setminus V_D \tag{6}$$

$$\sum_{\substack{i \in V_L, \\ i \neq w}} (x_{iw} + y_{iw} + \hat{y}_{iw}) \\ + \sum_{s \in V_S} \sum_{k \in K} z^k_{sw} = 1 : \forall w \in V_D \tag{7}$$

In the TSP-D-RS, every customer must be served exactly once. Constraints (6) cover the customers

that can be served exclusively by the truck. For the remaining customers, constraints (7) guarantee that they are served exactly once by either truck, drone, or robot.

$$y_{iw} \leq \sum_{\substack{j \in V_R \\ i \neq j}} x_{ij} : \forall i \in V_L, w \in V_D, i \neq w \tag{8}$$

$$\hat{y}_{iw} \leq \sum_{\substack{j \in V_R \\ i \neq j}} x_{ij} : \forall i \in V_L, w \in V_D, i \neq w \tag{9}$$

$$y'_{wj} \leq \sum_{\substack{i \in V_L \\ i \neq j}} x_{ij} : \forall w \in V_D, j \in V_R, w \neq j \tag{10}$$

$$\sum_{\substack{i \in V_L \\ i \neq w}} y_{iw} = \sum_{\substack{j \in V_R \\ w \neq j}} y'_{wj} : \forall w \in V_D \tag{11}$$

The TSP-D-RS has intensive operational *synchronization* requirements. For the truck and the drone, these are handled through constraints (8) - (10). To be more precise, through constraints (8) (respectively, (9)), the truck is required to visit every vertex i that serves as a starting point for a multi-leg (respectively, round trip) operation. Furthermore, when a drone is retrieved at some vertex j after a multi-leg operation, the truck is required to visit the vertex due to constraints (10). Finally, constraints (11) guarantee that the flow is conserved during multi-leg operations.

$$a_i \leq s_i : \forall i \in V \tag{12}$$

$$M(x_{ij} - 1) + s_i + t_{ij} \leq a_j \\ : \forall i \in V_L, j \in V_R, i \neq j \tag{13}$$

Temporal constraints for the truck are given through constraints (12) and (13). The first set of constraints applies the arrival time as a lower bound on the departure time. In addition, the second set of constraints provides

a lower bound on the arrival time at j for every arc (i, j) that is used by the truck.

$$M(y_{iw} - 1) + \bar{s}_i + (\bar{t}_l + \bar{t}_{iw}) \leq \bar{r}_w$$
$$: \forall i \in V_L, w \in V_D, i \neq w \quad (14)$$

$$M(y'_{wj} - 1) + \bar{s}_w + (\bar{t}_{wj} + \bar{t}_r) \leq \bar{r}_j$$
$$: \forall w \in V_D, j \in V_R, w \neq j \quad (15)$$

For every multi-leg (i, w, j) performed by the drone, constraints (14) and (15) provide lower bounds on the release time of the drone at the customer and retrieval location. These constraints incorporate the departure time of the drone as well as the overhead times and flight times associated with each leg.

$$\bar{r}_i + \sum_{\substack{w \in V_D, \\ i \neq w}} (\bar{t}_l + \bar{t}_{iw} + \bar{t}_{wi} + \bar{t}_r) \hat{y}_{iw}$$
$$\leq \bar{s}_i : \forall i \in V \quad (16)$$

$$\bar{s}_i + \bar{t}_l \sum_{\substack{w \in V_D, \\ i \neq w}} y_{iw} \leq s_i : \forall i \in V \quad (17)$$

$$a_j + \bar{t}_r \sum_{\substack{w \in V_D, \\ w \neq j}} y'_{wj} \leq \bar{r}_j : \forall j \in V \quad (18)$$

Due to constraints (16), the earliest possible departure time $\bar{s}_i$ of the drone depends on its release time $\bar{r}_i$ at i and the time spent performing round trips at the same vertex. Moreover, the earliest possible departure time of the truck is bounded by the earliest possible release time of the drone. If a drone starts a multi-leg, the overhead time to launch the drone must be applied which is stated through constraints (17). Similarly, expressed through constraints (18), when a drone is retrieved after a multi-leg, the overhead time to retrieve the drone must be considered before the drone can be released again.

$$(\bar{t}_{iw} + \bar{t}_{wi}) \hat{y}_{iw} \leq \mathcal{E}_d : \forall i \in V_L, w \in V_D, i \neq w \quad (19)$$

$$\sum_{\substack{i \in V_L, \\ i \neq w, \\ i \neq j}} \bar{t}_{iw} y_{iw} + (\bar{r}_j - \bar{s}_w) \leq \mathcal{E}_d + M(1 - y'_{wj})$$
$$: \forall w \in V_D, j \in V_R \quad (20)$$

Constraints (19) and (20) define the maximum endurance for round trips and multi-leg drone flights.

$$(1 - x_{ij}) + y_i + \sum_{\substack{w \in V_D, \\ w \neq i \\ w \neq j}} (-y_{iw} + y'_{wj})$$
$$\geq y_j : \forall i \in V_L, j \in V_R, i \neq j \quad (21)$$

Constraints (21) account for the availability of the drone: for every arc (i, j) used by the truck, these constraints place a valid upper bound on the variable y_j depending on the previous availability, i.e., y_i, as well as starts of a multi-leg at i and recoveries after a multi-leg at j.

$$\sum_{\substack{w \in V_D, \\ w \neq i}} y_{iw} \leq y_i : \forall i \in V_L \quad (22)$$

$$\sum_{\substack{w \in V_D, \\ w \neq j}} y'_{wj} \leq y_j : \forall j \in V_R \quad (23)$$

$$\sum_{\substack{w \in V_D, \\ i \neq w}} \hat{y}_{iw} \leq n y_i : \forall i \in V_L \quad (24)$$

If a drone is available, it can start or end a multi-leg at most once and perform an arbitrary number of round trips at the same location. This behavior is expressed through constraints (22) - (24). At this point, we have fully specified the interaction of the truck and drone. Next, we will integrate the robot stations into the TSP-D-RS.

$$\sum_{\substack{i \in V_L}} \sum_{\substack{s \in V_S \\ i \neq s}} x_{is} \leq C \quad (25)$$

$$\sum_{k \in K} \sum_{w \in V_D} z_{sw}^k \leq n \sum_{\substack{i \in V_L, \\ i \neq s}} x_{is} : \forall s \in V_S \quad (26)$$

We say that a station is *used* if it is visited by the truck. Through constraints (25), we require that at most $C \in \mathbb{Z}_{\geq 0}$ stations may be used. Furthermore, constraints (26) guarantee that an arbitrary number of robot deliveries might be performed at each station s that is visited by the truck. We conclude our model with constraints (27) that guarantee the maximum endurance to be respected for all robot operations.

$$(\tilde{t}_{sw} + \tilde{t}_{ws}) z_{sw}^k \leq \mathcal{E}_r : \forall k \in K, s \in V_S, w \in V_D \quad (27)$$

Even though MILP (1) - (27) formulates the TSP-D-RS, the linear relaxation can be improved by including some simple yet highly effective valid inequalities (see [14]). For the purpose of this work, we adapt them through constraints (28) and (29) as follows:

$$\sum_{i \in V_L} \sum_{\substack{j \in V_R, \\ i \neq j}} t_{ij} x_{ij}$$
$$+ \sum_{i \in V_L} \sum_{\substack{w \in V_D, \\ i \neq w}} \bar{t}_l y_{iw} + \sum_{w \in V_D} \sum_{\substack{j \in V_R, \\ w \neq j}} \bar{t}_r y'_{wj}$$
$$+ \sum_{i \in V_L} \sum_{\substack{w \in V_D, \\ i \neq w}} (\bar{t}_l + \bar{t}_{iw} + \bar{t}_{wi} + \bar{t}_r) \hat{y}_{iw} \leq \tau \quad (28)$$

$$\sum_{\substack{i\in V_L}}\sum_{\substack{w\in V_D,\\i\neq w}}(\bar{t}_l+\bar{t}_{iw})y_{iw}$$
$$+\sum_{\substack{w\in V_D}}\sum_{\substack{j\in V_R,\\w\neq j}}(\bar{t}_{wj}+\bar{t}_r)y'_{wj} \qquad (29)$$
$$+\sum_{\substack{i\in V_L}}\sum_{\substack{w\in V_D,\\i\neq w}}(\bar{t}_l+\bar{t}_{iw}+\bar{t}_{wi}+\bar{t}_r)\hat{y}_{iw}\leq\tau$$

These constraints state that the makespan is bounded by the time spent traveling by either the truck or drone and the time shared by both vehicles when they perform *synchronized* actions: these actions occur during the launches and recoveries associated with multi-leg operations, i.e., whenever the overhead times $\bar{t}_l$ and $\bar{t}_r$ are applied, as well as during round trips.

As a general remark, our model loosely follows the MILP models presented in [6, 14, 15]. However, with respect to modelling the interaction between the truck and drone, the model provided in this paper is far more compact. In particular, through the introduction of constraints (21), we explicitly account for the availability of the drone at every vertex without tracking *circumjacent* drone operations (refer to [13, 14]). Through this procedure, we *significantly* reduce the number of variables and constraints involved in the formulation. In fact, we only require $\mathcal{O}(|V|^2)$ variables and constraints for modelling the problem. Furthermore, in contrast to existing models (see, e.g., [6, 8, 15]), we require no explicit *subtour elimination constraints*. In fact, constraints (13) - (15) already eliminate the possibility of any subtour. Finally, we need only a few M-type constraints. To be more precise, these constraints only appear in the temporal constraints (13) - (15) and endurance constraints (20), which helps to strengthen the linear relaxation further. As a result of these improvements, to the best of our knowledge, at this time, we are the first to effectively solve a multitude of instances with 10 or more customers to optimality within acceptable runtime and to approach instances with up to 20 customers reasonably well through a general-purpose MILP solver (refer to Section 4).

3.2. Minimal Operational Cost TSP-D-RS

As an alternative to the *makespan minimization*, we might be interested in *cost minimization* instead. While minimizing the cost, we might be interested in the variable cost-per-mile that might be associated with the truck, drone, and every robot. To this end, we might use the following objective function:

$$\min\quad c_t\sum_{\substack{i\in V_L}}\sum_{\substack{j\in V_R\\i\neq j}}d_{ij}x_{ij}$$
$$+c_d\sum_{\substack{i\in V_L}}\sum_{\substack{w\in V_D,\\i\neq w}}\left(\left(\bar{d}_{iw}+\bar{d}_{wi}\right)\hat{y}_{iw}+\bar{d}_{iw}y_{iw}\right)$$
$$+c_d\sum_{\substack{w\in V_D}}\sum_{\substack{j\in V_R,\\w\neq j}}\bar{d}_{wj}y'_{wj} \qquad (30)$$
$$+c_r\sum_{s\in V_S}\sum_{k\in K}\sum_{w\in V_D}(\tilde{d}_{sw}+\tilde{d}_{ws})z_{sw}^k$$

In this objective function, the parameters $c_t, c_d, c_r \in \mathbb{R}^+$ determine the relative cost for each mile that the truck, drone, and robots are in operation, respectively.

As a concluding remark, as the TSP-D-RS (with either objective) generalizes the $\mathcal{NP}$-hard Traveling Salesman Problem, the TSP-D-RS is also $\mathcal{NP}$-hard.

4. Computational Experiments and their Numerical Results

We implemented the model (1) - (29) and solved it by the MILP solver Gurobi Optimizer 8.1.0 [4] with a time limit of 30 minutes. We carried out all experiments on compute nodes in an Intel Xeon E5-2670 CPU cluster where each node was allocated 8 GB of RAM.

4.1. Instance Generation

Several different assumptions have been made with regards to the technical coefficients that apply to trucks, drones, and robots. For trucks, these assumptions include speeds that range from 15 . . . 35 mph (refer to [2, 5, 7, 8, 12]) and, as drones are not exposed to the limitations of the road network, it is commonly assumed that they move faster than trucks. In the literature, for drones, the speed varies from 25 . . . 50 mph along with 15 . . . 40 minutes of flight time (refer to [5, 7, 8, 12]).

Generally, when it is of interest to minimize the makespan, faster drones are more favorable; however, the marginal utility is diminishing as we speed up drones [1, 14]. For the purpose of this work, we conservatively assume that we have a truck that moves at 20 mph and a drone that moves at 25 mph with a maximum flight time of 20 minutes. Moreover, we differentiate between robots that move close to walking speed, i.e., 5 mph, and faster robots that move at 15 mph [2, 15]. For both cases, we set the robots' endurance to 40 minutes. Further, we assume that the Manhattan distance applies to the truck while the Euclidean distance applies to the drone and robots. This is a reasonable assumption to make because

Table 2. Summary of the technical coefficients that are used for studying the TSP-D-RS

Scenario	Truck		Drone			Robots		
	v_t [mph]	c_t	v_d [mph]	$\mathcal{E}_D$ [min]	c_d	v_r [mph]	$\mathcal{E}_r$ [min]	c_r
Slow robot	20	1	25	20	1/10	5	40	1/20
Fast robot	20	1	25	20	1/10	15	40	1/20

drones might move as the crow flies and robots might follow routes that are inaccessible to the truck such as, e.g., pedestrian zones. For both of these networks, we assume that the distances are symmetric. Finally, we set the overhead times associated with a launch and recovery of the drone to $\bar{t}_l = \bar{t}_r = 1$ minute [5, 8].

To the best of our knowledge, there is no work that quantifies the variable cost-per-mile associated with trucks in comparison to (delivery) drones and robots. However, related works consider drones to be 10 to 25 times more cost-efficient than trucks (refer to [5, 12]). In this study, we accept these assumptions and use the cost ratios $c_t = 1, c_d = 1/10, c_r = 1/20$ for the minimal cost objective in (30). Table 2 contains the technological coefficients, under which the problem is studied.

Apart from the problem parameters, an instance is specified by a graph $\mathcal{G}(V, E)$ that determines the locations of the depot, stations, and customers as well as the distances between them. As the TSP-D-RS is a new concept, we generated several graphs that match to the problem.[2] More precisely, for each number of customers $n \in \{10, 15, 20\}$, we generated 10 instances as follows:

- The locations of the depot and n customers have been *randomly* placed within an 8×8 mile square region [8].

- For comparative purposes, we placed two robot stations that might be used. The first robot station is placed *randomly* within the square grid. The second robot station is placed at the coordinates (x_s, y_s) that depend on the locations of the n customers according to formula (31). This station might be located closer to customer clusters.

$$(x_s, y_s) := \frac{1}{n} \sum_{w \in V_N} (x_w, y_w) \qquad (31)$$

Afterwards, we solved these instances in four different ways. As a baseline, we treated them as a TSP. For comparative purposes, we also solved them (under the minimal makespan and cost objectives) as a TSP-D, i.e., when $C = 0$. For the TSP-D-RS, under the minimal makespan objective, we did the experiments with $C \in \{1, 2\}$ and $|K| \in \{1, 2\}$. In contrast, under the minimal cost objective, as no fixed cost is incurred for deploying the second robot, we only tested with $|K| = 1$.

For the TSP-D-RS, based on our choice of parameters $C, |K|$, and the two different scenarios portrayed in Table 2, we have 2^3 parameter vectors for the minimal makespan objective and 2^2 parameter vectors for the minimal cost objective. Therefore, given the 30 different problem instances, we have a total of 30 TSP, 60 TSP-D, as well as 120 minimal cost TSP-D-RS and 240 minimal makespan TSP-D-RS, which are all solved by the MILP solver. Notably, for a problem that is solved according to the minimal makespan objective, we also calculate the cost that can be associated with the solution. Analogously, under the minimal cost objective, we also compute the corresponding makespan. For solving the instances, we assumed that V_N coincides with V_D, i.e., all customers are drone- and robot-eligible. From a computational point of view, due to the complicating constraints (7), this is the most challenging case. As the decision of whether a customer should be served by the truck, drone, or robot must be made for all customers, $V_N = V_D$ provides the largest solution space.

4.2. Numerical Results

In this section, we provide the numerical results of our computational study. In particular, we comment on the performance of the MILP solver in addressing the proposed instances and the potential makespan and cost savings. Afterwards, we analyze the way in which the drone and robots are utilized depending on the selected objective function.

MILP Solver Performance Table 3 provides some insights into the performance of the MILP solver. More precisely, this table contains the average runtime and *Mixed Integer Programming* (MIP) gaps for the studied cases. Moreover, the table indicates how often the MILP solver was able to identify the provably optimal solution within the runtime limit of 30 minutes.

Regarding the TSP-D cases, overall, the numerical results show that our MILP formulation is well-suited for solving small-sized problems. Under the minimal cost objective, we can solve all instances to proven optimality within reasonable runtime. Moreover, under the minimal makespan objective, our formulation can solve instances with 10 customers to optimality within short runtime. Furthermore, we are able to approach

Table 3. Detailed results obtained through Gurobi Solver 8.1.0. This table contains the average runtime (in seconds), the average MIP gap after runtime, and the number of times (opt.) in which the provably optimal solution was found (values are averaged over 10 instances). The parameters n, v_r, $|K|$, and C are the number of customers, velocity of the robots, the amount of robots, and the number of stations that can be utilized.

| | | | TSP-D ($C = |K| = 0$) | | | | | | TSP-D-RS | | | | | | | | |
| | | | min cost | | | min makespan | | | min cost | | | min makespan ($|K| = 1$) | | | min makespan ($|K| = 2$) | | |
n	v_r	C	time	gap	opt.	time	gap	opt.	time	gap	opt.	time	gap	opt.	time	gap	opt.
10	5	1	1.2	0.0%	10/10	18.9	0.0%	10/10	1.7	0.0%	10/10	101.8	0.0%	10/10	62.9	0.0%	10/10
		2							2.2	0.0%	10/10	144.3	0.0%	10/10	108.6	0.0%	10/10
	15	1							1.1	0.0%	10/10	12.4	0.0%	10/10	5.6	0.0%	10/10
		2							1.2	0.0%	10/10	13.2	0.0%	10/10	4.8	0.0%	10/10
15	5	1	12.5	0.0%	10/10	739.1	1.5%	7/10	29.2	0.0%	10/10	883.8	1.8%	8/10	711.1	1.8%	8/10
		2							40.9	0.0%	10/10	949.0	2.0%	7/10	896.0	2.3%	7/10
	15	1							4.7	0.0%	10/10	774.0	2.0%	8/10	204.8	0.0%	10/10
		2							4.9	0.0%	10/10	699.0	1.8%	8/10	160.3	0.0%	10/10
20	5	1	193.8	0.0%	10/10	1,645.2	9.0%	1/10	365.1	0.0%	10/10	1,678.4	9.5%	1/10	1,571.9	9.8%	2/10
		2							451.4	0.0%	10/10	1,637.1	11.1%	2/10	1,614.8	9.4%	2/10
	15	1							46.4	0.0%	10/10	1,711.1	10.1%	2/10	1,780.6	10.6%	1/10
		2							45.9	0.0%	10/10	1,800.0	15.9%	0/10	1,580.2	10.5%	3/10
Average			69.2	0.0%	30/30	801.1	3.5%	18/30	82.9	0.0%	120/120	867.0	4.5%	76/120	725.1	3.7%	83/120

instances with 15 customers with low MIP gaps and can even solve 7 out of 10 to proven optimality. However, it also becomes apparent that performance degrades as the instance size is increased to $n = 20$, in particular, when we are optimizing the makespan instead of cost.

For the TSP-D-RS, we are able to solve instances with 10 to 15 customers reasonably well and achieve optimality in almost all cases. Indeed, under the minimal cost objective, we can even approach instances with 20 customers and achieve performance that rivals the TSP-D cases. Specifically, in the presence of fast robots, i.e., $v_r = 15$, the problem often becomes easier to solve. However, it also becomes apparent that, in general, the minimal makespan problem is much more difficult to solve: for $n = 20$ the remaining MIP gaps after runtime increase and the number of instances that are solved to optimality decreases significantly.

Makespan versus Cost Minimization For the purpose of illustrating the benefits of utilizing the drone and robots with regards to *cost versus makespan reduction*, we introduce the following metrics:

$$\Delta_\tau = \frac{\tau}{\tau_{\text{TSP}}^*} \quad \text{and} \quad \Delta_c = \frac{c}{c_{\text{TSP}}^*}, \tag{32}$$

where, τ and c are the objective values returned by the solver and τ_{TSP}^* as well as c_{TSP}^* stand for the optimal makespan and cost of the TSP solution, i.e., where no drone is present and no station is visited or used.

Figure 2 highlights the numerical results for the minimal makespan objective. Specifically, this figure shows the average savings over the TSP, i.e., Δ_τ and Δ_c, based on the number of available robots $|K|$ and their velocity v_r for a given value of using at most $C = 1$ stations. Note that, in this figure, the values for $K = 0$ are taken from the TSP-D solution (where $C = |K| = 0$ but the drone can be used). Notably, under the makespan objective, the costs are also reduced significantly. Based on the selected technological coefficients (refer to Table 2), the savings are of similar magnitude. In particular, robots that move close to walking speeds of 5 mph have a hardly noticeable influence under the minimal makespan objective. In contrast, as their speed is increased to 15 mph, they become increasingly useful.

Figure 3 visualizes the results for the minimal cost objective. As expected, under this objective, costs of much smaller magnitude can be realized. However, in contrast to the makespan objective, the costs and delivery time show a contrary development (see also [5]): as we increase the speed of the robots, the relative costs decrease further but there is a sharp increase in the makespan. As a possible explanation, an increase in speed also raises the robots' range (see constraints (27)); and while operating robots might be cheap (due to the cost factor c_r), this throttles the makespan nonetheless.

Station Utilization Regarding the use of the robot stations, we present Table 4, on which we can make the following observations. When the objective is to minimize the operational cost or if the number of used stations is limited to $C = 1$, it is much more likely that the weighted station is utilized. In contrast, this ratio becomes less definite under the minimal makespan objective, where it can also make sense to visit the randomly placed station. In these cases, it might also be useful to utilize a station as an intermediate hub in order to launch or recover the drone (see also [13]).
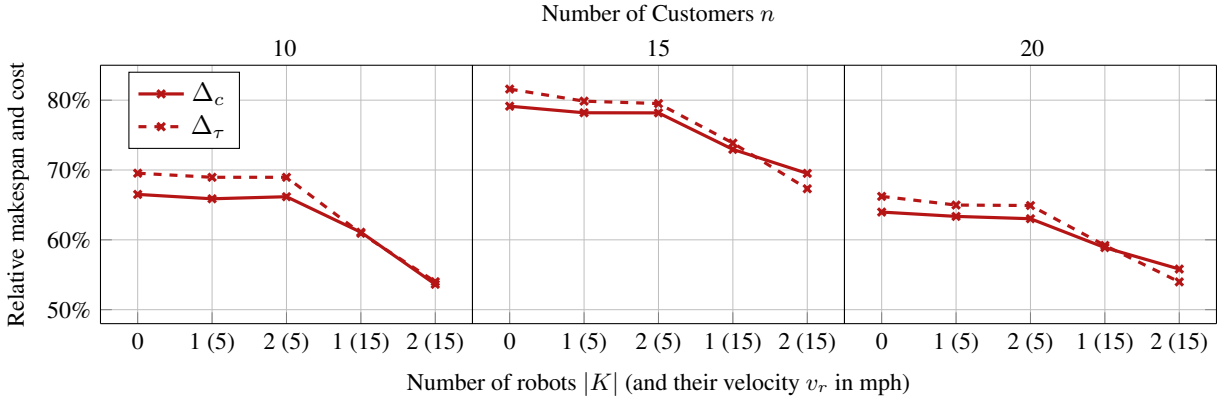
Figure 2. The relative makespan Δ_τ and cost Δ_c for the minimal makespan objective. These values are shown for $C = 1$ ($C = 0$ for the TSP-D solution, i.e., where $|K| = 0$) and averaged over 10 instances.
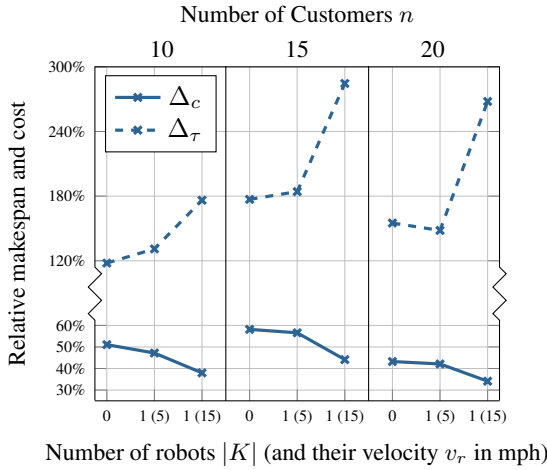

Figure 3. The relative makespan Δ_τ and cost Δ_c for the minimal cost objective. These values are shown for $C = |K| = 1$ ($C = 0$ for the TSP-D solution, i.e., where $|K| = 0$) and averaged over 10 instances.

Table 4. The share of times the weighted or randomly located stations are used under the minimal cost and minimal makespan (time) objective (averaged over 30 instances for $n \in \{10, 15, 20\}$).

v_r	C	min cost		min time ($|K|=1$)		min time ($|K|=2$)	
		weighted	random	weighted	random	weighted	random
5	1	63.3%	16.7%	16.7%	43.3%	26.7%	40.0%
	2	70.0%	16.7%	40.0%	50.0%	36.7%	43.3%
15	1	96.7%	3.3%	63.3%	33.3%	83.3%	16.7%
	2	96.7%	10.0%	93.3%	66.7%	100.0%	76.7%

considered, a round trip clearly is the most cost-effective operation that a drone can perform. More precisely, consider a feasible multi-leg $(i, w, j) \in P$. Based on the objective defined in (30), if $\overline{d}_{iw} < \overline{d}_{wj}$, it is always feasible and, in particular, more cost-efficient to perform the round trip (i, w, i) instead. Note that, analogous considerations are possible for the case of $\overline{d}_{iw} > \overline{d}_{wj}$.

On the contrary, when the objective is to achieve minimal makespan, the utilization of the truck ramps up significantly. Equally important, the predominant mode of drone delivery is the multi-leg and the share of customers served by robots declines noticeably. It is interesting to note that when fast robots are used, the shares of customers served by the drone and robot, respectively, are of a similar magnitude.

5. Conclusion

In this work, we introduced the TSP-D-RS, which combines the concepts of a truck-drone tandem with the possibility of robot stations. After a brief literature review, we formulated the problem as a MILP, and presented the numerical results of our computational study. We have shown that a state-of-the-art solver yields acceptable results within reasonable runtime for

Operational Characteristics We are also interested to investigate the operational characteristics of the drone and robots. For this purpose, Table 5 illustrates the share of customers that are served by the truck, through a multi-leg or a round trip, and by a robot.

If the objective is to achieve minimal operational cost, the utilization of the truck is expensive. Moreover, the cheaper modes of delivery, i.e., the drone and in particular the robots, are utilized to a large extent. Indeed, the predominant mode of operating the drone is the round trip. Noteworthy, to the best of our knowledge, most research works (existing in the literature) that study related problems under a minimal cost objective, do not permit drones to perform round trips (see, e.g., [5, 12]). However, if only the variable cost-per-mile is

Table 5. Share of customers served by each vehicle (averaged over 10 instances for $C = 1$).

n	v_r	minimal cost ($\|K\| = 1$)				minimal makespan ($\|K\| = 1$)				minimal makespan ($\|K\| = 2$)			
		truck	multi-leg	round trip	Robot	truck	multi-leg	round trip	Robot	truck	multi-leg	round trip	Robot
10	5	20.0%	0.0%	74.0%	6.0%	63.0%	31.0%	0.0%	6.0%	63.0%	31.0%	0.0%	6.0%
	15	14.0%	0.0%	16.0%	70.0%	54.0%	22.0%	0.0%	24.0%	40.0%	25.0%	0.0%	35.0%
15	5	21.3%	0.0%	72.7%	6.0%	67.3%	26.7%	0.0%	6.0%	64.7%	25.3%	0.0%	10.0%
	15	12.0%	0.0%	14.7%	73.3%	60.7%	21.3%	0.0%	18.0%	50.7%	18.7%	0.0%	30.7%
20	5	23.5%	0.0%	65.5%	11.0%	69.0%	25.0%	0.0%	6.0%	67.5%	25.5%	0.0%	7.0%
	15	12.0%	0.0%	15.5%	72.5%	62.5%	21.5%	0.0%	16.0%	53.0%	16.5%	0.0%	30.5%

instances with up to 20 customers. According to the numerical results, combining a truck-drone tandem along with properly situated robot stations can bring significant reductions in the delivery time or operational costs. Furthermore, drone round trips are a suitable mode of operation when the objective is to minimize the costs; however, we believe that one must be careful with the impact that this might have on the delivery time.

As the TSP-D-RS defines a new concept, the future research directions are numerous. A subsequent sensitivity analysis on parameters (e.g., drone speed) might provide some more in-depth insights on their respective impact. While our MILP formulation has shown favorable behavior, it will be necessary to rely on heuristic methods in order to address larger instances. Moreover, in the context of same-day-delivery, it might be of interest to formulate the TSP-D-RS as a multi-objective problem: to achieve an improved trade-off, we might weigh time against cost savings and include advanced cost metrics such as, e.g., a cost that is imposed for waiting (refer to [5]). Finally, it is also interesting to consider the impact of multiple trucks, several drones per truck or the possibility of enhanced robots with multiple compartments (see, e.g., [14, 16]).

Acknowledgements

References

[1] Agatz, N., Bouman, P., Schmidt, M.: Optimization Approaches for the Traveling Salesman Problem with Drone. Transportation Science **52**(4), 965–981 (2018)

[2] Boysen, N., Schwerdfeger, S., Weidinger, F.: Scheduling last-mile deliveries with truck-based autonomous robots. European Journal of Operational Research **271**(3), 1085–1099 (2018)

[3] Drexl, M.: Synchronization in Vehicle Routing—A Survey of VRPs with Multiple Synchronization Constraints. Transportation Science **46**(3), 297–316 (2012)

[4] Gurobi Optimization, LLC: Gurobi Optimizer Reference Manual (2019)

[5] Ha, Q.M., Deville, Y., Pham, Q.D., Hà, M.H.: On the min-cost Traveling Salesman Problem with Drone. Transportation Research Part C: Emerging Technologies **86**, 597–621 (2018)

[6] Kim, S., Moon, I.: Traveling Salesman Problem With a Drone Station. IEEE Transactions on Systems, Man, and Cybernetics: Systems **49**(1), 42–52 (2018)

[7] Moeini, M., Salewski, H.: A Genetic Algorithm for Solving the Truck-Drone-ATV Routing Problem. Advances in Intelligent Systems and Computing **991**, 1023–1032 (2020)

[8] Murray, C.C., Chu, A.G.: The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery. Transportation Research Part C: Emerging Technologies **54**, 86–109 (2015)

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

[10] Poikonen, S., Golden, B., Wasil, E.A.: A Branch-and-Bound Approach to the Traveling Salesman Problem with a Drone. INFORMS Journal on Computing **31**(2), 335–346 (2019)

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

[12] Sacramento, D., Pisinger, D., Ropke, S.: An adaptive large neighborhood search metaheuristic for the vehicle routing problem with drones. Transportation Research Part C: Emerging Technologies **102**, 289–315 (2019)

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

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

[15] Schermer, D., Moeini, M., Wendt, O.: The Traveling Salesman Drone Station Location Problem. Advances in Intelligent Systems and Computing **991**, 1129–1138 (2020)

[16] Sonnenberg, M.O., Leyerer, M., Kleinschmidt, A., Knigge, F., Breitner, M.H.: Autonomous Unmanned Ground Vehicles for Urban Logistics: Optimization of Last Mile Delivery Operations. Proceedings of the 52nd Hawaii International Conference on System Sciences pp. 1538–1547 (2019)

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