Highlights

Two-indexed formulation of the traveling salesman problem with multiple drones performing sidekicks and loops[1]

Alexander Rave

- Deciding on the routing of one truck equipped with multiple drones that may perform both loops and sidekicks while minimizing the makespan (TSPmD)
- Introducing a two-indexed MILP formulation for the TSPmD that can be easily implemented and outperforms MILP formulations for the single drone case from the literature
- Generating new optimal solutions for instances from the literature with up to 28 customers for benchmark purposes
- Presenting managerial insights on the impact on the makespan when allowing drones to perform loops

[1]Working Paper submitted to SSRN

Two-indexed formulation of the traveling salesman problem with multiple drones performing sidekicks and loops

Alexander Rave[1,1*]

[1*]Department of Operations, Catholic University Eichstätt-Ingolstadt, Ingolstadt School of Management, Auf der Schanz 49, Ingolstadt, 85049, Germany.

Corresponding author(s). E-mail(s): ARave@ku.de;

Abstract

Aerial drone delivery has great potential to improve delivery time in package delivery, as drones can fly autonomously over obstacles at a possibly higher speed than trucks. The benefits of drones in delivery can even be increased in a truck-and-drone tandem where a truck carries one or multiple drones and releases them at advantageous places. This problem is known as the traveling salesman problem with multiple drones (TSPmD). We focus on a variant of this problem, where the drones have two options after serving the customer: they can return to a node the truck visits at a later stage (sidekick) or return to the same node they were launched from (loop).

While several variants of the TSPmD have been studied, considering sidekicks and loops in combination with minimizing the makespan is not represented so far. We fill this gap and introduce an efficient two-indexed mixed-integer linear program (MILP) formulation that provides new optimal solutions for instances with up to 28 nodes for benchmark purposes. Our MILP formulation outperforms two state-of-the-art MILP formulations from the literature considering one drone. In a case study, we analyze the impact on the minimization of the makespan when allowing drones to perform loops. Loops mainly become relevant when drones travel faster than trucks resulting in average makespan savings of up to 2.7%.

Keywords: Unmanned aerial vehicles, Routing, Last-mile delivery, Mixed-integer linear program, Benchmark instances

(a) Performed sidekick by a drone. (b) Performed loop by a drone.

Fig. 1: Illustration of drone routes.

1 Introduction

Parcel delivery via aerial drones is much considered in academia [e.g., 1] and also in practice [e.g., 2]. One known delivery option is the interaction of one or more drones with a delivery truck. The routing of both a truck and one drone is known in the literature as traveling salesman problem with drone (TSPD). In the case of multiple drones, the problem is called traveling salesman problem with multiple drones (TSPmD) [e.g., 3]. Initially introduced by [1], many publications on the truck-and-drone tandem only consider drones performing sidekicks, which means that they return to a node the trucks visits at any later stage in its tour (see Figure 1a). However, only a few publications consider additionally drones performing loops, meaning drones return to the same node they are launched from (see Figure 1b). Thereby, [4] already found that drones perform loops in the final found routing if these are allowed.

Considering multiple drones that can perform both sidekicks and loops while minimizing the makespan is not represented in the literature, and thus, the literature also lacks suitable benchmark instances. We fill this gap and introduce a compact two-indexed mixed-integer linear program (MILP) formulation of this TSPmD. This MILP formulation is easily implemented and outperforms MILP formulations for the single drone case from the literature for larger instances, i.e., it finds the optimal solution faster.

This paper contributes to the literature as follows: First, we introduce the TSPmD with drones performing both sidekicks and loops while minimizing the makespan. Second, we formulate the problem as MILP with only two-indexed variables that outperforms MILP formulations from the literature for the single drone case. Third, we present new optimal solutions for the instances of [1] and [5] for up to 28 nodes for benchmark purposes. Fourth, in a numerical study, we analyze the impact of allowing drone loops for different numbers of drones on minimizing the makespan and the runtime of a MILP solver and present managerial insights.

The structure of this paper is as follows. In Section 2, we describe the decisions and assumptions in detail. In Section 3, we present the relevant literature and delimit our

paper to this literature. Next, we introduce the MILP in Section 4. In Section 5, we describe the instances for which we present benchmark results, analyze the runtime, and give managerial insights on allowing drone loops. Last, in Section 6, we summarize the results and give a brief outlook.

2 Problem setting

We decide on the routing of one truck and its equipped multiple drones that need to visit certain nodes, i.e., customers. Additionally, we also decide if a customer is served by a truck or else by drone. For this, we minimize the makespan, i.e., the time until the last vehicle has returned to the depot, starting with the first release at time zero. The peculiarity of the considered problem setting is that we consider multiple drones and these can both perform sidekicks and loops. Additionally, we make the following assumptions:

- Traveling times of the truck and the drones generally differ and thus, different speeds and distance metrics can be considered.
- Drones have an endurance limit, i.e., maximum flight time per flight.
- Assumptions regarding customers:
  - All customers must be served once by truck or by drone.
  - Some customers cannot be served by drones, but all by truck [e.g., 1].
  - The truck has a sufficient high payload to serve all customers in one tour.
  - A drone can serve one customer per trip and has to return to the truck afterwards.
- Assumptions regarding synchronization:
  - If drones return to a node, the truck continues its tour at this node as soon as all returning drones have landed.
  - Drones do not wait for the truck but reduce their speed instead to a certain extent [e.g., 1, 6]. This allows to avoid unrealistic routes where the drone has to wait for almost the complete truck's tour at a node, for example, at the end of the tour.
  - Drones only return to the truck at a node the truck visits. This node must be the same node they are launched from (loop) or a node the truck visits later in its tour (sidekick).
  - Drones can perform loops at the depot, but these are only allowed at the end of the truck's tour, which results in waiting times at the end of the tour [e.g., 7].
  - Each drone can only perform at maximum one loop at a node.
- There are no service, preparation, or rendezvous times for trucks and drones.

- There are no loading or battery change times for drones.

Exemplary routing plan

For a better understanding of the problem setting, Figure 2 presents an example routing of the truck equipped with two drones serving ten customers. Customers (C_1,...,C_{10}) are numbered in the order of serving. Starting at the depot, C_1 is served by the truck first, where one drone is launched to perform a sidekick to serve C_2. The truck and the drone meet again at C_3, where a second drone performs a loop and serves C_4. Note that the truck continues its tour as soon as both drones have returned from serving C_2 and C_4. Next, the truck serves customer C_5 and launches a drone to perform a sidekick to serve C_7. Note that the drone can return to any node, the truck visits at a later stage as long as the flight time is within the endurance limit. At C_6, the second drone is launched to perform a sidekick to serve C_8. Both drones return to the truck at C_9. The truck and its drones return to the depot afterwards, where again, one drone is launched to serve C_{10} in a loop. The tour is finished as soon as this drone returns. Thus, the truck waits at the depot for the drone to return.

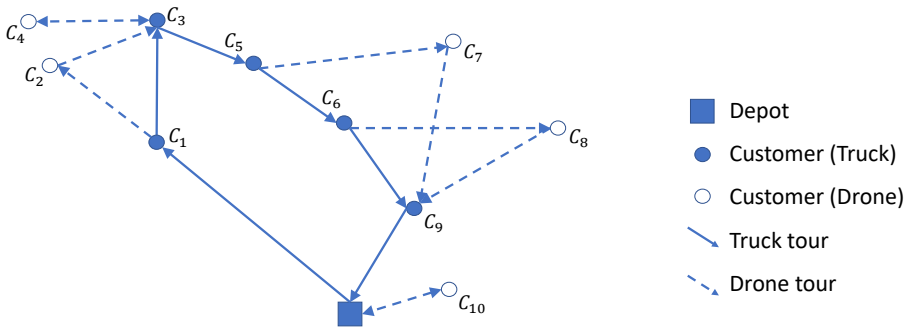

Fig. 2: Exemplary routing plan for ten customers served by one truck equipped with two drones.

3 Related literature

In this section, we differentiate this paper to the relevant literature regarding truck-and-drone tandems, where one or multiple trucks are equipped with at least one drone. Especially, we only focus on literature that includes a MILP formulation without special assumptions like, e.g., battery consumption or a sole supply of drones where the truck only has auxiliary means to launch drones at advantageous locations. First,

we present the literature where drones only perform sidekicks, and second, we present the literature where drones can perform both sidekicks and loops. We recommend the paper of [8] for an extensive review of drone operations and [9] for a more recent literature review.

Sidekicks

The truck-and-drone tandem is initially introduced by [1] who present a three-indexed MILP formulation with makespan minimization, which has difficulties to be solved to optimality, if instances with eleven nodes are considered. As a result, [10] and [11] published new and improved modelling approaches, which accelerate a standard solver. Further, multiple publications consider close problem settings with either a different objective or extensions like an increased number of trucks or drones. So [3] and [12] consider multiple drones and [13] and [14] additionally multiple trucks. Cost minimization is considered by [15] (single truck) and [16] (multiple trucks) who present heuristic solution approaches.

Sidekicks and loops

Drone loops are rarely represented in the literature, however, [4] show that the final found routing might include drone loops as well. The authors analyze – among other things – the number of loops and sidekicks performed in a truck-and-drone tandem with multiple trucks and drones. Drones' speed was found to be a significant influencing factor on performed loops. [17] propose two MILP formulations and analyze – among other things – the impact of loops on the objective of cost minimization, if the truck is equipped with an unlimited number of drones. The authors find that allowing drones to perform loops can significantly reduce routing costs. However, an analysis of the impact on minimizing the makespan when permitting loops and for different numbers of drones is missing. [18] present a multiple TSPD problem where the assumption that a drone might only return to the same truck is relaxed and thus drones can return to a different truck. The authors consider loops but only if performed at the depot. [7] derive several benchmark results for the instances of [1] for different drone settings including loops, however, multiple drones are missing. [19] and [20] present a two-indexed formulation of a TSPD. While [19] additionally introduce an exact branch-and-price approach, [20] accelerate the solver by adding upper and lower bounds by, e.g., starting with an initial solution that is generated by a greedy insertion heuristic, and additionally assess the data in a pre-processing step to reduce the number of possible drone arcs. Considering multiple drone delivery options, [6] present

both a MILP formulation and heuristics solution approach for a problem setting where drones may launch from trucks and additionally from the depot or microdepots.

Differentiation to the literature

Table 1 summarizes the major assumptions, the objective, the largest number of variables' indices in the MILP formulation, and the largest instance size (including the depot once) solved to optimality with the MILP formulation. While there are three two-indexed formulations for the TSPD [10, 19, 20], there is one publication with a two-indexed formulation for the TSPmD [3]. However, among other things, the author does not consider loops for drone delivery.

The truck-and-drone tandem presented in this paper extends the one of [6]. In contrast to their paper, we minimize the makespan, have additional assumptions (e.g., the return time of drones to the depot is sufficient), and an enhanced modelling version of some constraints accelerating the runtime of the solver.

	Assumptions			Objective			
	# Trucks	# Drones (per truck)	Loops	Costs	Make-span	# Variables' indices	Instance size
[1]	1	1			✓	3	-
[15]	1	1		✓		3	11
[10]	1	1			✓	2	14
[11]	1	1			✓	5	11
[16]	n	1		✓		3	13
[3]	1	m			✓	2	14
[12]	1	m			✓	3	25
[13]	n	m			✓	4	9
[14]	n	m			✓	5	9
[20]	1	1	✓		✓	2	32*
[7]	1	1	✓		✓	3	11
[19]	1	1	✓		✓	2	10
[17]	1	m	✓	✓		3	20
[4]	n	m	✓		✓	3	11
[18]	n	m	✓**		✓	3	10
[6]	n	m	✓	✓		3	15
This paper	1	m	✓		✓	2	28

Table 1: Comparison to the relevant literature. Note that each publication considers drones performing sidekicks.
* The authors added lower and upper bounds and a pre-processing step before running the MILP, ** drone loops only allowed at the depot.

To sum up, this paper introduces the first TSPmD considering both loops and sidekicks while minimizing the makespan. Moreover, we present a two-indexed MILP formulation that can solve instances with more nodes.

4 Two-indexed TSPmD formulation

In this section, we present the MILP formulation for the TSPmD. First, in Section 4.1, we describe the proceeding of creating the two-indexed formulation. Second, in Section 4.2, the used index sets, parameters, and variables are introduced. Third, in Section 4.3, the MILP formulation is presented, and last, in Section 4.4, the adjustment for preventing loops is described, as this is needed in the numerical study to show the impact of loops.

4.1 Proceeding of two-indexed formulation

A full flight of a drone can be interpreted as a three-node sortie (i, c, j) including the launch node i, the served customer c, and the landing node j [1], and thus consisting of the outbound flight to the customer and the return flight to the truck. To formulate a TSPD using only two-indexed variables, the drone's flights are split up into two variables instead of one: one for the outbound $\overrightarrow{y}_{i,c}$ and one for the return flights $\overleftarrow{y}_{c,j}$ and these flights are matched in the constraints [e.g., 10].

Considering multiple drones per truck, each vehicle is typically represented in an index set for vehicles [e.g., 13], which is, however, not necessary, i.e., the TSPmD can be formulated without the index set for vehicles. Each drone flight can be defined over the same variables $\overrightarrow{y}_{i,c}$ and $\overleftarrow{y}_{c,j}$ with a flow formulation [3]. For this, an auxiliary variable $(\pi_{i,j})$ is introduced to monitor the number of drones on the truck traveling from i to j.

To also include loops in the MILP, which only affect the truck's waiting time, a variable for the truck's waiting time $\phi_{i,j}$ is defined with two indices. $\phi_{i,j}$ determine the truck's waiting time at node j for each node i where a drone is launched, and thus the variable includes sidekicks $(i \neq j)$ and loops $(i = j)$. The truck's waiting time at node j results from the variable $\phi_{i,j}$ for each i. Thus, in contrast to the literature for the TSPD [e.g., 19, 20], there is no need for variables that track drones traveling an arc while staying on the truck, i.e., the drone is not flying.

4.2 Decisions and relevant parameters

We consider the index set for customers $\mathcal{I}$, customers including the depot $\mathcal{I}_0$, and customers that can be served by both the truck and drones $\mathcal{I}_D$. In contrast to, e.g., [7], we include the depot only once in the index set $\mathcal{I}_0$. Therefore, constraints regarding a launch or return at the depot are handled differently.

The decisions taken are the routing of the truck ($x_{i,j}$) and the routing of the drones ($\overrightarrow{y}_{i,c}$, $\overleftarrow{y}_{c,j}$). Additionally, the model decides on the customers served by truck (z_c^T) and drones (z_c^D).

The truck is equipped with m drones, each with an endurance e per flight. Similar to, e.g., [7], the different speeds or distance metrics of the truck and drones are also considered in $t_{i,j}^T$, and $t_{i,j}^D$, respectively.

Index sets	
$\mathcal{I}$, $\mathcal{I}_0$, $\mathcal{I}_D$	Node sets for customers, customers and the depot, and customers that can be served by drones.
Parameters	
e	Endurance time for each drone flight.
m	Number of drones on the truck.
M_l^{big}	Big M for $l = 1, 2, 3, 4$.
$t_{i,j}^T$, $t_{i,j}^D$	Truck's (drones') traveling time from node $i \in \mathcal{I}_0$ to node $j \in \mathcal{I}_0$.
Decision variables	
$x_{i,j}$	Binary variable indicating if the truck travels from node i to j $(i, j \in \mathcal{I}_0)$.
$\overrightarrow{y}_{i,c}$	Binary variable indicating if a drone is launched at node $i \in \mathcal{I}_0$ to serve customer $c \in \mathcal{I}_D$.
$\overleftarrow{y}_{c,j}$	Binary variable indicating if a drone returns from customer $c \in \mathcal{I}_D$ to node $j \in \mathcal{I}_0$.
z_c^T, z_k^D	Binary variable indicating if customer $c \in \mathcal{I}$ $(k \in \mathcal{I}_D)$ gets served by truck (drone).
Auxiliary variables	
u_c	Real-value variable for subtour elimination by [21] $(c \in \mathcal{I})$.
$\lambda_{i,c}$	Binary variable indicating if customer $c \in \mathcal{I}_D$ is served in a drone loop launched at node $i \in \mathcal{I}_0$.
$\pi_{i,j}$	Positive integer variable monitoring the number of drones that travel on the truck from node i to j $(i, j \in \mathcal{I}_0)$ and thus do not serve a customer at that time.
$\tau_{i,j}$	Positive real-value variable indicating the accumulated travel time of the truck traveling from node i to j including the waiting times for drones at node $j \in \mathcal{I}$. Note that waiting times at the depot are excluded and must be considered separately in the objective function.
τ^*	Positive real-value variable indicating the makespan.
$\phi_{i,j}$	Positive real-value variable indicating the waiting times of the truck on the last returning drone at node j, if the drone performs a sidekick $(i \neq j)$ or a loop $(i = j)$.

Table 2: Index sets, parameters, and decision and auxiliary variables.
Note that binary variables have a value of 1, if they are true, and 0 otherwise.

Table 2 describes index sets, parameters, and decision and auxiliary variables in detail.

4.3 MILP

Objective function

The objective is to minimize the makespan.

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

Definition of the makespan

τ^* is defined in Constraint 2 as the truck's accumulated travel time $\tau_{i,0}$ when arriving at the depot plus the truck's waiting time for returning drones at the depot, as these are not included in $\tau_{i,0}$ when returning to the depot. This objective function results in runtime issues, as a solver has a severe problem finding a lower bound [3]. To overcome this issue, we add another definition of the makespan, which helps to find strong lower bounds, particularly in initial iterations (Constraint 3). Note that either Constraint 2 or Constraint 3 can be chosen. The solver has the best runtime if both constraints are chosen.

$$\tau^* = \sum_{i \in \mathcal{I}_0} \left(\tau_{i,0} + \phi_{i,0}\right) \tag{2}$$

$$\tau^* = \sum_{i,j \in \mathcal{I}_0} \left(t^T_{i,j} \cdot x_{i,j} + \phi_{i,j}\right) \tag{3}$$

Set partitioning problem

Constraints 4 ensure that the problem is a set partitioning problem. Constraints 5 determine the customers that are served by truck and Constraints 6 the customers that are served by drones.

$$z^T_i + \left\{ \begin{array}{ll} z^D_i & , i \in \mathcal{I}_D \\ 0 & , else \end{array} \right\} = 1 \qquad \forall i \in \mathcal{I} \tag{4}$$

$$\sum_{i \in \mathcal{I}_0} x_{i,j} = z^T_j \qquad \forall j \in \mathcal{I} \tag{5}$$

$$\sum_{i \in \mathcal{I}_0} \overrightarrow{y}_{i,c} = \sum_{j \in \mathcal{I}_0} \overleftarrow{y}_{c,j} = z^D_c \quad \forall c \in \mathcal{I}_D \tag{6}$$

Truck related constraints

Constraints 7 conserve the truck's flow. Subtours are eliminated by Constraints 8 [21]. M^{big}_1 can be set as the number of nodes $|\mathcal{I}_0|$. The truck is launched at maximum once

from the depot (Constraint 9). Note that a solution might be that all customers are served by drones that perform a loop at the depot and thus the truck does not leave the depot. Constraints 10 ensure that the truck only travels to a different node.

$$\sum_{i \in \mathcal{I}_0} x_{j,i} - \sum_{i \in \mathcal{I}_0} x_{i,j} = 0 \qquad \forall j \in \mathcal{I}_0 \tag{7}$$

$$u_i + 1 \leq u_j + M_1^{big} \cdot (1 - x_{i,j}) \quad \forall i, j \in \mathcal{I} \tag{8}$$

$$\sum_{j \in \mathcal{I}_0} x_{0,j} \leq 1 \tag{9}$$

$$\sum_{i \in \mathcal{I}_0} x_{i,i} = 0 \tag{10}$$

Drone related constraints

Constraint 11 defines the number of drones the truck carries starting from the depot. Constraints 12 are balance constraints during the tour to ensure that the number of drones on the truck that leaves a node is equal to the number of drones that were previously on the truck minus launching plus landing drones. The number of drones carried by truck in its complete tour is limited in Constraints 13.

Drones may only launch and land on nodes, the truck has visited (Constraints 14). Drone flights must not exceed an endurance (Constraints 15). Additionally, the truck must pick up the drone within the considered endurance limit (Constraints 16, if launched at node $i \in \mathcal{I}$ and Constraints 17, if launched at the depot). M_2^{big} can be set as the truck's maximum possible tour length.

$$\sum_{j \in \mathcal{I}} \pi_{0,j} + \sum_{c \in \mathcal{I}_D} \left(\overrightarrow{y}_{0,c} - \lambda_{0,c} \right) = m \tag{11}$$

$$\sum_{j \in \mathcal{I}_0 : i \neq j} \pi_{i,j} = \sum_{k \in \mathcal{I}_0 : k \neq i} \pi_{k,i} + \sum_{c \in \mathcal{I}_D} \left(\overleftarrow{y}_{c,i} - \overrightarrow{y}_{i,c} \right) \qquad \forall i \in \mathcal{I} \tag{12}$$

$$\pi_{i,j} \leq m \cdot x_{i,j} \qquad \forall i, j \in \mathcal{I}_0 : i \neq j \tag{13}$$

$$\overrightarrow{y}_{j,c} + \overleftarrow{y}_{c,j} \leq 2 \cdot \sum_{i \in \mathcal{I}_0} x_{i,j} \qquad \forall c \in \mathcal{I}_D, j \in \mathcal{I} \tag{14}$$

$$t_{i,c}^D \cdot \overrightarrow{y}_{i,c} + t_{c,j}^D \cdot \overleftarrow{y}_{c,j} \leq e \qquad \forall i, j \in \mathcal{I}_0, c \in \mathcal{I}_D \tag{15}$$

$$\sum_{f \in \mathcal{I}_0} \left(\tau_{f,j} - \tau_{f,i} \right) - M_2^{big} \cdot (2 - \overrightarrow{y}_{i,c} - \overleftarrow{y}_{c,j}) \leq e \quad \forall i \in \mathcal{I}, j \in \mathcal{I}_0, c \in \mathcal{I}_D \tag{16}$$

$$\sum_{f \in \mathcal{I}_0} \tau_{f,j} - M_2^{big} \cdot (2 - \overrightarrow{y}_{0,c} - \overleftarrow{y}_{c,j}) \leq e \qquad \forall j \in \mathcal{I}_0, c \in \mathcal{I}_D \tag{17}$$

Setting up the truck's travel time

The truck's accumulated travel times at the beginning of the tour are defined in Constraints 18. Constraints 19 defines the lower and Constraints 20 the upper bound for the accumulated travel times during the truck's tour. These are equal if the truck travels from node i to node j. The accumulated travel time is zero, if the truck does not travel from i to j (Constraints 21).

$$\tau_{0,j} = t^T_{0,j} \cdot x_{0,j} \qquad \forall j \in \mathcal{I} \qquad (18)$$

$$\tau_{i,j} \geq t^T_{i,j} \cdot x_{i,j} + \sum_{f \in \mathcal{I}_0} \Big(\tau_{f,i} + \phi_{f,i} \Big) - M_2^{big} \cdot (1 - x_{i,j}) \quad \forall i \in \mathcal{I}, j \in \mathcal{I}_0 \qquad (19)$$

$$\tau_{i,j} \leq t^T_{i,j} \cdot x_{i,j} + \sum_{f \in \mathcal{I}_0} \Big(\tau_{f,i} + \phi_{f,i} \Big) + M_2^{big} \cdot (1 - x_{i,j}) \quad \forall i \in \mathcal{I}, j \in \mathcal{I}_0 \qquad (20)$$

$$\tau_{i,j} \leq M_2^{big} \cdot x_{i,j} \qquad \forall i, j \in \mathcal{I}_0 \qquad (21)$$

Sidekicks

Constraints 22 (at the beginning of the tour) and Constraints 23 (during the tour) ensure that drones do not return to a node that was visited by the truck previously on its tour. M_3^{big} can be set as twice the reciprocal value of the truck's shortest travel time $t^T_{i,j}$ between two nodes. M_4^{big} is dependent on the values of M_2^{big} and M_3^{big} and has to be at least $M_2^{big} \cdot M_3^{big}$. There are at maximum m sidekicks per node (Constraints 24). The lower bound of the truck's waiting time arriving at node j is presented in Constraints 25 at the beginning of the tour and in Constraints 26 during the tour. The inequality exists since the truck waits at a node until the last drone has returned.

$$\overrightarrow{y}_{0,c} + \overleftarrow{y}_{c,j} \leq M_3^{big} \cdot \sum_{f \in \mathcal{I}_0} \tau_{f,j} + M_4^{big} \cdot (2 - \overrightarrow{y}_{0,c} - \overleftarrow{y}_{c,j}) \qquad \forall j \in \mathcal{I}, c \in \mathcal{I}_D \qquad (22)$$

$$\overrightarrow{y}_{i,c} + \overleftarrow{y}_{c,j} \leq M_3^{big} \cdot \sum_{f \in \mathcal{I}_0} \Big(\tau_{f,j} - \tau_{f,i} \Big) + M_4^{big} \cdot (2 - \overrightarrow{y}_{i,c} - \overleftarrow{y}_{c,j}) \qquad \forall i \in \mathcal{I}, j \in \mathcal{I}_0, i \neq j, c \in \mathcal{I}_D \qquad (23)$$

$$\sum_{c \in \mathcal{I}_D} \big(\overrightarrow{y}_{i,c} - \lambda_{i,c} \big) \leq m \qquad \forall i \in \mathcal{I}_0 \qquad (24)$$

$$\phi_{0,j} \geq t^D_{0,c} \cdot \overrightarrow{y}_{0,c} + t^D_{c,j} \cdot \overleftarrow{y}_{c,j}$$

$$- \sum_{f \in \mathcal{I}_0} \tau_{f,j} - M_2^{big} \cdot (2 - \overrightarrow{y}_{0,c} - \overleftarrow{y}_{c,j}) \qquad \forall j \in \mathcal{I}, c \in \mathcal{I}_D \tag{25}$$

$$\begin{aligned} \phi_{i,j} \geq\ & t_{i,c}^D \cdot \overrightarrow{y}_{i,c} + t_{c,j}^D \cdot \overleftarrow{y}_{c,j} \\ & - \sum_{f \in \mathcal{I}_0} \left(\tau_{f,j} - \tau_{f,i} - \phi_{f,i}\right) \\ & - M_2^{big} \cdot (2 - \overrightarrow{y}_{i,c} - \overleftarrow{y}_{c,j}) \qquad \forall i \in \mathcal{I}, j \in \mathcal{I}_0, c \in \mathcal{I}_D : i \neq j \end{aligned} \tag{26}$$

Loops

The truck waits at node i until all drones planned to return have returned to node i (Constraints 27). Constraints 28 ensure that drones are only launched to perform loops if there is a sufficient number of drones on the truck at node j. Constraints 29 set up $\lambda_{i,c}$, if a customer is supplied in a loop. If a drone starts from node i to serve customer c and returns to node i, $\lambda_{i,c}$ must be set to 1 (left side). If $\lambda_{i,c}$ is set to 1, a drone must start from and return to node i (right side).

$$\phi_{i,i} \geq 2 \cdot t_{i,c}^D \cdot \lambda_{i,c} \qquad \forall i \in \mathcal{I}_0, c \in \mathcal{I}_D \tag{27}$$

$$\sum_{i \in \mathcal{I}_0} \pi_{i,j} + \sum_{c \in \mathcal{I}_D} \left(\overleftarrow{y}_{c,j} - 2 \cdot \lambda_{j,c}\right) \geq 0 \qquad \forall j \in \mathcal{I}_0 \tag{28}$$

$$\frac{\overrightarrow{y}_{i,c} + \overleftarrow{y}_{c,i}}{2} \geq \lambda_{i,c} \geq \overrightarrow{y}_{i,c} + \overleftarrow{y}_{c,i} - 1 \quad \forall i \in \mathcal{I}_0, c \in \mathcal{I}_D \tag{29}$$

Definition of variables

Last, decision and auxiliary variables are defined.

$$x_{i,j}, \overrightarrow{y}_{i,c}, \overleftarrow{y}_{c,j}, \lambda_{i,c} \in \{0,1\}, \pi_{i,j} \in \mathbb{N}, \phi_{i,j}, \tau_{i,j}, \tau^* \in \mathbb{R}^+ \quad \forall i,j \in \mathcal{I}_0, c \in \mathcal{I}_D \tag{30}$$

$$z_i^T, z_c^D \in \{0,1\}, u_i \in \mathbb{R} \qquad \forall i \in \mathcal{I}, c \in \mathcal{I}_D \tag{31}$$

4.4 Adjustment for preventing drone loops

Variable structures are generally chosen such that drones might perform loops. Thus, to prevent drone loops the variable $\lambda_{i,c}$ needs to be removed from the Constraints 11, 24, and 30, and the Constraints 32 need to replace Constraints 27, 28, and 29. These constraints prevent drones to launch and return to the same node.

$$\overrightarrow{y}_{i,c} + \overleftarrow{y}_{c,i} \leq 1 \quad \forall i \in \mathcal{I}_0, c \in \mathcal{I}_D \tag{32}$$

5 Numerical experiments

In this section, we describe the instances for which we present new benchmark solutions (Section 5.1). Next, we analyze the runtime solving the MILP and compare the runtime of our MILP to MILPs for the TSPD from the literature (Section 5.2). Last, we show the impact of considering drone loops on minimizing the makespan and the solver's runtime for different drone speeds and endurance limits (Section 5.3).

The MILP is implemented in OPL, solved using CPLEX v12.10., and executed on an AMD Ryzen 9 3950X with 32 GB of RAM (single thread). Each instance has a runtime limit of 3600 seconds.

5.1 Benchmark instances

In the following, we describe two published instance sets. Detailed solutions of each individual instance for benchmark purposes can be found in Appendix A for the instances of [1] and in Appendix B for the instances of [5]. We present results for these instances, if one ($m = 1$), three ($m = 3$), and a theoretically unlimited number of drones ($m = \infty$) are considered.

Instances of [1]

[1] published twelve instances with eleven nodes (ten customers and the depot) whose locations are uniformly distributed in an 8×8 mile region. The authors consider two endurance limits of 20 and 40 minutes and three different drone-to-truck speed ratios $\alpha \in \{0.6, 1.0, 1.4\}$. Thus, there are 72 instances for which we present results for benchmark purposes. Within the instances, the truck follows a Manhattan distance, while the drone flies the Euclidean path. 80 - 90% of all customers can be served via drones.

Instances of [5]

[5] published instances with 20 and 50 customers that are uniformly distributed with coordinates from 0 to 100. From these instances, we consider 16, 20, 24, 28, and 32 nodes as in [20]. These are drawn by taking the first 16, 20, 24, 28, and 32 nodes from the instance with the next-largest number of nodes, i.e., 16 and 20 nodes from instances with 20 nodes, and 24, 28, and 32 nodes from instances with 50 nodes. The endurance limit is set to 30, and drones have the same speed as the truck. Both drones and the truck travel the Euclidean path. All customers can be served by drones.

5.2 Runtime analysis

For the runtime analysis, the instances are aggregated by the number of nodes $|\mathcal{I}_0|$ and the endurance e. Table 3 presents for each number of drones ($m = 1$, $m = 3$, and $m = \infty$) the number of instances solved to optimality, the average runtime needed in seconds and the average optimality gap, if the solver could not find the optimal solution within 3600 seconds.

			$m = 1$			$m = 3$			$m = \infty$		
	$\|\mathcal{I}_0\|$	e	Opt	CPU [s]	Gap	Opt	CPU [s]	Gap	Opt	CPU [s]	Gap
[1]	11	20	36/36	104	0%	36/36	6	0%	36/36	4	0%
	11	40	33/36	880	1%	36/36	16	0%	36/36	4	0%
	Summary		69/72	492		72/72	11		72/72	4	
[5]	16	30	10/10	11	0%	10/10	6	0%	10/10	11	0%
	20	30	10/10	247	0%	10/10	299	0%	10/10	319	0%
	24	30	8/10	1327	1%	8/10	1319	1%	8/10	1387	1%
	28	30	1/10	3294	4%	1/10	3305	4%	2/10	3237	4%
	32	30	0/10	3600	12%	0/10	3600	10%	0/10	3600	11%
	Summary		29/50	1696		29/50	1706		30/50	1711	

Table 3: Aggregated results for instances of [1] and [5].

Findings: For the instances of [1], we could find all optimal solutions except three if one drone is considered. In particular, it is noticeable for all these instances that the optimal solution is found much faster when multiple drones are considered. For the instances of [5], we could find all optimal solutions for instances with 16 and 20 nodes, eight of ten optimal solutions for instances with 24 nodes, and up to two optimal solutions for instances with 28 nodes when considering one, three, and an infinite number of drones. In contrast to the instances of [1], there is no significant change in runtime if multiple drones are considered.

Runtime comparison to the literature

The two-indexed MILP formulation of the TSPmD presented in this paper can solve instances with more nodes than the literature (see also Table 1). So, [1] could not solve one of the 72 instances with eleven nodes to optimality. Considering the instances of [5], [20] could find optimal solutions for up to two out of the ten instances with 24 nodes. However, it is difficult to compare the MILPs based on results in the literature, as they have slightly different assumptions, a different solver (or version of solver) is used, and they were run on a computer with different RAM. Thus, to show the efficiency of our MILP formulation in comparison to the literature, we implemented

the MILPs of [19] and [20] in OPL. [19] have an arc-based formulation and [20] a node-based formulation of the TSPD, where additional variables track the drone's tour, even if it is carried on the truck. Both MILP formulations are chosen for comparison, as they are efficient two-indexed MILP formulations for the TSPD and include loops while minimizing the makespan.

			MILP of [19]			MILP of [20]			This paper		
	$\lvert\mathcal{I}_0\rvert$	e	Opt	CPU [s]	Gap	Opt	CPU [s]	Gap	Opt	CPU [s]	Gap
[1]	11	20	24/24	238	0%	24/24	93	0%	24/24	148	0%
	11	40	24/24	76	0%	23/24	399	1%	21/24	870	2%
	Summary		48/48	157		47/48	246		45/48	509	
[5]	16	30	0/10	3600	19%	10/10	27	0%	10/10	11	0%
	20	30	0/10	3600	28%	10/10	576	0%	10/10	247	0%
	24	30	0/10	3600	30%	5/10	2165	4%	8/10	1327	1%
	28	30	0/10	3600	36%	0/10	3600	9%	1/10	3294	4%
	32	30	0/10	3600	42%	0/10	3600	17%	0/10	3600	12%
	Summary		0/50	3600		25/50	1994		29/50	1696	

Table 4: Aggregated results running the MILPs of [19], [20], and from this paper for the instances of [1] and [5]. All MILP formulations represent a TSPD (single drone) with the same objective and assumptions.

We tested both MILP formulations for the instances of [1] and [5]. However, only a subset of the instances of [1] is suitable, because both MILP formulations require a drone speed that is at least the same as the truck's speed. Thus, only 48 of the 72 instances, but all of [5] are considered. The MILP of [20] is additionally adjusted by endurance constraints that limit the waiting times of drones (similar to Constraints 16 and 17). This is already considered by [19]. Note that for implementing the MILPs, we also consider the enhanced modelling assumptions to accelerate the MILP solver presented in the papers (similar to Constraints 3), but no lower or upper bounds or any pre-processing steps as in [20]. Table 4 presents the aggregated results solving the three MILPs. The column gap shows the gap to the own lower bound of the respective model. Note that all MILP formulations represent the same problem setting of the TSPD with one drone that can perform both sidekicks and loops.

Findings: The MILP of [19] works best for the instances of [1], as all 48 instances could be solved optimally with the lowest runtime on average, but, on the other hand, not one instance with 16 or more nodes could be solved. Running the MILP of [20], optimal solutions for all ten instances with 16 and 20 nodes and five out of ten instances with 24 nodes could be found. In contrast, our MILP formulation could additionally be solved to optimality for three more instances with 24 nodes and one

instance with 28 nodes. Moreover, the average runtime and the optimality gap are significantly lower. Thus, the MILP presented in this paper outperforms the literature for larger instances. Additionally, it should be noted that our MILP formulation is a more general formulation as drones might have lower speeds than trucks and multiple drones can be considered without any increase in runtime.

5.3 Impact of drone loops

Drone loops as an additional routing option for drones might reduce the makespan τ^*, but on the other hand, considering loops might increase the solver's runtime on average, which in turn can be problematic, as truck-and-drone tandems are difficult to solve even for heuristic solution approaches [e.g., 16]. So, [4] already found that drones perform loops in the solutions, and [17] found that loops might reduce total routing costs. However, in the solutions presented by [7] drones do not perform a single loop. Thus, in the following, we give detailed insights on the makespan reduction and the runtime increase when allowing loops compared to prohibiting loops for the instances of [1] and [5] with a varying number of drones m, endurance limits e, and truck-to-drone speed ratios α. For this, we only compare instances solved optimally for the case with and without drone loops.

Instances of [1]

Table 5 presents the reduction of the makespan τ^* and the increase in runtime of the MILP solver in percent if drones can perform loops. For this, the makespan considering loops is compared to the makespan, if loops are forbidden. We generate results for three different numbers of drones ($m = 1$, $m = 3$, $m = \infty$), two endurance limits ($e = 20$, $e = 40$), and five speed factors ($\alpha = 0.6$, $\alpha = 1.0$, $\alpha = 1.4$, $\alpha = 2.0$, $\alpha = 2.8$). These are two additional speed factors ($\alpha = 2.0$, $\alpha = 2.8$) compared to the data considered in [1] as the speed has a significant impact on the number of performed loops [4]. Each entry presents the average τ^* reduction or runtime increase of up to twelve instances. Note that there might be less than twelve instances considered if they are not solved to optimality. A negative entry in column "Runtime increase [%]" means there is a decrease in runtime.

Findings. Considering instances of [1] with eleven nodes, loops can reduce the makespan τ^* by up to 1.9% on average. For these instances it can be stated that there are only improvements in the objective considering loops if drones travel faster than trucks ($\alpha > 1$). Note that in these instances, the drone follows a Euclidean path contrary to the truck that travels the Manhattan distance. The improvements

considering loops occur especially when few drones are considered. However, slow-traveling drones are more likely to perform loops when the truck is equipped with multiple drones. If the truck is equipped with an unlimited number of drones, there are nearly no reductions of τ^*. Moreover, endurance does not have an impact on performed loops. The runtime, on the other hand, increases significantly by up to 113%, considering loops as an additional routing option. The runtime increase is significantly higher if the truck is equipped with only one drone.

			$m = 1$		$m = 3$		$m = \infty$	
$\lvert\mathcal{I}_0\rvert$	e	α	τ^* reduction [%]	Runtime increase [%]	τ^* reduction [%]	Runtime increase [%]	τ^* reduction [%]	Runtime increase [%]
11	20	0.6	-	62.2	-	-2.2	-	7.3
		1.0	-	23.4	-	33.5	-	-11.7
		1.4	-	-4.8	0.1	16.5	0.6	18.8
		2.0	0.6	79.1	0.1	9.4	-	3.8
		2.8	1.7	57.4	0.5	4.5	-	5.9
	40	0.6	-	48.7	-	34.4	-	37.8
		1.0	-	23.2	-	14.4	-	10.7
		1.4	-	29.1	0.3	-6.0	-	7.2
		2.0	0.6	113.3	0.1	46.2	-	-0.4
		2.8	1.9	5.0	0.5	32.0	-	1.6
Average			0.5	43.7	0.2	18.3	0.1	8.1

Table 5: Makespan reduction and the solver's runtime increase, if drones can perform loops, for the instances of [1].

Instances of [5]

Similar to the previous table, Table 6 presents the results for 16 considered nodes for two different endurance limits ($e = 30$, $e = 60$) and three speed factors ($\alpha = 1$, $\alpha = 2$, $\alpha = 3$). The additional endurance limit and speed factors are similar as in [20]. An analysis of instances with 20 or more nodes is not meaningful, as there are only a few instances solved to optimality for both the consideration and the ban of loops. Each entry presents the average τ^* reduction or runtime increase of up to 10 instances.

Findings Both the makespan reduction and runtime increase are on average larger for these instances compared to the instances of [1]. So, allowing drones to perform loops may reduce τ^* by up to 2.7%. Again, the endurance e has no significant impact on τ^*, but a larger drone-to-truck speed ratio α has. Contrary to the results for the instances of [1], a larger number of drones now results in a larger reduction of τ^*. Additionally, if drones and trucks have the same speed ($\alpha = 1$), there is even an instance where one loop is performed in the optimal solution resulting in slight

reductions of τ^*. Note that drones and trucks follow both the Euclidean path. The runtime increases significantly by up to 194.4% and is especially high if one drone is considered. However, considering an infinite number of drones, there is no significant runtime increase, but the impact on τ^* is high.

			$m = 1$		$m = 3$		$m = \infty$	
$\lvert\mathcal{I}_0\rvert$	e	α	τ^* reduction [%]	Runtime increase [%]	τ^* reduction [%]	Runtime increase [%]	τ^* reduction [%]	Runtime increase [%]
16	30	1	-	57.8	0.1	41.9	0.1	44.5
		2	1.0	63.8	1.5	-14.4	2.0	-24.6
		3	0.3	194.4	1.7	6.3	1.3	-11.4
	60	1	-	99.1	-	44.3	-	26.1
		2	./.	./.	-	19.2	1.7	9.5
		3	./.	./.	-	-2.7	2.7	-33.0
Average			0.3	103.8	0.6	15.8	1.3	1.9

Table 6: Makespan reduction and the solver's runtime increase, if drones can perform loops, for the instances of [5]. Entry ./. indicates that no instance was solved to optimality for both the consideration and the ban of loops.

6 Conclusion

This paper considers the TSPmD with drones performing sidekicks and loops while minimizing the makespan. We introduce the problem as a two-indexed MILP formulation that outperforms MILP formulations for the single drone case from the literature. We present new benchmark solutions for instances of [1] and [5], which we could solve to optimality for up to 28 nodes. Additionally, we generate managerial insights on the impact of allowing drone loops on makespan minimization.

We can find that makespan savings of up to 2.7% can be achieved by allowing drones to perform loops. The savings are essentially dependent on the drone-to-truck speed ratio, but not on the endurance. It follows that there are no drone loops if the drones' speeds are too slow. On the other hand, drone loops make the problem significantly harder to solve, even if in the optimal solution drones do not perform loops. Therefore, for the TSPmD with any number of drones, drone loops should only be considered if drones have at least the truck's speed.

Compact and efficient MILP formulations for truck-and-drone tandems are good competitors to exact algorithms. However, an efficient optimal solution approach is still required for the TSPmD that can solve instances with more than 28 nodes to optimality.

Appendix A: Benchmark results for instances of [1]

The following two tables present results for the instances of [1] for the endurance of $e = 20$ and $e = 40$. The instance names are similar to [7].

	$m = 1$			$m = 3$			$m = \infty$		
Instance	τ^*	CPU [s]	Gap	τ^*	CPU [s]	Gap	τ^*	CPU [s]	Gap
20140810T123437v1	54.3926	12	-	51.3825	2	-	51.3825	2	-
20140810T123437v2	51.6111	12	-	51.6111	7	-	51.6111	6	-
20140810T123437v3	54.0684	15	-	52.8225	10	-	52.8225	8	-
20140810T123437v4	66.8684	9	-	65.6225	11	-	65.6225	8	-
20140810T123437v5	45.3353	326	-	32.0655	10	-	24.4390	2	-
20140810T123437v6	43.9153	210	-	28.9400	7	-	24.0264	1	-
20140810T123437v7	46.5813	18	-	43.2069	9	-	39.8664	7	-
20140810T123437v8	59.3813	34	-	57.7700	12	-	53.1454	7	-
20140810T123437v9	39.2035	821	-	24.9912	6	-	21.8536	1	-
20140810T123437v10	36.9077	343	-	25.7710	11	-	21.9720	2	-
20140810T123437v11	39.3002	107	-	33.7779	7	-	32.3077	2	-
20140810T123437v12	51.5645	61	-	47.7507	8	-	45.1077	3	-
20140810T123440v1	46.4304	52	-	43.8455	6	-	43.8455	6	-
20140810T123440v2	49.3737	56	-	46.2455	11	-	46.2455	14	-
20140810T123440v3	53.6616	13	-	51.6277	8	-	51.6277	7	-
20140810T123440v4	66.4616	14	-	64.4277	6	-	64.4277	10	-
20140810T123440v5	39.7498	115	-	32.2263	3	-	31.6066	1	-
20140810T123440v6	40.3790	91	-	34.0066	2	-	34.0066	1	-
20140810T123440v7	43.3126	21	-	40.7304	2	-	40.7304	2	-
20140810T123440v8	55.7960	20	-	53.5304	2	-	53.5304	1	-
20140810T123440v9	35.5331	14	-	31.6066	1	-	31.6066	0	-
20140810T123440v10	36.0764	10	-	34.0066	1	-	34.0066	1	-
20140810T123440v11	40.7304	4	-	40.7304	1	-	40.7304	1	-
20140810T123440v12	53.5304	7	-	53.5304	1	-	53.5304	1	-
20140810T123443v1	69.5865	8	-	69.5865	4	-	69.5865	7	-
20140810T123443v2	72.0639	5	-	71.7473	3	-	71.5839	3	-
20140810T123443v3	76.1447	4	-	76.0673	1	-	75.9039	1	-
20140810T123443v4	89.1839	2	-	88.8673	2	-	88.7039	2	-
20140810T123443v5	54.7521	236	-	42.2634	3	-	39.7845	3	-
20140810T123443v6	55.1268	112	-	45.0943	11	-	43.9992	5	-
20140810T123443v7	62.2491	36	-	60.8333	5	-	60.8333	5	-
20140810T123443v8	80.0971	34	-	76.3401	27	-	73.8724	5	-
20140810T123443v9	41.9314	749	-	27.6100	1	-	24.5760	1	-
20140810T123443v10	42.9348	153	-	30.9760	1	-	30.9760	1	-
20140810T123443v11	51.4056	17	-	43.7760	3	-	43.7760	2	-
20140810T123443v12	62.6175	9	-	56.5760	1	-	56.5760	1	-

Table 1: Results for instances of [1] with an endurance of 20.

	$m = 1$			$m = 3$			$m = \infty$		
Instance	τ^*	CPU [s]	Gap	τ^*	CPU [s]	Gap	τ^*	CPU [s]	Gap
20140810T123437v1	49.1189	3369	-	35.5186	23	-	33.8109	8	-
20140810T123437v2	46.3113	442	-	35.3759	29	-	32.6510	7	-
20140810T123437v3	52.6868	248	-	49.2858	87	-	40.9090	28	-
20140810T123437v4	65.4868	272	-	62.5748	112	-	54.5333	56	-
20140810T123437v5	42.8354	2663	-	28.4013	31	-	24.4390	2	-
20140810T123437v6	41.6015	1344	-	28.4688	10	-	24.0264	3	-
20140810T123437v7	43.3913	173	-	36.4363	13	-	35.8290	8	-
20140810T123437v8	56.1913	267	-	50.5161	13	-	48.6273	7	-
20140810T123437v9	37.8082	3600	12.2%	24.9912	12	-	21.8536	1	-
20140810T123437v10	36.9077	635	-	25.7710	15	-	21.9720	2	-
20140810T123437v11	39.3002	312	-	29.3975	19	-	26.4022	1	-
20140810T123437v12	51.2536	115	-	40.7507	7	-	37.3724	1	-
20140810T123440v1	45.1029	686	-	35.5331	29	-	32.3895	1	-
20140810T123440v2	44.4461	866	-	37.2360	60	-	34.0066	1	-
20140810T123440v3	52.3083	421	-	45.3532	16	-	45.3532	7	-
20140810T123440v4	66.3969	581	-	58.9284	14	-	58.1532	7	-
20140810T123440v5	39.7498	310	-	32.2263	3	-	31.6066	0	-
20140810T123440v6	39.6581	207	-	34.0066	1	-	34.0066	1	-
20140810T123440v7	43.3126	58	-	40.7304	1	-	40.7304	1	-
20140810T123440v8	55.7960	40	-	53.5304	1	-	53.5304	1	-
20140810T123440v9	35.5331	13	-	31.6066	1	-	31.6066	1	-
20140810T123440v10	36.0764	9	-	34.0066	1	-	34.0066	1	-
20140810T123440v11	40.7304	5	-	40.7304	1	-	40.7304	1	-
20140810T123440v12	53.5304	5	-	53.5304	1	-	53.5304	1	-
20140810T123443v1	54.0129	1121	-	37.5843	7	-	37.4695	2	-
20140810T123443v2	56.5729	2149	-	38.5950	11	-	35.1953	1	-
20140810T123443v3	66.1746	298	-	54.6474	11	-	47.4513	1	-
20140810T123443v4	81.4603	350	-	69.2538	13	-	65.6295	3	-
20140810T123443v5	49.0759	3600	25.0%	31.2183	19	-	24.2916	0	-
20140810T123443v6	48.4864	2640	-	35.0597	12	-	30.0160	0	-
20140810T123443v7	56.8793	777	-	44.3748	3	-	42.8160	0	-
20140810T123443v8	68.3988	225	-	56.5760	1	-	55.6160	0	-
20140810T123443v9	41.9314	3600	6.3%	27.6100	3	-	23.6160	0	-
20140810T123443v10	42.9348	240	-	30.9760	1	-	30.0160	0	-
20140810T123443v11	51.3950	38	-	42.8160	0	-	42.8160	0	-
20140810T123443v12	62.6175	7	-	55.6160	0	-	55.6160	0	-

Table 2: Results for instances of [1] with an endurance of 40.

Appendix B: Benchmark results for instances of [5]

The following five tables present the results for the instances of [5]. The instance names are similar to [20] and the results are split up for each considered number of nodes.

$\lvert\mathcal{I}_0\rvert = 16$									
	$m = 1$			$m = 3$			$m = \infty$		
Instance	τ^*	CPU [s]	Gap	τ^*	CPU [s]	Gap	τ^*	CPU [s]	Gap
Uniform-61-n20	347.0919	74	-	346.9223	27	-	346.9223	75	-
Uniform-62-n20	353.7019	2	-	353.7019	2	-	353.7019	4	-
Uniform-63-n20	372.9289	2	-	370.1349	2	-	370.1349	1	-
Uniform-64-n20	357.9113	10	-	357.9113	11	-	357.9113	9	-
Uniform-65-n20	368.6511	5	-	368.6511	4	-	368.6511	4	-
Uniform-66-n20	426.5167	2	-	426.5167	2	-	426.5167	2	-
Uniform-67-n20	372.7803	2	-	372.7803	1	-	372.7803	1	-
Uniform-68-n20	423.3365	2	-	423.3365	2	-	423.3365	3	-
Uniform-69-n20	363.4143	7	-	363.4143	6	-	363.4143	5	-
Uniform-70-n20	410.1439	8	-	410.1439	4	-	410.1439	8	-

Table 3: Results for instances of [5] with 16 nodes.

$\lvert\mathcal{I}_0\rvert = 20$									
	$m = 1$			$m = 3$			$m = \infty$		
Instance	τ^*	CPU [s]	Gap	τ^*	CPU [s]	Gap	τ^*	CPU [s]	Gap
Uniform-61-n20	351.5212	588	-	351.4622	1526	-	351.4622	399	-
Uniform-62-n20	374.1195	431	-	374.1195	481	-	374.1195	1308	-
Uniform-63-n20	394.5000	34	-	391.7060	31	-	391.7060	78	-
Uniform-64-n20	368.7314	575	-	368.7314	506	-	368.7314	581	-
Uniform-65-n20	390.5601	224	-	381.0652	52	-	381.0652	67	-
Uniform-66-n20	436.2728	166	-	435.1632	62	-	433.3361	101	-
Uniform-67-n20	391.4744	46	-	391.4744	34	-	391.4744	53	-
Uniform-68-n20	435.6068	25	-	435.6068	29	-	435.6068	36	-
Uniform-69-n20	380.4329	336	-	380.4329	181	-	380.4329	525	-
Uniform-70-n20	422.6549	42	-	422.6549	84	-	422.6549	42	-

Table 4: Results for instances of [5] with 20 nodes.

$\lvert\mathcal{I}_0\rvert = 24$									
	$m = 1$			$m = 3$			$m = \infty$		
Instance	τ^*	CPU [s]	Gap	τ^*	CPU [s]	Gap	τ^*	CPU [s]	Gap
Uniform-71-n50	415.8466	1059	-	415.8466	1449	-	415.8466	1219	-
Uniform-72-n50	410.1562	28	-	410.1562	30	-	410.1562	29	-
Uniform-73-n50	394.5155	495	-	391.3396	324	-	391.3564	316	-

Continued on next page

Continued from previous page

Uniform-74-n50	467.6828	865	-	467.6828	502	-	467.6828	552	-
Uniform-75-n50	463.6015	1072	-	463.6015	863	-	463.6015	1742	-
Uniform-76-n50	394.2924	3600	2.7%	394.2924	3600	4.8%	394.2924	3600	2.8%
Uniform-77-n50	458.7309	565	-	452.9030	98	-	452.9030	106	-
Uniform-78-n50	412.8484	1428	-	410.8227	1263	-	410.8227	1217	-
Uniform-79-n50	412.1437	554	-	412.1437	1460	-	412.1437	1492	-
Uniform-80-n50	397.2048	3600	4.6%	391.7442	3600	2.8%	391.7442	3600	2.7%

Table 5: Results for instances of [5] with 24 nodes.

$\lvert\mathcal{I}_0\rvert = 28$									
	$m = 1$			$m = 3$			$m = \infty$		
Instance	τ^*	CPU [s]	Gap	τ^*	CPU [s]	Gap	τ^*	CPU [s]	Gap
Uniform-71-n50	446.9016	3600	6.7%	441.6016	3600	4.8%	441.6016	3600	3.2%
Uniform-72-n50	449.6584	541	-	449.6584	645	-	449.6584	412	-
Uniform-73-n50	449.0387	3600	2.6%	445.8796	3600	2.6%	445.8628	3600	1.2%
Uniform-74-n50	470.8468	3600	1.0%	470.8405	3600	1.3%	470.8468	3600	1.7%
Uniform-75-n50	473.0050	3600	4.0%	473.0050	3600	5.9%	473.0050	3600	6.8%
Uniform-76-n50	407.7456	3600	9.2%	407.4875	3600	8.8%	407.7456	3600	9.5%
Uniform-77-n50	485.5172	3600	2.6%	480.5688	3600	1.0%	480.5687	3159	-
Uniform-78-n50	462.1742	3600	1.5%	459.4857	3600	1.2%	459.4857	3600	2.1%
Uniform-79-n50	447.7367	3600	7.9%	447.7367	3600	8.6%	447.7367	3600	10.3%
Uniform-80-n50	454.8248	3600	9.1%	450.3997	3600	8.4%	449.3641	3600	8.9%

Table 6: Results for instances of [5] with 28 nodes.

$\lvert\mathcal{I}_0\rvert = 32$									
	$m = 1$			$m = 3$			$m = \infty$		
Instance	τ^*	CPU [s]	Gap	τ^*	CPU [s]	Gap	τ^*	CPU [s]	Gap
Uniform-71-n50	483.6558	3600	7.5%	478.3559	3600	7.7%	478.3559	3600	6.5%
Uniform-72-n50	493.0942	3600	3.4%	493.0942	3600	3.6%	493.0942	3600	3.6%
Uniform-73-n50	497.0019	3600	7.5%	490.4027	3600	6.9%	490.4758	3600	7.3%
Uniform-74-n50	484.4209	3600	15.1%	479.3757	3600	14.1%	492.1320	3600	17.6%
Uniform-75-n50	502.6546	3600	12.5%	502.5212	3600	12.0%	504.6453	3600	12.6%
Uniform-76-n50	494.6067	3600	19.5%	474.7205	3600	15.6%	483.1039	3600	19.0%
Uniform-77-n50	521.4612	3600	7.2%	511.1739	3600	4.9%	511.1739	3600	4.7%
Uniform-78-n50	507.7490	3600	7.3%	504.8168	3600	5.3%	505.4795	3600	5.5%
Uniform-79-n50	483.3885	3600	16.1%	452.3963	3600	10.4%	459.7861	3600	12.1%
Uniform-80-n50	473.7993	3600	21.9%	445.4466	3600	16.9%	451.4449	3600	19.0%

Table 7: Results for instances of [5] with 32 nodes.

References

[1] 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) https://doi.org/10.1016/j.trc.2015.03.005

[2] Gartner, J.: JD.com's Drone Delivery Program Takes Flight in Rural China. World Wide Web (2016). Last access: May 17, 2022

[3] Seifried, K.: The traveling salesman problem with one truck and multiple drones. Available at SSRN 3389306 (2019)

[4] 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) https://doi.org/10.1016/j.trc.2019.06.016

[5] Bouman, P., Agatz, N., Schmidt, M.: Instances for the TSP with Drone (and some solutions) (v1.2). Zenodo (2018). Last access: February 14, 2023

[6] Rave, A., Fontaine, P., Kuhn, H.: Drone location and vehicle fleet planning with trucks and aerial drones. European Journal of Operational Research **308**(1), 113–130 (2023) https://doi.org/10.1016/j.ejor.2022.10.015

[7] Dell'Amico, M., Montemanni, R., Novellani, S.: Benchmark instances and optimal solutions for the traveling salesman problem with drone. arXiv preprint arXiv:2107.13275 (2021)

[8] 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) https://doi.org/10.1002/net.21818

[9] Boysen, N., Fedtke, S., Schwerdfeger, S.: Last-mile delivery concepts: a survey from an operational research perspective. OR Spectrum **43**(1), 1–58 (2021) https://doi.org/10.1007/s00291-020-00607-8

[10] Dell'Amico, M., Montemanni, R., Novellani, S.: Drone-assisted deliveries: New formulations for the flying sidekick traveling salesman problem. Optimization Letters **15**, 1617–1648 (2021) https://doi.org/10.1007/s11590-019-01492-z

[11] Freitas, J.C., Penna, P.H.V., Toffolo, T.A.M.: Exact and heuristic approaches to truck–drone delivery problems. EURO Journal on Transportation and Logistics **12**, 100094 (2023) https://doi.org/10.1016/j.ejtl.2022.100094

[12] Cavani, S., Iori, M., Roberti, R.: Exact methods for the traveling salesman problem with multiple drones. Transportation Research Part C: Emerging Technologies **130**, 103280 (2021) https://doi.org/10.1016/j.trc.2021.103280

[13] Murray, C.C., Raj, R.: The multiple flying sidekicks traveling salesman problem: Parcel delivery with multiple drones. Transportation Research Part C: Emerging Technologies **110**, 368–398 (2020) https://doi.org/10.1016/j.trc.2019.11.003

[14] Tamke, F., Buscher, U.: A branch-and-cut algorithm for the vehicle routing problem with drones. Transportation Research Part B: Methodological **144**, 174–203 (2021) https://doi.org/10.1016/j.trb.2020.11.011

[15] 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) https://doi.org/10.1016/j.trc.2017.11.015

[16] Sacramento, D., Pisinger, D., Røpke, S.: An adaptive large neighborhood search metaheuristic for the vehicle routing problem with drones. Transportation Research. Part C: Emerging Technologies **102**, 289–315 (2019) https://doi.org/10.1016/j.trc.2019.02.018

[17] Tiniç, G.O., Karasan, O.E., Kara, B.Y., Campbell, J.F., Ozel, A.: Exact solution approaches for the minimum total cost traveling salesman problem with multiple drones. Transportation Research Part B: Methodological **168**, 81–123 (2023) https://doi.org/10.1016/j.trb.2022.12.007

[18] Kitjacharoenchai, P., Ventresca, M., Moshref-Javadi, M., Lee, S., Tanchoco, J.M.A., Brunese, P.A.: Multiple traveling salesman problem with drones: Mathematical model and heuristic approach. Computers & Industrial Engineering **129**, 14–30 (2019) https://doi.org/10.1016/j.cie.2019.01.020

[19] Roberti, R., Ruthmair, M.: Exact methods for the traveling salesman problem with drone. Transportation Science **55**(2), 315–335 (2021) https://doi.org/10.1287/trsc.2020.1017

[20] El-Adle, A.M., Ghoniem, A., Haouari, M.: Parcel delivery by vehicle and drone. Journal of the Operational Research Society **72**(2), 398–416 (2021) https://doi.org/10.1080/01605682.2019.1671156

[21] Miller, C.E., Tucker, A.W., Zemlin, R.A.: Integer programming formulation of traveling salesman problem. Journal of the ACM **7**(4), 326–329 (1960) https://doi.org/10.1145/321043.321046