OR Spectrum
https://doi.org/10.1007/s00291-024-00785-9

ORIGINAL ARTICLE

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

Alexander Rave[1]

Received: 31 January 2024 / Accepted: 28 July 2024

Abstract

Aerial drone delivery has great potential to improve package delivery time, 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 multiple drones and releases them at advantageous places, i.e., the traveling salesman problem with multiple drones (TSPmD). We focus on a general version of this problem with makespan minimization, where the drones have two options after serving the customer: they can return to any node the truck visits at a later stage (sidekick) or return to the same node they were launched from (loop) — even at the depot. We introduce an efficient two-indexed mixed-integer linear program (MILP) for this TSPmD and show how to adapt the MILP to cover two problem variants, namely the multiple flying sidekick traveling salesman problem and the traveling salesman problem with drone. Our MILP formulation is an efficient formulation, as it outperforms eight state-of-the-art MILP formulations for these problem variants in nearly all larger instances. In a numerical study, we provide new optimal solutions with up to 28 nodes for benchmark purposes. Moreover, we analyze the impact of allowing drone loops on makespan minimization in general and at the depot. We find that 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

✉ Alexander Rave
ARave@ku.de

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

Published online: 29 August 2024

1 Introduction

Parcel delivery via aerial drones is much considered in academia (e.g., Murray and Chu 2015) and also in practice (e.g., Gartner 2016) as drones can increase the delivery speed (e.g., Murray and Raj 2020). The benefits of drones can even be increased when a truck releases the drones at advantageous places to serve customers. This problem was initially introduced by Murray and Chu (2015) under the name flying sidekick traveling salesman problem (FSTSP) and investigated by many publications (e.g., Ha et al. 2018). In the FSTSP, drones have to perform a sidekick, meaning they return to a node the truck visits at any later stage in its tour (see Fig. 1a). This is contrasted by drones performing loops, which means that they return to the same node they are launched from (see Fig. 1b). We call the problem when a drone launched from a mobile truck may perform both sidekicks and loops traveling salesman problem with drone (TSPD) and traveling salesman problem with multiple drones (TSPmD) if the truck is equipped with multiple drones. Only a few publications consider loops, thereby (Schermer et al. 2019) already found that drones perform loops in the final found routing if these are allowed. As truck-and-drone tandems are challenging to solve and only a few publications consider loops, the following question arises: How high is the impact of additionally considering loops on minimizing the total delivery time and, in contrast, how much effort (in terms of runtime increase) does it take?

In this paper, we introduce a general version of the TSPmD with sidekicks and loops while minimizing the makespan. The general version means that drone flights are not restricted by limitations that are not set physically (e.g., an endurance limit), i.e., in our version, drones can launch and return to any truck's node and even perform loops at the depot. These depot loops are, for example, not considered by Tiniç et al. (2023). This problem is not represented in the literature, and thus, the literature also lacks suitable benchmark instances. We fill this gap and introduce a compact and efficient two-indexed mixed-integer linear program (MILP) formulation of this TSPmD. Moreover, we show how to adjust this MILP to cover the well-known variants multiple flying sidekick traveling salesman problem (mFSTSP) and the TSPD. The MILP formulations for the TSPmD and the two problem variants are easily implemented and outperform in total eight state-of-the-art MILP formulations for these problem variants, i.e., they find the optimal solution faster. As a result, the MILP allows us to present optimal solutions with up to 28 nodes for all three problems for benchmark purposes. In a numerical study, we analyze the impact of allowing drone loops on minimizing the makespan and the runtime of a MILP solver. Moreover, we analyze the impact

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

Fig. 1 Illustration of drone routes

of loops performed at the depot and conduct an a-posteriori cost analysis to show the impact of loops on last-mile delivery costs.

This paper contributes to the literature as follows First, we introduce a general version of the TSPmD with makespan minimization. Second, we present an efficient MILP formulation with only two-indexed variables and adjust the MILP to cover the well-known problem variants mFSTSP and TSPD. We benchmark our MILP to four MILP formulations for the mFSTSP (three with three drones, one with a single drone) and four MILP formulations for the TSPD and find that our MILP formulation outperforms all of them in larger instances. Third, we present new optimal solutions for the TSPmD, the mFSTSP, and the TSPD for the instances of Murray and Chu (2015) and Bouman et al. (2018) for up to 28 nodes for benchmark purposes. Fourth, we present managerial insights on allowing drone loops in general and specifically performed at the depot - also including an a-posteriori cost analysis.

The structure of this paper is as follows. In Sect. 2, we describe the decisions and assumptions in detail. In Sect. 3, we present the relevant literature and delimit our problem setting and methodology to this literature. Next, we introduce the MILP in Sect. 4. We show how to adjust the MILP to cover the problem variants mFSTSP and TSPD in Sect. 5. In Sect. 6, we describe the instances for which we present benchmark results, analyze and benchmark the runtime, and give managerial insights on allowing drone loops. Last, in Sect. 7, we summarize the results and give a brief outlook.

2 Problem setting

We decide on the routing of one truck and its equipped multiple homogeneous drones that need to visit certain nodes, i.e., customers. Additionally, we decide if a customer is served by truck or 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 perform both sidekicks and loops - even at the depot. 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.
- The truck has an unlimited capacity. It follows that all customers can be served by truck in one tour, and also the number of drones carried by the truck is unrestricted. This unlimited number of drones is also considered by (e.g., Tiniç et al. 2023). We show how to limit the number of drones in Sect. 5.3, but at the expense of allowing a single loop per node.
- Multiple drones can launch or land at the same time (e.g., Rave et al. 2023), i.e., we consider unlimited launch and return platforms. This might be a limitation to the proposed MILP formulation.
- Drones have an endurance limit, i.e., maximum flight time per flight.
- Assumptions regarding customers (e.g., Murray and Chu 2015):

  - All customers must be served once by truck or by drone.
  - Some customers cannot be served by drones, but all by truck.
  - 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 must be picked up by the truck within a certain time (e.g., Murray and Chu 2015; Rave et al. 2023). 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. From a practical point of view, we assume that drones reduce their speed to a certain extent when arriving at a node before the truck, following the drone must be picked up within its endurance limit.
  - 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).
  - There may be multiple loops at each node.
  - Drones can perform loops at the depot. Without loss of generality, these are performed at the end of the truck's tour, which results in waiting times at the end of the tour (e.g., Dell'Amico et al. 2021). Note that in contrast to the literature, 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 provided that all customers can be served by drones and are in endurance range.
- There are no service, preparation, or rendezvous times for trucks and drones.
- We do not consider loading or battery change times for drones as this could be done while the truck is traveling (e.g., Rave et al. 2023). However, we show how to add manual battery change times to our MILP formulation.
- For benchmark tests and further analysis, we consider the following two problem variants:
  - mFSTSP: We limit the number of drones to m and prevent drones from performing loops.
  - TSPD: We limit the number of drones to one.

2.1 Exemplary routing plan

For a better understanding of the problem setting, Fig. 2 presents an example routing of the truck and its equipped 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

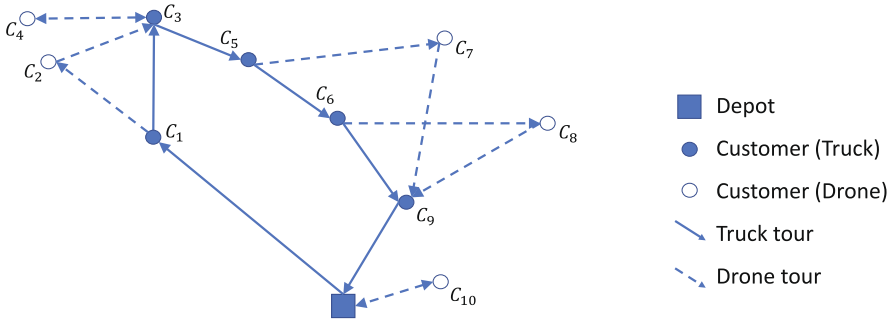


Fig. 2 Exemplary routing plan for ten customers served by a truck and its equipped drones

visits at a later stage as long as the flight time is within the endurance limit. At C_6, a 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.

3 Related literature

In this section, we differentiate this paper from the relevant literature regarding truck-and-drone tandems, where one or multiple trucks are equipped with at least one drone. We mainly focus on literature that includes a MILP formulation without complex assumptions that are not considered in our problem setting, e.g., battery consumption. First, we present the literature where drones only perform sidekicks, and second, we present the literature where drones can perform both sidekicks and loops. Last, we present further variants of truck-and-drone tandems. We recommend the paper of Otto et al. (2018) for an extensive review of drone operations and Boysen et al. (2021) for a more recent literature review.

3.1 Sidekicks

The FSTSP is initially introduced by Murray and Chu (2015) 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, Dell'Amico et al. (2021), Freitas et al. (2023), Yu et al. (2023) and Boccia et al. (2023) introduced improved modeling approaches, which accelerate a standard solver. Boccia et al. (2023) additionally introduce a branch-and-cut approach that outperforms their MILP formulation. Further, multiple publications consider close problem settings with either a different objective or extensions like an increased number of trucks or drones. So, Murray and Raj (2020) introduce the mFSTSP, where the truck is equipped with multiple drones. Multiple drones are also considered by Seifried (2019), Cavani et al. (2021), Dell'Amico et al. (2021) and Tamke

and Buscher (2021) additionally consider multiple trucks. The authors present a MILP formulation, and Cavani et al. (2021) and Tamke and Buscher (2021) also an exact method based on a branch-and-cut algorithm. Ha et al. (2018) (single truck) and Sacramento et al. (2019) (multiple trucks) focus on cost minimization and present both a MILP and heuristic solution approaches. Moshref-Javadi et al. (2020) consider an objective that minimizes the customers' waiting times, which equals a makespan minimization but without consideration of the vehicles' return times to the depot.

3.2 Sidekicks and loops

There are only a few publications considering both sidekicks and loops. However, Schermer et al. (2019) 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. Tiniç et al. (2023) propose two MILP formulations and a branch-and-cut approach 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 of minimizing the makespan when permitting loops, especially at the depot, is missing. Moreover, the authors have certain assumptions that restrict drone flights, e.g., drones are not permitted to perform loops at the depot.

Dell'Amico et al. (2021) derive multiple benchmark results for the instances of Murray and Chu (2015) for different drone settings, including loops, considering a single drone. Schermer et al. (2020), Roberti and Ruthmair (2021), El-Adle et al. (2021) and Dell'Amico et al. (2022) present two-indexed formulations of a TSPD. While Schermer et al. (2020) additionally introduce an exact brach-and-cut approach and Roberti and Ruthmair (2021) a branch-and-price approach that are capable of solving larger instances, El-Adle et al. (2021) 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. Dell'Amico et al. (2022) present an enhanced 2-indexed MILP formulation and outperform state-of-the-art MILP formulations from the literature, e.g., Roberti and Ruthmair (2021). Wang et al. (2017) analyze potential makespan savings by drones and derive multiple worst-case results. Considering multiple drone delivery options, Rave et al. (2023) 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.

3.3 Variants of truck-and-drone tandems

Karak and Abdelghany (2019) and Moshref-Javadi et al. (2020) evaluate truck-and-drone tandems where the truck does not serve customers but only launches drones at

advantageous places to perform loops. Salama and Srinivas (2020) extend this problem by additionally allowing the truck to serve customers. Chang and Lee (2018) also only consider drone loops and set the drone launching spots flexible by making use of the k-means clustering algorithm. The truck routing is determined by solving the TSP in a second step. Agatz et al. (2018) (single drone) and Morandi et al. (2023) (multiple drones) consider a variant where drones have the possibility of retraversing arcs. In addition, Morandi et al. (2023) allow drones to visit multiple nodes per trip. This is also considered by Poikonen and Golden (2020), who additionally take the drones' energy consumption into account. Kitjacharoenchai et al. (2019) and İbroşka et al. (2023) consider 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. On the contrary, the number of drone flights is restricted by considering at most one launch and return per node, and loops may be only performed at the depot. If a loop is performed at the depot, the drone is excluded from the truck's tours.

3.4 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 and with other exact method if introduced. The literature of paragraph "Variants of truck-and-drone tandems" is not included due to the large difference of the problem setting.

Our problem setting major differs from the literature by the unrestricted number of drones, the objective, and the general flight settings of drones, i.e., we also take drones performing loops at the depot into account. As a result, our problem setting is the most general TSPmD formulation so far.

From a methodological point of view, our MILP is the only one that has two-indexed variables while considering drone loops and multiple drones. There are four two-indexed formulations for the TSPD (El-Adle et al. 2021; Schermer et al. 2020; Roberti and Ruthmair 2021; Dell'Amico et al. 2022) and one publication with a two-indexed formulation for the mFSTSP (Seifried 2019). Moreover, we can solve instances with the largest number of customers without any pre-processing steps or lower and upper bounds.

The truck-and-drone tandem presented in this paper extends the one of Rave et al. (2023). In contrast to their paper, we minimize the makespan, have additional assumptions (e.g., we consider the return time of drones to the depot), and an enhanced modeling version of some constraints accelerating the runtime of the solver.

4 Two-indexed TSPmD formulation

In this section, we present the MILP formulation for the TSPmD. First, in Sect. 4.1, we describe the proceeding of creating the two-indexed formulation. Second, in Sect. 4.2, the used index sets, parameters, and variables are introduced. Third, in Sect. 4.3, the MILP formulation is presented.

Table 1 Comparison of the relevant literature on truck-and-drone tandems

	Assumptions			Objective			Instance size	
	# Trucks	# Drones (per truck)	Loops	Costs	Makespan	# Variables' indices	MILP	Other exact method
Murray and Chu (2015)	1	1			✓	3	–	
Ha et al. (2018)	1	1		✓		3	11	
Dell'Amico et al. (2021)	1	1			✓	2	14	
Freitas et al. (2023)	1	1			✓	5	11	
Boccia et al. (2023)	1	1			✓	3	20	40
Yu et al. (2023)	1	1			✓	3	11	
Sacramento et al. (2019)	n	1		✓		3	13	
Seifried (2019)	1	m			✓	2	14	
Murray and Raj (2020)	1	m			✓	4	9	
Moshref-Javadi et al. (2020)	1	m				4	10	
Cavani et al. (2021)	1	m			✓	3	25	25
Dell'Amico et al. (2021)	1	m			✓	3	11	
Tamke and Buscher (2021)	n	m			✓	5	9	30
El-Adle et al. (2021)	1	1	✓		✓	2	32*	
Dell'Amico et al. (2021)	1	1	✓		✓	3	11	
Roberti and Ruthmair (2021)	1	1	✓		✓	2	10	40
Dell'Amico et al. (2022)	1	1	✓		✓	2	20	
Schermer et al. (2020)	1	1	✓		✓	2	20	20
Tiniç et al. (2023)	1	∞	✓**	✓		3	13	20
Wang et al. (2017)	n	m	✓		✓			
Schermer et al. (2019)	n	m	✓***		✓	3	11	
Rave et al. (2023)	n	m	✓	✓		3	15	
This paper	1	∞	✓		✓	2	28	

Note that each publication listed considers drones performing sidekicks. * The authors added lower and upper bounds and a pre-processing step before running the MILP, ** no drone loops allowed at the depot, *** drone loops allowed at the depot, but the drone is then excluded from its truck's tour

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 (Murray and Chu 2015), and thus consisting of the outbound flight to the customer and the return flight to the

truck. To formulate it mathematically using only two-indexed variables, the drone's flights are split up into two variables instead of one: one for the outbound $\vec{y}_{i,c}$ and one for the return flights $\overleftarrow{y}_{c,j}$ and these flights are matched in the constraints (e.g., Dell'Amico et al. 2021).

Considering multiple drones per truck, typically an index set for vehicles is introduced (e.g., Murray and Raj 2020), which is, however, not necessary, i.e., the drone routing can be formulated without the index set for vehicles. Each drone flight can be defined over the same variables $\vec{y}_{i,c}$ and $\overleftarrow{y}_{c,j}$ with a flow formulation (Seifried 2019), instead. On the contrary, it should be noted that an index per vehicle allows the use of heterogeneous vehicles.

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., El-Adle et al. 2021; Roberti and Ruthmair 2021), 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., Dell'Amico et al. (2021), 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 main decisions taken are the routing of the truck ($x_{i,j}$), the routing of the drones ($\vec{y}_{i,c}$, $\overleftarrow{y}_{c,j}$), and the decision if a customer is served in a loop ($\lambda_{i,c}$). Additionally, the model decides on the customers served by truck (z_c^T) and drones (z_c^D).

Drones have an endurance e per flight. Similar to, e.g., Dell'Amico et al. (2021), 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.

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 (Seifried 2019). To

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

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_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 $(i, j \in \mathcal{I}_0)$
$\vec{y}_{i,c}$	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 $j \in \mathcal{I}_0$
$\lambda_{i,c}$	Binary variable indicating if customer $c \in \mathcal{I}_D$ is served in a drone loop launched at node $i \in \mathcal{I}_0$
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 Miller et al. (1960) $(c \in \mathcal{I})$
$\tau_{i,j}$	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 $(i = j)$

overcome this issue, we add another definition of the makespan (Constraint 3), which helps to find strong lower bounds, particularly in initial iterations. Note that either Constraint 2 or Constraint 3 can be chosen. The solver has the best runtime if both constraints are chosen.

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

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

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.

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

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

$$\sum_{i \in \mathcal{I}_0} \vec{y}_{i,c} = \sum_{j \in \mathcal{I}_0} \overleftarrow{y}_{c,j} = z_c^D \qquad \forall c \in \mathcal{I}_D \tag{6}$$

Note that the MILP can be formulated without variables z_c^T and z_c^D and, thus, without Constraints 5 and 6 (right side). However, the introduced MILP has a better runtime when these constraints are included.

Truck related constraints

Constraints 7 conserve the truck's flow. Subtours are eliminated by Constraints 8 (Miller et al. 1960). M_1^{big} can be set as the number of nodes $|\mathcal{I}_0|$. The truck is launched at maximum once from the depot (Constraint 9). Constraints 10 ensure that the truck only travels to a different node.

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

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

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

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

Drone related constraints

Drones may only launch and land at nodes the truck has visited (Constraints 11). Drone flights must not exceed the endurance (Constraints 12). Additionally, the truck must pick up the drone within the considered endurance limit (Constraints 13, if launched at node $i \in \mathcal{I}$ and Constraints 14, if launched at the depot). M_2^{big} can be set as the truck's maximum possible tour length.

$$\vec{y}_{j,c} + \overleftarrow{y}_{c,j} \leq 2 \cdot \sum_{i \in \mathcal{I}_0} x_{i,j} \qquad \forall c \in \mathcal{I}_D, j \in \mathcal{I} \tag{11}$$

$$t_{i,c}^D \cdot \vec{y}_{i,c} + t_{c,j}^D \cdot \overleftarrow{y}_{c,j} \leq e \qquad \forall i, j \in \mathcal{I}_0, c \in \mathcal{I}_D \tag{12}$$

$$\sum_{f \in \mathcal{I}_0} \left(\tau_{f,j} - \tau_{f,i}\right) - M_2^{big} \cdot (2 - \vec{y}_{i,c} - \overleftarrow{y}_{c,j}) \leq e \qquad \forall i \in \mathcal{I}, j \in \mathcal{I}_0, c \in \mathcal{I}_D \tag{13}$$

$$\sum_{f \in \mathcal{I}_0} \tau_{f,j} - M_2^{big} \cdot (2 - \vec{y}_{0,c} - \overleftarrow{y}_{c,j}) \leq e \qquad \forall j \in \mathcal{I}_0, c \in \mathcal{I}_D \tag{14}$$

Setting up the truck's travel time

The truck's accumulated travel times at the beginning of the tour are defined in Constraints 15. Constraints 16 define the lower and Constraints 17 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 18).

$$\tau_{0,j} = t^T_{0,j} \cdot x_{0,j} \qquad \forall j \in \mathcal{I} \tag{15}$$

$$\tau_{i,j} \geq t^T_{i,j} \cdot x_{i,j} + \sum_{f \in \mathcal{I}_0} \left(\tau_{f,i} + \phi_{f,i} \right) - M^{big}_2 \cdot (1 - x_{i,j}) \qquad \forall i \in \mathcal{I}, j \in \mathcal{I}_0 \tag{16}$$

$$\tau_{i,j} \leq t^T_{i,j} \cdot x_{i,j} + \sum_{f \in \mathcal{I}_0} \left(\tau_{f,i} + \phi_{f,i} \right) + M^{big}_2 \cdot (1 - x_{i,j}) \qquad \forall i \in \mathcal{I}, j \in \mathcal{I}_0 \tag{17}$$

$$\tau_{i,j} \leq M^{big}_2 \cdot x_{i,j} \qquad \forall i, j \in \mathcal{I}_0 \tag{18}$$

Sidekicks

Constraints 19 (at the beginning of the tour) and Constraints 20 (during the tour) ensure that drones do not return to a node that was visited by the truck previously on its tour. M^{big}_3 can be set as twice the reciprocal value of the truck's shortest travel time $t^T_{i,j}$ between two nodes. M^{big}_4 is dependent on the values of M^{big}_2 and M^{big}_3 and has to be at least $M^{big}_2 \cdot M^{big}_3$. The lower bound of the truck's waiting time arriving at node j is presented in Constraints 21 at the beginning of the tour and in Constraints 22 during the tour. The inequality exists since the truck waits at a node until the last drone has returned.

$$\begin{aligned} \vec{y}_{0,c} + \overleftarrow{y}_{c,j} \leq\ & M^{big}_3 \cdot \sum_{f \in \mathcal{I}_0} \tau_{f,j} \\ & + M^{big}_4 \cdot (2 - \vec{y}_{0,c} - \overleftarrow{y}_{c,j}) \qquad \forall j \in \mathcal{I}, c \in \mathcal{I}_D \end{aligned} \tag{19}$$

$$\begin{aligned} \vec{y}_{i,c} + \overleftarrow{y}_{c,j} \leq\ & M^{big}_3 \cdot \sum_{f \in \mathcal{I}_0} \left(\tau_{f,j} - \tau_{f,i} \right) \\ & + M^{big}_4 \cdot (2 - \vec{y}_{i,c} - \overleftarrow{y}_{c,j}) \qquad \forall i \in \mathcal{I}, j \in \mathcal{I}_0, i \neq j, c \in \mathcal{I}_D \end{aligned} \tag{20}$$

$$\begin{aligned} \phi_{0,j} \geq\ & t^D_{0,c} \cdot \vec{y}_{0,c} + t^D_{c,j} \cdot \overleftarrow{y}_{c,j} \\ & - \sum_{f \in \mathcal{I}_0} \tau_{f,j} - M^{big}_2 \cdot (2 - \vec{y}_{0,c} - \overleftarrow{y}_{c,j}) \qquad \forall j \in \mathcal{I}, c \in \mathcal{I}_D \end{aligned} \tag{21}$$

$$\begin{aligned}\phi_{i,j} \geq\ & t^D_{i,c} \cdot \vec{y}_{i,c} + t^D_{c,j} \cdot \overleftarrow{y}_{c,j} \\ & - \sum_{f \in \mathcal{I}_0} \left(\tau_{f,j} - \tau_{f,i} - \phi_{f,i}\right) \\ & - M_2^{big} \cdot (2 - \vec{y}_{i,c} - \overleftarrow{y}_{c,j}) \qquad \forall i \in \mathcal{I}, j \in \mathcal{I}_0, c \in \mathcal{I}_D : i \neq j\end{aligned} \tag{22}$$

Loops

The truck waits at node i until all drones planned to return have returned to node i (Constraints 23). Constraints 24 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).

$$\phi_{i,i} \geq 2 \cdot t^D_{i,c} \cdot \lambda_{i,c} \qquad \forall i \in \mathcal{I}_0, c \in \mathcal{I}_D \tag{23}$$

$$\frac{\vec{y}_{i,c} + \overleftarrow{y}_{c,i}}{2} \geq \lambda_{i,c} \geq \vec{y}_{i,c} + \overleftarrow{y}_{c,i} - 1 \qquad \forall i \in \mathcal{I}_0, c \in \mathcal{I}_D \tag{24}$$

Definition of variables

Last, decision and auxiliary variables are defined.

$$x_{i,j}, \vec{y}_{i,c}, \overleftarrow{y}_{c,j} \in \{0,1\}, \phi_{i,j}, \tau_{i,j}, \tau^* \in \mathbb{R}^+ \qquad \forall i,j \in \mathcal{I}_0, c \in \mathcal{I}_D \tag{25}$$

$$z_i^T, z_c^D \in \{0,1\}, u_i \in \mathbb{R} \qquad \forall i \in \mathcal{I}, c \in \mathcal{I}_D \tag{26}$$

$$\lambda_{i,c} \in \{0,1\} \in \mathbb{N} \qquad \forall i \in \mathcal{I}_0, c \in \mathcal{I}_D \tag{27}$$

As mentioned in our problem setting, batteries are recharged during the tour. If batteries need to be changed manually, i.e., by the truck driver, a battery change time s needs to be added to the return flight in Constraints 21 and 22, i.e., $(t^D_{c,j} + s)$ instead of $t^D_{c,j}$. If batteries, on the other hand, are swapped automatically, this only increases the makespan by s when a drone is the last vehicle to return to the depot. This is because there are no second flights for drones required for an unlimited fleet of drones, as there are always sufficient drones available.

5 Problem variants: mFSTSP and TSPD

In this section, we show how our MILP can be adopted to cover the problem variants mFSTSP, TSPD, and TSPmD when the drone fleet is limited. First, in Sect. 5.1, we limit the number of drones that are, moreover, only able to perform sidekicks as considered by, e.g., Murray and Raj (2020) and Dell'Amico et al. (2021) (mFSTSP). In Sect. 5.2, we adjust the MILP formulation to cover a single drone performing sidekicks and loops as considered by, e.g., Roberti and Ruthmair (2021), El-Adle

et al. (2021), and Dell'Amico et al. (2022) (TSPD). Last, in Sect. 5.3, we adjust the MILP formulation for the TSPmD to cover a limited drone fleet.

5.1 mFSTSP: limited number of drones performing only sidekicks

To cover a problem with a limited number of drones $m \in \mathbb{N}$ performing sidekicks, we add the variable $\pi_{i,j}$ that tracks if the drone travels on the truck from node i to j ($i, j \in \mathcal{I}_0$, $i \neq j$) and thus does not serve a customer at that time. The following constraints need to be added to Constraints 2-22, 25 and 26:

$$\sum_{j \in \mathcal{I}} \pi_{0,j} + \sum_{c \in \mathcal{I}_D} \vec{y}_{0,c} = m \tag{28}$$

$$\sum_{j \in \mathcal{I}_0 : i \neq j} \pi_{i,j} = \sum_{k \in \mathcal{I}_0 : k \neq i} \pi_{k,i} + \sum_{c \in \mathcal{I}_D} \left(\overleftarrow{y}_{c,i} - \vec{y}_{i,c} \right) \qquad \forall i \in \mathcal{I} \tag{29}$$

$$\pi_{i,j} \leq m \cdot x_{i,j} \qquad \forall i, j \in \mathcal{I}_0 \, : \, i \neq j \tag{30}$$

$$\sum_{c \in \mathcal{I}_D} \vec{y}_{i,c} \leq m \qquad \forall i \in \mathcal{I}_0 \tag{31}$$

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

$$\pi_{i,j} \in \mathbb{N} \qquad \forall i, j \in \mathcal{I}_0 \, : \, i \neq j \tag{33}$$

Constraint 28 defines the number of drones the truck carries starting from the depot. Constraints 29 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 30. There are at maximum m sidekicks per node (Constraints 31). Variable structures are generally chosen such that drones might perform loops. Thus, Constraints 32 prevent drones from launching and returning to the same node. Last, variable $\pi_{i,j}$ is defined.

5.2 TSPD: single drone performing sidekicks and loops

To cover the TSPD with a drone performing sidekicks and loops, the following constraints need to be added to Constraints 2–22, 24–27, 29 and 33. Note that $\pi_{i,j}$ now tracks the number of drone loops at a node i, if $i = j$, following $\pi_{i,j} \in \mathbb{N}$.

$$\sum_{j \in \mathcal{I}} \pi_{0,j} + \sum_{c \in \mathcal{I}_D} \left(\vec{y}_{0,c} - \lambda_{0,c} \right) = 1 \tag{34}$$

$$\pi_{i,j} \leq x_{i,j} \qquad \forall i,j \in \mathcal{I}_0 : i \neq j \tag{35}$$

$$\sum_{c \in \mathcal{I}_D} \left(\vec{y}_{i,c} - \lambda_{i,c}\right) \leq 1 \qquad \forall i \in \mathcal{I}_0 \tag{36}$$

$$\sum_{i \in \mathcal{I}_0} \pi_{i,j} + \sum_{c \in \mathcal{I}_D} \left(\overleftarrow{y}_{c,j} - 2 \cdot \lambda_{j,c}\right) \geq 0 \qquad \forall j \in \mathcal{I}_0 \tag{37}$$

$$\lambda_{j,k} \leq \sum_{c \in \mathcal{I}_D} \left(\overleftarrow{y}_{c,j} - \lambda_{j,c}\right) + \sum_{i \in \mathcal{I}_0 : i \neq j} \pi_{i,j} \qquad \forall j \in \mathcal{I}_0, k \in \mathcal{I}_D \tag{38}$$

$$\phi_{i,i} \geq \sum_{c \in \mathcal{I}_D} 2 \cdot t^D_{i,c} \cdot \lambda_{i,c} \qquad \forall i \in \mathcal{I}_0 \tag{39}$$

Constraints 34–36 are variants of Constraints 28, 30 and 31 considering a single drone and loops. Constraints 37 ensure that the drone only launches to perform loops if it was carried by the truck at node j. There may be only loops if the drone returns from a sidekick or a drone travels on the truck to node j (Constraints 38). Last, Constraints 39 ensure that there may be multiple loops per node.

5.3 TSPmD: limited number of drones

To consider a TSPmD with a limited number of drones $m \in \mathbb{N}$, the following constraints need to be added to Constraints 2–27, 29, 33 and 37-39:

$$\sum_{j \in \mathcal{I}} \pi_{0,j} + \sum_{c \in \mathcal{I}_D} \left(\vec{y}_{0,c} - \lambda_{0,c}\right) = m \tag{40}$$

$$\pi_{i,j} \leq m \cdot x_{i,j} \qquad \forall i,j \in \mathcal{I}_0 : i \neq j \tag{41}$$

$$\sum_{c \in \mathcal{I}_D} \left(\vec{y}_{i,c} - \lambda_{i,c}\right) \leq m \qquad \forall i \in \mathcal{I}_0 \tag{42}$$

These constraints are similar to Constraints 34–36, however, they limit the number of drones used to m instead of 1. Note that a limitation of this formulation with multiple drones is that each drone can perform a maximum of one loop per node.

6 Numerical experiments

In this section, we describe the instances for which we present new benchmark solutions (Sect. 6.1). Next, we analyze the runtime solving the MILP and compare the runtime of our MILP to MILPs for the mFSTSP and the TSPD from the literature

(Sect. 6.2). Last, we analyze the impact of loops in general and especially at the depot on makespan minimization. We also conduct an a-posteriori cost analysis.

The MILPs are 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 s.

6.1 Benchmark instances

For experiments, we consider the publicly available instance sets of Murray and Chu (2015) and Bouman et al. (2018). Detailed solutions of each individual instance for benchmark purposes can be found in Appendix A for the instances of Murray and Chu (2015) and in Appendix B for the instances of Bouman et al. (2018). We present results for these instances for the TSPmD, the mFSTSP ($m = 3$), and the TSPD.

6.1.1 Instances of Murray and Chu (2015)

Murray and Chu (2015) 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 min and three different drone-to-truck speed ratios $\alpha \in \{0.6, 1.0, 1.4\}$. Similar endurance limits are considered by, e.g., Rave et al. (2023). 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. Note that we do not consider the launching and rendezvous times that are included in these instances.

6.1.2 Instances of Bouman et al. (2018)

Bouman et al. (2018) 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 El-Adle et al. (2021). 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. This is an uncommon assumption. However, the travel time is important, which depends on the speed and distance, so we vary the travel time by increasing the drone speed in the analyses. All customers can be served by drones.

6.2 Runtime analysis

To show the efficiency of our MILP formulation for the TSPmD and also for the two problem variants, we analyze their runtime in this section. First, in Sect. 6.2.1, we compare the runtime of the TSPmD, the mFSTSP, and the TSPD. Second, in

Sect. 6.2.2, we benchmark our MILP's runtime with a total of eight MILP formulations from the literature, solving our problem variants.

6.2.1 Runtime analysis for the TSPmD and the two problem variants

In this section, we analyze the runtime of solving our MILP for the TSPmD and the two problem variants, the mFSTSP with $m = 3$ and the TSPD. Table 3 reports the results for the instances aggregated by the number of nodes $|\mathcal{I}_0|$ and the endurance e. The table shows 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 s.

Findings For the instances of Murray and Chu (2015), we could find all optimal solutions except two if one drone is considered (TSPD). In particular, it is noticeable for all these instances that the optimal solution is found much faster when multiple drones are considered and when the endurance is lower. This is because there are more drone flights in the optimal solution considering the TSPmD, resulting in shorter truck routes. In addition, a lower endurance reduces the number of feasible flight options.

For the instances of Bouman et al. (2018), 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 three optimal solutions for instances with 28 nodes when considering the TSPmD, the mFSTSP, and the TSPD. In contrast to the instances of Murray and Chu (2015), no significant change in runtime can be observed if multiple drones are considered. This is because drones are less competitive towards the truck in these instances as both follow the Euclidean path and, thus, the computation time mainly arises from the truck routing.

6.2.2 Runtime comparison to the literature

The two-indexed MILP formulation of the TSPmD, mFSTSP, and TSPD presented in this paper can solve instances with more nodes than the MILP formulations from the literature (see also Table 1). So, Murray and Chu (2015) could not solve one of the 72 instances with eleven nodes to optimality. Considering the instances of Bouman et al. (2018), El-Adle et al. (2021) 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 in total eight MILP formulations from the literature.

6.2.2.1 Problem variant: mFSTSP We implemented the four-indexed MILP of Murray and Raj (2020) with an automated launch and recovery system, the three-indexed MILP of Cavani et al. (2021), and the three-indexed MILP with crossing sortie variables of Dell'Amico et al. (2021) in OPL. The MILP formulations mainly differ from our MILP by the larger number of indices and constraints and by considering the depot twice, i.e., for the start and end of the tour. The MILP formulations were cho-

Table 3 Aggregated results for instances of Murray and Chu (2015) and Bouman et al. (2018)

			TSPmD			Problem variant 1 mFSTSP (m = 3)			Problem variant 2 TSPD		
	$\lvert\mathcal{I}_0\rvert$	e	Opt	CPU [s]	Gap (%)	Opt	CPU [s]	Gap (%)	Opt	CPU [s]	Gap (%)
Murray and Chu (2015)	11	20	36/36	3	0	36/36	6	0	36/36	90	0
	11	40	36/36	3	0	36/36	14	0	34/36	850	0
	Summary		72/72	3		72/72	10		70/72	470	
Bouman et al. (2018)	16	30	10/10	6	0	10/10	6	0	10/10	11	0
	20	30	10/10	349	0	10/10	227	0	10/10	275	0
	24	30	8/10	1211	1	8/10	1093	0	8/10	1301	1
	28	30	2/10	3197	3	3/10	3139	3	2/10	3285	4
	32	30	0/10	3600	9	0/10	3600	11	0/10	3600	12
	Summary		30/50	1673		31/50	1613		30/50	1694	

sen, as Murray and Raj (2020) initially introduced the mFSTSP and Cavani et al. (2021) and Dell'Amico et al. (2021) developed a MILP that is easier to solve. We tested the MILP formulations for the instances of Murray and Chu (2015) and Bouman et al. (2018). Note that the MILP of Cavani et al. (2021) requires a drone speed that is at least the truck's speed. Thus, only a subset of the instances of Murray and Chu (2015) is considered. Furthermore, we adjusted the MILP of Cavani et al. (2021) by endurance constraints similar to Constraints 13 and 14, and we added constraints ensuring that some customers cannot be served by drone, but by truck. This is already included in the MILP formulations of Murray and Raj (2020) and Dell'Amico et al. (2021).

Table 4 presents the aggregated results solving all three MILPs and our MILP formulation for the mFSTSP. The column Gap shows the average optimality gap if the solver cannot find the optimal solution within 3600 s. Note that all MILP formulations represent the same problem setting of the mFSTSP with three drones that can only perform sidekicks. For implementing the MILPs, we also consider the enhanced modeling assumptions to accelerate the MILP solver presented in the papers (similar to Constraint 3).

Findings The MILP of Murray and Raj (2020) cannot not solve any instance considered. For the instances of Bouman et al. (2018), the MILP cannot even find a single lower bound. This is in line with the findings of Murray and Raj (2020) themselves, who only found optimal solutions for 66% of the considered instances with eight customers. Contrary, the MILPs of Cavani et al. (2021) and Dell'Amico et al. (2021) can solve the instances of Murray and Chu (2015). Considering larger instances, the MILP of Cavani et al. (2021) solves up to two instances with 24 nodes, and the MILP of Dell'Amico et al. (2021) solves seven instances with 16 nodes to optimality. Our MILP formulation, however, not only finds all optimal solutions for the instances of Murray and Chu (2015) with a low runtime of 10 s but also up to three optimal solutions for instances of Bouman et al. (2018) with 28 nodes. If the optimal solution cannot be found, the gaps are rather small. Only the MILP of Cavani et al. (2021) has a lower runtime for the instances of Murray and Chu (2015) but can only solve a subset of all considered instances. Thus, our MILP formulation for the mFSTSP outperforms the MILP formulations of Murray and Raj (2020) and Dell'Amico et al. (2021) for all considered instances, and the MILP formulation of Cavani et al. (2021) for larger instances.

In "Appendix C", we further benchmark our MILP to the MILP of Boccia et al. (2023) for the special case of $m = 1$, finding that we outperform it for larger instances.

6.2.2.2 Problem variant: TSPD We implemented the MILPs of Schermer et al. (2020), Roberti and Ruthmair (2021) and El-Adle et al. (2021), and the two-indexed MILP of Dell'Amico et al. (2022) for a TSPD in OPL. The formulations of Roberti and Ruthmair (2021) and El-Adle et al. (2021) differ from our MILP by considering additional variables tracking the drone's tour, even if it is carried on the truck. Moreover, Schermer et al. (2020), Roberti and Ruthmair (2021) and Dell'Amico et al. (2022) consider the depot twice, i.e., for the start and end of the tour. The MILP formulations are chosen for comparison, as they are efficient MILP formulations for the TSPD.

Table 4 Aggregated results running the MILPs of Murray and Raj (2020) and Dell'Amico et al. (2021), and from this paper for the instances of Murray and Chu (2015) and Bouman et al. (2018)

			MILP of Murray and Raj (2020)			MILP of Cavani et al. (2021)*			MILP of Dell'Amico et al. (2021)			Our MILP (mFSTSP)		
	$\lvert\mathcal{I}_0\rvert$	e	Opt	CPU [s]	Gap (%)	Opt	CPU [s]	Gap (%)	Opt	CPU [s]	Gap (%)	Opt	CPU [s]	Gap (%)
Murray and Chu (2015)	11	20	0/36	3600	93	24/24	3	0	36/36	108	0	36/36	6	0
	11	40	0/36	3600	98	24/24	2	0	36/36	132	0	36/36	14	0
	Summary		0/72	3600		48/48	2		72/72	120		72/72	10	
Bouman et al. (2018)	16	30	0/10	3600	100	10/10	390	0	7/10	1646	5	10/10	6	0
	20	30	0/10	3600	100	3/10	2881	7	0/10	3600	25	10/10	227	0%
	24	30	0/10	3600	100	2/10	3070	9	0/10	3600	34	8/10	1093	0
	28	30	0/10	3600	100	0/10	3600	12	0/10	3600	46	3/10	3139	3
	32	30	0/10	3600	100	0/10	3600	16	0/10	3600	64	0/10	3600	11
	Summary		0/50	3600		15/50	2708		7/50	3209		31/50	1613	

All MILP formulations represent an mFSTSP (three drones, sidekicks, no loops) with the same objective and assumptions. * We modified the MILP by endurance constraints

Again, we tested the MILP formulations for the instances of Murray and Chu (2015) and Bouman et al. (2018). However, only a subset of the instances of Murray and Chu (2015) is suitable for Roberti and Ruthmair (2021) and El-Adle et al. (2021) because both MILP formulations require a drone speed that is at least the same as the truck's speed. The MILPs of Schermer et al. (2020) and El-Adle et al. (2021) are additionally adjusted by endurance constraints that limit the waiting times of drones (similar to Constraints 13 and 14). This is already considered by Roberti and Ruthmair (2021) and Dell'Amico et al. (2022). Note that for implementing the MILPs, we also consider the enhanced modeling assumptions to accelerate the MILP solver presented in the papers (similar to Constraint 3), but no lower or upper bounds or any pre-processing steps as in El-Adle et al. (2021). Table 5 presents the aggregated results solving the MILPs. 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 Roberti and Ruthmair (2021) works quite well for instances of Murray and Chu (2015), solving all of them to optimality. For larger instances, however, not one instance with 16 or more nodes could be solved. The MILP of Schermer et al. (2020) works best for the instances of Murray and Chu (2015), as all 72 instances could be solved optimally with the lowest runtime on average, but, on the other hand, it has difficulties solving the instances with 16 or more nodes. The MILP of Dell'Amico et al. (2022) has a similar performance compared to the MILP of Schermer et al. (2020) for the instances of Murray and Chu (2015) but works better for larger instances. Running the MILP of El-Adle et al. (2021), it outperforms our MILP for instances of Murray and Chu (2015) where the drone has at least the truck's speed. Furthermore, regarding the instances of Bouman et al. (2018), optimal solutions for all ten instances with 16 and 20 nodes and five out of ten instances with 24 nodes can be found. In contrast, our MILP formulation can additionally be solved to optimality for three more instances with 24 nodes and two instances with 28 nodes. Moreover, the average runtime and the optimality gap are significantly lower. Thus, our MILP formulation for the TSPD outperforms the literature for larger instances. Additionally, it should be noted that our MILP formulation is more general compared to Roberti and Ruthmair (2021) and El-Adle et al. (2021) as drones might have lower speeds than trucks, and also in comparison to Schermer et al. (2020) and Dell'Amico et al. (2022) as multiple drones can be considered without any increase in runtime.

Summary Our two-indexed MILP formulation outperforms all eight considered MILP formulations from the literature for larger instances, finding solutions for instances with a larger number of nodes. If optimality cannot be proven, the optimality gap is, moreover, the lowest compared to all other MILPs. Only for the instances of Murray and Chu (2015) our MILP formulation does not perform best. Please note that other exact solution methods, e.g., the branch-and-price of Roberti and Ruthmair (2021), might still outperform our MILP formulation.

6.2.2.3 Value of linear relaxation at the root node Further, we compare the value of the linear relaxation at the root node for all considered MILPs for the mFSTSP and the TSPD. For this, Table 6 shows the gap between the optimal solution, if available, and else the best-found solution solving all MILPs and the value of the

Table 5 Aggregated results running the MILPs of Schermer et al. (2020), El-Adle et al. (2021) and Dell'Amico et al. (2022), and from this paper for the instances of Murray and Chu (2015) and Bouman et al. (2018)

			MILP of Schermer et al. (2020)*			MILP of Roberti and Ruthmair (2021)			MILP of El-Adle et al. (2021)*			MILP of Dell'Amico et al. (2022)			Our MILP (TSPD)		
	$\lvert\mathcal{I}_0\rvert$	e	Opt	CPU [s]	Gap (%)	Opt	CPU [s]	Gap (%)	Opt	CPU [s]	Gap (%)	Opt	CPU [s]	Gap (%)	Opt	CPU [s]	Gap (%)
Murray and Chu (2015)	11	20	36/36	9	0	24/24	238	0	24/24	93	0	36/36	11	0	36/36	90	0
	11	40	36/36	17	0	24/24	76	0	23/24	399	1	36/36	28	0	34/36	850	0
	Summary		72/72	13		48/48	157		47/48	246		72/72	19	0	70/72	470	
Bouman et al. (2018)	16	30	9/10	418	16	0/10	3600	19	10/10	27	0	9/10	441	1	10/10	11	0
	20	30	3/10	3185	9	0/10	3600	28	10/10	576	0	6/10	2531	2	10/10	275	0
	24	30	0/10	3600	11	0/10	3600	30	5/10	2165	4	1/10	3292	7	8/10	1301	1
	28	30	0/10	3600	13	0/10	3600	81	0/10	3600	9	0/10	3600	11	2/10	3285	4
	32	30	0/10	3600	17	0/10	3600	88	0/10	3600	17	0/10	3600	15	0/10	3600	12
	Summary		12/50	2880		0/50	3600		25/50	1994		16/50	2693		30/50	1694	

All MILP formulations represent a TSPD (single drone, sidekicks, and loops) with the same objective and assumptions. * We modified the MILP by endurance constraints

Table 6 Gap [%] between the optimal solution if available and the value of the linear relaxation at the root node for all considered MILPs for the instances of Murray and Chu (2015) and Bouman et al. (2018)

			mFSTSP				TSPD				
			Murray and Raj (2020) (%)	Cavani et al. (2021)* (%)	Dell'Amico et al. (2021) (%)	Our MILP (%)	Schermer et al. (2020)* (%)	Roberti and Ruthmair (2021) (%)	El-Adle et al. (2021)* (%)	Dell'Amico et al. (2022) (%)	Our MILP
Murray and Chu (2015)	11	20	100	63	69	68	56	59	57	69	70
	11	40	100	56	63	62	53	57	54	67	68
	Summary		100	59	66	65	54	58	55	68	69
Bouman et al. (2018)	16	30	100	76	83	97	61	62	82	79	97
	20	30	100	75	87	99	62	63	86	81	98
	24	30	100	73	90	99	58	59	91	79	98
	28	30	100	72	90	99	57	58	92	78	98
	32	30	100	72	89	99	56	57	93	78	98
	Summary		100	74	88	99	59	60	89	79	98

* We modified the MILP by endurance constraints

linear relaxation at the root node. The results for all considered MILPs for the mFSTSP and the TSPD are reported.

Findings We observe that our MILP formulation never has the lowest gap of the linear relaxation at the root node for both the mFSTSP and the TSPD. Considering the mFSTSP Cavani et al. (2021) has the lowest gap. Considering the TSPD, on the other hand, the MILP of El-Adle et al. (2021) has the best value of linear relaxation for the instances of Murray and Chu (2015) (TSPD), and the MILP of Roberti and Ruthmair (2021) for the instances of Bouman et al. (2018). However, we find that the gap at the root node is not directly a good indicator for an efficient MILP formulation as Roberti and Ruthmair (2021) could not solve a single instance of Bouman et al. (2018) to optimality despite having a good value of the linear relaxation at the root node. Moreover, we observe that our MILPs' value of the linear relaxation at direct subsequent nodes increases significantly for all considered problem variants.

6.3 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., Sacramento et al. 2019). So, Schermer et al. (2019) already found that drones perform loops in the solutions, and Tiniç et al. (2023) found that loops might reduce total routing costs. However, in the solutions presented by Dell'Amico et al. (2021), 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 all kinds of loops for the instances of Murray and Chu (2015) and Bouman et al. (2018) solving the TSPD and the TSPmD with varying 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.

6.3.1 Instances of Murray and Chu (2015)

Table 7 presents the reduction of the makespan τ^* and the increase in the 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 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 Murray and Chu (2015) as the speed has a significant impact on the number of performed loops (Schermer et al. 2019). 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.

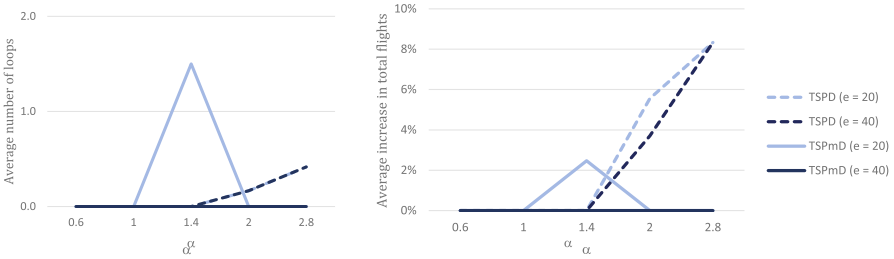
Findings Considering instances of Murray and Chu (2015) with eleven nodes, loops can reduce the makespan τ^* by up to 1.6% on average. For these instances, we find that there are only improvements in the objective considering loops if drones travel faster than trucks ($\alpha > 1$). Note that in these instances, the drone follows an Euclidean path contrary to the truck that travels the Manhattan distance. Improvements in considering loops occur, especially when a single drone is considered. Considering the TSPmD, there are nearly no reductions of τ^* as loops are only performed in two instances. Conversely, the runtime increases significantly by up to 98.4%, considering loops as an additional routing option. The runtime increase is significantly higher when drone loops become a more relevant and competitive flight option.

Further, we analyze the average number of performed loops (Fig. 3a) and the average increase in flights, including both sidekicks and loops (Fig. 3b) for the TSPD and the TSPmD for both endurance limits. We find that only a few loops are performed and only when considering a single drone. However, considering loops, the total number of drone flights increases if $\alpha > 1$ as drones perform loops in addition to sidekicks. Moreover, we find that both endurance limits do not have an impact on performed loops for these instances. A higher endurance allows more customers to be reached within drone loops, but this is contrasted by longer waiting times for the truck.

Last, we analyze the impact on the number of drones carried by the truck when considering loops. Figure 4 shows the average number of drones used solving the TSPmD for the two endurance limits and the case with and without drone loops. As the number of drone flights increases with increasing α, so does the number of drones. In addition, there are more drones when a larger endurance is considered

Table 7 Makespan reduction and the solver's runtime increase, if drones can perform loops, for the instances of Bouman et al. (2018)

			TSPD		TSPmD	
$\lvert\mathcal{I}_0\rvert$	e	α	τ^* Reduction (%)	Runtime Increase (%)	τ^* Reduction (%)	Runtime Increase (%)
11	20	0.6	–	25.6	–	17.2
		1.0	–	17.9	–	20.8
		1.4	–	18.9	0.5	25.9
		2.0	0.4	54.3	–	7.0
		2.8	1.6	66.0	–	20.6
	40	0.6	–	33.1	–	−12.4
		1.0	–	−7.5	–	4.4
		1.4	–	30.3	–	21.0
		2.0	0.4	75.6	–	0.3
		2.8	1.6	98.4	–	11.4
Average			0.4	42.3	0.0	11.6



(a) Average number of loops per instance.

(b) Average increase in drone flights, if loops are additionally considered

Fig. 3 Average number of loops and increase in drone flights for the TSPD and the TSPmD for different endurance limits when solving the instances of Murray and Chu (2015)

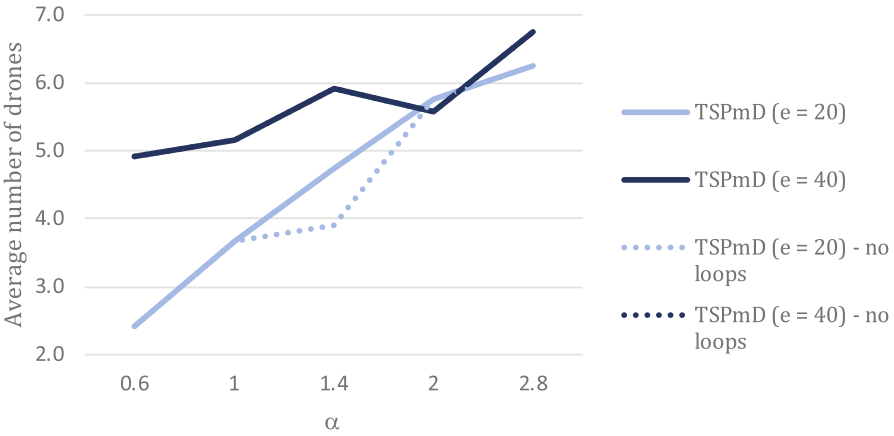


Fig. 4 Average number of drones in the TSPmD for different endurance limits with and without drone loops when solving the instances of Murray and Chu (2015)

because drones have longer flights on average, resulting in less availability. The number of drones with and without loops equals except one outlier, where all loops are performed at a single node, resulting in an increase in drones.

6.3.1.1 Instances of Bouman et al. (2018) Similar to the previous table, Table 8 presents the results for instances of Bouman et al. (2018) with 16 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 El-Adle et al. (2021). 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 We find that allowing drones to perform loops may reduce τ^* by up to 2.7%. Again, we do not find that the endurance e has a significant impact on τ^*, but

a larger drone-to-truck speed ratio α has. Contrary to the results for the instances of Murray and Chu (2015), there is a larger reduction of τ^* when considering the TSPmD as loops are performed in nearly all instances. This might be because there are more customers in the instances who can all be supplied by drones. 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 Euclidean paths. Again, the runtime increases significantly by up to 91.4% and is especially high if one drone is considered. However, considering the TSPmD, there is no runtime increase, but the impact on τ^* is high. One reason for this is that sidekicks are less competitive against loops in these instances for the considered parameter settings, and thus, loops occur more frequently.

Further, Fig. 5 shows the average number of performed loops (left) and the average increase in drone flights if loops can perform loops (right). Note that considering the TSPD with an endurance $e = 60$, no sufficient large number of instances could be solved optimally, and thus, it is not represented in the figure. We find that many more loops are performed in the instances of Bouman et al. (2018), especially if multiple drones are considered. This is because there is a larger number of customers that can be all served by drones. Considering the TSPmD with an endurance of $e = 30$, the increase in total flights decreases with additional drone speeds as there have already been many drone flights. Drone loops only replace certain sidekicks.

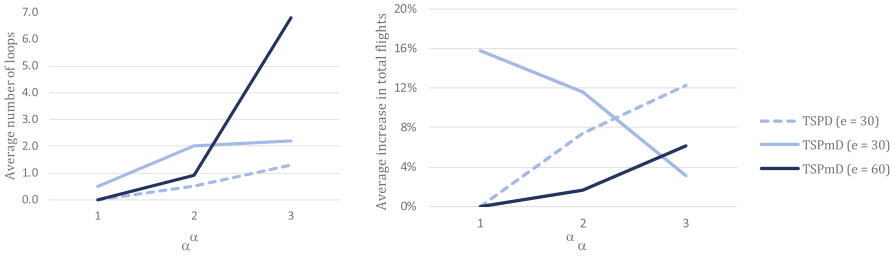
Last, we analyze the impact on the number of drones carried by the truck when considering loops in the TSPmD (Fig. 6). Again, there are more drones carried by the truck with larger values of α and e. In addition, there are slightly more drones if drone loops are allowed, as these loops are altogether performed at a single node.

6.3.1.2 Impact of loops performed at the depot Further, we analyze if it is worth considering loops performed at the depot. For this, we compare the results when allowing and prohibiting loops performed at the depot for the TSPD and the TSPmD

Table 8 Makespan reduction and the solver's runtime increase, if drones can perform loops, for the instances of Bouman et al. (2018)

			TSPD		TSPmD	
			τ^*	Runtime	τ^*	Runtime
$\lvert\mathcal{I}_0\rvert$	e	α	Reduction (%)	Increase (%)	Reduction (%)	Increase (%)
16	30	1	–	55.0	0.1	49.5
		2	0.8	43.5	1.4	3.5
		3	0.8	36.0	1.3	−28.6
	60	1	–	91.4	–	20.7
		2	./.	./.	1.7	−25.2
		3	./.	./.	2.7	−22.7
Average			0.4	49.9	1.2	−0.5

Entry./. indicates that no instance was solved to optimality for both the consideration and the ban of loops



(a) Average number of loops per instance.

(b) Average increase in drone flights, if loops are additionally considered

Fig. 5 Average number of loops and increase in drone flights for the TSPD and the TSPmD for different endurance limits when solving the instances of Bouman et al. (2018)

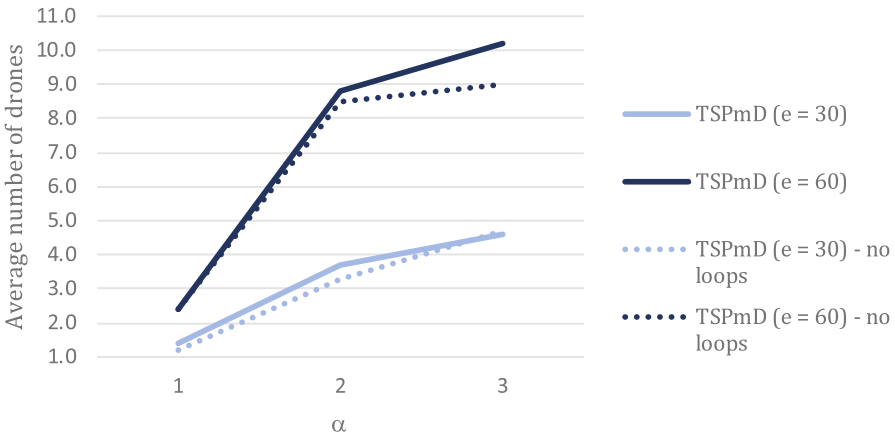


Fig. 6 Average number of drones in the TSPmD for different endurance limits with and without drone loops when solving the instances of Bouman et al. (2018)

for the same settings as before. We find that there is a single loop performed at the depot for one instance of Bouman et al. (2018) in the optimal solution. For the instances of Murray and Chu (2015), there are no depot loops. This is because the depot's location is set at the outskirt of 2/3 of all instances of Murray and Chu (2015) and all instances of Bouman et al. (2018), so there are hardly any customers nearby.

For further analysis, we thus place the depot in the center of gravity for the instances of Bouman et al. (2018) (see Table 9). We find for the considered instances that there are loops performed at the depot for the TSPmD as soon as $\alpha \geq 2$ and $e = 60$, resulting in large makespan savings of up to 18.5 on average. On the contrary, the runtime increases by up to 91.4% even if there are no loops performed at the depot in the optimal solution. Considering the TSPD, no instance could be solved to optimality for $\alpha \geq 2$ and $e = 60$.

Table 9 Makespan reduction and the solver's runtime increase, if drones can perform loops at the depot, for the instances of Bouman et al. (2018) with the depot placed in the center of gravity

			TSPD		TSPmD	
			τ^*	Runtime	τ^*	Runtime
$\lvert\mathcal{I}_0\rvert$	e	α	Reduction (%)	Increase (%)	Reduction (%)	Increase (%)
16	30	1	–	27.1	–	13.7
		2	–	7.0	–	27.6
		3	–	23.6	–	10.0
	60	1	–	17.9	–	−1.3
		2	./.	./.	8.2	−25.5
		3	./.	./.	18.5	39.3
Average			–	18.3	4.4	10.6

Entry./. indicates that no instance was solved to optimality for both the consideration and the ban of loops at the depot

6.4 A-posteriori cost analysis

In this section, we analyze the impact of permitting drones to perform loops on routing costs. This is done by an a-posteriori cost analysis, which means that costs are assigned to the makespan minimal routing. We consider the instances with the largest average makespan savings of 2.7%, i.e., the instances of Bouman et al. (2018) with $e = 60$ and $\alpha = 3$.

We consider costs for each traveled arc by truck and drone and waiting costs for the truck if the truck waits for drones (Rave et al. 2023). The costs for the outbound and return flights are considered the same because the reduced payload after serving a customer only leads to a few reductions in costs (Rave et al. 2023). The cost ratios vary within the literature. Thus, within the analysis, we take the different truck-to-drone cost ratios and waiting-to-operating cost ratios into account. Note that Tiniç et al. (2023) consider two truck-to-drone cost ratios. Table 10 shows the cost ratios and the cost savings when considering loops.

We find that the cost savings are highly dependent on the cost parameter values and vary between a cost reduction of 9.4% and even an increase in costs of up to 5.9%. This is because the consideration of loops increases the number of flights, which also increases the operating costs for drones. On the contrary, the truck travels less, resulting in lower operating costs. In addition, the truck's waiting time and, thus, the waiting costs increase. As a result, the largest cost savings are achieved with the parameter values of Sacramento et al. (2019) as they consider relatively low drone operating costs and no waiting costs. Similar but slightly lower cost savings are obtained with the costs as considered in Rave et al. (2023), taking waiting costs into account. Contrary, considering the cost values of Tiniç et al. (2023), drone operating costs and truck waiting costs are rather large compared to truck operating costs, and, thus, the additional drone tours lead to an increase in costs.

Table 10 Truck to drone cost ratios traveling an arc and waiting-to-operating cost ratios considered in the literature and the resulting cost savings. A negative entry indicates an increase in costs

	Truck to drone cost ratio (operating cost)	Waiting-to-operating cost ratio (truck)	Cost savings (%)
Sacramento et al. (2019)	10.0	0.00	9.4
Rave et al. (2023)	23.3	0.53	6.7
Tiniç et al. (2023)	2.0/4.0	0.75	−5.9/−4.2

7 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 that outperforms in total eight MILP formulations from the literature for the mFSTSP and the TSPD. We present new benchmark solutions for instances of Murray and Chu (2015) and Bouman et al. (2018), which we could solve to optimality for up to 28 nodes. We generate managerial insights on the impact of allowing drone loops in general and especially at the depot on makespan minimization. Moreover, we conduct an a-posteriori cost analysis.

We find that makespan savings of up to 2.7% can be achieved by allowing drones to perform loops. For the considered instances, 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, when minimizing the makespan, we recommend considering drone loops as soon as drones have at least the truck's speed. We also find that it is worth considering loops that are performed at the depot if the depot is placed centrally and the drones' speed and endurance are sufficiently large.

Compact and efficient MILP formulations for truck-and-drone tandems are good competitors to other exact algorithms, which, however, may still outperform efficient MILP formulations. So, for example, Roberti and Ruthmair (2021) could solve instances to optimality with 40 nodes in comparison to our MILP solving instances with 28 nodes. However, an efficient optimal solution approach is still required for the TSPmD that can solve instances with more than 28 nodes to optimality. Moreover, some assumptions can be strengthened, such as a consideration of limited (e.g., one or two) launch or return platforms for drones. This leads to a scheduling problem that needs to be solved to avoid drones exceeding their endurance limit. Further research can consider an automatic swap of batteries within a limited drone fleet. A battery change time s would only delay a drone's next flight if the next flight is planned before the battery could have been swapped as mentioned in Sect. 4.2.

Appendix A: Benchmark results for instances of Murray and Chu (2015)

The following Tables 11 and 12 present results for the instances of Murray and Chu (2015) for the endurance of $e = 20$ and $e = 40$. The instance names are similar to Dell'Amico et al. (2021).

Appendix B: Benchmark results for instances of Bouman et al. (2018)

The following Tables 13, 14, 15, 16 and 17 present the results for the instances of Bouman et al. (2018). The instance names are similar to El-Adle et al. (2021) and the results are split up for each considered number of nodes.

Appendix C: Benchmark test with the MILP of Boccia et al. (2023)

We further test the MILP of Boccia et al. (2023) because this MILP seems to be an efficient formulation. This MILP only considers a single drone performing sidekicks. Thus, we additionally benchmark our MILP for the special case of an mFSTSP with $m = 1$ to their three-indexed MILP formulation. Similar to our MILP, the MILP of Boccia et al. (2023) considers two-indexed variables for drone flights. However, a three-indexed variable tracks if the truck travels an arc (i, j) while the drone serves a customer k. Moreover, the MILP is an arc-based formulation. We tested the MILP formulation for the instances of Murray and Chu (2015) and Bouman et al. (2018). As the instances of Murray and Chu (2015) consider the case that only a subset of customers can be served by drone, the MILP of Boccia et al. (2023) is adjusted by constraints similar to Constraints 4 to 6.

Considering the instances of Murray and Chu (2015), the MILP formulation of Boccia et al. (2023) performs better than our MILP formulation (Table 18). However, when considering larger instances, the MILP has severe runtime issues, while our MILP formulation can solve 2 instances with 28 nodes.

One major drawback of the MILP formulation of Boccia et al. (2023) is that they eliminate subtours by a formulation that is based on subsets. As a result, there is a huge amount of constraints, which cannot even be generated in OPL in a reasonable amount of time. To tackle this problem, Boccia et al. (2023) develop a branch-and-cut approach that ignores the subtour elimination in the first step.

Table 11 Results for instances of Murray and Chu (2015) with an endurance of 20

				Problem variant 1:			Problem variant 2:		
	TSPmD			mFSTSP (m = 3)			TSPD		
Instance	τ^*	CPU (s)	Gap	τ^*	CPU (s)	Gap	τ^*	CPU (s)	Gap
20140810T123437v1	51.3825	4	–	51.3825	5	–	54.3926	14	–
20140810T123437v2	51.6111	5	–	51.6111	6	–	51.6111	8	–
20140810T123437v3	52.8225	8	–	52.8225	10	–	54.0684	17	–
20140810T123437v4	65.6225	5	–	65.6225	8	–	66.8684	9	–
20140810T123437v5	24.4390	1	–	32.0655	8	–	45.3353	239	–
20140810T123437v6	24.0264	2	–	28.9400	8	–	43.9153	241	–
20140810T123437v7	39.8664	7	–	43.2069	13	–	46.5813	28	–
20140810T123437v8	53.1454	7	–	57.7700	16	–	59.3813	20	–
20140810T123437v9	21.8536	1	–	24.9912	9	–	39.2035	687	–
20140810T123437v10	21.9720	1	–	25.7710	8	–	36.9077	160	–
20140810T123437v11	32.3077	2	–	34.4110	8	–	39.3002	52	–
20140810T123437v12	45.1077	3	–	47.7507	9	–	51.5645	29	–
20140810T123440v1	43.8455	4	–	43.8455	7	–	46.4304	31	–
20140810T123440v2	46.2455	7	–	46.2455	12	–	49.3737	42	–
20140810T123440v3	51.6277	9	–	51.6277	8	–	53.6616	19	–
20140810T123440v4	64.4277	6	–	64.4277	9	–	66.4616	10	–
20140810T123440v5	31.6066	2	–	32.2263	2	–	39.7498	198	–
20140810T123440v6	34.0066	2	–	34.0066	1	–	40.3790	85	–
20140810T123440v7	40.7304	1	–	40.7304	2	–	43.3126	29	–
20140810T123440v8	53.5304	1	–	53.5304	2	–	55.7960	31	–
20140810T123440v9	31.6066	1	–	31.6066	1	–	35.5331	9	–
20140810T123440v10	34.0066	1	–	34.0066	1	–	36.0764	9	–
20140810T123440v11	40.7304	1	–	40.7304	1	–	40.7304	4	–
20140810T123440v12	53.5304	1	–	53.5304	1	–	53.5304	8	–
20140810T123443v1	69.5865	1	–	69.5865	4	–	69.5865	5	–
20140810T123443v2	71.5839	4	–	71.7473	3	–	72.0639	7	–
20140810T123443v3	75.9039	2	–	76.0673	2	–	76.1447	2	–
20140810T123443v4	88.7039	3	–	88.8673	2	–	89.1839	5	–
20140810T123443v5	39.7845	4	–	42.2634	4	–	54.7521	222	–
20140810T123443v6	43.9992	5	–	45.0943	6	–	55.1268	106	–
20140810T123443v7	60.8333	5	–	60.8333	5	–	62.2491	11	–
20140810T123443v8	73.8724	6	–	76.3401	26	–	80.0971	79	–
20140810T123443v9	24.5760	1	–	27.6100	2	–	41.9314	724	–
20140810T123443v10	30.9760	1	–	30.9760	1	–	42.9348	86	–
20140810T123443v11	43.7760	2	–	43.7760	1	–	51.4056	11	–
20140810T123443v12	56.5760	1	–	56.5760	1	–	62.6175	8	–

Table 12 Results for instances of Murray and Chu (2015) with an endurance of 40

				Problem variant 1:			Problem variant 2:		
	TSPmD			mFSTSP (m = 3)			TSPD		
Instance	τ^*	CPU (s)	Gap	τ^*	CPU (s)	Gap	τ^*	CPU (s)	Gap
20140810T123437v1	33.8109	8	–	35.5186	75	–	49.1189	2946	–
20140810T123437v2	32.6510	7	–	35.3759	25	–	46.3113	490	–
20140810T123437v3	40.9090	10	–	49.2858	71	–	52.6868	221	–
20140810T123437v4	54.5333	47	–	62.5748	58	–	65.4868	238	–
20140810T123437v5	24.4390	1	–	28.4013	27	–	42.8354	3600	7.8%
20140810T123437v6	24.0264	1	–	28.4688	11	–	41.6015	1155	–
20140810T123437v7	35.8290	5	–	36.4363	9	–	43.3913	220	–
20140810T123437v8	48.6273	6	–	50.5161	12	–	56.1913	397	–
20140810T123437v9	21.8536	1	–	24.9912	6	–	37.8082	1771	–
20140810T123437v10	21.9720	2	–	25.7710	9	–	36.9077	879	–
20140810T123437v11	26.4022	1	–	29.3975	18	–	39.3002	86	–
20140810T123437v12	37.3724	1	–	42.1352	17	–	51.2536	134	–
20140810T123440v1	32.3895	1	–	35.5331	22	–	45.1029	993	–
20140810T123440v2	34.0066	1	–	37.2360	53	–	44.4461	1014	–
20140810T123440v3	45.3532	10	–	45.3532	16	–	52.3083	620	–
20140810T123440v4	58.1532	10	–	58.9284	14	–	66.3969	610	–
20140810T123440v5	31.6066	1	–	32.2263	2	–	39.7498	497	–
20140810T123440v6	34.0066	0	–	34.0066	1	–	39.6581	157	–
20140810T123440v7	40.7304	1	–	40.7304	1	–	43.3126	37	–
20140810T123440v8	53.5304	1	–	53.5304	1	–	55.7960	36	–
20140810T123440v9	31.6066	1	–	31.6066	1	–	35.5331	24	–
20140810T123440v10	34.0066	0	–	34.0066	1	–	36.0764	11	–
20140810T123440v11	40.7304	1	–	40.7304	1	–	40.7304	8	–
20140810T123440v12	53.5304	1	–	53.5304	1	–	53.5304	1	–
20140810T123443v1	37.4695	2	–	37.5843	5	–	54.0129	2117	–
20140810T123443v2	35.1953	1	–	38.5950	7	–	56.5729	1894	–
20140810T123443v3	47.4513	1	–	54.6474	23	–	66.1746	314	–
20140810T123443v4	65.6295	1	–	69.2538	7	–	81.4603	418	–
20140810T123443v5	24.2916	1	–	31.2183	9	–	48.5789	3600	20.7%
20140810T123443v6	30.0160	0	–	35.0597	9	–	48.4864	1993	–
20140810T123443v7	42.8160	0	–	44.3748	3	–	56.8793	354	–
20140810T123443v8	55.6160	0	–	56.5760	2	–	68.3988	181	–
20140810T123443v9	23.6160	0	–	27.6100	2	–	41.9314	3086	–
20140810T123443v10	30.0160	0	–	30.9760	2	–	42.9348	473	–
20140810T123443v11	42.8160	0	–	42.8160	0	–	51.3950	30	–
20140810T123443v12	55.6160	0	–	55.6160	0	–	62.6175	8	–

Table 13 Results for instances of Bouman et al. (2018) with 16 nodes

$|\mathcal{I}_0| = 16$

				Problem variant 1:			Problem variant 2:		
	TSPmD			mFSTSP (m = 3)			TSPD		
Instance	τ^*	CPU (s)	Gap	τ^*	CPU (s)	Gap	τ^*	CPU (s)	Gap
Uniform-61-n20	346.9223	28	–	347.0919	27	–	347.0919	73	–
Uniform-62-n20	353.7019	2	–	353.7019	2	–	353.7019	2	–
Uniform-63-n20	370.1349	2	–	372.9289	2	–	372.9289	2	–
Uniform-64-n20	357.9113	9	–	357.9113	9	–	357.9113	10	–
Uniform-65-n20	368.6511	4	–	368.6511	3	–	368.6511	4	–
Uniform-66-n20	426.5167	2	–	426.5167	2	–	426.5167	2	–
Uniform-67-n20	372.7803	1	–	372.7803	1	–	372.7803	2	–
Uniform-68-n20	423.3365	2	–	423.3443	2	–	423.3365	3	–
Uniform-69-n20	363.4143	5	–	363.4143	5	–	363.4143	7	–
Uniform-70-n20	410.1439	7	–	410.1439	6	–	410.1439	7	–

Table 14 Results for instances of Bouman et al. (2018) with 20 nodes

$|\mathcal{I}_0| = 20$

				Problem variant 1:			Problem variant 2:		
	TSPmD			mFSTSP (m = 3)			TSPD		
Instance	τ^*	CPU (s)	Gap	τ^*	CPU (s)	Gap	τ^*	CPU (s)	Gap
Uniform-61-n20	351.4622	1113	–	351.4622	895	–	351.5212	958	–
Uniform-62-n20	374.1195	1497	–	374.1195	452	–	374.1195	379	–
Uniform-63-n20	391.7060	28	–	394.5000	33	–	394.5000	57	–
Uniform-64-n20	368.7314	425	–	368.7314	524	–	368.7314	578	–
Uniform-65-n20	381.0652	42	–	381.0652	39	–	390.5601	215	–
Uniform-66-n20	434.9542	57	–	435.1632	111	–	436.2728	108	–
Uniform-67-n20	391.4744	40	–	391.4744	29	–	391.4744	40	–
Uniform-68-n20	435.6068	24	–	435.6068	22	–	435.6068	40	–
Uniform-69-n20	380.4329	218	–	380.4329	126	–	380.4329	331	–
Uniform-70-n20	422.6549	41	–	422.6549	38	–	422.6549	40	–

Table 15 Results for instances of Bouman et al. (2018) with 24 nodes

$|\mathcal{I}_0| = 24$

				Problem variant 1:			Problem variant 2:		
	TSPmD			mFSTSP (m = 3)			TSPD		
Instance	τ^*	CPU (s)	Gap	τ^*	CPU (s)	Gap	τ^*	CPU (s)	Gap
Uniform-71-n50	415.8466	681	–	415.8466	625	–	415.8466	1057	–
Uniform-72-n50	410.1562	30	–	410.1562	31	–	410.1562	32	–
Uniform-73-n50	391.3397	317	–	391.3397	173	–	394.5155	487	–
Uniform-74-n50	467.6828	345	–	467.6828	297	–	467.6828	472	–
Uniform-75-n50	463.6015	1810	–	463.6015	973	–	463.6015	680	–
Uniform-76-n50	394.2924	3600	5.3%	394.2924	3600	1.2%	394.2924	3600	2.1%
Uniform-77-n50	452.9030	139	–	452.9030	197	–	458.7309	467	–
Uniform-78-n50	410.8227	1040	–	410.8227	886	–	412.8484	2192	–
Uniform-79-n50	412.1437	547	–	412.1437	553	–	412.1437	422	–
Uniform-80-n50	391.7442	3600	1.6%	391.7442	3600	1.6%	397.2048	3600	4.0%

Table 16 Results for instances of Bouman et al. (2018) with 28 nodes

$|\mathcal{I}_0| = 28$

				Problem variant 1:			Problem variant 2:		
	TSPmD			mFSTSP (m = 3)			TSPD		
Instance	τ^*	CPU (s)	Gap	τ^*	CPU (s)	Gap	τ^*	CPU (s)	Gap
Uniform-71-n50	441.6016	3600	3.0%	441.6016	3600	3.8%	446.9016	3600	5.6%
Uniform-72-n50	449.6584	355	–	449.6584	470	–	449.6584	448	–
Uniform-73-n50	445.8629	3600	1.9%	445.8796	3600	0.3%	449.0387	3600	2.2%
Uniform-74-n50	470.8405	3600	0.4%	470.8406	2843	–	470.8407	3600	0.4%
Uniform-75-n50	473.0050	3600	4.7%	473.2393	3600	4.1%	473.0050	3600	4.9%
Uniform-76-n50	407.4875	3600	7.5%	407.4875	3600	7.9%	407.7456	3600	10.5%
Uniform-77-n50	480.5687	2820	–	480.5687	2879	–	485.5172	3600	2.4%
Uniform-78-n50	459.4857	3600	1.4%	459.4857	3600	1.0%	462.1742	3600	1.5%
Uniform-79-n50	447.7367	3600	7.5%	447.7367	3600	6.4%	447.7367	3600	7.4%
Uniform-80-n50	449.3641	3600	6.2%	449.3641	3600	7.1%	454.8248	3600	9.0%

Table 17 Results for instances of Bouman et al. (2018) with 32 nodes

$\lvert\mathcal{I}_0\rvert = 32$									
				Problem variant 1:			Problem variant 2:		
	TSPmD			mFSTSP (m = 3)			TSPD		
Instance	τ^*	CPU (s)	Gap	τ^*	CPU (s)	Gap	τ^*	CPU (s)	Gap
Uniform-71-n50	478.3559	3600	5.8%	478.3559	3600	6.2%	483.6558	3600	7.2%
Uniform-72-n50	493.0942	3600	2.5%	493.0942	3600	3.8%	493.0942	3600	4.3%
Uniform-73-n50	488.0547	3600	3.9%	490.4018	3600	6.2%	497.0028	3600	7.7%
Uniform-74-n50	478.5537	3600	11.2%	518.4635	3600	17.4%	480.7702	3600	13.6%
Uniform-75-n50	500.3800	3600	9.8%	510.0112	3600	13.4%	525.9658	3600	16.3%
Uniform-76-n50	496.0703	3600	20.1%	480.0877	3600	16.7%	484.6235	3600	19.2%
Uniform-77-n50	515.3738	3600	5.6%	511.1739	3600	4.5%	518.2364	3600	7.4%
Uniform-78-n50	504.8168	3600	5.2%	505.1834	3600	4.7%	507.7490	3600	5.4%
Uniform-79-n50	454.3164	3600	9.9%	479.9965	3600	15.5%	467.9651	3600	13.2%
Uniform-80-n50	447.2753	3600	13.2%	458.6088	3600	18.4%	481.3873	3600	24.6%

Table 18 Aggregated results running the MILPs of Boccia et al. (2023) and from this paper for the instances of Murray and Chu (2015) and Bouman et al. (2018)

			MILP of Boccia et al. (2023)			Our MILP (mFSTSP, $m = 1$)		
	$\lvert\mathcal{I}_0\rvert$	e	Opt	CPU (s)	Gap	Opt	CPU (s)	Gap
Murray and Chu (2015)	11	20	36/36	34	0%	36/36	94	0%
	11	40	36/36	28	0%	35/36	783	0%
	Summary		72/72	31		71/72	439	
Bouman et al. (2018)	16	30	10/10	720	0%	10/10	9	0%
	20	30	0/10	–	–	10/10	139	0%
	24	30	0/10	–	–	8/10	1115	3%
	28	30	0/10	–	–	2/10	3111	5%
	32	30	0/10	–	–	0/10	3600	10%
	Summary		10/50	–		30/50	1595	

Acknowledgements The authors gratefully thank the anonymous reviewers, the department editor, and the editor for their valuable recommendations, which have significantly improved our paper.

Funding Open Access funding enabled and organized by Projekt DEAL.

References

Agatz N, Bouman P, Schmidt M (2018) Optimization approaches for the traveling salesman problem with drone. Transp Sci 52(4):965–981. https://doi.org/10.1287/trsc.2017.0791

Boccia M, Mancuso A, Masone A, Sterle C (2023) A new MILP formulation for the flying sidekick traveling salesman problem. Networks 82(3):254–276. https://doi.org/10.1002/net.22172

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

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

Cavani S, Iori M, Roberti R (2021) Exact methods for the traveling salesman problem with multiple drones. Transp Res Part C Emerg Technol 130:103280. https://doi.org/10.1016/j.trc.2021.103280

Chang YS, Lee HJ (2018) Optimal delivery routing with wider drone-delivery areas along a shorter truck-route. Expert Syst Appl 104:307–317. https://doi.org/10.1016/j.eswa.2018.03.032

Dell'Amico M, Montemanni R, Novellani S (2021) Drone-assisted deliveries: new formulations for the flying sidekick traveling salesman problem. Optim Lett 15:1617–1648. https://doi.org/10.1007/s11590-019-01492-z

Dell'Amico M, Montemanni R, Novellani S (2021) Modeling the flying sidekick traveling salesman problem with multiple drones. Networks 78(3):303–327. https://doi.org/10.1002/net.22022

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)

Dell'Amico M, Montemanni R, Novellani S (2022) Exact models for the flying sidekick traveling salesman problem. Int Trans Oper Res 29(3):1360–1393. https://doi.org/10.1111/itor.13030

El-Adle AM, Ghoniem A, Haouari M (2021) Parcel delivery by vehicle and drone. J Oper Res Soc 72(2):398–416. https://doi.org/10.1080/01605682.2019.1671156

Freitas JC, Penna PHV, Toffolo TAM (2023) Exact and heuristic approaches to truck-drone delivery problems. EURO J Transp Logist 12:100094. https://doi.org/10.1016/j.ejtl.2022.100094

Gartner, J.: JD.com's Drone delivery program takes flight in Rural China. World Wide Web (2016). Last Access 17 May 2022

Ha QM, Deville Y, Pham QD, Hà MH (2018) On the min-cost traveling salesman problem with drone. Transp Res Part C Emerg Technol 86:597–621. https://doi.org/10.1016/j.trc.2017.11.015

İbroşka B, Özpeynirci S (2023) Özpeynirci: multiple traveling salesperson problem with drones: General variable neighborhood search approach. Comput Oper Res 160:106390. https://doi.org/10.1016/j.cor.2023.106390

Karak A, Abdelghany K (2019) The hybrid vehicle-drone routing problem for pick-up and delivery services. Transp Res Part C Emerg Technol 102:427–449. https://doi.org/10.1016/j.trc.2019.03.021

Kitjacharoenchai P, Ventresca M, Moshref-Javadi M, Lee S, Tanchoco JMA, Brunese PA (2019) Multiple traveling salesman problem with drones: mathematical model and heuristic approach. Comput Indus Eng 129:14–30. https://doi.org/10.1016/j.cie.2019.01.020

Miller CE, Tucker AW, Zemlin RA (1960) Integer programming formulation of traveling salesman problem. J ACM 7(4):326–329. https://doi.org/10.1145/321043.321046
Morandi N, Leus R, Matuschke J, Yaman H (2023) The traveling salesman problem with drones: the benefits of retraversing the arcs. Transp Sci 57(5):1340–1358. https://doi.org/10.1287/trsc.2022.0230
Moshref-Javadi M, Hemmati A, Winkenbach M (2020) A truck and drones model for last-mile delivery: a mathematical model and heuristic approach. Appl Math Model 80:290–318. https://doi.org/10.1016/j.apm.2019.11.020
Moshref-Javadi M, Lee S, Winkenbach M (2020) Design and evaluation of a multi-trip delivery model with truck and drones. Transp Res Part E Logist Transp Rev 136:101887. https://doi.org/10.1016/j.tre.2020.101887
Murray CC, Chu AG (2015) The flying sidekick traveling salesman problem: optimization of drone-assisted parcel delivery. Transp Res Part C Emerg Technol 54:86–109. https://doi.org/10.1016/j.trc.2015.03.005
Murray CC, Raj R (2020) The multiple flying sidekicks traveling salesman problem: parcel delivery with multiple drones. Transp Res Part C Emerg Technol 110:368–398. https://doi.org/10.1016/j.trc.2019.11.003
Otto A, Agatz N, Campbell J, Golden B, Pesch E (2018) Optimization approaches for civil applications of unmanned aerial vehicles (UAVs) or aerial drones: a survey. Networks 72(4):411–458. https://doi.org/10.1002/net.21818
Poikonen S, Golden B (2020) Multi-visit drone routing problem. Comput Oper Res 113:104802. https://doi.org/10.1016/j.cor.2019.104802
Rave A, Fontaine P, Kuhn H (2023) Drone location and vehicle fleet planning with trucks and aerial drones. Eur J Oper Res 308(1):113–130. https://doi.org/10.1016/j.ejor.2022.10.015
Roberti R, Ruthmair M (2021) Exact methods for the traveling salesman problem with drone. Transp Sci 55(2):315–335. https://doi.org/10.1287/trsc.2020.1017
Rave A, Fontaine P, Kuhn H (2023) Drone network design for emergency resupply of pharmacies and ambulances. Available at SSRN 4569199
Sacramento D, Pisinger D, Røpke S (2019) An adaptive large neighborhood search metaheuristic for the vehicle routing problem with drones. Transp Res Part C Emerg Technol 102:289–315. https://doi.org/10.1016/j.trc.2019.02.018
Salama M, Srinivas S (2020) Joint optimization of customer location clustering and drone-based routing for last-mile deliveries. Transp Res Part C Emerging Technol 114:620–642. https://doi.org/10.1016/j.trc.2020.01.019
Schermer D, Moeini M, Wendt O (2019) A matheuristic for the vehicle routing problem with drones and its variants. Transp Res Part C Emerg Technol 106:166–204. https://doi.org/10.1016/j.trc.2019.06.016
Schermer D, Moeini M, Wendt O (2020) A branch-and-cut approach and alternative formulations for the traveling salesman problem with drone. Networks 76(2):164–186. https://doi.org/10.1002/net.21958
Seifried, K.: The traveling salesman problem with one truck and multiple drones. Available at SSRN 3389306 (2019)
Tamke F, Buscher U (2021) A branch-and-cut algorithm for the vehicle routing problem with drones. Transp Res Part B Methodol 144:174–203. https://doi.org/10.1016/j.trb.2020.11.011
Tiniç GO, Karasan OE, Kara BY, Campbell JF, Ozel A (2023) Exact solution approaches for the minimum total cost traveling salesman problem with multiple drones. Transp Res Part B Methodol 168:81–123. https://doi.org/10.1016/j.trb.2022.12.007
Wang X, Poikonen S, Golden B (2017) The vehicle routing problem with drones: several worst-case results. Optim Lett 11(4):679–697. https://doi.org/10.1007/s11590-016-1035-3
Yu VF, Lin S-W, Jodiawan P, Lai Y-C (2023) Solving the flying sidekick traveling salesman problem by a simulated annealing heuristic. Mathematics 11(20):4305