Annals of Operations Research
https://doi.org/10.1007/s10479-025-06829-9

ORIGINAL RESEARCH

Truck drone arc covering problem with an application and case study in disaster management

Alexander Rave[1] · **Pirmin Fontaine**[1]

Received: 19 September 2024 / Accepted: 25 August 2025

Abstract
River exploration during, before, or after floods enables operators in civil protection and disaster control to better prepare for or even prevent disasters. While typically, this river exploration is done by boat, truck, helicopter, or even not at all, autonomous flying drones equipped with a camera can enhance this process. Moreover, interaction between a truck and a drone can enable the drone to be used flexibly and extend its short range. Thus, the Bavarian Red Cross equipped a truck with a drone for river coverage. Based on this real case, we introduce a truck drone arc covering problem (TD-ACP) for the application of river coverage. We formulate the TD-ACP as a mixed-integer linear program and introduce valid inequalities that strengthen the formulation and allow us to solve realistic-sized instances to optimality. In a real-world case study involving an actual river, we demonstrate that using drones for river coverage can reduce coverage time by 56.3% compared to boats and by 28.1% compared to trucks. Additionally, we propose a manual planning heuristic that is straightforward for practitioners to apply and achieves an optimality gap of 4.0% on this specific river.

Keywords Unmanned aerial vehicles · Routing · Mixed-integer linear program · Valid inequalities · Case study

1 Introduction

Inland floods are increasing, especially near rivers, leading to costly and humanitarian disasters (DRK, 2021). One example of this is the flood in the Ahr valley in Germany in 2021, in which 134 people died, 766 people were hurt (Landesregierung Rheinland-Pfalz, 2021),

✉ Alexander Rave
ARave@ku.de

Pirmin Fontaine
Pirmin.Fontaine@ku.de

[1] Ingolstadt School of Management and Mathematical Institute for Machine Learning and Data Science, Catholic University Eichstätt-Ingolstadt, Auf der Schanz 49, 85049 Ingolstadt, Germany

Published online: 15 September 2025

and overall economic damage amounted to 560 million Euro occurred (Der Spiegel, 2021). As a result, politicians and relief organizations, e.g., the Red Cross, are very keen to take measures to contain these disasters. One possibility is the river exploration in flood areas during, before, and after flood disasters. Covering the river during or before a float helps relief organizations to prepare better or even prevent disasters. For example, people can be rescued or evacuated, or dikes can be additionally strengthened. After disasters, relief organizations can get an overview of flood damages or even search for missing persons (Gstaiger et al., 2022; BRK, 2022). For these cases, it is of particular importance that the river is covered as fast as possible.

There are various possibilities for this river exploration. On the one hand, fixed sensors and cameras can be useful, but especially in more rural areas, where cameras cannot be placed, the river is not covered at all. Alternatively, it is driven next by car, by boat, or flying over by helicopter. However, this river coverage via boat is not always possible, e.g., during strong floods, and additionally requires multiple skilled workers. Another option would be pictures via satellites, but these require a cloudless sky, resulting in a delayed provision, and are limited in their depth of detail (Gstaiger et al., 2022).

Now, an aerial drone can enhance this process, as the drone can fly economically efficiently, fast, and autonomously (e.g., Gstaiger et al., 2022; Kippnich et al., 2022; Liu et al., 2019a) over the river to cover. Additionally, drones can take much better pictures than boats as they can cover a much larger area with one image and have a better angle (e.g., Nemer et al., 2020). This means that they can also cover far to the right and left of the river, which is particularly important in the event of a disaster (Gstaiger et al., 2022; Kippnich et al., 2022). Furthermore, launching a drone is relatively easy or can be carried out autonomously (e.g., Otto et al., 2018) in comparison to launching a boat, such that not many skilled workers are required. On the contrary, the drone's endurance is limited, and its flight is restricted for legal, data protection, and safety reasons. To overcome these disadvantages, a drone can be launched from a mobile truck (e.g., Murray & Chu, 2015) that is equipped with this drone. Therefore, the Bavarian Red Cross tested at the river Ahr whether its coverage via drone is practicable by equipping a pickup truck with a single drone (BRK, 2022; Quantum Systems, 2024). Because of the successful practical tests, the following questions arise: How does the optimal truck and drone routing look like, and how time-efficient is the use of drones in comparison to coverage via boat or truck?

Based on a real-life case study, we introduce a truck drone arc covering problem (TD-ACP) for river coverage. The peculiarity of covering a river is that the drone only travels certain subsequent arcs. Moreover, due to legal, data protection, and safety reasons, the drone is not allowed to leave the river arbitrarily. This means that it may not leave the river at a certain point just to enter at a far-away point, and the drone may only return to the truck node from the closest river point.

To solve this problem, we formulate the TD-ACP as a mixed-integer linear program (MILP) and introduce valid inequalities that enhance the MILP's runtime. These model enhancements allow us to solve instances with 105 nodes to optimality. For practitioners, we introduce and test a manual planning heuristic that enables them to plan truck and drone routing easily. In our numerical study, we analyze the routing and time savings for both the river Ahr and, in general, for synthetic sine-shaped rivers.

We contribute to the literature as follows: First, we introduce a new routing problem for river coverage, the TD-ACP. Second, we formulate the problem as an MILP and intro-

duce valid inequalities allowing us to solve realistic-sized instances to optimality. Third, we introduce a manual planning heuristic, which can also be applied in situations where there is no time or capacity to generate the full set of truck and drone nodes. Fourth, we introduce a novel realistic case study for the River Ahr and present managerial insights for it and rivers in general.

The structure of this paper is as follows: We formulate the problem setting in detail in Sect. 2. Then, in Sect. 3, we present the most relevant literature and delimit our contribution from it. Next, Sect. 4 introduces the MILP and valid inequalities. In Sect. 5, we present the manual planning heuristic for practitioners. We introduce the numerical study in Sect. 6 and, last, our results are summarized in Sect. 7.

2 Problem setting

2.1 Decision problem

We consider a problem, where a single river needs to be covered, i.e., overflown, by a single drone. The drone belongs to a truck that launches and picks up the drone at certain nodes where it fits best for both the truck and the drone. We decide on the routing of the truck and the drone, as well as the launch and rendezvous nodes, such that the makespan, i.e., the maximum of the completion times of the truck's and the drone's complete tour, is minimized.

2.2 Truck and drone routing

The truck has a supporting role and does not cover the river itself; it only launches and receives the drone. For this, the truck may visit arbitrarily many nodes at a maximum of once. The truck is synchronized with its drone, i.e., when the drone is launched, the truck and the drone can rendezvous again at the same node the drone has been launched from (including the depot) or a node the truck visits at a later stage within its tour. However, a return to the truck's next but one node would be unreasonable because of the truck's supporting role. It follows that the drone can perform loops (return to the same node it has been launched from) and sidekicks (return to a different node it has been launched from). When the drone performs a loop at the depot, we assume that this loop is performed at the end of the truck's tour. Note that a reasonable solution might be that the truck does not leave the depot, as the drone might cover the total river when launched at the depot (Rave, 2025).

Figure 1 illustrates the synchronization of the truck and the drone. On the left, the drone is launched at a node, and the drone returns to the same node from which it was launched. During this flight, the truck waits. Then, the truck continues its tour with the drone onboard to the next node, where another loop is performed. On the right, the drone is launched at a node and returns to the truck at a different node. Theoretically, drones may return to the same or any later node. However, a return to a different node than the same or the next node cannot be optimal, as the truck's task is to place the drone at advantageous locations and, then, a node visit must include a drone operation (launch or return of the drone).

Drone flights are limited by an endurance time per flight. When returning to the truck, the drone's battery is swapped with a loaded one, which the truck carries for a limited number. This is for practical and legal reasons, as commercial drones may only launch in

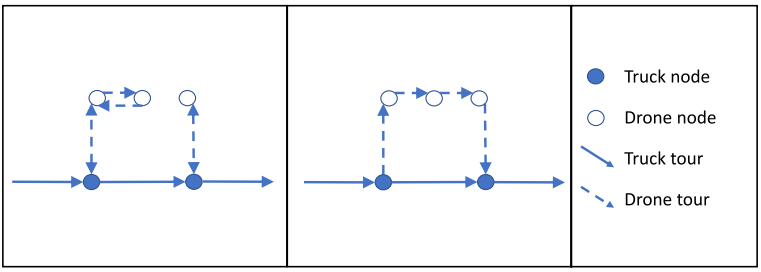

Fig. 1 Exemplary truck and drone synchronization

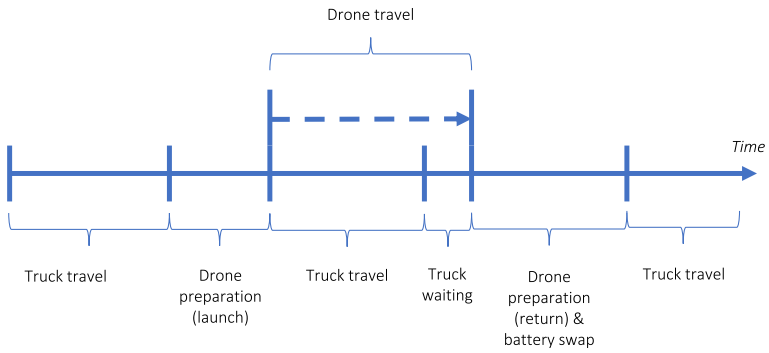

Fig. 2 Sequence and time of events for an exemplary routing

Germany if battery capacity is larger than 65% (Rave et al., 2023b). It follows that the drone can perform as many flights as there are batteries, since the truck cannot load the batteries. Moreover, there is a preparation time for launching and returning, and also a time for battery swap if the drone is planned to perform another trip after it has returned to the truck. Please note that the time required for launch preparation, return procedures, and battery swapping is not counted towards the drone's endurance limit, as these activities take place while the drone is on board the truck. In contrast, if the drone arrives at a node before the truck and has to wait, this waiting time is included in the endurance limit, since we assume the drone remains airborne during that period (e.g., Murray & Chu, 2015; Rave 2025).

Figure 2 illustrates the time that occurs during an exemplary tour. First, the truck travels to a launch node with the drone onboard. Then, the drone is launched, and preparation time ensues. The drone travels along the river. The drone's travel time equals the truck's travel time and the truck's waiting time for the drone. Preparation time occurs for the drone's return. Additionally, a time arises for a battery swap. Last, the truck continues its tour with the drone onboard. Please note that the drone might also wait for the truck. Then, the drone waiting time arises instead of the truck waiting time.

2.3 River coverage

To cover a river, we overlap the river by a fine raster of squares, with a node in the center (Nedjati et al., 2016). A peculiarity of river coverage is that all nodes to cover are set in a

Fig. 3 Picture of Ahr made by drone's equipped camera (Quantum Systems, 2024)

Fig. 4 Exemplary coverage of squares

line, as the drone's camera can cover the entire river width (see Fig. 3). Thus, we assume the river is covered if all arcs between direct neighbor nodes are traveled by the drone at least in one direction.

Figure 4 shows an example of a river overlapped by three squares, each with a node in its center and linked with arcs to the directly subsequent nodes. The river is covered if the drone travels arcs $(i, i+1)$ or $(i+1, i)$ and $(i+1, i+2)$ or $(i+2, i+1)$. A peculiarity of this problem is that the drone is not allowed to travel along arc $(i, i+2)$, as this would not ensure that the drone only flies over the river. This would be problematic due to legal, data protection, and safety reasons. Moreover, due to a potential river curvature, the river is not covered. Thus, the drone only travels to one of the two neighboring nodes within the river and does not skip these (Chen et al., 2017). Moreover, when returning to the truck, the drone can only leave the river at the closest river node to the truck node.

Figure 5 shows an exemplary routing plan for covering a river. The drone is launched twice to cover the river while the truck travels to three of the four truck nodes.

Note that approximating the river by considering only the end nodes and potential drone entry points, while omitting intermediate nodes, can lead to different or even infeasible solutions. This is because such an approximation ignores the possibility of dividing a single river subsection into two separate drone flights. In situations where the drone s endurance is insufficient to cover an entire subsection in one flight, representing the subsection as multiple shorter arcs ensures that feasible solutions remain possible.

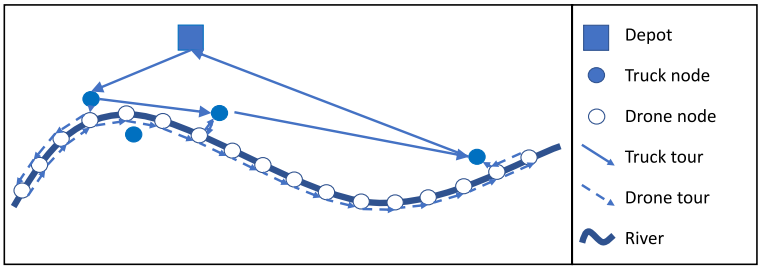

Fig. 5 Exemplary routing plan to cover a river

3 Literature review

In this section, we present the most relevant literature. First, in Sect. 3.1, we look at papers that consider drones visiting single nodes, and second, in Sect. 3.2, we present the literature that considers drones traveling multiple nodes. Third, in Sect. 3.3, we review publications on the two-echelon vehicle routing problem (2E-VRP). Last, in Sect. 3.4, we delimit our paper from the literature.

3.1 Drones visiting single nodes

Truck and drone tandems were initially introduced by Murray and Chu (2015) for parcel delivery, who consider a truck equipped with a single drone, and this drone may only visit a single node in its trip because of its limited payload. This problem setting has been extended by various publications, e.g., Sacramento et al. (2019), who consider multiple trucks equipped with a single drone with cost minimization, or Tamke and Buscher (2021), who consider multiple trucks and drones. Salama and Srinivas (2022) consider a variant where the truck can visit additional nodes that are not customers to launch drones, and Mirzapour Al-e-Hashem et al. (2024) take weather uncertainties into account. Due to the complexity of the synchronization between trucks and drones, Salama and Srinivas (2022), Mirzapour Al-e-Hashem et al. (2024), and Sacramento et al. (2019) present a heuristic, and Tamke and Buscher (2021) a branch-and-cut approach. On the contrary, Dell Amico et al. (2021), Freitas et al. (2023), or Rave (2025) focus on enhanced modeling approaches for MILP formulations. These formulations, however, are not capable of solving realistic-sized instances.

3.2 Drones visiting multiple nodes

Windras Mara et al. (2022) and Boccia et al. (2024) extend the truck-and-drone tandem of Murray and Chu (2015) with the drone traveling multiple nodes within its tour and present both a MILP and a heuristic. In this setting, the drone is not allowed to return to the same node from which it was launched. As the use case is still in parcel delivery, the number of subsequent drone nodes is rather low. Leaving parcel delivery, Liu et al. (2019a) consider the powerline inspection via drone that is launched from a truck, where a drone has to visit the complete powerline exactly once that consists of multiple straight arcs. The authors only present a heuristic solution approach. Liu et al. (2019b) extend this paper by considering

additionally a truck return to a different depot. The authors present a mathematical model that is nonlinear due to multiple absolute value functions. Moreover, they introduce a heuristic solution approach. Aljalaud et al. (2023) also consider a close use case of pipeline inspection, where all arcs, however, need to be traversed by multiple drones. No truck is considered.

Nedjati et al. (2016) introduce a general area coverage problem, where drones launched from fixed hubs have to cover certain areas. The authors decide, among other things, on the number of drones and the location of hubs. Xia et al. (2023) also consider a general area coverage problem, where drones launched from a truck have to fully cover certain areas flying in a lawn mowing or spiral pattern. Each drone can only cover a single area per flight, and the authors decide on the truck routing, the drone launching points, and whether the area is covered via lawn mowing or spiral patterns. Another comparable use case for truck-and-drone tandems is road patrol, where certain arcs and nodes need to be covered either by truck or by drone. Luo et al. (2019) consider this problem setting for a single drone, and Wu et al. (2022) and Xu et al. (2023) for multiple drones. Cai et al. (2025) introduce a location routing problem where drones are launched from multiple hubs to visit wind farms. The authors decide, among other things, on the number of spare batteries used.

3.3 2E-VRP

As our truck-and-drone tandem can be interpreted as a variant of a 2E-VRP, where the truck route is the first echelon and the drone route the second echelon, we now consider the literature on the 2E-VRP. Within the 2E-VRP, the second echelon's vehicles typically return to the same node where they have started their tours (e.g., Perboli et al., 2011); however, considering drones, these may return to the next node within the truck's tour. Li et al. (2020) introduce this problem, where multiple trucks equipped with multiple drones have to visit certain customers exactly once, either by truck or by drone. Rave et al. (2023a) also consider the two-echelon structure for their drone routing problem. However, the authors consider only a single customer visit per drone, but instead fixed hubs and trucks, each equipped with multiple drones. Dukkanci et al. (2023) introduce a 2E-VRP with drones for relief logistics after an earthquake. The peculiarity of this problem is that it includes demand and road network uncertainty. Again, the drone only visits a single node per flight.

3.4 Differentiation from the literature

Table 1 distinguishes our paper from the previously introduced literature. The table reports the number (0 = none, 1 = single, n = multiple) of trucks and drones considered and nodes per drone flight, as well as the methodology used. Further, it is mentioned if all arcs/nodes need to be visited at most once (set packing), exactly once (set partitioning), or at most once (set covering).

Our considered problem has elements of classical truck and drone tandems (e.g., Roberti & Ruthmair, 2021), but our drone routing is more complex as each drone tour includes more than a single outbound and return flight. Our problem extends a 2E-VRP with mobile hubs and the drone as a vehicle of the second echelon as we combine it with area coverage explicitly done by the drone. Moreover, the drone can only travel certain arcs due to legal, data protection, and safety reasons, which is also represented in our mathematical model.

Table 1 Differentiation from the literature

	#Trucks/#Drones	#Nodes per drone flight	Setting	Sole drone inspection	Methodology
Murray and Chu (2015)	1/1	1	Partitioning		MILP, heuristic
Dell Amico et al. (2021)	1/1	1	Partitioning		MILP*
Freitas et al. (2023)	1/1	1	Partitioning		MILP, heuristic
Salama and Srinivas (2022)	1/*n*	1	Partitioning		MILP, heuristic
Mirzapour Al-e-Hashem et al. (2024)	1/*n*	1	Partitioning		MILP, heuristic
Rave (2025)	1/*n*	1	Partitioning		MILP
Sacramento et al. (2019)	*n*/1	1	Partitioning		MILP, heuristic
Tamke and Buscher (2021)	*n*/*n*	1	Partitioning		MILP*, branch-and-cut
Rave et al. (2023a)	*n*/*n*	1	Partitioning		MILP, heuristic
Dukkanci et al. (2023)	*n*/*n*	1	Packing		MILP, heuristic
Nedjati et al. (2016)	0/*n*	*n*	Covering	✓	MILP*
Aljalaud et al. (2023)	0/*n*	*n*	Covering	✓	Heuristic
Cai et al. (2025)	0/*n*	*n*	Partitioning	✓	MILP*, Benders decomposition
Windras Mara et al. (2022)	1/1	*n*	Partitioning		MILP, heuristic
Boccia et al. (2024)	1/1	*n*	Partitioning		MILP*, branch-and-cut, heuristic
Liu et al. (2019a)	1/1	*n*	Partitioning	✓	heuristic
Liu et al. (2019b)	1/1	*n*	Partitioning	✓	MINLP***, heuristic
Xia et al. (2023)	1/1	**	Covering	✓	MILP, heuristic
Luo et al. (2019)	1/1	*n*	Covering		MILP, heuristic
Wu et al. (2022)	1/*n*	*n*	Covering		MILP, heuristic
Xu et al. (2023)	1/*n*	*n*	Covering		MILP, heuristic
Li et al. (2020)	*n*/*n*	*n*	Partitioning		MILP, heuristic
Our paper	1/1	*n*	Covering	✓	MILP*, heuristic

*Valid inequalities. **Drone flies spiral or lawn mowing pattern. ***Mixed integer non-linear program

Considering multiple visits per drone flight, as considered by, e.g., Liu et al. (2019a, b), we have multiple different assumptions. On the one hand, our drone flights are more restrictive by allowing the drone only to travel certain arcs, but on the other hand, our assumptions are less restrictive, e.g., the drone may also wait for the truck. In comparison to Luo et al. (2019), we consider more restrictive flight situations and drone coverage only. Therefore, in solutions of Luo et al. (2019), the drone only travels a few arcs, i.e., one to around five, in one flight. Within our solutions, this is up to 86 arcs in one flight. Moreover, we consider the special case where all arcs are set in a line. This allows us in our MILP formulation to formulate the drone routing without subtour elimination constraints; following that, we can solve realistic-sized instances to optimality.

Our problem setting with its assumptions has not been covered in the literature so far. Thus, MILP formulations and algorithms cannot simply be adopted to our problem settings. Main differences are the flexibility of trucks and drones waiting for each other (e.g., Liu et

al., 2019a, b) and the ability to return to the same or the next truck node it has been launched from. Furthermore, we have a finer division of the arcs so that two drone flights can cover river subsections of the flow if the endurance is no longer sufficient. Regarding heuristics for classical truck and drone tandems, e.g., the adaptive large neighborhood search of Sacramento et al. (2019), drone flights are limited to cover a single node per flight. Additionally, the number of batteries and, thus, drone flights is limited in our study, and drones must enter the river at the closest truck node. These are practical and necessary limitations and a peculiarity of our problem.

4 TD-ACP

We consider the set of truck nodes $\mathcal{I}^T = \{1, \ldots, n\}$ that may be visited by the truck at maximum once and the set of drone nodes $\mathcal{I}^D = \{n+1, \ldots, n+m\}$ that may be visited at any frequency. These nodes are ordered from the river's start to the river's end. As the drone cannot enter or leave the river to cover arbitrarily, some nodes might be visited multiple times. The set $\mathcal{I}_0 = \mathcal{I}_0^T \cup \mathcal{I}^D$ is the set of all nodes, where $\mathcal{I}_0^T = \mathcal{I}^T \cup \{0\}$ is the set of truck nodes including the depot. The arc set $A = \{(i,j) | i,j \in \mathcal{I}^D, j = i+1 \vee j = i-1\}$ includes all arcs that need to be covered by the drone in at least one direction. Arc set $\hat{A} = A \cup \{(i,j) | (i \in \mathcal{I}_0^T, j \in \mathcal{I}^D) \vee (i \in \mathcal{I}^D, j \in \mathcal{I}_0^T)\}$ further includes the arcs with nodes where the truck and the drone visit. Moreover, we consider the set of fully loaded batteries $\mathcal{D}$, which can also be interpreted as the number of drone tours, as the battery is swapped whenever the drone returns to the truck, but batteries are not reloaded on the truck.

The drone has a limited endurance e and needs certain traveling times $t_{i,j}^D$ when traveling arc $(i,j) \in \hat{\mathcal{A}}$. Similarly, the truck has certain traveling times $t_{i,j}^T$ $(i,j \in \mathcal{I}_0^T)$, too. Moreover, preparation s^P and battery swap times s^B arise, including times for launch, return, and battery swap, which do not affect endurance.

The main decision variables decide on the routing of the truck ($x_{i,j}$) and the drone ($y_{i,j,d}$). Variables $z_{i,j,d}$ show the launch and rendezvous node of the truck and the drone. Note that the drone may also fly loops, i.e., $z_{i,i,d} = 1$. Table 2 provides an overview of all sets, parameters, and variables.

Figure 6 shows the values of variables $x_{i,j}$ and $z_{i,j,d}$ for an exemplary routing plan to cover a river. For all arcs (i,j) that are not illustrated, the variables have a value of zero, e.g., $x_{1,2} = 0$. The truck travels from depot 0 to node 1 ($x_{0,1} = 1$), launches the drone, and both meet again at node 3, i.e., $x_{1,3} = 1$ and $z_{1,3,1} = 1$. At node 3, the battery is swapped, and the drone is launched again using its second battery. Drone and truck meet again at node 4, i.e, $x_{3,4} = 1$ and $z_{3,4,2} = 1$. Last, the truck returns to the depot ($x_{4,0} = 1$).

4.1 MILP

Objective and definition of makespan

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

Table 2 Index sets, parameters, and decision and auxiliary variables

Index sets	
$\mathcal{I}_0$	Node set for all considered nodes including depot 0.
$\mathcal{I}^T, \mathcal{I}_0^T$	Node set of truck nodes (including the depot 0).
$\mathcal{I}^D$	Node set of drone nodes starting with node $n+1$ and ending with node $n+m$.
$\mathcal{D}$	Set of batteries.
$\hat{\mathcal{A}}$	Arc set of all possible drone arcs.
$\mathcal{A}$	Arc set that needs to be traveled in at least one direction.
Parameters	
e	Maximum endurance per drone flight.
n	Number of truck nodes (without depot).
m	Number of drone nodes.
s^P, s^B	Drone's preparation and battery swap time.
$t_{i,j}^T$	Traveling times for the truck traveling from node $i \in \mathcal{I}_0^T$ to node $j \in \mathcal{I}_0^T$.
$t_{i,j}^D$	Traveling times for the drone traveling arc $(i,j) \in \hat{\mathcal{A}}$.
Variables	
$x_{i,j}$	Truck tour $i, j \in \mathcal{I}_0^T$.
$y_{i,j,d}$	Drone tour $(i,j) \in \hat{\mathcal{A}}, d \in \mathcal{D}$.
$z_{i,j,d}$	Drone released at node $i \in \mathcal{I}_0^T$ for tour $d \in \mathcal{D}$ to return at node $j \in \mathcal{I}_0^T$.
τ	Makespan of both truck and drone.
v_i^T	Subtour elimination variable (Miller et al., 1960).
$\phi_{j,d}^T$	Truck's waiting time at node $j \in \mathcal{I}_0^T$ for the drone for drone flight $d \in \mathcal{D}$.

Note that binary variables have a value of 1 if they are true and 0 otherwise

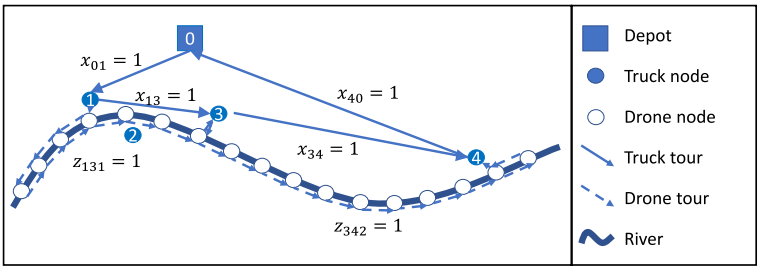

Fig. 6 Main decision variables for an exemplary routing plan to cover a river

$$s.t.$$

$$\tau \geq \sum_{i,j \in \mathcal{I}_0^T} t_{i,j}^T \cdot x_{i,j} + \sum_{j \in \mathcal{I}_0^T, d \in \mathcal{D}} \phi_{j,d}^T + s^P \cdot \sum_{i,j \in I_0^T, d \in D} z_{i,j,d} + s^B \cdot \left(\sum_{i,j \in I_0^T, d \in D} z_{i,j,d} - 1 \right) \quad (2)$$

The objective is to minimize the makespan τ, i.e., the time both the truck and the drone have returned to the depot starting at time 0. τ is set in Constraint (2) as the truck's traveling time (first sum) plus the truck's waiting time on the drone (second sum) plus the preparation time

for each drone flight (third sum) plus the battery swap time if a next flight is planned. Preparation times for launch, return, and battery swap may only influence the number of batteries used, i.e., fewer drone tours are preferred.

Area coverage

$$\sum_{d\in\mathcal{D}} y_{i,j,d} + y_{j,i,d} \geq 1 \qquad \forall (i,j) \in \mathcal{A} \tag{3}$$

The TD-ACP is formulated as a set covering problem (Constraints (3)). Thus, all arcs $(i,j) \in \mathcal{A}$ need to be traveled in at least one direction.

Truck routing

$$\sum_{i\in\mathcal{I}_0^T} \left(x_{i,j} - x_{j,i}\right) = 0 \qquad \forall j \in \mathcal{I}_0^T \tag{4}$$

$$v_i^T + 1 \leq v_j^T + |\mathcal{I}_0^T| \cdot (1 - x_{i,j}) \qquad \forall i,j \in \mathcal{I}^T \tag{5}$$

$$\sum_{j\in\mathcal{I}_0^T} x_{0,j} \leq 1 \tag{6}$$

Constraints (4) ensure that the truck's tour conserves flow. Subtours are eliminated in Constraints (5) (Miller et al., 1960). Constraint (6) ensures that there is at most one truck. Note that a feasible solution might be for the truck not to leave the depot and the drone to only perform one or multiple loops from it.

Drone routing

$$\sum_{i,j\in\mathcal{I}_0^T} z_{i,j,d} \leq 1 \qquad \forall d \in \mathcal{D} \tag{7}$$

$$x_{i,j} \geq \sum_{d\in\mathcal{D}} z_{i,j,d} \qquad \forall i,j \in \mathcal{I}_0^T : i \neq j \tag{8}$$

$$\sum_{i\in\mathcal{I}_0^T} x_{i,j} \geq z_{j,j,d} \qquad \forall j \in \mathcal{I}^T, d \in \mathcal{D} \tag{9}$$

$$\sum_{i\in\mathcal{I}_0:(i,j)\in\hat{\mathcal{A}}} \left(y_{i,j,d} - y_{j,i,d}\right) = 0 \qquad \forall j \in \mathcal{I}^D, d \in \mathcal{D} \tag{10}$$

$$y_{i,i+1,d} \leq y_{i+1,i+2,d} + \sum_{j\in\mathcal{I}_0^T:(i+1,j)\in\hat{\mathcal{A}}} y_{i+1,j,d} \qquad \forall i \in \mathcal{I}^D, d \in \mathcal{D} : i < n+m-1 \tag{11}$$

$$y_{i,i-1,d} \leq y_{i-1,i-2,d} + \sum_{j \in \mathcal{I}_0^T : (i-1,j) \in \hat{\mathcal{A}}} y_{i-1,j,d} \qquad \forall i \in \mathcal{I}^D, d \in \mathcal{D} : i > n+2 \tag{12}$$

$$\sum_{k \in \mathcal{I}^D : (i,k) \in \hat{\mathcal{A}}} y_{i,k,d} + \sum_{k \in \mathcal{I}^D : (k,j) \in \hat{\mathcal{A}}} y_{k,j,d} \geq z_{i,j,d} \qquad \forall i,j \in \mathcal{I}_0^T, d \in \mathcal{D} \tag{13}$$

$$\sum_{k \in \mathcal{I}^D : (i,k) \in \hat{\mathcal{A}}} y_{i,k,d} + \sum_{k \in \mathcal{I}^D : (k,j) \in \hat{\mathcal{A}}} y_{k,j,d} \leq z_{i,j,d} + 1 \qquad \forall i,j \in \mathcal{I}_0^T, d \in \mathcal{D} \tag{14}$$

The drone may perform at most one sidekick per node (Constraints (7)), but loops are not restricted. Considering sidekicks, these nodes need to be visited by the truck in a direct sequence (Constraints (8)). Constraints (9) ensure that a drone only performs loops at a node the truck has visited. Constraints (10) are flow conservation constraints for the drone. The drone may only return to the truck or travel subsequent (Constraints (11)) or previous nodes (Constraints (12)). These constraints make complex and time-consuming subtour elimination constraints unnecessary. However, area coverage in general, e.g., covering lakes, could be represented by this MILP formulation if Constraints (11) and (12) are replaced by subtour elimination constraints. In the "Appendix", we show how this could be formulated.

Constraints (13) and Constraints (14) connect the drone tour variables $y_{i,j,d}$ with the variable $z_{i,j,d}$ indicating if there is a drone flight from i to j. If $z_{i,j,d} = 1$, there is a drone flight (Constraints (13)). Conversely, if there is a drone flight, then $z_{i,j,d} = 1$ (Constraints (14)).

Synchronization

$$\phi_{j,d}^T \geq \sum_{(i,k) \in \hat{\mathcal{A}}} \left(t_{i,k}^D \cdot y_{i,k,d}\right) - \sum_{i \in \mathcal{I}_0^T} \left(t_{i,j}^T \cdot x_{i,j}\right) - e \cdot \left(1 - \sum_{i \in \mathcal{I}_0^T : i \neq j} z_{i,j,d}\right) \qquad \forall j \in \mathcal{I}_0^T, d \in \mathcal{D} \tag{15}$$

$$\phi_{j,d}^T \geq \sum_{(i,k) \in \hat{\mathcal{A}}} \left(t_{i,k}^D \cdot y_{i,k,d}\right) - e \cdot (1 - z_{j,j,d}) \qquad \forall j \in \mathcal{I}_0^T, d \in \mathcal{D} \tag{16}$$

$$\phi_{j,d}^T \leq e - \sum_{i \in I_0^T : i \neq j} t_{i,j}^T \cdot z_{i,j,d} \qquad \forall j \in \mathcal{I}_0^T, d \in \mathcal{D} \tag{17}$$

The truck's waiting time $\phi_{j,d}^T$ at a node $j \in \mathcal{I}_0^T$ is set in Constraints (15) if the drone performs a sidekick and in Constraints (16) if the drone performs a loop. The truck's waiting time may be at most the endurance e without considering the truck's travel time to node j (Constraints (17)). Note that the truck's waiting time $\phi_{j,d}^T$ must be set for each node j to be

the link between $y_{k,l,d}$ and $z_{i,j,d}$. As multiple returns at a node j are possible, $\phi^T_{j,d}$ has each drone flight d as an additional index.

Definition of variables

$$\sum_{i \in \mathcal{I}_0^T} x_{i,i} = 0 \tag{18}$$

$$x_{i,j}, z_{i,j,d} \in \{0,1\}, \phi^T_{j,d}, \tau \in \mathbb{R}^+ \qquad \forall i,j \in \mathcal{I}_0^T, d \in \mathcal{D} \tag{19}$$

$$y_{i,j,d} \in \{0,1\} \qquad \forall (i,j) \in \hat{\mathcal{A}}, d \in \mathcal{D} \tag{20}$$

$$v_i^T, w_j \in \{0,1\} \qquad \forall i \in \mathcal{I}^T, j \in \mathcal{I}^D \tag{21}$$

Last, variables are defined.

4.2 Valid inequalities

We further introduce valid inequalities to reduce the runtime of solving the MILP using a standard solver. We show the impact of valid inequalities in Sect. 6.2.2. Please note that valid inequalities (22)–(26) also fit for area coverage in general as shown in the "Appendix".

Lower bound

$$\tau \geq t^{min} \tag{22}$$

Constraint (22) limits the makespan τ by a lower bound t^{min} that is determined in a preprocessing step. This constraint is expected to increase the value of the linear relaxation at the root node. We set t^{min} as follows. It equals a theoretical (truck-independent) drone's round trip, which neglects the endurance limit and arc limitations:

$$t^{min} = t^D_{0,n+1} + \frac{\sum_{(i,j) \in \mathcal{A}} t^D_{i,j}}{2} + t^D_{n+m,0} + \left\lceil \frac{\sum_{(i,j) \in \mathcal{A}} t^D_{i,j}}{2 \cdot e} \right\rceil \cdot (s^P + s^B)$$

t^{min} equals the time the drone travels to the first river's node starting from the depot, plus the shortest time for covering the area, plus the time for returning to the depot. In addition, a minimum number of preparation and battery swap times $s^P + s^B$ is added. Note that if the truck travels faster than the drone, e.g., because of a highway, then the travel times $t^D_{i,j}$ need to be adjusted by a drone-truck speed ratio.

Makespan redefinition

$$\tau \geq \sum_{i,j \in \mathcal{I}_0^T} t_{i,j}^T \cdot x_{i,j} + s^P \cdot \sum_{i,j \in I_0^T, d \in D} z_{i,j,d} + s^B \cdot \left(\sum_{i,j \in I_0^T, d \in D} z_{i,j,d} - 1 \right) + \sum_{(i,j) \in \hat{\mathcal{A}}, d \in \mathcal{D}} t_{i,j}^D \cdot y_{i,j,d} + \sum_{d \in \mathcal{D}} \phi_d^D - \sum_{i,j \in \mathcal{I}_0^T, d \in \mathcal{D}: i \neq j} t_{i,j}^T \cdot z_{i,j,d} \tag{23}$$

$$\phi_d^D \geq \sum_{i,j \in \mathcal{I}_0^T: i \neq j} t_{i,j}^T \cdot z_{i,j,d} - \sum_{(i,j) \in \hat{\mathcal{A}}} t_{i,j}^D \cdot y_{i,j,d} \qquad \forall d \in \mathcal{D} \tag{24}$$

$$\phi_d^D \in \mathbb{R}^+ \qquad \forall d \in \mathcal{D} \tag{25}$$

While in Constraint (2), τ is set from the truck's perspective, Constraint (23) sets τ additionally from the drone's perspective, i.e., the truck's total traveling time (first sum) plus the preparation time for each drone flight (second sum) plus the battery swap time if a next flight is planned (third sum) plus the drone traveling (fourth sum) and its waiting time (fifth sum). When the drone performs a sidekick, i.e., the truck picks up the drone at a different node, the truck traveling time between both nodes is subtracted as it is already included in the drone traveling and waiting time (sixth sum). Constraints (24) introduce the drone's waiting time for each flight. The drone waits for the truck if it arrives earlier at a node when performing a sidekick. In the solutions, we observe that in some instances, the drone waits, albeit not for a long time compared to the truck's waiting time. Note that Constraints (2) and (23) are equivalent, and for runtime improvements, both should be used. The redefinition of τ is an idea inspired by Rave (2025) for truck-and-drone tandems, who found that it significantly lowers the solver's runtime.

Node visit with drone operations

$$\sum_{i \in \mathcal{I}_0^T} x_{i,j} \leq \sum_{i \in \mathcal{I}_0^T, d \in \mathcal{D}} (z_{i,j,d} + z_{j,i,d}) \qquad \forall j \in \mathcal{I}^T \tag{26}$$

A feasible solution might be that the truck visits nodes without launching or returning the drone; however, this may only result in delays and thus can never be optimal. Thus, Constraints (26) ensure that when the truck is traveling to a node $j \in \mathcal{I}^T$, then a drone launches from or returns to that node. Note that these constraints only hold if $t_{i,j}^T \leq t_{i,k}^T + t_{k,j}^T$, which is, however, given in our numerical study.

Arc limitations

$$\sum_{i \in \mathcal{I}_0: (i,j) \in \hat{\mathcal{A}}} y_{i,j,d} \leq 2 \qquad \forall j \in \mathcal{I}^D, d \in \mathcal{D} \tag{27}$$

Each node has two outgoing arcs within $\mathcal{A}$ (except the first and the last node). Thus, each node can only be visited twice, which is ensured by Constraints (27).

5 Manual planning heuristic

In this section, we introduce a simple heuristic procedure that can be applied in practice manually. This is particularly important from a practical point of view, as the MILP might be complex to implement, even if written down, has runtimes of just under an hour (depending on the instance), and requires exact geographical coordinates between the nodes. The MILP, on the other hand, is required to generate optimal solutions and for benchmark tests for the following heuristics. This heuristic is set as follows and ensures that the lowest number of drone flights is used:

1. The truck travels to the node that is closest to one end of the river with the drone onboard.
2. The drone is released at that node, traveling to one river's end node. Then, the drone turns 180 degrees in its direction.
3. The drone flies as far as the endurance is not exceeded, and the truck and drone meet again at the farthest distant node.
4. The drone is released again at the same node and continues its tour. Continue with step 3 until the river is fully covered. Then, the truck and drone travel back to the depot.

Note that if the depot is the closest node to one end of the river, the drone directly launches at or returns to the depot. Figure 5 in Sect. 2 shows a solution generated by applying this heuristic. We show the performance of this heuristic in comparison to optimal solutions generated by our MILP in Sect. 6.2.3.

6 Case study

In this section, we first introduce the real case of the Bavarian Red Cross. Next, we show the impact of our valid inequalities, and third, we present our numerical results. The MILP is implemented in OPL and solved using CPLEX version 22.1.1. All experiments are conducted on an AMD Ryzen 9 5950X with 128 GB RAM.

6.1 Numerical setup

The dedicated drone is the Trinity F90+, which has a constant speed of $17m/s$ or $61.2km/h$ and an endurance of $e = 90$ minutes. This drone is ideally suited for this use case as it is a fixed-wing drone with tilt rotors that flies safely and quietly and is typically equipped with a camera (Quantum Systems, 2024). We assume that the drone directly flies the Euclidean path. However, when the drone returns to the truck, it should not fly freely in the terrain for safety reasons but follow the river (Chen et al., 2017). We consider a launch and return time of one minute each and a battery swap time of two minutes, i.e., $s^P = s^B = 2$.

6.1.1 River Ahr

The river Ahr is an 85.1km long river in Germany, which flows into the Rhine and was affected by a flood disaster in 2021. We split up the river into 85 squares, each with a length

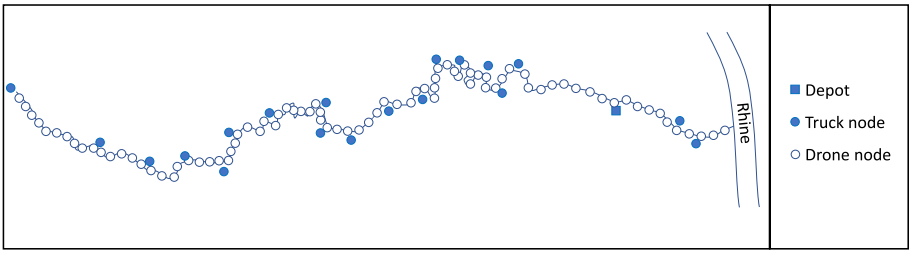

Fig. 7 Model of the river Ahr with all truck and drone nodes

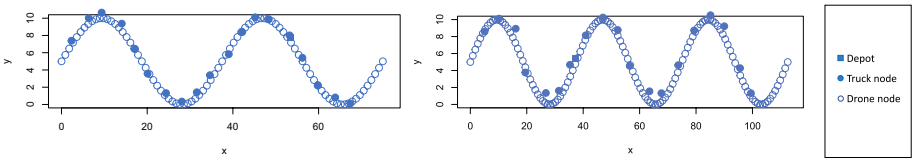

Fig. 8 Example of the sine shaped river with a period of 4π (left) a period of 6π (right)

of 1 km, with a node in the center, i.e., each arc has a length of one km. The truck nodes are next to 19 villages or cities, respectively, next to the Ahr, and were affected by the flood disaster. The depot is located at the center of the German Red Cross, which is also quite close to the River Ahr. The pickup truck has travel times taken from Google Maps for the case study of the river Ahr. Figure 7 shows the model of the river Ahr with all truck (blue-filled circles) and drone nodes (unfilled circles). The blue square indicates the depot.

6.1.2 Synthetic rivers

We also generate sine-shaped rivers (e.g., Song et al., 2016) to test our MILP on more data and show the effects of shorter or longer rivers. We consider ten sine-shaped rivers with a period of 4π that are stretched within a certain area of 75×10 km. These rivers are equal in length to the river Ahr. In a sensitivity analysis, we vary the length of these rivers (both the size of the area and the period) by a factor of 0.5, 1.5, 2, and 2.5, resulting in a period of 2π, 6π, 8π, and 10π. Within these instances, we consider $n = 19$ truck nodes (plus depot), and the rivers are approximated by one node each km. The truck is assumed to have an average speed of 35 km/h and travels the Euclidean distance corrected by a factor of 1.1. Figure 8 shows an example of the sine-shaped river with a period of 4π (left) and a period of 6π (right) with truck and drone nodes. Note that with ten instances and five different periods, a total of 50 synthetic instances is considered in our case study.

6.2 River approximation and performance tests

In this section, we test our chosen node number to approximate the rivers and test our valid inequalities and the manual planning heuristic.

6.2.1 Determining the number of nodes to approximate a river

First, we show that in our case study, our chosen number of nodes does not distort our objective function values, as with too few nodes, river curvature is neglected. For this, we optimally solve an instance with $|\mathcal{D}| = 2$ and an approximated sine-shaped river (4π) with an increasing number of drone nodes starting with 20 and ending with 300. An approximation by 85 nodes equals an arc length of one km between two nodes next to each other.

Figure 9 shows the objective values with an increasing number of drone nodes $\mathcal{I}^D$. The dotted red line indicates the objective value considering 300 nodes, which is expected to be close to convergence. The objective value deviations fluctuate due to the increasing consideration of the river curvature and because of the position at which the drone enters the river. From 80 nodes onwards, the deviations are rather small, with 80 nodes having a deviation of less than 0.05% compared to considering 300 nodes. Therefore, we find that considering at least 80 nodes is sufficient to approximate rivers. For practical reasons, 85 nodes are chosen because then the distance between the node centers is equal to one km.

6.2.2 Impact of valid inequalities

We now show that the presented valid inequalities reduce the runtime of the solver. For this, we take the ten instances with a period of 4π for the synthetic rivers with $|\mathcal{D}| = 3$ and solve the MILP (runtime limit of 1 h) with different added valid inequalities. Table 3 shows the average upper and lower bound found, the resulting optimality gap, the number of optimal solutions found, and the average runtime in seconds when solving the MILPs.

We solve the MILP without any valid inequalities, with each of the four valid inequalities individually, and with all valid inequalities within a runtime limit of 1 h. Moreover, we test all valid inequalities, except the lower bound (Constraint (22)), as its impact on the performance cannot be observed directly.

Without considering any valid inequalities, three out of the ten instances could be solved to optimality with an average gap of 13.0%. We observe that Constraints (23)–(27) improve the solver's performance with the makespan redefinition (Constraints (23)–(25)) having the largest improvements. As a result, up to nine instances could be solved to optimality, and the optimality gap could be reduced significantly, including each of these valid inequalities individually. Constraint (22), however, reduces the optimality gap but also increases the runtime, following that no instance could be solved to optimality. Thus, we additionally tested considering all valid inequalities and when not including Constraint (22). We find that we can only solve all instances optimally if Constraint (22) is included, also having the

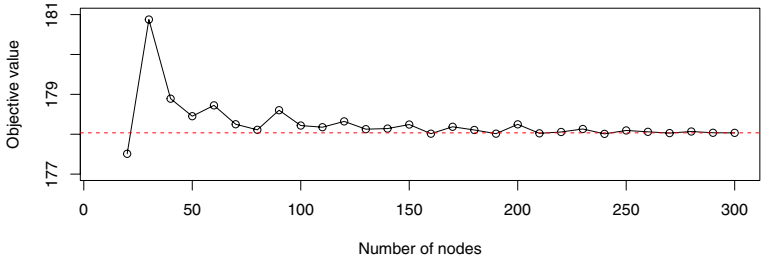

Fig. 9 Objective values with increasing number of drone nodes

Table 3 Impact of valid inequalities

	Upper bound	Lower bound	Gap [%]	#Opt.	CPU [s]
MILP without additional constraints	183.6399	159.8401	13.0%	3/10	3045
MILP with Constraint (22)	183.6399	163.8912	10.8%	0/10	3600
MILP with Constraints (23)–(25)	183.6399	181.6070	1.1%	9/10	407
MILP with Constraints (26)	183.6399	180.5437	1.7%	9/10	737
MILP with Constraints (27)	183.7764	164.5497	10.5%	7/10	2047
MILP with Constraints (23)–(27)	183.6399	182.4664	0.6%	9/10	438
MILP with Constraints (22)–(27)	183.6399	183.6399	0.0%	10/10	318

Table 4 Benchmark tests

	Heuristic		MILP		
	Objective	# Batteries	Objective	# Batteries	Δ [%]
River Ahr	160.1	1	153.7	2	4.0
Sine-shaped river (2π)	94.9	1	84.7	2	10.8
Sine-shaped river (4π)	192.8	2	183.6	2	4.8
Sine-shaped river (6π)	333.9	2	327.1	2.3	2.0
Sine-shaped river (8π)	502.4	3	472.3	3.2	6.0
Sine-shaped river (10π)	673.8	4	618.6	4.2	8.2
Average	355.6		333.7		6.2

lowest runtime on average. As the average runtime is significantly reduced from 3045 s on average to a tenth of 318 s, we include all valid inequalities in our numerical study in the following.

6.2.3 Performance tests of manual planning heuristic

Next, we benchmark the manual planning heuristic introduced in Sect. 5 against optimal solutions of our MILP formulation. Table 4 shows the average objective values and number of batteries used applying this heuristic and for our MILP formulation, when considering the river Ahr and the sinus-shaped rivers with a period of 2π, 4π, 6π, 8π, and 10π. Last, the table presents the gap Δ between our MILP formulation and the heuristic. Please note that for the synthetic rivers, the average of ten instances is reported. Each instance has the same number of truck nodes (19) and drone nodes (85). The instances vary in the location of truck nodes and the location of the depot. Moreover, the network size increases with larger river periods.

Considering the river Ahr, a 4% worse solution was found compared to the optimal solution. This is mainly because only one drone flight is performed in the heuristic solution, but carrying out two drone flights improves the solution. Considering the MILP with $|\mathcal{D}| = 1$, there is, however, still a gap of 1.1% as a different node than the closest to one river's end would be better.

The gap is similar when considering a sine-shaped river with a period of 4π but significantly larger with a smaller river (2π). This is because the launch and return of the first drone's tour are chosen inappropriately with the heuristic. The gap decreases for rivers with a period of 6π because multiple drone tours are required, so the disadvantages of the first and last truck node selection become less relevant. Conversely, for river periods of 8π and 10π, the optimality gap increases again. This is primarily due to the fixed number of 19 truck nodes: as the river length grows, a suboptimal choice of launch or return node by the heuristic has a growing impact on the objective. Nevertheless, our manual planning heuristic performs reasonably well, yielding an average optimality gap of 6.2%.

Our MILP formulation enables the pre-evaluation of routing strategies in areas at risk of flooding. Thanks to valid inequalities that significantly strengthen the formulation, even larger instances can be solved to optimality within an hour. However, in actual disaster situations, time is of critical importance, and there is no opportunity for data generation in flooded areas if it has not been done before the disaster occurs. In such cases, our manual planning heuristic proves useful. Its key advantage lies in its simplicity, which allows routing decisions to be made by hand. Despite its simplicity, the heuristic achieves relatively low optimality gaps and is therefore highly practical for real-world applications.

For larger rivers that exceed the capabilities of our MILP formulation, a metaheuristic (e.g., a genetic algorithm) could support the pre-evaluation phase. This aspect, however, is beyond the scope of this paper, as our case study based on the River Ahr and the considered synthetic instances, even with 2.5 times the size, could be solved to optimality in under 1 h.

6.3 Managerial insights

In this section, we present managerial insights based on the river Ahr and synthetic rivers. For this, we show the time savings a drone has compared to coverage via boat, a coverage via truck, and the optimal number of drone flights.

To illustrate river coverage via boat, we assume the following: The boat is placed next to the depot at the river to cover, which is only a short walk away. The preparation time for launch and return is increased by a factor of 10 ($s^{boat} = 20$), which also includes the time traveling to the boat. On the contrary, no battery swap is necessary. The endurance is no longer limited ($e = \infty$), and a speed of 30 km/h is considered. As a result, the time for river coverage via boat is computed simply by $\tau^{boat} = \sum_{(i,j)\in\mathcal{A}} t^D_{i,j} \cdot \frac{61.2}{30} + s^{boat}$, which is the sum of the traveling time for all subsequent arcs increased by the drone and boat speed ratio plus the preparation and return time s^{boat}. Note that this speed typically only holds if the river is navigable by boat; otherwise, in practice, a much lower speed is considered, or the boat cannot cross the river at all. Thus, the results presented in comparison to the boat show the least time savings from drone usage.

To illustrate coverage via a truck that observes the river by traveling next to it, we assume that there is a road directly next to the river. Thus, to cover the river, the truck traverses

to one end of the river and travels next to the river to the other end of the river. Then it returns to the depot. As a result, the time for river coverage via truck is computed simply by $\tau^{truck} = t^T_{0,n+1} + \frac{1}{2} \cdot \sum_{(i,j)\in\mathcal{A}} t^T_{i,j} + t^T_{n+m,0}$. Note that for this computation, $t^T_{i,j}$ must be defined for all $(i, j) \in \hat{\mathcal{A}}$. In practice, certain detours must be taken into account for the truck; thus, the results show the least time savings from drone usage in comparison to truck coverage.

6.3.1 Covering the river Ahr

Figure 10 shows the time savings when covering the river via drone in comparison to boat and truck coverage for different numbers of batteries used, starting with a single battery, i.e., one flight to cover the river. For this, we set the MILP's runtime to 1 h, and all MILPs could be solved to optimality in one second, considering a single battery, and 2740 s, considering four batteries.

We find that the river can be covered by a drone with a single battery, resulting in significant time savings of 56.3% in comparison to coverage via boat and 26.0% in comparison to coverage via truck. Moreover, considering additional batteries leads only to small additional makespan reductions of up to 1.3 (for boat) or 2.1 (for truck) percentage points, if at all. So, an improvement hardly offsets the runtime increase of route optimization with more batteries. Covering the river Ahr, the drone needs a total of 158.3 min, which is contrasted by 362.7 min for boat coverage and 213.9 min for a theoretical truck coverage. Please note that these time savings might be much larger, as the boat cannot always navigate the river during or after a flood, and there might not always be a road next to the river for the truck.

Figure 11 illustrates the optimal truck and drone routing considering $|\mathcal{D}| = 2$. In this case, the drone starts its first tour at the depot and returns to the truck at the second outermost truck node. Then, the truck travels (with the drone onboard) to the other end of the river and launches the drone for its second tour. Both the truck and the drone meet again at the depot. The truck arrives earlier than the drone at both nodes, resulting in waiting times of 3 and 20 min.

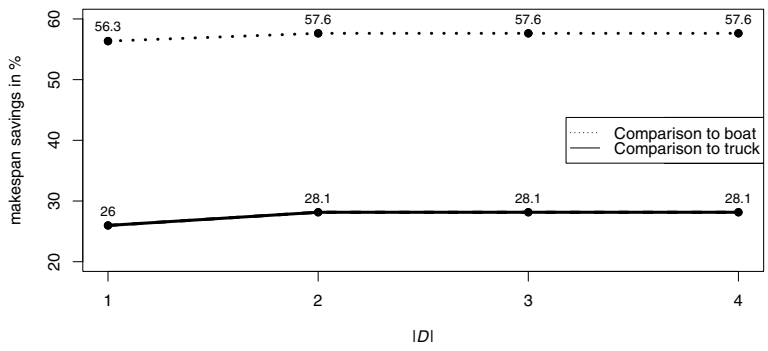

Fig. 10 Makespan savings when using a drone instead of a boat and a truck for river coverage with an increasing number of batteries

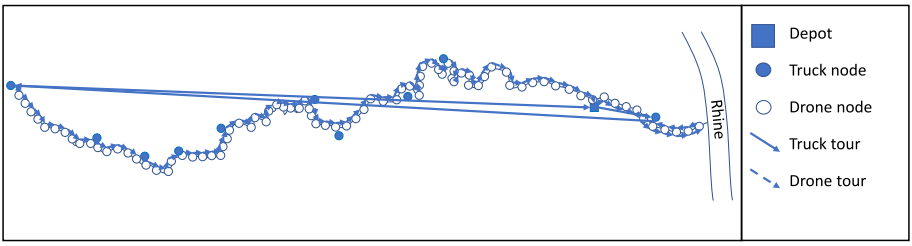

Fig. 11 Optimal solution covering the river Ahr with $|\mathcal{D}| = 2$

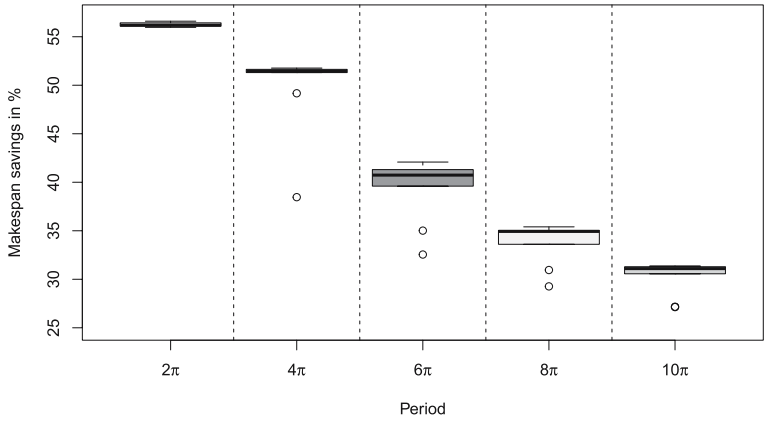

Fig. 12 Time savings of drone in comparison to boat coverage

6.3.2 Covering synthetic rivers

To generalize the results, we also consider synthetic rivers of varying lengths. Again, we compare the results of drone coverage to boat and truck coverage. The number of drone tours $|\mathcal{D}|$ is set sufficiently high, and all instances are solved to optimality with a runtime limit of 1 h. Figure 12 shows the makespan savings in comparison to boat coverage for the five different river lengths in the form of a boxplot diagram. Note that each boxplot consolidates ten instances.

Using a drone, average time savings are 56% (2π), 50% (4π), 40% (6π), 34% (8π), and 30% (10π). Apart from a few outliers, similar time savings are achieved for each river length in each instance, i.e., the standard deviation is relatively low. The time savings are reduced significantly as river length increases, despite the boat traveling along each arc in both directions, whereas the drone only does this at the outer points. The drone needs more time to cover the river because the total drone preparation time adapts to the boat with increasing river length, as one drone flight is optimal for a river length of 2π, two drone flights for 4π, 2–3 drone flights for 6π, 3–4 drone flights for 8π, and 4–5 drone flights for 10π. Please note that a solution with fewer drone flights, e.g., two instead of three, might still be feasible, but a few additional time savings can be achieved by having more drone flights.

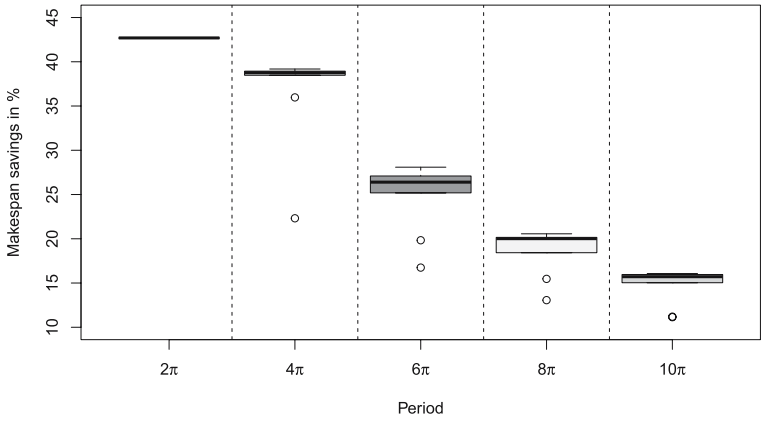

Fig. 13 Time savings of drone in comparison to truck coverage

Similarly, Fig. 13 shows the makespan savings of drone usage in comparison to coverage via a truck that traverses next to the river. Makespan can be reduced by up to 42.7% when using a drone. However, similarly to boat coverage, savings decrease with longer rivers. The biggest advantage of the drone compared to truck coverage is that the faster drone can avoid unnecessarily long and slow truck trips to the river's ends. However, as the length of the river increases, these detours account for a smaller proportion of the total time, so the time advantages decrease.

7 Conclusion

Inspired by a real case, we introduced the truck drone arc covering problem for river coverage, where a single drone launched from a truck must cover a river completely. We formulated the problem as an MILP and introduced valid inequalities that strengthened the formulation. These valid inequalities led to significant runtime improvements so that the runtime could be reduced to a tenth, following that all considered instances could be solved to optimality. Moreover, we introduced and tested a manual planning heuristic that practitioners can easily apply. In benchmark tests, we observed that this heuristic has an average deviation of 6.2% from the optimal solution.

In our case study based on covering the river Ahr, we find that drone usage leads to time savings of 56.3% in comparison to boat coverage, and 28.1% in comparison to truck coverage if the truck directly travels next to the river. Moreover, in a sensitivity analysis, we observed that these time savings decrease as the river length increases.

Future research could examine the benefits of using multiple drones per truck or multiple trucks equipped with one or multiple drones each. Moreover, retraversing certain arcs for the truck might be a reasonable option (Morandi et al., 2023) if there are only limited truck nodes available, which might be the case during floods. Alternatively, the drone could also take off from a moving truck (Thomas et al., 2024). Another interesting topic would be the search for missing persons after a disaster. This, however, requires a dynamic consideration of time, as people could be swept down the river during the operation.

In practical applications, factors such as weather conditions (e.g., precipitation) or reduced drone endurance due to strong winds are important considerations. These aspects are not accounted for in our study and therefore constitute limitations. Precipitation effects could be mitigated by deploying weather-resistant drones. In contrast, strong winds can be handled deterministically by modifying $t^D_{i,k}$ in Constraints (15) (17) and Constraints (24), applying a wind adjustment factor dependent on the flight direction. In our heuristic, wind can be taken into account by considering two route options obtained by traversing the heuristic path in opposite directions. The route that aligns more favorably with prevailing tailwinds is then selected.

Another relevant aspect not captured in our model is the presence of obstacles. Particularly during or after floods, some truck arcs may become impassable, often without prior notice. This can be addressed through a dynamic or stochastic extension of the model (e.g., Dukkanci et al., 2023).

Appendix: Adjustment for general area coverage

Our MILP introduced is capable of covering areas in general, e.g., lakes, where the nodes are not set in line when the following constraints replace Constraints (11) and (12). All other constraints [(2)–(10), (13)–(21)] remain unchanged. Note that it is no longer necessary to order the node set $\mathcal{I}^D$ along the river line, which is also not even possible for areas in general, e.g., lakes.

$$\upsilon^D_{i,d} + 1 \leq \upsilon^D_{j,d} + |\mathcal{I}_0| \cdot (1 - y_{i,j,d}) \qquad \forall (i,j) \in \mathcal{A}, d \in \mathcal{D} \tag{1}$$

$$\upsilon^D_{j,d} \in \mathbb{R} \qquad \forall j \in \mathcal{I}^D, d \in \mathcal{D} \tag{2}$$

These constraints are subtour elimination constraints by Miller et al. (1960). Note that these constraints are not necessary for covering a river, as the drone can only fly in a line or return to the truck.

Acknowledgements This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors. The authors would like to thank the anonymous reviewers and the editors for their valuable recommendations, which have significantly improved the paper.

Funding Open Access funding enabled and organized by Projekt DEAL.

Declarations

Conflict of interest Alexander Rave declares that he has no conflict of interest. Pirmin Fontaine declares that he has no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

Aljalaud, F., Kurdi, H., & Youcef-Toumi, K. (2023). Autonomous multi-UAV path planning in pipe inspection missions based on booby behavior. *Mathematics*, *11*(9). https://doi.org/10.3390/math11092092

BRK. (2022). Drohnen sammeln Daten für schnelle Katastrophenhilfe: Übung mit Rettungskräften im Ahrtal. https://www.brk.de/aktuell/presse/meldung/drohnen-sammeln-daten-fuer-schnelle-katastrophenhilfe-uebung-mit-rettungskraeften-im-ahrtal.html. Last access: August 30, 2024

Boccia, M., Mancuso, A., Masone, A., et al. (2024). Exact and heuristic approaches for the truck drone team logistics problem. *Transportation Research Part C: Emerging Technologies, 165*, Article 104691. https://doi.org/10.1016/j.trc.2024.104691

Cai, L., Li, J., Wang, K., et al. (2025). Optimal allocation and route design for station-based drone inspection of large-scale facilities. *Omega, 130*, Article 103172. https://doi.org/10.1016/j.omega.2024.103172

Chen, M., Hu, Q., Fisac, J. F., et al. (2017). Reachability-based safety and goal satisfaction of unmanned aerial platoons on air highways. *Journal of Guidance, Control, and Dynamics, 40*(6), 1360–1373. https://doi.org/10.2514/1.G000774

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

Der Spiegel. (2021). Unternehmen im Ahrtal verzeichnen mehr als halbe Milliarde Euro Schäden. https://www.spiegel.de/wirtschaft/unternehmen/flutkatastrophe-im-ahrtal-ueber-halbe-milliarde-euro-schaeden-fuer-unternehmen-a-21a2442d-ebaf-4792-b473-ba9a05795ac7, last access: July 04, 2024.

DRK. (2021). Überschwemmungen und Hochwasser. https://www.drk.de/hilfe-weltweit/wann-wir-helfen/katastrophe/ueberschwemmungen/. Last access: July 04, 2024.

Dukkanci, O., Koberstein, A., & Kara, B. Y. (2023). Drones for relief logistics under uncertainty after an earthquake. *European Journal of Operational Research, 310*(1), 117–132. https://doi.org/10.1016/j.ejor.2023.02.038

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

Gstaiger, V., Merkle, N., Rosenbaum, D., et al. (2022). Aus dem All und aus der Luft frisch auf den (Lage-) Tisch: Der Nutzen von Luft- und Satellitendaten für die Lageerfassung. *Im Einsatz* 56–61. https://elib.dlr.de/186343/

Kippnich, M., Kippnich, U., Erhard, H., et al. (2022). Weiterentwicklung im Katastrophenschutz: Ziel, Strategie und Taktik am Beispiel der Hochwasserkatastrophe 2021 im Ahrtal. *Notfall + Rettungsmedizin*. https://doi.org/10.1007/s10049-022-01089-7

Landesregierung, R. (2021). Aktuelle Situation - Zahlen und Fakten. https://web.archive.org/web/20210731110520/. https://hochwasser-ahr.rlp.de/de/aktuelle-lage/zahlen-und-fakten/, last access: July 04, 2024

Li, H., Wang, H., Chen, J., et al. (2020). Two-echelon vehicle routing problem with time windows and mobile satellites. *Transportation Research Part B: Methodological, 138*, 179–201. https://doi.org/10.1016/j.trb.2020.05.010

Liu, Y., Jianmai, S., Zhong, L., et al. (2019a). Two-layer routing for high-voltage powerline inspection by cooperated ground vehicle and drone. *Energies*, *12*(7). https://doi.org/10.3390/en12071385

Liu, Y., Luo, Z., Liu, Z., et al. (2019b). Cooperative routing problem for ground vehicle and unmanned aerial vehicle: The application on intelligence, surveillance, and reconnaissance missions. *IEEE Access, 7*, 63504–63518. https://doi.org/10.1109/ACCESS.2019.2914352

Luo, H., Zhang, P., Wang, J., et al. (2019). Traffic patrolling routing problem with drones in an urban road system. *Sensors*, *19*(23). https://doi.org/10.3390/s19235164

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

Mirzapour Al-e-Hashem, S. M. J., Hejazi, T. H., Haghverdizadeh, G., et al. (2024). Optimizing last-mile delivery services: A robust truck-drone cooperation model and hybrid metaheuristic algorithm. *Annals of Operations Research*. https://doi.org/10.1007/s10479-024-06164-5

Morandi, N., Leus, R., Matuschke, J., et al. (2023). The traveling salesman problem with drones: The benefits of retraversing the arcs. *Transportation Science, 57*(5), 1340–1358. https://doi.org/10.1287/trsc.2022.0230

Murray, C. C., & Chu, A. G. (2015). The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery. *Transportation Research Part C: Emerging Technologies, 54*, 86–109. https://doi.org/10.1016/j.trc.2015.03.005

Nedjati, A., Izbirak, G., Vizvari, B., et al. (2016). Complete coverage path planning for a multi-UAV response system in post-earthquake assessment. *Robotics*, *5*(4). https://doi.org/10.3390/robotics5040026

Nemer, I. A., Sheltami, T. R., & Mahmoud, A. S. (2020). A game theoretic approach of deployment a multiple UAVs for optimal coverage. *Transportation Research Part A: Policy and Practice, 140*, 215–230. https://doi.org/10.1016/j.tra.2020.08.004

Otto, A., Agatz, N., Campbell, J., et al. (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

Perboli, G., Tadei, R., & Vigo, D. (2011). The two-echelon capacitated vehicle routing problem: Models and math-based heuristics. *Transportation Science, 45*(3), 364–380. https://doi.org/10.1287/trsc.1110.0368

Quantum Systems. (2024). Trinity F90+ Disaster Relief Mission. https://quantum-systems.com/trinity-f90-disaster-relief-mission/. Last access: March 13, 2024

Rave, A. (2025). Two-indexed formulation of the traveling salesman problem with multiple drones performing sidekicks and loops. *OR Spectrum, 47*(67), 104. https://doi.org/10.1007/s00291-024-00785-9

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

Rave, A., Fontaine, P., & Kuhn, H. (2023b). Drone network design for emergency resupply of pharmacies and ambulances. Available at SSRN, 4569199.

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

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

Salama, M. R., & Srinivas, S. (2022). Collaborative truck multi-drone routing and scheduling problem: Package delivery with flexible launch and recovery sites. *Transportation Research Part E: Logistics and Transportation Review, 164*, Article 102788. https://doi.org/10.1016/j.tre.2022.102788

Song, X., Xu, G., Bai, Y., et al. (2016). Experiments on the short-term development of sine-generated meandering rivers. *Journal of Hydro-Environment Research, 11*, 42–58. https://doi.org/10.1016/j.jher.2016.01.004

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

Thomas, T., Srinivas, S., & Rajendran, C. (2024). Collaborative truck multi-drone delivery system considering drone scheduling and en route operations. *Annals of Operations Research* 693–739. https://doi.org/10.1007/s10479-023-05418-y

Windras Mara, S. T., Rifai, A. P., & Sopha, B. M. (2022). An adaptive large neighborhood search heuristic for the flying sidekick traveling salesman problem with multiple drops. *Expert Systems with Applications, 205*, Article 117647. https://doi.org/10.1016/j.eswa.2022.117647

Wu, G., Zhao, K., Cheng, J, et al. (2022). A coordinated vehicle-drone arc routing approach based on improved adaptive large neighborhood search. *Sensors*, *22*(10). https://doi.org/10.3390/s22103702

Xia, Y., Chen, C., Liu, Y., et al. (2023). Two-layer path planning for multi-area coverage by a cooperative ground vehicle and drone system. *Expert Systems with Applications, 217*, Article 119604. https://doi.org/10.1016/j.eswa.2023.119604

Xu, B., Zhao, K., Luo, Q., et al. (2023). A GV-drone arc routing approach for urban traffic patrol by coordinating a ground vehicle and multiple drones. *Swarm and Evolutionary Computation, 77*, Article 101246. https://doi.org/10.1016/j.swevo.2023.101246