Delft University of Technology

Delft Center for Systems and Control

Technical report 22-019

Real-Time UAV Routing Strategy for Monitoring and Inspection for Postdisaster Restoration of Distribution Networks*

J. Fu, A. Núñez, and B. De Schutter

To cite this work, please refer to the published version:

J. Fu, A. Núñez, and B. De Schutter, "Real-time UAV routing strategy for monitoring and inspection for postdisaster restoration of distribution networks," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 4, pp. 2582–2592, Apr. 2022. doi:10.1109/TII.2021.3098506

Delft Center for Systems and Control
Delft University of Technology
Mekelweg 2, 2628 CD Delft
The Netherlands
phone: +31-15-278.24.73 (secretary)
URL: https://www.dcsc.tudelft.nl

* This report can also be downloaded via https://dspub.eu/22-019

Real-time UAV routing strategy for monitoring and inspection for postdisaster restoration of distribution networks

Jianfeng Fu, Alfredo Núñez, *Senior Member, IEEE,* and Bart De Schutter, *Fellow, IEEE*

Abstract

After a natural disaster, a quick inspection of all damaged components is crucial to recover the functionality of distribution networks. Unmanned aerial vehicles (UAVs) can perform inspection tasks, particularly for damages that are difficult to access for human repair crews. Additionally, UAVs can monitor the transmission lines to find potential dangers and early-stage damages, and to monitor the road infrastructure to provide real-time information about traffic conditions so that repair crews can select the best ways to reach damages. Besides, due to unpredictable events during restoration, the UAV routing strategy (UAVRS) needs to be updated in real time. Thus, the proposed UAVRS in this paper determines the optimal routes for the UAVs allocated to inspect damages as well as the optimal routes for the UAVs to monitor transmission lines and roads in real-time for distribution networks. To tackle the multi-time-scale characteristic of the proposed UAVRS, a two-layer decision-making architecture is proposed. A bi-level programming problem is solved in the first layer for the large time scale problem and a mixed-integer linear programming (MILP) problem is solved for the small time scale problem in the second layer. A case study based on the distribution network in Zaltbommel and its neighbor areas, in the Netherlands, illustrates the effectiveness of our real-time method compared to the off-line methods. Furthermore, different solvers are studied and compared in view of the real-time requirement.

Index Terms

Distribution network post-disaster restoration, unmanned aerial vehicles routing strategy, monitoring and inspection coordination, real-time routing for unpredictable events

I. Introduction

DISASTERS, e.g., hurricanes, floods, and earthquakes, can damage components of distribution networks. To reduce the impact of the disasters on distribution networks, a variety of post-disaster countermeasures have been proposed in the literature [1]–[3].

Before implementing countermeasures, a clear assessment of types of damages, their locations, statuses and causes can facilitate and improve the performance of the repair crews. Thus, post-disaster inspection should be carried out in a fast manner, so as to significantly improve the reliability and resilience of the power system. UAVs can inspect the status of the components under unsafe and unreachable conditions, e.g., in case of blocked roads due to floods or mudslides. In this way, the human repair crews can work more safely and efficiently. For instance, in [4], a multi-UAV routing strategy is proposed to start post-disaster inspection as quick as possible. A two-stage optimization problem is formulated to firstly determine the starting locations of the UAVs by minimizing setup cost and to secondly determine the inspection routes by minimizing the completion time of inspection. In [5], a fault inspection strategy for UAVs is proposed, where the UAVs establish the information exchange network themselves in areas with telecommunication network coverage. Hoang *et al.* [6] present a strategy for surface inspection using

This work was supported by CSC (China Scholarship Council) with funding number 201806280023.

Jianfeng Fu and Bart De Schutter are with the Delft Center for Systems and Control, Delft University of Technology, Delft, The Netherlands, (e-mail: J.Fu-1@tudelft.nl, B.DeSchutter@tudelft.nl).

Alfredo Núñez is with the Department of Engineering Structures, Delft University of Technology, (e-mail:A.A.NunezVicencio@tudelft.nl).

UAVs. The Internet of Things is utilized as the communication network and a particle swarm optimization algorithm is proposed to route the UAVs. Zheng *et al.* [7] propose a cooperative strategy for transmission line inspections involving UAV schedules and human-team schedules.

Different from the inspection tasks performed by hovering above the damages, monitoring tasks are performed by fast passing along the transmission lines and roads as shown in the blue trajectories (monitoring) and red trajectory (inspection) in Fig. 1. In this paper, UAVs are used for monitoring networks after disasters with two purposes. Firstly, because of some potential dangers and early-stage damages, the components might still be working but with a risk of becoming damaged soon, e.g., leaning but still working towers. By monitoring the transmission lines, potential dangers and early-stage damages can be found. Additionally, fixed non-movable sensors might have become damaged and usually these are not able to cover the whole distribution network. Thus, this paper considers using UAVs to frequently monitor the whole distribution network after the disaster. Secondly, the repair crews require to know the real-time condition of road infrastructure, so as to select the best routes to reach damaged components during restoration. Thus, using UAVs to monitor the road infrastructure after the disaster is also considered in this paper to prevent repair crews from using routes that are dangerous or not accessible.

In literature, monitoring routing strategies have been proposed for other applications [8]–[16]. For instance, in [8]–[10], routing strategies for monitoring of transmission lines are studied. However, these papers focus on how the UAVs can examine the transmission lines from different locations of a particular tower, but not on how to determine the optimal monitoring routes for the whole network. In [11], a UAV monitoring scheme is developed targeting energy efficiency for transmission lines. In [12], [13], UAV monitoring routing strategies for road traffic are proposed. Other applications include wildlife rescue, surveillance, and offshore wind farm monitoring [14]–[16].

In current strategies, inspection and monitoring are not integrated nor coordinated. Therefore, this paper focuses on an integrated and coordinated UAVRS that combines the inspection routing and the monitoring routing. Note that our proposed UAVRS only determines the routes to damages required to be inspected, the detailed routes for how to inspect the damages and manipulate the UAVs, e.g., how to hover, are not considered; the total inspection times for damages are considered instead. Furthermore, the UAV routing strategy should be implemented in real time. That is because during the restoration process, some unpredictable events may happen. New damages may emerge because of, e.g., a subsequent earthquake. Besides, some of the components could already have been inspected by human repair crews, which implies that they do no need to be inspected anymore by UAVs. Furthermore, the position of a UAV might deviate from the originally planned route because of, e.g., a blast of heavy wind, flying speed changes, or communication with the UAVs being temporally disrupted during restoration. Thus, the routes of UAVs should be adapted to these unpredictable events in real time.

To obtain the optimal routes for UAVs, there are a lot of decision-making strategies in literature. For instance, branch-and-bound solvers [17], evolutionary optimization algorithms [18], [19], and meta-heuristic algorithms for multi-objective programming [20] are adopted to solve the UAV routing problems. However, these strategies cannot be applied for multi-time-scale programming as they may easily become intractable for large problem. In order to get a computationally tractable approach, we consider a control architecture with 3 layers: layer 1 gives the results of mode allocation, inspection, and rough monitoring; layer 2 gives the trajectory of the detailed monitoring; and layer 3 takes care of the manipulation of the UAVs, e.g., flight control and trajectory tracking. However, layer 3 has been widely studied in the literature [21], [22], therefore we do not consider layer 3 in detail in this paper.

The contributions of the current paper are as follows:

- To facilitate the restoration process and to make the human crews work safely and efficiently, a UAV routing strategy integrating of monitoring roads and lines, and inspection of damages is proposed for distribution networks.
- To adapt to unpredictable events, the UAV routing strategy is implemented in real time by adopting a receding horizon strategy.

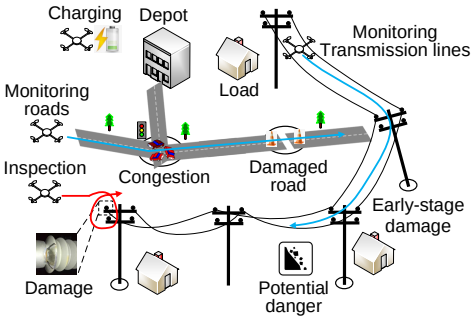


Fig. 1. Illustration of the restoration problem and the modes of the UAVs

- A two-layer decision-making architecture is adopted due to inconsistency between time scales of inspection and monitoring. The architecture proposes to use a bi-level programming problem in the first layer to make the first-layer problem tractable.

The paper is organized as follows. Section II describes the UAVRS problem and explains the proposed two-layer real-time UAVRS architecture. In Section III, the bi-level programming problem of the first layer is described. Then in Section IV, the monitoring problem solved by the second layer is formulated. Section V analyzes a case study based on a real-life distribution network to evaluate the proposed UAVRS. Section VI concludes the paper.

II. Problem Description and Two-Layer UAVRS

A. Problem description

After disasters, aggravating situations such as damages (e.g., insulator flashover), potential dangers (e.g. mudslides, floods near the poles), and early-stage damages (e.g. severely tilted but still working poles) may occur in distribution networks, and aggravating situations such as heavily damaged roads and congestion may occur in road traffic networks as shown in Fig. 1. Thus, the problem of real-time UAVRS for post-disaster restoration is to determine the UAV routes to provide more and real-time information of the distribution network and of the road traffic network, so as to help restore the distribution networks. Note that this problem is dynamic. Thus, a predictive decision-making approach with a moving prediction window is proposed to avoid having a short-sighted real-time strategy. In such an approach, the prediction window contains several decision steps. Optimal decisions are obtained for one prediction horizon window, but only the decisions determined for the first step are implemented. At the next time step, the whole optimization over a shifted prediction window is performed again in the next step with updated information.

According to the categories of the events and the charging requirements of UAVs, in this paper, three UAV operation modes are defined: monitoring mode, inspection mode, and charging mode. Possible trajectories for these three modes are illustrated in Fig. 1. For the monitoring mode (blue trajectories in Fig. 1), the UAVs fly along the transmission lines and the road infrastructure. By monitoring the distribution network, UAVs will detect potential dangers, early-stage damages, and damages that cannot be monitored by fixed sensors. By monitoring the road infrastructure, UAVs will update repair crews about the road traffic conditions in order to guide the repair crews to reach the damages. For the inspection mode (red trajectory in Fig. 1), the UAVs visit located damages to provide more information that facilitates the repair crews in judging the type of damage, and in selecting the required spare components and the required repair materials. In this mode, UAVs fly from their current position to the located damages directly and then stay a while to inspect them carefully by hovering. In the charging mode, UAVs return to the nearest depots before their batteries are depleted.

TABLE I
PROPOSED TWO-LAYER ARCHITECTURE FOR REAL-TIME UAVRS

Layers	Network model	Time scale	Formulated problem	Obtained results
First layer	Graph	Inspection time step (e.g., 5 min)	Top level of bi-level programming	Mode allocation
			Bottom level of bi-level programming	Inspection routes
				Rough monitoring routes
Second layer	Hexagons	Monitoring time step (e.g., 30 s)	Mixed-integer programming	Detailed monitoring routes

B. Two-layer architecture for real-time UAVRS

In the charging mode, the UAVs are recalled back to the nearest depot (considering detours for avoiding obstacles). Then the UAVs in the charging mode do not need to be routed. Next, the models for the monitoring mode and the inspection mode are explained.

For monitoring routing, as shown in Fig. 2.1, the radius R of the monitoring footprints of UAVs is determined by the cameras installed on the UAVs and the flying height of the UAVs. Thus, when the type of UAVs and their equipment are given, R is also known. Since this paper focuses on route planning, the radius of the footprints of the UAVs for monitoring is assumed to be the same for all UAVs. According to R, the distribution network and the road traffic network are divided into hexagons that are inscribed hexagons[1] of the footprints, so that the movements of the UAVs can be modeled. Then the monitoring process can be seen as the movement from the center of one hexagon to the center of one of the six neighbor hexagons (6 options for moving directions) to monitor the lines/roads inside the hexagons as shown in the flight trajectory in Fig. 2.1. For concerns about small obstacles (including no-fly areas and buildings) in monitoring, the UAVs avoid obstacles to monitor by passing through the centers of nearby hexagons that do not contain obstacles as shown in Fig. 2.4. Besides, the time taken for one movement from the center of one hexagon to the center of a neighbor hexagon is defined as the monitoring time step (e.g., for R=173 m and UAV speed 0.6 km/min, a monitoring time step takes 30 s), which is the basic time unit for monitoring.

In the inspection mode, the inspection routes are planned based on the graph without dividing into hexagons as shown in Fig. 2.2. The UAVs travel from their current positions to damages directly, and they do not need to travel along the lines/roads (seen as the straight flight trajectory in Fig. 2.2). Note that our proposed real-time UAVRS only determines the routes to damages required to be inspected; the detailed movements at the manipulation layer for inspecting the damages, e.g., how to hover, are not considered; the total inspection times for damages are considered instead. For concerns about obstacles in inspection, the UAVs make a detour on the travel to avoid the obstacles on the way to the damages as shown in Fig. 2.5. The flights to remote damages may take tens of minutes, and the duration of inspections take several or tens of minutes in practical distribution networks. Thus, the basic time unit for inspection is several minutes, which is defined as the inspection time step.

In general, the length of the monitoring time step (e.g., 30 s) is not comparable to that of the inspection time step (e.g., 5 min in Section V) in practice. Thus, the real-time UAVRS has an inherent multi-time-scale characteristic. In order to coordinate the time scales of monitoring and inspection problems and to reduce the computational complexity, we use a two-layer architecture as described in Table I. In the first layer, the mode allocation, the inspection routes, and the rough monitoring routes are determined with the inspection time step as basic time unit, where a graph is used to model the problem. For tractability, a bi-level linear programming problem [23] is proposed to formulate the first-layer programming problem. More specifically, in the bi-level programming problem, if the results of the top-level problem (mode allocation results) are fixed, the inspection and rough monitoring programming problems at the bottom level can be separated and solved as two individual programming problems. If we would formulate the

[1]The footprint defines a circle and we consider the hexagon inscribed in that circle.

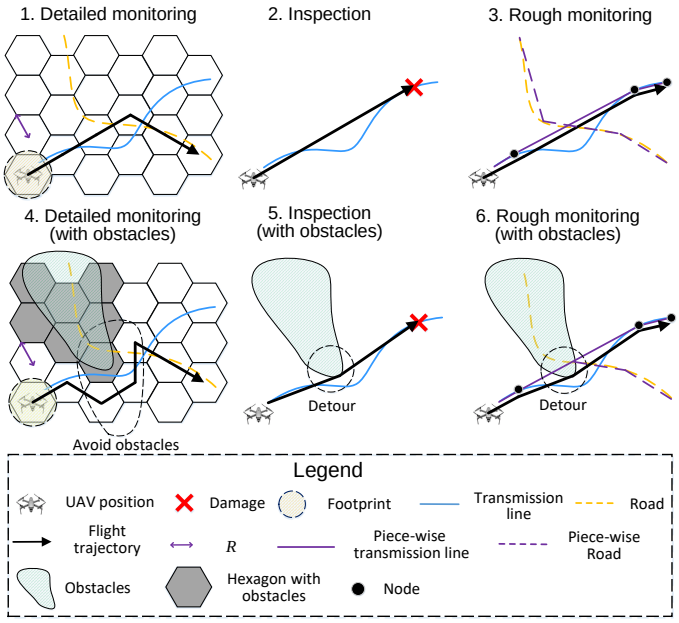


Fig. 2. Difference of monitoring and inspection and obstacle concerns

first-layer problem in a single-level problem, the scale of this problem will be very large. Furthermore, the inspection programming problem and the rough programming problem can then no longer be separated and be solved individually. This is why a bi-level approach is adopted. For the bi-level programming problem, the inspection process has been explained earlier in this subsection, as is also shown in Fig. 2.2 and Fig. 2.5. The rough monitoring process can be described as the UAVs traveling along the transmission lines and roads as shown in Fig. 2.3 and Fig. 2.6 to obtain rewards by finishing monitoring tasks, where a monitoring task corresponds to the monitoring of one transmission line or one road. In the rough monitoring problem, two assumptions are made: firstly, the winding parts and branches of transmission lines and roads are approximated, by considering a sequence of straight lines using some intermediate points as shown in Fig. 2.3 and Fig. 2.6. Secondly, every UAV monitors a transmission line or a road starting from one terminal node to another without turning around until another terminal node is reached. We believe that these two assumptions make sense, because on one hand, the rough monitoring does not need an exact model, as we still have a subsequent detailed monitoring problem (see below). On the other hand, we can let the piece-wise curve approximate the real graph of the lines and nodes arbitrarily well by considering more straight pieces (at the cost of increased computation time). In Section III, we will formulate the first-layer problem and discuss how to solve it.

Then in the second layer, a more detailed monitoring problem with the monitoring time step as basic time unit and a hexagonal grid is proposed to determine the detailed monitoring routes. The detailed monitoring problem takes care of feasibility by tracking the rough monitoring routes as closely as possible given the exact layout of the distribution network, including obstacles. The detailed monitoring process is explained earlier in this subsection as shown in Fig. 2.1 and Fig. 2.4. In Section IV, we will formulate the second-layer problem.

III. Bi-level Programming Problem Formulation

A. *Objective function*

Considering that the inspection processes may take a longer period than one inspection time step, it is assumed that the UAVs in inspecting tasks cannot be interrupted, except when their energy levels are below their thresholds. In addition, in one inspection time step, only one mode can be allocated for each

UAV. The mode allocation problem is formulated as follows:

$$\min_{\delta} \min_{y_1,y_2} (R_{\mathrm{I}}(\delta, y_1) + \gamma R_{\mathrm{M}}(\delta, y_2)) \tag{1}$$

$$\text{s.t.} \quad \sum_{m\in\mathcal{M}} \delta_{n,m} = 1,\ \forall n \in \mathcal{N} \tag{2}$$

$$\delta_{n,3} E_n \le E_{\mathrm{set},n},\ (1-\delta_{n,3}) E_{\mathrm{set},n} \le E_n,\ \forall n \in \mathcal{N} \tag{3}$$

$$y_1 \in \Phi_1(\delta),\ y_2 \in \Phi_2(\delta) \tag{4}$$

where $\delta_{n,m}$ equals 1 if UAV n is allocated to mode m, $\gamma \ge 0$ is a weight coefficient indicating the relative importance of monitoring with respect to inspection, $R_{\mathrm{I}}(\cdot)$ and $R_{\mathrm{M}}(\cdot)$ are the extra cost of inspection and the total monitoring reward respectively (see below for details), $\mathcal{N}$ represents the set of UAVs, $\mathcal{M} = \{1, 2, 3\}$ is the set of modes (the monitoring mode, inspection mode, and charging mode corresponding to mode 1, 2, and 3 respectively), y_1 and y_2 are the vectors of variables of monitoring and inspection respectively that will be defined in Sections 3.2 and 3.3. Constraint (2) indicates that one UAV only can be allocated to one mode. Constraint (3) forces that when E_n, the energy of UAV n, is lower than the threshold value $E_{\mathrm{set},n}$, UAV n should be allocated to the charging mode ($m = 3$). In (4), $\Phi_1(\delta)$ and $\Phi_2(\delta)$ define the feasibility sets of variable vectors y_1 and y_2 in the monitoring mode routing problem and the inspection mode routing problem respectively.

B. Rough monitoring routing problem

In this section, a stochastic rough monitoring routing problem is formulated. The load profiles and the monitoring times of the transmission lines and roads are considered as stochastic parameters with a scenario-based approach. The set of scenarios $\mathcal{S}$ is defined by the stochastic distribution of the loads obtained by the autoregressive moving average approach [24] and the distribution of the monitoring times obtained by discrete Gaussian distributions [25].

Then the rough monitoring routing problem can be formulated by maximizing the rewards obtained in one prediction horizon. The rewards for monitoring a transmission line or a road can be calculated according to their importance. The reward $r_{i,j,s}$ of monitoring from node i to j at the current inspection time step of scenario s is calculated by:

$$r_{i,j,s} = r^{\mathrm{T}}_{i,j,s} + r^{\mathrm{R}}_{i,j},\ \forall i \in \mathcal{I},\ \forall j \in \mathcal{I}^{\mathrm{N}}_i,\ \forall s \in \mathcal{S} \tag{5}$$

where $\mathcal{I}$ is the set of nodes, $\mathcal{I}^{\mathrm{N}}_i$ is the set of neighbor nodes connected to node i via transmission lines or roads, $r^{\mathrm{T}}_{i,j,s}$ and $r^{\mathrm{R}}_{i,j}$ are the rewards of monitoring the transmission line and the road between nodes i and j respectively. If there is no transmission line or no roads between nodes i and j, correspondingly $r^{\mathrm{T}}_{i,j,s} = 0$ or $r^{\mathrm{R}}_{i,j} = 0$. Otherwise, $r^{\mathrm{T}}_{i,j,s}$ and $r^{\mathrm{R}}_{i,j}$ can be calculated by:

$$\begin{aligned} r^{\mathrm{T}}_{i,j,s} &= \zeta^{\mathrm{T}}_{i,j}\ r^{\mathrm{T}}_{i,j,\min} + (1-\zeta^{\mathrm{T}}_{i,j})\ r^{\mathrm{T}-}_{i,j,s}\ e^{P_{i,j,s}}, \\ r^{\mathrm{R}}_{i,j} &= \zeta^{\mathrm{R}}_{i,j}\ r^{\mathrm{R}}_{i,j,\min} + (1-\zeta^{\mathrm{R}}_{i,j})\ r^{\mathrm{R}-}_{i,j}\ e^{Q_{i,j}}, \\ &\forall i \in \mathcal{I},\ \forall j \in \mathcal{I}^{\mathrm{N}}_i\ \forall s \in \mathcal{S} \end{aligned} \tag{6}$$

where $\zeta^{\mathrm{T}}_{i,j}$ or $\zeta^{\mathrm{R}}_{i,j}$ equals 1 if the transmission line or the road from terminal i to j has been monitored at the previous inspection time step, else it equals 0. In (6), $r^{\mathrm{T}}_{i,j,\min}$ and $r^{\mathrm{R}}_{i,j,\min}$ are the minimum rewards for the transmission line and road between nodes i and j, $r^{\mathrm{T}-}_{i,j,s}$ is the reward of monitoring the transmission line between nodes i and j at the previous inspection time step for scenario s, $r^{\mathrm{R}-}_{i,j}$ is the rewards of monitoring the road between nodes i and j at the previous inspection time step, $P_{i,j,s}$ is the downstream power load of the transmission line from node i to j at the current inspection time step of scenario s, $Q_{i,j}$ is the importance of road from node i to j at the current inspection time step, which is determined by whether the repair crews intend to pass by this road to reach the damages or not. The repair crews can propose one or several preferred routes to reach a given damage before they depart. Then, UAVs can

monitor their preferred routes with a priority so that the real-time situation of the roads can be provided to the repair crews. The objective function of the rough monitoring routing problem is defined as:

$$R_{\mathrm{M}}(\cdot) = -\mathbb{E}_{\mathcal{S}}\Big(\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{I}}\delta_{n,1}\ z_{i,j}\ r_{i,j,s}\Big) \tag{7}$$

where $\mathbb{E}_{\mathcal{S}}$ represents the expectation over all the scenarios in set $\mathcal{S}$, and the binary variable $z_{i,j}$ indicates if the set of UAVs have monitored from node i to j at the current time step:

$$z_{i,j} = \begin{cases} 1 & \text{if } \sum_{n\in\mathcal{N}}\delta_{n,1}\ \Delta^{\mathrm{M}}_{n,i,j} \geq 1 \\ 0 & \text{otherwise} \end{cases} \forall i \in \mathcal{I},\ \forall j \in \mathcal{I} \tag{8}$$

where $\Delta^{\mathrm{M}}_{n,i,j}$ is the variable indicating whether UAV n travels from node i to j or not.

When UAV n is allocated to the rough monitoring mode (i.e. $\delta_{n,1} = 1$), it starts from its starting point, while if it is not allocated, there is no constraint for the starting point of UAV n. So:

$$(\delta_{n,1} - 1)\ M_{\mathrm{big}} \leq \sum_{i\in\mathcal{I}}\Delta^{\mathrm{M}}_{n,\varrho_n,i} - 1 \leq (1-\delta_{n,1})\ M_{\mathrm{big}},\ \forall n \in \mathcal{N} \tag{9}$$

where ϱ_n is the starting point of UAV n, M_{big} is an extremely large positive constant. The UAVs allocated to the rough monitoring mode are not supposed to remain stationary at one node i:

$$(\delta_{n,1} - 1)\ M_{\mathrm{big}} \leq \Delta^{\mathrm{M}}_{n,i,i} \leq (1-\delta_{n,1})\ M_{\mathrm{big}},\ \forall n \in \mathcal{N}, \forall i \in \mathcal{I} \tag{10}$$

The time constraints for UAVs are:

$$\begin{aligned}(\delta_{n,1}\Delta^{\mathrm{M}}_{n,i,j} - 1)\ M_{\mathrm{big}} \leq T^{\mathrm{M}}_{n,j,s} - T^{\mathrm{M}}_{n,i,s} - \tau^{\mathrm{M}}_{i,j,s} \leq \\ (1 - \delta_{n,1}\Delta^{\mathrm{M}}_{n,i,j})\ M_{\mathrm{big}},\ \forall n \in \mathcal{N},\ \forall i \in \mathcal{I}',\ \forall j \in \mathcal{I},\ \forall s \in \mathcal{S}\end{aligned} \tag{11}$$

where $\mathcal{I}' = \mathcal{I} \cup \{\varrho_n\}$, $T^{\mathrm{M}}_{n,i,s}$ is the time instant at which UAV n reaches node i in scenario s, $\tau^{\mathrm{M}}_{i,j,s}$ is the travel time from node i to j for scenario s. Note that the travel time should consider the detouring time for obstacles. Furthermore, if node i and j are not connected and i is not the starting point, $\tau^{\mathrm{M}}_{i,j,s}$ is an extremely large positive value for all the scenarios. For $i = \varrho_n$, $\tau^{\mathrm{M}}_{\varrho_n,j,s}$ equals the travel time from the starting point to node j for scenario s. For the starting point ϱ_n of UAV n, $T^{\mathrm{M}}_{n,\varrho_n,s} = 0$. When UAV n departs from node i, it must have arrived at node i:

$$\sum_{j\in\mathcal{I}}\delta_{n,1}\ \Delta^{\mathrm{M}}_{n,i,j} \leq \sum_{j\in\mathcal{I}'}\delta_{n,1}\ \Delta^{\mathrm{M}}_{n,j,i},\ \forall n \in \mathcal{N},\ \forall i \in \mathcal{I} \tag{12}$$

The monitoring process for all UAVs cannot exceed the prediction horizon:

$$0 \leq T^{\mathrm{M}}_{n,i,s} \leq T_{\mathrm{D}},\ \forall n \in \mathcal{N},\ i \in \mathcal{I},\ \forall s \in \mathcal{S} \tag{13}$$

Then the feasibility set can be defined by constraints (5)-(13), and the optimization vector is $y_1 = [\Delta^{M}_{n,i,j}, T^{\mathrm{M}}_{n,j,s}]^{\mathrm{T}}_{n\in\mathcal{N},i\in\mathcal{I}',j\in\mathcal{I},s\in\mathcal{S}}$.

C. Inspection routing problem

In this subsection, a stochastic inspection routing problem is formulated. The inspection time, the traveling time, and the load loss cost are stochastic variables included in the problem via scenarios. When formulating the inspection routing problem, a target time for each inspection is considered. If the earliest UAV reaches one component later than the target time for inspection, an extra cost of inspection will be added to penalize this delay. In this paper, the extra cost of inspection is defined to be equal to the load loss costs from the target time to the earliest arrival time of UAVs. If the UAVs do not reach the component in one prediction horizon, the extra cost of inspection equals the load loss costs from the target time to the end of the prediction horizon. Furthermore, at the next inspection time step, the target time

for components not yet reached in the previous inspection time step will be a negative value to capture that their inspection is delayed from the previous inspection time step(s). In addition, the absolute value of this negative value equals the time between the starting time of the current inspection time step and the target time of this component.

Furthermore, the space-varying characteristics of the load loss costs should be considered, such that damages in different locations will lead to different load loss costs, e.g., some damages can lead to large outages while others only influence small loads. Then, the extra cost of inspection is:

$$\begin{aligned} R_{\mathrm{I}}(\cdot) = \mathbb{E}_{\mathcal{S}'}\Big(\sum_{q\in\mathcal{P}} \sum_{n\in\mathcal{N}} \delta_{n,2}\Big(\sum_{p\in\mathcal{P}'_n} (\Delta^{\mathrm{I}}_{n,p,q} \cdot \max\{(T^{\mathrm{I}}_{n,q,s'} - \\ w_q)\cdot C_{q,s'}, 0\}) + \max\{(1 - \sum_{p\in\mathcal{P}'_n} \Delta^{\mathrm{I}}_{n,p,q}), 0\} \cdot T_{\mathrm{D}} \cdot C_{q,s'}\Big)\Big) \end{aligned} \tag{14}$$

where $\mathcal{P}$ is the set of components that have to be inspected, $\mathcal{P}'_n = \mathcal{P} \cup \{\varrho_n\}$, $\mathcal{S}'$ is the set of scenarios in inspection routing problem. At the beginning of each inspection time step, some UAVs have finished monitoring or inspection, or are flying through the distribution network or road infrastructure, or are in the depots charging their batteries, so the starting points of UAVs will be different. In (14), the extra costs of inspections is separated into two terms. The first term is the total inspection cost. If time instant of scenario s', $T^{\mathrm{I}}_{n,q,s'}$ at which UAV n finishes inspecting component q is later than target time w_q, there will be an additional cost of inspection. If not, the extra cost of inspection will be zero. The second term is the total non-inspection cost when components were not inspected in one prediction horizon. In (14), the load loss cost for damaged component q of scenario s' is $C_{q,s'}$ per unit time, $\Delta^{\mathrm{I}}_{n,p,q}$ is the variable indicating whether UAV n travels from damaged component or starting point p to damaged component q.

When UAV n is allocated to the inspection mode ($\delta_{n,2} = 1$), it starts from its starting point, while if not allocated, there is no constraint for the starting point of UAV n, such that:

$$(\delta_{n,2} - 1)\ M_{\mathrm{big}} \le \sum_{q\in\mathcal{P}} \Delta^{\mathrm{I}}_{n,\varrho_n,q} - 1 \le (1 - \delta_{n,2})\ M_{\mathrm{big}},\ \forall n \in \mathcal{N} \tag{15}$$

The UAVs allocated to the inspection mode are not supposed to remain stationary at one location p:

$$(\delta_{n,2} - 1)\ M_{\mathrm{big}} \le \Delta^{\mathrm{I}}_{n,p,p} \le (1 - \delta_{n,2})\ M_{\mathrm{big}},\ \forall n \in \mathcal{N}, \forall p \in \mathcal{P} \tag{16}$$

To simplify the expression of the following constraints, intermediate variables are introduced:

$$x_{n,p,q} = \delta_{n,2}\ \Delta^{\mathrm{I}}_{n,p,q}\ \forall n \in \mathcal{N},\ \forall p, q \in \mathcal{P}'_n \tag{17}$$

The time constraints for UAVs are:

$$\begin{aligned} (x_{n,p,q} - 1)\ M_{\mathrm{big}} \le T^{\mathrm{I}}_{n,q,s'} - T^{\mathrm{I}}_{n,p,s'} - \tau^{\mathrm{I}}_{p,q,s'} - \sigma_{q,s'} \le \\ (1 - x_{n,p,q})\ M_{\mathrm{big}},\ \forall n \in \mathcal{N},\ \forall p \in \mathcal{P}'_n,\ \forall q \in \mathcal{P},\ \forall s' \in \mathcal{S}', \end{aligned} \tag{18}$$

where $\tau^{\mathrm{I}}_{p,q,s'}$ and $\sigma_{q,s'}$ are the travel time from component p to q and the inspection time for component q of scenario s' respectively. Notice that, the travel time should consider the detouring time for obstacles. Besides, for the starting point ϱ_n of UAV n, $T^{\mathrm{I}}_{n,\varrho_n,s'} = 0$ and $\sigma_{\varrho_n,s'} = 0$ are defined for all $s' \in \mathcal{S}'$. Each component can only be reached once and the UAVs can only depart once from each component:

$$\sum_{n\in\mathcal{N}} \sum_{p\in\mathcal{P}'_n} x_{n,p,q} \le 1,\ \forall q \in \mathcal{P},\ \sum_{n\in\mathcal{N}} \sum_{q\in\mathcal{P}} x_{n,p,q} \le 1,\ \forall p \in \mathcal{P} \tag{19}$$

When UAV n departs from component p, it must have arrived at component p:

$$\sum_{q\in\mathcal{P}} x_{n,p,q} \le \sum_{q\in\mathcal{P}'_n} x_{n,q,p},\ \forall n \in \mathcal{N},\ \forall p \in \mathcal{P} \tag{20}$$

The beginning time of inspections for all the UAVs cannot exceed the prediction horizon:

$$0 \leq T^{\mathrm{I}}_{n,p,s'} \leq T_{\mathrm{D}},\ \forall n \in \mathcal{N},\ p \in \mathcal{P}'_n,\ \forall s' \in \mathcal{S}' \tag{21}$$

Then the feasibility set can be defined by constraints (14)-(21), and the optimization vector is $y_2 = [\Delta^{\mathrm{I}}_{n,p,q}, T^{\mathrm{I}}_{n,p,s'}]^{\mathrm{T}}_{n\in\mathcal{N},p\in\mathcal{P}'_n,q\in\mathcal{P},s'\in\mathcal{S}'}$.

D. Solution approach

To reduce the computational burden and to avoid conservative solutions, a chance-constrained method is used to transform the stochastic constraints into deterministic constraints [26]. Then, the mixed-integer linear bi-level programming problem can be solved by a branch-and-bound computing structure [23]. This branch-and-bound computing structure solves the bi-level programming problem by generating the feasible subsets of the solution set in the top-level problem. Then, it solves the bottom-level problem by substituting these feasible subsets into the bottom-level problem. After that, the solutions of the bottom-level problems associated with the subsets will be obtained and compared to find the optimal solution of the top-level problem via the branch-and-bound algorithm.

Four kinds of bi-level solvers are studied and compared in this paper, including Matlab+CPLEX solver, Matlab+Intlinprog solver[2], Matlab+GA (genetic algorithm) solver, and Matlab+Greedy (greedy algorithm) solver. All these solvers use a branch-and-bound computing structure that can be implemented in Matlab at the top level of the bi-level programming problem, while the difference of these solvers lies in the solvers for the bottom-level mixed-integer linear programming problem. CPLEX and Intlinprog bottom-level solvers can be implemented by the functions "cplexmilp" and "intlinprog" respectively. Here we explain the mechanism of the GA and Greedy algorithms applied in this paper.

The GA algorithm has been implemented in Matlab M files for convenience. Furthermore, as can be seen in Section III.B and Section III.C, when the mode allocation results are fixed by the top-level branch-and-bound solver, the inspection and the rough monitoring problems are two individual problems. Then, the inspection programming problem and the rough monitoring programming problem at the bottom level can be solved separately by using the mechanism for the multiple traveling salesman problem presented in [27]. In detail, the inspection problem is to route UAVs from their starting points to the locations of the damages, and the rough monitoring problem is to route UAVs from nodes to the locations of the damages. Thus, they are both essentially multiple traveling salesman problems.

For the greedy algorithm, the strategy for inspection is as follows: for each inspection time step, first we select the damage that is most urgently required to be inspected, but has not yet been inspected. Then we select the nearest UAV to inspect that damage. Then repeat this process until all UAVs are allocated or all damages are allocated to be inspected. Furthermore, the strategy for rough monitoring is such that: for each inspection time step, we select the line/road with the highest monitoring reward, and select the nearest UAV to monitor it. Then we repeat this process until all UAVs are allocated or all lines/roads are allocated to be monitored.

IV. Detailed monitoring problem

After mode allocation and determining the inspection routes and the rough monitoring routes, the detailed monitoring routing problem is formulated and solved. At each monitoring time step, the positions of the UAVs are updated as:

$$\chi_n(\phi+1) = \chi_n(\phi) + l \cdot \begin{bmatrix} \cos(\theta_n(\phi)) \\ \sin(\theta_n(\phi)) \end{bmatrix},\ \forall n \in \mathcal{N}',\ \forall \phi \in \Phi' \tag{22}$$

where $\chi_n(\phi)$ is the geographical position of UAV n at monitoring time step ϕ, and $\chi_n(0)$ is the initial position of UAV n. The variable $\theta_n(\phi)$ represents the moving direction of UAV n at monitoring time step

[2]Because CPLEX is implemented in object code whereas GA and the greedy algorithm implemented in Matlab, to have a fair comparison with the MILP solvers, we also use the MILP solver implemented in Matlab i.e. Intlinprog.

ϕ where $\theta_n(\phi) \in \Theta = \{ \frac{\pi}{6}, \frac{\pi}{2}, \frac{5\pi}{6}, \frac{7\pi}{6}, \frac{3\pi}{2}, \frac{11\pi}{6} \}$, Φ' is the set of monitoring time steps in the one prediction horizon but excluding the last monitoring time step, and $\mathcal{N}'$ is the set of UAVs that are working in monitoring mode. To use the UAVs more efficiently in one inspection time step, once UAVs finish their charging and inspection tasks, at the remaining monitoring time steps of the current inspection time step, they can be put into the set $\mathcal{N}'$ and used to perform monitoring tasks until the beginning of the next inspection time step. For example, consider that there are 5 monitoring time steps in one inspection time step. When a UAV finishes inspection or charging at the third monitoring time step, then for the remaining two monitoring time steps, this UAV will be allocated to monitoring tasks by solving the detailed monitoring routing problem.

Considering that one inspection time step includes multiple monitoring time steps, solving a programming problem for one inspection time step is time consuming. Thus, a predictive solver using receding horizons is applied to reduce the computation burden. The objective function of the detailed monitoring problem is defined as:

$$\max_{\theta_n(\phi)} \mathbb{E}_{\mathcal{S}} \Big(\sum_{\phi \in \Phi} \sum_{c \in \mathcal{C}} \eta_c(\phi) r^{\mathrm{M}}_{c,s}(\phi) \Big) \tag{23}$$

where $\eta_c(\phi)$ equals 1 if hexagon c is monitored at monitoring time step ϕ, $r^{\mathrm{M}}_{c,s}(\phi)$ is the reward for monitoring hexagon c in monitoring time step ϕ for scenario s, Φ is the set of monitoring time steps in one prediction horizon of the detailed monitoring problem, and $\mathcal{C}$ is the set of hexagons in the whole distribution and traffic network. The objective of the detailed monitoring problem is to monitor the lines and roads to maximize the total reward. The rewards are modeled in the detailed monitoring problem considering two factors. Firstly, when the transmission lines or roads have been monitored recently by UAVs, the condition, e.g., potential failure, traffic jam, of these transmission lines or roads are less uncertain. Secondly, the longer the time since the transmission lines or roads have been monitored, their conditions are more uncertain. Thus, the reward $r^{\mathrm{M}}_{c,s}(\phi)$ for monitoring hexagon c in monitoring time step ϕ can be defined as:

$$\begin{gathered} r^{\mathrm{M}}_{c,s}(\phi) = \sum_{(i,j) \in \mathcal{T}_c} r'_{i,j,s}(\phi) + \sum_{(i,j) \in \mathcal{R}_c} r''_{i,j}(\phi), \\ \forall \phi \in \Phi, \forall c \in \mathcal{C}, \forall s \in \mathcal{S} \end{gathered} \tag{24}$$

where (i,j) represents the transmission line or road between nodes i and j, $\mathcal{T}_c$ and $\mathcal{R}_c$ are the sets of transmission lines and roads included in hexagon c respectively. For $r'_{i,j,s}(\phi)$ and $r''_{i,j}(\phi)$:

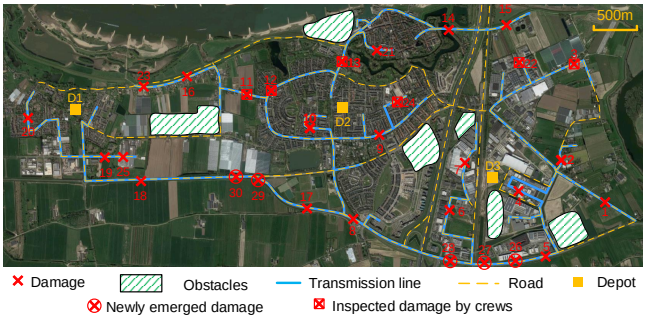
$$\begin{gathered} r'_{i,j,s}(\phi) = \eta_c(\phi) \cdot r^{\mathrm{T}}_{i,j,\min} + (1 - \eta_c(\phi)) \cdot r^{\mathrm{T}}_{i,j,s}, \\ (i,j) \in \mathcal{T}_c, \forall \phi \in \Phi, \forall c \in \mathcal{C}, \forall s \in \mathcal{S} \\ r''_{i,j}(\phi) = \eta_c(\phi) \cdot r^{\mathrm{R}}_{i,j,\min} + (1 - \eta_c(\phi)) \cdot r^{\mathrm{R}}_{i,j}, \\ (i,j) \in \mathcal{R}_c, \forall \phi \in \Phi, \forall c \in \mathcal{C}, \forall s \in \mathcal{S} \end{gathered} \tag{25}$$

Note that $r^{\mathrm{T}}_{i,j,s}$ and $r^{\mathrm{R}}_{i,j}$ vary at each inspection time step but not at each monitoring time step, because the loads and the roads preferred by the repair crews are assumed to not change drastically within tens of seconds (one monitoring time step). Then, $\eta_c(\phi)$ can be obtained by:

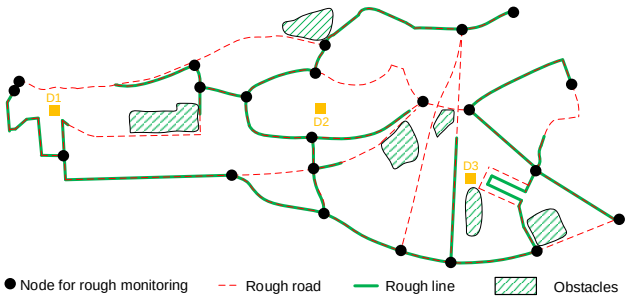
$$\eta_c(\phi) = \begin{cases} 0 & \text{if } \chi_n(\phi) = \chi_c \\ 1 & \text{otherwise} \end{cases} \forall n \in \mathcal{N}', \forall \phi \in \Phi, \forall c \in \mathcal{C} \tag{26}$$

where χ_c represents the geographical position of the center of hexagon c.

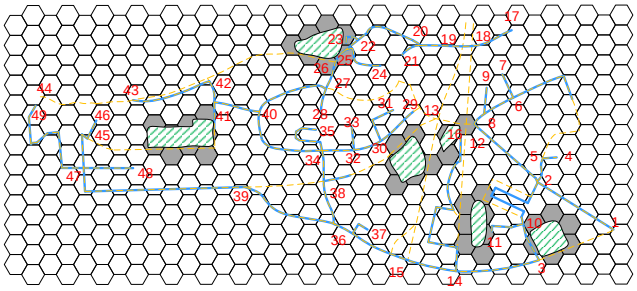
The detailed monitoring problem can be solved by firstly transforming the non-linear constraints into mixed-integer form, and then using branch-and-bound method.



(a) Map for Zaltbommel (*Source: Google map*)



(b) Map for rough monitoring



(c) Map for detailed monitoring

Fig. 3. Maps for the case study

V. Case Study

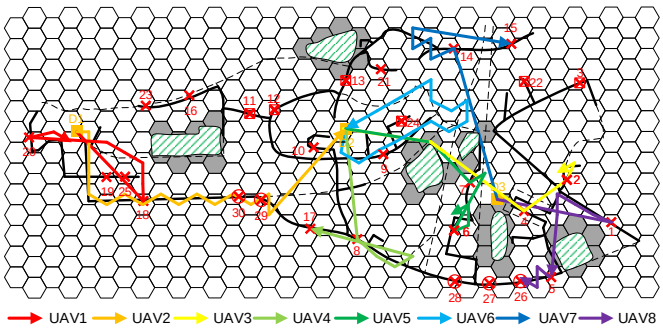
A. General settings of the case study

A real-life distribution network and traffic network in the urban and rural area of Zaltbommel and its neighbors, the Netherlands, is considered as shown in Fig. 3. In this case study, obstacles and a heterogeneous UAV team with multi-rotor UAVs and fixed-wing UAVs are considered. There are 6 multi-rotor UAVs (UAVs 1, 3, 4, 5, 7 and 8) that can be applied for inspection and monitoring and 2 fixed-wing UAVs (UAVs 2 and 6) that can only be used for monitoring.

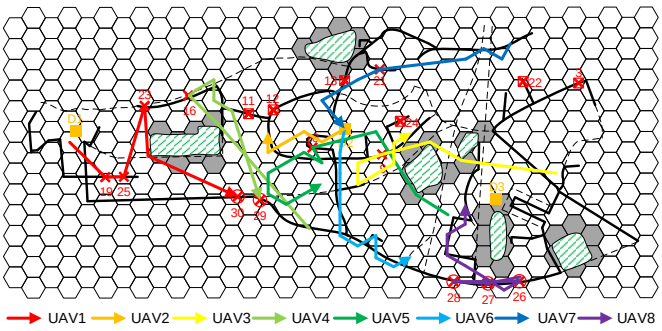
During the restoration process, some unpredictable events will also be considered in this case study, including changes of damages (i.e. newly emerged damages, and damages inspected by repair crews) and the unpredictable position shifting of the UAVs. We will show that the proposed real-time UAVRS can handle these unpredictable events in time by comparing its performance to the off-line methods, which do not handle these unpredictable events in time. The off-line method used for changes of damages determines the UAVRS plans before implementing the inspection and monitoring tasks and does not change them during implementation. This off-line method is characterized by two aspects. Firstly, when a new damage emerges, the off-line method allocates UAVs to inspect the newly emerged damage after the off-line inspection tasks are all accomplished. Secondly, when a damage is inspected by human repair crews, the off-line method still allocates UAVs to inspect that damage but when the UAVs reach that damage (and find that the damage has been inspected by human repair crews), the UAVs move to another inspection task immediately. While, the off-line method used for position shifting considers that when a position shifting occurs, the UAV first moves back to its original route and then follows the original route.

In addition, the algorithms mentioned in Section III.D will be compared w.r.t. the optimality and the solving speed.

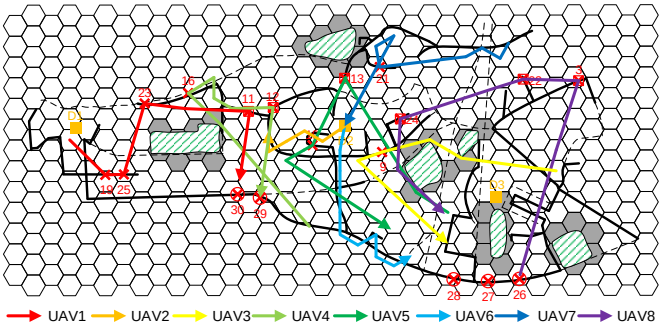
In this case study, the flying speed of all the UAVs is 0.3 km/min, each inspection time step takes 5 min, and the prediction horizon is 15 min ($T_{\mathrm{D}} = 15$). In the detailed monitoring problem, the distance between the center of a hexagon and that of its neighbor hexagon is 300 m. Moreover, there are 5 monitoring time steps in one inspection time step. Furthermore, the prediction horizon of the detailed monitoring is 4 ($|\Phi| = 4$). At the beginning, UAVs 1 and 2 are in Depot 1, UAVs 3 to 6 are in Depot 2, and UAVs 7



(a) Simulation results for steps 1 to 5



(b) Simulation results of the real-time UAVRS for steps 6 to 9



(c) Simulation results of the off-line UAVRS for steps 6 to 9

Fig. 4. Routes for changes of damages

and 8 are in Depot 3. The charging times for UAVs are 15 min. The coefficients are $\gamma = 0.5$, $r^{\mathrm{T}}_{i,j,\min} = 1$, and $r^{\mathrm{R}}_{i,j,\min} = 1$ ($\forall i \in \mathcal{I}$, $\forall j \in \mathcal{I}^{\mathrm{N}}_i$).

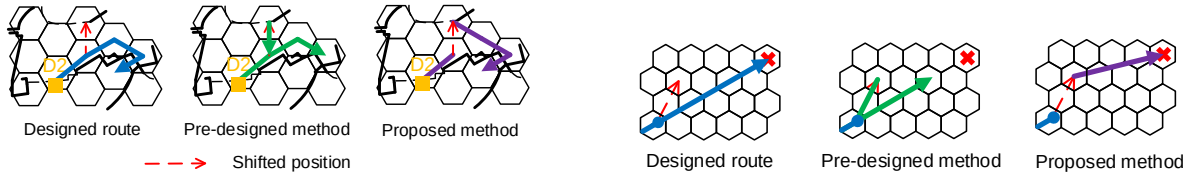
B. Settings and results for changes of damages

The following unpredictable events are considered in the case of changes of damages: At $t = 20$ min, Damages 26 to 30 newly emerge. At $t = 25$ min, $t = 30$ min, and $t = 40$ min, Damages 3 and 22, Damages 11 to 13, and Damage 24 have been inspected by the human repair crews respectively. During inspection time steps 1 to 5, the preferred paths of the repair crew are 3-14, 39-47, 29-34, and 8-29-26, while during inspection time steps 6 to 9, the preferred paths are 3-14, 36-39-38, and 34-40 as shown in Fig. 3c. All these optimal routes in Fig. 4 are obtained by the branch-and-bound bi-level solver (Matlab+CPLEX).

Since there are no changes of damages at the first five inspection time steps, the routes for the proposed and the off-line method are the same at inspection time steps 1 to 5 (Fig. 4a). For inspection time steps 6 to 9, the routes obtained with the proposed method are shown in Fig. 4b, while the routes of the off-line method used for changes of damages are shown in Fig. 4c.

In Fig. 4a, UAV 1 is allocated to inspect Damages 18 and 20 and reaches them at $t = 3.54$ min and $t = 17.85$ min. UAV 2 is allocated to monitor paths 45-47-39-36 and then to the charging mode. UAV 3 is allocated to inspect Damages 4 and 2 and reaches them at 9.72 min and 17.1 min. UAV 4 is allocated to Damages 8 and 17 and reaches them at $t = 4.12$ min and $t = 18.77$ min. UAV 5 is allocated to Damages 7 and 6 and reaches them at $t = 8.72$ min and $t = 17.18$ min. UAV 6 is allocated to monitor paths 34-32-30-29-31-27 and then to the charging mode. UAV 7 is allocated to inspect Damages 14 and 15 and reaches them at $t = 5.88$ min and $t = 19.88$ min. UAV 8 is allocated to inspect Damages 1 and 5 and reaches them at $t = 4.32$ min and $t = 17.18$ min.

Then in Fig. 4b, UAV 1 is allocated to Damages 19, 25, 23, and 30 and reaches them at $t = 25.9$ min, $t = 30.64$ min, $t = 37.68$ min, and $t = 44.6$ min respectively. UAV 2 is fully charged at $t = 31.5$ min and then start monitoring paths 34-36-38-15. UAV 3 is allocated to inspect Damage 9 and reaches it at $t = 31.34$ min and then monitor paths 30-32 and 38-13. UAV 4 is allocated to Damages 16 and 29 and



(a) Position shifting when monitoring

(b) Position shifting on the way to inspect

Fig. 5. Routes for position shifting

reaches them at $t = 31.36$ min and $t = 42.98$ min. UAV 5 is allocated to Damages 10 and reaches it at $t = 32.27$ min and then to monitor paths 34-40 and 39-38. UAV 6 is fully charged at $t = 35.5$ min and start to monitor paths 34-35 and 34-40. UAV 7 is allocated to Damage 21 and reaches it at $t = 33.22$ and then to charging mode. UAV 8 is allocated to Damages 27, 28, and 26 and reaches them at $t = 26.16$ min, $t = 31.28$ min, and $t = 37.44$ min respectively. The whole restoration process is accomplished in 9 inspection time steps for the proposed real-time method.

For the off-line method (see Fig. 4c), UAV 1 reaches Damages 19, 25, 23, and 30 at $t = 25.9$ min, $t = 30.64$ min, $t = 37.68$ min, and $t = 45.89$ min respectively. UAV 2 travels the same routes with those of Fig. 4b. UAV 3 reaches Damages 9 and 27 at $t = 31.44$ min and $t = 47.1$ min. UAV 4 reaches Damages 16 and 29 at $t = 31.36$ min and $t = 44.92$ min. UAV 5 reaches Damages 10 and 28 at $t = 34.55$ min and $t = 49.5$ min. UAV 6 travels the same routes with those of Fig. 4b. UAV 7 reaches Damage 21 at $t = 33.22$ min. UAV 8 reaches Damage 26 at $t = 51.7$ min. The whole restoration process is not accomplished in 9 inspection time steps (actually 11 inspection time steps) for the off-line method.

The total inspection cost for the real-time method is 3.88 for the whole process, while the inspection cost for the off-line method is 146.07. Thus, to handle the newly emerged damages and adapt to the inspected damages by the human repair crews in time are very important in UAVRS for restoration. The monitoring rewards for the whole process are 378.41 and 313.31 for the real-time method and the off-line method respectively. The objective function value of the whole UAVs routing process for the real-time UAVRS is smaller than that of the off-line UAVRS[3].

C. Settings and results of position shifting

In this subsection, unpredictable position shifting on the way to inspect, and when monitoring is considered. The proposed real-time UAVRS is compared with the off-line method used for position shifting as described in Section V.A. The simulation results are shown in Fig. 5.

In Fig. 5a, the designed route for a period of 4 min is shown. However, at t=2 min, the UAV is shifted to a position out of the designed route. The off-line method searches the way back to the trace, while the detailed monitoring routing strategy in the proposed real-time UAVRS architecture determines a better route at t=2 min. By implementing the proposed method, the UAV monitors one more hexagon than the off-line method.

In Fig. 5b, when the UAV reaches the blue node, the designed route is determined for the next inspection time step. However, because of position shifting, the UAV reaches the position indicated by red dashed arrow, not the blue node. The off-line method searches the way back to the off-line routes. On the contrary, the proposed real-time UAVRS architecture adjusts to the new position and determines a new direct route. In this case, the damage can be inspected earlier than with the off-line method.

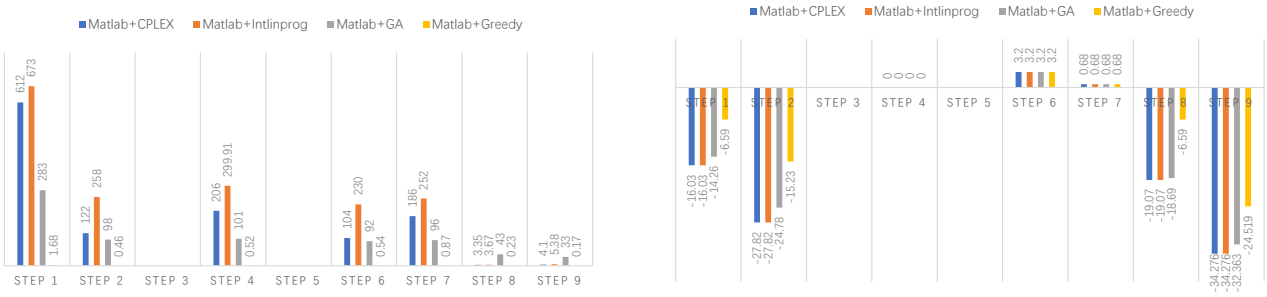
D. Comparison of algorithms and satisfaction of real-time requirements

Since for the detailed monitoring problem at the second layer, the solutions can be obtained in several seconds, which is much shorter than one monitoring time step, the real-time requirement is satisfied easily

[3]To assess whether the proposed approach also perform well for other set-ups, 10 additional cases have been studied in the supplement [28], where a similar conclusion is obtained.

TABLE II
SCALES OF THE PROGRAMMING PROBLEM FOR EACH INSPECTION TIME STEP

	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6	Step 7	Step 8	Step 9
Number of damages	25	22	-	19	-	16	11	5	2
Number of available UAVs	8	5	-	6	-	5	6	3	7



(a) CPU times for the solvers

(b) Objective values of the solvers

Fig. 6. Performance comparison of the solvers

for each time step. This subsection mainly discusses and compares the algorithms mentioned in Section III.D for the first-layer bi-level programming problem.

The scales of the bi-level programming problem for each inspection time step are listed in Table II. The available UAVs represent the UAVs that are not inspecting or charging. The CPU times and objective function values are shown in Fig. 6. In Fig. 6, for inspection time step 3 and 5, there is no UAV available and consequently there is no bi-level programming problem to be solved at these two inspection time steps. Note that the CPU times and objective function values of the GA are obtained by the average value over ten runs. Furthermore, the maximum number of generations to obtain results in one inspection time step is 800, and the number of chromosomes is 601.

From the simulation results, it can be seen that, although the Matlab+CPLEX and the Matlab+Intlinprog solvers can obtain the optimal solution, the computation time may be larger than one inspection time step if the programming problem involves more than 25 damages and 8 available UAVs, while the Matlab+CPLEX and the Matlab+Intlinprog solvers can only handle cases with up to 19 damages and 6 available UAVs. For the Matlab+GA solver, the computing times can be kept smaller than one inspection time step, and the gaps between the Matlab+GA solutions and the optimal solution are less than 11% for large-scale problems (Step 1, 2, 8, 9), with no gaps at all for small scale problems (Step 6, 7). For the Matlab+Greedy solver, the computation time is quite low, but the optimality fluctuates heavily for different inspection time steps. Therefore, the Matlab+GA solver is more suitable to solve the bi-level programming problem in the first layer of our proposed real-time UAVRS.

VI. CONCLUSIONS AND FUTURE WORK

This paper has proposed a real-time UAV routing strategy that can facilitate the fast restoration of distribution networks after disasters. By using UAVs, the road traffic conditions will not influence the inspection and monitoring operations, and the level of safety and efficiency of the human repair crews after the disaster will increase. Then the distribution network operators will obtain updated statuses of the components of the distribution network and of the road infrastructure condition, which benefits the operators and repair crews to take resilience enhancement measures in time and also guides the way for repair crews to reach the damages considering that some roads might be blocked due to the disasters. In addition, the proposed real-time UAV routing strategy can also adapt to unpredictable events, e.g., newly emerged damages, damages already inspected by human repair crews, and position shift of UAVs.

Further research will consider a really larger network, and the multi-level methods will be developed for large-scale instances of the UAVRS problem.

REFERENCES

[1] Y. Wang, C. Chen, J. Wang, and R. Baldick, "Research on resilience of power systems under natural disasters—A review," *IEEE Trans. Power Syst.*, vol. 31, pp. 1604–1613, 2015.

[2] S. Yu, L. Zhang, H. H. Lu, T. Fernando, and K. P. Wong, "A DSE-based power system frequency restoration strategy for PV-integrated power systems considering solar irradiance variations," *IEEE Trans. Ind. Inform.*, vol. 13, no. 5, pp. 2511–2518, 2017.

[3] N. Ganganath, J. V. Wang, X. Xu, C. Cheng, and C. K. Tse, "Agglomerative clustering-based network partitioning for parallel power system restoration," *IEEE Trans. Ind. Inform.*, vol. 14, no. 8, pp. 325–333, 2018.

[4] G. J. Lim, S. Kim, J. Cho, Y. Gong, and A. Khodaei, "Multi-UAV pre-positioning and routing for power network damage assessment," *IEEE Trans. Smart Grid*, vol. 9, pp. 3643–3651, 2016.

[5] S. Y. Derakhshandeh, Z. Mobini, M. Mohammadi, and M. Nikbakht, "UAV-assisted fault location in power distribution systems:An optimization approach," *IEEE Trans. Smart Grid*, vol. 10, pp. 628–636, 2018.

[6] V. T. Hoang, M. D. Phung, T. H. Dinh, and Q. P. Ha, "System architecture for real-time surface inspection using multiple UAVs," *IEEE Syst. J.*, vol. PP, no. 99, 2019.

[7] Y. J. Zheng, Y. C. Du, Z. L. Su, H. F. Ling, M. X. Zhang, and S. Y. Chen, "Evolutionary human-UAV cooperation for transmission network restoration," *IEEE Trans. Ind. Inform.*, vol. 17, no. 3, pp. 648–657, 2021.

[8] S. Kim, D. Kim, S. Jeong, J. W. Ham, J. K. Lee, and K. Y. Oh, "Fault diagnosis of power transmission lines using a UAV-mounted smart inspection system," *IEEE Access*, vol. 8, pp. 99–109, 2020.

[9] Y. Wu, G. Zhao, J. Hu, Y. Ouyang, S. X. Wang, J. He, F. Gao, and S. Wang, "Overhead transmission line parameter reconstruction for UAV inspection based on tunneling magnetoresistive sensors and inverse models," *IEEE Trans. Power Deliv.*, vol. 34, pp. 819–827, 2019.

[10] T. He, Y. Zeng, and Z. Hu, "Research of multi-rotor UAVs detailed autonomous inspection technology of transmission lines based on route planning," *IEEE Access*, vol. 7, pp. 955–965, 2019.

[11] Z. Zhou, C. Zhang, C. Xu, F. Xiong, Y. Zhang, and T. Umer, "Energy-efficient industrial internet of UAVs for power line inspection in smart grid," *IEEE Trans. Ind. Inform.*, vol. 14, no. 6, pp. 2705–2714, 2018.

[12] M. M. Hosseini, A. Umunnakwe, M. Parvania, and T. Tasdizen, "Intelligent damage classification and estimation in power distribution poles using unmanned aerial vehicles and convolutional neural networks," *IEEE Trans. Smart Grid*, vol. 11, pp. 3325–3333, 2020.

[13] Z. Cong, B. De Schutter, M. Burger, and R. Babuška, "Monitoring of traffic networks using mobile sensors," in *17th International IEEE Conference on Intelligent Transportation Systems*, 2014, pp. 92–97.

[14] J. C. Hodgson, S. M. Baylis, R. Mott, A. Herrod, and R. H. Clarke, "Precision wildlife monitoring using unmanned aerial vehicles," *Sci Rep*, vol. 6, p. 22574, 2016.

[15] H. Huang and A. V. Savkin, "An algorithm of reactive collision free 3-D deployment of networked unmanned aerial vehicles for surveillance and monitoring," *IEEE Trans. Ind. Inform.*, vol. 16, no. 1, pp. 32–40, 2020.

[16] H. Chung, S. Maharjan, Y. Zhang, F. Eliassen, and K. Strunz, "Placement and routing optimization for automated inspection with UAVs: A study in offshore wind farm," *IEEE Trans. Ind. Inform.*, pp. 1–1, 2020.

[17] R. G. Ribeiro, J. R. C. Júnior, L. P. Cota, T. A. M. Euzébio, and F. G. Guimarães, "Unmanned aerial vehicle location routing problem with charging stations for belt conveyor inspection system in the mining industry," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 10, pp. 4186–4195, 2020.

[18] Y. Fu, M. Ding, C. Zhou, and H. Hu, "Route planning for unmanned aerial vehicle (UAV) on the sea using hybrid differential evolution and quantum-behaved particle swarm optimization," *IEEE Trans. Syst. Man Cybern. Syst.*, vol. 43, no. 6, pp. 1451–1465, 2013.

[19] W.-C. Chiang, Y. Li, J. Shang, and T. L. Urban, "Impact of drone delivery on sustainability and cost: Realizing the UAV potential through vehicle routing optimization," *Appl. energy*, vol. 242, pp. 64–75, 2019.

[20] B. N. Coelho, V. N. Coelho, I. M. Coelho, L. S. Ochi, R. Haghnazar, D. Zuidema, M. S. Lima, and A. R. da Costa, "A multi-objective green UAV routing problem," *Comput. Oper. Res.*, vol. 88, pp. 306–315, 2017.

[21] S. Farí, X. Wang, S. Roy, and S. Baldi, "Addressing unmodeled path-following dynamics via adaptive vector field: A UAV test case," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 56, no. 2, pp. 1613–1622, 2020.

[22] J. Zhang, J. Yan, M. Lv, X. Kong, and P. Zhang, "UAV formation flight cooperative tracking controller design," in *2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV)*. IEEE, 2018, pp. 856–861.

[23] J. T. Moore and J. F. Bard, "The mixed integer linear bilevel programming problem," *Oper. Res.*, vol. 38, pp. 911–921, 1990.

[24] J. Fu, A. Nunez, and B. De Schutter, "A short-term preventive maintenance scheduling method for distribution networks with distributed generators and batteries," *IEEE Trans. Power Syst.*, pp. 1–1, 2020.

[25] Z. Ye, C. Chen, B. Chen, and K. Wu, "Resilient service restoration for unbalanced distribution systems with distributed energy resources by leveraging mobile generators," *IEEE Trans. Ind. Inform.*, vol. 17, no. 2, pp. 1386–1396, 2021.

[26] Z. Su, A. Jamshidi, A. Núnez, S. Baldi, and B. De Schutter, "Multi-level condition-based maintenance planning for railway infrastructures – a scenario-based chance-constrained approach," *Transp. Res. Pt. C-Emerg. Technol.*, vol. 84, pp. 92–123, 2017.

[27] A. E. Carter and C. T. Ragsdale, "A new approach to solving the multiple traveling salesperson problem using genetic algorithms," *Eur. J. Oper. Res.*, vol. 175, no. 1, pp. 246–257, 2006.

[28] https://www.researchgate.net/publication/353274308_supplement Accessed July 16, 2021.