



# Exploration of a disrupted road network after a disaster with an online routing algorithm

Lorena Reyes-Rubiano<sup>1</sup> · Jana Voegl<sup>1</sup> · Klaus-Dieter Rest<sup>1</sup> · Javier Faulin<sup>2</sup> · Patrick Hirsch<sup>1</sup>

Received: 3 May 2019 / Accepted: 6 November 2020 / Published online: 30 November 2020  
© The Author(s) 2020

## Abstract

This paper considers the problem of supporting immediate response operations after a disaster with information about the available road network to reach certain locations. We propose an online algorithm that aims to minimize the route length required by an unmanned aerial vehicle (UAV) to explore the road accessibility of potential victim locations. It is assumed that no information about disruptions in the road network is available at the start of the exploration. The online algorithm applies two movement and three orientation strategies. Additionally, a cutting strategy is used to restrict the search space after new information about the state of single roads is obtained. We consider a road and an aerial network for the movements of the UAV, since it is not necessary to follow the route of a road any longer, if it can be marked as disrupted. In extensive numerical studies with artificial and real-world test instances, it is evaluated for different disruption levels, which combinations of movement and orientation strategies perform best. Additionally, we propose different refuelling strategies for the UAV and present how they differ in the number of refuelling operations and the required additional route length. The results show that an efficient online algorithm can save valuable exploration time.

**Keywords** Disrupted road network · Immediate response operations · Online algorithm · Aerial vehicles for humanitarian operations · Online exploration strategies · Disaster management

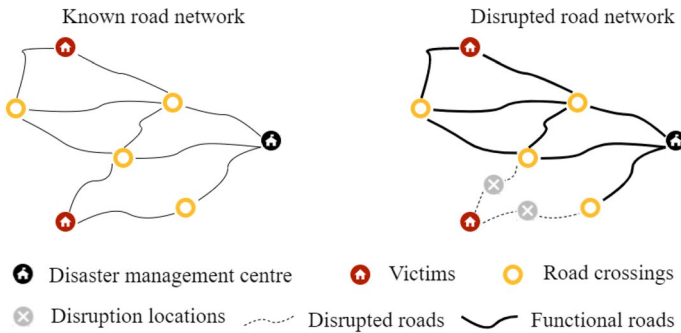
## 1 Introduction

Disasters are unexpected events which are characterized by uncertainty in their impact (Kovács and Spens 2007). These events may generate disruptions on the road network, such as collapsed bridges, or roads covered by water or debris. The affected

---

✉ Patrick Hirsch  
patrick.hirsch@boku.ac.at

Extended author information available on the last page of the article



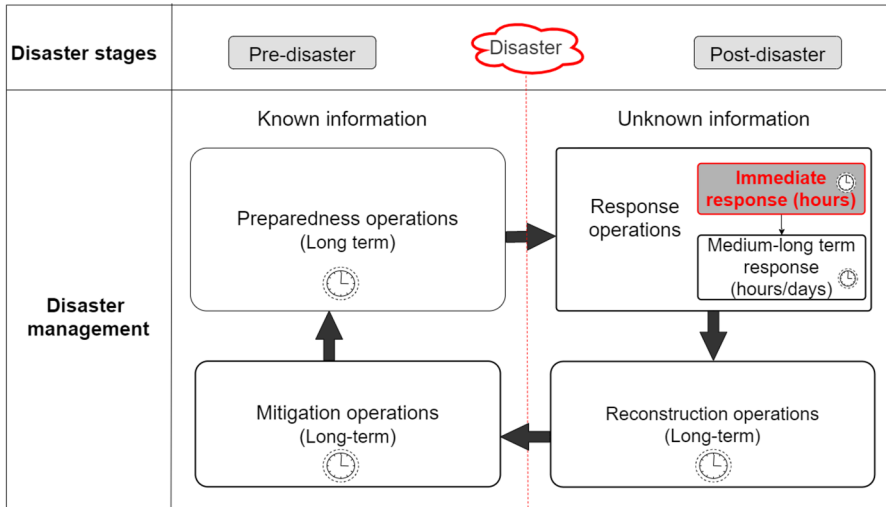
**Fig. 1** Road network states: known road state and disrupted road state

roads cannot be used to travel on. Disruptions reduce the number of roads to reach victims, i.e. to people affected by the disaster (Noyan et al. 2015). Their impacts are particularly strong in networks with a limited number of roads, e.g. in rural zones.

In this paper, a disrupted road network is defined as a subnetwork of the known road network that contains the roads that still work. Figure 1 shows a simplified version of a known and a disrupted road network. These networks each include a disaster management centre (black), where all humanitarian actions and resources are centralized and coordinated, victims (red), i.e. locations of people where humanitarian aid is required, road crossings (yellow) and functional roads (solid bold line). Figure 1 (right) shows the disruption locations (gray) and disrupted roads (dotted line).

A disaster has two stages which involve several logistic operations (Daud et al. 2016). Figure 2 describes the operation cycle of disaster management. The pre-disaster stage involves preparedness operations in a known environment. In contrast, the response and reconstruction operations are performed in an unknown environment in the post-disaster stage (Kovács and Spens 2007; Daud et al. 2016). The immediate response refers to an initial assessment that determines the information for planning of the medium- to long-term response operations (Oruc and Kara 2018). In disaster management, limited information generates a lack of coordination among humanitarian aid organizations. This can cause some victims not to be assisted, while others receive relief more than once (Clarke and Campbell 2015; Torabi et al. 2016; Shao et al. 2020).

Despite the growing number of studies in humanitarian logistics, there is a limited number of research addressing response operations after a disaster and the use of unmanned aerial vehicles (UAVs) (Coutinho et al. 2018; Oruc and Kara 2018; Farahani et al. 2020; Macrina et al. 2020; Pamukcu and Balcik 2020; Rojas Vilorria et al. 2020). Organizations in disaster management, such as the Humanitarian OpenStreetMap Team (HOT 2019) and WeRobotics (WeRobotics 2019), have been starting to include new technologies (FSD 2019). The use of UAVs in humanitarian logistics led to a decrease in the operational times needed for immediate explorations; a detailed overview of UAV use is presented in FSD (2019). In current practice, a UAV tries to cover the whole affected zone using a zigzag trajectory (Tanzi



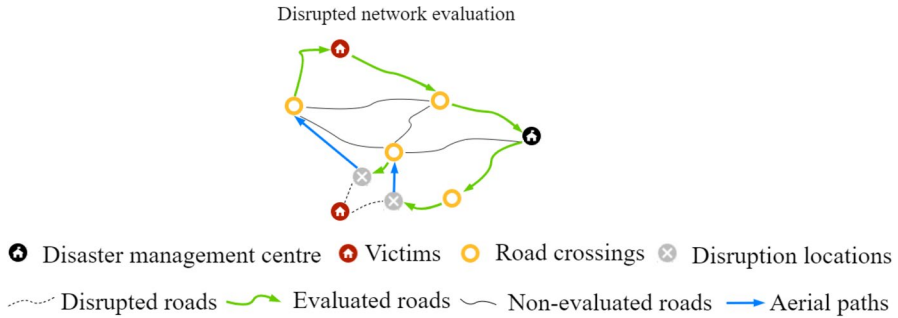
**Fig. 2** Logistic operations over the life cycle of a disaster with focus on immediate response. Own illustration based on Kovács and Spens (2007), Daud et al. (2016) and Shao et al. (2020)

et al. 2016; FSD 2019). However, a UAV usually can only cover a limited number of geographical points of the affected zone within a reasonable time (FSD 2019). Routing operations to plan the flight of a UAV are inefficient (Murphy et al. 2016), and they do not meet the needs of humanitarian logistics (Özdamar and Ertem 2015). As a result, several commercial and industrial projects are looking to develop tools to improve the reliability of routing decisions, to reduce expenses and flight times of UAVs. In particular, in cases in which flight routes have to be defined on the fly (PwC and Agoria 2018). This leads us to the following research question:

How can the road access from the disaster management centre (DMC) to the victim locations after a disaster be evaluated with a UAV within minimum time?

The objective of this paper is to develop an online algorithm with different exploration strategies to guide the UAV routing in such a way that the route length required to evaluate the accessibility of victims by road is minimized. The UAV flies over the known road network to capture and transfer road information revealing the disrupted road network. If a disruption is detected, the UAV has to be redirected according to the exploration strategy using the aerial network. Figure 3 shows evaluated roads (green line), non-evaluated roads (solid line) and an exemplary route of a UAV. Starting from the DMC, the UAV follows the disrupted road network until a disruption is detected, to then use an aerial path (blue line) to travel to the next node, defined by the exploration strategy.

In this paper, we propose the use of a UAV to capture and transfer road information by real-time video. From the DMC, a trained pilot evaluates the state of the road remotely. For this, we develop and implement an online algorithm to provide routing decisions for the UAV based on the current road information. Intuitively, the route could be guided by travel distance or time, but these factors could lead the UAV to blind-spots, i.e. to road crossings which do not connect



**Fig. 3** Example of the disrupted road network evaluation using a UAV

the DMC to victims. We assign priority values for each road according to predefined rules, e.g. redundancy of a road. Based on the gathered information, these priority values are updated. We develop different exploration strategies. These are compared in extensive numerical studies on a realistic instance and on various instances from the literature with different sizes as well as for several disruption percentages in the road network. Additionally, we introduce an extension of the exploration strategies to consider the limited flying range of UAVs due to restricted fuel or battery capacity. It is assumed that the UAV can be refuelled at the DMC.

In summary, the contribution of this article is to present an online algorithm with different exploration strategies for making routing decisions for a UAV in real time based on continuous updates of road network information.

The remainder of the paper is structured as follows: We review previous work related to disaster management and the use of UAVs in Sect. 2. The problem definition in Sect. 3 is followed by the description of the proposed solution approach in Sect. 4. Computational results of a realistic instance and instances from the literature are reported and discussed together with a comparison of exploration strategies in Sect. 5. Finally, concluding remarks are given in Sect. 6.

## 2 Literature review

The number of studies focusing on optimization problems for humanitarian logistics has significantly increased in the last years (OCHA 2019). Existing work covers a broad set of optimization problems for pre-disaster and post-disaster stages. Anaya-Arenas et al. (2014), Leiras et al. (2014), Dascioglu et al. (2019) and Macrina et al. (2020) present comprehensive surveys about humanitarian logistics management. These works provide an overview of qualitative and quantitative models for disaster management related to accessibility, distribution and facility location problems. This section discusses the application of operations research methods and UAVs in the post-disaster phase.

## 2.1 Operations research methods applied to disrupted road networks

This subsection focuses on literature related to disrupted road networks. First, we present studies related to road network evaluation. Second, we consider strategies to deal with disruptions.

Currently, there is a limited number of studies that focus on road network evaluation in the post-disaster situation. Huang et al. (2013) focus on routing operations to evaluate the state of the disaster area. The authors consider a fixed number of vehicles without driving range limitations and assume that all nodes can be visited. The objective is to minimize the sum of arrival times at victim locations. Similarly, Oruc and Kara (2018) assume a motorcycle and a UAV are used to evaluate the road network. The objective is to determine the state of the road network in a given time. The authors affirm that this strategy could lead the vehicle to multiple evaluations of parts of the network with few victims. Coutinho et al. (2019) propose a mixed-integer linear programming model to determine the route of a fleet of UAVs with minimum travel time. The authors evaluate the state of the infrastructure of hospitals and schools. Likewise, Pamukcu and Balcik (2020) solve the routing problem considering two modes of information sharing. In the first one, the UAVs have to return to a DMC to report the state of the affected zone and, in the second one, the UAVs remotely transfer real-time information to the DMC.

To deal with a disrupted road network in the response operations, Eftekhari et al. (2014) design and develop policies for managing disruptions associated with the demand of victims and the availability of resources. These policies are aimed at mitigating the effect of uncertainty in the relief supply distributions. Akkermans and Van Wassenhove (2018) propose operational strategies for managing the availability of relief supplies at the DMC as well as for the estimation of the demand of victims. The authors affirm that operational strategies depend on disaster magnitude and distribution network characteristics.

Response operations are connected to accessibility problems (Noyan et al. 2015). Current solution approaches focus on minimizing operational costs and maximizing the coverage of humanitarian assistance. To deal with limited accessibility after a disaster, Kunz and Van Wassenhove (2019) propose a model to determine the vehicle fleet required to reach all locations of victims in a disrupted road network. The authors minimize the cost while estimating the size of the vehicle fleet to determine the right trade-off between these contradicting objectives. Maya Duque and Sørensen (2011) develop a model to determine the roads that need to be repaired to enhance the accessibility of victims. In this study, each road in the disrupted network is associated with an importance value which reflects the types of connected nodes, e.g. victim locations, DMC or road crossings. Nolz et al. (2010) focus on delivering drinking water in a region partly destroyed by an earthquake, flood or tsunami. The authors propose a model to locate additional water tanks near victims to enhance their accessibility to this resource. Similarly, Rath and Gutjahr (2014) aim to locate additional warehouses in the affected zone to deal with disruptions associated with shortages of relief supplies.

Most works addressing the problem of unknown information in a disrupted road network assume that disruptions and uncertainty can be modelled by a set of

probable scenarios. In this research line, Günneç and Salman (2011) simulate the disaster magnitude and location to determine the disrupted road network. Hatefi et al. (2015) and Elçi and Noyan (2018) develop a stochastic programming model with several probable scenarios for determining the expected disrupted road network after an earthquake. They focus on determining the location and capacity of DMCs to cope with shortages of relief supplies. The authors affirm that disrupted road networks induce inequality problems due to the limited accessibility of victims. Noyan et al. (2017) develop a two-stage model considering different types of decision-makers. Hatefi and Jolai (2014) propose a robust optimization model to minimize the operational cost and to define the disrupted road network probable to work well after a disaster. The authors also consider disruptions related to the shortage of relief supplies at the DMC.

Table 1 summarizes the studies focusing on managing a disrupted road network after a disaster. Besides presenting the disruption location (i.e. at a node or on a road), it also shows the implemented solution approaches for decision making. Most of the disruptions at a node refer to the limited resource stock at DMCs, while disruptions on edges are associated with missing connections to access and supply humanitarian assistance. In the context of disrupted road networks, most studies consider the following assumptions: (i) the disruptions in the road network follow a probability distribution, (ii) knowledge of the road network state is static for response operations and (iii) all victim locations have to be visited. However, in real-world situations, the disruptions in the road network and therefore, the accessibility of victims are unknown information. Disaster managers first need to evaluate the disrupted road network to know if the use of road-based vehicles in response operations is possible.

## 2.2 UAVs in humanitarian operations

Organizations that support disaster response are encouraging the use of UAVs (FSD 2019). The main applications of this technology are to search for victims, fast delivery of medical supplies and to extract ground information by continuously transferring a real-time video of the affected zone (Bravo et al. 2019; Rojas Vilorio et al. 2020). Bravo et al. (2019) focus on planning the UAV route to find victims from UAVs' images. They deal with the uncertainty of the state of the affected area through a stochastic model based on a partially observable Markov decision process. The authors solve the UAV routing problem by giving priority to areas that are more likely to have victims. They highlight the relevance of this technology for finding victims and the lack of method evaluation for route planning in realistic scenarios. Estrada and Ndoma (2019) highlight the benefits of using UAVs to deal with damage to infrastructure such as transportation systems or information and communication networks in the context of humanitarian assistance. Yu et al. (2020) focus on determining the UAV route plan by considering external factors such as mountains and weather. The main objective of their study is to find a route that can meet all the technical restrictions of the UAV. do C Martins et al. (2021) solve a real-time UAV routing problem by considering a two-echelon network. The objective is to satisfy

**Table 1** Summary of literature related to decision making in a disrupted road network

Study	Disruption location	Solution approach	Objective	Problem nature
Nolz et al. (2010)	Node	HM	Multi objective (i) Minmax. tour length from depot to victims (ii) Max. coverage of victims	<i>D</i>
Günneç and Salman (2011)	Edge	SP	Measure reliability of disrupted network	<i>S</i>
Maya Duque and Sörensen (2011)	Edge	M	Min. weighted sum of shortest paths from depot to victims	<i>D</i>
Hatefi and Jolai (2014)	Node	CP	Determine optimal location and number of nodes	<i>S</i>
Rath and Gutjahr (2014)	Node	MT	Multi objective (i) Min. transport costs (ii) Min. fixed costs for depots and vehicles (iii) Max. covered demand	<i>D</i>
Hatefi et al. (2015)	Node	CP	Min. operational costs	<i>S</i>
Noyan et al. (2015)	Node	SP	Multi objective (i) Max. relief supply accessibility (ii) Max. equity of last-mile relief delivery	<i>S</i>
Torabi et al. (2016)	Node	FO	Min. operational costs	<i>S</i>
Noyan et al. (2017)	Node	SP	Min. associated risk of accessibility and equity issues	<i>S</i>
Zhalechian et al. (2018)	Node	SP	Multi objective (i) Min. operational costs (ii) Max. resilience level of hub network	<i>S</i>
Akkermans and Van Wassenhove (2018)	Node Edge	A	Describe disaster impact on supply chain	<i>S</i>
Elçi and Noyan (2018)	Node	SP	Min. total shipping costs of relief supplies	<i>S</i>

*CP* Constrained programming, *SP* stochastic programming, *FO* fuzzy optimization, *M* metaheuristic, *HM* hybrid metaheuristic, *MT* matheuristic, *A* descriptive analysis, *Min.*: minimize, *Max.*: Maximize, *Minmax.*: minimize the maximum, *D* Deterministic problem, *S* stochastic problem

the demand of the victims in the shortest possible time. The authors refer to the use of UAVs for the delivery of medical supplies in disaster situations. They address a deterministic routing problem but highlight the importance of planning UAV routes considering dynamic information.

Sah et al. (2020) analyse the inclusion of UAVs in logistics operations. They conclude that practitioners must deal with government restrictions and limited information on the logistics operations. Farahani et al. (2020) and Rojas Vilorio et al. (2020) present a literature review on vehicle routing problems and the use of UAVs. The authors conclude that in practice, most logistics applications focus on parcel distribution, collecting information or surveillance. Macrina et al. (2020) provide another literature review on operations research contributions of UAV-aided routing problems. The authors affirm that most studies focus on minimizing transportation costs. In addition, Farahani et al. (2020) and Macrina et al. (2020) highlight the lack of contributions regarding uncertain information.

The extraction of road information by aerial images is a relevant research topic in the field of automated systems (Zhang et al. 2019). Coutinho et al. (2018) affirm that most studies determine the flight route neglecting the dynamism of the environment. In addition, the most common UAV applications rely on trained pilots to control and evaluate the information (Coutinho et al. 2018; Estrada and Ndoma 2019). For instance, the pilot determines whether the extracted image refers to a functional or a disrupted road.

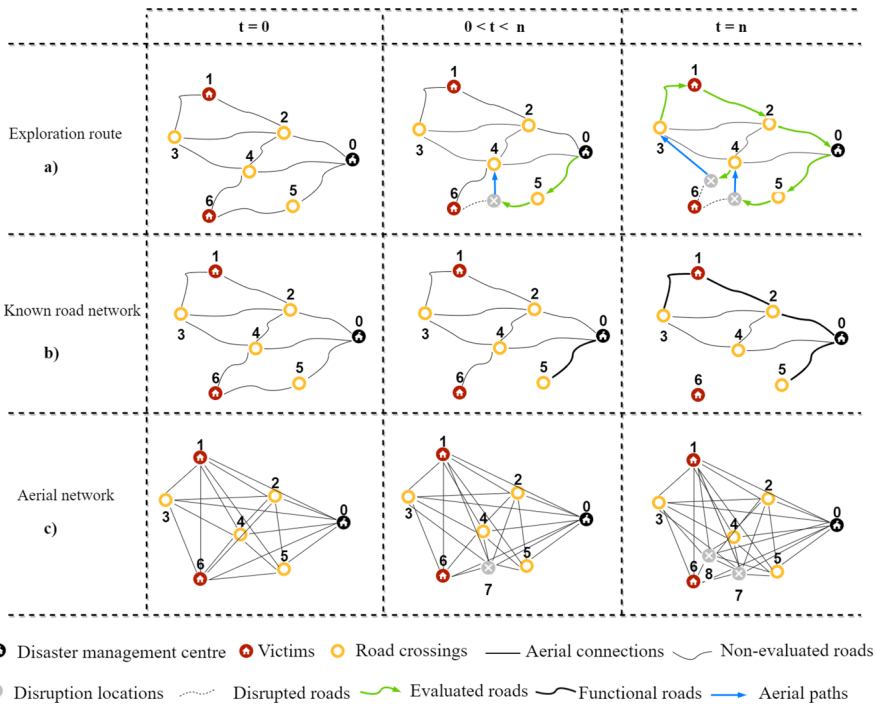
The use of UAVs is subject to technical limitations such as the battery life and therefore, the flying range. To deal with the limited flying range, in some cases the road network is equipped with recharging stations for UAVs, particularly for the exploration of large geographical areas (Yanmaz et al. 2018). In this research line, plenty of studies have been focusing on developing new technologies to solve technical limitations (Shakhatreh et al. 2019).

In summary, most studies in the post-disaster phase focus on maximizing the coverage and minimizing the operational costs of response operations. Scenario-based methodologies are widely applied to address problems related to unknown information and disrupted road networks. Particularly, in UAV routing problems, unknown information is treated by stochastic models where the state of the road network depends on a probability distribution. Our review reveals that there is a lack of immediate response operations strategies for an efficient exploration of a disrupted road network. The online algorithm presented in this paper fills this research gap.

### 3 Problem description

The tackled problem consists of determining exploration strategies to evaluate the accessibility of victims by road using a UAV. The evaluation of the known road network is addressed as a real-time UAV routing problem, aimed at minimizing the required route length to determine the accessibility of all victim locations by road. Figure 4 illustrates the information of the disrupted road network at three different times  $t$  of the exploration route.





**Fig. 4** State of exploration route, known road- and aerial network at different times  $t$  of exploration

Figure 4a shows the exploration route of the UAV. The number of the edges in the known road network decreases as disruptions are detected (see Fig. 4b). Let  $G_t = (N, E_t)$  be the known road network at time  $t$  where the set of nodes  $N$  includes the DMC, victim locations and road crossings, while the set of edges  $E_t$  represents the functional and non-evaluated roads at the time  $t$ . In contrast, the aerial network expands when an interruption is detected. The current position of the UAV at time  $t$  becomes the origin node of a set of aerial edges. Each aerial edge connects the current position of the UAV with each node of the known road network (see Fig. 4c). Let  $\hat{G}_t = (N_t, \hat{E}_t)$  represent the aerial network where  $N_t$  and  $\hat{E}_t$  are the set of nodes and aerial connections at time  $t$ .

The UAV flies over the known road network until a disruption is detected. The disruption may take place at any location on the road under exploration. When a disruption is detected, the affected road is no longer considered in the known network. In Fig. 4, victim 1 can be accessed by road, while victim 6 needs to be accessed by other means of transport (e.g. by air).

Some examples of disruption locations are given in Fig. 5. Figure 5a shows a disruption location on edge (1, 2), thus, the associated road is unavailable (dashed line). The disrupted network is still connected to the DMC, but the accessibility of victim node 1 is reduced to one connection. A similar case is given in Fig. 5b, where accessibility to road crossing node 3 is limited to one connection. The missing edges are not necessary for reaching the victim nodes. Nevertheless,

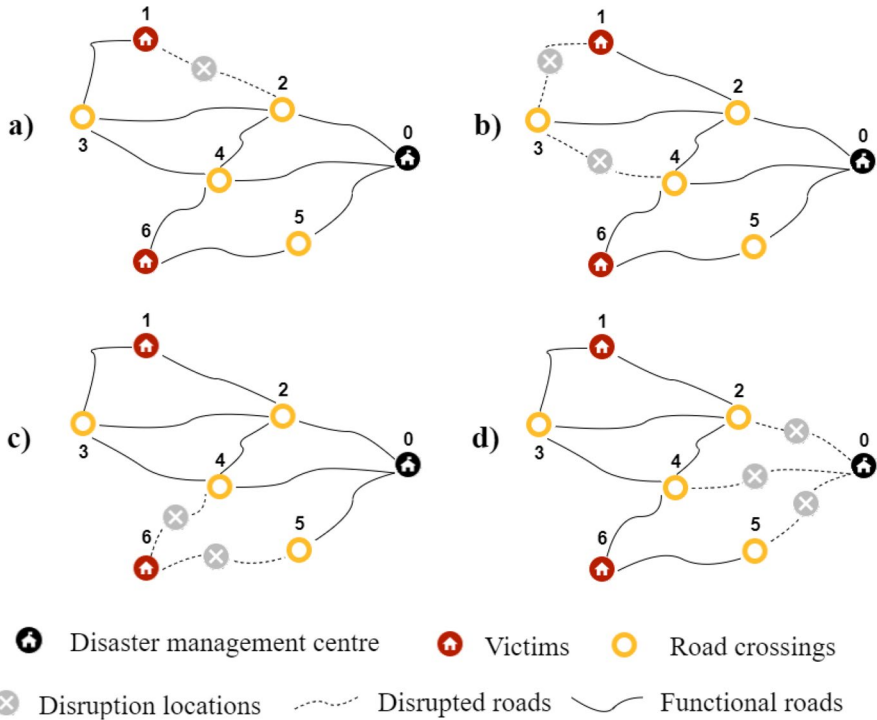


Fig. 5 Type of disruptions in the road network

these disruptions might change the direction of the route for reaching the victims. Figure 5c presents a case where victim node 6 is not connected to the road network anymore. It becomes an unreachable node because there is no possible path to the DMC. As shown in Fig. 5d, some disruptions can impact the accessibility of the DMC. In that situation, no victim location can be reached from the DMC. Consequently, there is no road network to deploy humanitarian aids.

For the tackled problem, we made the following assumptions:

1. The road and the aerial network are undirected networks.
2. Both the road and the aerial network are used: If an interruption is detected on a road, the route of the UAV is redirected using the aerial network.
3. The UAV route starts and ends at the DMC.
4. No information is known about the state of the road network after the disaster. (Note: Nevertheless, if the state of some roads is known in advance, the disrupted roads can be removed from the known road network and the functional roads can be marked as evaluated. This additional information is then used as input for the algorithm.)
5. The length of each edge on the road and the aerial network is known.

6. The disruptions are not known in advance and might occur on any edge of the road network after the disaster.
7. Based on the practical application of UAVs, we assume the use of a fixed-wing UAV which is commonly used to fly large areas (FSD 2019). We select a *Q01* UAV to perform the exploration route. This UAV type is designed for road-based operations and has a large range with up to 8334 km. Its endurance and maximum speed are about 50 h and 333 km/h, respectively (Centre for a New American Security 2019).
8. The large range and endurance of *Q01* allow us to assume an unlimited flying range for our application area. We do not consider the influence of the UAV load, the UAV speed and rates of energy consumption.
9. We assume a constant speed of the UAV.
10. A pilot on the ground monitors the flight progress at the DMC.
11. The pilot determines if a road is disrupted or not.
12. Based on the extracted real-time information, it is determined if a victim location is reachable by road or not.

In an extension of our problem, we account for UAVs with lower flying ranges than *Q01* and make the following differing assumptions:

1. In case that the remaining flying range of a UAV goes below a certain level during an exploration, it needs to return to the DMC for refuelling before it can continue its route.
2. Refuelling is only possible at the DMC.

#### 4 An online algorithm to explore a road network after a disaster

We develop exploration strategies to guide the UAV routing to minimize the route length required to evaluate the accessibility of victims by road. The presented solution approach is implemented in Java. As Fig. 6 shows, the exploration strategy consists of an orientation and a movement strategy. The orientation strategy determines how the priorities of edges are defined, while the movement strategy changes the exploration route of the UAV when a disruption is detected. The online algorithm considers one of three orientation criteria used to assign a priority to each edge

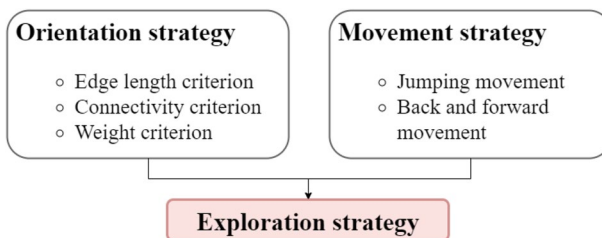


Fig. 6 Overview of components of the exploration strategy

which needs to be explored. Depending on the orientation criterion, the edges with the highest priority are the shortest ones (*Edge length*), those edges that connect the important nodes in the network, i.e. the DMC and the victim nodes (*Connectivity*) or a combination of these criteria (*Weight*). Two movement strategies are applied; these are *jumping movement* and *back and forwards movement*.

The proposed solution approach uses the exploration strategy based on a labelled network and a sort insertion algorithm. The labelled network is a network in which the road edges have an assigned priority. The algorithm only values the road edges that allow reaching the victim nodes by road. As a result, some edges are cut off from the road network to avoid unnecessary explorations. The online algorithm recomputes the priority values when a road edge is explored. It uses the sort insertion procedure to rank the edges according to their priority. This insertion procedure chooses the edge with the highest priority for insertion into the UAV route. Figure 7 shows a general framework of the proposed online algorithm. We describe the details of the cutting procedure of the road network, the labelled network, the movement strategies and the sort insertion procedure in the following subsections.

### 4.1 Cutting procedure

This procedure cuts non-valuable nodes and edges from the known road network. In practical terms, road edges and road crossings that do not belong to a path from the DMC to at least one victim node are not relevant for the exploration. However, computing and storing all paths between the DMC and the victim nodes implies a great computational effort and a long execution time. We propose a heuristic procedure to cut the unnecessary parts of the road network without computing all the possible paths from the DMC to victim nodes. Cutting decisions are based on obtained information from the road network at each time  $t$  of the exploration. As the UAV advances in its exploration, the online algorithm identifies which nodes are reachable from the DMC using a set of functional road edges.

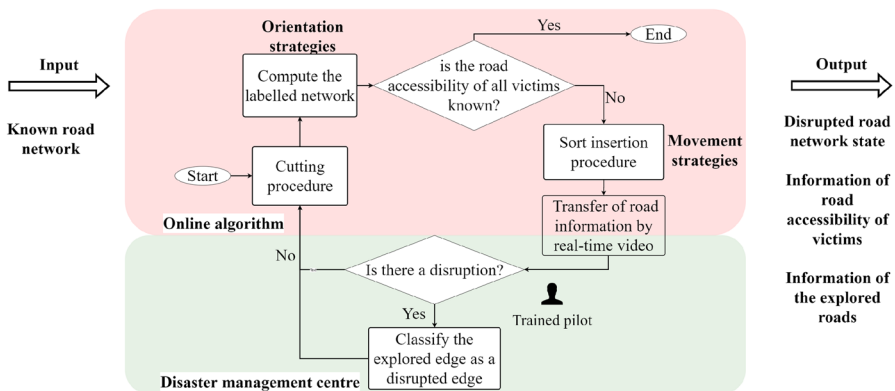


Fig. 7 General framework of the online algorithm

Connected nodes and connected edges are defined as nodes and edges that are reachable from the DMC by a set of functional edges.

The proposed heuristic procedure is described in Algorithm 1. This procedure first identifies the structure of the road network. Algorithm 1 considers the information of the known road network (*knownRoadNetwork*), at time *t* of the exploration, to cut off nodes and road edges not relevant to reach the victim nodes.

---

**Algorithm 1** Procedure to cut the road network.

---

```

1: procedure CUTTINGPROCEDURE(knownRoadNetwork, visitedEdges, connectedElements)
2:   nodeList  $\leftarrow$  excludingRoadCrossings(knownRoadNetwork)
3:   edgeList  $\leftarrow$  redundantEdges(knownRoadNetwork, connectedEdges)
4:   spanningTree  $\leftarrow$  anyTree(nodeList, edgeList)
5:   nodeList  $\leftarrow$  selectingNodes(spanningTree)
6:   edgeList  $\leftarrow$  selectingEdges(spanningTree)
7:   for each (node in nodeList) do
8:     relevantNode  $\leftarrow$  recursiveMethod(node, edgeList) ▷ boolean variable
9:     if relevantNode is false then
10:       exclude(node, nodeList)
11:     end if
12:   end for
13:   knownRoadNetwork  $\leftarrow$  updateInformation(nodeList, edgeList)
14:   return knownRoadNetwork
15: end procedure

```

---

The rules presented in Algorithm 1 are explained below:

1. Select all the nodes of the known road network (*nodeList*), cutting all road crossings which have only one adjacent edge (line 2). For example, in Fig. 8a, first, node 6 is cut and then node 5.
2. Select all the edges of the known road network (*edgeList*) and cut the road edges that have not been visited yet, and link connected nodes to each other (line 3). In Fig. 8b, the edge (4, 6) is cut because it is a disrupted edge that has been explored. The edges (0, 4) and (2, 3) are cut because they are unexplored edges that link connected nodes.
3. Cut nodes and edges that are disconnected from the DMC (lines 5 and 6). The algorithm identifies the disconnected nodes and edges by computing a spanning tree from the DMC (line 5). The cut nodes and edges are the ones which belong to the known road network but are not connected to the spanning tree. Figure 8c illustrates this step, edges (4, 6), (0, 5) and (0, 7) are explored and marked as disrupted. Indeed, the victim nodes 6 and 7 as well as the road crossing 5, with their respective adjacent edges, are unreachable from the DMC by road. As a result, these nodes and edges are cut from the known road network, as shown in Figure 8c.
4. Cut all nodes that do not lead to edges connecting at least one victim node and the DMC (lines 9–11). Using a recursive method the algorithm evaluates the relevance of each node to reach a victim node from the DMC (lines 7–12). For

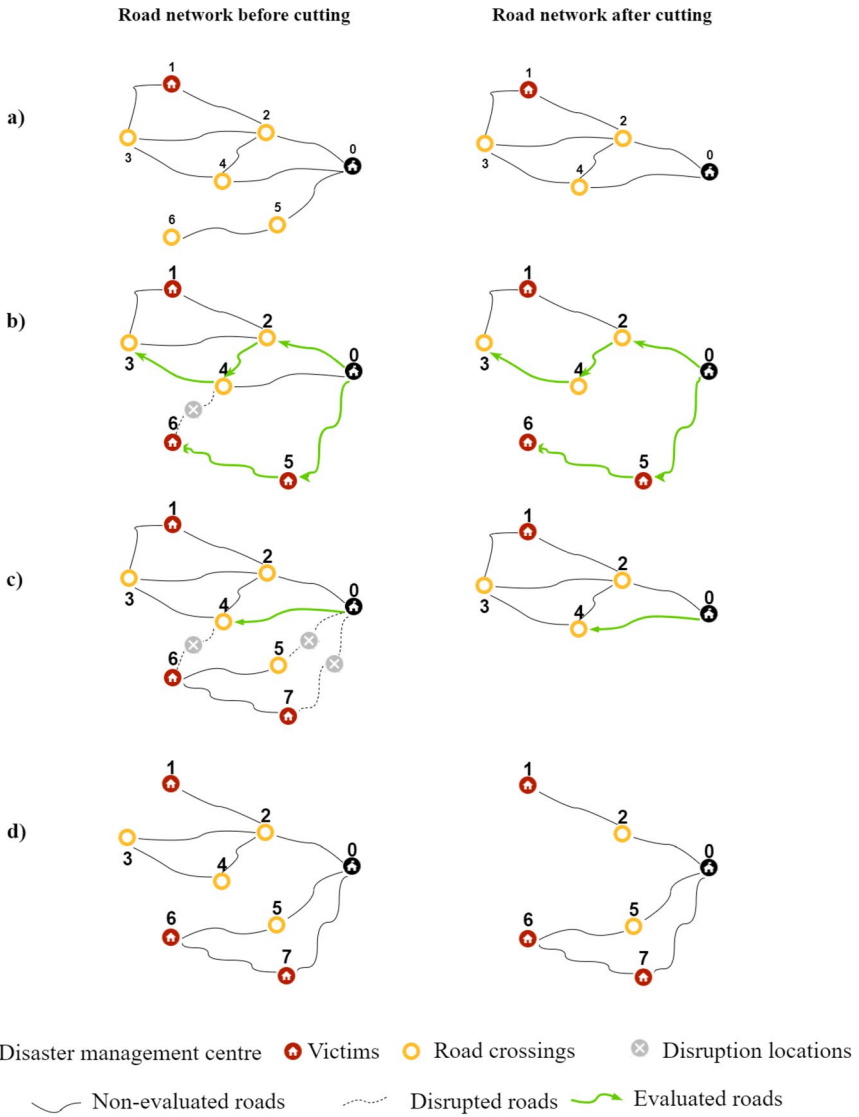


Fig. 8 Example of the cutting procedure

example in Fig. 8d, road crossings 3 and 4 as well as their adjacent edges are cut, because they only lead to a road crossing.

Once the cutting procedure is applied, the known road network is updated (line 13). Applying the cutting procedure after each evaluation of a road edge avoids unnecessary exploration steps of the UAV.

## 4.2 Labelled road network

After the cutting procedure the online algorithm computes a labelled network. In this network, each edge has a label that represents its priority to be explored. The labelled network is calculated only if two conditions are met: (1) there are edges connecting the DMC to the road network; (2) the accessibility of at least one victim node is unknown.

The labelled network is used to orient the UAV to get the accessibility information of the victim nodes within the shortest route length. An overview of calculating the labelled road network is given in Algorithm 2. We define the following three orientation criteria to determine the priority value for each road edge:

1. *Edge length* For this criterion, the priority value refers to the length of the edge.
2. *Edge connectivity* Based on the ideas of Guimera et al. (2005), this criterion represents the relevance of an edge to reach a victim node from the DMC.
3. *Edge weight* This orientation criterion depends on the edge connectivity and the edge length. The priority value of each edge is the weighted sum of both criteria.

---

### Algorithm 2 Procedure to compute the labelled network.

---

```

1: procedure LABELLEDNETWORK(knownRoadNetwork, visitedEdges, orientationP)
    $\triangleright$  orientationParameters: parameters  $\alpha$ , criterion, ss
    $\triangleright$   $\alpha$ : parameter for the weight criterion
    $\triangleright$  criterion: length, connectivity or weight
    $\triangleright$  ss: static score for the DMC, victim and road crossing nodes
2:   edgeList  $\leftarrow$  selectEdges(knownRoadNetwork)
3:   if There are unexplored victim nodes then
    $\triangleright$  Road accessibility
    $\triangleright$  Assignment of a priority value per criterion to each edge
    $\triangleright$  Edge length criterion
4:     labelledNetwork.calculateRoadDistance(knownRoadNetwork, visitedEdges, orientationP)
5:     if criterion = Edge connectivity || criterion = Edge weight then
    $\triangleright$  Edge connectivity. Algorithm 3
6:       labelledNetwork.edgeConnectivity(knownRoadNetwork, visitedEdges, orientationP)
7:     end if
8:     if criterion = Edge weight then
    $\triangleright$  Edge weight. Algorithm 4
9:       labelledNetwork.edgeWeight(knownRoadNetwork, visitedEdges, orientationP)
10:    end if
11:   end if
12:   return labelledNetwork
13: end procedure

```

---

The following sections describe how the priority value of each edge is determined according to the connectivity and weight criterion.

### 4.2.1 Connectivity criterion

Algorithm 3 shows the procedure to calculate the edge connectivity for the labelled network. This procedure starts with the identification of the existing nodes in the road network at time  $t$  of the exploration. Initially, each node has a static score ( $ss$ ) used to distinguish the types of nodes, i.e. the DMC, victim nodes and road crossings (line 5). Each  $ss$  has a positive value. For the DMC and victim nodes, the value of  $ss$  initializes with a higher score than the one assigned to road crossings. Once the exploration has been started and it is proven that there is a functional edge that connects a road crossing  $i$  with a victim node or the DMC, the road crossing  $i$  obtains a new  $ss$  equal to the one assigned to the DMC and the victim nodes. For example, in Fig. 9 the  $ss$  of the road crossing node 5 is equal to the one of the victim node 6. Taking the values of  $ss$  as a reference, the connectivity value ( $c$ ) is calculated for each node. The connectivity of a node is defined as the sum of its  $ss$  value and those of its adjacent nodes (lines 5–8).

Once the connectivity value for each node is computed, it is transferred to the edges directly connected to the given node. The edge connectivity ( $edgeC$ ) is the sum of the connectivity values of its origin and end node (in Algorithm 3, lines 9–12).

---

#### Algorithm 3 Procedure to compute the edge connectivity.

---

```

1: procedure EDGECONNECTIVITY(knownRoadNetwork, visitedEdges, ss)
    ▷  $ss = \{a, b\}$ 
    ▷  $a$ :  $ss$  for the DMC and victim nodes
    ▷  $b$ :  $ss$  for road crossing nodes
2:   nodeList ← selectNodes(knownRoadNetwork)
3:   edgeList ← selectEdges(knownRoadNetwork)
4:   assignStaticScore(nodeList, visitedEdges, ss)
    ▷ Node type identification
5:   for each (node in nodeList) do
6:      $c$  ← calculateNodeConnectivity(node, knownRoadNetwork)
7:     node.assignNodeConnectivity( $c$ )
    ▷ Node connectivity
8:   end for
9:   for each (edge in edgeList) do
10:    edgeC ← sumNodesConnectivity(edge)
    ▷ Edge connectivity
11:    edge.assignEdgeConnectivity(edgeC)
12:   end for
13: end procedure

```

---

### 4.2.2 Weight criterion

This orientation criterion depends on the connectivity and the length of an edge. The priority value of each edge is the weighted sum of the edge connectivity ( $edgeC$ ) and the edge length ( $edgeL$ ). An importance level  $\alpha$  in the co-domain  $[0.1, 0.9]$  is assigned to the edge length criterion. Equation 1 computes the edge weight criterion ( $edgeW$ )



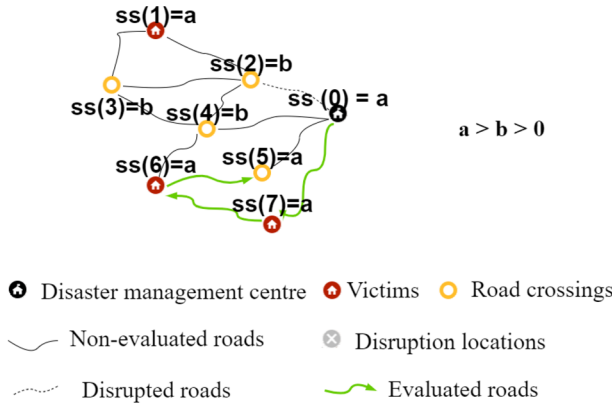


Fig. 9 Example of an *ss* assignment

for each edge (in Algorithm 4, lines 5–8). The parameter *longest* denotes the length of the longest edge and the parameter *highestC* is the highest edge connectivity value; the parameters are determined from the known road network and updated at each time *t* when the UAV obtains new information.

$$edgeW := \alpha \cdot \left( 1 - \frac{edgeL}{longest} \right) + (1 - \alpha) \cdot \frac{edgeC}{highestC} \tag{1}$$

---

**Algorithm 4** Procedure to compute the edge weight.

---

```

1: procedure EDGEWEIGHT(knownRoadNetwork, visitedEdges,  $\alpha$ )
    ▷  $\alpha$ : importance level for the edge length
2:   edgeList ← selectEdges(knownRoadNetwork)
3:   highestC ← maxEdgeConnectivity(edgeList)
4:   longest ← maxEdgeDistance(edge)
5:   for each (edge in edgeList) do
6:     edgeW ← calculateWeight(edgeList, highestC, longest) ▷ Edge weight. Equation 1
7:     edge.assignEdgeWeight(edgeW)
8:   end for
9: end procedure
    
```

---

For the exploration strategy one of the three mentioned criteria is chosen for the orientation of the route. Following the connectivity and weight criterion, road edges are sorted in descending order; thus, the edge with the highest value has the highest priority. In contrast, when the edge length criterion is used, the edge with the shortest length has the highest priority value.

To apply the orientation criteria, we develop a sort insertion procedure to determine an efficient UAV routing. An overview of this procedure is given in the next section.

### 4.3 Routing decisions

The sort insertion procedure selects the best edge to be explored next, i.e. the edge with the highest priority. In each iteration, the online algorithm inserts the best edge into the exploration route of the UAV. Once a road edge is explored, the labelled network has to be updated. The movement strategy is used when there is an interruption in the exploration. An interruption can occur when a disrupted edge is detected or when the node which is the current position of the UAV has no adjacent edges to visit. The online algorithm deals with interruptions by redirecting the UAV, using one of two movement strategies; the *jumping movement* strategy or the *back and forwards movement* strategy.

Similar to the cutting procedure, the routing decisions are made if these two conditions are met: (1) there are edges connecting the DMC to the road network, (2) the accessibility of at least one victim node is unknown. In the following, we describe the procedure of defining the exploration route. An overview of the routing decisions is given in Algorithm 5.

---

#### Algorithm 5 Sort insertion procedure

---

```

1: procedure INSERTIONPROCEDURE(knownRoadNetwork, labelledRoadNetwork)
2:   position  $\leftarrow$  DisasterManagementCentreLocation
3:   victimsInformation  $\leftarrow$  evaluatedRoadAccessibility(knownRoadNetwork)   ▷ List of explored victims
4:   explorationRoute  $\leftarrow$  newRoute   ▷ Exploration route is empty
5:   while unexploredVictims(victimsInformation) do   ▷ Victim nodes to be explored
6:     edgeToInsert  $\leftarrow$  selectBestEdge(position, labelledRoadNetwork)   ▷ Selection of an edge
7:     if edgeToInsert is null then
8:       edgeToInsert  $\leftarrow$  redirectTheUAV(position, explorationRoute)   ▷ Exploration strategies
9:       explorationRoute  $\leftarrow$  setExplorationRoute(edgeToInsert, explorationRoute)
10:    end if
11:    if edgeToInsert is a disrupted edge then   ▷ Evaluation of the selected edge
12:      knownRoadNetwork  $\leftarrow$  cuttingProcedure(knownRoadNetwork)   ▷ Algorithm 1
13:      labelledRoadNetwork  $\leftarrow$  LabelledNetwork(knownRoadNetwork)   ▷ Algorithm 2
14:      aerialEdge  $\leftarrow$  redirectTheUAV(position, explorationRoute)
15:      explorationRoute  $\leftarrow$  setExplorationRoute(aerialEdge, explorationRoute)
16:    end if
17:    knownRoadNetwork  $\leftarrow$  cuttingProcedure(knownRoadNetwork)
18:    labelledRoadNetwork  $\leftarrow$  labelledNetwork(knownRoadNetwork)
19:    knownRoadNetwork  $\leftarrow$  update(explorationRoute, victims)   ▷ Update of road information
20:    position  $\leftarrow$  updateUAVPosition(explorationRoute)
21:  end while
22:  return explorationRoute, knownRoadNetwork
23: end procedure

```

---

Taking the chosen route orientation criterion and the current location of the UAV as references, the best next edge to explore is selected (method *selectBestEdge*, line 6). Initially, the UAV position is the DMC (line 2). If there is no adjacent edge to explore, the movement strategy (method *redirectTheUAV*, line 8) is applied. Otherwise, the best adjacent edge to be explored is selected. The following rules describe the selection of that best next edge to explore:

1. Determine the node in the road network which is the current position of the UAV.
2. Considering the known road network, select the adjacent edges to the node determined in step 1.
3. Take the edges selected in step 2, check if there are still unvisited edges.
  - (a) If there are unvisited edges, one of the following two strategies is used as a parameter for the online algorithm.
    - (i) Choose the unvisited edge with the highest priority value.
    - (ii) Choose an unvisited edge which connects directly to a victim node. In case of a tie, choose the edge with the higher priority value.
  - (b) If there are no unvisited edges, apply the movement strategy (line 8).

Once the best next edge to explore (*edgeToInsert*) is determined, the UAV flies over the selected road edge. Then, the road information of the last exploration is transferred and the known road network is updated (lines 17 and 18). The procedure of updating the information consists of applying the cutting procedure and calculating the labelled road network.

This procedure iterates until the online algorithm has the information of the road accessibility of all victim nodes from the DMC. The result of this procedure is the exploration route and the set of functional edges.

The following subsections describe how each movement strategy deals with interruptions in the exploration route.

### 4.3.1 Back and forwards movement strategy

This strategy redirects the UAV based on the connected edges and nodes that have been identified up to the moment the interruption takes place. An aerial edge or a road edge, or if useful a combination of both, is used by the UAV to return to the closest connected node with unexplored adjacent edges.

Figure 10 illustrates the route of the UAV after an interruption. The UAV returns to the closest connected node with unexplored adjacent edges. For example, in Fig. 10 (left), the current position of the UAV is node 7, with no adjacent edges to visit. In this case, the UAV returns to the DMC because it is the closest connected node with unexplored adjacent edges to its current position. In a first step all unexplored edges from the DMC, namely (0, 5), (0, 4) and (0, 2), are evaluated in order to find the closest node adjacent to these edges to node 7 via an aerial edge. Since node 5 is the closest one to node 7, the UAV uses the aerial edge to node 5 and

explores the road edge (5, 0) on its way back to the DMC. In case of Fig. 10 (right), the interruption is caused by a disruption. There, the UAV needs to return to node 2 as it is the closest connected node with unexplored adjacent edges. Since road edge (3, 2) is still unexplored and node 3 is closer than nodes 2 and 4 to the disruption location, the UAV uses the aerial edge to node 3 and explores the road edge (3, 2).

The following steps are used for the selection of the closest connected node:

1. Select all connected edges (green arrows, in Fig. 10).
2. Select all nodes connected to the edges selected in step 1.
3. From the nodes selected in step 2, select the closest connected node with unexplored adjacent edges.
4. Check if there is a road edge connecting the current position of the UAV with the node selected in step 3 (closest connected node).
  - (a) If there is a road edge to the closest connected node, evaluate this road edge.
  - (b) If there is no road edge to the closest connected node, evaluate all adjacent nodes from the unexplored edges from the closest connected node and select the node which can be reached within the shortest aerial distance from the current position. In case that the closest connected node is selected, create an aerial edge from the current position to it, else use the aerial edge to the selected node and evaluate the road edge to the closest connected node afterwards.
5. Update the current position of the UAV. The current position is the last node visited.
6. Transfer the road information to update the known road network.

### 4.3.2 Jumping movement strategy

This movement strategy deals with interruptions by using an aerial edge to the closest node of an unexplored road edge from the current position. The road edge is

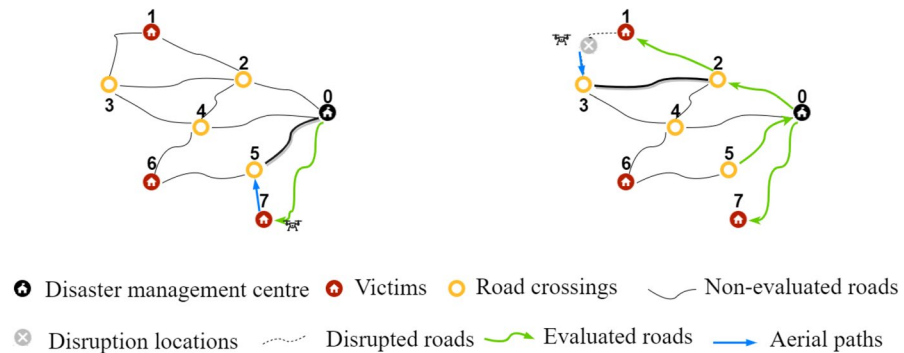


Fig. 10 Examples for the back and forwards strategy

chosen based on a selection criterion that considers the priority value of the unexplored edges as well as the current position of the UAV. For example, in Fig. 11 (left), the strategy selects the unexplored edge (5, 6). Thus, the UAV uses the aerial edge to the closest node from 7, which is 5, to then evaluate the road edge (5, 6). In Fig. 11 (right), the UAV flies to node 3 from the disruption location to explore the selected road edge (3, 4).

The steps for the selection of the closest node are the following:

1. Select the unexplored road edges  $e$  in the known road network.
2. For each road edge  $e$ , compute the aerial distance from the current position of the UAV to its two adjacent nodes. Select the shortest aerial distance to this road edge  $de$ .
3. For each road edge  $e$  compute the selection criterion  $ae = \frac{\text{priority value of } e}{de}$ .
4. Select the road edge with the highest value of  $ae$ . Then, use the aerial edge to get to the closest adjacent node of that edge and evaluate it.
5. Update the current position of the UAV. The current position is the last node visited.
6. Transfer the road information to update the known road network.

Next, we present an extension of the exploration strategies to consider a limited flying range of the UAV.

#### 4.4 Exploration route with a flying range limitation of the UAV

The refuelling strategies are based on a dynamic buffer which ensures that the UAV has enough flying capacity to finish the exploration route at the DMC at any time. We assume that the UAV can only be refuelled at the DMC. Forced refuelling operations are a type of interruption in the online algorithm. We assume that the buffer is defined in distance units. Thus, the dynamic buffer represents the distance that the UAV has to be able to fly to explore the next selected unexplored road edge

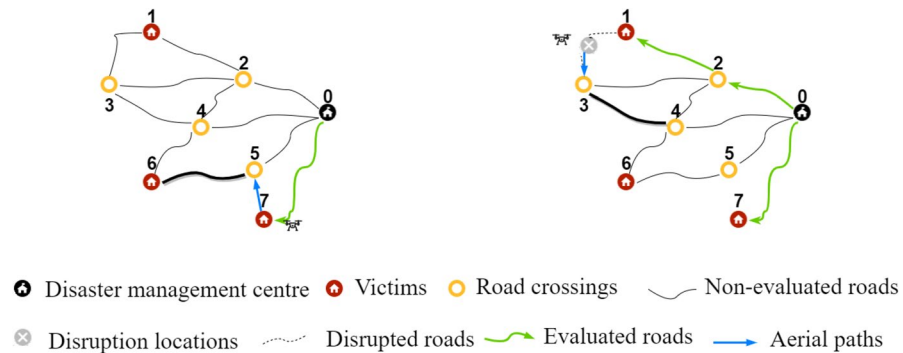


Fig. 11 Examples for the jumping strategy

(*edgeToInsert*) and then return to the DMC. Figure 12 shows the variations of the dynamic buffer. Figure 12a, b describes the buffer when the *edgeToInsert* connects the DMC and the current position of the UAV. Figure 12c depicts the case when the *edgeToInsert* is adjacent to the current position of the UAV but not to the DMC. Finally, Fig. 12d depicts the buffer when the *edgeToInsert* connects nodes other than the current position of the UAV and the DMC. The steps for computing the buffer are described below:

1. Determine the current position of the UAV. The current position can be any node in the known road network or a disruption location.
2. Determine the origin and end node of the *edgeToInsert*.
  - (a) The origin node is the node closer to the current position of the UAV.
  - (b) The end node is the node further away from the current position of the UAV.

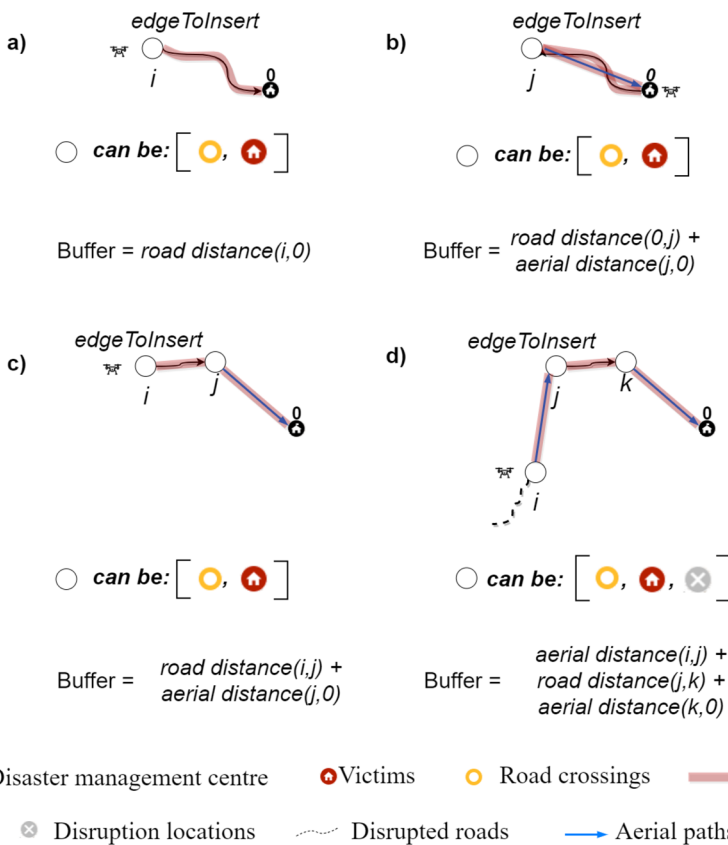


Fig. 12 Examples for the dynamic buffer of a UAV

3. Compute the buffer, which is the sum of the aerial distance between the current position of the UAV and the origin node of *edgeToInsert*, the road distance of *edgeToInsert* and the aerial distance between the end node of *edgeToInsert* and the DMC.

As shown in Fig. 12b, the UAV could also be brought back to the DMC by a routing decision. Given this fact, three strategies are designed to reduce forced returns to the DMC for refuelling operations. Once the UAV is at the DMC, one of the following refuelling strategies can be applied: (1) refuel each time when at the DMC (2) refuel when the flying capacity is lower than 50% of the flying range (3) refuel when the flying capacity is lower than the buffer needed at that time.

The sort insertion procedure applies an additional function to control the dynamic buffer and the flying capacity of the UAV (lines 6, 8 and 14 in Algorithm 5). The steps of this function are described below:

1. Compute the buffer needed at that time of the exploration.
2. Compute the distance travelled and the flying capacity of the UAV at that time of the exploration.
3. Is the UAV located at the DMC?
  - (a) If yes, apply the selected refuelling strategy and explore the *edgeToInsert*.
  - (b) If not, determine if the flying capacity is bigger than or equal the required buffer.
    - (i) If yes, explore the *edgeToInsert*.
    - (ii) If not, fly over an aerial edge from the current position of the UAV to the DMC. Refuel the UAV at the DMC and use an aerial edge to start the exploration of the *edgeToInsert*.
4. Update the current position of the UAV. The current position is the last node visited.

## 5 Computational experiments and results

In this section, we present the design of the computational experiments and their results. We evaluate the performance of the proposed exploration strategies of the online algorithm under different conditions. Sections 5.1 to 5.4 describe the artificial test instances and the real world-based instance as well as their solutions.

The proposed online algorithm does not require any parameters. Only the Edge weight orientation strategy uses the importance level  $\alpha$  for the edge length and  $(1 - \alpha)$  for the connectivity criterion. We gradually increase the value of  $\alpha$  in steps of 0.1 in the interval [0.1, 0.9]. The files with the instances and their detailed parameterization are available at [short.boku.ac.at/instances](http://short.boku.ac.at/instances).

## 5.1 Artificial test instances

We test the performance of the proposed solution method on a set of generic instances, which are obtained from Chao et al. (1996). The number of nodes ranges from 20 to 101. To adapt these instances to our problem, we define a road network and an aerial network. The aerial network is based on Euclidean distances between all nodes of the graph. For the road network, we compute first the minimum spanning tree of the aerial network and add cycles to this tree in a next step to mimic a realistic road network. After that, we define additional nodes, called breakpoints (BP). We consider two BPs per edge on the road network, one BP is set to the right and the other one to the left of each edge. This is done in a way to avoid new crossings with other edges. The length of each edge in the road network is then the sum of the Euclidean distances on the path  $originNode - BP1 - BP2 - endNode$ . Depending on the number of cycles in the road network, the number of nodes in the road network is at least tripled by adding the BPs. The BPs are also used to mimic potential disruption locations on the road network. The instances were first tested using a type *Q01* UAV with a large flying range with up to 8334 km. Then, they were tested using a UAV with a lower flying range. This UAV was selected according to the length of the longest edge in each instance (see Table 2).

Table 2 summarizes the characteristics of the artificial instances used in the experiment.

The following numerical experiments were conducted:

1. Comparison of the orientation strategies: edge length, edge connectivity and edge weight (taking into account 11  $\alpha$  values).
2. Comparison of the movement strategies: back and forwards movement and jumping movement.
3. Refuelling strategies: refuel each time when at the DMC ( $D$ ), refuel when the flying capacity is lower than 50% of the flying range ( $H$ ), refuel when the flying capacity is lower than the buffer needed at that time ( $R$ ).

**Table 2** Characteristics of artificial instances and UAV type

Instance	#Nodes	#Victim nodes	Average number of adjacent edges per node	Average edge length	UAV type*
p1.2.b	32	9	2.65	31.20	Super BAT Flying range: 643 km
p2.2.a	20	9	2.50	24.37	
p3.2.a	32	12	2.50	34.60	
p4.2.a	99	28	2.48	27.68	K-Max Flying range: 1852 km
p5.2.a	65	25	3.47	19.82	Super BAT Flying range: 643 km
p6.2.a	63	22	2.63	14.64	
p7.2.a	101	31	2.45	6.90	

1 Disaster management centre, 1 UAV

\*UAV type with a lower flying range than the *Q01*



4. Evaluation of different disruption levels {30%, 50%, 70%}.
5. For each disruption level, 10 different disrupted road networks are evaluated per instance (10 seeds).

## 5.2 Comparison of the exploration strategies

In the first stage of the experiment, the performance of the exploration strategies was evaluated considering the preference of edges which are directly connecting to victim nodes, as mentioned in Sect. 4.3, step 3(a), *i* and *ii*. We test all combinations of movement (4) and orientation strategies (11) for the seven artificial instances with ten disruption seeds (70) and three disruption levels (3). As a result, 9240 experiments were performed. We found that biasing the route decision as described in Sect. 4.3, step 3(a), *ii* led to much better results for all exploration strategies. Hence, the results presented in the following sections are obtained under this preference.

We computed the exploration route for each instance and each seed with a fixed disruption level for all  $\alpha$  values and movement strategies. Then, we determined the shortest exploration route for each instance and each seed. This route is compared to all other routes of the same instance and seed. We computed the percentage deviation from the length of the shortest route for all other routes from the same instance and seed. In the next step, for each instance and  $\alpha$  value, the average percentage deviation over all tested seeds was calculated for each disruption level. Then, these values were taken to present average values over all instances. Thus, low deviations indicate the best exploration strategy, i.e. the best movement strategy and the best  $\alpha$  setting. Table 3 presents for each movement strategy this average deviation for each disruption level. In "Appendix 1", Table 8 provides more details about each instance.

In road networks with a 30% disruption level, the routes obtained by the back and forwards movement strategy are on average 10% longer than the shortest route while the routes computed by the jumping movement strategy are about 69% longer on average. In road networks with a disruption level of 50% and 70%, the average deviation for the back and forwards movement strategy is 5% and 3%, while for the jumping strategy the average deviation is 109% and 97%, respectively. This indicates that the routes provided by the back and forwards movement strategy are shorter than the routes obtained by the jumping strategy.

Table 4 presents the behaviour of each movement strategy using different  $\alpha$  values. Here, the shortest exploration route is determined for each movement strategy independently. All other steps are the same as for Table 3. Table 4 shows the deviation from the shortest route for each  $\alpha$  value and level of disruption, over all instances and seeds. In "Appendix 1", Tables 9 and 10 show more detailed results for each instance; they also provide the absolute values of the route lengths. It can be observed that the route lengths obtained by the jumping strategy are much more variable compared to those of the back and forwards movement strategy, for all  $\alpha$  values. This can be explained by the broader exploration of the jumping strategy, which may be undertaken in unknown areas after a disruption is detected. In general, the lengths of the exploration routes obtained by the back and forwards movement

**Table 3** Average deviation from the shortest exploration route over all movement strategies and tested instances for each  $\alpha$  value and disruption level

Disruption level	Connectivity (%)	Weight (%)									Length (%)
		0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	
<i>Back and forwards movement strategy</i>											
0.3	12	11	<b>10</b>	12	12	13	12	13	14	13	13
0.5	6	6	6	6	7	6	<b>5</b>	6	6	6	6
0.7	5	4	4	5	4	<b>3</b>	4	4	4	4	4
<i>Jumping movement strategy</i>											
0.3	87	80	72	76	77	77	73	<b>69</b>	86	80	89
0.5	149	156	132	116	128	116	<b>109</b>	126	149	155	142
0.7	123	130	<b>97</b>	107	131	134	135	159	180	195	194

The smallest average deviation is marked in bold

**Table 4** Average deviation from the shortest exploration route over all tested instances for each movement strategy,  $\alpha$  value and disruption level

Disruption level	Connectivity (%)	Weight (%)									Length (%)
		0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	
<i>Back and forwards movement strategy</i>											
0.3	8	7	<b>6</b>	7	8	8	7	8	9	8	8
0.5	4	4	3	4	5	4	<b>3</b>	4	4	4	4
0.7	3	3	3	3	3	<b>1</b>	3	2	2	2	2
<i>Jumping movement strategy</i>											
0.3	42	41	34	38	38	40	33	<b>31</b>	41	41	45
0.5	74	72	46	35	53	44	<b>37</b>	58	71	85	70
0.7	52	70	<b>39</b>	42	59	61	59	80	89	110	102

The smallest average deviation is marked in bold

strategy are shorter than those of the jumping strategy. Nevertheless, there are few exceptions (see Tables 9 and 10 in "Appendix 1"). For example, the average route length obtained by the jumping strategy for test instance p1.2.b with a disruption level of 30% is shorter than the one obtained by the back and forwards movement strategy. Hence, it can be argued that in a few cases, especially when the disruption level is low, it can be advantageous to try out the jumping strategy. But in general, the back and forwards movement strategy seems to be the better strategy, since it delivers much better results on average.

Regarding the setting of the  $\alpha$  values, the average deviations presented in Table 4 allow us to estimate the best  $\alpha$  values for each movement strategy at each disruption level. The back and forwards movement strategy provides on average shorter routes with an  $\alpha$  value equal to 0.2, 0.6 and 0.5 for each disruption level, respectively. In

general, the setting of the  $\alpha$  value has less impact in case of the back and forwards movement strategy; this can be explained by the dominance of the movement strategy over the orientation strategy. On the other hand, the  $\alpha$  value impacts the solution quality much more in the jumping movement strategy; this strategy reaches on average shorter routes when the  $\alpha$  values are set to 0.7, 0.6 and 0.2 for each disruption level, respectively. The edge weight criterion seems to be the best orientation strategy with values in the range of [0.2, 0.6] for the back and forwards movement strategy and values in the range of [0.2, 0.7] for the jumping movement strategy.

### 5.3 Performance of the refuelling strategies

This section presents the results of the three proposed refuelling strategies  $D$ ,  $H$  and  $R$ . Each refuelling strategy was tested with the generic test instances described in Sect. 5.1. The objective of these experiments was to determine the best refuelling strategy to use in each movement strategy. Obviously, having no refuelling operation will lead to the best solution. Hence, the single refuelling strategies are benchmarked against these solution values. From the previous Sect. 5.2, we determine over all  $\alpha$  values the shortest exploration route for each instance and seed, considering no refuelling operations. This route is used as a benchmark for routes obtained using the refuelling strategies with the same  $\alpha$  value setting. In order to make a fair comparison between the refuelling strategies, we select only those instances and seeds where refuelling operations are needed. Tables 5 and 6 present the average number of refuelling operations and the average deviation from the route length of the benchmark for each instance and refuelling strategy over all meaningful seeds for both movement strategies.

Table 5 presents the performance of the single refuelling strategies for the back and forwards movement. Not surprisingly, refuelling each time when the UAV is at the DMC is the best refuelling strategy in terms of route length. However, this will lead to more refuelling operations. The  $H$  and  $R$  strategies perform on average less refuelling operations, but their average deviations from the best route length are up to 3 percentage points higher than the average deviation of  $D$ .

Table 6 presents the performance of the refuelling strategies for the jumping movement strategy. The performances of the refuelling strategies are similar to those reported for the back and forwards movement strategy.

Tables 5 and 6 report deviations equal to zero in cases where only one seed was considered for that particular instance. This is because the exploration route of the refuelling strategy coincides with the benchmark route. This behaviour is given when the UAV is at the DMC because of a regular routing decision and the condition for performing a refuelling operation is met at the same time. Depending on the time required to perform a refuelling operation, decision-makers can choose the best refuelling strategy. If refuelling is cheap with respect to time, strategy  $D$  seems to be the best one.

**Table 5** Back and forwards movement strategy: comparison of refuelling strategies

Instance	Disruption level	<i>D</i>		<i>H</i>		<i>R</i>	
		RO	Dev. (%)	RO	Dev. (%)	RO	Dev. (%)
p1.2.b	0.3	3.20	2	2.30	3	2.22	5
p2.2.a		2.00	0	2.00	10	2.00	10
p3.2.a		3.80	2	2.67	4	2.57	6
p4.2.a		4.00	2	2.60	5	2.20	6
p5.2.a		4.00	8	3.70	7	3.60	8
p6.2.a		2.60	8	2.43	11	2.43	11
p7.2.a		3.00	0	2.00	2	2.00	3
p1.2.b	0.5	3.90	0	2.00	0	2.00	5
p3.2.a		4.00	1	2.17	2	2.00	5
p4.2.a		4.30	0	2.00	0	2.00	3
p5.2.a		3.40	4	3.86	6	3.71	7
p6.2.a		2.13	2	2.00	6	2.00	6
p3.2.a	0.7	4.00	0	2.00	0	2.00	0
p5.2.a		2.20	0	2.00	1	2.00	2
Average		3.32	2.07	2.41	4.07	2.34	5.50

RO, number of refuelling operations. Dev., average deviation from the shortest exploration route obtained with no refuelling operation

**Table 6** Jumping movement strategy: comparison of refuelling strategies

Instance	Disruption level	<i>D</i>		<i>H</i>		<i>R</i>	
		RO	Dev. (%)	RO	Dev. (%)	RO	Dev. (%)
p1.2.b	0.3	2.70	5	2.10	5	2.00	6
p3.2.a		3.50	4	2.50	6	2.25	8
p4.2.a		3.20	1	2.38	2	2.13	4
p5.2.a		4.00	7	3.60	8	3.50	8
p6.2.a		2.50	9	2.38	10	2.38	12
p7.2.a		2.40	0	2.00	1	2.00	4
p1.2.b	0.5	3.80	2	2.00	3	2.00	7
p3.2.a		3.40	1	2.20	7	2.00	7
p4.2.a		4.00	1	2.25	3	2.00	3
p5.2.a		4.00	4	4.57	8	4.57	9
p6.2.a		2.38	5	2.00	7	2.00	9
p1.2.b	0.7	4.10	0	2.00	1	2.00	4
p3.2.a		3.90	0	2.00	6	2.00	13
p5.2.a		2.40	2	2.40	3	2.40	7
p6.2.a		2.00	0	2.00	12	2.00	12
Average		3.22	2.73	2.43	5.47	2.35	7.53

RO, number of refuelling operations; Dev., average deviation from the shortest exploration route obtained with no refuelling operation

## 5.4 Real world-based instance

To test our solution approach under more realistic conditions, we have designed an instance based on a rural area in northern Colombia. In Colombia, the awareness for the need of coordinated humanitarian support has evolved significantly over the last decade. In addition, recent major natural disasters continue to show an increasing necessity for a good disaster preparedness.

The real-world instance is based on OpenStreetMap (OSM) data of Colombia provided by Geofabrik GmbH Karlsruhe (2020) and was prepared to be used with the developed algorithm using QGIS 3.4.15 with GRASS 7.8.2. First, we deleted road edges with a low hierarchy. Roads with the following hierarchy tags were kept: 'primary', 'primary\_link', 'secondary', 'secondary\_link', 'tertiary', 'tertiary\_link', 'unclassified', 'unknown', 'trunk', 'trunk\_link' and 'track 4 and 5'. In the following, low hierarchy roads were manually added, in case they were essential for connecting major parts of the network. Next, those roads were merged into one object, and isolated islands, i.e. disconnected parts of the network, were removed. Road intersections were extracted and manually checked, to exclude intersections where one road becomes another road (e.g. due to different road types). The victim nodes, defined based on the settlements in the exploration area, e.g. towns and villages, were extracted. The town of Planeta Rica was chosen to be the DMC, as it is located on the edge of the exploration area and at the intersection of two high level roads. In addition, it has a police station, which could be used as a DMC. Nodes were then snapped to the closest kept road edge as their centre was not always directly located at a higher level road. The road object was cut at each intersection and at each victim node to create single edges connecting these nodes. The position and type of the start and end nodes of these edges as well as their length were extracted and used in the algorithm. The real world-based instance has 742 nodes enumerated from 0 to 741 (52 victim locations, 689 road crossings, one DMC) and 814 undirected edges (roads). To create the disruption nodes, the seventh vertex of each road object was extracted. Short or very straight roads with less than seven vertexes were, thus, not included and the maximum number of disrupted edges is therefore 278. In addition, 10 disruption nodes were selected manually to be able to check how a low disruption level affects the routing of the UAV, when important connections are disrupted. According to the results obtained for the artificial instances, the best movement strategy is selected to evaluate its performance under realistic conditions. Therefore, this section presents the performance of the back and forwards movement strategy, considering a UAV with a limited (Super BAT - 643 km) and an unlimited flying range. The experiments of this section are summarized below:

1. Comparison of the orientation strategies: edge length, edge connectivity and edge weight (taking 11 different  $\alpha$  values into account).
2. Refuelling strategies:  $D$ ,  $H$  and  $R$ .

3. Considering the maximum number of 278 disrupted edges, we use 5 disruption levels: {4%, 30%, 50%, 70%, 100%}. The disruption locations for the disruption level of 4% are fixed. For the disruption levels of {30%, 50%, 70%}, we have generated 10 different disrupted road networks (10 seeds).

### 5.4.1 Results of the real world-based instance

In total, 352 numerical experiments have been performed. The exploration routes obtained for each disruption level guarantee the exploration of the road accessibility of the 52 victim locations in the network. As for the artificial test instances, the route length decreases as the disruption level increases. As in the previous analyses, we compute the exploration route for each disruption level, each seed (for a disruption level of 30%, 50% and 70%) and for all  $\alpha$  values. We determine the shortest exploration route for each seed and disruption level and compute the percentage deviation of the length of the other routes. Then, the average over all seeds is reported for each disruption level.

Table 7 shows that if there is a disruption level of 4%, the exploration strategy provides the shortest route when  $\alpha$  is set to 0.8 or 0.9; but taking another  $\alpha$  value does not have a strong impact on the solution quality. Even though, the best  $\alpha$  values are different for the following three disruption levels, it stays clear that the  $\alpha$  value has only a minor influence on the solution quality in the back and forwards strategy. For a disruption level of 100% the  $\alpha$  value has no impact on the solution quality. This can be explained by the limited choice for the orientation strategy. These results underpin our assumptions made about the back and forwards strategy for the artificial test instances; it seems to be a movement strategy, which is almost independent from the parameter setting.

The shortest routes without refuelling over all  $\alpha$  values for each seed and disruption level are used as benchmark routes to evaluate the performance of the refuelling strategies. The general picture is again similar to the artificial test instances. The

**Table 7** Back and forwards movement strategy: average deviation from the shortest exploration route over all seeds for each  $\alpha$  value and disruption level

Disruption level	Conne- ctivity (%)	Weight (%)									Length (%)
		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	
4%*	1	3	2	2	2	2	2	2	<b>0</b>	<b>0</b>	1
30%	<b>4</b>	<b>4</b>	5	6	6	7	7	8	7	6	6
50%	4	3	<b>2</b>	4	3	3	3	3	<b>2</b>	<b>2</b>	3
70%	3	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	2	2	2	2	2	3
100%*	0	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>

The smallest average deviation is marked in bold  
 A maximum of 278 disrupted edges is considered.  
 \*Fixed disrupted edges

results show that strategy  $D$  provides lower deviations, but the number of refuelling operations is higher, while the  $H$  and  $R$  strategies perform less refuelling operations, but the deviations from the benchmark routes for  $H$  and  $R$  are higher.

## 6 Conclusions

This paper presented an online algorithm, which guides the exploration of a road network after a disaster under unknown information about the road network state. It is evaluated if victim locations can be reached from a DMC by road or not. Since time is a critical element in immediate response operations, this information should be known as soon as possible by the disaster managers, in order to find alternative modes to reach the victim locations. The evaluation of the road network is done by a UAV, which has the advantage compared to ground-based vehicles that it is usually faster and is not bound to the road network. Hence, the online algorithm uses the road and the aerial network.

The proposed online algorithm consists of a cutting strategy that removes unnecessary parts of the exploration space each time new information about the road network is obtained. The exploration strategy consists of movement and orientation strategies that are controlled by a labelled network and a sort insertion procedure. This combination leads to essential savings in the exploration time of the UAV.

The online algorithm was implemented in Java and tested on a set of artificial instances and a real world-based instance located in a high-risk natural disaster area in Colombia. The computation times of the online algorithm are suitable to support real-world operations. Depending on the size of the road network the test instances could be solved in a few seconds.

Our aim was to find the best combination of orientation and movement strategies in order to save exploration time. These combinations are benchmarked against each other to prove the quality of the proposed exploration strategy. During our research, we discovered the high importance of a well-designed cutting strategy, in order to provide a solid backbone for the exploration strategies. We conclude that the best exploration strategy involves the edge weight criterion and the back and forwards movement strategy. The edge weight with values in the range of  $[0.2, 0.6]$  and the back and forwards movement strategy provide on average shorter exploration routes. However, in the back and forwards movement strategy, the setting of the  $\alpha$  value has only a minor impact on its performance. This can also be seen as an advantage of this strategy, since the setting of the  $\alpha$  parameter does not influence the solution quality considerably. Nevertheless, we also obtained few solutions where the jumping movement strategy in combination with the edge weight criterion led to better results. This was the case when the disruption level was low and therefore the exploration space tended to be large. Here, that movement strategy can be useful. The orientation strategy is much more important for the jumping movement strategy; hence, the setting of the  $\alpha$  value dominates the performance of this strategy considerably. Therefore, we would recommend the use of the back and forwards movement strategy in combination with the edge weight criterion for practical applications.

The real world-based instance consists of 742 nodes and 814 road edges. In such a complicated road network, it is important that the pilot of a UAV is supported with online information, since she/he cannot decide easily, which parts of the road network are worth to explore next if new information is obtained and of course it is also very complicated to cut the road network accordingly. When we checked the results of the test instances for feasibility, we observed quite often that we would have chosen other—in our opinion more useful—options to move forwards during single steps of the exploration. Nevertheless, that would have led us to worse solutions than that of the algorithm, because we could not take into account all the control mechanisms of the cutting and exploration strategy.

Additionally, the paper presents three refuelling strategies to deal with a limited flying range of the UAV. We assume that the UAV can only be refuelled at the DMC. Those forced returns increase the length of the exploration route. The refuelling strategy where the UAV refuels each time it is at the DMC ( $D$ ) provides shorter routes of the UAV but leads to a higher number of refuelling operations than the strategies  $H$  and  $R$ . Depending on the time required for a refuelling operation, disaster managers can decide on the best strategy.

Further research paths based on the findings presented in this paper could include the use of additional UAVs in the exploration process. Hence, the routing decisions of these UAVs need to be aligned continuously to avoid unnecessary explorations. It could be also advantageous to setup other refuelling locations than the DMC in disaster-prone areas in the pre-disaster stage. This could reduce the mileage of a UAV during the exploration. Furthermore, the solution approach could be extended by considering a multi-period exploration in order to catch the dynamism of the disaster. To conclude, this research can be extended in some interesting research areas dealing with inter-dependencies of decisions in a dynamic environment, as it is often the case in disaster management.

**Acknowledgements** This work has been partially supported by the Österreichische Nationalbank (OeNB) (17703) and the Erasmus+ Program (2018-1-ES01-KA103-049767). We also acknowledge the support of the UPNA doctoral program and the PID2019-111100RB-C22/AEI/10.13039/501100011033 project.

**Funding** Open access funding provided by University of Natural Resources and Life Sciences Vienna (BOKU).

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## Appendix 1

See Tables 8, 9, 10



**Table 8** Average deviation from the shortest exploration route over all movement strategies

Disruption level	Instance	Back and forwards movement strategy (%)										Jumping movement strategy (%)											
		0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
0.3	p1.2.b	19	20	14	13	13	13	13	13	14	14	14	29	29	38	37	38	36	33	30	24	26	
	p2.2.a	11	10	10	15	19	19	19	19	25	25	25	76	58	43	46	45	55	60	93	159	166	
	p3.2.a	8	7	6	8	8	8	8	8	8	7	7	63	44	23	27	17	36	35	39	41	45	46
	p4.2.a	15	14	13	10	11	12	7	8	8	8	8	111	99	105	108	134	115	120	105	131	118	140
	p5.2.a	26	20	19	27	27	26	26	26	26	21	20	29	29	16	36	35	34	34	45	48	50	55
	p6.2.a	6	7	8	8	8	6	6	10	12	13	13	130	111	109	110	100	97	96	65	106	62	93
	p7.2.a	2	3	3	3	4	4	4	4	4	4	3	4	173	189	174	167	173	167	133	105	88	97
Average	12	11	<b>10</b>	12	12	13	12	13	14	13	14	13	87	80	72	76	77	77	73	<b>69</b>	86	80	89
0.5	p1.2.b	8	4	3	4	4	4	4	4	4	4	4	104	107	121	80	60	56	52	58	57	55	
	p2.2.a	11	11	11	11	11	5	5	5	5	5	5	129	89	83	83	106	49	32	39	65	66	
	p3.2.a	6	6	6	6	6	6	6	6	6	5	5	116	101	90	67	68	92	76	68	68	73	78
	p4.2.a	4	2	3	7	7	7	5	5	5	5	5	212	177	238	188	199	138	133	141	172	178	177
	p5.2.a	5	8	6	6	7	7	7	7	7	6	6	109	133	39	38	36	44	45	46	53	157	77
	p6.2.a	1	1	1	1	1	1	2	4	4	4	4	92	127	117	111	108	122	100	96	175	90	109
	p7.2.a	9	8	8	8	11	11	11	10	11	11	11	277	355	235	247	320	307	322	433	455	467	433
Average	6	6	6	6	7	6	<b>5</b>	6	6	6	6	149	156	132	116	128	116	<b>109</b>	126	149	155	142	
0.7	p1.2.b	5	6	6	1	1	1	1	1	1	1	1	37	104	56	36	38	41	40	41	41	42	35
	p2.2.a	14	13	13	13	13	2	11	11	10	10	10	68	42	51	51	51	25	52	52	64	64	64
	p3.2.a	0	0	0	0	0	0	0	0	0	0	0	86	85	38	37	40	39	37	36	38	42	42
	p4.2.a	0	0	0	10	6	6	6	6	6	6	6	144	207	218	185	223	207	185	187	220	221	277
	p5.2.a	2	1	0	1	1	1	1	1	1	1	1	180	200	92	210	248	258	302	317	378	439	417
	p6.2.a	1	1	1	1	1	1	1	1	1	1	0	97	97	104	104	107	130	99	109	153	154	117
	p7.2.a	9	9	9	9	9	9	9	9	9	9	9	248	177	122	124	209	237	230	371	366	405	404
Average	5	4	4	5	4	<b>3</b>	4	4	4	4	4	123	130	<b>97</b>	107	131	134	135	159	180	195	194	

The smallest average over all instances is marked in bold

**Table 9** Back and forwards movement strategy: average deviation from the shortest exploration route over all seeds for each instance,  $\alpha$  value and disruption level

Disruption level (%)	Instance	Min. distance*	Avg. distance*	Max. distance*	Connectivity	Weight (%)										Length
						0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1	
30	p1.2.b	500.33	934.03	1267.93	9	9	3	2	2	2	2	3	3	3	3	3
	p2.2.a	96.45	379.97	709.78	6	6	6	10	14	14	14	21	21	21	21	21
	p3.2.a	318.81	919.23	1499.34	3	1	1	3	3	3	3	3	3	2	2	2
	p4.2.a	700.86	1981.75	3747.72	13	12	12	8	10	11	5	6	6	6	6	6
	p5.2.a	1506.35	1753.46	1943.40	16	11	10	17	17	17	17	17	17	11	11	11
	p6.2.a	56.57	799.97	1423.50	4	5	6	6	6	6	4	7	10	10	11	11
	p7.2.a	69.84	371.69	802.61	2	3	3	3	4	4	4	4	4	3	4	4
Average					8	7	6	7	8	7	8	9	8	8	8	
50	p1.2.b	323.92	526.40	755.39	8	4	3	4	4	4	4	4	4	4	4	
	p2.2.a	88.62	221.43	476.02	11	11	11	11	11	5	5	5	5	5	5	
	p3.2.a	223.01	545.81	980.25	1	1	1	1	0	0	0	0	0	0	0	
	p4.2.a	499.53	1031.25	2570.13	4	2	3	7	7	7	5	5	5	5	5	
	p5.2.a	40.00	1302.00	2362.59	3	6	4	4	6	6	6	6	6	5	5	
	p6.2.a	56.57	343.09	749.03	1	1	1	1	1	1	2	4	4	4	4	
	p7.2.a	54.98	94.74	323.44	1	0	0	0	2	2	2	2	2	2	2	
Average				4	4	3	4	5	4	3	4	4	4	4		
70	p1.2.b	269.65	385.86	655.35	5	6	6	1	1	1	1	1	1	1	1	
	p2.2.a	88.62	133.93	196.64	11	10	10	10	10	0	9	9	8	8	8	
	p3.2.a	223.01	321.42	756.29	0	0	0	0	0	0	0	0	0	0	0	
	p4.2.a	346.46	558.95	779.46	0	0	0	10	6	6	6	6	6	6	6	
	p5.2.a	40.00	333.21	891.88	2	1	0	1	1	1	1	1	1	1	0	
	p6.2.a	56.57	121.30	468.42	1	1	1	1	1	1	1	1	1	1	0	
	p7.2.a	49.59	62.13	76.22	1	0	0	0	0	0	0	0	0	0	0	
Average				3	3	3	3	3	1	3	2	2	2	2		

The smallest average over all instances is marked in bold

\*Over all 10 seeds for each instance, disruption level and  $\alpha$  value

**Table 10** Jumping movement strategy: average deviation from the shortest exploration route over all seeds for each instance,  $\alpha$  value and disruption level

Disruption level (%)	Instance	Min. distance*	Avg. distance*	Max. distance	Connectivity	Weight (%)										Length
						0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1	
30	p1.2.b	693.71	916.32	1139.81	17	15	24	23	24	22	22	20	18	13	15	
	p2.2.a	115.43	395.40	609.27	35	33	21	23	22	32	36	54	109	115	117	
	p3.2.a	318.81	904.34	1399.92	56	37	15	20	11	29	27	32	34	36	37	
	p4.2.a	1325.14	2566.04	3697.96	22	16	16	21	35	26	28	22	33	28	39	
	p5.2.a	1398.28	1752.87	2271.74	20	21	8	27	26	26	25	36	38	41	45	
	p6.2.a	56.57	853.88	1286.96	56	51	50	52	43	45	30	19	29	21	30	
	p7.2.a	69.84	518.76	890.14	91	112	105	100	104	100	64	34	25	30	31	
Average				42	41	34	38	38	40	33	31	41	41	45		
50	p1.2.b	339.16	707.70	1011.35	57	57	67	29	16	14	10	14	14	14	12	
	p2.2.a	88.62	273.62	509.75	76	48	43	43	66	14	4	7	32	32	33	
	p3.2.a	223.01	618.66	1110.39	83	68	54	38	36	58	44	37	37	42	47	
	p4.2.a	872.76	1729.41	2930.33	68	49	80	60	64	31	28	34	55	47	53	
	p5.2.a	40.00	1611.23	2809.39	81	106	9	8	8	12	12	14	19	124	40	
	p6.2.a	56.57	517.21	969.61	12	29	25	21	19	24	18	19	40	15	19	
	p7.2.a	49.23	177.82	485.14	145	145	44	46	164	157	144	283	299	322	289	
Average				74	72	46	35	53	44	37	58	71	85	70		
70	p1.2.b	269.65	441.94	868.75	21	85	40	22	23	25	24	26	26	26	20	
	p2.2.a	88.62	147.91	226.48	45	19	28	28	28	12	38	38	48	48	48	
	p3.2.a	223.01	349.76	804.08	75	71	30	28	30	29	27	27	29	33	33	
	p4.2.a	346.46	873.49	1397.82	67	114	124	100	134	130	103	105	130	132	174	
	p5.2.a	40.00	611.85	1623.00	60	109	9	72	90	101	111	124	151	254	175	
	p6.2.a	56.57	215.18	685.85	20	20	24	24	26	33	17	20	32	29	16	
	p7.2.a	46.91	111.21	256.84	78	69	19	19	81	98	91	224	206	249	246	
Average				52	70	39	42	59	61	59	80	89	110	102		

The smallest average over all instances is marked in bold

\*Over all 10 seeds for each instance, disruption level and  $\alpha$  value

## References

- Akkermans H, Van Wassenhove LN (2018) Supply chain tsunamis: research on low-probability, high-impact disruptions. *J Supply Chain Manag* 54(1):64–76. <https://doi.org/10.1111/jscm.12162>
- Anaya-Arenas AM, Renaud J, Ruiz A (2014) Relief distribution networks: a systematic review. *Ann Oper Res* 223(1):53–79. <https://doi.org/10.1007/s10479-014-1581-y>
- Bravo RZB, Leiras A, Cyrino Oliveira FL (2019) The use of UAVs in humanitarian relief: an application of POMDP-based methodology for finding victims. *Prod Oper Manag* 28(2):421–440. <https://doi.org/10.1111/poms.12930>
- Centre for a New American Security (2019) The Drone Database. <http://drones.cnas.org/drones/>. Accessed 2020-10-24
- Chao IM, Golden BL, Wasil EA (1996) The team orienteering problem. *Eur J Oper Res* 88(3):464–474. [https://doi.org/10.1016/0377-2217\(94\)00289-4](https://doi.org/10.1016/0377-2217(94)00289-4)
- Clarke PK, Campbell L (2015) Exploring coordination in humanitarian clusters. ALNAP Study, London
- Coutinho WP, Battarra M, Fliege J (2018) The unmanned aerial vehicle routing and trajectory optimisation problem, a taxonomic review. *Comput Ind Eng* 120:116–128. <https://doi.org/10.1016/j.cie.2018.04.037>
- Coutinho WP, Fliege J, Battarra M (2019) Glider routing and trajectory optimisation in disaster assessment. *Eur J Oper Res* 274(3):1138–1154. <https://doi.org/10.1016/j.ejor.2018.10.057>
- Dascioglu BG, Vayvay O, Kalender ZT (2019) Humanitarian supply chain management: Extended literature review. In: *Industrial engineering in the big data era*. Springer, Berlin, pp 443–459. [https://doi.org/10.1007/978-3-030-03317-0\\_36](https://doi.org/10.1007/978-3-030-03317-0_36)
- Daud MSM, Hussein M, Nasir M, Abdullah R, Kassim R, Suliman M, Saludin M (2016) Humanitarian logistics and its challenges: the literature review. *Int J Supply Chain Manag* 5(3):107–110
- do Martins LC, Hirsch P, Juan AA (2021) Agile optimization of a two-echelon vehicle routing problem with pickup and delivery. *Int Trans Oper Res* 28(1):201–221. <https://doi.org/10.1111/itor.12796>
- Eftekhari M, Masini A, Robotis A, Van Wassenhove LN (2014) Vehicle procurement policy for humanitarian development programs. *Prod Oper Manag* 23(6):951–964. <https://doi.org/10.1111/poms.12108>
- Elçi Ö, Noyan N (2018) A chance-constrained two-stage stochastic programming model for humanitarian relief network design. *Transp Res Part B Methodol* 108:55–83. <https://doi.org/10.1016/j.trb.2017.12.002>
- Estrada MAR, Ndoma A (2019) The uses of unmanned aerial vehicles-UAV's-(or drones) in social logistic: natural disasters response and humanitarian relief aid. *Procedia Comput Sci* 149:375–383. <https://doi.org/10.1016/j.procs.2019.01.151>
- Farahani RZ, Lotfi M, Baghaian A, Ruiz R, Rezapour S (2020) Mass casualty management in disaster scene: a systematic review of OR&MS research in humanitarian operations. *Eur J Oper Res* 287(3):787–819. <https://doi.org/10.1111/itor.12783>
- FSD (2019) Drones in humanitarian actions. Technical report, Fondation Suisse de Déménagement (FSD), Geneva. <https://fsd.ch/en/innovation-and-technology>. In: section 3, pp 24–33. Accessed: 2020-10-24
- Geofabrik GmbH Karlsruhe (2020) Geofabrik. <http://www.geofabrik.de/>. Accessed: 2020-10-24
- Guimera R, Mossa S, Turtschi A, Amaral LN (2005) The worldwide air transportation network: anomalous centrality, community structure, and cities' global roles. *Proc Nat Acad Sci* 102(22):7794–7799. <https://doi.org/10.1073/pnas.0407994102>
- Günneç D, Salman FS (2011) Assessing the reliability and the expected performance of a network under disaster risk. *OR Spectrum* 33(3):499–523. <https://doi.org/10.1007/s00291-011-0250-7>
- Hatefi S, Jolai F (2014) Robust and reliable forward-reverse logistics network design under demand uncertainty and facility disruptions. *Appl Math Model* 38(9–10):2630–2647. <https://doi.org/10.1016/j.apm.2013.11.002>
- Hatefi SM, Jolai F, Torabi SA, Tavakkoli-Moghaddam R (2015) A credibility-constrained programming for reliable forward-reverse logistics network design under uncertainty and facility disruptions. *Int J Comput Integr Manuf* 28(6):664–678. <https://doi.org/10.1080/0951192X.2014.900863>
- HOT (2019) Disaster management through geo-spatial data: DMI. [www.hotosm.org](http://www.hotosm.org). Accessed 2020-10-24
- Huang M, Smilowitz KR, Balcik B (2013) A continuous approximation approach for assessment routing in disaster relief. *Transp Res Part B: Methodol* 50:20–41. <https://doi.org/10.1016/j.trb.2013.01.005>

- Kovács G, Spens KM (2007) Humanitarian logistics in disaster relief operations. *Int J Phys Distrib Logist Manag* 37(2):99–114. <https://doi.org/10.1108/09600030710734820>
- Kunz N, Van Wassenhove LN (2019) Fleet sizing for UNHCR country offices. *J Oper Manag* 65(3):282–307. <https://doi.org/10.1002/joom.1013>
- Leiras A, de Brito Jr I, Queiroz Peres E, Rejane Bertazzo T, Tsugunobu Yoshida Yoshizaki H (2014) Literature review of humanitarian logistics research: trends and challenges. *J Humanit Logist Supply Chain Manag* 4(1):95–130. <https://doi.org/10.1108/JHLSCM-04-2012-0008>
- Macrina G, Pugliese LDP, Guerriero F, Laporte G (2020) Drone-aided routing: a literature review. *Transp Res C: Emerg Technol*. <https://doi.org/10.1016/j.trc.2020.102762>
- Maya Duque P, Sörensen K (2011) A grasp metaheuristic to improve accessibility after a disaster. *OR Spectrum* 33(3):525–542. <https://doi.org/10.1007/s00291-011-0247-2>
- Murphy RR, Tadokoro S, Kleiner A (2016) Disaster robotics. In: Springer handbook of robotics. Springer, Berlin, pp 1577–1604
- Nolz PC, Doerner KF, Gutjahr WJ, Hartl RF (2010) A bi-objective metaheuristic for disaster relief operation planning. *Stud Comput Intell* 272:167–187. [https://doi.org/10.1007/978-3-642-11218-8\\_8](https://doi.org/10.1007/978-3-642-11218-8_8)
- Noyan N, Balci B, Atakan S (2015) A stochastic optimization model for designing last mile relief networks. *Transp Sci* 50(3):1092–1113. <https://doi.org/10.1287/trsc.2015.0621>
- Noyan N, Meraklı M, Küçükayavuz S (2017) Two-stage stochastic programming under multivariate risk constraints with an application to humanitarian relief network design. *Math Program*. <https://doi.org/10.1007/s10107-019-01373-4>
- OCHA (2019) Un disaster assessment and coordination (undac). Technical report, Office for the coordination of humanitarian affairs, available at: <https://www.unocha.org/our-work/coordination/un-disaster-assessment-and-coordination-undac>. Accessed: 2020-10-24
- Oruc BE, Kara BY (2018) Post-disaster assessment routing problem. *Transp Res Part B: Methodol* 116:76–102. <https://doi.org/10.1016/j.trb.2018.08.002>
- Özdamar L, Ertem MA (2015) Models, solutions and enabling technologies in humanitarian logistics. *Eur J Oper Res* 244(1):55–65. <https://doi.org/10.1016/j.ejor.2014.11.030>
- Pamukcu D, Balci B (2020) A multi-cover routing problem for planning rapid needs assessment under different information-sharing settings. *OR Spectrum*. <https://doi.org/10.1007/s00291-019-00563-y>
- PwC and Agoria (2018) A drone's eye view. Technical report companies associated in PricewaterhouseCoopers International Limited (PwCIL), Belgium. <https://www.pwc.be/en/documents/20180518-drone-study.pdf>. Accessed: 2020-10-24
- Rath S, Gutjahr WJ (2014) A math-heuristic for the warehouse location-routing problem in disaster relief. *Comput Oper Res* 42:25–39. <https://doi.org/10.1016/j.cor.2011.07.016>
- Rojas Viloria D, Solano-Charris E, Muñoz-Villamizar A, Montoya-Torres J (2020) Unmanned aerial vehicles/drones in vehicle routing problems: a literature review. *Int Trans Oper Res*. <https://doi.org/10.1111/itor.12783>
- Sah B, Gupta R, Bani-Hani D (2020) Analysis of barriers to implement drone logistics. *Int J Logist Res Appl*. <https://doi.org/10.1080/13675567.2020.1782862>
- Shakhatareh H, Sawalmeh AH, Al-Fuqaha A, Dou Z, Almaita E, Khalil I, Othman NS, Khreishah A, Guizani M (2019) Unmanned aerial vehicles (UAVs): a survey on civil applications and key research challenges. *IEEE Access* 7:48572–48634. <https://doi.org/10.1109/ACCESS.2019.2909530>
- Shao J, Wang X, Liang C, Holguín-Veras J (2020) Research progress on deprivation costs in humanitarian logistics. *Int J Disaster Risk Reduct*. <https://doi.org/10.1016/j.ijdrr.2019.101343>
- Tanzi TJ, Chandra M, Isnard J, Camara D, Sebastien O, Harivelo F (2016) Towards “drone-borne” disaster management: future application scenarios. In: ISPRS annals of the photogrammetry, remote sensing and spatial information sciences vol 3, pp 181–189. <https://doi.org/10.5194/isprs-annals-III-8-181-2016>
- Torabi S, Namdar J, Hatefi S, Jolai F (2016) An enhanced possibilistic programming approach for reliable closed-loop supply chain network design. *Int J Prod Res* 54(5):1358–1387. <https://doi.org/10.1080/00207543.2015.1070215>
- WeRobotics (2019) Open AI caribbean challenge: mapping disaster risk from aerial imagery. <https://blog.werobotics.org/2019/10/08/open-ai-caribbean-challenge-mapping-disaster-risk-from-aerial-image-ry/>. Accessed 2020-10-24
- Yanmaz E, Yahyanejad S, Rinner B, Hellwagner H, Bettstetter C (2018) Drone networks: communications, coordination, and sensing. *Ad Hoc Netw* 68:1–15. <https://doi.org/10.1016/j.adhoc.2017.09.001>

- Yu X, Li C, Zhou J (2020) A constrained differential evolution algorithm to solve UAV path planning in disaster scenarios. *Knowl Based Syst.* <https://doi.org/10.1016/j.knosys.2020.106209>
- Zhalechian M, Torabi SA, Mohammadi M (2018) Hub-and-spoke network design under operational and disruption risks. *Transp Res Part E: Logist Transp Rev* 109:20–43. <https://doi.org/10.1016/j.tre.2017.11.001>
- Zhang X, Han X, Li C, Tang X, Zhou H, Jiao L (2019) Aerial image road extraction based on an improved generative adversarial network. *Remote Sens.* <https://doi.org/10.3390/rs11080953>

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

## Affiliations

Lorena Reyes-Rubiano<sup>1</sup>  · Jana Voegl<sup>1</sup>  · Klaus-Dieter Rest<sup>1</sup>  ·  
Javier Faulin<sup>2</sup>  · Patrick Hirsch<sup>1</sup> 

Lorena Reyes-Rubiano  
lorena.reyes-rubiano@boku.ac.at

Jana Voegl  
jana.voegl@boku.ac.at

Klaus-Dieter Rest  
klausdieter.rest@boku.ac.at

Javier Faulin  
javier.faulin@unavarra.es

<sup>1</sup> Institute of Production and Logistics, University of Natural Resources and Life Sciences, Vienna, Austria

<sup>2</sup> Institute of Smart Cities, Public University of Navarra, Pamplona, Spain