



# Integrated optimal scheduling and routing of repair crew and relief vehicles after disaster: a novel hybrid solution approach

Sakineh Lakzaei<sup>1</sup> · Donya Rahmani<sup>1</sup> · Babak Mohamadpour Tosarkani<sup>2</sup> · Sepideh Nasiri<sup>1</sup>

Accepted: 11 May 2023 / Published online: 10 June 2023

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2023

## Abstract

After a natural disaster or a large-scale incident, some of the utility lines and essential roads around the damaged centers may be blocked by debris from the disaster. This incident can immediately restrict crucial supply distribution and affect rescue plans. Therefore, the recovery of damaged roads is an essential step to help increase the effectiveness of the relief operations and to reduce victims' suffering. In this study, we develop a mixed-integer programming model for optimal scheduling and routing of repair crew and relief vehicles after the disaster. The objective function is to minimize the total relief time of the disaster-affected nodes and the recovery time of damaged nodes. Demand nodes are manually weighed according to a presumed degree of damage. Given the NP-hardness nature of the problem, a hybrid approach is proposed based on the Benders decomposition method and a heuristic algorithm. The heuristic algorithm is employed as an initiation algorithm to generate an initial solution for the Benders decomposition algorithm. The routing of the repair crew and relief vehicles is drawn using the Benders decomposition algorithm. To demonstrate the performance of the hybrid algorithm, an ant colony optimization algorithm is also developed. The computational results demonstrate that the proposed approach is capable of computing a near-optimal solution in less CPU time. Furthermore, the performance of this hybrid approach is distinctively superior when it comes to routing of medium to large-scale problems.

**Keywords** Disaster · Debris management · Scheduling · Road restoration · Routing · Benders decomposition · Ant colony optimization

---

✉ Donya Rahmani  
drahmani@kntu.ac.ir

Sakineh Lakzaei  
lakzaie@email.kntu.ac.ir

Babak Mohamadpour Tosarkani  
babak.tosarkani@ubc.ca

Sepideh Nasiri  
nasiri@kntu.ac.ir

<sup>1</sup> Department of Industrial Engineering, K. N. Toosi University of Technology, Tehran, Iran

<sup>2</sup> School of Engineering, University of British Columbia, Kelowna, BC, Canada

## 1 Introduction and literature review

Natural disasters cause thousands of deaths annually and damage to personal property and public infrastructure (Thomas & Kopczak, 2005). The rapid growth of the world's population, along with the increase in population density in hazardous environments, has led to increased frequency and severity of natural disasters. In addition to humans' and financial losses, leaving large amounts of debris in the affected areas would slow down relief and reconstruction operations in the aftermath of disasters. Restoring accessibility in the damaged area is serious after a disaster (Kasaei & Salman, 2016). Debris management investigates road restoration operations by the repair crew by determining the sequence of visits to the damaged areas. The problem of integrated optimal scheduling and routing of the repair crew and relief vehicles after the disaster (IOSRRCRVD) is one of debris management's most significant and severe issues. This problem considers a wide geographical area where roads are spread out. Most urban areas are densely populated and physically connected by several roads and routes. In cases where many communication routes exist between different areas, disaster response can be achieved by receiving relief aid and transferring victims through the undamaged roads and routes. However, in rural areas with fewer roads and communication routes, disasters can destroy the main roads, thereby isolating some areas. With the isolation of the people and the destruction of the roads, it will be difficult to deliver aid to these areas (Aksu & Ozdamar, 2014). Furthermore, the lack of sufficient resources makes it impossible to rebuild and remove debris from all roads simultaneously. For this reason, repair of the destroyed roads should be done based on specific scheduling to have access to demand nodes.

Aksu and Ozdamar (2014) developed a dynamic path-based mathematical model for maximizing network accessibility to rescue survivors, repair damage, and clean up roadside debris. In their study, a path contains a sequence of links between each source and destination, which is assumed to be open when all links are empty of debris. A weight is assigned to prioritize each route, indicating the urgency of retrieving each route based on the expected traffic flow on that route. Their model output is a retrieval schedule for clearing the blocked links. They also developed an algorithm for solving their proposed retrieval problem in a reasonable time.

Yan and Shih (2009) presented an integrated road restoration and relief distribution model to minimize operation completion time. The model applies time-based network flow considering repair crew trips and relief flows along the reconstructed roads. Their three-step heuristic approach includes prioritizing the blocked links, crew scheduling, and optimization of material flow. In their proposed heuristic-phased approach, scheduling repair of the damaged nodes and optimizing the flow of relief aid are not considered simultaneously.

Furthermore, Yan and Shih (2012) developed an integer network flow model with side constraints to address the problem of road repair. They also presented a hybrid algorithm based on ant colony algorithms to solve large-scale problems to compensate for the limitations of the previous studies. They applied their proposed approach to the Chi-Chi earthquake in Taiwan.

Duque et al. (2013) presented a method with improving path accessibility with an integer linear programming model, in which resources must be allocated to improve the accessibility of network vertices. They proposed two solution approaches, one based on the Knapsack problem and the other based on the variable neighborhood search (VNS). They validated the performance of the two approaches with 480 random examples. Finally, they emphasized the importance of allocating limited resources to improve road accessibility.

Kim et al. (2017) improved the model proposed by Duque et al. (2013) from a short-term perspective, in which dimensions of damage change at specific times. A short-term perspective on the post-disaster relief chain is the situation immediately after a disaster, in which disaster-affected events are still occurring and changing. In this situation, accessibility of the supply chain is vital for the timely delivery of relief aid, rescuing victims, and transferring casualties. They presented a mixed-integer linear programming model minimizing the total weighted damage in the isolated areas and also completion time by the repair crew. A single repair crew is provided to obtain an optimal solution for reducing the complexity of the problem. They developed an ant colony algorithm that can also solve the problem of multiple repair crews. Their computational results showed that their model reduces further post-disaster damage to a greater extent than the model proposed by Duque et al. (2013).

Avci and Avci (2017) presented a traveling repairman problem with profits which is the extended version of the traveling repairman problem (TRP), in which visiting all nodes is not mandatory. A time-dependent profit is allocated to each node, and the objective function maximizes the total revenue. Their paper developed a mixed integer linear programming model to solve small-scale examples. A metaheuristic approach was presented to solve medium- and large-scale problems, including the greedy randomized adaptive search procedure (GRASP) to create a solution and the iterated local search (ILS) to improve the solution. The metaheuristic approach was tested on examples derived from the literature, and the results demonstrated the effectiveness of their solution approach.

Li et al. (2019) addressed the post-disaster relief problem. They presented a model for solving the rescuer allocation problem by considering several disaster areas, places for rescuers, rescue time, personal preferences, and the rescuers' degree of competence. Personal preferences and aid workers' degree of competence for relief work are combined to determine work relevance for aid workers and relief workers. They presented an integer programming model whose objective is maximizing the total degree of conformity between rescuers and rescue workers. This degree of conformity is obtained by integrating work and timely relevance. Compared to existing methods, in this study, more realistic conditions, such as several disaster areas, inaccuracies in transportation times, personal preferences of rescuers for each rescue work, etc., were considered.

Briskorn et al. (2020) investigated the basic simultaneous road clearance and distribution problem (BSRCDP). The novelty of their research was that in defining the problem, a hard time deadline was considered for delivery of relief goods to demand locations. Each demand node is assigned a certain amount of demand and a specific deadline. They developed a mixed-integer linear programming model and proposed an exact solution approach based on the Branch and Bound Algorithm.

Moreno et al. (2019) studied crew scheduling and routing problem (CSRP) in post-disaster situations and developed a mixed-integer linear programming model to minimize the inaccessibility time of the affected areas. Scheduling decisions in their study included the order in which the crew repairs the nodes blocked by debris in the network. The routing decisions included determining the routes that the crew must visit to repair the damaged nodes. Their proposed model determines when demand nodes are accessible. This time must be less than a predefined acceptable threshold. They proposed a branch-and-Benders-cut algorithm to solve the model. They considered the problem of routing of relief vehicles and determining the exact time of relief to each affected area with some other details.

Furthermore, Moreno et al. (2020) proposed precise, innovative, and hybrid problem-solving approaches to their previous research (Moreno et al. (2019)), including 1. Branch-and-Benders-cut method (BBC), which reinforces the previous approach by using a different variable partitioning scheme and new valid inequalities, 2. simulated annealing algorithm

and genetic algorithm, and 3. a precise hybrid method of BBC that combines the first two approaches. In this study, based on computational results using the modeled examples, it can be said that the proposed algorithms can solve the sample in larger dimensions compared to other methods presented in the literature.

Ulusan and Ergun (2021) investigated the stochastic network recovery problem considering planning restoration activities on infrastructure network links based on Markov decision process formulation.

Concerning simultaneous road repairs and relief aid, Shin et al. (2019) developed another mathematical model based on mixed-integer linear programming for the problem of integrated Optimal Scheduling of Repair Crew and Relief Vehicle after Disaster (IOSRCRVD). They assumed a relief vehicle for all demand nodes with sufficient capacity to service all nodes in the route. The mathematical model proposed by Shin et al. (2019) was coded by FICO XPRESS-IVE version 7.9. They developed an ant colony optimization algorithm to solve the problem in large-scale problems. The relevant works in the field of debris management are summarized in Table 1.

### 1.1 Research gaps and contributions

Considering the literature review, further research is needed to significantly minimize the repair time of the damaged areas and consequent damage in the aftermath of disasters (Altay & Green, 2006). Furthermore, a review of the literature shows that there are significant gaps in this area. Most crew studies have focused primarily on reducing the inaccessibility of the affected areas. To further minimize losses, it is crucial to consider repairing inaccessible areas and scheduling relief vehicles to demand areas. If road repairs and relief are not considered simultaneously, operations will not be optimal, either in scheduling reconstruction of the blocked roads or providing relief aid to demand areas. Relief aid depends on the scheduling of the repair crew, as it is only possible if roads are unblocked. Therefore, the repair crew should be sent to the mission on a scheduling basis with an accurate attitude to relief because operations, such as providing relief aid, largely depend on repair time. This inherent concurrence of the two issues has not been observed in many related studies in the literature.

To address this research gap, in this study, a mathematical model is proposed for integrating crew scheduling and routing to repair the damaged roads and also considering routing of relief vehicles to demand areas. Significant contribution of this study is providing a new mathematical model for IOSRCRVD. The proposed model investigates post-disaster debris management and relief for the affected areas simultaneously. The objective function of this model is to minimize the arrival time of relief vehicles to demand nodes that have been weighed according to the degree of deterioration of each demand node and minimize the repair time of the damaged nodes in the network. The other main contributions of this study are as follows:

- Novel integrated mathematical formulation with fewer variables and constraints in comparison with the literature
- Considering the priority of each demand node and the existence of the destroyed nodes in the path
- Multi relief vehicle routing for demand nodes from depot simultaneously
- Proposing an efficient novel hybrid solution approach based on the Benders decomposition algorithm and a previously proposed heuristic algorithm
- Proposing a new heuristic method to provide initial incumbent solutions to accelerate the convergence of the solution methods

**Table 1** Summary of existing studies related to debris management

References	Decision				Modeling							Objective function		Solution approach		
	Location	Repair Routing	Relief Routing	Scheduling	Allocation	LP	IP	MIP	RMIP	MINLP	Single	Multi	Exact solution approach	Decision-support tool	Heuristics & MetaHs	
Yan and Shih (2012)	✓			✓			✓					✓				✓
Duque et al. (2013)				✓			✓				✓					✓
Şahin (2013)		✓		✓			✓				✓					✓
Aksu and Ozdamar (2014)				✓			✓					✓				✓
Pramudita and Taniguchi (2014)		✓								✓						✓
Özdamar et al. (2014)				✓					✓							✓
Lorca et al. (2015)	✓			✓						✓				✓		✓
Çelik et al. (2015)				✓		✓										✓

Table 1 (continued)

References	Decision		Modeling							Objective function		Solution approach			
	Location	Repair Routing	Relief Routing	Scheduling	Allocation	LP	IP	MIP	RMIP	MINLP	Single	Multi	Exact solution approach	Decision-support tool	Heuristics & MetaHs
						Allocation	LP	IP	MIP	RMIP	MINLP	Single			
Aveci and Aveci (2017)	✓						✓				✓				✓
Kim et al. (2017)			✓				✓				✓				✓
Lorca et al. (2017)	✓									✓	✓			✓	
Habib and Sarkar (2017)	✓				✓		✓				✓			✓	
Akbari and Salman (2017)		✓		✓			✓					✓			✓
Wang et al. (2018)				✓	✓		✓				✓				✓
Moreno et al. (2019)		✓		✓			✓				✓		✓		
Alinaghian et al. (2019)	✓						✓				✓				✓

**Table 1** (continued)

References	Modeling										Objective function		Solution approach			
	Decision		Repair Routing	Relief Routing	Scheduling	Allocation	LP	IP	MIP	RMIP	MINLP	Single	Multi	Exact solution approach	Decision-support tool	Heuristics & MetaHs
	Location	Allocation														
Shin et al. (2019)	✓		✓					✓			✓					✓
Li et al. (2019)				✓							✓					
Li et al. (2020)	✓		✓					✓								✓
Briskorn et al. (2020)	✓		✓					✓			✓					
Moreno et al. (2020)	✓		✓					✓			✓					✓
This research	✓		✓					✓			✓					✓

The remainder of this study is organized as follows: Sect. 2 introduces the proposed mathematical model and numerical example showing validation of the model; Sect. 3 describes the proposed solution approaches. The computational results are given in Sect. 4 by solving the generated test problems on different scales. Finally, the conclusions and future research are presented in Sect. 5.

## 2 Mathematical formulation

### 2.1 Problem description

This section presents a novel mathematical model for the problem IOSRRCRVD. Since the defined problem deals with several areas and routes; the necessary components are designed by a network model in this paper. The considered IOSRRCRVD problem is defined by a connected and undirected graph  $G = (V, \varepsilon)$ , a set of nodes, and a set of edges (roads) connecting areas. The demand nodes ( $V^d \subset V$ ) represent the affected areas, and the damaged nodes ( $V^r \subset V$ ) specify the destroyed areas in the network blocked by debris and, thus, require debris cleanup. There is a demand for humanitarian assistance in demand nodes ( $d_i$ ). In addition to these nodes, transshipment or intersection nodes may indicate the intersection of two or more edges. There is one depot (node 0) that is a supply node to be connected with the demand nodes and from which the repair crew initially departs to repair the damaged nodes.

Due to many inaccessible areas, the possible starting nodes are also limited for the repair crew and relief vehicles. Since relief vehicles can only cross intact or repaired roads, their starting point should be close to the repair crew. For these reasons, the repair crew and relief vehicles' starting point is considered a depot (Shin et al., 2019). Also, the repair crew departs to repair the damaged nodes at the beginning of the period. One of the main assumptions of this study, as the most relevant investigation to our research, is that there is enough capacity to cover all demand points with one vehicle. This assumption is not practical in many real-world situations. Therefore, we propose a multi-vehicle problem in which a separate schedule is considered for each demand point. In the proposed model, in addition to scheduling the repair crew for the damaged nodes, routing the relief vehicles to each demand node and their exact arrival time are also determined.

Due to the critical situation, it is assumed that enough relief vehicles are available for relief of the demand nodes. These vehicles start moving from the depot, and if there are no damaged nodes on the path from the depot to the demand node, they can start moving from moment zero; otherwise, they must wait until the repair of the damaged nodes. After repairing the last damaged node on the route, that demand node will be accessible, and the relief vehicle can continue on its path. The time to arrive at a demand node is obtained from the repair time of the last damaged node in the path of that demand node plus the passage time of this repaired node to that demand node. Duration of relief to the demand nodes is also considered. If relief duration is added up to arrival time at the demand node, the exact time of completing relief to each demand node will be obtained. The assumptions related to the proposed model are as follows:

- The repair crew works together to repair the damaged nodes. In other words, there is one repair team.
- The repair crew cannot move between nodes that are not connected by an edge.
- The repair crew can cross each node or edge several times.



- Each relief vehicle has sufficient capacity to supply humanitarian assistance to a demand node.
- A relief vehicle can also cross each node or edge several times.

The final decisions in the proposed model are as follows:

1. Scheduling of the repair crew for repairing all the damaged nodes;
2. The routes that the repair crew must travel between two consecutive damaged nodes;
3. Routing of each relief vehicle from depot to each demand node;
4. Determining the exact time for arriving and assisting each demand node.

The proposed problem's objective function is minimizing the arrival time of relief vehicles to demand nodes and minimizing the restoration time of the damaged nodes by debris in the network.

## 2.2 Mathematical modeling

The notations used in the model are as follows:

### Sets

- $V$  Set of nodes  
 $V^d \subset V$  Set of the demand nodes requiring relief goods  
 $V^r \subset V$  Set of the damaged nodes requiring repair  
 $\varepsilon$  Set of edges  
 $\varepsilon k_i \subseteq V$  Set of nodes adjacent to node  $i$

### Parameters

- $d_i$  Relief demand in node  $i$ .  $\forall i \in V^d$   
 $s_i$  Time required to repair the damaged node  $i$ .  $\forall i \in V^r$   
 $t_{ij}$  Travel time between nodes  $i$  and  $j$ .  $\forall i, j \in V$   
 $l_i$  Time required to supply relief goods at demand node  $i$  after the relief vehicle arrives there  $\forall i \in V^d$   
 $M$  Big number

### Decision variables

- $x_{ij}$  = 1 if node  $j$  is repaired immediately after node  $i$ . = 0 otherwise.  $\forall i, j \in V^r \cup \{0\}$   
 $U_{ijkl}$  = 1 if edge between node  $l$  and node  $k$  is used on the path from the damaged node  $i$  or the depot to the damaged node  $j$  or the depot, = 0 otherwise.  $\forall k, l \in V; \forall i, j \in V^r \cup \{0\}$   
 $N_{kij}$  = 1 if node  $k$  is used on the path from the damaged node  $i$  or the depot to the damaged node  $j$  or the depot, = 0 otherwise.  $\forall k \in V; \forall i, j \in V^r \cup \{0\}$   
 $w_{ij}^k$  = 1 if node  $j$  is visited immediately after node  $i$  on the path to demand node  $k$  = 0 otherwise.  $\forall i, j \in V. \forall k \in V^d$   
 $z_i^r$  Exact time at which the reparation of damaged node  $i$  is started.  $\forall i \in V^r$   
 $z_{ij}^d$  Exact time at which the relief vehicle arrives to node  $i$  on the path from the depot to demand node  $j$ .  $\forall i \in V. \forall j \in V^d$   
 $z_i^d$  Exact time at which the demand node  $i$  is serviced.  $\forall i \in V^d$

The variable  $x_{ij}$  determines the scheduling of the repair crew, or in other words, the sequence of the damaged nodes that need to be repaired. It does not specify the crew route, as they are defined only for the sequence of visits to the damaged nodes. The complete path is obtained from variables  $U_{ijkl}$  and  $N_{kij}$ , determine the edges and nodes that must be visited in the route between two consecutive nodes  $i$ - $j$  in the determined scheduling.

On the other hand, the variable  $w_{ij}^k$  determines the exact routing between the depot and each demand node  $k$ . In other words, it determines which nodes must pass through to reach each demand node  $k$  from the depot node. The model is formulated as follows:

$$\text{Min } Z = \sum_{i \in V^d} d_i \cdot z_i^d + \sum_{i \in V^r} z_i^r \tag{1}$$

S.t.:

$$\sum_{j \in V^r} x_{ij} = 1 \quad \forall i \in V^r \cup \{0\} \tag{2}$$

$$\sum_{i \in V^r} x_{ij} = 1 \quad \forall i \in V^r \cup \{0\} \tag{3}$$

$$\sum_{k \in \varepsilon_k} U_{ijik} = x_{ij} \quad \forall i \in V^r \cup \{0\}; \forall j \in V^r \tag{4}$$

$$\sum_{k \in \varepsilon_k} U_{ijkj} = x_{ij} \quad \forall i \in V^r \cup \{0\}; \forall j \in V^r \tag{5}$$

$$\sum_{l \in \varepsilon_k} U_{ijkl} + \sum_{l \in \varepsilon_k} U_{ijlk} = 2N_{kij} \quad \forall i \in V^r \cup \{0\}; \forall j \in V^r; \forall k \in V \setminus \{i, j\} \tag{6}$$

$$N_{kij} \leq x_{ij} \quad \forall i, j \in V^r \cup \{0\}; \forall k \in V \setminus \{i, j\} \tag{7}$$

$$\sum_{i \in \varepsilon_k} w_{ij}^j = 1 \quad \forall j \in V^d \tag{8}$$

$$\sum_{i \in \varepsilon_{k_0}} w_{0i}^i = 1 \quad \forall j \in V^d \tag{9}$$

$$\sum_{i \in \varepsilon_k} w_{ij}^p = \sum_{k \in \varepsilon_k} w_{jk}^p \quad \forall p \in V^d; \forall j \in V^d \tag{10}$$

$$z_{jk}^d \geq z_{ik}^d + t_{ij} - (1 - w_{ij}^k) \cdot M \quad \forall i \in V; \forall j \in V; \forall k \in V^d \tag{11}$$

$$z_{jk}^d \geq z_{ik}^d + t_{ij} + s_i - (1 - w_{ij}^k) \cdot M \quad \forall i \in V^r; \forall j \in V; \forall k \in V^d \tag{12}$$

$$z_{jk}^d \geq z_i^r + t_{ij} + s_i - (1 - w_{ij}^k) \cdot M \quad \forall i \in V^r; \forall j \in V; \forall k \in V^d \tag{13}$$

$$z_j^r \geq z_i^r + \sum_{e, l \in V} U_{ijel} \cdot t_{el} + s_i - (1 - x_{ij}) \cdot M \quad \forall i \in V^r \cup \{0\}; \forall j \in V^r \tag{14}$$

$$z_j^r \geq z_k^r - (1 - N_{kij}) \cdot M \quad \forall i \in V^r \cup \{0\}; \forall j, k \in V^r \tag{15}$$

$$z_i^d \geq z_{ii}^d + l_i \quad \forall i \in V^d \tag{16}$$

$$x_{ij} \in \{0, 1\} \quad \forall i, j \in V^r \cup \{0\} \tag{17}$$

$$U_{eij}, N_{kij} \in \{0, 1\} \quad \forall i \in V^r \cup \{0\}; \forall j \in V^r; \forall k \in V; \forall e \in \varepsilon \tag{18}$$

$$w_{ij}^p \in \{0, 1\} \quad \forall i, j \in V; \forall p \in V^d \tag{19}$$

$$z_i^r \geq 0 \quad \forall i \in V^r \cup \{0\} \quad (20)$$

$$z_{ij}^d \geq 0 \quad \forall i, j \in V^d \quad (21)$$

$$z_i^d \geq 0 \quad \forall i, j \in V^d \quad (22)$$

The objective function (1) consists of two parts; minimizing the total relief time of demand nodes that have been weighted based on their demand and the total restoration time of the damaged nodes.

Constraints (2) and (3) represent that each damaged node must be visited only once during the scheduling of the repair crew. Constraints (4) to (6) establish the flow in the route between the damaged nodes  $i$  and  $j$ . If there is a route between the damaged nodes  $i$  and  $j$  ( $x_{ij} = 1$ ), Constraint (4) is applied to an edge incident on node  $i$  in the route, while Constraint (5) is applied to an edge incident on node  $j$  in the route. In addition, for each node  $k$  in the route from node  $i$  to  $j$  ( $N_{kij} = 1$ ), one edge exits, and one edge enters the node  $k$  in the considered route, applied by Constraint (6). Constraint (7) ensures that no routing is specified if the damaged node  $j$  is not visited after the damaged node  $i$  ( $x_{ij} = 0$ ).

Similarly, for the demand nodes, Constraints (8) and (9) specify that each demand node is visited only once during routing. Constraint (10) also establishes the flow in the route between the depot and each demand node. Constraint (11) specifies the time to reach each demand node.

Constraints (12) and (13) explain the additional assistance time to each demand required for completing the final service time. In other words, these constraints ensure that a damaged node on the route between the depot to the demand node is repaired before the relief vehicle passes through the node. To guarantee this relationship, the main constraint is defined in the form of nonlinear constraint (23) and then is replaced by Constraints (12) and (13) for the purpose of linearization.

$$z_{jk}^d \geq \max(z_{ik}^d, z_i^r) + t_{ij} + s_i - (1 - w_{ij}^k) \cdot M \quad \forall i \in V^r, \forall j \in V, \forall k \in V^d \quad (23)$$

Constraint (14) determines the exact time the damaged nodes are cleaned up. For node  $j$ , the exact time is obtained from the sum of the time at which the previously damaged node  $i$  is repaired plus the passage time between nodes  $i$  and  $j$  and the time it takes for node  $j$  to be repaired. These constraints derived from the model proposed by Moreno et al. (2019) act as sub-tour removal constraints and are based on the Miller-Tucker-Zemlin formula of the traveling salesman problem (Miller et al., 1960), which has some constraints depending polynomially on the number of nodes.

Constraint (15) ensures that any damaged node in the route from node  $i$  to  $j$  must be repaired before node  $j$ ; in other words, the damaged nodes that are not repaired cannot be used to arrive at node  $j$ . Constraint (16) calculates the final time of relief completion for each demand node. Finally, Constraints (17) to (22) impose the domains of the decision variables.

## 2.3 Numerical example

In this section, a numerical example is considered to show the validation the proposed model and analyze its performance. The provided example is solved using BARON solver in GAMS software version 24.7.3 by a PC with 8 GB of RAM.

The numerical example, investigated in this section, is derived from the research presented by Moreno et al. (2019) (Fig. 1). The objective function of the model proposed by Moreno

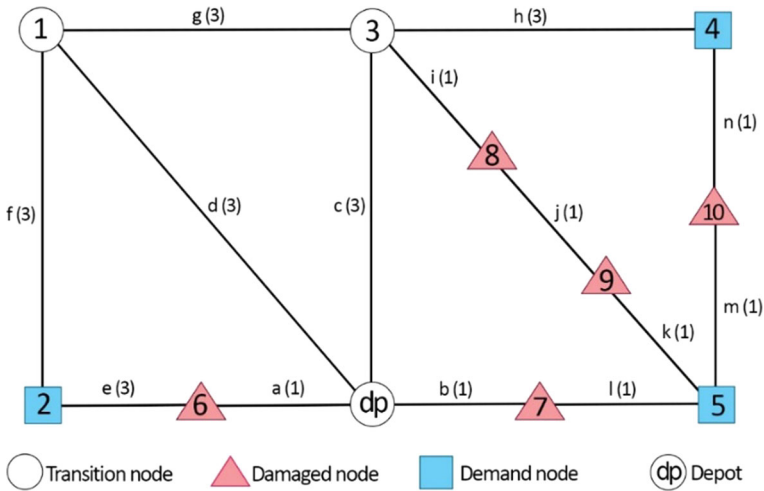


Fig. 1 The numerical example presented by Moreno et al. (2019)

et al. (2019) is minimizing the time at which the demand nodes remain inaccessible from the depot, weighted by their corresponding demands. The accessibility time of the demand nodes will be determined considering the damaged nodes existing in the route from the depot to each demand node and the repair time of the damaged nodes. Modeling is done so that the maximum time to reach each demand node for relief is less than an acceptable time threshold. Consider the CSRPs presented by Moreno et al. (2019) for further details.

Values of the parameters for numerical example are shown in Table 2. Each edge  $(i, j)$  is named by a letter to match the model presented by Moreno et al. (2019), which is shown in Table 2 and Fig. 2.

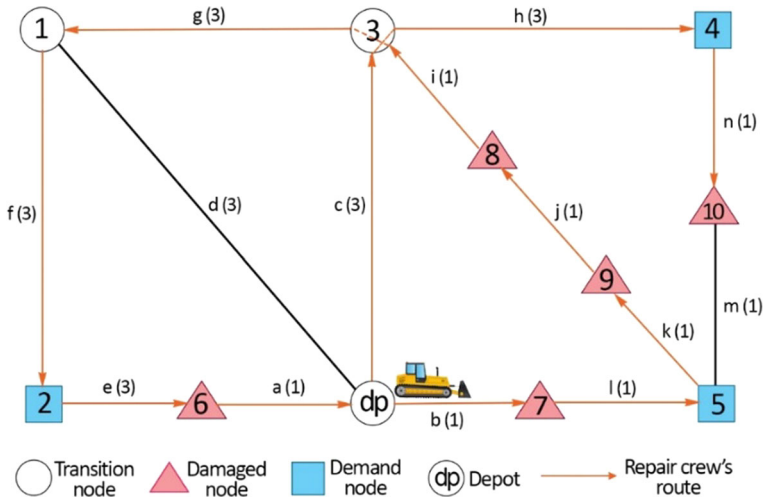
The schedule for repairing the damaged nodes is in the following order: 0-7-9-8-6-10, which are obtained by solving the model presented by Moreno et al. (2019). The route details of the crew to repair the damaged nodes, the order in which the damaged nodes are repaired, and the time to reach each node are provided in Fig. 2 and Table 3.

Furthermore, the nodes and edges on the route from the depot to the demand nodes and accessibility time of each node in numerical example are given in Table 4.

For a better understanding, the routing of each relief vehicle from the depot to each demand node is shown separately in Fig. 3. It is noteworthy that any relief vehicle can start moving from time zero.

Table 2 Values of parameters in the numerical example

Parameter	Value
$d_i$	(2,5), (4,4), (5,3)
$s_i$	(6,1), (7,2), (8,1), (9,2), (10,2)
Travel time	(a,1), (b,1), (c,3), (d,3), (e,3), (f,3), (g,3), (h,3), (I,1), (j,1), (k,1), (l,1), (m,1), (n,1)
$l_i$	(2,2), (4,2), (5,1)



**Fig. 2** Routing details in numerical example based on solving the model proposed by Moreno et al. (2019)

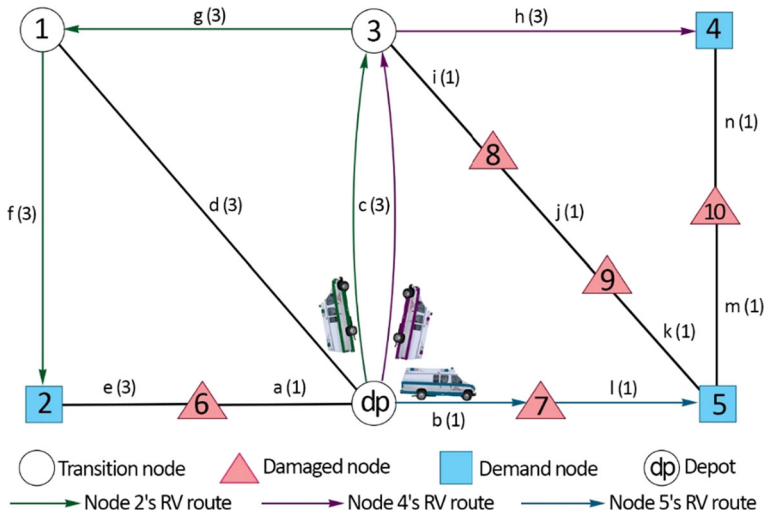
**Table 3** Routing details and scheduling of the damaged nodes in the numerical example obtained by solving the model proposed by Moreno et al. (2019)

Sequence	Damaged Node <i>i</i>	Nodes on the path from the previous node	Edges on the path from the previous node	Exact time at which the node <i>i</i> is repaired
1	7	–	b	3
2	9	5	l, k	7
3	8	–	j	9
4	6	3,1,2	i, g, f, e	20
5	10	0,3,4	a, c, h, n	30

**Table 4** Routing details and accessibility time of the demand nodes in numerical example obtained by solving the model proposed by Moreno et al. (2019)

Demand Node	Nodes on the route	Edges on the path from depot	Exact time at which the node becomes accessible
2	3,1	c, g, f	0
4	3	c, h	0
5	7	b, l	3

As can be seen, there is at least one damaged node on each possible path to the demand node 5. As mentioned earlier, the model proposed by Moreno et al. (2019) calculates the accessibility time of the demand node based on the damaged nodes existing on the path from the depot to the demand node. This accessibility time is equal to the last damaged node’s repair time. Therefore, the accessibility time of the demand nodes 2 and 4 are equal to zero.



**Fig. 3** Routing details of relief vehicles in numerical example obtained by solving of the model proposed by Moreno et al. (2019)

The numerical example provided by Moreno et al. (2019) was also investigated using the model proposed in this study, in which the time required to relieve demand nodes 2, 4, and 5, is equal to 2, 2, and 1, respectively. The order for restoration of the damaged nodes and the time to reach each node based on the proposed model are given in Table 5.

Scheduling and routing of the repair crew to repair the damaged nodes and the time to reach each node are shown on each node in Fig. 4.

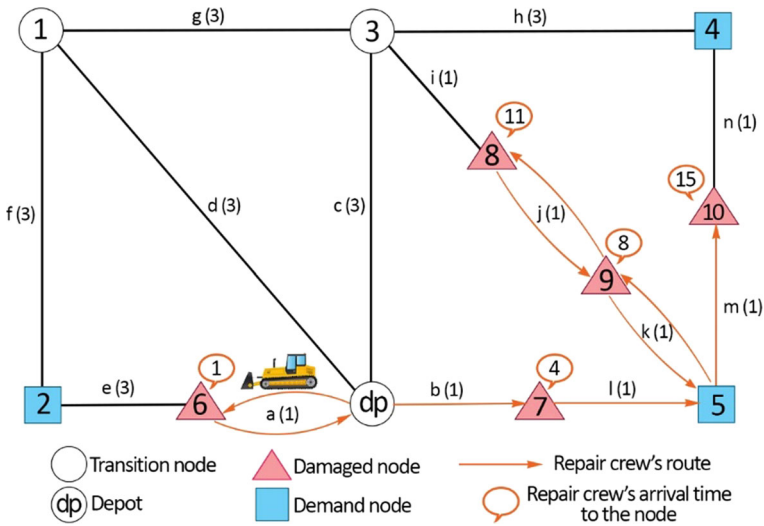
For the demand nodes, the routing of relief vehicles from the depot to each demand node and the exact time of completion of a relief operation are shown in Table 6 and Fig. 5.

Now, to clarify the main contribution of the proposed model, the need for proper modeling regarding the routing of relief vehicles to the demand nodes is taken into account.

For this purpose, changing the length of some edges is considered. If there are no damaged nodes in the path of a demand node, from the moment zero, this demand node is considered “accessible.” In fact, by changing the lengths of the edges in the path of this demand, the “accessibility time” of it does not change, but the real-time of “reaching” to it changes.

**Table 5** Routing details and scheduling of the damaged nodes in numerical example obtained by solving the model proposed in this study

Sequence	Node <i>i</i>	Nodes on the path from the previous damaged node	Edges on the path from the previous damaged node	Exact time at which the node <i>i</i> is visited
1	6	–	A	1
2	7	0	a, b	4
3	9	5	l, k	8
4	8	–	j	11
5	10	9, 5	j, k, m	15



**Fig. 4** Routing of the damaged nodes in numerical example obtained by solving the model proposed in this study

**Table 6** Routing details and scheduling of relief vehicles in numerical example obtained by solving the model proposed in this study

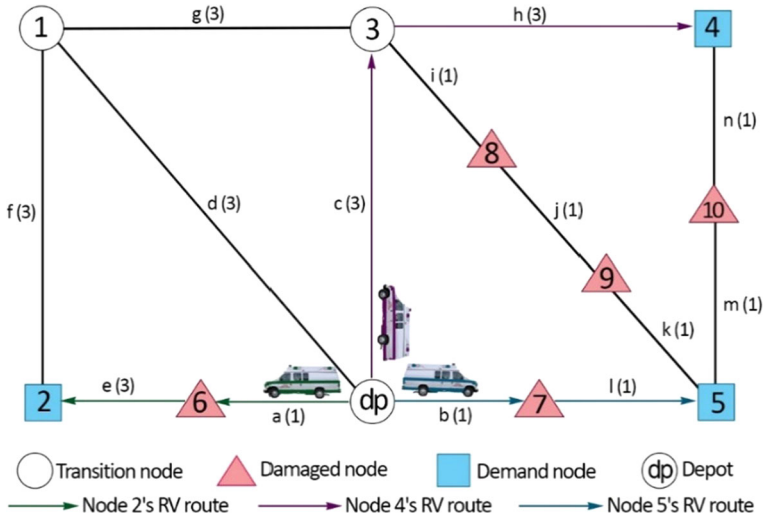
Node	Nodes on the path from depot	Edges on the path from depot	Exact time at which relief vehicle arrives at the node i	Exact time at which the node i is serviced
2	6	a, e	5	7
4	3	c, h	6	8
5	7	b, l	7	8

Such changes can lead to prioritizing repairing the damaged nodes in the route of demand nodes. Our proposed model aims to control all these issues of great importance in real-world problems.

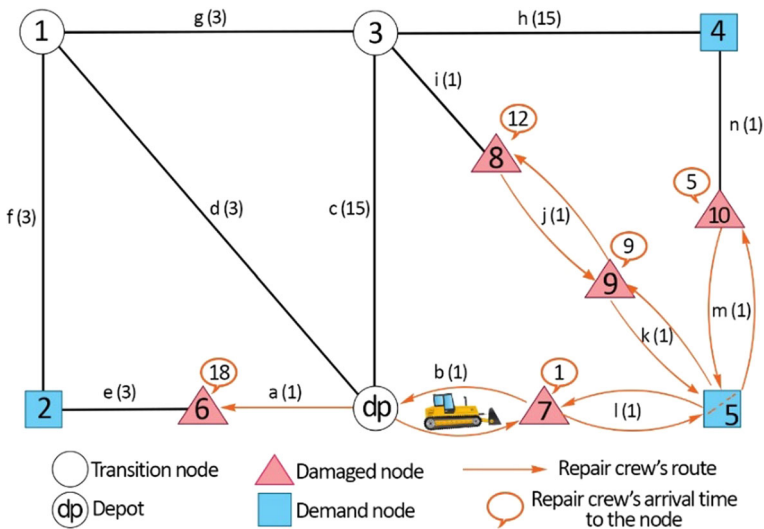
For example, lengths of edges *c* and *h* are increased to 15. After changing the lengths of these edges, Example No.1 is solved again based on the proposed model. The results demonstrate the performance and sensitivity of our proposed model to this problem. The scheduling and routing of the damaged node repair are altered (see Fig. 6).

Due to the increase in length of the edges in the route to the demand node 4 and the lengthening of this route, with the repair of the damaged nodes 7 and 10 (Fig. 6), the proposed model provides a shorter route to reach that demand node, as shown in Fig. 7.

The model proposed in this study can choose between currently open routes and the routes containing the damaged nodes to choose a route that reaches the demand nodes sooner. In the proposed model, considering the length of all edges and the time of repairing the damaged nodes, the exact route and time of relief to each demand node will be calculated, which can be the basis of strategic decisions for disaster management teams.



**Fig. 5** Routing details of relief vehicles in numerical example obtained by solving the model proposed in this study

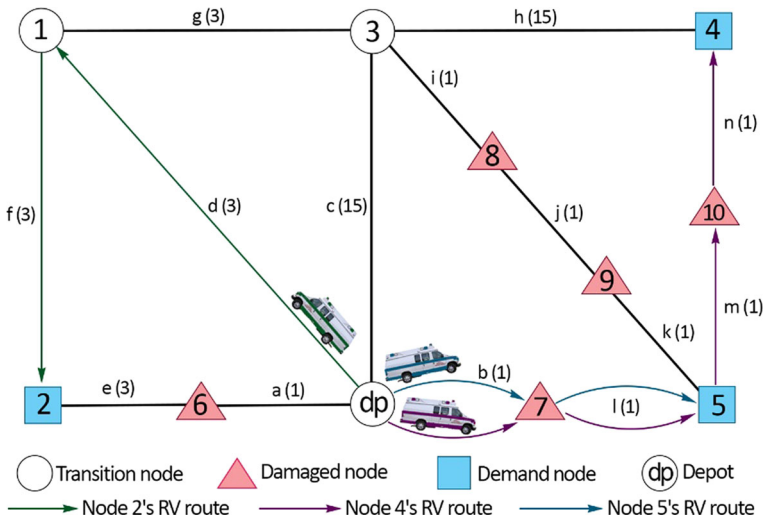


**Fig. 6** Routing details and scheduling of the damaged nodes in numerical example after changing the length of the edges obtained by solving the proposed model in this study

### 3 Solution approaches

Due to the NP-hard nature of the considered problem, a hybrid solution approach is developed based on Benders Decomposition Algorithm and a new heuristic algorithm. The proposed hybrid algorithm tries to find a near-optimal solution in an acceptable CPU time. For solving the problem, it was divided into two phases. The first phase involves determining the





**Fig. 7** Routing details in numerical example after changing the length of the edges obtained by solving the proposed model in this study

scheduling of the repair crew for the damaged nodes, determined using a heuristic algorithm described in Sub-Sect. 3.1. The value of  $X_{ij}$ , as the output of the heuristic algorithm, is considered as a parameter in the next phase. In the second phase, the Benders decomposition algorithm is applied to determine the routing of relief vehicles to service the demand nodes. An ant colony optimization (ACO) is also developed to demonstrate the performance of the proposed hybrid method.

### 3.1 The proposed Heuristic algorithm

As mentioned previously, an efficient algorithm is designed to determine the scheduling of the repair crew of the damaged nodes as an initial solution for the second phase. The steps of this algorithm are as follows:

*Step 0* The network and its related information (including nodes and edges, the time required to cross each edge, demand nodes and their demand values, the damaged nodes, and the time required to repair them) are defined. The repair crew and any relief vehicles are located in the depot, initially. First, a list of all the demand nodes (*DemandNodesList*) is created and they are sorted in descending order based on their demand value starting from the first node.

*Step 1* Using the Dijkstra’s algorithm (Dijkstra, 1959), the shortest path between the depot and demand node ( $Path_a$ ) is determined. The path found for relevant relief vehicle is stored in the variable of *ReliefVehiclePath* [ $i$ ] to be returned as the algorithm’s output.

*Step 2* Completion time of relief to the desired node (*FulfillmentTime* [ $i$ ]) is equal to the time required for passing the determined route plus relief (service) time at the demand node.

*Step 3* Now, if there are any number of damaged nodes in the specified route, the time required for the relief vehicle to reach that node from the depot is called as  $t_a$ . Using the Dijkstras’ algorithm, a path like  $Path_c$  is also determined for the repair crew to reach the damaged node from its current node (*CTCNode*) (at the beginning, this node is the depot

but changes to the last damaged node repaired by the repair crew in any iteration of the algorithm). The found path is stored in the variable of *CrewTeamPath* to be returned at the end as another output.

*Step 4* The arrival time of the repair crew at the damaged node is called as  $t_c$ . Given the existence of only one repair crew, it is necessary to consider the last time of debris cleanup of the repair crew (*LRNTime*) and add it to  $t_c$ . There may also be more than one damaged node in  $Path_c$ , so the time required to repair each damaged node is added to  $t_c$ . If  $t_c$  is greater than  $t_a$ , it means that the relief vehicle will reach the damaged node before the repair crew arrives and removes the debris. In this case, the relief vehicle must wait until the end of cleaning up of debris. This waiting time ( $t_c - t_a$ ) will be added to the service completion time of the relief vehicle.

*Step 5* The network is updated after each debris cleanup, and the damaged nodes that have been repaired will henceforth be considered transition nodes.

*Step 6* Steps of the algorithm are repeated until all the damaged nodes in  $Path_a$  are cleaned up, and in practice, the relief vehicle arrives to the related demand node, and its service is performed. These steps are then repeated for the next demand node in the *DemandNodesList*.

*Step 7* Once all the demand nodes have been serviced, the network is checked for the damaged nodes. If there is any damaged node, it starts finding  $Path_c$  from the first accessible damaged node and calculates the time it takes to arrive and repair the node, and updates *LRNTime*.

The steps are repeated for all the damaged nodes to achieve all problem objectives. The final output of the algorithm, i.e., the completion time of the operations (i.e., *TotalTime*) is calculated by finding the maximum value of the service fulfillment time of the relief vehicles (i.e., *FulfillmentTime*) and the last debris cleanup (i.e., *LRNTime*).

Pseudocode of the heuristic algorithm is described in Display 1.

**Display 1. Pseudocode of the heuristic algorithm****Initializing:**

Graph  $G = (V, \mathcal{E})$ ; (Defining the network and parameters like demands amount, time needed to repair the damaged nodes, time needed to service at demand nodes, etc.)  
 LRNTIME  $\leftarrow 0$ ; (Time at which the last damaged node was repaired)  
 $t_c \leftarrow 0$ ; (Temporary time for crew team movements)

**Output:**

CrewTeamPath (An array of nodes showing movement order of the crew team)  
 ReliefVehiclePath (An array of nodes showing movement order for each demand node)  
 TotalTime (Time of the end of all operations)

**Start:**

```

DemandNodesList  $\leftarrow$  Demand Nodes in V in descending order by demand value;
for (i = 0; i < DemandNodesList.size; i++)
  Patha  $\leftarrow$  DijkstraPath (depot, DemandNodesList[i]);
  Add Patha to the ReliefVehiclePath[i];
  FulfillmentTime[i]  $\leftarrow$  PassTimeCalc(Patha.firstNode, Patha.lastNode)+ServiceTime(i);
  ta  $\leftarrow$  PassTimeCalc(depot, Patha.DamagedNodes.firstNode);
  while Patha.DamagedNodes.count > 0 do
    Pathc  $\leftarrow$  DijkstraPath(CTCNode, Patha.DamagedNodes.firstNode);
    Add Pathc to the end of CrewTeamPath;
    while Pathc.DamagedNodes.count > 0 do
      tc  $\leftarrow$  PassTimeCalc(CTCNode, Pathc.DamagedNodes.firstNode) +
        Pathc.DamagedNodes.firstNode.repairTime + LRNTIME;
      LRNTIME  $\leftarrow$  tc;
      CTCNode  $\leftarrow$  Pathc.DamagedNodes.firstNode;
      Update G;
    end while
  end while
  if tc > ta then
    FulfillmentTime[i]  $\leftarrow$  FulfillmentTime[i] + (tc - ta)
  end if
  tc  $\leftarrow$  0;
end for
while there are still damaged nodes in G do
  Pathc  $\leftarrow$  DijkstraPath (CTCNode, G. Damaged Nodes.firstNode);
  Add Pathc to the end of CrewTeamPath;
  while Pathc.DamagedNodes.count > 0 do
    tc  $\leftarrow$  PassTimeCalc(CTCNode, Pathc.DamagedNodes.firstNode) +
      Pathc.DamagedNodes.firstNode.repairTime + LRNTIME;
    LRNTIME  $\leftarrow$  tc;
    CTCNode  $\leftarrow$  Pathc.DamagedNodes.firstNode;
    Update G;
  end while
end while
TotalTime  $\leftarrow$  max (FulfillmentTime, LRNTIME);
return TotalTime, CrewTeamPath, ReliefVehiclePath

```

This algorithm is coded using the C# Programming language, and its output is used to continue the problem-solving in the second phase via GAMS.

### 3.2 Heuristic-based Benders decomposition algorithm (Hybrid approach)

Benders decomposition algorithm is a decomposition-based technique whose goal is tackling problems with complicating variables (Benders, 1962; Costa, 2005; Martins de Sá et al.,

2013). It was developed by Benders (1962) to solve mixed-integer programming problems. In this method, the problem is divided into smaller problems. Because the computational complexity of mathematical models is increased dramatically with an increasing number of variables and constraints, solving these smaller problems in iterative procedures is more efficient than solving a large problem all at once (Taşkin, 2010).

The general idea behind the Benders decomposition algorithm to divide the problem into two parts: a linear sub-problem and a master problem with complicated integer variables. This algorithm strategy is fixing the hard integer variables in the sub-problem and solving its linear dual sub-problem. Then, the algorithm generates several cuts (constraints) to add to the master problem using the solution obtained from the dual sub-problem. This process continues until the master problem has enough cuts to achieve an optimal solution.

In the model proposed in this study, the variables  $U_{ijkl}$  and  $N_{kij}$ , determining the routing of the repair crew, and the variables  $w_{ij}^p$ , determining the routing of relief vehicles to the demand nodes are considered complicated variables. As mentioned previously, the integer variable  $x_{ij}$ , showing the scheduling of the repair crew for the damaged nodes, is determined by the explained heuristic algorithm in the first phase. So, it is just defined as an input parameter in using the Benders decomposition algorithm.

### 3.2.1 Sub-problem

At the beginning of the Benders decomposition algorithm, the initial feasible solution, or value of the model’s complicated variables, is determined by the heuristic algorithm. They are obtained in each subsequent iteration by solving the master problem in any iteration. The sub-problem of the model presented in this study is defined as follows:

$$(SP) \text{Min} Z_{SP} = \sum_{i \in V^d} d_i \cdot z_i^d + \sum_{i \in V^r} z_i^r \tag{24}$$

S.t.:

$$z_{jk}^d \geq z_{ik}^d + t_{ij} - (1 - \bar{w}_{ij}^k) \cdot M \quad \forall i \in V; j \in V; k \in V^d \tag{25}$$

$$z_{jk}^d \geq z_{ik}^d + t_{ij} + s_i - (1 - \bar{w}_{ij}^k) \cdot M \quad \forall i \in V^r; j \in V; k \in V^d \tag{26}$$

$$z_{jk}^d \geq z_i^r + t_{ij} + s_i - (1 - \bar{w}_{ij}^k) \cdot M \quad \forall i \in V^r; j \in V; k \in V^d \tag{27}$$

$$z_j^r \geq z_i^r + \sum_{e, l \in V} U_{ijel} \cdot t_{el} + s_i - (1 - \bar{x}_{ij}) \cdot M \quad \forall i \in V^r \cup \{0\}; \forall j \in V^r \tag{28}$$

$$z_j^r \geq z_k^r - (1 - \bar{N}_{kij}) \cdot M \quad \forall i \in V^r \cup \{0\} \forall j; k \in V^r \tag{29}$$

$$z_i^d \geq z_{ii}^d + l_i \quad \forall i \in V^d \tag{30}$$

and Constraints (20) to (22).

The sub-problem’s objective function is an upper bound (UB) for the originally proposed problem. Constraints (25) to (30) are equivalent to Constraints (11) to (16) in the initial problem, whose complicated variables are assumed to be constant.

### 3.2.2 Master problem

The master problem in the Benders decomposition algorithm leads to the generation of a lower bound (LB) for the original problem, which is defined as follows:

$$(MP) \text{ Min}Z_{MP} = \psi \tag{31}$$

S.t.:

$$(1 - x_{ij}) \cdot R_j \geq R_i + 1 - |V^r \cup \{0\}| \quad \forall i \in V^r; j \in V^r \cup \{0\} \tag{32}$$

$$\psi \geq \psi_{LB} \tag{33}$$

$$\psi \in R \tag{34}$$

$$u_{ijk}^1 \cdot u_{ijk}^2 \cdot u_{ijk}^3 \cdot u_{ijk}^4 \cdot u_{ijk}^5 \cdot u_{ijk}^6 \in R \tag{35}$$

and Constraints (2) to (10), (17) to (19).

The objective function in the originally proposed model includes only continuous variables in the sub-problem but cannot be included in the master problem. Therefore, the objective function of the master problem, i.e., relation (31), only consists of a free variable in the  $\psi$  notation, which denotes the algorithm’s lower bound.

It should be noted that the sub-tour elimination constraints in the original model will not remain in the master problem. Therefore, as stated by Moreno et al. (2019) the Constraint (32) is added as a sub-tour elimination constraint together with the auxiliary variables  $R_j$  to the master problem. Constraints (33) and (35) determine the domain of those variables.

### 3.2.3 Feasibility and optimality cuts

The Benders optimality cut moves towards the optimal solution, added to the master problem. The guideline for adding the Benders cut in each iteration is defined as follows: if an obtained solution is not applied to the stop condition, an optimality cut is added to the master problem. Otherwise, if the solution is infeasible for the sub-problem, a feasibility cut (Constraint 36) must be added to the master problem, as described in the following:

$$\begin{aligned} \psi \geq & \sum_{k \in V^d} \sum_{i,j \in V} (t_{ij} - M(1 - w_{ij}^k)) \cdot u_{ijk}^1 + \sum_{k \in V^d} \sum_{j \in V} \sum_{i \in V^r} (t_{ij} + s_i - M(1 - w_{ij}^k)) \cdot u_{ijk}^2 \\ & + \sum_{k \in V^d} \sum_{j \in V} \sum_{i \in V^r} (t_{ij} + s_i - M(1 - w_{ij}^k)) \cdot u_{ijk}^3 \\ & + \sum_{i,j \in V^r} \left( \sum_{e,l \in V} U_{ijel} \cdot t_{el} + s_i - M(1 - x_{ij}) \right) \cdot u_{ij}^4 + \sum_{i,j \in V^r} \left( \sum_{e,l \in V} U_{ijel} \cdot t_{el} + s_i - M(1 - x_{ij}) \right) \cdot u_{ij}^4 \\ & + \sum_{i,j,r \in V^r} (M(N_{kij} - 1)) \cdot u_{ijr}^5 + \sum_{i \in V^d} (t_i) \cdot u_i^6 \end{aligned} \tag{36}$$

In Constraint (36), the right side comprises the sum of values of the right side of each sub-problem constraint multiplied by their dual variables. This value must be less than the auxiliary variable  $\psi$  to improve the objective function in any iteration. The variable  $u$  also denotes the dual variables of each constraint.

Constraint (37) is also a feasibility cut of the Benders decomposition algorithm, preventing the original problem from becoming unbounded.

$$\begin{aligned}
 & \sum_{k \in V^d} \sum_{i, j \in V} \left( t_{ij} - M(1 - w_{ij}^k) \right) \cdot u_{ijk}^1 + \sum_{k \in V^d} \sum_{j \in V} \sum_{i \in V^r} \left( t_{ij} + s_i - M(1 - w_{ij}^k) \right) \cdot u_{ijk}^2 \\
 & + \sum_{k \in V^d} \sum_{j \in V} \sum_{i \in V^r} \left( t_{ij} + s_i - M(1 - w_{ij}^k) \right) \cdot u_{ijk}^3 \\
 & + \sum_{i, j \in V^r} \left( \sum_{e, l \in V} U_{ijel} \cdot t_{el} + s_i - M(1 - x_{ij}) \right) \cdot u_{ij}^4 \\
 & + \sum_{i, j, r \in V^r} (M(N_{kij} - 1)) \cdot u_{ijr}^5 + \sum_{i \in V^d} (l_i) \cdot u_i^6 \leq 0
 \end{aligned} \tag{37}$$

### 4 Computational experiments

In this section, the performance of the proposed solution approach is assessed based on different generated test problems.

To compare the performance of the recommendation algorithm, an ACO algorithm also developed based on the research of Shin et al. (2019). In the proposed ACO algorithm, two types of ants, are operated in the considered network (Shin et al., 2019). The repair ant and relief ant represent the ants corresponding to the repair crew and the relief vehicle, respectively. Each ant tries to determine a feasible route from the perspective of repair and transfer, respectively. Because the route of the repair ant affects the route of the relief ant, the route of the repair ant is determined first. Constraint (23) is employed to determine the arrival time of relief vehicle to demand node by term  $\max(z_{ik}^d, z_i^r)$ . For details, refer to Shin et al. (2019) and Thiruvady et al. (2016).

First, in small-sized test problems, the results of the proposed hybrid approach are compared with those solved using the CPLEX solver in GAMS. Then, since solving accurate large-scale test problems is not possible in CPLEX in acceptable CPU time, the efficiency of the hybrid algorithm to solve large-scale problems is considered. It is noteworthy that the heuristic algorithm is coded using C# programming language, the Benders decomposition algorithm is coded in GAMS version 24.7.3, and the ACO algorithm is coded by MATLAB software on a PC with 8 GB of RAM, a CPU of Intel (R) Core (TM) i7-7500U. 25 test problems in different sizes are generated to evaluate the solution methods (Table 7).

The number of nodes and edges of the network, the number of demand nodes, and damaged nodes in each test problem are given in Table 7. A  $(VN + 1, VN + 1)$  matrix is used to produce connecting edges in which VN demonstrates the number of nodes. The elements of this matrix are generated randomly as the binary numbers (0 or 1), where each 1 for element  $ij$  in the matrix indicates the presence of an edges between nodes ( $i$  to  $j$ ). The total number of 1 for each node is also randomly generated in such a way as to ensure the feasibility of the problem.

Also, the values of the model parameters are generated randomly by using a uniform function (see Table 8). The number of main parameters of developed ACO are:  $\alpha = 0.3$ ,  $\beta = 1$ , and  $\rho = 0.01$ .

**Table 7** Size of test problems

Problem size	No.	Number of nodes	Number of edges	Number of $V^d$ nodes	Number of $V^r$ nodes
Small	1	5	5	2	1
	2	7	7	3	3
	3	8	10	4	3
	4	9	13	4	4
	5	9	12	4	4
	6	10	14	4	4
	7	10	15	4	5
	8	11	13	5	4
	9	13	16	6	6
	10	15	20	7	7
Medium	11	20	25	9	9
	12	23	28	10	9
	13	25	30	11	10
	14	27	34	13	11
	15	30	35	14	12
	16	40	45	15	15
	17	45	50	20	15
	18	50	60	20	20
	19	55	70	25	20
	20	60	75	27	25
Large	21	65	80	27	30
	22	70	80	30	30
	23	75	90	35	30
	24	80	100	40	35
	25	90	120	40	40

**Table 8** Parameter design

Parameters	Description
$d_i$	Uniformint (1,10)
$s_i$	Uniformint (1,5)
$t_{ij}$	Uniformint (1,30)
$l_i$	Uniformint (1,3)

#### 4.1 Performance evaluation of the proposed hybrid solution approach

For evaluating the performance of the proposed hybrid approach, first, 10 small-size test problems presented in Table 7 are solved, and the results are compared with those obtained

**Table 9** Comparison of the results obtained by proposed approaches and CPLEX

No.	Objective function				$GAP_{Sol}^*$ (%)		
	CPLEX	Heuristic algorithm	Hybrid approach	ACO	Heuristic algorithm	Hybrid approach	ACO
1	74	75.07	74.00	74.01	1.45	0.00	0.01
2	109	112.82	110.70	109.16	3.50	1.56	0.15
3	228	236.96	230.67	230.44	3.93	1.17	1.07
4	946	961.51	961.51	973.15	1.64	1.64	2.87
5	1078	1079.19	1078.00	1092.45	0.11	0.00	1.34
6	1386	1656.74	1386.00	1396.78	4.46	0.00	0.09
7	1128	1174.47	1149.54	1138.04	4.12	1.91	0.89
8	1280	1346.82	1280.00	1307.65	5.22	0.00	2.16
9	2370	2456.27	2406.02	2420.96	3.64	1.52	2.15
10	2462	2638.77	2506.32	2505.82	7.18	1.80	1.78

from CPLEX. It should be noted that the performance of the heuristic algorithm without hybridizing it with the Benders decomposition algorithm is assessed, as well.

It is noteworthy that in the two proposed algorithms, the stop condition is to reach a CPU time of 150 s. The CPU time taken to solve all 10 test problems by CPLEX was less than 10 s. As indicated in Table 9, the CPU time in these cases (i.e., the best UB obtained by these three algorithms at this stopping CPU time) is reported as the values of the objective function. The  $GAP_{Sol}^*$  calculated using Eq. (38), denotes the gap between these solution methods, where  $Sol$  is the objective function of each algorithm. The  $Sol_{Optimal}$  is the value of the objective function obtained by CPLEX.

$$GAP_{Sol}^* \% = \left( \frac{Sol - Sol_{Optimal}}{Sol_{Optimal}} \right) * 100 \tag{38}$$

As shown in Table 9, the CPU time taken to solve the small-size test problems, provided in each solution method, are short. As the dimension of the problem increases, solving the mathematical model in CPLEX becomes time-consuming. Due to the obtained results and acceptable CPU time of the proposed heuristic algorithm, it is reasonable to use this algorithm to generate an initial solution for the Benders decomposition algorithm.

As shown in Table 9 and Fig. 8, in small-size test problems (i.e., No. 1 to 10), all the solutions provided by the heuristic algorithm and ACO are feasible and have less than a 7.2% and 3% gap from the optimal solution, respectively. In these cases, the hybrid approach produces more efficient solutions in less than 2% gap from the optimal solution. These results, in addition to the validity of the hybrid algorithm, also explain the more efficient performance of the hybrid method compared to the others. Figure 9 shows convergence diagrams of the proposed hybrid approach in Test problem 6, which converges to the optimal solution in iteration 114.

The  $GAP_{hyb}$  reported in Table 10 is calculated by Eq. (39) to show the performance of the hybrid algorithm in large-size problems. It should be noted that due to the inability of CPLEX to solve large-size problems efficiently, upper bound (UB) and lower bound (LB) are calculated for the test problems. The UB and LB are obtained by the hybrid algorithm in



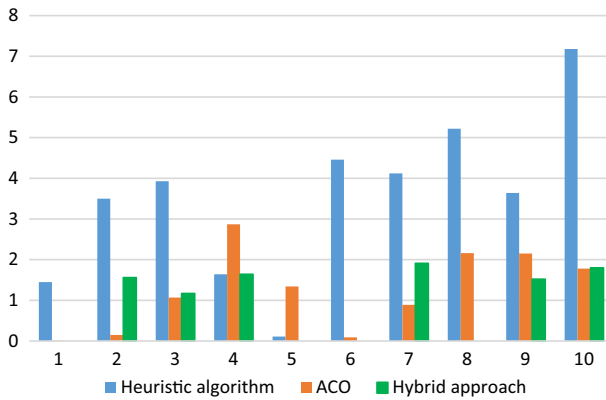


Fig. 8 Comparison of the generated gaps by solution approaches

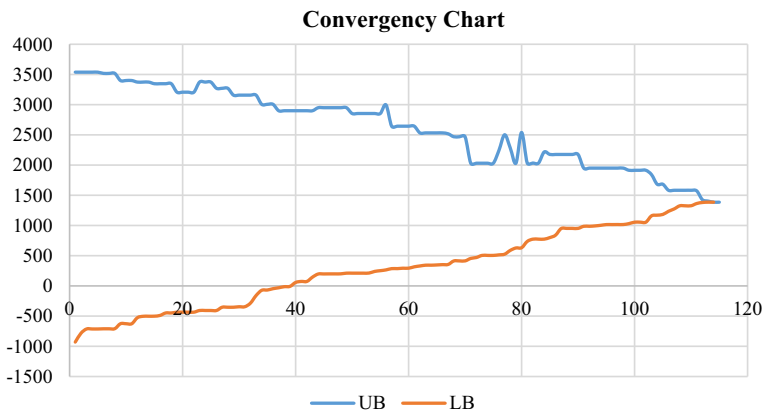


Fig. 9 Progression of UB and LB values for the proposed hybrid approach (Test problem 6)

3600 s (as stop condition). The results are provided in Table 10.

$$GAP_{hyb}\% = \left( \frac{UB - LB}{UB} \right) * 100 \tag{39}$$

The larger the dimensions of the studied test problems, the better the efficiency of the proposed hybrid approach. In medium and large-scale test problems (i.e., Test problems No.11 to 25), the solution obtained using CPLEX is highly inefficient and produces a solution with  $GAP$  above 50% for medium-size test problems up to 3600 s CPU time. However, as shown in Table 10, in the proposed hybrid approach, in large-size test problems, up to 3600 s, the solution offers a  $GAP_{hyb}$  of less than 5%.

As demonstrated in Table 10, in medium dimensions (i.e., Problems 11 to 20) the hybrid method reaches the optimal solution in 3600 s. However, the UB of the ACO shows that the optimal solution is not achieved and the maximum gap is 1.5%. In large dimensions, the convergence speed of the hybrid method to the optimal solution decreases and the gap value increases when the algorithm stops at 3600 s. In Problems 23 and 25, the UB of the ACO

**Table 10** Efficiency of the proposed hybrid algorithm compared to ACO in medium- and large-size test problems

No.	Hybrid algorithm			ACO
	<i>UB</i>	<i>LB</i>	<i>GAP<sub>hyb</sub>(%)</i>	UB
11	3561	3561	0.00	3640
12	3987	3987	0.00	4024
13	4563	4563	0.00	4621
14	4289	4289	0.00	4289
15	5738	5738	0.00	5738
16	7987	7987	0.00	8014
17	7856	7856	0.00	7856
18	7309	7309	0.00	7413
19	8892	8892	0.00	8913
20	6573	6552	0.32	6651
21	8236	8236	0.00	8377
22	11,385	11,218	1.47	11,914
23	13,628	13,343	2.09	12,971
24	18,945	18,049	4.73	19,106
25	23,671	22,549	4.74	22,535

is lower and therefore it performs better. This result emphasizes that the performance of the hybrid algorithm is superior in large dimensions without time limitation.

In summary, the comparison between the hybrid approach and CPLEX in terms of the CPU time demonstrates that as the size of the test problem increases, the performance of the two approaches differs significantly in terms of CPU time. Accordingly, the CPU time of CPLEX increases exponentially as the size of the test problem increases, making the method inefficient in practice. While the proposed hybrid approach reaches an efficient solution in less time to manage disaster-related activities.

## 5 Concluding remarks

As noted in the literature review, few studies have covered repairing the damaged roads and providing relief to the affected areas simultaneously in a single model. In addition to calculating the required accessibility time of each demand node, this research determines the routing of every relief vehicle from the depot to each demand node in a prioritized manner. To achieve more realistic scenario, the blocking caused by damaged nodes along the route was scored and considered, and completion time of the relief operation to each demand node was determined. Compared to the literature, the mathematical model presented in this research offers the desired output with fewer variables and constraints, making it less complex. A hybrid approach was presented to solve the proposed model which attains the near-optimal solution in less time compared to an exact solver (i.e., CPLEX). The proposed approach is based on the Benders decomposition method and a heuristic algorithm. The initial solution for the Benders decomposition method and repair schedule of the damaged nodes is determined

using the heuristic algorithm. Then, the routing of the repair crew and relief vehicles is solved using the Benders decomposition method.

In order to obtain a more realistic scenario, in future works, more than one repair crew can be included in the model to work on the damaged and blocked roads simultaneously, or multi-mode scheduling can be employed. Setting multiple crews to clear roads from debris and coordinating between them will lead to accelerate unblocking of each road. However, such cooperation prevents simultaneous clearance of different blocked roads due to the lack of uniform distribution of crew across all demand points. This aspect has been addressed in problems focusing solely on roads' debris clearance, while problems with simultaneous debris clearing and post-disaster relief have suffered further research. Other suggested topics for the future research are as follows:

- Presenting a model considering the priority of roads where the priority of different roads is weighted regarding the type and location of debris based on data acquired from a case study prior to implementing the model.
- Considering uncertainty in both the supply and demand capacity in a post-disaster phase.
- Measuring a minimum and optimal number of resources required for road repair and debris clearance operation within a specified time.
- Considering a time window to aid each demand node.
- Considering traffic in routes of the relief vehicles and repair crews.

## References

- Akbari, V., & Salman, F. S. (2017). Multi-vehicle synchronized arc routing problem to restore post-disaster network connectivity. *European Journal of Operational Research*, 257(2), 625–640.
- Aksu, D. T., & Ozdamar, L. (2014). A mathematical model for post-disaster road restoration: Enabling accessibility and evacuation. *Transportation Research Part e: Logistics and Transportation Review*, 61, 56–67.
- Alinaghian, M., Aghaie, M., & Sabbagh, M. S. (2019). A mathematical model for location of temporary relief centers and dynamic routing of aerial rescue vehicles. *Computers & Industrial Engineering*, 131, 227–241.
- Altay, N., & Green, W. G., III. (2006). OR/MS research in disaster operations management. *European Journal of Operational Research*, 175(1), 475–493.
- Avci, M., & Avci, M. G. (2017). A GRASP with iterated local search for the traveling repairman problem with profits. *Computers & Industrial Engineering*, 113, 323–332.
- Benders, J. (1962). Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4(1), 238–252.
- Briskorn, D., Kimms, A., & Olschok, D. (2020). Simultaneous planning for disaster road clearance and distribution of relief goods: a basic model and an exact solution method. *OR Spectrum: Quantitative Approaches in Management*, pp.1–29.
- Çelik, M., Ergun, Ö., & Keskinocak, P. (2015). The post-disaster debris clearance problem under incomplete information. *Operations Research*, 63(1), 65–85.
- Costa, A. M. (2005). A survey on benders decomposition applied to fixed-charge network design problems. *Computers & Operations Research*, 32(6), 1429–1450.
- Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1), 269–271.
- Duque, P. A. M., Coene, S., Goos, P., Sörensen, K., & Spieksma, F. (2013). The accessibility arc upgrading problem. *European Journal of Operational Research*, 224(3), 458–465.
- Habib, M. S., & Sarkar, B. (2017). An integrated location-allocation model for temporary disaster debris management under an uncertain environment. *Sustainability*, 9(5), 716.
- Kasaei, M., & Salman, F. S. (2016). Arc routing problems to restore connectivity of a road network. *Transportation Research Part e: Logistics and Transportation Review*, 95, 177–206.
- Kim, S., Park, Y., Lee, K., & Moon, I. (2017). Repair crew scheduling considering variable disaster aspects. In IFIP International Conference on Advances in Production Management Systems (pp. 57–63). Springer, Cham.

- Li, S., Ma, Z., & Teo, K.L. (2020). A new model for road network repair after natural disasters: integrating logistics support scheduling with repair crew scheduling and routing activities. *Computers & Industrial Engineering*, p.106506.
- Li, M. Y., Zhao, X. J., Fan, Z. P., Cao, P. P., & Qu, X. N. (2019). A model for assignment of rescuers considering multiple disaster areas. *International Journal of Disaster Risk Reduction*, 38, 101201.
- Lorca, Á., Çelik, M., Ergun, Ö., & Keskinocak, P. (2015). A decision-support tool for post-disaster debris operations. *Procedia Eng*, 107, 154–167.
- Lorca, Á., Çelik, M., Ergun, Ö., & Keskinocak, P. (2017). An optimization-based decision-support tool for post-disaster debris operations. *Production and Operations Management*, 26(6), 1076–1091.
- Martins de Sá, E. M., De Camargo, R. S., & De Miranda, G. (2013). An improved Benders decomposition algorithm for the tree of hubs location problem. *European Journal of Operational Research*, 226(2), 185–202.
- Miller, C. E., Tucker, A. W., & Zemlin, R. A. (1960). Integer programming formulation of traveling salesman problems. *Journal of the ACM (JACM)*, 7(4), 326–329.
- Moreno, A., Munari, P., & Alem, D. (2019). A branch-and-Benders-cut algorithm for the crew scheduling and routing problem in road restoration. *European Journal of Operational Research*, 275(1), 16–34.
- Moreno, A., Munari, P., & Alem, D. (2020). Decomposition-based algorithms for the crew scheduling and routing problem in road restoration. *Computers & Operations Research*. <https://doi.org/10.1016/j.cor.2020.104935>
- Özdamar, L., Aksu, D. T., & Ergüneş, B. (2014). Coordinating debris cleanup operations in post disaster road networks. *Socio-Economic Planning Sciences*, 48(4), 249–262.
- Pramudita, A., & Taniguchi, E. (2014). Model of debris collection operation after disasters and its application in urban area. *International Journal of Urban Sciences*, 18(2), 218–243.
- Şahin, H. (2013). Debris removal during disaster response phase: a case for Turkey (Doctoral dissertation, Bilkent University).
- Shin, Y., Kim, S., & Moon, I. (2019). Integrated optimal scheduling of repair crew and relief vehicle after disaster. *Computers & Operations Research*, 105, 237–247.
- Taşkın, Z.C. (2010). Benders decomposition. *Wiley Encyclopedia of Operations Research and Management Science*.
- Thiruvady, D., Ernst, A. T., & Singh, G. (2016). Parallel ant colony optimization for resource constrained job scheduling. *Annals of Operations Research*, 242, 355–372.
- Thomas, A. S., & Kopczak, L. R. (2005). From logistics to supply chain management: The path forward in the humanitarian sector. *Fritz Institute*, 15(1), 1–15.
- Ulusan, A., & Ergun, Ö. (2021). Approximate dynamic programming for network recovery problems with stochastic demand. *Transportation Research Part e: Logistics and Transportation Review*, 151, 102358.
- Wang, S., Liu, F., Lian, L., Hong, Y., & Chen, H. (2018). Integrated post-disaster medical assistance team scheduling and relief supply distribution. *The International Journal of Logistics Management*.
- Yan, S., & Shih, Y. L. (2009). Optimal scheduling of emergency roadway repair and subsequent relief distribution. *Computers & Operations Research*, 36(6), 2049–2065.
- Yan, S., & Shih, Y. L. (2012). An ant colony system-based hybrid algorithm for an emergency roadway repair time-space network flow problem. *Transportmetrica*, 8(5), 361–386.

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

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.