



# Work crew routing problem for infrastructure network restoration

Nazanin Morshedlou, Andrés D. González, Kash Barker\*

School of Industrial and Systems Engineering, University of Oklahoma, Norman, OK, USA

## ARTICLE INFO

### Article history:

Received 16 April 2018

Revised 3 September 2018

Accepted 4 October 2018

### Keywords:

Restorative capacity planning

Routing problem

Infrastructure networks

Heuristic

Network resilience

## ABSTRACT

This paper introduces a synchronized routing problem for planning and scheduling restorative efforts for infrastructure networks in the aftermath of a disruptive event. In this problem, a set of restoration crews are dispatched from depots to a road network to restore the disrupted infrastructure network. Two mathematical formulations are presented to scheduling and sequencing disrupted network components to restoration crews and route the crews towards disrupted components to maximize network resilience progress in any given time horizon. In the first formulation, the number of restoration crews assigned to each disrupted component, the arrival time of each assigned crew to each disrupted component and consequently the restoration rate associated with each disrupted component are considered as variables to increase the flexibility of the model in the presence of different disruptive events. Along with the contributions applies in the first formulation, in the second formulation, each disrupted component can be partially active during its restoration process. To find the coordinated routes, we propose a relaxed mixed integer program as well as a set of valid inequalities which relates the planning and scheduling efforts to decision makers policies. The integration of the relaxed formulation and valid inequalities results in a lower bound for the original formulations. We further introduce a feasibility algorithm to derive a strong initial solution for the routing restorative capacity problem. Computational results on gas, water, and electric power infrastructure network instances from Shelby County, TN data, demonstrates both the effectiveness of the proposed model formulation, in solving small to medium scale problems, the strength of the initial solution procedure, especially for large scale problems.

© 2018 Elsevier Ltd. All rights reserved.

## 1. Introduction and motivation

Critical infrastructure networks are referred to as physical and virtual systems and assets that provide services which form the basis of society, and as such they are relied upon as “the backbone of the nation’s economy, security, and health” (White House, 2013). In particular, electric power, gas, and water are considered key critical infrastructure networks as they enable functionality and productivity across all other critical infrastructures (White House, 2013). According to Presidential Policy Directive (PPD) 21 (White House, 2013), such networks “must be secure and able to withstand and rapidly recover from all hazards.” In other words, these critical infrastructure networks must be *resilient*.

\* Corresponding author.

E-mail address: [kashbarker@ou.edu](mailto:kashbarker@ou.edu) (K. Barker).

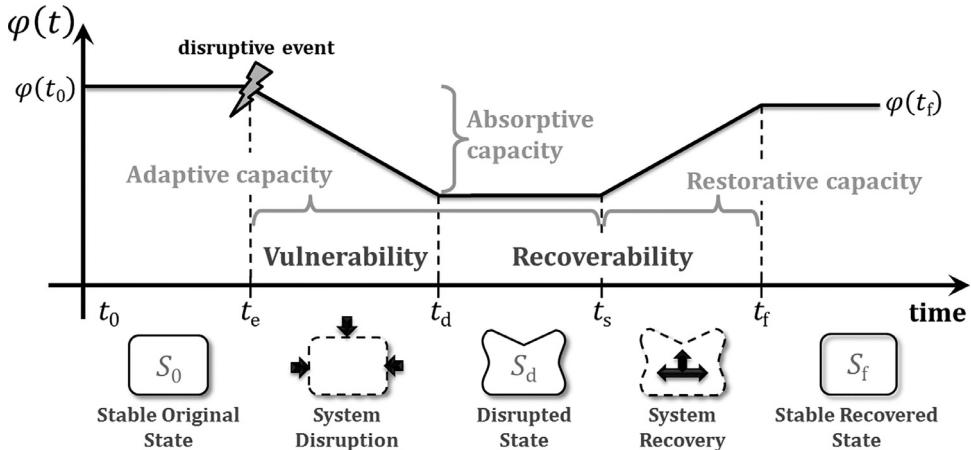


Fig. 1. Trajectory of performance across system states, adapted from [Henry and Ramirez-Marquez \(2012\)](#).

Despite the progress toward enhancing the resilience of infrastructure networks, “aging equipment, capacity bottlenecks, as well as increasing storms and climate change” increase the vulnerability of infrastructure networks in the face of terrorist attacks, natural disasters, and manmade hazards ([American Society for Civil Engineers \(ASCE\), 2017](#)). According to the [ASCE Infrastructure Report \(2017\)](#), the US energy sector was recently given a grade of D+, and the water and wastewater sector was given a grade of D, citing that (i) service was at its full capacity in 640,000 miles of the high voltage transmission lines in the lower 48 states, (ii) many of the drinking water and wastewater pipelines and gas distribution lines should be replaced as they are approaching the end of their life expectancy, and (iii) without incorporating the negative effect of disruptions, there are still an estimated 240,000 water major breaks per year.

The increasing vulnerability in these critical infrastructure networks structure, combined with the more frequent and severe natural disasters and malevolent attacks, challenge our traditional planning for responding to and restoring from disruptive events. For example, when Hurricane Harvey struck the southern coast, it caused about \$200 billion in damages, and \$20–\$30 billion in lost economic output ([CNBC, 2017](#)). According to [FEMA \(2017\)](#), nearly 40,000 people were in shelters in Texas and Louisiana, most without essential lifeline services. Over 160 drinking water systems were damaged, with 50 of them totally shut down, and 800 water waste facilities were partially damaged. Also, more than 300,000 customers were without power for more than 24 h ([Commission to Rebuild Texas, 2017](#)). Just from Hurricane Harvey and its consequences, we realize the extent to which a severe disruption to critical infrastructure networks can adversely impact the health, security, and the productivity of a society.

In response to this, several studies have defined, modeled, and assessed resilience across different critical infrastructure sectors ([Hosseini et al., 2016](#); [Celik 2017](#)). Many studies focus on reducing vulnerability or enhancing restoration in energy sectors ([Bienstock and Mattia, 2007](#); [Nurre et al., 2012](#); [Nan and Sansavini, 2017](#)), water and wastewater sectors ([Nurre et al., 2012](#); [D'Ambrosio et al., 2015](#)), transportation and emergency response ([Baroud et al., 2014](#); [Celik et al., 2015](#); [Iloglu and Albert, 2018](#)), and the interconnectivity and associated interdependency among various types of networks ([Sharkey et al., 2015](#); [González et al., 2016b](#); [Barker et al., 2017](#); [González et al., 2017](#); [Smith et al., 2017](#)).

To illustrate the effects of a disruptive event, [Fig. 1](#) depicts the changes in the performance of an infrastructure network prior, during, and after its occurrence ([Henry and Ramirez-Marquez, 2012](#); [Barker et al., 2013](#); [Pant et al., 2014](#)). The performance (e.g., the total demand that is satisfied at time  $t$ ) and the resilience (e.g., the proportion between the performance at time  $t$  and the performance before the disruptive event  $t_e$ ) are denoted by  $\varphi(t)$  and  $\varphi(t|t_e)$ , respectively.

[Vugrin and Camphouse \(2011\)](#) defined *resilience capacity* as a function of (i) *absorptive capacity*, or the extent to which an infrastructure network absorbs the negative effect of a disruptive event, (ii) *adaptive capacity*, or the extent to which an infrastructure network adapts to new conditions in the aftermath of a disruptive event by temporary means, and (iii) *restorative capacity*, or the extent to which an infrastructure network is recovered in long-term manner. The collection of absorptive and adaptive capacities addresses the vulnerability mitigation of the network, or to what extent an infrastructure network withstands a disruptive event (i.e.,  $\varphi(t_e) - \varphi(t_d)$ ). The restorative capacity is analogous to recoverability, or the ability of the network to recover to a desire level of performance in a timely manner. As [Fig. 1](#) indicates, the dimensions of vulnerability and recoverability combine to account for resilience. Among several studies, optimization approaches to increase powerlines capacities to prevent large scale cascading blackouts in power network ([Bienstock and Mattia, 2007](#)) is an example of absorptive capacity. The robust adaptive strategies to respond to the dramatic climate change in water management systems ([Lempert and Groves, 2010](#)), where simulation models of several disruption scenarios to the network are implemented to illustrate the vulnerabilities and assess the options to ameliorate those vulnerabilities is an example of adaptive capacity. Finally, debris removal from a transportation network after a natural disaster ([Celik et al., 2015](#)) is an example of restorative capacity.

In this paper, we focus on enhancing restorative capacity of infrastructure networks after a large disruptive event. The proposed formulations and techniques in this paper can be applied to the restoration efforts of a variety of infrastructure systems. While other works have proposed optimization formulations to assign resources or schedule work crews for interdependent network restoration (González et al., 2016a; Almogathawi et al., 2017; Sharkey et al., 2015), proposed here is a formulation that integrates the work crew scheduling problem with a vehicle routing problem to address the practical problem of traversing a given road network to recover other infrastructure networks. The main contribution of this research is to propose two mixed integer linear routing models that assign a set of disrupted components to each restoration crew and identify the route with the minimum total traveling time associated with that restoration crew. In the first routing model, referred to as the *Binary Active Restoration Crew Routing* model, each disrupted component is not operational unless it is fully recovered. In the second routing model, referred to as *Proportional Active Restoration Crew Routing*, each disrupted component can be partially operational in the network while it is being recovered. Disrupted components have component-specific characteristics, including specific restoration rates and disruption levels. After a disruptive event, various restoration crews can be assigned to a disrupted component and accelerate its restoration trajectory. Each of the assigned restoration crews can arrive at a time that does not depend on the arrival time of other assigned crews. However, a restoration crew cannot arrive at a disrupted component after its restoration process is completed by other crews. We also note that each disrupted component may experience a different increase in the restoration rate when a new crew joins to the restoration process. The optimal assignment, schedule, and route of restoration crews can significantly reduce the restoration time of the entire set of infrastructure networks. This paper assumes that any disruption affects only the infrastructure network, not the transportation network. As the formulation proposed here is already substantially complex, we consider this paper to be a first step toward eventually analyzing simultaneous disruptions to infrastructure and transportation networks.

The remainder of the paper is organized as follows. Section 2 introduces an overview of the infrastructure resilience literature focusing on restorative capacity and restoration crew routing problems. Section 3 proposes a multiple restoration crew routing formulation to distribute restoration crews to disrupted network components through a routing network and to update the model to incorporate the proportionally operational components in each time period. We then propose a lower bound for the restoration crew routing problem by introducing a relaxed formulation of the original model and a heuristic algorithm to provide a feasible initial solution aligned with policies for enhancing infrastructure network resilience. Section 4 illustrates the applicability of the proposed formulations with the system of gas, water, and electric power networks derived from those in Shelby County, Tennessee. We also discuss the computational results associated with the illustrative examples and investigate the efficacy of the proposed algorithm to find the best lower bound. Concluding remarks and prospective future work are provided in Section 5.

## 2. Background literature

Several studies in recent years have focused on optimization models and algorithms to improve the restoration process after disruptive events. Celik (2017) provides a comprehensive overview of the literature on large-scale infrastructure network restoration in the aftermath of catastrophes and malevolent attacks.

Many of the fundamental studies in the field of post disruption infrastructure network resilience do not address the issue of routing, instead focusing on scheduling and sequencing disrupted network components to restoration crews. Nurre et al. (2012) introduce a design and scheduling formulation to improve the infrastructure network construction and restoration process. Aligned with particular decision making policies, the authors develop a dispatching rule based heuristic to identify the next set of network components to be restored by crews. Liberatore et al. (2014) present a restoration planning formulation for disrupted transportation networks through which emergency goods are distributed to affected populations, noting that the routing problem is not considered in their proposed formulation. Sharkey et al. (2015) propose a mathematical formulation that incorporates the restoration interdependencies among different infrastructure networks (e.g., water, power, transportation) into the design and scheduling problems. They also investigate the effects of centralized decision making (i.e., where one decision maker dispatches all recovery resources through all infrastructure networks) and decentralized decision making (i.e., where decision makers associated with each infrastructure determine restoration efforts independently and communicate with other decision makers responsible for other infrastructure networks). Özdamar and Ertem (2015) study a variety of humanitarian operations, including relief delivery, casualty transportation, and mass evacuation after large-scale disruptions. While the aforementioned works deal with various aspects of service networks engaged after a disruption, none consider the routes of restoration crews, and none account for the routing time of each crew and its effect on the restoration plan and schedule. In the field of supply chain networks, Wang et al. (2016) focus on the interdependencies among several supply chain networks and their environment and how their resilience structure prevents financial crises and economic recession. González et al. (2016a, b) propose a mathematical model to recover a damaged system of interdependent networks while considering limited resources and diverse operational constraints. Their model considers not only physical interdependencies among the different networks in the system, but also cost reductions associated with recovering multiple co-located components simultaneously. Furthermore, considering the high computational complexity associated with optimizing the recovery of a system of interdependent networks, González et al. (2017) propose a reduced-order linear representation based on data-driven system identification, denominated the recovery operator, which reproduces the main recovery dynamics of the system and can be used to generate efficient recovery strategies. Extending from the approach by Sharkey et al. (2015), Smith et al. (2017) propose a sequential game theoretic model to determine effi-

cient recovery strategies that depict decentralized decision making processes with partial information under a time-discrete non-cooperative configuration. Chapman et al. (2017) show that such a recovery operator can be used to efficiently model decentralized decisions, by constructing a layered Cartesian form of the studied system. Ouyang and Fang (2017) establish a decision making formulation to protect and restore critical infrastructure networks after malevolent attacks. Their proposed decomposition algorithm minimizes network vulnerability by fortifying network components and/or building new supporting lines prior to a disruption and enhances the network restoration process after the disruption.

With regard to routing time of restoration crews as well as scheduling and sequencing disrupted network components to each of those crews, we found several works that specifically focus on the road network restoration process itself (e.g., debris cleaning and disposal, snow removal). Previous works may differ from the restoration of other infrastructure networks as the disruptions in transportation networks result in the loss of physical connections. As a result, the accessibility to some disrupted network components depends on the operational state of other components. Considering the restoration of disrupted infrastructure networks, Wang et al. (2010) provide a multi-objective optimization model to apply contraflow techniques and plan optimal emergency resources to repair roads in a timely manner. Hu and Sheu (2013) present a reverse logistics approach for post-disruption debris that proposes a multi-objective linear model to minimize the total reverse logistics cost, environmental and operational costs, and required medical treatment in the affected location. Faturechi and Miller-Hooks (2014) propose a bi-level three-stage mathematical formulation to maximize the connectivity of roadway networks and optimize traveling time through those networks after a disruptive event. Celik et al. (2015) develop a partially observable Markov decision model to solve a stochastic debris removal problem to determine the optimal schedule of blocked links over discrete time periods. To reconnect a disrupted transportation network in the minimum time horizon, Kasaei and Salman (2016) propose an arc routing formulation that identifies the restoration schedule and sequence of blocked roads. For large-scale routing problems, they develop a heuristic algorithm to maximize the benefit gained by network connectivity promptly. Iloglou and Albert (2018) propose a p-median formulation to find the minimized weighted distance between the emergency responders and disrupted locations in a transportation network. Wang et al. (2017) study the urgent evacuation problem through transportation networks, proposing an algorithm to connect the transportation network to minimize the traveling time between each pair of locations. Akbari and Salman (2017) extend the arc routing formulation to dispatch more than one restoration crew through the disrupted network such that a closed road cannot be traversed unless its restoration procedure is completed. They then propose a local search algorithm to find a set of synchronized routes resulting in minimum required time to reach to the complete network connectivity. Xu et al. (2018) propose a data-driven approach to manage multiple types of emergency fleets and successfully strike a balance between dynamic rescue demands and restoration vehicles supplies in post disruption urban flood control to dispatch restoration vehicles through disrupted areas using real time disruption scenarios.

We integrate the two areas of literature described in the previous two paragraphs, (i) infrastructure restoration and (ii) transportation network dispatch, to propose a new problem that addresses the dependent relationship between a disrupted infrastructure network and the routing network that connects all disrupted components. The proposed model is an extended form of the multiple restoration crew routing problem and the fundamental constraints employed in such a model (e.g., the routing network, subgraph elimination, crew arrival time consideration).

### 3. Problem formulation

Mentioned in Sections 1 and 2, the purpose of this paper is to establish the optimal restoration plan for a disrupted infrastructure network by determining, among others, the best schedule and sequence of disrupted components assigned to each crew through an underlying routing network. The infrastructure network is represented by an undirected connected graph  $G = (N, A)$ , where  $N$  is the set of nodes and  $A$  is the set of links. There is a set of supply nodes  $N_+ \subseteq N$ , where each supply node  $i \in N_+$  supplies amount  $o_i$  in each time period, a set of demand nodes  $N_- \subseteq N$ , where each demand node  $i \in N_-$  demands amount  $b_i$  in each time period, and a set of transition nodes  $N_0 \subseteq N$ . There is also a set of links  $A' \subseteq A$ , that are affected by a disruptive event. Each link  $(i, j) \in A$  has a pre-defined capacity  $u_{ij}$  and a pre-disruption flow value  $f_{ij}$  calculated based on the total amount of demand. In some cases, some demand nodes need to be prioritized over others as they might be located in more critical areas (e.g., more populated, located near hospitals or other critical facilities). To incorporate the relative importance of each demand node  $i \in N_-$ , we define weight  $w_i$  to give priority such nodes. The principal goal of our formulation is to send maximum flow from supply nodes to demand nodes, while respecting the flow capacity of links and supply/demand capacities.

Separate from the *infrastructure* network, we model the *routing* network as a complete undirected graph  $\tilde{G} = (\tilde{N}, \tilde{A})$ , where  $\tilde{N}$  is the set of nodes and  $\tilde{A}$  is the set links defined between each pair of nodes. Relating the infrastructure network to the routing network, the disrupted locations on disrupted links  $A' \subseteq A$  in the infrastructure network, are the set of nodes  $\tilde{N}_{A'} \subseteq \tilde{N}$  forming the routing network  $\tilde{G}$ . There is a set of depots  $\tilde{N}_D \subseteq \tilde{N}$  from which the restoration crews are dispatched and a dummy sink node  $n+1 \in \tilde{N}$  to which all the restoration routes end. For directed routing networks, we simply assign  $x_{ijk} = 0$ , for  $k \in K$ ,  $\forall i, j \in \tilde{N}$ , if there is no path from node  $i$  to node  $j$ , where  $x_{ijk}$  is a binary variable that equals 1 if crew  $k$  travels from node  $i$  to node  $j$  and 0, otherwise. The traveling time through each link  $(\bar{i}, \bar{j})$  is  $c_{\bar{i}\bar{j}}$ . Similar to Akbari and Salman (2017), we assume that the traveling time from  $\bar{i}$  to  $\bar{j}$  is equivalent to the traveling time from  $\bar{j}$  to  $\bar{i}$  (i.e.,  $c_{\bar{i}\bar{j}} = c_{\bar{j}\bar{i}}$ ). When a restoration crew arrives to a node in  $\tilde{N}_{A'} \subseteq \tilde{N}$ , or its counterpart disrupted location in the infrastructure network, it

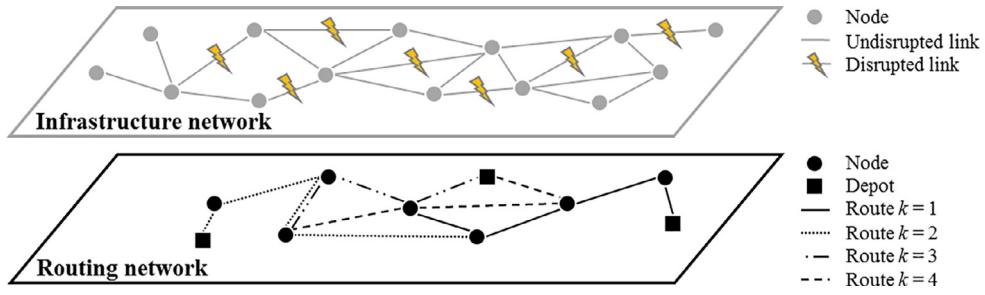


Fig. 2. The geographical relation between the routing network and the infrastructure network.

spends a particular amount of time to restore that location, whether alone or with the aids of other crews, and then leaves the node to join another restoration task. As more restoration tasks are completed in the routing network, we see the infrastructure network performance enhanced (i.e., the total flow reaching to the demand nodes increases). Fig. 2 depicts the relation between the infrastructure network and its corresponding routing network after a disruptive event. In Fig. 2, the disrupted locations on links in the infrastructure network, shown by lightning bolts, are the nodes in the routing network, shown by circular nodes. Each particular line in the routing network (e.g., bold, bold dashing, long and small dashes), is associated with the route of a particular restoration crew starting from a depot, shown by a square node, and may intersect and share nodes with other restoration routes. As mentioned in Section 1, the routing network itself remained unaffected by the disruptive effects in such way that there is always a path which connects each pair of nodes in the routing network, whether the pre-disruption shortest path unaffected by disruptions or the shortest undisrupted path between those corresponding nodes.

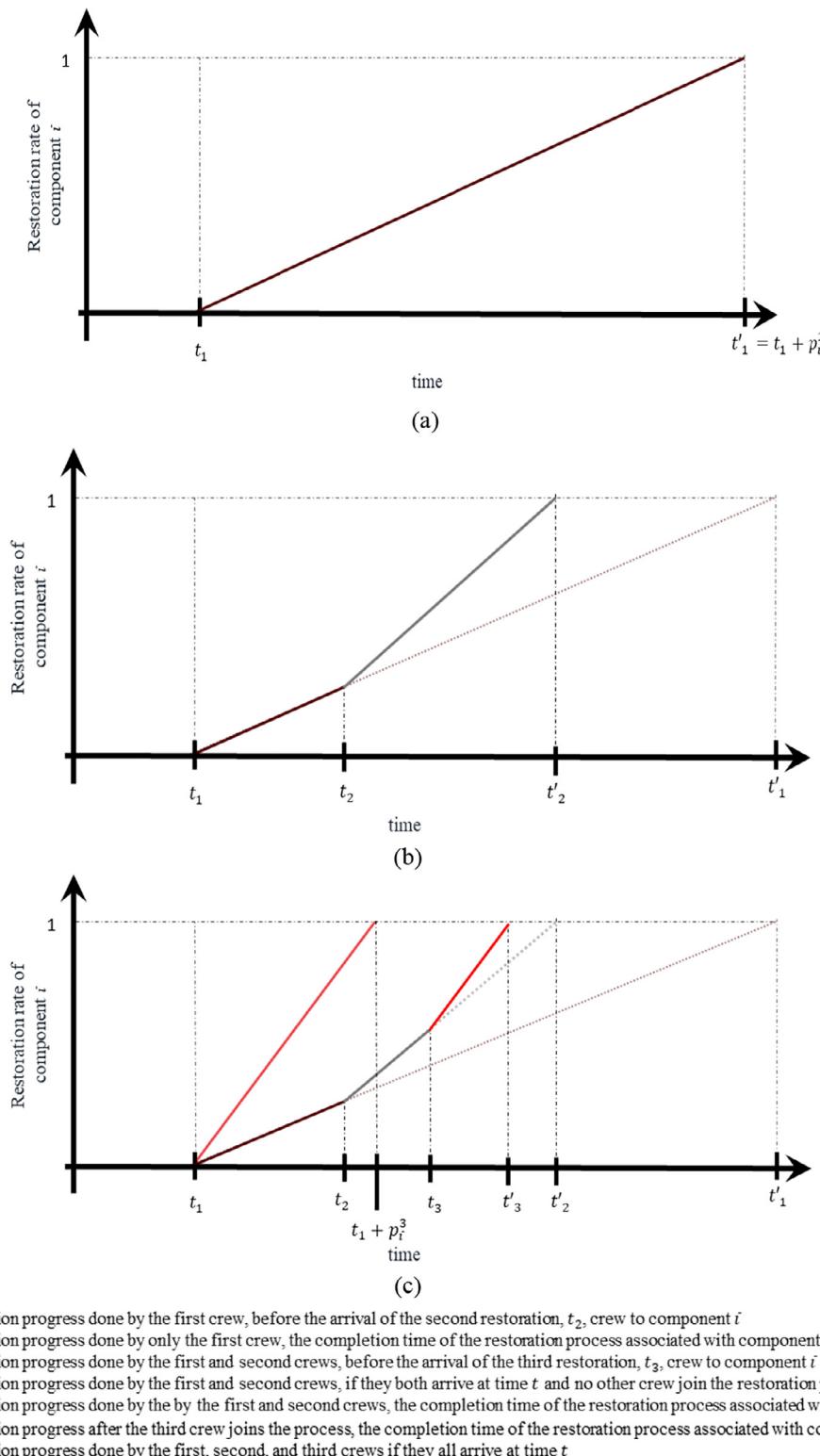
After a disruptive event, the mixed integer programming (MIP) formulation determines  $|K|$  open routes on the complete graph  $\tilde{G} = (\tilde{N}, \tilde{A})$  such that network  $G = (N, A)$  becomes fully operational after all disrupted arcs in  $A' \subseteq A$  are restored. Without loss of generality, we model disrupted nodes using disrupted links, since each node can be represented as two nodes and one link. Each restoration crew starts its route from its originating depot and ends in a dummy sink node,  $(n+1)$ . The disrupted links in network  $G$  are assigned to multiple parallel restoration crews, where the total number of crews available to work on each link is  $|K|$ , and the maximum number of restoration crews that are able to work at each period of time is  $L$ . Note that more than one crew is able to work on a disrupted link simultaneously. The processing time of each disrupted link  $(i, j) \in A'$  depends on the characteristics of that link, its level of disruption, the number of restoration crews assigned to it, and the arrival time of each crew to that link. In this paper, we consider the fact that doubling or tripling a working crew does not lead to a doubled or tripled as much as the restoration rate associated with it (e.g., due to some conflicts among crews and their role in the restoration process). Furthermore, we consider that the increase in the number of working crews would not necessarily result in the increase in the restoration rate (i.e., the restoration rate might remain the same). To implement this consideration, we choose 10% of total disrupted components randomly and set their rate of recovery (processing time) in such a way that after adding a certain number of crews, no increase would be observed in the rate of recovery.

### 3.1. Dynamic restoration process

Depending on its originating depot and route, the arrival time of each restoration crew assigned to each disrupted node  $\bar{i} \in \tilde{N}_{A'}$  may be different from other crews assigned to that node. The restoration process starts as soon as the first crew arrives to node  $\bar{i} \in \tilde{N}_{A'}$  and each time a new crew joins the restoration process, it accelerates the remaining restoration process and consequently decreases the remaining processing time of node  $\bar{i}$ . Fig. 3 illustrates how the restoration rate accelerates when a new crew joins the restoration process (relative to when crews arrive at the same time). We assume that  $p_{\bar{i}}^l$  is an integer parameter that represents the processing time of node  $\bar{i} \in \tilde{N}_{A'}$  if  $l$  restoration crews are assigned to  $\bar{i}$  and start their restoration tasks at the same time. Note that when all crews arrive to node  $\bar{i}$  at the same time, the restoration process will be completed sooner than when work crews arrive at different times. In Fig. 3(a), at time  $t_1$ , the first assigned crew arrives to node  $\bar{i} \in \tilde{N}_{A'}$  and starts the restoration process. If no other crew joins to the restoration process, it will be completed at time  $t_1'$ , in  $p_{\bar{i}}^1$  time periods. Fig. 3(b) illustrates when a second crew arrives at time  $t_2$ , showing how the restoration rate accelerates, and the process is completed at time  $t_2'$ , where  $t_1 + p_{\bar{i}}^2 < t_2' < t_1'$ . Finally, in Fig. 3(c), a third crew arrives and at time  $t_3$ , and the restoration process is completed in time  $t_3'$ , where  $t_1 + p_{\bar{i}}^3 < t_3' < t_2'$ .

The restoration of each link should be processed without interruption, and the model prevents time conflicts by calculating the arrival time of each crew  $k \in K$  at each node  $\bar{i} \in \tilde{N}_{A'}$ , or its counterpart disrupted location on a link  $A' \subseteq A$  in the infrastructure network  $G = (N, A)$ .

The objective of the optimization model is to maximize resilience of the infrastructure network over time, where resilience is measured as a time-dependent function of recovered network performance to total performance loss, as adapted from Henry and Ramirez-Marquez (2012). Noting that only the infrastructure network contains demand nodes, shown by



**Fig. 3.** Illustration of restoration rates as new crews arrive at a disrupted component.

$i \in N_-$ , the resilience measure tracks the trajectory of restoration at each time,  $t=1, \dots, T$ , by determining the maximum weighted flow, denoted by  $\sum_{i \in N_-} w_i \varphi_{it}$ , that reaches to demand nodes in the infrastructure network. The resilience measure is calculated for a particular disruption  $e^j$  following Eq. (1), where  $\sum_{i \in N_-} w_i \varphi_{it_e}$  is the total weighted flow reaching to demand nodes before the disruption and  $\sum_{i \in N_-} w_i \varphi_{it_d}$  is the total weighted flow of reaching to the demand nodes immediately after disruption  $e$  when no restoration crew takes part in the restoration process. Time periods  $t_e$  and  $t_d$  are illustrated in Fig. 1 as the pre- and post-disruption time periods.

$$\mathfrak{R}_\varphi(t|e) = \frac{\sum_{i \in N_-} w_i \varphi_{it} - \sum_{i \in N_-} w_i \varphi_{it_d}}{\sum_{i \in N_-} w_i \varphi_{it_e} - \sum_{i \in N_-} w_i \varphi_{it_d}} \quad (1)$$

### 3.2. Mathematical model

We present two variations on the proposed MIP restoration routing problem. In the *Binary Active* model, it is assumed that each disrupted link remains inoperable until the related recovery process is completed in full. Although the *Binary Active* model has many applications in many realistic case studies (e.g., water pipe networks, railways), other applications (e.g., road transportation networks, the physical structure of internet networks) assume that disrupted links can be partially operable during their restoration. As such, the *Proportional Active* model addresses this latter category of restoration problems in which the level of operability in each disrupted link  $(i, j) \in A'$  increases during its restoration process until it becomes completely operational. The *Proportional Active* model may not always be directly applicable to power, water, and gas networks. However, we used this model on the three proposed network instances, as it provides relevant insights on their performance and behavior. In particular, by using this model we can: (i) study the behavior of the model on various sizes of problem instances, from small size networks (i.e., gas network) to large size networks (i.e., power network), and (ii) consider the applicability of the model the problem instances where temporary and emergency components are installed along with the main infrastructure network to satisfy a portion of demand. Considering both formulations, the indices, parameters, and decision variables for the infrastructure network and routing network are found in Table 1.

In this paper, we consider that, in the case where the transportation network is damaged, the model uses the remaining undamaged subgraph. We further assume that the network is still connected after the disruptions. In cases where the subgraph does not connect all the disrupted locations in the network, the model restores only the accessible locations.

#### 3.2.1. MIP model for *Binary Active* network restoration

$$\max \sum_{t \in T} \mathfrak{R}_\varphi(t|e) \quad (2)$$

$$\sum_{k \in K} \sum_{\bar{j} \in \bar{N}_{A'} : (\bar{i}, \bar{j}) \in \bar{A}} x_{\bar{i}\bar{j}}^k = v_{\bar{i}} \quad \forall \bar{i} \in \bar{N}_D \quad (3)$$

$$\sum_{\bar{i} \in \bar{N} \setminus (n+1) : (\bar{i}, \bar{j}) \in \bar{N}} x_{\bar{i}\bar{j}}^k \leq 1 \quad \forall \bar{j} \in \bar{N}_{A'} \cup (n+1), k \in K \quad (4)$$

$$\sum_{\bar{j} \in \bar{N}_{A'} : (\bar{i}, \bar{j}) \in \bar{N}} x_{\bar{i}\bar{j}}^k \leq 1 \quad \forall \bar{i} \in \bar{N} \setminus (n+1), k \in K \quad (5)$$

$$\sum_{\bar{i} \in \bar{N}_{A'} \cup \bar{N}_D : (\bar{i}, \bar{j}) \in \bar{A}} x_{\bar{i}(n+1)}^k = 1 \quad k \in K \quad (6)$$

$$\sum_{\bar{i} \in \bar{N} \setminus (n+1) : (\bar{i}, \bar{j}) \in \bar{A}} x_{\bar{i}\bar{j}}^k - \sum_{\bar{i} \in \bar{N}_{A'} \cup (n+1) : (\bar{i}, \bar{j}) \in \bar{A}} x_{\bar{j}\bar{i}}^k = 0 \quad \forall \bar{j} \in \bar{N}_{A'}, k \in K \quad (7)$$

$$\sum_{\bar{i} \in \bar{N} \setminus (n+1) : (\bar{i}, \bar{j}) \in \bar{A}} x_{\bar{i}\bar{j}}^k = \sum_{t=1}^T \tau_{\bar{j}t}^k \quad \forall \bar{j} \in \bar{N}_{A'}, k \in K \quad (8)$$

$$\sum_{k \in K} \sum_{t=1}^T \tau_{\bar{i}t}^k = \sum_{l=1}^L l z_{\bar{i}}^l \quad \forall \bar{i} \in \bar{N}_{A'}, l = 1, \dots, L \quad (9)$$

$$\sum_{l=1}^L z_{\bar{i}}^l \leq 1 \quad \forall \bar{i} \in \bar{N}_{A'} \quad (10)$$

**Table 1**

Notation for the Binary and Proportional Active restorative capacity routing problems.

Infrastructure network notation	
$N$	Set of nodes in network $G = (N, A)$
$A$	Set of links in network $G = (N, A)$
$A'$	Set of disrupted links in network $G = (N, A)$
$\{1, \dots, T\}$	Set of time periods in the restoration horizon
Routing network notation	
$\tilde{N}$	Set of nodes in network $\tilde{G} = (\tilde{N}, \tilde{A})$
$\tilde{N}_{A'}$	Set of nodes in network $\tilde{G} = (\tilde{N}, \tilde{A})$ corresponding to disrupted links in network $G = (N, A)$
$\tilde{N}_D$	Set of depots from which recovery crews commence their routes
$\tilde{A}$	Set of links in network $\tilde{G} = (\tilde{N}, \tilde{A})$ which connects the nodes corresponding to disrupted links in network $G = (N, A)$
$K$	Set of restoration crews, where $ K $ is the maximum number of available crews through the restoration horizon
$\{1, \dots, L\}$	Set of restoration crews assigned to each node $\bar{i} \in \tilde{N}_{A'}$ , where $L$ is the maximum number of crews that can be assigned to each disrupted component
$(n+1)$	The dummy sink node where the routes of all restoration crews ends
Parameters	
$v_{\bar{i}}$	Maximum number of restoration crews sent from depot $\bar{i} \in \tilde{N}_D$
$p_{\bar{i}}^l$	Processing time of node $\bar{i} \in \tilde{N}_{A'}$ when $l$ crews are assigned to its restoration
$c_{\bar{i}\bar{j}}$	Traveling time from node $\bar{i}$ to node $\bar{j}$ , $(\bar{i}, \bar{j}) \in \tilde{A}$
$\theta_{\bar{i}\bar{j}}$	Binary parameter equals to 1 if node $\bar{i} \in \tilde{N}_{A'}$ represents link $(i, j) \in A'$ in graph $G = (N, A)$
$u_{\bar{i},te}$	Capacity of node $\bar{i} \in \tilde{N}_{A'}$ , or its corresponding link $(i, j) \in A'$ , before the disruptive event
$u_{\bar{i},td}$	Capacity of node $\bar{i} \in \tilde{N}_{A'}$ , or its corresponding link $(i, j) \in A'$ , immediately after the disruptive event
$b_i$	Capacity of demand node $i \in N_-$
$M$	Very large number
Decision variables	
$x_{\bar{i}\bar{j}}^k$	Binary variable equal to 1 if restoration crew $k \in K$ travels link $(\bar{i}, \bar{j}) \in \tilde{A}$
$z_{\bar{i}}^l$	Binary variable equal to 1 if $l$ restoration crews are assigned to node $\bar{i} \in \tilde{N}_{A'}$
$\tau_{\bar{i}t}^k$	Binary variable equal to 1 if restoration crew $k$ arrives to node $\bar{i} \in \tilde{N}_{A'}$ at time $t$
$g_{\bar{i}t}^l$	Binary variable equal to 1 if the $l^{th}$ restoration crew arrives to node $\bar{i} \in \tilde{N}_{A'}$ at time $t$
$\beta_{\bar{i}}^l$	Continuous variable representing the completion time of the restoration process associated with node $\bar{i}$ when $l$ crews are assigned
$f_{ijt}$	Integer variable representing the flow on link $(i, j) \in A$ at time $t$
$\phi_{it}$	Integer variable representing the flow reaching to demand node $i \in N_-$ at time $t$
$\alpha_{ijt}$	Binary variable equal to 1 if restoration task on link $(i, j)$ finishes at time $t$
$y_{\bar{i}}^k$	Binary variable equal to 1 if restoration crew $k \in K$ is assigned to node $\bar{i} \in \tilde{N}_{A'}$
$f_{\bar{i}\bar{j}}^k$	Integer variable representing the flow of restoration crew $k$ on link $(\bar{i}, \bar{j}) \in \tilde{A}$

Note: All subscripts with an overbar refer to nodes in the routing network

$$\sum_{t=1}^T t \tau_{\bar{j}t}^k \geq c_{\bar{i}\bar{j}} + \sum_{t=1}^T t \tau_{\bar{i}t}^k + p_{\bar{i}}^l - M(1 - x_{\bar{i}\bar{j}}^k) - M(1 - z_{\bar{i}}^l) \quad \forall \bar{i} \in \tilde{N}_{A'}, \forall \bar{j} \in \tilde{N}_{A'}, l = 1, \dots, L, k \in K \quad (11)$$

$$\sum_{t=1}^T t \tau_{\bar{j}t}^k \geq c_{\bar{i}\bar{j}} + \beta_{\bar{i}}^l - M(1 - x_{\bar{i}\bar{j}}^k) - M(1 - z_{\bar{i}}^l) \quad \forall \bar{i} \in \tilde{N}_{A'}, l = 2, \dots, L, k \in K \quad (12)$$

$$\sum_{t=1}^T t g_{\bar{i}t}^{l+1} \geq \sum_{t=1}^T t g_{\bar{i}t}^l - M \left( 1 - \sum_{\bar{i}=l+1}^L z_{\bar{i}}^l \right) \quad \forall \bar{i} \in \tilde{N}_{A'}, l = 1, \dots, L \quad (13)$$

$$\sum_{t=1}^T t g_{\bar{i}t}^{l+1} \leq \beta_{\bar{i}}^l + M \left( 1 - \sum_{\bar{i}=l+1}^L z_{\bar{i}}^l \right) \quad \forall \bar{i} \in \tilde{N}_{A'}, l = 1, \dots, L \quad (14)$$

$$\sum_{k \in K} \tau_{\bar{i}t}^k \geq \sum_{l=1}^L g_{\bar{i}t}^l \quad \forall \bar{i} \in \tilde{N}_{A'}, t = 1, \dots, T \quad (15)$$

$$\sum_{t=1}^T g_{\bar{i}t}^l = \sum_{\bar{i}=l}^L z_{\bar{i}}^l \quad l = 1, \dots, L, \forall \bar{i} \in \tilde{N}_{A'} \quad (16)$$

$$\sum_{j:(i,j) \in A} f_{ijt} - \sum_{j:(j,i) \in A} f_{jbt} \leq O_i \quad \forall i \in N_+, t = 1, \dots, T \quad (17)$$

$$\sum_{j:(i,j) \in A} f_{ijt} - \sum_{j:(j,i) \in A} f_{jit} = 0 \quad \forall i \in N_-, t = 1, \dots, T \quad (18)$$

$$\sum_{j:(i,j) \in A} f_{ijt} - \sum_{j:(i,j) \in A} f_{jit} = -\varphi_{it} \quad \forall i \in N_-, t = 1, \dots, T \quad (19)$$

$$0 \leq \varphi_{it} \leq b_i \quad \forall i \in N_-, t = 1, \dots, T \quad (20)$$

$$0 \leq f_{ijt} \leq u_{ij} \quad \forall (i, j) \in A, t = 1, \dots, T \quad (21)$$

$$0 \leq f_{ijt} \leq \sum_{s=1}^t \alpha_{ijs} u_{ij} \quad \forall (i, j) \in A', t = 1, \dots, T \quad (22)$$

$$\sum_{s=1}^T s \alpha_{ijs} \geq \beta_i^l - M(1 - z_i^l) - M(1 - \theta_{ij}) \quad \forall h \in \bar{N}_{A'}, \forall (i, j) \in A', k \in K \quad (23)$$

$$\sum_{s=1}^T \alpha_{ijs} \leq 1 \quad \forall (i, j) \in A' \quad (24)$$

$$z_i^l = \{0, 1\} \quad l = 1, \dots, L, \forall \bar{i} \in \bar{N}_{A'} \quad (25)$$

$$\tau_{it}^k = \{0, 1\}, g_{it}^l = \{0, 1\} \quad \forall \bar{i} \in \bar{N}_{A'}, k \in K, l = 1, \dots, L, t = 1, \dots, T \quad (26)$$

$$x_{ij}^k = \{0, 1\} \quad (\bar{i}, \bar{j}) \in \bar{A}, k \in K \quad (27)$$

$$\varphi_{it} > 0, i \in N_- \quad \forall i \in N_-, t = 1, \dots, T \quad (28)$$

$$\alpha_{ijt} = \{0, 1\}, f_{ijt} > 0 \quad (i, j) \in A, t = 1, \dots, T \quad (29)$$

The objective function focuses on the performance of the infrastructure network as determined by its resilience measure over the horizon of restoration. Eqs. (3)–(8) are restoration crew routing balance equations. Eq. (3) requires that at most  $v_i$  restoration crews can be dispatched from each depot  $\forall \bar{i} \in \bar{N}_D$ . Eqs. (4) and (5) ensure that each restoration crew travels through each link  $(\bar{i}, \bar{j}) \in \bar{A}$  and visits each node  $\bar{i} \in \bar{N}_{A'}$  at most once, respectively. In Eq. (6), a dummy sink node,  $(n+1)$ , is considered for crews where their routes end, where  $x_{\bar{i}(n+1)}^k$ ,  $\bar{i} \in \bar{N}_D$  is equal to 1 for crew  $k \in K$  when it is not used in the restoration process and does not leave its depot at all. In Eq. (7), each crew  $k \in K$  that enters a node  $\bar{i} \in \bar{N}_{A'}$  should leave that node after restoration tasks are completed. In Eq. (8), no crew travels link  $(\bar{i}, \bar{j}) \in \bar{A}$  unless it is scheduled to restore node  $\bar{j} \in \bar{N}_{A'}$ . Eq. (9) ensures that no crew visits node  $\bar{i}$  unless it is assigned to that corresponding node. Eq. (10) ensures that when a certain number of restoration crews are assigned to link  $(i, j) \in A'$ , or its counterpart node  $h \in \bar{N}_{A'}$ , then the number of crews cannot be changed during the restoration process.

Eqs. (11)–(14) determine the arrival time related to each restoration crew,  $k \in K$ ,  $\bar{i} \in \bar{N}_{A'}$ , and the processing time associated with each disrupted link. Eqs. (11) and (12) calculate the arrival time of each restoration crew  $k \in K$  to each disrupted node  $\bar{j} \in \bar{N}_{A'}$  from node  $\bar{i} \in \bar{N} \setminus (n+1)$ . It is assumed that all restoration crews work independently, and each crew starts the restoration process as soon as it arrives to any disrupted node. Consider disrupted node  $\bar{j}$  to which we assigned  $l$  restoration crews. After completing the restoration process associated with node  $\bar{i} \in \bar{N}$  at time  $\beta_{\bar{i}}^l$ , the first crew arrives at time  $t\tau_{\bar{j}t}^k = t\tau_{\bar{j}t}^1 = \beta_{\bar{i}}^l + c_{\bar{i}\bar{j}}$ , for  $\bar{j} \in \bar{N}_{A'}$ ,  $k \in K$ , and commences restoration operations with the recovery rate  $\lambda_{\bar{i}}^1$ , then the second crew arrive at time  $t'\tau_{\bar{j}t'}^k = t'\tau_{\bar{j}t'}^2 = \beta_{\bar{i}}^l + c_{\bar{i}\bar{j}}$ , after completing the restoration process of node  $\bar{i}$ , and so forth. Each time a new crew joins to the restoration process of a disrupted link, its rate of increases. Therefore, the processing time of each disrupted link  $(i, j) \in A'$ , or its counterpart node  $\bar{j} \in \bar{N}_{A'}$ , is a function of the arrival time of that crew to that corresponding link. Eqs. (13) and (14) set a time window for  $l$ th restoration crew,  $l = 1, \dots, L$ , arriving to node  $\bar{i} \in \bar{N}_{A'}$  starting from the arrival time of the prior restoration crew and ending to the completion time of the restoration process of by  $l-1$  restoration crews. Eqs. (15) and (16) sort the arrival time associated with crews assigned to each disrupted link.

Eqs. (17)–(22) are flow balance equations through supply nodes, transition nodes, and demand nodes. Eq. (20) ensures that the amount of flow reaching to each demand node  $i \in N_-$  does not exceed the capacity of that demand node. Eqs. (21) and (22) require that the flow of each link  $(i, j) \in A$ , whether undisrupted, disrupted, or recovered, does not exceed

the capacity of that link. [Eq. \(23\)](#) demonstrates that once the restoration process of each link  $(i, j) \in A'$ , or its counterpart node  $\bar{i} \in \bar{N}_{A'}$ , is completed, it becomes fully operational. [Eq. \(24\)](#) ensures that none of the disrupted links receives restoration services more than once.

To clarify how  $\beta_{\bar{i}}^l$  is calculated and implemented in the mathematical model, consider node  $\bar{i} \in \bar{N}_{A'}$  in routing network  $\bar{G} = (\bar{N}, \bar{A})$ , to which  $l = 1, \dots, L$  restoration crews are assigned, where  $L$  is the maximum number of crews that can be assigned to  $\bar{i}$ . Each assigned crew arrives at a particular time  $t = 1, \dots, T$ , that might be different from the arrival time of other assigned crews. In this paper, the restoration progress in node  $\bar{i}$  is measured in terms of the increase in the capacity of node  $\bar{i}$  in a given time window  $\Delta t = t_{l'+1} - t_{l'}$ ,  $l' = 1, \dots, l$  (i.e., between the arrival time of  $l'$ th and  $(l'+1)$ st assigned crews).

Referred to as  $F_{\bar{i}}^{l'}(\Delta t)$ , the restoration progress related to node  $\bar{i}$  is calculated as follows. As the first restoration crew arrives to  $\bar{i}$ , the restoration process begins. The restoration progress of node  $\bar{i}$  in the time window after the arrival of the first crew,  $t_1$ , and before the arrival of the second crew,  $t_2$ , is  $F_{\bar{i}}^1(t - t_1)$ ,  $t_1 \leq t < t_2$ . Immediately after the arrival of the second crew, the restoration progress is accelerated and upgraded to  $F_{\bar{i}}^2$ . The restoration progress in the time window after the arrival of the second crew,  $t_2$ , and before the arrival of the third crew,  $t_3$ , is  $F_{\bar{i}}^2(t - t_2)$ ,  $t_2 \leq t < t_3$ , and the total restoration progress to that point is  $F_{\bar{i}}^1(t_2 - t_1) + F_{\bar{i}}^2(t - t_2)$ ,  $t_2 \leq t < t_3$ . Finally, after the arrival of the  $l$ th crew, the restoration progress is upgraded to  $F_{\bar{i}}^l(t)$ ,  $t_l < t$ , and the total restoration progress is calculated as  $\sum_{n=1}^{l-1} F_{\bar{i}}^n(t_{n+1} - t_n) + F_{\bar{i}}^l(t - t_l)$ .

Considering parameters  $u_{\bar{i}te}$  as the capacity of node  $\bar{i} \in \bar{N}_{A'}$ , or its corresponding link  $(i, j) \in A'$ , before the disruption (i.e., at time  $t_e$ ), and  $u_{\bar{i}td}$  as its residual capacity after the disruption (i.e., at time  $t_d$ ), the restoration process continues until node  $\bar{i}$  is completely operational, or its capacity is fully restored (i.e.,  $u_{\bar{i}td} + \sum_{h=1}^{l-1} F_{\bar{i}}^h(t_{n+1} - t_n) + F_{\bar{i}}^l(t_l) \simeq u_{\bar{i}te}$ ).

Being familiar with the performance of restoration progress,  $\{F_{\bar{i}}^{l'}\}^{-1}(\Delta u)$  is the inverse function of restoration progress and calculates the time required to have  $\Delta u$  progress in the restoration process of node  $\bar{i} \in \bar{N}_{A'}$  with  $l'$  crews assigned.  $\Delta u$  is the amount of disruption in node  $\bar{i}$  that should be restored. The domain of  $\Delta u$  is  $0 \leq \Delta u \leq u_{\bar{i}te} - u_{\bar{i}td}$ , and the domain of the inverse function is  $0 \leq \{F_{\bar{i}}^{l'}\}^{-1}(\Delta u) \leq p_{\bar{i}}^{l'}$ . Considering  $(u_{\bar{i}te} - u_{\bar{i}td})$  as the total loss in the capacity of node  $\bar{i}$ , when the  $l$ th crew arrives to node  $\bar{i}$ , there is exactly  $(u_{\bar{i}te} - u_{\bar{i}td}) - \sum_{h=2}^{l-1} F_{\bar{i}}^{(h-1)}(t_h - t_{h-1})$  units of capacity left disrupted, and consequently  $\{F_{\bar{i}}^l\}^{-1}((u_{\bar{i}te} - u_{\bar{i}td}) - \sum_{h=2}^{l-1} F_{\bar{i}}^{(h-1)}(t_h - t_{h-1}))$  is the time required to finish the restoration process of node  $\bar{i}$  after the arrival of  $l$ th assigned crew.

To illustrate the calculation of the completion time of the restoration process associated with each disrupted node  $\bar{i} \in \bar{N}_{A'}$ , consider the example in which the first crew arrives at node  $\bar{i}$  at time  $t_1$  and starts the restoration process. If only one crew is assigned to node  $\bar{i} \in \bar{N}_{A'}$ , the restoration process will be completed in  $\{F_{\bar{i}}^1\}^{-1}(u_{\bar{i}te} - u_{\bar{i}td}) = p_{\bar{i}}^1$  time periods, where  $p_{\bar{i}}^1$  is the processing time of node  $\bar{i} \in \bar{N}_{A'}$  when only one crew is assigned to node  $\bar{i}$ . Otherwise, the next restoration crew arrives at time  $t_2$ , where  $t_1 \leq t_2 < t_1 + \{F_{\bar{i}}^1\}^{-1}(u_{\bar{i}te} - u_{\bar{i}td})$  (i.e., the second crew arrives before the restoration process of node  $\bar{i}$  is completed), accelerates the restoration process of the remaining task, completing it in  $\{F_{\bar{i}}^2\}^{-1}((u_{\bar{i}te} - u_{\bar{i}td}) - F_{\bar{i}}^1(t_2 - t_1))$ . The third crew arrives at time  $t_3$ ,  $t_2 \leq t_3 < t_2 + \{F_{\bar{i}}^2\}^{-1}((u_{\bar{i}te} - u_{\bar{i}td}) - F_{\bar{i}}^1(t_2 - t_1))$ , accelerates the restoration process of the remaining work, and will complete it in  $\{F_{\bar{i}}^3\}^{-1}(u_{\bar{i}te} - u_{\bar{i}td}) - (\sum_{l'=2}^3 F_{\bar{i}}^{(l'-1)}(t_{l'} - t_{l-1}))$ . Finally, the  $l$ th crew is the last restoration crew arriving at time  $t_l$ ,  $t_{l-1} \leq t_l < t_{l-1} + \{F_{\bar{i}}^{(l-1)}\}^{-1}((u_{\bar{i}te} - u_{\bar{i}td}) - \sum_{l'=2}^{l-1} F_{\bar{i}}^{(l'-1)}(t_{l'} - t_{l-1}))$  and the remaining tasks will be completed in  $\{F_{\bar{i}}^l\}^{-1}((u_{\bar{i}te} - u_{\bar{i}td}) - \sum_{h=2}^l F_{\bar{i}}^{(h-1)}(t_{h'} - t_{h-1}))$  time periods. The total restoration process associated with node  $\bar{i} \in \bar{N}_{A'}$  is calculated as  $t_l + \{F_{\bar{i}}^l\}^{-1}((u_{\bar{i}te} - u_{\bar{i}td}) - \sum_{h=2}^l F_{\bar{i}}^{(h-1)}(t_{h'} - t_{h-1}))$ . In general, the arrival time of  $l$ 'th restoration crew,  $l' = 1, \dots, l$ , to node  $\bar{i} \in \bar{N}_{A'}$  is shown by  $tg_{\bar{i}t}^{l'}$ , where  $g_{\bar{i}t}^{l'}$  is a binary variable, equals to 1 if the  $l$ 'th restoration crew arrives at node  $\bar{i}$ . Considering  $l$  crews assigned to each node  $\bar{i} \in \bar{N}_{A'}$ , the completion time of each node  $\bar{i} \in \bar{N}_{A'}$ ,  $\beta_{\bar{i}}^l$ , is equal to the completion time of its counterpart link  $(i, j) \in A'$  and calculated with [Eq. \(30\)](#).

$$\beta_{\bar{i}}^l = \sum_{t=1}^T tg_{\bar{i}t}^l + \{F_{\bar{i}}^l\}^{-1} \left( (u_{\bar{i}te} - u_{\bar{i}td}) - \sum_{l'=2}^l F_{\bar{i}}^{(l'-1)} \left( \sum_{t=1}^T tg_{\bar{i}t}^{l'} - \sum_{t=1}^T tg_{\bar{i}t}^{(l'-1)} \right) \right) \quad (30)$$

Without loss of generality, we consider a linear relationship between the progress in the restorative capacity of each link and restoration time of that corresponding link, updating [Eq. \(30\)](#) with [Eq. \(31\)](#). In [Eq. \(31\)](#), node  $\bar{i}$  is restored with rate  $\lambda_{\bar{i}}^{l'}$  when the  $l'$  restoration crews are working on node  $\bar{i}$ ,  $F_{\bar{i}}^{l'}(\Delta t) = \lambda_{\bar{i}}^{l'} \Delta t$ . The recovery time for node  $\bar{i}$  when one restoration crew is assigned is  $p_{\bar{i}}^1$ , therefore  $p_{\bar{i}}^l$  is the recovery time when  $l$  crews commence restoring node  $\bar{i}$  at the same time, where  $l = 1, \dots, L$  is the number of crews assigned to a link (i.e., its counterpart node in network  $\bar{G}$ ).

$$\beta_{\bar{i}}^l = \sum_{t=1}^T tg_{\bar{i}t}^l + \left( \frac{(u_{\bar{i}te} - u_{\bar{i}td}) - \sum_{l'=2}^l \lambda_{\bar{i}}^{(l'-1)} \left( \sum_{t=1}^T tg_{\bar{i}t}^{l'} - \sum_{t=1}^T tg_{\bar{i}t}^{(l'-1)} \right)}{\lambda_{\bar{i}}^l} \right) \quad (31)$$

### 3.2.2. MIP model for Proportional Active network restoration

In the Proportional Active formulation, the processing time of each link  $(i, j) \in A'$  is presented as a function of: (i) the number of assigned restoration crews to that link, (ii) the level of disruption associated with that link and the set of required tasks for its restoration, and (iii) the characteristics of that link, such as the level of disruption it experiences and the series of required task for its recovery. It is assumed that each recovery task should be processed without interruption. The formulation has many of the same constraints as the Binary Active model with the addition of Eq. (32), which calculates the improvement in the restoration process of each disrupted link  $(i, j) \in A'$  in each time period after its restoration process commences.

$$\max \sum_{t \in T} \mathfrak{R}_\varphi(t|e) \quad (32)$$

(4)-(22)

$$f_{ijt} \leq \sum_{s=1}^t (t-s) \left( \lambda_{\bar{i}}^1 g_{\bar{i}s}^1 + \sum_{h=1}^{l-1} (\lambda_{\bar{i}}^{h+1} g_{\bar{i}s}^{h+1} - \lambda_{\bar{i}}^h g_{\bar{i}s}^h) \right) + M(1 - z_{\bar{i}}^l) + M(1 - \theta_{ij\bar{i}}) \quad \forall (i, j) \in A', t = 1, \dots, T, l = 1, \dots, L, \bar{i} \in \bar{N}_{A'} \quad (33)$$

### 3.3. MIP model for relaxed network restoration

Since the model is complicated by tracking the arrival time of crews to each node  $\bar{j} \in \bar{N}_{A'}$ , and consequently calculating the restoration processing time, we present a relaxed formulation of the proposed problem such that the timing of restoration crews is ignored. This helps to reduce the number of binary variables from  $O(n^2K + m^2T)$  to  $O(n^2K)$ . However, it may result in solutions that are infeasible for the original formulations as (i) it may assign crews to a disrupted node  $\bar{i} \in \bar{N}_{A'}$  which arrive after the completion of restoration process to that corresponding node, and (ii) it may form direct cycles in the network, meaning that two or more crews are present in two different locations at the same time. To tackle this problem, we derive a feasible solution by modifying the assignment of restoration tasks to crews using *Initial Solution Preprocessing & Feasibility* algorithm in Section 3.3.1.

$$\min \Omega \quad (34)$$

(3)-(7)

$$\sum_{k \in K} \sum_{(\bar{i}, \bar{j}) \in \bar{A}} c_{\bar{i}\bar{j}} x_{\bar{i}\bar{j}}^k + \sum_{l=1}^L \sum_{\bar{i} \in \bar{N}_{A'}} p_{\bar{i}}^l z_{\bar{i}}^l \leq \Omega \quad (35)$$

$$\sum_{k \in K} \sum_{\bar{i} \in \bar{N} \setminus (n+1) : (\bar{i}, \bar{j}) \in \bar{N}} x_{\bar{i}\bar{j}}^k \geq 1 \quad \forall \bar{j} \in \bar{N}_{A'} \quad (36)$$

$$\sum_{\bar{i} \in \bar{N} \setminus (n+1) : (\bar{i}, \bar{j}) \in \bar{A}} \bar{f}_{\bar{i}\bar{j}}^k - \sum_{i \in \bar{N}_{A'} \cup \bar{N}_D : (\bar{i}, \bar{j}) \in \bar{A}} \bar{f}_{\bar{j}\bar{i}}^k = y_{\bar{j}}^k \quad \forall \bar{j} \in \bar{N}_{A'}, k \in K \quad (37)$$

$$\bar{f}_{\bar{i}\bar{j}}^k - \sum_{\bar{i} \in \bar{N}_{A'} : (\bar{i}, \bar{j}) \in \bar{A}} y_{\bar{i}}^k \geq -|\bar{N}| x_{\bar{i}\bar{j}}^k \quad \forall \bar{i} \in \bar{N}_D, \forall j \in \bar{N}_{A'}, k \in K \quad (38)$$

$$x_{\bar{i}\bar{j}}^k \leq Q_{\bar{i}\bar{j}}^k \leq |\bar{N}_{A'}| x_{\bar{i}\bar{j}}^k \quad \forall (\bar{i}, \bar{j}) \in \bar{A}, k \in K \quad (39)$$

Eq. (35) minimizes the total traveling and restoration time, and Eq. (36) ensures that all disrupted links and their counterpart nodes should be visited. Eqs. (37)–(39) provide flow balance. In Eq. (37), the net flow of each node  $\bar{i} \in \bar{N}_{A'}$  equals the number of the crews assigned to the corresponding node. For each depot, the net flow is the total number of nodes assigned to each restoration crew that starts its route from the corresponding depot, as shown in Eq. (38). Eq. (39) does not allow a crew to travel on a link unless it is traveled by that crew, and if a link is used by a crew, then there must be a positive amount of flow associated with the restoration crew passing through that link.

To incorporate  $w_i$  in the relaxed formulation, we first find the set of paths that push required flow to prioritized demand nodes. Among those paths, we determine the level of importance of each link from one of a number of importance measure types representing different graph theoretical (e.g., edge betweenness) or flow-based measures (e.g., edge flow centrality, maximum flow edge count, flow capacity impact) (Nicholson et al. 2016). We use  $I_\pi$  to refer to the importance measure calculated for each link  $(i, j) \in A'$ , or its counterpart node  $\bar{i} \in \bar{N}_{A'}$ , of type  $\pi$ . In this paper we consider three types of importance measure: (i) max flow edge count,  $I_{\text{MFcount}} = \frac{1}{n(n-1)} \sum_{\bar{s}, \bar{t} \in N} \mu_{\bar{s}\bar{t}}(i, j)$ , where  $\mu_{\bar{s}\bar{t}}(i, j)$  is a binary parameter and equals 1 if link  $(i, j)$  is used in a given source-sink max flow path, (ii) edge flow centrality,  $I_{\text{Flow}} = \frac{\sum_{\bar{s}, \bar{t} \in N} \omega_{\bar{s}\bar{t}}(i, j)}{\sum_{\bar{s}, \bar{t} \in N} \omega_{\bar{s}\bar{t}}}$ , where  $\omega_{\bar{s}\bar{t}}(i, j)$

**Table 2**

Order of disrupted links assigned to each crew and their restoration process completion time, adapted from [Akbari and Salman \(2017\)](#).

Crew	Order of disrupted links
1	$b_{11} \rightarrow b_{12} \rightarrow b_{13} \dots b_{1n_1}$
2	$b_{21} \rightarrow b_{22} \rightarrow b_{23} \dots b_{2n_2}$
$\vdots$	$\vdots$
$K$	$b_{K1} \rightarrow b_{K2} \rightarrow b_{K3} \dots b_{Kn_k}$
Completion time of restoration tasks	
1	$\tilde{F}_{11} = c_{Db_{11}} + \tilde{p}_{ib_{11}} \rightarrow \tilde{F}_{12} = \tilde{F}_{11} + c_{b_{11}b_{12}} + \tilde{p}_{ib_{12}} \dots \tilde{F}_{1n_1} = \tilde{F}_{1n_1-1} + c_{b_{1n_1-1}b_{1n_1}} + \tilde{p}_{ib_{1n_1}}$
2	$\tilde{F}_{21} = c_{Db_{21}} + \tilde{p}_{ib_{21}} \rightarrow \tilde{F}_{22} = \tilde{F}_{21} + c_{b_{21}b_{22}} + \tilde{p}_{ib_{22}} \dots \tilde{F}_{2n_2} = \tilde{F}_{2n_2-1} + c_{b_{2n_2-1}b_{2n_2}} + \tilde{p}_{ib_{2n_2}}$
$\vdots$	$\vdots$
$K$	$\tilde{F}_{K1} = c_{Db_{K1}} + \tilde{p}_{ib_{K1}} \rightarrow \tilde{F}_{K2} = \tilde{F}_{K1} + c_{b_{Kn_1}b_{Kn_2}} + \tilde{p}_{ib_{Kn_2}} \dots \tilde{F}_{Kn_1} = \tilde{F}_{Kn_1-1} + c_{b_{Kn_1-1}b_{Kn_1}} + \tilde{p}_{ib_{Kn_1}}$

is the flow on link  $(i, j)$  for all possible source-sink paths, and (iii) *flow capacity rate*,  $I_{\text{FCR}} = \frac{1}{n(n-1)} \frac{\sum_{\tilde{s}, \tilde{t} \in N} \omega_{\tilde{s}\tilde{t}}(i, j)}{c_{ij}}$ , where  $c_{ij}$  is the capacity of link  $(i, j)$ . More details about the calculation of these three importance measures are found in [Nicholson et al. \(2016\)](#).

We then cluster disrupted links based on their importance measure. Aligned with decision making policies, we may define various thresholds for clusters and represent different number of clusters. The more the number of defined clusters is, the more accurate the demand nodes are prioritized. Yet, the obtained solution may be different from the optimal solution which merely focuses on maximizing the network resilience enhancement. For example, if the importance measure of each link falls into the range of  $[0, 0.9]$ , links with the importance measure equal to or greater than 0.6 are categorized in cluster 1, or the most important set of links,  $\tilde{N}_{A'_1}$ , links with the importance measure between 0.3 and 0.5 are categorized in cluster 2,  $\tilde{N}_{A'_2}$ , and links with importance measure less than 0.3 are categorized in cluster 3,  $\tilde{N}_{A'_3}$ , or the least important set of links, where  $\tilde{N}_{A'_1} \cup \tilde{N}_{A'_2} \cup \dots \cup \tilde{N}_{A'_\Gamma} = \tilde{N}_{A'}$ . [Eq. \(40\)](#) then ensures that the disrupted links in  $\tilde{N}_{A'_1}$  should be restored before the disrupted links in  $\tilde{N}_{A'_2}$ , and disrupted link in cluster two should be served before disrupted links in  $\tilde{N}_{A'_3}$  and so forth.

$$\sum_{k \in K} \sum_{i \in \tilde{N}_{A'_{\gamma+1}}} \sum_{j \in \tilde{N}_{A'_\gamma}} x_{ij}^k = 0, \quad \gamma = 1, \dots, \Gamma - 1 \quad (40)$$

**Proposition 1.** The optimal recovery scheduling of the Relaxed Restorative Capacity problem,  $S_{R, RC}^*$ , builds a lower bound for the optimal solution to the original formulation.

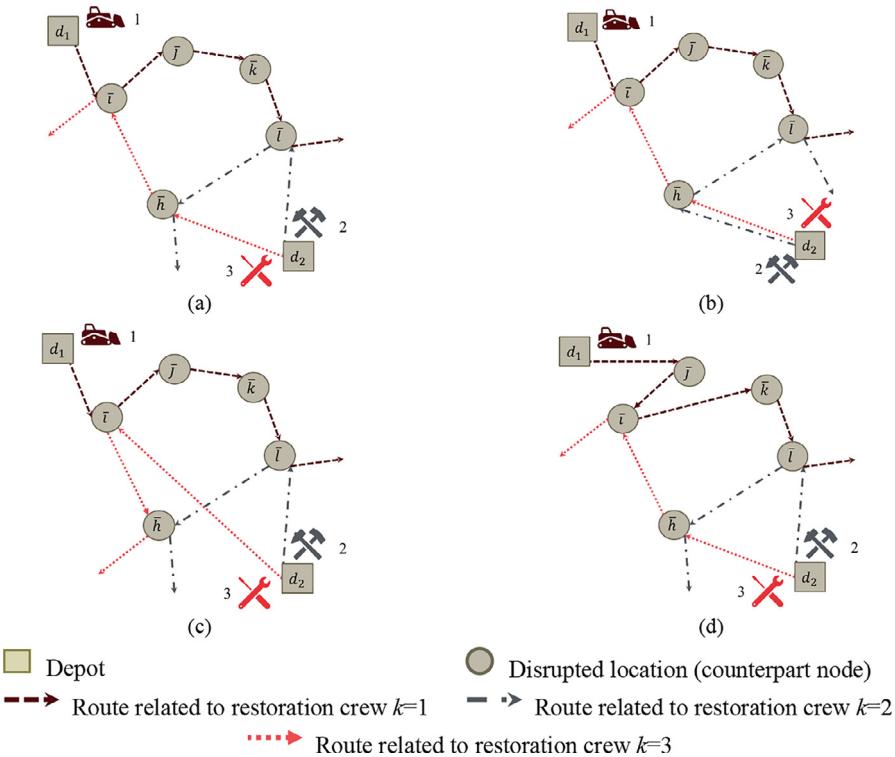
The proof of [Proposition 1](#) is given in [Appendix A-1](#).

### 3.3.1. Solution approach

Since time-related variables (e.g.,  $\tau_{\tilde{i}}^k$ ,  $g_{\tilde{i}}^l$ ,  $\alpha_{ijt}$ ,  $\phi_{it}$ ,  $f_{ijt}$ ) are not considered, the routing time associated with each restoration crew should be evaluated to be synchronized with other routes with which it has restoration tasks in common. To achieve this, we start by obtaining a lower bound for the original problem by using the relaxed formulation. Then, we use [Algorithm 2](#) to obtain a feasible solution for the original problem. Afterwards, we compare this feasible solution with the solution obtained from solving the original formulations. Note that if the original formulation could not be solved to optimality due to limited computation time, we may simply use the best solution achieved.

The proposed feasibility algorithm modifies the optimal solution obtained by the relaxed formulation as follows. First, adapted from [Akbari and Salman \(2017\)](#), with the results obtained from the relaxed formulation, we form a solution table such as [Table 2](#), to illustrate the scheduled set of disrupted links assigned to each restoration crew and the completion time of each restoration task assigned to that crew. Here,  $b_{kh}$  is the  $h$ th disrupted link  $(i, j) \in A'$  visited by crew  $k$ , and  $\tau_{ki}$  is the time when the restoration process of this link is completed. As the number of disrupted links assigned to each restorative crew can vary, to facilitate the update of restoration orders, we construct  $|K|$  lists in [Table 2](#) in which there are  $n_k$  elements in row  $k$ .

Then we detect whether there is a directed cycle in the graph (i.e., whether there is a particular restoration crew that is present in two different locations at the same time). For example, from the relaxed formulation results, we know that two crews  $k$  and  $\kappa$  share disrupted nodes  $i$  and  $j$ . Crew  $k$  is scheduled to restore node  $i$  then node  $j$ , while crew  $\kappa$  is scheduled to restore node  $j$  and then node  $i$ . This schedule prevents the recovery task completion of nodes  $i$  and  $j$  as two crews should be present in two different locations at the same time and therefore it is an infeasible solution for original formulation. Inspired by the Depth First Search (DFS) algorithm, we present the *Direct Cycle Elimination* algorithm using the DFS algorithm to identify the direct cycles and eliminate them by reversing the restoration order of one of the involved routes that intersect with the corresponding cycle. One input to the algorithm is a list including the scheduled set of disrupted links assigned to each restoration crew,  $B$ . Another input is a dictionary, named *graph*, whose *keys* are the all nodes,  $\tilde{i} \in \tilde{N}_{A'}$ , in the routing network and the *values* associated with each *key* are the nodes,  $\tilde{j} \in \tilde{N}_{A'}$ , where  $x_{ij}^k = 1$ ,  $k \in K$ , and  $x = 1$ . The output is a list of



**Fig. 4.** Illustrative example of [Algorithm 1](#), the Direct Cycle Elimination algorithm: (a) the routing network contains the direct cycle  $\bar{i} \rightarrow \bar{j} \rightarrow \bar{k} \rightarrow \bar{l} \rightarrow \bar{h} \rightarrow \bar{i}$ , (b) the direct cycle is eliminated by changing the route of crew 2 from  $d_2 \rightarrow \bar{l} \rightarrow \bar{h}$  to  $d_2 \rightarrow \bar{h} \rightarrow \bar{l}$ , (c) the direct cycle is eliminated by changing the route of crew 3 from  $d_2 \rightarrow \bar{h} \rightarrow \bar{i}$  to  $d_2 \rightarrow \bar{i} \rightarrow \bar{h}$ , and (d) the direct cycle is eliminated by changing the route of crew 1 from  $d_1 \rightarrow \bar{i} \rightarrow \bar{j} \rightarrow \bar{k} \rightarrow \bar{l}$  to  $d_1 \rightarrow \bar{j} \rightarrow \bar{i} \rightarrow \bar{k} \rightarrow \bar{l}$ .

scheduled links to each crew forming a routing network without any direct cycle. The steps of the proposed algorithm are as follows.

Using the *Convert Procedure*, we convert the obtained list  $B$  to its counterpart *dictionary graph*. In the *Cycle Detection Procedure*, we use the DFS algorithm to find all cycles in the routing network and put them in a list, named  $all_{path}$ . Then, in the *Elimination Procedure*, we pick the first cycle in  $all_{path}$ , find the sequence of a pair of nodes  $(\bar{i} \rightarrow \bar{j})$  in that cycle with the maximum repetition in all cycles in  $all_{path}$ , change the order of the corresponding nodes  $(\bar{i} \rightarrow \bar{j}) \rightarrow (\bar{j} \rightarrow \bar{i})$  in the schedule of each crew in  $B$ , and delete all cycles that include  $(\bar{i} \rightarrow \bar{j})$  from  $all_{path}$ . The output of this procedure is the updated list  $B$ . The algorithm repeats until it finds no further direct cycles the routing network formed from  $B$  (i.e.,  $x=0$ ). [Fig. 4](#) presents an illustration of the performance of the *Direct Cycle Elimination* algorithm. As [Fig. 4a](#) shows, the restoration crews one, two, and three form the direct cycle  $\bar{i} \rightarrow \bar{j} \rightarrow \bar{k} \rightarrow \bar{l} \rightarrow \bar{h} \rightarrow \bar{i}$ . To eliminate this cycle, we can change the restoration order associated with crew 2 from  $d_2 \rightarrow \bar{l} \rightarrow \bar{h}$  to  $d_2 \rightarrow \bar{h} \rightarrow \bar{l}$ , as shown in [Fig. 4b](#), or the restoration order associated with crew 3 from  $d_2 \rightarrow \bar{h} \rightarrow \bar{i}$  to  $d_2 \rightarrow \bar{i} \rightarrow \bar{h}$ , shown in [Fig. 4c](#), or the restoration order associated with crew 1 from  $d_1 \rightarrow \bar{i} \rightarrow \bar{j} \rightarrow \bar{k} \rightarrow \bar{l}$  to  $d_1 \rightarrow \bar{j} \rightarrow \bar{i} \rightarrow \bar{k} \rightarrow \bar{l}$ , shown in [Fig. 4d](#). According to [Algorithm 1](#), as changing the route of crew two has the least increase in traveling cost, we update the routing network accordingly in [Fig. 4b](#). The updated routing network does not contain a direct cycle.

Next, we present the *Initial Solution Preprocessing & Feasibility* algorithm to detect whether there is a timing conflict among the restoration crews assigned to each disrupted link. That is, an assigned restoration crew  $k \in K$  arrives to a disrupted link, or its corresponding node  $\bar{i} \in \bar{N}_{A'}$ , after that link has been restored. In this situation, two options are considered: (i) the position of node  $\bar{i} \in \bar{N}_{A'}$  in the outlier crew schedule in [Table 2](#) is swapped with one of the precedent nodes in the schedule where the arrival time of crew  $k$  to node  $\bar{i}$  falls in the restoration task time window (*Shift Procedure*), and (ii) node  $\bar{i} \in \bar{N}_{A'}$  is simply deleted from the restoration schedule of the crew  $k$  (*Delete Procedure*).

**3.3.1.1. Shift Procedure.** Considering the order of the corresponding disrupted node,  $\bar{j}$ , in the schedule of the outlier crew  $k$  in [Table 2](#),  $b_{kj}$ , look for a precedent disrupted node,  $\bar{j}$  (e.g.,  $b_{k\bar{j}} < \bar{j} < j$  in the outlier crew schedule) that is not shared with any other crew. If the summation of the arrival time to node,  $\bar{j}$ , and  $p_j^1$  is greater than maximum arrival time of other restoration crews to node  $j$ , we change the order of disrupted node to  $b_{kj}^{\text{new}} = b_{k\bar{j}}$  and consequently  $b_{kj}^{\text{new}} = b_{kj}$ . In cases where there is no precedent disrupted node,  $\bar{j}$ , with the defined characteristics, simply apply the *Delete Procedure*.

**Algorithm 1** Direct Cycle Elimination.

---

```


$$\begin{bmatrix} B_{11} & \dots & B_{1n_1} \\ \vdots & \ddots & \vdots \\ B_{K1} & \dots & B_{Kn_k} \end{bmatrix}$$


1: Input  $B = [\dots]$ ,  $graph$ ,  $x = 1$ 

Procedures:
2: Convert  $graph$  as an empty dictionary with all  $key \in V$  and all  $value = []$ 
3: for each  $key$  in  $V$  do:
4:   for all  $B_{ij} = key$   $graph$  [ $key$ ]. $value \leftarrow B_{ij+1}$ 
5: Cycle Detection Set  $graph \leftarrow$  Convert procedure,  $all\_path = []$ 
6: for  $i \in V_A$  do:
7:   Set  $start \leftarrow i$ ,  $end \leftarrow i$  and  $temp = [(i, [ ])]$ 
8:   While  $temp$  do
9:     if  $start = end$  do
10:     $all\_cycles \leftarrow path$  and go to line 8
11:    for  $next$  in  $graph$  [ $state$ ] do
12:      if  $next$  not in  $path$  do
13:         $path \leftarrow next$ 
14:         $state \leftarrow next$ 
15:         $temp = [(next, path)]$ 
16: Elimination Set  $all\_cycle \leftarrow$  Cycle detection
17: While  $all\_path \neq \emptyset$  do
18:   Among all links  $all\_path[1][i] \rightarrow all\_cycle[1][i+1]$ ,  $i = 1, \dots, |all\_cycle[1]|$  find one with the
19:   maximum repetition in cycles in  $all\_path$ 
20:   Substitute  $all\_cycle[1][i+1] \rightarrow all\_cycle[1][i]$  for all  $all\_cycle[1][i] \rightarrow all\_cycle[1][i+1]$  in  $B$ 
21:    $all\_path \leftarrow$  all cycles in  $all\_path$  but ones contain  $all\_path[1][i] \rightarrow all\_cycle[1][i+1]$ 
22: While  $x \neq 0$ 
23:    $B = Elimination(Cycle\ detection(graph), B)$ 
24:    $graph = Convert(V, B)$ 
25:    $all\_path = (graph)$ 
26:    $x = |all\_path|$ 
27: Return  $B$ 

```

---

**3.3.1.2. Delete Procedure.** Remove the disrupted link from the restoration schedule of the outlier crew and update its schedule by shifting the orders of the following disrupted links one step backward.

As multiple crews can restore a single disrupted link, it may appear in the restoration order of more than one crew. However, naturally, restoration of a link should not be repeated. We further define  $\tilde{\mathcal{F}}$  and  $\tilde{\mathcal{F}}$  as lists of  $|K|$  lists to track *fixed* and *unfixed* completion time, respectively. We note that each sub-list is related to a crew and has a defined length equal to the number of nodes assigned to that corresponding crew. During the algorithm, the completion time related to the restoration process of each node  $\tilde{i} \in \tilde{N}_{A'}$  is marked as *unfixed* if there is at least one unscheduled task ahead for that corresponding node in  $B$ , when a disrupted link (its corresponding node) is restored we mark its completion time as *fixed*, that is, no more restoration task remains as unscheduled for that corresponding link. To determine a restoration crew as an outlier for a disrupted link, all precedent restoration tasks assigned to the crew, and consequently their completion time, must be marked as *fixed*. After determining the exact time when crew  $k$ ,  $k \in K$ , arrives to the  $i$ th node (e.g., node  $\tilde{i}$ ),  $\tilde{\tau}_{ki}$ , we face three options: (i) the node is only assigned to one crew and the restoration time is  $p_i^1$ , in which case  $\tilde{\mathcal{F}}_{ki} = \tilde{\mathcal{F}}_{ki} = \tilde{\tau}_{ki} + p_i^1$ , (ii) the arrival time of crew  $k$  is greater than the completion time of the restoration task associated with that link, in which case we apply the *Shift* or *Delete* Procedures, or (iii) considering  $\bar{\kappa}$  restoration crews assigned and working on that corresponding node, the arrival time of crew  $k$ ,  $\tilde{\tau}_{ki}$ , to the  $i$ th node is smaller than the calculated completion time of the restoration process done by  $\bar{\kappa}$  restoration crews, yet  $\tilde{\tau}_{ki} + p_i^1$  is greater than the maximum arrival time related to  $\bar{\kappa}$  restoration crews, in which case crew  $k$  joins to the restoration task of node  $\tilde{i}$  and the restoration rate of the remaining restoration work is accelerated. In cases where  $\tilde{\tau}_{ki} + p_i^1$  is smaller than the maximum arrival time related to  $\bar{\kappa}$  restoration crews, we apply the *Shift* or *Delete* Procedures on the restoration crew with the maximum arrival time and update the maximum arrival time related to  $\bar{\kappa}$  restoration crews, in case the *Shift* Procedure was applicable, or the maximum arrival time related to  $\bar{\kappa} - 1$  restoration crews. This procedure continues until  $\tilde{\tau}_{ki} + p_i^1$  is greater than the updated maximum arrival time related to remained crews working on the  $i$ th node. **Algorithm 2** continues until all  $b_{ik}$ ,  $k \in K$ ,  $i = 1, \dots, n_k$  are visited and their completion time is marked as *fixed*.

**Proposition 2.** The initial solution resulting from solving the *Initial Solution Preprocessing & Feasibility* algorithm is at least equal to the  $\max_{k \in K} \tilde{\mathcal{F}}_{kn_k}$  (i.e., the optimal solution obtained from Relaxed RCRP) and at most equal to  $|K| \times \max_{k \in K} \tilde{\mathcal{F}}_{kn_k}$  (i.e., the maximum routing time obtained from the Relaxed RCRP), which is at most equal to  $|K| \times \max_{k \in K} \tilde{\mathcal{F}}_{kn_k}$  obtained from the original Binary and Proportional RCRP.

The proof of **Proposition 2** is presented in **Appendix A-2**.

**Algorithm 2** Initial Solution Preprocessing & Feasibility Algorithm.

---

```

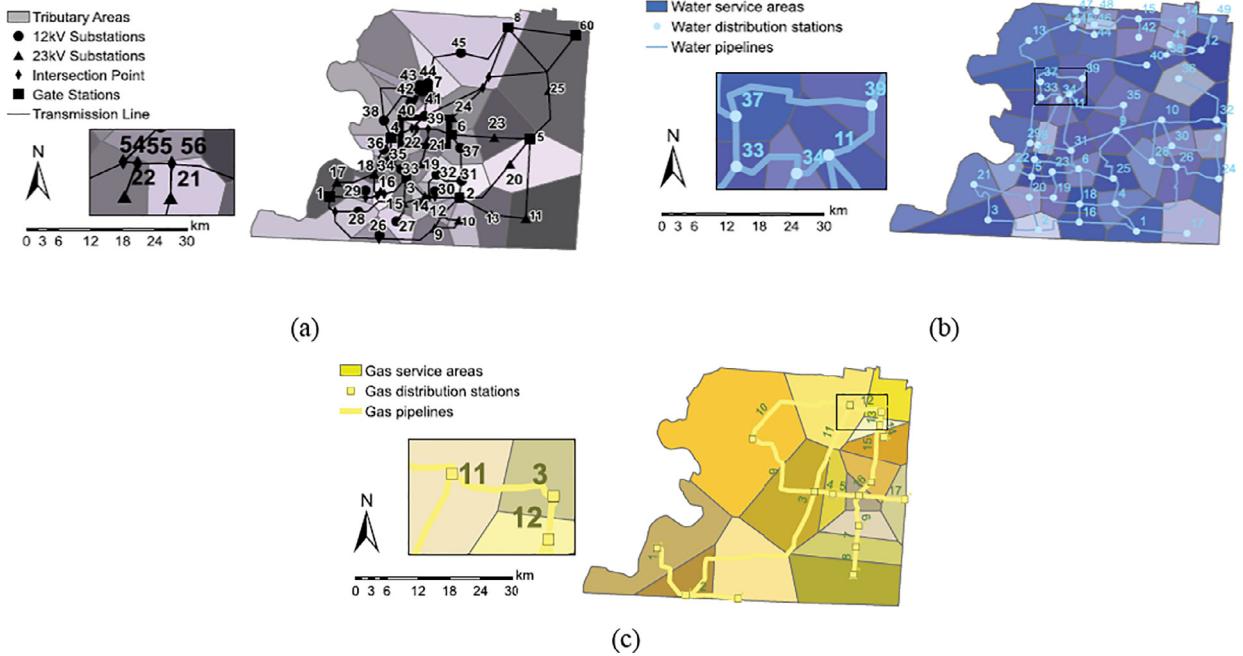
1: Input  $B, C$ 
2: Apply the Direct Cycle Algorithm on the input
3: Set  $\tilde{\mathcal{F}} = []$ ,  $\tilde{\mathcal{F}} = []$ ,  $\tilde{P} = []$ , and  $\tilde{\tau} = []$  for all disrupted links; each of which is a list of  $K$  lists and the length each list
   is equal to the number of disrupted nodes assigned to each restoration crew
4: for  $r = 1, \dots, K$  do
5:    $\tilde{\mathcal{F}}[r][1] = 1$ ,  $\tilde{\mathcal{F}}[r][1] = 1$ ,  $\tilde{P}[r][1] = 1$ , and  $\tilde{\tau}[r][1] = 1$ 
6: for  $h = 1, \dots, \max_{r=1,\dots,K}(|B[r]|)$  do
7:   for  $r = 1, \dots, K$  do
8:     if  $\tilde{\mathcal{F}}[r][h-1] > 0$ 
9:       Case1: if  $z_{B[r][h]}^1 = 1$  (only one crew is assigned to  $B[r][h]$ ):
10:         $\tilde{P}[r][h] = p_{B[r][h]}^1$ ,  $\tilde{\mathcal{F}}[r][h] = \tilde{\mathcal{F}}[r][h] = \tilde{\tau}[r][h] + p_{B[r][h]}^1$ ,
11:         $\tilde{\tau}[r][h+1] = \tilde{\mathcal{F}}[r][h] + C[B[r][h], B[r][h+1]]$ 
12:       end if
13:       Case2: if  $z_{B[r][h]}^1 = 1, l > 1$  (more than one crew is assigned to  $B[r][h]$ ) and  $B[r][h]$  is visited for
14:          the first time:
15:           $\tilde{P}[r][h] = p_{B[r][h]}^1$ ,  $\tilde{\mathcal{F}}[r][h] = \tilde{\tau}[r][h] + p_{B[r][h]}^1$ ,
16:           $\tilde{\tau}[r][h+1] = \tilde{\mathcal{F}}[r][h] + C[B[r][h], B[r][h+1]]$ 
17:       end if
18:       Case3: if  $z_{B[r][h]}^1 = 1, l > 1$  ( $B[r][h]$  is visited  $l - \tilde{l}$  times before):
19:          form  $\mathcal{X} = \{\tilde{\tau}[i][j] | B[i][j] = B[r][h]\}$ ,  $\mathcal{Y} = \{\tilde{\mathcal{F}}[i][j] | B[i][j] = B[r][h]\}$ , sorted non-increasing order
20:          Case3-1: if  $\min_{i=1,\dots,|\mathcal{X}|} \mathcal{X}[i] < \tilde{\tau}[r][h] < \max_{i=1,\dots,|\mathcal{Y}|} \mathcal{Y}[i]$ :
21:            for  $\forall B[i][j] = B[r][h]$ 
22:               $\tilde{\mathcal{F}}[i][j] = \max_{i=1,\dots,|\mathcal{X}|} \mathcal{X}[i] + \frac{(\lambda_{B[r][h]}^1 p_{B[r][h]}^1 - \sum_{h=2}^{l-1} \lambda_{B[r][h]}^1 (\mathcal{X}[l] - \mathcal{X}[l-1]))}{\lambda_{B[r][h]}^1}$ 
23:               $\tilde{P}[i][j] = \tilde{\mathcal{F}}[i][j] - \tilde{\tau}[i][j]$ 
24:              if  $\tilde{l} = 1$ :
25:                 $\tilde{\mathcal{F}}[i][j] = \tilde{\mathcal{F}}[i][j]$ 
26:              end if
27:            end for
28:          end if
29:          Case3-2: if  $\max_{i=1,\dots,|\mathcal{Y}|} \mathcal{Y}[i] < \tilde{\tau}[r][h]$ :
30:            Shift Procedure: if  $\exists B[r][\tilde{h}]$  where  $z_{B[r][\tilde{h}]}^1 = 1, \tilde{h} < h, \tilde{\tau}[r][\tilde{h}] + p_{B[r][\tilde{h}]}^1 < \max_{i=1,\dots,|\mathcal{X}|} \mathcal{X}[i]$ :
31:              Change the position of  $B[r][\tilde{h}]$  and  $B[r][h]$  in  $B$ 
32:              Go to Case3-1 (line 17)
33:              Go to Update procedure
34:            Delete Procedure: else Delete  $B[r][h]$  in  $B$ 
35:              Go to Update procedure
36:            end if
37:          end if
38:        end if
39:      end for
40:    end for
41:    Update Procedure: for  $h' = \tilde{h} + 1, \dots, |B[r]|$  :
42:      if  $\tilde{\tau}[r][h'] > 0$  :
43:         $\tilde{\tau}[r][h'] = \tilde{\mathcal{F}}[r][h' - 1] + C[B[r][h' - 1], B[r][h']]$ 
44:      end if
45:      if  $\tilde{\mathcal{F}}[r][h'] > 0$  :
46:         $\tilde{\mathcal{F}}[r][h'] = \tilde{\mathcal{F}}[r][h'] = \tilde{\tau}[r][h'] + \tilde{P}[r][h']$ 
47:        examine Case1, Case2, and Case3 for  $B[r][h' + 1]$ 
48:      end if
    end for

```

---

**4. Illustrative examples based on power grid transmission, water, and gas networks in Shelby County, TN**

To test the performance of the mathematical formulations and heuristic algorithm, we apply the restoration crew routing problem to realistic data sets representing three different infrastructure systems, including the electric power, water, and gas networks in Shelby County, Tennessee. Shelby County, which is located in the New Madrid Seismic Zone, is home to Memphis, a city with a population of over 650,000. Shown in Fig. 5, the power network is composed by 136 components (i.e., 60 nodes and 76 links), the water network by 120 components (i.e., 49 nodes and 71 links), and the gas network by 33 components (i.e., 16 nodes and 17 links) (González et al., 2016b). The combination of these three networks has a total number of 289 components (i.e., 125 nodes and 164 links). Based on González et al. (2016b), we consider a realistic earthquake scenario with epicenter at 35.3° N, and 90.3° W located 33 km northwest of the center of Memphis, including magnitudes within the range of  $M_w \in [6, 9]$ . On average, for the simulated earthquakes with  $M_w = 6$ , about 6.2% of all network components are destroyed, for  $M_w = 7$ , 9.3% are destroyed, for  $M_w = 8$ , 16.6% are destroyed, and for  $M_w = 9$ , 22.8% are destroyed. We distribute the disruptions among the components of the three networks randomly. Since each restoration



**Fig. 5.** Graphical representations of the (a) power grid transmission, (b) water, and (c) gas networks at a transmission level in Shelby County, TN (González et al., 2016b).

crew is accompanied with personnel and equipment, the increase in the number of restoration crews adds additional costs to the problem. The maximum number of crews working in each time period is 6, 5, and 4 for the power grid, water, and gas networks, respectively. The distance between each pair of disrupted locations is the shortest unblocked path (in miles) obtained through ArcGIS and Google Earth. Each of the disrupted infrastructure links may experience a certain level of damage and require a specific restoration time that depends on the number of assigned restorative crews, its level of damage, and other characteristics. After the occurrence of the disruption, crews should be dispatched from the depots. For each disruption scenario associated with each infrastructure network, we select 5, 10, and 15 potential locations for depots respectively. To simplify the calculation of the distance between a depot node and the location of disrupted links, the potential depot nodes are chosen from the nodes of each infrastructure network. In different scenarios, the depots are randomly chosen from the potential depots nodes. The traveling time between each pair of nodes  $c_{ij}$ ,  $(i, j) \in \bar{A}$  is equal the distance between  $i$  and  $j$  divided by the speed of restoration crew, 60 mph on average (Akbari and Salman 2017). We consider the restoration time horizon as 100 h or about four days (FEMA, 2017).

#### 4.1. Computational experiment

In this section, we present the computational results on 60 instances, where for each network we test five instances for each level of magnitude vibration, and the disruption links are randomly distributed through each network. The computational experiments for both mathematical formulations and heuristic are performed on an Intel Core™ i7-7500U CPU 2.90GHz (with 32GB RAM) using Gurobi 7.0.2 on Python 2.7.13. The outputs of the heuristics are compared with the exact solutions obtained by solving the Binary and Proportional Active restoration crew routing problems. For cases where the Gurobi Solver is not able to return the exact solution within the allowed time of one hour, we compare the lower bound obtained by [Algorithm 1](#) with the lower bounds and the best obtained solution found by Gurobi in the given time limit. While the Proportional Active model may not always be directly applicable to power, water, and gas networks, to provide relevant insights, we study the behavior of these networks under Proportional Active assumptions for illustrative purposes. Furthermore, the Proportional Active is also applicable in some cases where, along with the installation of main infrastructure networks (e.g., power lines), temporary and emergency lines are installed and used to meet a portion of demands. The model is also applicable in cases where the redundant components (e.g., powerline and transformers, water pipelines and pumps or gas lines) are installed to play the same role in parallel. Finally, we apply the Proportional Active model to all three infrastructure networks to evaluate the behavior of the model for small size problems (i.e., gas network), medium size (i.e., water network), and large size (i.e., power network) problems.

[Table 3](#) through [Table 5](#), which show the results for the power grid, water, and gas network instances in Shelby County, represent the results of Binary Active and Proportional Active restoration crew routing models and the initial solution obtained from the Relaxed-Restoration Crew Routing formulation and Initial Solution Preprocessing & Feasibility algorithm. For

**Table 3**

Percentage gap and solution time for the Relaxed-Based Initial Solution Algorithm, Binary Active, and Proportional Active restoration crew routing problem for the electric power network.

Ins. (1)	$M_w$ (2)	K (3)	Initial solution		Binary Active formulation				Proportional Active formulation			
			CPU(s) (4)	T (5)	Gap <sup>G*</sup> (6) (%)	Gap <sup>Z</sup> (7) (%)	CPU(s) (8)	T (9)	Gap <sup>G*</sup> (10) (%)	Gap <sup>Z</sup> (11) (%)	CPU(s) (12)	T (13)
1	6	2	2.9	96	11.5	19.9	1800	70	3.86	15.7	1800	85
2		3	2.4	67	7.45	32.9	1800	43	0.9	58.5	1800	35
3		4	2.7	47	4.5	16.4	1800	41	0.1	35.3	1800	28
4		5	5.8	47	5.55	19.1	1800	35	0.1	26.5	1800	28
5	7	2	2.0	83	11	36.8	1800	69	2.73	12.4	1800	66
6		3	6.3	62	9.1	31.2	1800	53	0.4	45.8	1800	37
7		4	2.3	60	6.67	37.7	1800	44	0.7	49.4	1800	34
8		5	3.9	57	5.85	37.2	1800	39	0.1	45.5	1800	26
9	8	4	2.3	111	–	–	1800	–	–	–	1800	–
10		5	4.4	103	–	–	1800	–	–	–	1800	–
11		6	4.9	95	23	27.5	1800	86	14.10	21.8	1800	85
12		7	2.9	83	18.6	16.1	1800	74	12.3	15.5	1800	72
13	9	4	4.2	75	–	–	1800	–	–	–	1800	–
14		5	2.6	72	–	–	1800	–	–	–	1800	–
15		6	6.8	68	–	–	1800	–	–	–	1800	–
16		7	3.4	65	–	–	1800	–	–	–	1800	–

**Table 4**

Percentage gap and solution time for the Relaxed-Based Initial Solution Algorithm, Binary Active, and Proportional Active restoration crew routing problem for the water network.

Ins. (1)	$M_w$ (2)	K (3)	Initial solution		Binary Active formulation				Proportional Active formulation			
			CPU(s) (4)	T (5)	Gap <sup>G*</sup> (6) (%)	Gap <sup>Z</sup> (7) (%)	CPU(s) (8)	T (9)	Gap <sup>G*</sup> (10) (%)	Gap <sup>Z</sup> (11) (%)	CPU(s) (12)	T (13)
1	6	2	6.05	77	3.45	18.4	1800	65	2.37	31.5	1800	49
2		3	3.72	42	2.67	12.7	1800	36	2.32	1.1	1800	40
3		4	4.65	32	3.08	–2.7	1800	35	2.43	8.1	1800	29
4		5	7.71	22	2.45	–2.4	1800	32	1.54	–6.1	1800	25
5	7	2	3.44	89	3.42	3.8	1800	76	3.67	17.2	1800	59
6		3	7.24	68	3.05	27	1800	47	3.73	20.0	1800	47
7		4	6.78	54	3.71	38.2	1800	35	3.52	8.5	1800	46
8		5	7.88	40	3.93	18.3	1800	34	4.23	–3.7	1800	42
9	8	2	3.77	99	6.64	5.2	1800	95	5.5	26.8	1800	74
10		3	3.46	71	6.24	5.1	1800	53	5.76	7.8	1800	63
11		4	3.48	59	6.43	4.3	1800	51	5.75	2.2	1800	55
12		5	4.90	51	6.34	1.8	1800	48	5.43	3.2	1800	45
13	9	3	7.84	90	12	3.3	1800	79	9.85	8.1	1800	77
14		4	6.69	81	23	–3.5	1800	85	14.56	45.3	1800	54
15		5	7.37	69	13.5	8.6	1800	60	9.5	25.9	1800	38
16		6	6.05	43	12.7	1.5	1800	41	9.6	24.2	1800	30

each infrastructure network, the level of disruption (i.e., earthquake magnitude) and the number of restoration crews are shown in the first and second columns. The fourth and fifth columns provide the CPU time required for the computation of the initial solution and the makespan or the restoration time, provided by the initial solution,  $T$ , respectively. The optimality gaps for each scenario obtained from Gurobi optimization,  $Gap^{G^*}$ , are shown for Binary and Proportional formulations (columns 6 and 10, respectively). In the Gurobi optimization process, first a relaxed formulation is driven from the original model and a lower bound is determined for the optimization process. Then using the Branch and Bound algorithm, the best bound is obtained through the CPU limitation time.  $Gap^{G^*}$  measures how far the best bound is from the relaxed formulation obtained by Gurobi. The percentages of difference between improvement in network resilience associated with the initial solution (obtained from the heuristic algorithm) and the optimization model solved by Gurobi,  $Gap^Z$ , is calculated in Eq. (41). Results are shown for both the Binary and Proportional formulations (columns 7 and 11, respectively). Tables 3–5 also show the CPU times required for the computation of initial solution, Binary and Proportional formulations (columns 8 and 12, respectively), as well as the Makespan related to the two formulations (columns 9 and 13, respectively).

$$Gap^Z = \frac{\sum_{t \in T} \vartheta_\varphi(t|e^j)_{\text{initial}} - \max \sum_{t \in T} \vartheta_\varphi(t|e^j)}{\max \sum_{t \in T} \vartheta_\varphi(t|e^j)} \quad (41)$$

According to Table 3 through Table 5, all instances under the disruption scenarios with  $M_w=6$ , 7, and 8 were solved within a reasonable amount of time (i.e., 1800s) and with an average optimality gap of 4.5%. The variation of the optimality gap is larger for the disruption scenario with  $M_w=9$ , where the minimum, maximum, and average gaps are 3.03%, 23%, and

**Table 5**

Percentage gap and solution time for the Relaxed-Based Initial Solution Algorithm, Binary Active, and Proportional Active restoration crew routing problem for the gas network.

Ins. (1)	$M_w$ (2)	K (3)	Initial solution		Binary Active formulation				Proportional Active formulation			
			CPU(s) (4)	T (5)	Gap <sup>G</sup> (%) (6)	Gap <sup>Z</sup> (%) (7)	CPU(s) (8)	T (9)	Gap <sup>G</sup> (%) (10)	Gap <sup>Z</sup> (%) (11)	CPU(s) (12)	T (13)
1	6	2	4.22	27	0	0	25	27	0	6.5	18	9
2		3	3.57	21	0	0	45	21	0	20.5	32	10
4	7	2	1.13	11	0.03	0	1800	51	0	1.6	38	21
5		3	4.72	22	0.06	0	1800	27	0	28.8	41	11
7	8	2	4.54	38	1.93	0	1800	59	0.3	26.3	1800	42
8		3	1.97	32	2.32	0	1800	55	0.95	22.6	1800	21
10	9	4	3.68	28	2.47	0	1800	43	1.56	10.4	1800	20
11		3	2.27	36	3.34	0	1800	59	2.42	11.1	1800	32
12		4	2.23	31	3.03	0	1800	44	2.65	37.8	1800	25

7.89%, respectively, for the Binary formulation, and 1.56%, 14.5%, and 5.93%, respectively, for the Proportional formulation. For water and gas networks, the initial solution demonstrates the efficacy of the [Algorithm 2](#) in providing a strong initial feasible solution for any solution improvement algorithm for the Restoration Routing Problems. It is also shown that in some cases the initial solution algorithm provides the better upper bound for the Binary and Proportional formulations (e.g., instances 1, 2, and 8–12 in gas network and instances 4, 5, and 14 for water network).

For the power network instances, as an example for large scale problems, [Algorithm 2](#) reaches to a reasonable initial feasible solution, with the average optimality gap of 28.9% for Binary formulation and 36% for Proportional formulation, in a considerably short time of 3.5 seconds on average. For disruption scenario with  $M_w=9$ , as the size of the instances increase dramatically, the Binary and Proportional formulations fall short of solving the power network instances with the disruptions level of 22.8%. However, the algorithm obtained the initial solution in a considerably short time of 4.5 s on average. The exact formulations fall short in solving instances 10 and 11 simply because the number of restoration crews were not adequate to restore the whole network in the given time horizon (i.e.,  $T=100$ ).

Results suggest that there should be a balance between the number of restoration crews and the number of disrupted links in the infrastructure network to obtain the minimum optimality gap in a given solution time. For example, for the disruption scenario with  $M_w=6$ , using three crews to restore the water network and four crews to restore the power network results in the minimum optimality gap, and with  $M_w=7$  using three and five crews for restoring the water and power networks, respectively, results in the minimum optimality gap. The minimum optimality gap does not guarantee a minimum restoration time horizon, but rather it assures the best upper bound in a limited solution time.

Comparing the solutions with and without the consideration of routing time, we see that more disrupted components receive restoration services from multiple crews. In the former category, restoration crews are more scattered throughout the network, and there is a smaller number of disrupted components that receive restoration services from multiple crews because time spent traveling in the network is being accounted for. In the latter category which does not consider routing time, sets of restoration tasks assigned to crews may form direct cycles, which means, for example, two crews would incorrectly be assigned at two different geographical locations at the same time. In the latter category when routing time is not considered, the crews assigned to each component are assumed to start the restoration task at the same time. However, this might not always be the case for the former category as the difference in traveling time results in different crew arrival times. The difference of arrival time affects how the disrupted components are scheduled to each crew. Finally, regardless of the consideration of routing time, depending on the distance among the depots and the distribution of disruptions over the network (whether highly scattered or spatially focused), the number of disrupted components that are restored by multiple crews and the average number of crews assigned to them changes. As such, the closer the depots are together and/or the closer are the disrupted components, the number of disrupted components visited by multiple crews increases as well as the average number of crews assigned to each of those components.

We examine the effect of the weight  $w_i$ ,  $i \in N_-$ , for weighting the importance of demand nodes, where, for example, demand nodes located in highly populated areas could have a higher priority for restoration relative to other demand nodes. To incorporate  $w_i$  in both mathematical formulations, we update the objective function with [Eq. \(42\)](#).

$$\mathfrak{R}_\varphi(t|e) = \frac{\sum_{i \in N_-} w_i \varphi_{it} - \sum_{i \in N_-} w_i \varphi_{it_d}}{\sum_{i \in N_-} w_i \varphi_{it_e} - \sum_{i \in N_-} w_i \varphi_{it_d}} \quad (42)$$

For the relaxed formulation, we use three measures of importance (i.e.,  $I_{\text{MFCount}}$ ,  $I_{\text{flow}}$ ,  $I_{\text{FCR}}$ ) and categorize the important links to three clusters, one with IMs less than 0.3, one with IMs between 0.3 and 0.6, and one with IMs greater than 0.6. [Table 6](#) compares the performance of [Algorithm 2](#) and the Binary Active RCR when we incorporate demand nodes with priority weight  $w_i$ ,  $i \in N_-$ . To compare all results obtained from the Restoration Crew Routing formulation and [Algorithm 2](#), we examined the results under the disruption scenarios with  $M_w=6$ , 7, and 8. The type of the infrastructure network, the instance number, the magnitude of the earthquake, and the number of the restoration crews are shown in the first four columns of [Table 6](#). The effect of employing different importance measures in [Eq. \(40\)](#) on the CPU time and the required

**Table 6**

Percentage gap and solution time for Relaxed-Based Initial Solution Algorithm, Binary Active restoration crew routing problem under the employment of importance measures  $I_{MFCount}$ ,  $I_{FCR}$ , and  $I_{flow}$ .

	Ins.	$M_w$	K	$I_{MFCount}$		$I_{FCR}$		$I_{flow}$		Binary Active formulation							
				(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	Gap <sup>G*</sup> (%)	(10)	CPU(s)	T (11)	Gap <sup>Z</sup> (%)
Power	1	6	2	4.80	97	4.23	99	1.32	100	5.4	1800	92	6.7	9.5	18.2		
	2		3	3.52	68	5.24	69	4.40	70	2.2	1800	39.6	41	37.1	40.7		
	3		4	4.67	49	4.63	49	3.77	51	1.8	1800	34	27.5	27.5	29.5		
	4		5	4.96	49	2.93	49	5.64	49	2.1	1800	34	27.5	27.5	27.5		
	5	7	2	3.40	84	2.22	86	2.91	86	3.7	1800	73	13.6	15.0	15.0		
	6		3	6.36	64	3.60	65	2.09	64	2.1	1800	42	24.0	26.2	24.0		
	7		4	3.83	61	2.23	61	3.72	64	2.4	1800	38	35.3	35.3	37.3		
	8		5	3.94	59	5.53	59	5.97	59	1.8	1800	33	32.0	32.3	32.3		
	9	8	2	6.96	115	6.43	115	6.82	115	–	1800	–	–	–	–		
	10		3	6.67	112	6.93	113	6.29	112	–	1800	–	–	–	–		
	11		4	6.83	105	6.29	108	6.02	109	26	1800	95	15.6	15.8	15.8		
	12		5	6.71	92	6.92	94	6.12	94	21.3	1800	83	7.5	7.8	7.8		
	13	6	2	6.73	79	6.40	79	4.57	79	3.6	1800	54	32.7	32.7	32.7		
	14		3	2.76	44	6.46	45	6.69	43	3.9	1800	44	0	0.3	–0.3		
	15		4	4.11	33	1.17	33	3.59	34	4.3	1800	37	–2.8	–2.8	–2.5		
	16		5	4.28	24	3.34	23	1.01	25	2.6	1800	27	–2.9	–3.3	–1.4		
	17	7	2	2.62	90	1.60	90	2.06	93	5.1	1800	64	28.2	28.2	31.2		
	18		3	2.78	69	4.33	70	5.31	69	4.7	1800	52	16.3	19.3	16.3		
	19		4	6.99	55	5.67	55	3.43	56	5.1	1800	53	1.6	1.6	2.1		
	20		5	3.01	42	5.94	43	5.53	42	5.4	1800	47	–5.6	–6.6	–5.6		
	21	8	2	4.55	101	5.1	103	5.43	103	–	1800	–	–	–	–		
	22		3	5.1	67	4.91	69	5.21	68	5.41	1800	59	16.4	17.1	17.5		
	23		4	4.32	65	4.55	65	4.58	65	5.38	1800	55	19.3	19.3	19.3		
	24		5	4.68	75	4.69	75	4.96	76	6.59	1800	51	27.4	28.9	27.4		
Gas	25	6	2	3.64	29	2.91	29	3.48	29	0	22	14	37.3	37.3	37.3		
	26		3	3.99	22	3.39	22	3.70	22	0	45	15	28.9	28.9	28.9		
	27	7	2	3.50	12	2.10	14	2.93	14	0	49	28	–33.8	–32.8	–32.8		
	28		3	2.98	24	3.01	25	2.37	24	0	63	15	29.0	30.8	29.0		
	29	8	2	3.1	25	2.36	27	2.41	27	0	65	24	3.1	4.5	4.5		
	30		3	3.4	46	2.65	47	2.58	50	0	67	43	12.9	13.6	16.8		

restoration time is shown in columns 4 and 5 for  $I_{MFCount}$ , columns 6 and 7 for  $I_{FCR}$ , and columns 8 and 9, for  $I_{flow}$ . For each scenario the optimality gap,  $Gap^G$ , the CPU time, and the required restoration time obtained from Gurobi optimization are shown in columns 10 through 12. Finally, the percentage of difference between improvement in the network resilience measure associated with the initial solution for  $I_{MFCount}$ ,  $I_{FCR}$ , and  $I_{flow}$  and the optimization model,  $Gap^Z$ , is shown the final three columns.

According to Table 6, the Binary Active formulation solved all instances in a reasonable amount of time (i.e., 1800 s) and with an average optimality gap of 2.81%. Employing each of the three importance measures, all prioritized demand nodes are satisfied before others. In cases where we suppose to consider the importance of some demand nodes over the others, scaled  $w_i, i \in N_-$ , the implementation of all three importance measures provides strong initial solutions for any of the solution improvement algorithms. The average, maximum, and minimum of  $Gap^Z$  related to  $I_{MFCount}$  are 16.2%, 37.3%, and –33.8%, for  $I_{FCR}$  are 16.7%, 41%, and –32.8%, and for  $I_{flow}$  are 17.4%, 40.7%, and –32.8%. Among the importance measures the implementation of  $I_{MFCount}$  results in full network resilience in less required restoration time. This is because, regardless of the percentage of the network flow a link carries, the implementation of  $I_{MFCount}$  finds links that are shared in the maximum number of source-target paths. Therefore, restoring more important links, measured by the employment of  $I_{MFCount}$ , bring more paths into activation as well as satisfying the prioritized demand nodes contained on the paths. On the other hand, the implementation of  $I_{flow}$  and  $I_{FCR}$  leads the relaxed formulation to focus on the links carrying the highest percentage of flow relative to total network flow and their defined capacity, respectively, which may not come from many source-target paths. As such, the choice of importance measure is an important consideration in finding a good solution.

Depending on the magnitude of the disruption scenario, and the accessibility to each disrupted component (i.e., the ratio of the restoration time, related to each disrupted component, to its traveling time to other components and depots), there is a certain number of restoration crews,  $k^*$ , for which the results obtained from the Binary and Proportional formulations represent the maximum number of disrupted components that receive restoration services from more than one crew. Assigning more restoration crews than  $k^*$  may result in a more scattered routing network, where the length of the route assigned to each restoration crew may decrease, yet a smaller number of disrupted components may be assigned to more than one restoration crew.

To illustrate this behavior, Fig. 6 indicates different scheduling and routing patterns obtained from Binary Active formulation, and Algorithm 2. These methods are studied under the disruption scenario with magnitude  $M_w=9$  for the water

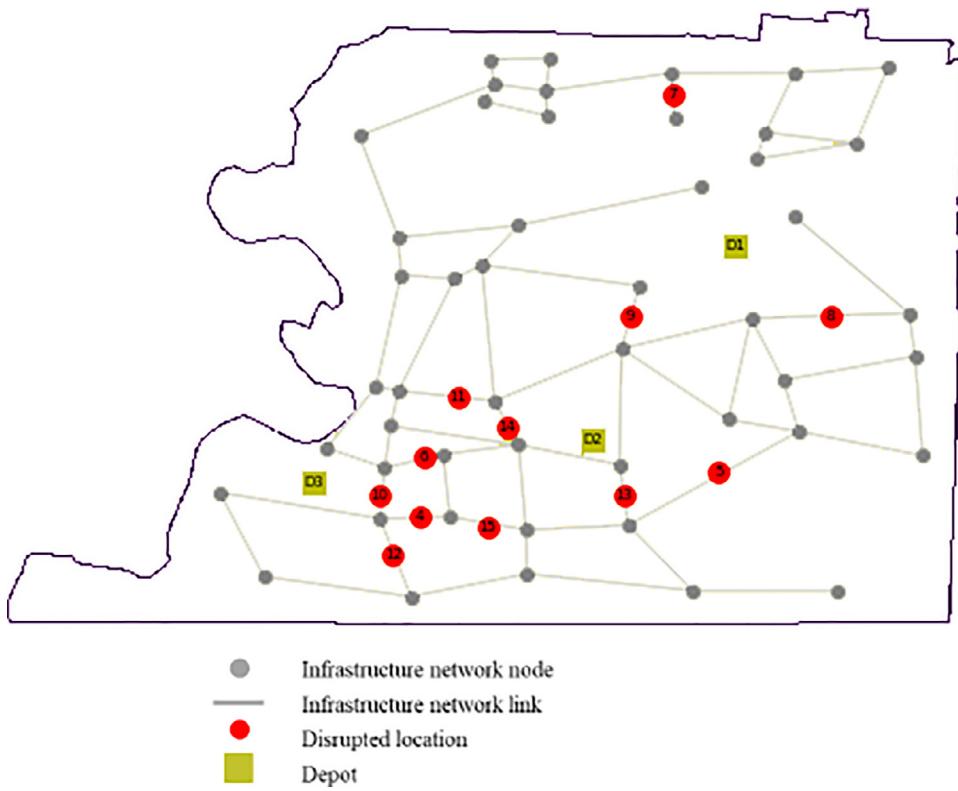


Fig. 6. The water network under the disruption scenario with magnitude  $M_w=9$ .

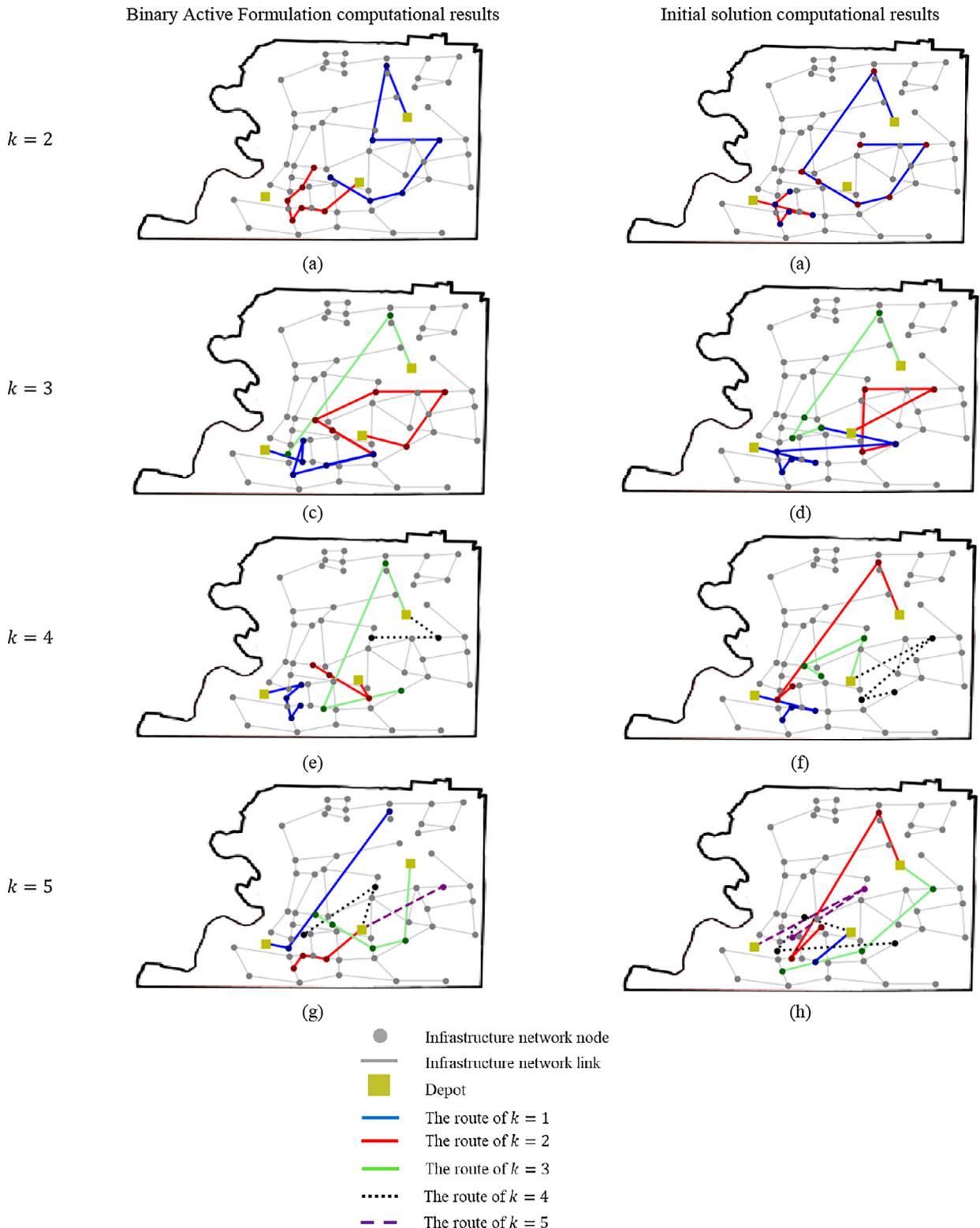
network. In Fig. 6, the rectangular nodes represent the available depots, and the numbered nodes represent the disrupted locations in the network after the occurrence of a disruptive event.

According to Fig. 7, for the Binary Active formulation, the maximum number of disrupted components which receive restoration services from more than one crew obtained when we incorporate three restoration crews. For the proposed algorithm, assigning three restoration crews to the disrupted network results in the maximum number of disrupted components scheduled to more than one crew. Accordingly, Considering Binary formulation, for power network, we may incorporate  $k^*=3, 4$  and  $3$ , under the disruption scenarios with magnitudes  $M_w=6, 7$  and  $8$ , respectively, to see the maximum number of disrupted components scheduled to more than one crew. For water network the number would be  $k^*=2, 2, 3$ , and  $3$ , under the disruption scenario with magnitudes  $M_w=6, 7, 8$ , and  $9$ , respectively. Finally, for the gas network, this number would be  $k^*=2, 2, 3$ , and  $4$ , under the disruption scenario with magnitudes  $M_w=6, 7, 8$ , and  $9$ , respectively, for the gas network. Considering the proposed algorithm, for power network, this number would be  $k^*=3, 4, 3$  and  $6$ , under the disruption scenario with magnitudes  $M_w=6, 7, 8$  and  $9$ , respectively. For water network, this number would be  $k^*=3, 3, 3$ , and  $3$ , under the disruption scenario with magnitudes  $M_w=6, 7, 8$ , and  $9$ , respectively. Finally, for the gas network,  $k^*=2, 2, 3$ , and  $3$ , under the disruption scenarios with magnitude  $M_w=6, 7, 8$ , and  $9$  respectively.

In the proposed relaxed formulation, the average number of crews assigned to each disrupted component is greater than the average number of assigned crews in the solution obtained from Binary and Proportional Active formulations. Note that although this feature proposes great initial solutions for heuristic algorithms to find near optimal solutions for restoration routing problems, to solve the timing conflicts, [Algorithm 2](#) has to change some routes and restoration sequences that may result in solutions with more than 10% of optimality gap from the solutions obtained from Binary and Proportional Active formulations.

## 5. Concluding remarks

Restoration capacity enhancement problems are often based on idealized assumptions (e.g., eliminating routing problems in restoration process, neglecting timing conflicts, considering fixed number of assigned crews), that may propose assumptions that may result in models that cannot be used in realistic contexts. With the implementation of routing among disrupted network components, we show that there is a considerable difference between these models and models that focus only on basic infrastructure network restoration. In other words, the basic infrastructure network restoration models may result in disrupted components schedules and sequences that are not applicable in realistic contexts, which must con-



**Fig. 7.** The computational results of water network under the disruption scenario with magnitude  $M_w = 9$ . (a) Binary Active model with two restoration crews, (b) initial solution using two restoration crews, (c) Binary Active model with three restoration crews, (d) initial solution using three restoration crews, (e) Binary Active model with four restoration crews, (f) initial solution using four restoration crews, (g) Binary Active model with five restoration crews, and (g) initial solution using five restoration crews.

sider traveling time of each restoration crew traveling time between each two disrupted locations it is assigned to, as the two or more restoration crews have to be in more than one location at the same time (i.e., timing conflicts). Also, they do not consider the difference in the arrival of time of each assigned crew to a disrupted location and the effects it has on the restoration rate of that corresponding component in each time period.

In this paper, we reinforced the applicability of restoration capacity problems to the real-world case studies by: (i) integrating routing and restoration problems by formulating Restoration Crew Routing problems, (ii) implementing dynamic restoration rate, where an idle restoration crew can join other crews working on a disrupted link and accelerate the remaining restoration process. Two model formulations are proposed: (i) the Binary Active Restoration Crew Routing model, in which disrupted links cannot play a role in network performance unless they are restored completely (e.g., railway network), and (ii) Proportional Active Restoration Crew Routing model, in which partially restored links are proportionally functional in the network, and as their restoration process progresses, their functionality in the network increases (e.g., the physical structure of internet networks).

Additionally, this paper presents a new algorithm to obtain the best initial solution for the infrastructure network restorative capacity enhancement problem. We first introduce a relaxed formulation of the proposed routing problems, which does not consider the arrival time of restoration crews to each disrupted component and their effects on the restoration rate of that component in each time period. Then, a cycle elimination algorithm is employed to solve all timing conflicts and bring the routes of restoration crew into synchronization. Finally, the Initial Solution Preprocessing and Feasibility Algorithm ([Algorithm 2](#)) calculates the routing time of each restoration crew and solve the timing conflicts caused by any restoration crew arrives to a disrupted link after its restoration process is completed. Using instances derived from real-life data from power grid, water, and gas networks in Shelby County, TN.

The computational results prove the efficacy of both Binary Active and Proportional Active formulations, especially in small to medium scale problems, by showing the small optimality gap for relatively small sized instances. The initial solution obtained from [Algorithm 2](#) are compared with the best upper bound obtained from original formulations. For the disruption scenarios with  $M_w = 6, 7$ , and  $8$  the optimality gap,  $Gap^2$ , about 4.5% on average, shows the credibility of the initial solution obtained from the heuristic algorithm. Decision makers are provided with insights into how the restoration tasks would be distributed through the network over the time and to what extent those tasks are centralized over the disrupted components. The demonstrated credibility of the heuristic algorithm provides a good approximation about the restoration process, such as how the choice of depots with different geographical locations affects the distribution of restoration tasks through the network, particularly under the large-scale disruption scenarios where both Binary Active and Proportional Model formulations fall short in finding the best bound in a limited time. We further can use the obtained initial solution in heuristic algorithm to find optimal or near optimal solution in a considerably short amount of time. To incorporate the prioritization of some demand nodes over the others (scaled  $w_i, i \in N_-$ ), we introduce a variation of the relaxed formulation that prioritizes the restoration process of disrupted links that play an important role in satisfying prioritized demand nodes. Given this strategy, decision makers can analyze the effects of prioritizing certain demand nodes on the restoration process from different aspects (e.g., prioritizing the locations near the hospitals, more populated areas, certain vulnerable communities), thus striking a balance between the recovery of the entire network and the priority of demands. The results in [Table 6](#) suggest the efficiency of the proposed variation in the relaxed formulation in serving the prioritized demand nodes before others. It also emphasizes that the performance of the proposed algorithm is aligned with the minimizing the restoration horizon as well as serving the prioritized demand nodes when we implement  $I_{MFCount}$  as the measure of importance.

An important (and realistic) direction for future work is to consider the effects of the disruptions (and posterior recovery) on the routing network itself and analyzing the behavior of both infrastructure and routing networks in terms of sharing restoration crews, rerouting, and restoration interdependencies. Another relevant direction in this area is to consider multiple restoration tasks associated with each component. In this case, crews that were assigned to a given disrupted component may finish their restoration tasks earlier than others working on the same component, and in consequence may leave before the full restoration process of the component is completed.

## Acknowledgment

This work was supported in part by the [National Science Foundation](#) through award [1635813](#).

## Appendix A

### A-1

**Proof.** Without loss of generality, we assume that no rerouting action takes place in the infrastructure network after the occurrence of a disruptive event. Hence, the time when the infrastructure network reaches to fully operational state is the same time it attains full recovery. Recall that, we can state that the purpose of original restoration model is to maximize the infrastructure network by minimizing the maximum routing time. We prove that statement by two lemmas.

**Lemma 1.** If  $\tilde{p}_{(i,j)}^l$ ,  $l = 1, \dots, L$  represents the restoration time of each disrupted link  $(i, j) \in A'$ , defined in the independent crew routing problem, and  $p_{(i,j)}^l$ ,  $l = 1, \dots, L$  represents the restoration time of each disrupted link associated with the relaxed formulation, then we have that

$$\tilde{p}_{(i,j)}^l \leq p_{(i,j)}^l \quad (A-1)$$

**Proof.** Table 7 gives an instance that shows the impact of each crew on the processing time of each disrupted link  $(i, j) \in A'$ . Each time a new crew  $k = 1, \dots, K$  arrives, it accelerates the restoration rate associated with the remaining disruption. Therefore, the portion of disruptions processed prior to the arrival of crew  $k$  cannot be affected by that crew. We assume that no crew visits a disrupted link after it has been recovered.

According to Table 7, we have,  $\tilde{p}_{(i,j)}^l = (\tilde{t}_{(i,j)}^l - \tilde{t}_{(i,j)}^1) + \frac{\lambda_{(i,j)}^1 p_{(i,j)}^1 - \sum_{h=2}^l \lambda_{(i,j)}^{h-1} (\tilde{t}_{(i,j)}^h - \tilde{t}_{(i,j)}^{h-1})}{\lambda_{(i,j)}^l}$ , and we also have  $p_{(i,j)}^l = \frac{\lambda_{(i,j)}^1 p_{(i,j)}^1}{\lambda_{(i,j)}^l}$ .

Through the method of proof by contradiction, suppose that

$$(\tilde{t}_{(i,j)}^l - \tilde{t}_{(i,j)}^1) + \frac{\lambda_{(i,j)}^1 p_{(i,j)}^1 - \sum_{h=2}^l \lambda_{(i,j)}^{h-1} (\tilde{t}_{(i,j)}^h - \tilde{t}_{(i,j)}^{h-1})}{\lambda_{(i,j)}^l} \leq \frac{\lambda_{(i,j)}^1 p_{(i,j)}^1}{\lambda_{(i,j)}^l} \quad (A-2)$$

$$\lambda_{(i,j)}^l (\tilde{t}_{(i,j)}^l - \tilde{t}_{(i,j)}^1) + \lambda_{(i,j)}^1 p_{(i,j)}^1 - \sum_{h=2}^l \lambda_{(i,j)}^{h-1} (\tilde{t}_{(i,j)}^h - \tilde{t}_{(i,j)}^{h-1}) \leq \lambda_{(i,j)}^1 p_{(i,j)}^1 \quad (A-3)$$

$$\sum_{h=2}^l \lambda_{(i,j)}^l (\tilde{t}_{(i,j)}^h - \tilde{t}_{(i,j)}^{h-1}) \leq \sum_{h=2}^l \lambda_{(i,j)}^{h-1} (\tilde{t}_{(i,j)}^h - \tilde{t}_{(i,j)}^{h-1}) \quad (A-4)$$

$$l \lambda_{(i,j)}^l \leq \lambda_{(i,j)}^1 + \lambda_{(i,j)}^2 + \dots + \lambda_{(i,j)}^l \quad (A-5)$$

Thus, we have the contradiction on (A-2) as it is clear that  $\lambda_{(i,j)}^l \geq \lambda_{(i,j)}^{\bar{i}}$ ,  $\bar{i} = 1, \dots, l-1$ . Therefore, (A-1) is proved. As the traveling times among disrupted links remain unchanged, we conclude that

$$Z_R^* \leq Z_R(S_{B/PAP}^*) \quad (A-6)$$

A-2

**Proof.** Building upon Akbari and Salman (2017), we know that the output of the Relaxed RCRP is a set of restoration routes which may share one node or more with one another. In the worst case, using cycle elimination algorithm results in traveling times related to some crews which are prolonged more than the time they save when they join to other crews on the restoration process of disrupted nodes (i.e.,  $\max_{k=1, \dots, K} \tilde{F}_{kn_k}$  (Relaxed RCRP)  $\leq \max_{k=1, \dots, K} \tilde{F}_{kn_k}$  (feasibility algorithm)). We also defined a version of Binary and Proportional restoration crew routing formulation, referred to as the Modified formulation, in which we substitute multiple crews to one crew. This lone crew starts its route from a depot, serves a number of nodes, may restore some of them completely and some others partially, returns to the same depot or a different one in zero time, and then again starts a new route. This procedure repeats until all disrupted nodes are fully recovered. Without loss of generality, we consider that no rerouting is possible in the network after disruption. Hence, to attain fully restored network resilience ( $\mathcal{R}_\phi(t|e^j)$ ), we need to restore all disrupted links. In Modified formulation, the crew comes back to partially restored nodes and restores all remaining disruptions or a proportion of them with restoration rate  $\lambda_{\bar{i}}^l$ , where  $l$  is the number of times the crew has visited node  $\bar{i} \in V_{A'}$  up to then. The maximum number of times the restoration crew starts a new route is equal to  $K$ . The optimal solution of the Modified formulation is equal to  $\sum_{k=1}^K \tilde{F}_{kn_k}$ . It is clear that

$\max_{k=1, \dots, K} \tilde{F}_{kn_k}$  (Algorithm 2)  $\leq \sum_{k=1}^K \tilde{F}_{kn_k}$ . Further, we can see that  $\frac{\sum_{k=1}^K \tilde{F}_{kn_k}}{K} \leq \max_{k=1, \dots, K} \tilde{F}_{kn_k}$  (Relaxed formulation). We then obtain  $\max_{k=1, \dots, K} \tilde{F}_{kn_k}$  (Relaxed formulation)  $\leq \max_{k=1, \dots, K} \tilde{F}_{kn_k}$  (Algorithm 2)  $\leq K \cdot \max_{k=1, \dots, K} \tilde{F}_{kn_k}$  (Relaxed formulation).

**Table 7**  
The processing time of disrupted link  $(i, j)$  for independent crew routing restoration problem.

$h^{th}$ crew	Assigned crews	Acceleration in remained restoration process	Processing time after the arrival of $h^{th}$ crew
$h=1$	$k=1$	–	$p_{(i,j)}^1$
$h=2$	$k=1, 2$	$\hat{p}_{(i,j)}^2 = \frac{\lambda_{(i,j)}^2 (p_{(i,j)}^1 - (\tilde{t}_{(i,j)}^2 - \tilde{t}_{(i,j)}^1))}{\lambda_{(i,j)}^2}$	$\hat{p}_{(i,j)}^2 = (\tilde{t}_{(i,j)}^2 - \tilde{t}_{(i,j)}^1) + \hat{p}_{(i,j)}^2$
$h=3$	$k=1, 2, 3$	$\hat{p}_{(i,j)}^3 = \frac{\lambda_{(i,j)}^3 (\hat{p}_{(i,j)}^2 - (\tilde{t}_{(i,j)}^3 - \tilde{t}_{(i,j)}^2))}{\lambda_{(i,j)}^3}$	$\hat{p}_{(i,j)}^3 = (\tilde{t}_{(i,j)}^3 - \tilde{t}_{(i,j)}^2) + \hat{p}_{(i,j)}^3$
⋮	⋮	⋮	⋮
$h=l$	$k=1, 2, \dots, l$	$\hat{p}_{(i,j)}^l = \frac{\lambda_{(i,j)}^{l-1} (\hat{p}_{(i,j)}^{l-1} - (\tilde{t}_{(i,j)}^l - \tilde{t}_{(i,j)}^{l-1}))}{\lambda_{(i,j)}^{l-1}}$	$\hat{p}_{(i,j)}^l = (\tilde{t}_{(i,j)}^l - \tilde{t}_{(i,j)}^{l-1}) + \hat{p}_{(i,j)}^l$

## References

- Akbari, V., Salman, F.S., 2017. Multi-vehicle synchronized arc routing problem to restore post-disaster network connectivity. *Eur. J. Oper. Res.* 257 (2), 625–640.
- Almoghathawi, Y., K. Barker, and L. Albert. 2017. Resilience-driven restoration model for interdependent infrastructure networks. *Reliab. Eng. Syst. Saf.* Manuscript submitted for publication.
- American Society of Civil Engineers (ASCE), 2017. Report Card for America's Infrastructure <http://www.infrastructurereportcard.org/oklahoma/oklahoma-overview/>.
- Barker, K., Lambert, J.H., Zobel, C.W., Tapia, A.H., Ramirez-Marquez, J.E., Albert, L., Nicholson, C.D., Caragea, C., 2017. Defining resilience analytics for interdependent cyber-physical-social networks. *Sustainable Resilient Infrastruct.* 2 (2), 59–67.
- Barker, K., Ramirez-Marquez, J.E., Rocco, C.M., 2013. Resilience-based network component importance measures. *Reliab. Eng. Syst. Saf.* 117, 89–97.
- Baroud, H., Barker, K., Ramirez-Marquez, J.E., Rocco, C.M., 2014. Importance measures for inland waterway network resilience. *Transp. Res. Part E* 62, 55–67.
- Bienstock, D., Mattia, S., 2007. Using mixed-integer programming to solve power grid blackout problems. *Discrete Optim.* 4 (1), 115–141.
- Çelik, M., Ergün, Ö., Keskinocak, P., 2015. The post-disaster debris clearance problem under incomplete information. *Oper. Res.* 63 (1), 65–85.
- Çelik, M., 2017. Network restoration and recovery in humanitarian operations: framework, literature review, and research directions. *Surv. Oper. Res. Manage. Sci.* 21, 47–61.
- Chapman, A., González, A.D., Mesbahi, M., Dueñas-Osorio, L., 2017. Data-guided control: clustering, graph products, and decentralized control. In: 56th IEEE Conference on Decision and Control (CDC2017). Melbourne, Australia, pp. 1–8.
- CNBC Economy. 2017. <https://www.cnbc.com/2017/09/11/harvey-and-irma-economic-hit-could-total-200-billion-moodys.html>
- Commission to Rebuild Texas, 2017. Issue 3. Office of Texas Governor. Greg Abbott <https://gov.texas.gov/news/post/commission-to-rebuild-texas-update-issue-3>.
- D'Ambrosio, C., Lodi, A., Wiese, S., Bragalli, C., 2015. Mathematical programming techniques in water network optimization. *Eur. J. Oper. Res.* 243 (3), 774–788.
- Faturechi, Miller-Hooks, R.E., 2014. Travel time resilience of roadway networks under disaster. *Transp. Res. Part B* 70, 47–64.
- FEMA, 2017. Hurricane Harvey Snapshot <https://www.fema.gov/news-release/2017/09/03/hurricane-harvey-snapshot#>.
- González, A.D., Chapman, A., Dueñas-Osorio, L., Mesbahi, M., 2017. Efficient infrastructure restoration strategies using the recovery operator. *Comput. Aided Civil Infrastruct. Eng.* 32 (12), 991–1006.
- González, A.D., Dueñas-Osorio, L., Medaglia, A.L., Sánchez-Silva, M., et al., 2016a. The time-dependent interdependent network design problem (td-INDP) and the evaluation of multi-system recovery strategies in polynomial time. In: Huang, H.W., et al. (Eds.), The 6th Asian-Pacific Symposium on Structural Reliability and its Applications. Shanghai, China, pp. 544–550.
- González, A.D., Dueñas-Osorio, L., Sánchez-Silva, M., Medaglia, A.L., 2016b. The interdependent network design problem for optimal infrastructure system restoration. *Comput. Aided Civil Infrastruct. Eng.* 31 (5), 334–350.
- Henry, D., Ramirez-Marquez, J.E., 2012. Generic metrics and quantitative approaches for system resilience as a function of time. *Reliab. Eng. Syst. Saf.* 99, 114–122.
- Hosseini, S., Barker, K., Ramirez-Marquez, J.E., 2016. A review of definitions and measures of system resilience. *Reliab. Eng. Syst. Saf.* 145, 47–61.
- Hu, Z.H., Sheu, J.B., 2013. Post-disaster debris reverse logistics management under psychological cost minimization. *Transp. Res. Part B* 55, 118–141.
- Iloglu, S., Albert, L.A., 2018. An integrated network design and scheduling problem for network recovery and emergency response. *Oper. Res. Perspect.* 5, 218–231.
- Kasaei, M., Salman, F.S., 2016. Arc routing problems to restore connectivity of a road network. *Transp. Res. Part E* 95, 177–206.
- Lempert, R.J., Groves, D.G., 2010. Identifying and evaluating robust adaptive policy responses to climate change for water management agencies in the American West. *Technol. Forecasting Soc. Change* 77 (6), 960–974.
- Liberatore, F., Ortúñoz, M.T., Tirado, G., Vitoriano, B., Scaparra, M.P., 2014. A hierarchical compromise model for the joint optimization of recovery operations and distribution of emergency goods in humanitarian logistics. *Comput. Oper. Res.* 42, 3–13.
- Nan, C., Sansavini, G., 2017. A quantitative method for assessing resilience of interdependent infrastructures. *Reliab. Eng. Syst. Saf.* 157, 35–53.
- Nicholson, C.D., Barker, K., Ramirez-Marquez, J.E., 2016. Flow-Based Vulnerability Measures for Network Component Importance: Experimentation with Preparedness Planning. *Reliab. Eng. Syst. Saf.* 145, 62–73.
- Nurre, S.G., Cavdaroglu, B., Mitchell, J.E., Sharkey, T.C., Wallace, W.A., 2012. Restoring infrastructure systems: an integrated network design and scheduling (INDS) problem. *Eur. J. Oper. Res.* 223 (3), 794–806.
- Ouyang, M., Fang, Y., 2017. A mathematical framework to optimize critical infrastructure resilience against intentional attacks. *Computer Aided Civil Infrastruct. Eng.* 32 (11), 909–929.
- Özdamar, L., Ertan, M.A., 2015. Models, solutions and enabling technologies in humanitarian logistics. *Eur. J. Oper. Res.* 244 (1), 55–65.
- Pant, R., Barker, K., Ramirez-Marquez, J.E., Rocco, C.M., 2014. Stochastic measures of resilience and their application to container terminals. *Comput. Ind. Eng.* 70, 183–194.
- Sharkey, T.C., Cavdaroglu, B., Nguyen, H., Holman, J., Mitchell, J.E., Wallace, W.A., 2015. Interdependent network restoration: on the value of information-sharing. *Eur. J. Oper. Res.* 244 (1), 309–321.
- Smith, A.M., González, A.D., Dueñas-Osorio, L., D'Souza, R.M., 2017. Interdependent Network Recovery Games. To appear in. *Risk Anal.*
- Vugrin, E.D., Camphouse, R.C., 2011. Infrastructure resilience assessment through control design. *Int. J. Crit. Infrastruct.* 7 (3), 240–260.
- Wang, J.W., Wang, H.F., Zhou, Y.M., Wang, Y., Zhang, W.J. On an integrated approach to resilient transportation systems in emergency situations Appearing online in *Nat. Comput.* <https://doi.org/10.1007/s11047-016-9605-y>.
- Wang, J., Muddada, R.R., Wang, H., Ding, J., Lin, Y., Liu, C., Zhang, W., 2016. Toward a resilient holistic supply chain network system: concept, review, and future direction. *IEEE Syst. J.* 10 (2), 410–421.
- Wang, J.W., Ip, W.H., Zhang, W.J., 2010. An integrated road construction and resource planning approach to the evacuation of victims from single source to multiple destinations. *IEEE Trans. Intell. Transp. Syst.* 11 (2), 277–289.
- White House, 2013. Presidential Policy Directive 21—Critical Infrastructure Security and Resilience. Office of the Press Secretary, Washington, DC.
- Xu, G., Wang, J., Huang, G.Q., Chen, C.H., 2018. Data-driven resilient fleet management for cloud asset-enabled urban flood control. *IEEE Trans. Intell. Transp. Syst.* 19 (6), 1827–1838.