Computers & Industrial Engineering 161 (2021) 107626

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

A heuristic approach to an interdependent restoration planning and crew routing problem

Nazanin Morshedlou [a,*], Kash Barker [b], Andrés D. González [b], Alireza Ermagun [c]

[a] Department of Industrial and Systems Engineering, Mississippi State University, Mississippi State, MS, USA
[b] School of Industrial and Systems Engineering, University of Oklahoma, Norman, OK, USA
[c] Department of Civil and Environmental Engineering, Mississippi State University, Mississippi State, MS, USA

ARTICLE INFO

Keywords:
Restoration planning
Vehicle routing problem
Interdependent infrastructure networks
Local search
Heuristic
Network resilience

ABSTRACT

This study proposes efficient solution methods to solve interdependent restoration planning and crew routing problems. The solutions provide reliable restoration plans regardless of the size and structure of disrupted infrastructure networks. We propose a relaxed mixed integer linear programming (MILP) model and two heuristic algorithms to find efficient feasible initial solutions for the restoration routing problem. We also propose a local search heuristic algorithm to find a near-optimal solution using the results obtained from the proposed model. Using electric power, water, and gas infrastructure network instances from Shelby County, TN, the computational results corroborate the efficacy of the mathematical formulation and shows that the heuristic algorithm obtains optimal or near-optimal solutions. In particular, we apply the sequence of the relaxed model, initial solution algorithm, and the local search algorithm for 62 scenarios with different magnitudes of disruption and different numbers of restoration crews. Analyzing the performance of the local search algorithm, we confirm the advantages of using the initial solution algorithm in producing the restoration schedules with reliable and relatively low total restoration time, particularly for large size problems. The observations also reveal how the scattering intensity in the distribution of disrupted locations affects the performance of relaxed models, and consequently, the integrated heuristic.

1. Introduction

Infrastructure networks, including power grids, water pipelines, and gas networks, are the backbone of the nation connecting businesses, communities, and residents (American Society of Civil Engineers, 2017). They have been subject to the effects of aging as explained by the steady D+ grade of the U.S. energy sector since 1988 (American Society of Civil Engineers, 2017). These aging structures, coupled with increasing storms and severe weather, have weakened the reliability of infrastructure networks before disruptive events, intensified their vulnerability to disruptive events, and lengthened their restoration process after disruptive events. As such, planning for their resilience is an important area of study.

Vugrin and Camphouse (2011) describe resilience capacity with three components: absorptive capacity, adaptive capacity, and restorative capacity. The *absorptive capacity* is the extent to which an infrastructure network can withstand the negative effect of a disruptive event. An example of absorptive capacity is the optimization methods employed in power grid networks to prevent large scale cascading blackouts before the occurrence of disruptions (Bienstock & Mattia, 2007). The *adaptive capacity* is the extent to which an infrastructure network can be adapted to a new condition after disruptions by temporary means. The short-term robust adaptive strategies implemented to the infrastructure network immediately after disruptions is a pertinent example (Lempert & Groves, 2010). The *restorative capacity* is the extent to which an infrastructure network is recovered in a long-term manner.

Previous research has attempted to introduce planning strategies for different components of resilience capacity in energy sectors (Bienstock & Mattia, 2007; Nan & Sansavini, 2017; Nurre et al., 2012), drinking water and wastewater (D'Ambrosio & Leone, 2015; Nurre et al., 2012), and transport and emergency response (Çelik et al., 2015). Research has also explored the interdependency between multiple infrastructure sectors (Barker et al., 2017; González et al., 2016; Sharkey et al., 2015; González et al., 2017). Little is known about incorporating work crew routing in restorative capacity planning models. Morshedlou et al. (2018) proposed two mixed integer programming (MIP) formulations

* Corresponding author at: Department of Industrial and Systems Engineering, 260 McCain Hall, Mississippi State University, USA.
E-mail address: morshedlou@ise.msstate.edu (N. Morshedlou).

https://doi.org/10.1016/j.cie.2021.107626
Received 6 January 2021; Received in revised form 2 July 2021; Accepted 14 August 2021
Available online 17 August 2021
0360-8352/

that combine the restoration crew scheduling problem with its associated vehicle routing problem to minimize crew movement time in the transport network. Each disrupted component exhibits specific characteristics, including the level of disruption, the rate of restoration, the importance in the network, and the increase in restoration rate when the next crew joins the process.

The current research is a natural extension of the restoration crew routing model proposed by Morshedlou et al. (2018). We offer two contributions to the existing literature. *First*, we propose a new mixed integer linear programming (MILP) model with a density-based spatial clustering of applications with a noise (DBSCAN)-based Mapping and Solution Algorithm (Algorithm 1). This helps obtain reliable lower bounds and upper bounds for the restoration routing problem while providing efficient feasible initial solutions using the outputs of a relaxed formulation. *Second*, acknowledging the NP-hard nature of routing and scheduling problems, we propose a local search heuristic algorithm to efficiently obtain a near-optimal feasible solution for large scale problems. We study the performance of the heuristic algorithm using the MILP and heuristic model results and illustrate them on small, medium, and large size problems. Examples represent individual power, water, and gas grid transmission networks in Shelby County, TN, and we study the interdependency of each infrastructure network and its counterpart routing network connecting disrupted locations.

The remainder of the paper is structured as follows. Section 2 reviews the literature of infrastructure network formulations focusing on restorative capacity and restoration arc routing problems. Section 3 formulates the relaxed MILP with summaries of the original model proposed by Morshedlou et al. (2018) in Appendix A. Section 4 presents a heuristic model to obtain lower bounds and upper bounds of the MILP model as well as a heuristic algorithm to obtain a near-optimal feasible solution aligned with restoration policies. Section 5 illustrates the use of the formulations and heuristic algorithm with realistic electric power, water, and gas networks in Shelby County, TN. We discuss the computational results and perform sensitivity analyses associated with the illustrative case study to showcase the efficacy of the proposed heuristic algorithm under diverse circumstances. Section 6 concludes the study by summarizing the major finding, the limitations of the study, and future research avenues.

2. Methodological background

In problems related to restoration planning, the aim is to find the sequence of disrupted network components assigned to and scheduled for each restoration crew such that network performance is improved at each time period of the restoration horizon. Within the context of network restoration, most studies focus on optimizing network flow balance by assigning and scheduling restoration crews to disrupted components, regardless of their geographical location and physical access. This category of problems is referred to as infrastructure network design and scheduling models. Limited restoration resources are allocated to disrupted components to maximize the weighted delivered residual flow in each time period (Arab et al., 2015; Nurre & Sharkey, 2014; Nurre et al., 2012; Sanci & Daskin, 2019). The nature of restoration scheduling problems is NP-hard, meaning the exact solution methods cannot find an optimal restoration schedule for large scale disruption scenarios. To find near-optimal solutions in a timely manner, previous research focused on heuristic and estimation algorithms, including a dispatching rule heuristic (Nurre et al., 2012), reduced-order linear representation (González et al., 2017), decomposition algorithm (Ouyang & Fang, 2017; Ghorbani-Renani et al., 2020), and conversion algorithm (Tan et al., 2019).

Other approaches modeled networks with multiple echelons, such as supply chain networks that consist of a transport echelon (Carvalho et al., 2012; Caunhye et al., 2020; Hishamuddin et al., 2013) and production and inventory echelon (Carvalho et al., 2012; Davis et al., 2021). Research also investigated these networks from protection (Behzadi et al., 2020; Behzadi et al., 2020; Carvalho et al., 2012; Dixit et al., 2020; Davis et al., 2021; Mao et al., 2020; Li et al., 2020) and restoration (Dixit et al., 2020; Mao et al., 2020) perspectives. The interdependency between restoration time and the routing time of crews was studied with a new class of problems referred to as multiple restoration crew routing problems. This was accompanied by efficient solution approaches as mathematical formulations were too complex to solve with exact solution methods. They include iterated greedy-randomized constructive procedures (Duque et al., 2016), ant colony algorithms (Kim et al., 2018; Shin et al., 2019), the Branch-and-Benders-cut approaches (Moreno, Munari, & Alem, 2020b), distributed optimization approaches (Su et al., 2019), Benders decomposition approaches (Moreno, Alem, Gendreau, & Munari, 2020a), consecutive heuristic approaches (Pamukcu & Balcik, 2020), tabu search algorithms (Pamukcu & Balcik, 2020), L-shape algorithms (Sanci & Daskin, 2021), rich local search algorithms (Akbari et al., 2021), and constructive heuristic and a simulated annealing algorithms (Atsiz et al., 2021).

Within the context of network restoration, in which crews are dispatched to disrupted network components through a routing network, studies mainly focused on road restoration, where the road network represents the lifeline infrastructure, such as debris or snow removal. These studies may differ from general infrastructure network restoration from two perspectives. First, disruptions may cause network components to become physically inaccessible for restoration crews. Second, incorporating the travel time between each pair of disrupted locations in the model may change the best restoration schedule. In this context, studies aim to find the restoration routing schedule that maximizes the connectivity of a disrupted network, or particularly access to different network components, in the minimum required time. As restoration routing problems are known to be NP-hard, novel solution approaches have been proposed to solve large scale problems in a timely manner, including dynamic path-based models (Aksu & Özdamar, 2014), relaxation-based constructive algorithms (Kasaei & Salman, 2016; Özdamar et al., 2014), Lagrangian relaxation (Akbari & Salman, 2017; Faturechi & Miller-Hooks, 2014), p-median (Iloglu & McLay, 2018), and Markov decision process under incomplete information (Çelik et al., 2015). Maya Duque et al. (2016) and Çelik (2017) offer comprehensive overviews of relatively recent advancements.

The proposed solution approaches in the current research stand apart from the previous studies as they are adaptable to a variety of infrastructures and restoration methodologies. We introduce a constructive heuristic algorithm that uses a local-search algorithm to improve the coordinated restoration routes obtained from the relaxed form of the original formulations.

3. Problem formulation

The proposed mathematical formulation involves two undirected networks: (1) physical lifeline infrastructure network and (2) the routing network. The physical lifeline infrastructure network (e.g., electric power) is represented by an undirected network $G = (N,A)$, where N is the set of nodes and A is the set of links. There are a set of supply nodes ($N_+ \subseteqq N$), a set of demand nodes ($N_- \subseteqq N$), and a set of transshipment nodes ($N_= \subseteqq N$). Each supply node $i \in N_+$ has a fixed supply capacity o_i in each time period, and each demand node $i \in N_-$ has a fixed demand requirement b_i in each time. Assume E is a set of possible disruptive events. Prior to disruption $\ell \in E$ at time t_0, each link has a baseline capacity, u_{ijt_0}, and a baseline flow value, f_{ijt_0}, calculated to satisfy all demand nodes in each period, calculated to satisfy all demand nodes in each period. Considering the flow capacity of each link, we aim to transfer flow from supply nodes to demand nodes and satisfy the total amount of demand. To prioritize demand nodes, the flow reaching each node, $i \in N_-$, is given a weight of w_i such that the more important a demand node is (e.g., located in a more populated area), the greater the weight assigned to it. Following a disruption at time t_d, there is a set of network components (links and nodes) that are adversely impacted.

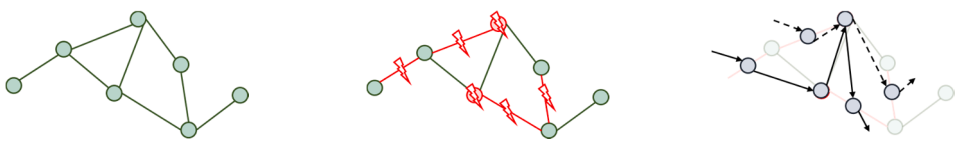

Fig. 1. Small illustrative example of the restoration crew routing problem.

Since any node can be represented as a pair of nodes and a link, we consider the disrupted components as a subset of links ($A'_\ell \subseteq A$). Each disrupted link has a new capacity, u_{ijt_d}, which can take any value less than u_{ijt_0}. The network $G = (N, A)$ changes to $G = (N, A/A'_\ell)$ also referred to as the residual network, where $N$is the set of nodes and A/A'_ℓ is the set of operational links.

The routing network is represented by a complete undirected graph $\bar{G}_\ell = \left(\bar{N}_\ell, \bar{A}_\ell\right)$, where $\bar{N}_\ell = \{1, \cdots, n\}$ is the set of disrupted locations and $\bar{A}_\ell$ is the set of links connecting each pair of locations. The routing network connects all disrupted locations in the infrastructure network $G = (N, A)$. There is a set of depots $\bar{N}_D \subseteq \bar{N}_\ell$ from which the restoration crews are distributed, a set of nodes $\bar{N}_\ell / \bar{N}_D$ that represent the disrupted locations on disrupted links $A'_\ell \subseteq A$ in graph $G = (N, A)$. A dummy sink node $n+1$ is also considered to be where all restoration routes end. K is the set of restoration crews that are distributed through the complete graph $\bar{G}_\ell = \left(\bar{N}_\ell, \bar{A}_\ell\right)$ such that all disrupted links $(i,j) \in A'_\ell$ in the infrastructure network, or their counterpart nodes in the routing network, $\bar{i} \in \bar{N}_\ell$, are fully restored. Parameter $|K|$ is the number of crews available to work on disrupted links in each period.

Each restoration crew can travel on each link $\left(\bar{i}, \bar{j}\right)$ in both directions with the same travel time $c_{\bar{i}\bar{j}}$. More than one crew can visit node $\bar{i} \in \bar{N}_\ell$ and the maximum number of crews that can be assigned to node $\bar{i}$ is $|L|$, where $|L| \leq |K|$. Without loss of generality, we assume that no crew visits node $\bar{i} \in \bar{N}_\ell$ unless it is assigned to restore that corresponding node. Fig. 1 illustrates a small example of infrastructure network G prior to disruption ℓ, network G after the disruption, and the counterpart restoration routing network $\bar{G}_\ell$, including a representation of parameter characteristics for each state of the network.

The objective function of the proposed model is to maximize a measure of resilience over the restoration horizon (Henry & Ramirez-Marquez, 2012). We define the flow reaching to demand node $i \in N_-$ as φ_{it}, and calculate it as the summation of flow reaching directly from transmission and supply nodes to demand nodes $\varphi_{it} = \sum_{j \in N_+ \bigcup N_-} f_{jit}$. The network performance at time $t = 1, \cdots, T$ is the total weighted flow reaching demand nodes $\sum_{i \in N_-} w_i \varphi_{it}$, where w_i refers to the prioritization weight assigned to each demand node. Before disruption ℓ, the total weighted demand is satisfied and equals $\sum_{i \in N_-} w_i \varphi_{it_0} = \sum_{i \in N_-} w_i \sum_{j \in N_+ \bigcup N_-} f_{jit_0}$. After the disruption, the total satisfied weighted demand across the residual network equals $\sum_{i \in N_-} w_i \varphi_{it_d} = \sum_{i \in N_-} w_i \sum_{j \in N_+ \bigcup N_-} f_{jit_d}$ and can be any number from zero (completely disrupted) to $\sum_{i \in N_-} w_i \varphi_{it_0}$ (completely operational).

The measure of resilience is a time-dependent ratio of the cumulative network restoration progress at each time period, $\sum_{i \in N_-} w_i \varphi_{it} - \sum_{i \in N_-} w_i \varphi_{it_d}$, to total performance loss, $\sum_{i \in N_-} w_i \varphi_{it_0} - \sum_{i \in N_-} w_i \varphi_{it_d}$. The progress of restoration is measured by the increase in the total weighted flow. Eq. (1) calculates the measure of resilience at each period. $\sum_{i \in N_-} w_i \varphi_{it_0}$ refers to the total weighted flow reaching demand nodes before the disruption at t_0, and $\sum_{i \in N_-} w_i \varphi_{it_d}$ refers to the total weighted flow when the full loss of network performance is felt at the time t_d.

$$Ð_{\varphi}(t|\ell) = \frac{\sum_{i \in N_-} w_i \varphi_{it} - \sum_{i \in N_-} w_i \varphi_{it_d}}{\sum_{i \in N_-} w_i \varphi_{it_0} - \sum_{i \in N_-} w_i \varphi_{it_d}} \quad (1)$$

Table 1
Parameters and variables used in the model formulation.

Indices	
A	Set of links, where $A'_\ell \subseteq A$ is the set of disrupted links
N	Set of nodes, where $N_+ \subseteq N$ is the set of supply nodes, $N_- \subseteq N$, is the set of demand nodes, and $N_= \subseteq N$ is the set of transmission nodes
$\bar{A}_\ell$	Set of routing links
$\bar{N}_\ell$	Set of routing nodes, where $\bar{N}_D \subseteq \bar{N}_\ell$, is the set of depots, and $\bar{N}_\ell / \bar{N}_D$, is the set of routing nodes (disrupted locations)
K	Set of restoration crews
L	Set of allowable crews working on a node simultaneously
t_0	Time immediately prior to disruptionℓ
t_d	Time after the end of disruptions
Parameters	
p_{ij}^k	Restoration time of link $(i,j) \in A'_\ell$ visited by crew$k \in K$
T	Restoration horizon limit
u_{ijt_0}	Baseline capacity of link$(i,j) \in$
f_{ijt_0}	Baseline flow on link$(i,j) \in A$
o_i	Capacity of supplier$i \in N_+$
b_i	Capacity of demand node$i \in N_-$
u_{ijt_d}	Capacity of disrupted link$(i,j) \in A'_\ell$
T	Recovery horizon limit
Variables	
f_{ijt}	Continuous variable, flow on link $(i,j) \in A'_\ell$ in the residual network at time$t$
φ_{it}	Continuous variable, total flow reaching demand node $i \in N_-$at time$t$
α_{kijt}	Binary variable, equal to 1 if $k \in K$ crews recover link $(i,j) \in A'_\ell$ at time t (Binary Active Model)

Appendix A represents the original formulations for restoration routing problems in detail. From the original model in Appendix A, the integration of the crew routing problem and the restorative capacity problem, along with the incorporation of the dynamic restoration process of each node, complicates the required execution time to solve both models. The execution time is positively correlated with the complexity of the model, and consequently, the complexity level of the model is positively correlated with the number of binary variables in the model. We measure the complexity level of the model by $O(|N|^2 \times |K| \times |A| \times T)$, where $|N|^2 \times |K| \times |A| \times T$ is the number of binary variables in the original model. Here, $|N|$ is the number of disrupted locations,$|K|$ is the number of crews,$|A|$ is the number of links, and T is the length of time horizon. The complexity of the original models increases the solution time drastically. To utilize the model in large scale problems, we propose effective bounds and heuristic solution approaches that reach to the near-optimal solution in a timely manner.

This Section proposes a relaxed formulation for restoration routing problems with multiple crews assigned to each disrupted location.

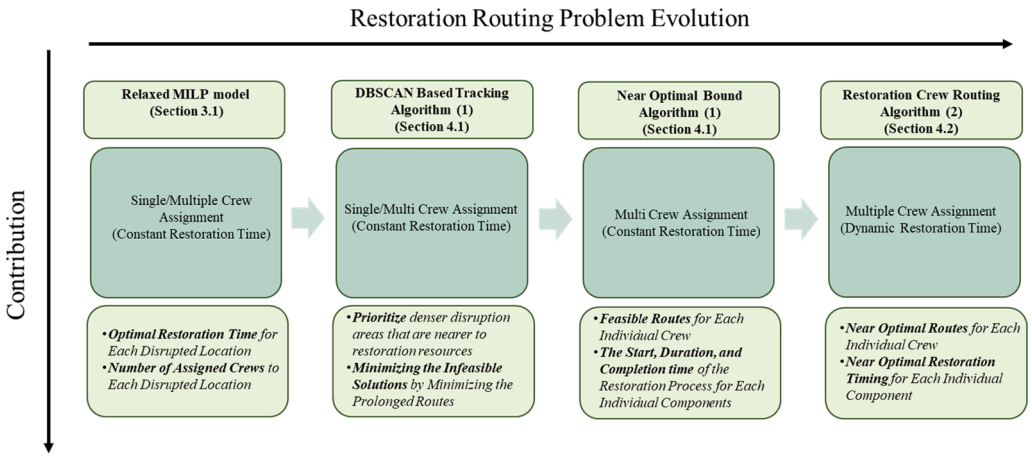

Fig. 2. Methodology flowchart of proposed models and their interdependencies.

Considering the restoration horizon, the proposed model, determines (i) the lower bound when it can only assign one crew to each disrupted component and (ii) the upper bound when it can assign multiple crews to each disrupted component. The proposed model maximizes the resilience measure in each period by incorporating the timing of restoration tasks as well as the number of assigned crews to each disrupted component, though the model cannot track the routing time of each crew individually.

3.1. Mixed linear integer programming formulation

The model identifies the start and completion time of restoration of each disrupted component. The model can only track the number of assigned crews considered in the models rather than specifying which crew is assigned to the node individually. We propose the mathematical formulation with the parameters and variables summarized in Table 1.

$$\max \sum_{t=1}^{T} Đ_{\varphi}(t) \tag{2}$$

$$\sum_{j:(i,j)\in A} f_{ijt} - \sum_{j:(j,i)\in A} f_{jit} \le o_i \quad \forall i \in N_+, t = 1, \cdots, T \tag{3}$$

$$\sum_{j:(i,j)\in A} f_{ijt} - \sum_{j:(j,i)\in A} f_{jit} = 0 \quad \forall i \in N_=, t = 1, \cdots, T \tag{4}$$

$$\sum_{j:(i,j)\in A} f_{ijt} - \sum_{j:(i,j)\in A} f_{jit} = -\varphi_{it} \quad \forall i \in N_-, t = 1, \cdots, T \tag{5}$$

$$0 \le \varphi_{it} \le b_i \quad \forall i \in N_-, t = 1, \cdots, T \tag{6}$$

$$u_{ijt_d} \le f_{ijt} \le u_{ijt_0} \quad \forall (i,j) \in A, t = 1, \cdots, T \tag{7}$$

$$u_{ijt_d} \le f_{ijt} \le \sum_{s=1}^{t} \alpha_{kijs} u_{ijt_0} \quad \forall (i,j) \in A', t = 1, \cdots, T \tag{8}$$

$$\sum_{s=1}^{T} \alpha_{ijs} \le 1 \quad \forall (i,j) \in A' \tag{9}$$

$$\sum_{t=1}^{p_{ij}^k - 1} \alpha_{kijt} = 0 \quad \forall (i,j) \in A', \forall k \in K \tag{10}$$

$$\sum_{k=1}^{K} \sum_{s=t}^{T} \left(1 + \left\lfloor \frac{t - \left(s - p_{ij}^k + 1\right)}{M} \right\rfloor\right) k\alpha_{kijs} \le |K| \quad \forall (i,j) \in A', t = 1, \cdots, T \tag{11}$$

Eqs. (3)–(8) are network flow balance constraints that send flow from supply nodes to the demand nodes through transshipment nodes and control the flow over the entire network. Eq. (3) ensures that the flow sent from each supply node $i \in N_+$ does not exceed its capacity. Eq. (4) requires that it must pass transmission nodes to reach demand nodes in subsequent steps. Eq. (5) calculates the amount of flow reaching each demand node $i \in N_-$. Eq. (6) ensures that the flow reaching each demand node will not exceed the capacity of that demand node. Eq. (7) ensures that the flow on any particular link will not exceed the capacity of that link. In Eq. (8), the flow on each link $(i,j) \in A'$, whether disrupted or restored, does not exceed the capacity of that link. Eq. (9) ensures that each disrupted link receives restoration services no more than once. Eq. (10) ensures that no link restoration process is completed before its required restoration time. Eq. (11) ensures that no more than $|K|$ restoration crews can work on disrupted components in each period.

4. Solution approach

Regarding the routing sequence and time, we propose a DBSCAN-based Mapping and Solution Algorithm (Algorithm 1) to minimize infeasible solutions using spatial clustering techniques as well as constructing efficient feasible initial solutions for the heuristic algorithm. Algorithm 1 prioritizes the restoration schedule of disrupted locations more densely found in affected areas relative to scattered locations in remote areas. We introduce a new linear constraint in Eq. (12) to include such clusters into the relaxed formulation. The updated model diminishes the size of the problem to $O(|A'| \times |K| \times T)$, which is a manageable number of constraints for medium sized problem instances.

To find an efficient initial solution, Algorithm 1 uses greedy techniques to provide (i) feasible routes for each crew and (ii) dynamic restoration time for each disrupted component with more than one assigned crew. The Restoration Crew Routing Heuristic Algorithm (Algorithm 2) uses the output of Algorithm 1 along with local search procedures to find feasible near-optimal solutions for large-scale problems. The flowchart in Fig. 2 portrays each model, its contribution, and connections to other proposed formulations for the ease of reading.

4.1. Discussion of DBSCAN-based Mapping and solution Algorithm

While benefiting from the simplicity of the relaxed formulations, two concerns may arise: (i) there is no control over the routing time of each crew due to long routes and (ii) the arrival time of each restoration crew assigned to a disrupted component might be different from the arrival time of other crews assigned to that component. Indeed, there might be cases where an assigned crew arrives after the restoration process of a disrupted component is completed by other crews. To eliminate both concerns, we propose a DBSCAN-based Mapping and Solution Algo-

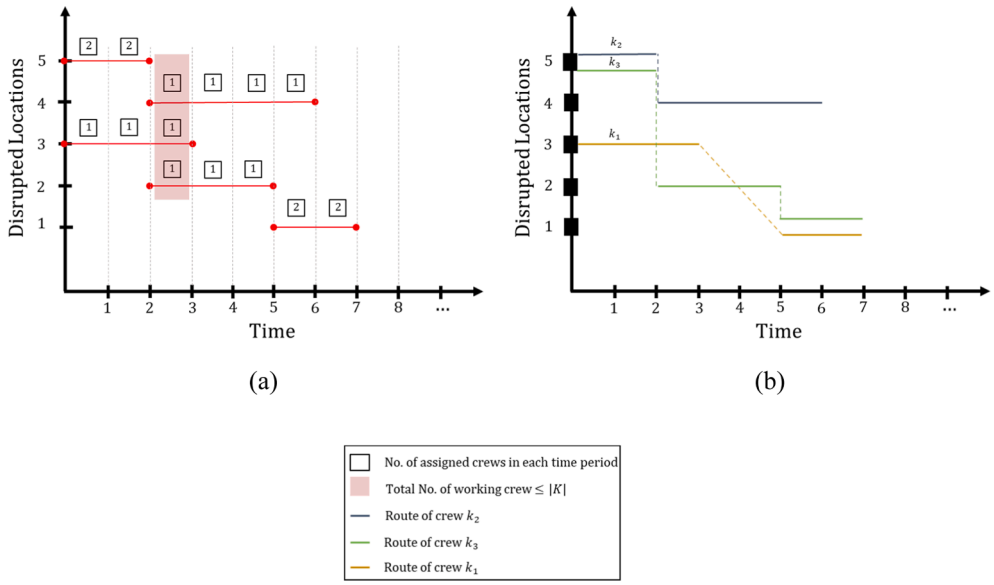

Fig. 3. Minimum distance-earliest release procedure for five disrupted locations and three restoration crews: (a) the Relaxed restoration problem optimal solution (input to the procedure), and (b) the route of each restoration crew (output of the procedure).

rithm, Algorithm 1 in Appendix B, which follows seven steps:

- **Step 1:** Use the DBSCAN method to cluster disrupted components that are spatially near one another to decrease the chance of prolonged or infeasible routes.
- **Step 2**: Rank the clusters based on the aggregate betweenness centrality of links included in each cluster, $I_{\mathbb{C}_i} = \sum_{(h,j)\in\mathbb{C}_i} \frac{f_{hjt0}}{u_{hjt_d}}$, over their average distance from depots, $D_{\mathbb{C}_i} = \frac{\sum_{j\in\overline{N}_{\mathscr{N}}/\overline{N}_D}\sum_{h\in\overline{N}_D} d(l_j,l_h)}{\left|\overline{N}_D\right|\times\left|\overline{N}_{\mathscr{N}}/\overline{N}_D\right|}$. Here, f_{hjt_0} is the flow on link $(h,j)\in\mathbb{C}_i$ in cluster $\mathbb{C}_i$ before the disruptive event, and u_{hjt_d} is the capacity of link (h,j) after the disruption. Distance $d(l_v, l_o)$ is the distance between the spatial location of link (h,j), shown by its counterpart node $v\in\overline{N}_{\mathscr{N}}/\overline{N}_D$ in the routing network, l_v, and the spatial location of depot, $o\in\overline{N}_D$, l_o. We call this measurement the total priority index $I^{\text{Total}}_{\mathbb{C}_i}$.
- **Step 3:** Implement Eq. (12) to distribute crews through the clustered network. The clusters with higher $I^{\text{Total}}_{\mathbb{C}_i}$ are prioritized to receive restoration services. D^{sort} is the set of sorted clusters in $D_{\mathbb{C}_i}$ and $D^{sort}_z \in D^{sort}$ represent each cluster of disrupted components where $z = 1,\cdots, |D^{sort}| - 1$.

$$\sum_{t=1}^{T}\sum_{k\in K}\alpha_{klht}(t-p^k_{lh}) \le \sum_{t=1}^{T}\sum_{k\in K}\sum_{(i,j)\in D^{sort}_z}\alpha_{kijt}(t-p^k_{ij}) \quad \forall (l,h)\in D^{sort}_{z+1} z = 1,\cdots,|D^{sort}|-1 \tag{12}$$

By recovering the most important links, this strategy guarantees a strong upper bound while it prevents crews assigned to disrupted locations with unrealistically heterogenous distances from one another. As the proposed model does not provide the trajectory of each individual crew, Algorithm 1 uses greedy techniques to provide a feasible route for each individual crew. The result of Algorithm 1 is the initial solution for the proposed heuristic algorithm.

- **Step 4:** Assign and direct each crew to the next nearest disrupted location immediately after it finishes the preceding restoration task.
- **Step 5:** Identify disrupted locations $\bar{i}\in\overline{N}_{\mathscr{N}}/\overline{N}_D$ to which multiple crews are assigned, or where $\alpha_{kijt} = 1, k\in K, k>1$.
- **Step 6:** Track the arrival time of each corresponding crew.

Table 2
The sequential assignment table and its corresponding dictionary.

Crew	Sequenced disrupted locations				Sequenced timing of disrupted locations restoration activities			
1	b_{11}	b_{12}	⋯	b_{1n_1}	$\Im^k_{b_{11}}, \mathscr{B}^k_{b_{11}}, \mathscr{T}^k_{b_{11}}$	$\Im^k_{b_{12}}, \mathscr{B}^k_{b_{12}}, \mathscr{T}^k_{b_{12}}$	⋯	$\Im^k_{b_{12}}, \mathscr{B}^k_{b_{12}}, \mathscr{T}^k_{b_{12}}$
2	b_{21}	b_{22}	⋯	b_{1n_2}	$\Im^k_{b_{21}}, \mathscr{B}^k_{b_{21}}, \mathscr{T}^k_{b_{21}}$	$\Im^k_{b_{22}}, \mathscr{B}^k_{b_{22}}, \mathscr{T}^k_{b_{22}}$	⋯	$\Im^k_{b_{22}}, \mathscr{B}^k_{b_{22}}, \mathscr{T}^k_{b_{22}}$
⋮	⋯							
$\lvert K\rvert$	$b_{\lvert K\rvert 1}$	$b_{\lvert K\rvert 2}$	⋯	$b_{\lvert K\rvert n_{\lvert K\rvert}}$	$\Im^k_{b_{\lvert K\rvert 1}}, \mathscr{B}^k_{b_{\lvert K\rvert 1}}, \mathscr{T}^k_{b_{\lvert K\rvert 1}}$	$\Im^k_{b_{\lvert K\rvert 2}}, \mathscr{B}^k_{b_{\lvert K\rvert 2}}, \mathscr{T}^k_{b_{\lvert K\rvert 2}}$	⋯	$\Im^k_{b_{\lvert K\rvert n_{\lvert K\rvert}}}, \mathscr{B}^k_{b_{\lvert K\rvert n_{\lvert K\rvert}}}, \mathscr{T}^k_{b_{\lvert K\rvert n_{\lvert K\rvert}}}$

- **Step 7:** Delete the late crews from the schedule of corresponding disrupted locations.

The relaxed model outcome provides the time sequence of the restoration process of disrupted links and the number of crews assigned to each disrupted link as shown in Fig. 3a. Following solving the relaxed formulation, Fig. 3b illustrates how Algorithm 1 (Appendix B) calculates the arrival time of assigned crew $\boldsymbol{k}\in\boldsymbol{K}$ to disrupted location $\bar{\boldsymbol{i}}\in\overline{\boldsymbol{N}}_{\mathscr{N}}/\overline{\boldsymbol{N}}_{\boldsymbol{D}}$, the processing time of each disrupted location $\bar{\boldsymbol{i}}\in\overline{\boldsymbol{N}}_{\mathscr{N}}/\overline{\boldsymbol{N}}_{\boldsymbol{D}}$, and the routing time of each crew $\boldsymbol{k}\in\boldsymbol{K}$. Algorithm 1 eliminates infeasible routes due to (i) the arrival of a crew at a disrupted location after the completion of its restoration process and (ii) the presence of a crew at two different disrupted locations simultaneously (i.e., cycle sub-graph formation (Morshedlou et al., 2018)).

Each row of Table 2 illustrates the sequence of disrupted links scheduled to crew $k\in K$ and its counterpart sequenced arrival, restoration, and departure times. Here, b_{kh} is the h^{th} link $(i,j)\in A'_{\mathscr{N}}$ visited by crew $k\in K$, and $|K|$ is the total number of rows. Its corresponding set of values represents the ith disrupted location in the schedule of crew $k\in K$, and its corresponding set of values is a vector which includes time when crew $k\in K$ starts ($\Im^k_{b_{kh}}$), proceeds ($\mathscr{B}^k_{b_{kh}}$), and ends ($\mathscr{T}^k_{b_{kh}}$) the restoration process of b_{kh}.

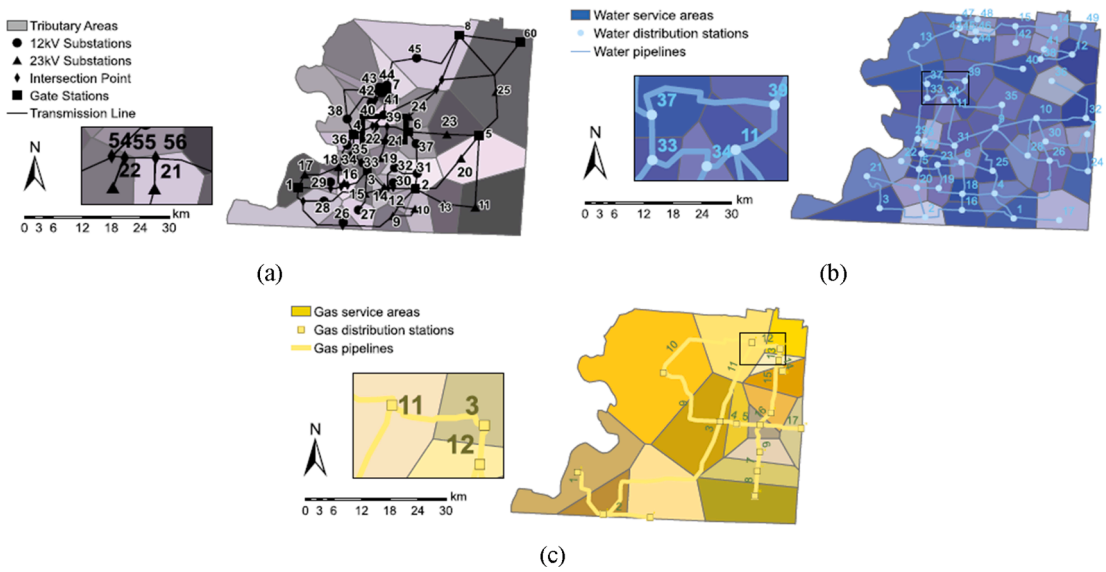

Fig. 4. Graphical representations of the (a) power grid transmission, (b) water, and (c) gas networks at a transmission level in Shelby County, TN (González et al., 2016).

4.2. A heuristic for the restorative capacity routing problem

In the context of enhancing restorative capacity, solving large instances in a shorter solution time is of great importance. We propose the *Restoration Routing Local Search* heuristic algorithm (Algorithm 2) using the feasible solution obtained from Algorithm 1, which is the sequential assignment depicted in Table 2. Algorithm 2 uses four procedures to obtain a near-optimal feasible solution for large-scale problems: (1) Shorten Routes, (2) Add Crew, (3) Remove Crew, and (4) Change Assignment.

Procedure Shorten Routes: Consider the routes and routing times in Table 2. This procedure is defined in two steps:

- **Step 1:** Find the route with the maximum routing time.
- **Step 2:** Among the crews that complete their routes sooner, check if there is any crew that can join the restoration time of the last node in the route and shortens the processing time of the node.

Procedure Add Crew: Consider the routes and routing times generated in Table 2. This procedure is defined in three steps:

- **Step 1**: Find the route with maximum routing time.
- **Step 2**: Among nodes on the route, find the one with the maximum weighted processing time, referred to as the bottleneck node. The weight of each node represents the betweenness centrality of that node.
- **Step 3**: Check if there is a crew working on a nearby node that can join the restoration process of the bottleneck node and improve the processing time. The routing time of the prolonged schedule must not exceed the maximum routing time.

Procedure Remove Crew: This procedure is defined in three steps:

- **Step 1:** Among nodes with more than one crew assigned to their restoration process, choose a node with the minimum weighted processing time, also called the least important node.
- **Step 2:** Among the assigned crews, chose the one that works on the immediate succeeding node located in// the minimum distance of the least important node.
- **Step 3:** Ensure that removing the crew from the schedule of the least important node does not worsen the maximum restoration time.

Procedure Change Assignment: This procedure is a hybrid of *Add Crew* and *Remove Crew* procedures.

The output leads to an improvement in the measure of total network resilience in each period representing: (i) shorter processing time for some disrupted links, (ii) shorter routing time for some restoration crews, and (iii) a change in the restoration schedule of each crew. The integration of procedures forms an easily implementable heuristic algorithm to obtain a near-optimal feasible solution.

5. Illustrative examples: power, water, and gas networks

To examine the efficacy of the proposed model and the heuristic algorithms, we employ three realistic data sets based on power grid transmission, water, and gas networks in Shelby County, TN. Located in the New Madrid Seismic Zone, Shelby County has over 650,000 and is home to Memphis. As illustrated in Fig. 4, the power network consists of 289 components (125 nodes and 164 links), the water network consists of 120 components (49 nodes and 71 links), and the gas network consists of 33 components (16 nodes and 17 links). Adapted from González et al. (2016), we consider four disruption scenarios $\ell = M_6, M_7, M_8, M_9$ over each network with the disruption levels $E = \{\ell = M_6(\text{Mild}), \ell = M_7(\text{Moderate}), \ell = M_8(\text{High}), \text{and} \ell = M_9(\text{Severe})\}$ with an average disruption of 6%, 9%, 17%, and 23% of all network components, respectively. The disruptions are selected among 6%, 9%, 17%, and 23% of the most critical network components (i.e., the network components with the highest proportional flow) in the disruption levels M_6, M_7, M_8, and M_9, respectively. The distance between each pair of disrupted locations is the shortest unblocked path obtained through ArcGIS and Google Earth. Each crew travels the shortest undisrupted path between each pair of consecutive locations on its restoration routing schedule. After a disruptive event, each restoration crew starts its route from its original depot. Without loss of generality, we eliminate the effect of the spatial distribution of depots on the routing schedule of crews by assuming that a depot has a distance equals zero from each disrupted location. Each disrupted location may experience a certain level of disruption and require a specific time interval to be restored, depending on the number of crews assigned to that component. The restoration time horizon is considered to be 150 time periods. For crews that may leave a disrupted location in the middle of its restoration process and after their restoration task is finished, we consider each processing time as a function of that corresponding component's characteristics.

For the infrastructure networks shown in Fig. 4, we evaluate the

Table 3
Solution time, percentage gap, and restoration time for the relaxed model (upper and lower bounds) and after implementing Algorithm 1 for the power network.

Ins.	⁄	$\|K\|$	Relaxed model, single crew lower bound			Relaxed model, multiple crew upper bound			Algorithm 1		
			CPU	Gap (%)	T	CPU	Gap (%)	T	CPU	Gap (%)	T
1	$M_6(10)$	2	1	0	50	6	0.01	38	10	0.02	45
2		3	1	0	46	6	0.24	29	3	0	30
3		4	1	0	38	7	0.03	27	1	0.01	28
4		5	1	0	36	8	0.05	20	2	0	22
5	$M_7(15)$	2	5	0	50	1	0	52	0	0	58
6		3	5	0	47	1	0	39	0	0	39
7		4	5	0	47	1	0.01	30	0	0	31
8		5	5	0	42	1	0.01	25	0	0	27
9	$M_8(28)$	2	35	0.04	38	145	0.09	60	66	0.16	64
10		3	35	0.03	28	21	0.05	45	450	0.04	49
11		4	50	0.04	23	24	0.04	38	325	0.04	40
12		5	75	0.01	23	20	0.04	30	690	0.04	34
13		6	65	0.02	23	35	0.01	29	925	0.04	31
14		7	25	0.01	20	29	0.01	25	825	0.02	27
15	$M_9(30)$	2	852	0.95	79	3180	1	127	3600	2.2	129
16		3	809	0.38	49	3600	0.6	91	3600	0.9	90
17		4	545	0.39	42	3600	0.45	69	3597	0.3	68
18		5	579	0.19	36	3600	0.3	58	3600	0.47	60
19		6	1271	0.11	30	3600	0.1	51	3600	0.54	50
20		7	1143	0.12	23	3600	0.1	48	3600	0.2	45

Table 4
Solution time, percentage gap, and restoration time for the relaxed model (upper and lower bounds) and after implementing Algorithm 1 for the water network.

Ins.	⁄	$\|K\|$	Relaxed model, single crew lower bound			Relaxed model, multiple crew upper bound			Algorithm 1		
			CPU	Gap (%)	T	CPU	Gap (%)	T	CPU	Gap (%)	T
21	$M_6(8)$	2	1	0	132	145	0.01	144	150	0.09	145
22		3	1	0	89	235	0.05	88	150	0.09	88
23		4	3	0	77	73	0.01	89	73	0.01	84
24		5	4	0	63	633	0.01	79	126	0.13	78
25		6	8	0	63	492	0.01	76	45	0.44	49
26		7	5	0	51	384	0	73	7	0.01	45
27	$M_7(13)$	2	4	0	183	800	0.03	189	575	0.02	189
28		3	10	0	137	375	0.03	123	951	0	129
29		4	16	0	100	375	0.03	123	1870	0.01	116
30		5	9	0	89	771	0.02	82	1304	0.01	104
31		6	17	0	89	1228	0.2	80	3551	0.01	100
32		7	19	0	55	3600	5.5	79	3600	0.2	94
33	$M_8(25)$	2	560	0	277	3600	2.79	272	3600	6	277
34		3	45	0.01	217	3600	0.96	177	3600	1.15	154
35		4	134	0.06	170	3600	0.96	104	3600	0.96	107
36		5	479	0	134	3600	0.96	102	3600	0.56	90
37		6	504	0	118	3600	0	79	3600	0.12	81
38		7	700	0	104	3600	2.75	65	2375	0.16	39
39	$M_9(30)$	2	785	0.01	418	1624	19	399	3600	10	398
40		3	992	0	287	3600	7	257	3600	2.66	196
41		4	2271	0	216	1624	3	138	3600	0.7	116
42		5	2045	0	171	1793	6	106	3600	0.25	93
43		6	1064	0	145	3600	0	81	3600	0.45	82
44		7	2745	0	126	3600	0	75	3600	0.46	76

performance of the proposed model, and Algorithm 1 on 62 instances (i.e., four disruption scenarios for each infrastructure network) with the percentage of disruptions varying between 6% and 23% of network components with high flow. Tables 3–5 shows the computational experiment. According to Tables 3–5, at least two and at most seven crews are dispatched through the network to recover disrupted components. l number of crews can be assigned to a disrupted location, where $l = \{1,2,\cdots,7\}$. If l number of crews works one disrupted location, it is recovered at the rate of λ_l units per each time period, where $\lambda_{l+1} \geq \lambda_l$.

5.1. Empirical observations

As expected, the results in Tables 3–5 show that a greater number of disrupted components result in a drastic increase in the solution time of the restoration problem and a greater optimality gap in a specific solution time horizon (e.g., one hour). However, results in Tables 3–5 suggest that the total restoration time resulting from solving Algorithm 1 is greater than the total restoration time obtained from solving the relaxed model. This is the result of employing DBSCAN technique in Algorithm 1 to divide the disrupted components under disruption scenarios M_6, M_7, M_8, and M_9, into (i) 7(1), 8(1), 10(1.5), and 14(1.9) clusters for the power network, (ii) 8(1), 10(1.3), 17(1.8), and 13(2.2) for the water network, and (iii) 2(1), 5(1), 7(1.3), and 7(1.3) for the gas network (with the average number of components in each cluster found in the parentheses). The application of these clusters in Eq. *(12)* limits the choices of the model for the next best set of disrupted components to restore. This is because no disrupted component from the clusters farther from the depots can be restored unless all disrupted components in the clusters nearer to depots are restored. The situation will be exacerbated if the number of disrupted components in nearer clusters is less than the number of available crews. Therefore, the optimal solution of Algorithm

Table 5
Solution time, percentage gap, and restoration time for the relaxed model (upper and lower bounds) and after implementing Algorithm 1 for the gas network.

Ins.	⁄	$\|K\|$	Relaxed model, single crew lower bound			Relaxed model, multiple crew upper bound			Algorithm 1		
			CPU	Gap (%)	T	CPU	Gap (%)	T	CPU	Gap(%)	T
45	$M_6(3)$	2	0	0	43	0	0	51	0	0	51
46		3	0	0	34	0	0	38	0	0	38
47	$M_7(5)$	2	0	0	71	0	0	88	0	0	88
48		3	0	0	58	0	0	63	0	0	63
49		4	0	0	47	0	0	64	0	0	64
50		5	0	0	47	0	0	56	0	0	56
51	$M_8(8)$	2	0	0	100	3	0	97	3	0	97
52		3	0	0	81	2.60	0	81	2.36	0	81
53		4	0.3	0	57	3	0	70	3	0	70
54		5	0.75	0	57	2.78	0	56	2.57	0	56
55		6	0.06	0	49	2.70	0	46	2.99	0	46
56		7	0.34	0	40	3	0	46	3	0	46
57	$M_9(10)$	2	0.96	0	87	3	0	94	3	0	94
58		3	0.64	0	63	3	0	81	3	0	81
59		4	1.31	0	50	2.80	0	75	2.33	0	75
60		5	1.37	0	43	3	0	65	3	0	65
61		6	1.40	0	39	2.51	0	56	3.05	0	56
62		7	1.16	0	38	3	0	50	3	0	50

1 is a lower bound for the optimal solution of the relaxed model.

For Tables 3–5, the first four columns on the left show the type of network, the number of instances, the level of the disruption, ⁄, and the number of crews engaged into the network restoration process, $|K|$, respectively. The next three columns indicate the results related to the single crew assignment (lower bound), multiple crew assignment (upper bound), and Algorithm 1. For each, the CPU time, the optimality gap, and the completion time of the restoration process of each disrupted network instances are provided. To solve these formulations, Gurobi 8 was used, along with Python 2.7.14. The reported CPU time is in seconds and all instances were tested on an Intel® Core i7-7500U CPU @ 2.70 GHz 2.90 GHz (two processor) with 32 GB RAM.

5.2. Computational results for heuristic approach

The optimal solution for the restoration routing problem with single crew assignment provides a lower bound for the original problem. We employ Algorithm 1 for each problem instance in Tables 3–5 to incorporate the routing time of each restoration crew into the optimal solutions for the relaxed single and multiple crew assignment models. For each problem instance, the output of Algorithm 1 is a feasible initial solution that provides an upper bound for the original problem. Using these initial solutions, Algorithm 2 modifies the solution obtained from Algorithm 1 to find a near-optimal solution for the original problem. Morshedlou et al. (2018) solved the original model, the solution for which appears in Appendix C for comparison. Fig. 5 compares the trajectory of total restoration time for the original model by Morshedlou et al. (2018), the relaxed model under single crew assignment assumption, Algorithm 1, as well as for Algorithm 2 when applied to the multiple crew assignment model and Algorithm 1. The algorithm is applied to the power (Fig. 5a), water (Fig. 5b), and gas (Fig. 5c) network instances in Shelby County, TN. The green, yellow, orange, and red areas indicate the trajectory of total restoration time under the disruption scenario levels M_6, M_7, M_8, and M_9, respectively.

In a one-hour CPU time, the solution of the original problem results in an optimality gap of 9.36, 7.58, and 1.92, for the power, water, and gas networks, respectively. As shown in Fig. 5a-c, the model cannot solve M_9 problem instances for water and power networks. Employing Algorithm 2, the initial solutions obtained from the relaxed model enhance by 5.90%, 3.82%, and 7.88%, with an average of 23.38, 12.69, and 26.18 s, respectively, for the power, water, and gas network instances. As for Algorithm 1, these values change to 7.49%, 8.94%, and 8.76%, with an average of 17.10, 16.39, and 6.70 s, respectively. As shown in Fig. 5, compared with the original model, the application of Algorithm 2 on the output of Algorithm 1 results in reliable near-optimal solutions. The results can be used as substitutes for the exact solutions in large-scale problems where the model fails to reach a reliable restoration planning schedule in a timely manner.

Fig. 5 shows that, on average, Algorithm 1 results in a shorter total restoration time than its counterpart relaxed model. However, some fluctuations seen in the behavior of the model prevent the ubiquity of such results. We can interpret this behavioral abnormality as a function of the dependency on the spatial location of the disrupted component. Fig. 6 illustrates the spatial distribution of disrupted component for power, water, and gas networks under disruption scenarios M_6, M_7, M_8, and M_9. Observations from Fig. 5 and Fig. 6 confirm that high scattering intensity (i.e., scenarios M_6 and M_7 for power, water, and gas networks in Fig. 6) disturbs the performance of Algorithm 1, resulting in greater restoration time than expected. As the spatial distribution of disrupted components becomes more congregated (scenarios M_8 and M_9 for power, water, and gas networks in Fig. 5), the performance of Algorithm 1 enhances considerably and results in a lower average restoration time as compared to the other proposed models, as shown especially with scenarios M_8 and M_9 in Fig. 5.

6. Concluding remarks

We proposed a new heuristic algorithm for the infrastructure network restoration routing problem to encompass realistic assumptions and contexts. Integrating the post disruption resilience problem and routing problems, we demonstrated that the restoration assignment and schedule represented in basic infrastructure network restoration models might not be feasible in realistic contexts as they do not consider (i) the travel time of a crew between each pair of assigned disrupted locations, and (ii) the difference in arrival time of each assigned crew to a disrupted location. Disregarding these two conditions results in the restoration schedules that include a crew potentially present in two different locations at the same time or a crew that reaches a disrupted location after its restoration process is completed. This research addressed these limitations by formulating:

- A relaxed restoration crew assignment restoration model to identify the efficient lower and upper bounds for the original problems;
- A DBSCAN-based Mapping and Solution Algorithm (Algorithm 1) to cluster disrupted locations, prioritize these clusters based on their average importance rather than their distance from depots, and assign components to crews accordingly; and

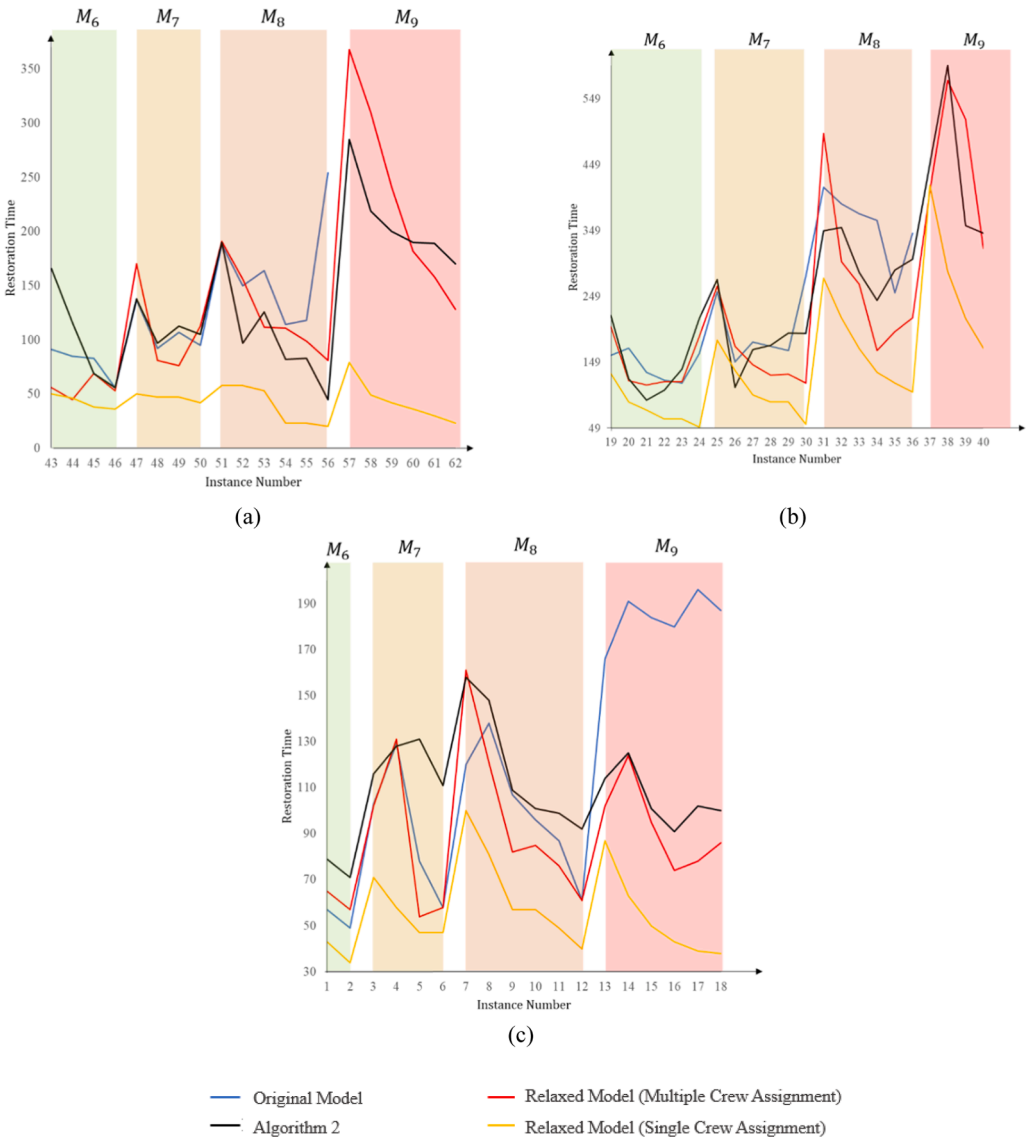

Fig. 5. The trajectory of total restoration time for the original model, relaxed model considering single crew assignment, the application of Algorithm 2 on the multiple crew assumption, and Algorithm 1 Formulations for the (a) power, (b) water, and (c) gas network instances in Shelby County, TN. The green, yellow, orange, and red zones indicate the trajectory of restoration time under the disruption scenario levels M_6, M_7, M_8, and M_9, respectively.

- A local search algorithm to improve the initial solution and obtain optimal or near-optimal solutions in a reasonable time.

We tested the relaxed model using instances derived from realistic case studies from power, water, and gas grid networks in Shelby County. We applied the sequence of the relaxed model, initial solution algorithm (Algorithm 1), and the local search algorithm (Algorithm 2) for 62 scenarios with different magnitude of disruptions (M_6, M_7, M_8, and M_9) and different numbers of restoration crews ranging from 2 to 7. The key practical findings are encapsulated in the following:

- Algorithm 1 provides the initial solutions that help local search heuristic reaches improved solutions in the same given solution time.
- The local search algorithm provides reliable solutions in reasonable time regardless of the initial solution that is given to it.
- The scattering intensity in the distribution of disrupted locations upgrades the performance of the relaxed model.
- The scattering intensity in the distribution of disrupted locations downgrades the performance of the Algorithm 1.
- The performance of Algorithm 1 overcomes the exact formulation in highly congregated locations.

This research compared the proposed heuristic approach with the exact solution and tried to find efficient bounds for the original model that also serve as efficient thresholds to validate the proposed heuristics along with the original model. This calls attention to further research on the comparison between Algorithm 2 and other existing heuristic algorithms to corroborate the efficiency of the algorithm under the circumstances that differ from those considered in this research. Along with that, this study has room to improve with further research. The routing network itself could be disrupted and disconnected physically in several locations, making it impossible for the crews to reach some disrupted locations. In representing this issue, the crews restoring a non-transport infrastructure must wait for the crews restoring the transportation infrastructure to open some blocked paths. Finally, some restoration crews might finish their restoration process earlier than others and leave the disruption component which they are assigned to before its restoration process is completed. In these cases, methods can be developed to improve the performance of the heuristic algorithm

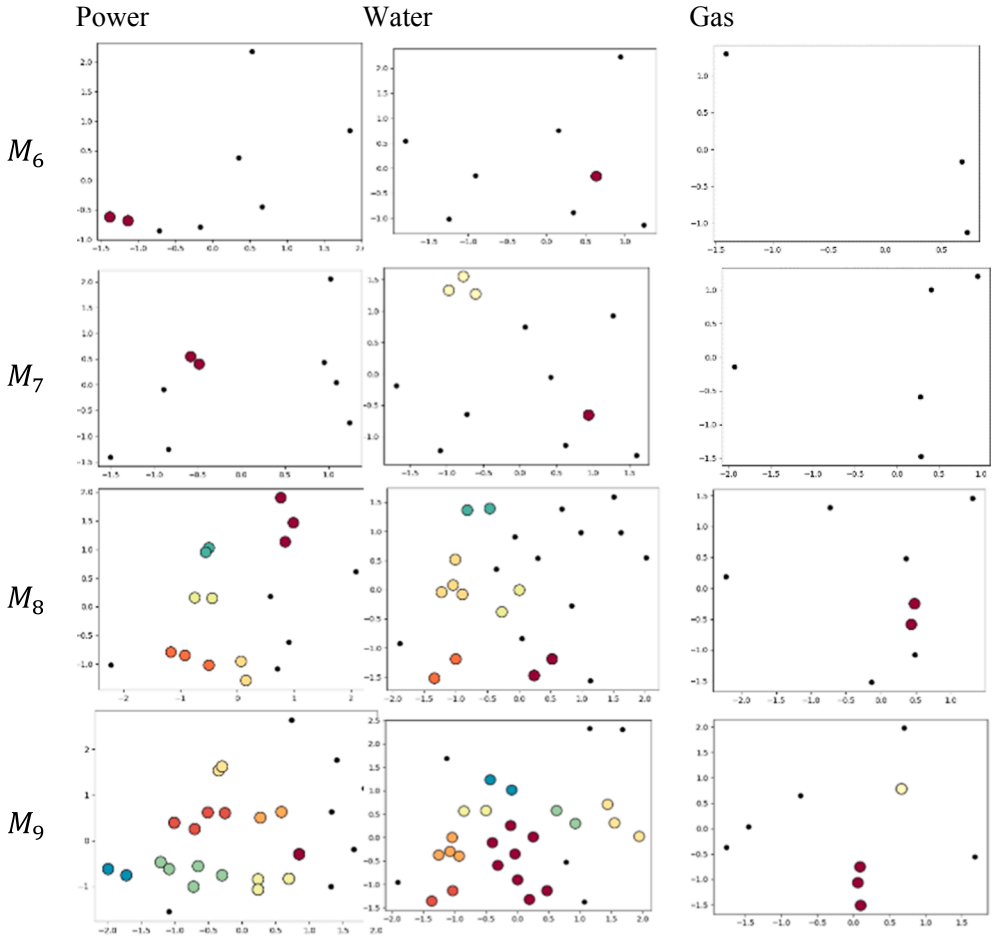

Fig. 6. Spatial location of disrupted component for power, water, and gas networks under disruption scenarios 6, 7, 8, and 9.

efficiently. However, additional work is needed to consider instances where parameters are subject to uncertainty.

Regardless of the limitations identified, this work can assist decision-makers gain analytical perspectives on the critical transportation-related factors affecting the performance of a restoration schedule. The proposed models reflect how the scattering intensity, and the distribution of disruptions affect the number of required crews and the average number of assigned locations to each crew. The results also help track the prepositioning candidates for crew depots with a high level of access under different disruptions scenarios.

CRediT authorship contribution statement

Nazanin Morshedlou: Conceptualization, Methodology, Validation, Formal analysis, Writing – original draft. **Kash Barker:** Conceptualization, Formal analysis, Visualization, Supervision, Project administration, Funding acquisition. **Andrés D. González:** Conceptualization, Formal analysis, Resources, Investigation. **Alireza Ermagun:** Validation, Formal analysis, Writing – original draft, Visualization.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A

A.1. Notations

There is a maximum number of crews, $v_{\bar{i}}$, that can be sent from each depot, $\bar{i} \in \overline{N}_D$. For each disrupted location $\bar{i} \in \overline{N}_{\diagup}/\overline{N}_D$, $z_{\bar{i}}^l$ is a binary variable that equals 1 if $l = \{1, \cdots, |L|\}$number of crews visit the corresponding location, and 0 otherwise. For the directed routing network, x_{ijk}, is a binary variable that equals 1 if restoration crew $k \in K$ travels from node $\bar{i} \in \overline{N}_{\diagup}$ to node $\bar{j} \in \overline{N}_{\diagup}/\overline{N}_D \bigcup (n+1)$, and 0 otherwise. The arrival time of each assigned crew $k \in K$ at disrupted location $\bar{i} \in \overline{N}_{\diagup}/\overline{N}_D$ is a binary variable $\tau_{\bar{i}t}^k$, which equals 1 if t is the arrival time of crew k to node $\bar{i}$, and 0 otherwise. The completion time of the restoration process associated with each disrupted link $(i,j) \in A^{'}$, or its counterpart node in routing network, $\bar{i} \in \overline{N}_{\diagup}/\overline{N}_D$, is a continuous variable, $\beta_{\bar{i}}^l$, where l is the number of restoration crews assigned to node $\bar{i} \in \overline{N}_{\diagup}/\overline{N}_D$. The maximum processing time of each node $\bar{i} \in \overline{N}_{\diagup}/\overline{N}_D$ equals to $p_{\bar{i}}^1$ when only one crew is assigned to that corresponding node, and $p_{\bar{i}}^l$ is the restoration time of node $\bar{i}$ when l crews are assigned to it. All start

Table A1
Notation for the original restoration routing problems.

Infrastructure and routing 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$G = (N,A)$
$\overline{N}$	Set of nodes in network$\overline{G} = (\overline{N},\overline{A})$
$\overline{N}_{A'}$	Set of nodes in network $\overline{G} = (\overline{N},\overline{A})$ corresponding to disrupted links in network$G = (N,A)$
$\overline{N}_D$	Set of depots from which recovery crews commence their routes
$(n+1)$	Dummy sink node where the path of all restoration crews ends
$\overline{A}$	Set of links in network $\overline{G} = (\overline{N},\overline{A})$ that connect the nodes corresponding to disrupted links in network$G = (N,A)$
K	Set of restoration crews, where $\|K\|$ is the total number of crews working during the restoration horizon
$\{1,\cdots,L\}$	Set of restoration crews assigned to each node $i \in \overline{N}_{A'}$, where L is the maximum number of crews that can be assigned to each disrupted component
$\{1,\cdots,T\}$	Set of time periods in the restoration horizon
Parameters	
$v_{\bar{i}}$	The maximum number of vehicles sent from depot$\bar{i} \in \overline{N}_D$
$p_{\bar{i}}^{l}$	The processing time of node$\bar{i} \in \overline{N}_{A'}$ when l crews are assigned to it
$\lambda_{\bar{i}}^{l}$	The progress of restoration process of node$\bar{i} \in \overline{N}_{A'}$ per time unit, when l crews are working on it
$c_{\bar{i}\bar{j}}$	The traveling time from node $i$to node $\bar{j}$,$(\bar{i},\bar{j}) \in \overline{A}$
$\theta_{ij\bar{i}}$	The binary parameter equals to 1 if node $\bar{i} \in \overline{N}_{A'}$ is a counterpart to link $(i,j) \in A'$, and 0 otherwise
$u_{\bar{i}te}$	Capacity of node $\bar{i} \in \overline{N}_{A'}$, or its corresponding link $(i, j) \in A'$, before the disruptive event
$u_{\bar{i}td}$	Capacity of node $\bar{i} \in \overline{N}_{A'}$, or its corresponding link $(i, j) \in A'$, immediately after the disruptive event
b_i	Capacity of demand node$i \in N_-$
o_i	Capacity of supply node$i \in N_+$
M	Very large number
Decision variables	
$x_{\bar{i}\bar{j}}^{k}$	Binary variable equals to 1 if crew$k \in K$ travels link$(\bar{i},\bar{j}) \in \overline{A}$
$z_{\bar{i}}^{l}$	Binary variable equals to 1 if $l \in L$ restoration crews are assigned to node$i \in \overline{N}_{\checkmark}/\overline{N}_D$
$\tau_{\bar{i}t}^{k}$	Binary variable equals to 1 if crew $k \in K$ arrives to node $\bar{i} \in \overline{N}_{\checkmark}/\overline{N}_D$ at time$t$
$g_{\bar{i}t}^{l}$	Binary variable equals to 1 if the lth crew arrives to node $\bar{i} \in \overline{N}_{\checkmark}/\overline{N}_D$ 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$t$
φ_{it}	Integer variable representing the flow reaching to demand node $i \in N_-$ at time$t$
α_{ijt}	Binary variable equals to 1 if restoration task on link (i,j) finishes at time$t$
$f_{\bar{i}\bar{j}}^{k}$	Integer variable representing the flow of restoration crew $k \in K$ on link$(\bar{i},\bar{j}) \in \overline{A}_{\checkmark}$

the restoration process at the same time. Starting from its originating depot, each crew travels a specific route, which is a sequence of disrupted locations and ends in the dummy sink node $(n+1)$. The restoration 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 restoration crew to that link.

To relate the two aforementioned networks, $\theta_{ij\bar{i}}$ is a binary parameter equal to 1 if node $\bar{i} \in \overline{N}_{\checkmark}/\overline{N}_D$ in the routing network is a counterpart to the disrupted location on link $(i,j) \in A'$ in the infrastructure network, and 0 otherwise. After the completion of the restoration process of each disrupted

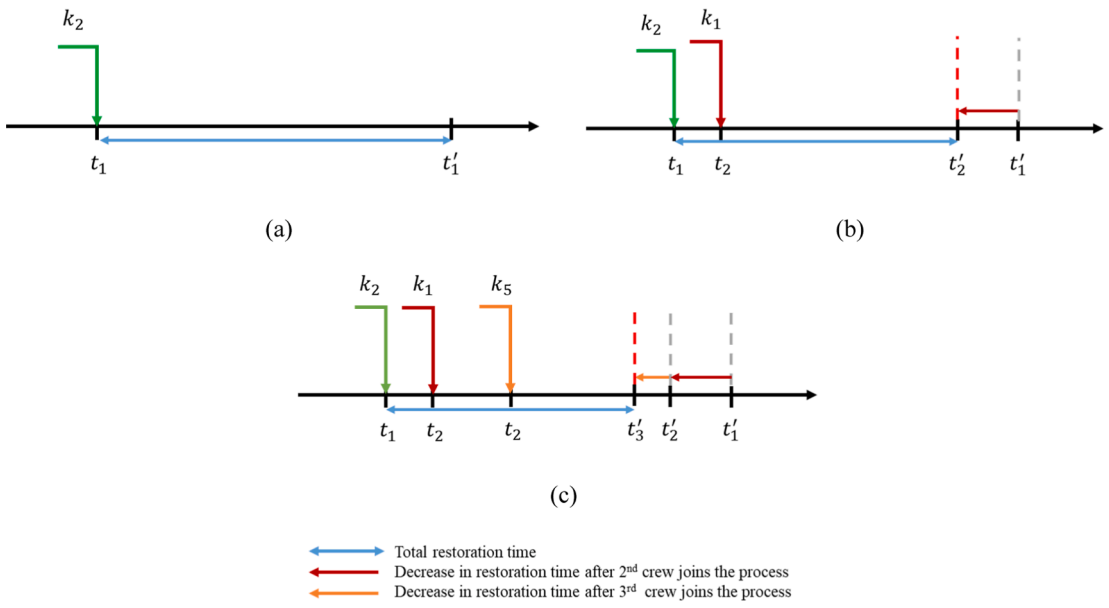

Fig. A1. Illustration of the dynamic restoration process when three crews are assigned to a disrupted location, (a) the first crew arrives at the disrupted location at time t1, (b) the second crew joins the process at time t2, and (c) the third crew joins the other two crews at time t3.

node $\bar{i} \in \overline{N}_{\diagup}/\overline{N}_D$ in the routing network, its counterpart link $(i,j) \in A^{'}(\theta_{ij\bar{i}} = 1)$ returns to its fully operational status in the infrastructure network. To formulate this fact mathematically, we first present a binary variable α_{ijt} that equals 1 if the restoration task on link $(i,j) \in A^{'}$ finishes at time t with the link becoming fully operational again, and otherwise 0. Then the model relates the operations on the routing network and the infrastructure network as follows: assume l restoration crews are assigned to disrupted node $\bar{i} \in \overline{N}_{\diagup}/\overline{N}_D$ in the routing network and complete the restoration task of that corresponding location at time $\beta_{\bar{i}}^{l}$. In the infrastructure network, the disrupted link $(i,j) \in A^{'}$, the counterpart of disrupted node $\bar{i} \in \overline{N}_{\diagup}/\overline{N}_D$, is recovered and becomes operational at the exact time $\beta_{\bar{i}}^{l}$. Therefore, binary variable α_{ijt} equals 1 at Time $t = \left[\beta_{\bar{i}}^{l}\right]$. Table A1 explains the notation in detail.

A.2. Dynamic restoration process

The traveling times on links in the routing network are heterogeneous and depending on the length of the links. Therefore, the arrival time of restoration crews assigned to node $\bar{i} \in \overline{N}_{A^{'}}$ are not necessarily the same. In this section, we explain how to calculate the processing time of each disrupted link $(i,j) \in A^{'}$, or its counterpart node $\bar{i} \in \overline{N}_{\diagup}/\overline{N}_D$, when l crews are assigned to it, yet they may not start their restoration tasks at the same time.

Fig. A1 illustrates the restoration process of the disrupted node $\bar{i} \in \overline{N}$, and how adding the next crew to the process affects the restoration time of node $\bar{i} \in \overline{N}_{\diagup}/\overline{N}_D$. After a disruptive event, node $\bar{i} \in \overline{N}_{\diagup}$, or its corresponding link $(i,j) \in A^{'}$ in network $G = (N,A)$, is considered disrupted if its post-disruption capacity ($u_{\bar{i}t_d}$ in network $\overline{G}$ or u_{ijt_d} in network G), is less than its pre-disruption capacity ($u_{\bar{i}t_0}$ in network or u_{ijt_0} in network G), and consequently $(u_{\bar{i}t_e} - u_{\bar{i}t_d})$ units of capacity must be restored. Each assigned crew arrives to node $\bar{i}$ at time $t = 1,\cdots,T$, which might be different from the arrival time of other assigned crews. In the mathematical model, to signal the sequential arrival of the assigned crews, we introduce $g_{\bar{i}t}^{l}$ as a binary variable equal to 1 if the lth crew arrives to node $\bar{i}$ at time t. When the first crew ($k_2 \in K$) arrives at node $\bar{i} \in \overline{N}_{\diagup}/\overline{N}_D$ at time t_1, $g_{\bar{i}t_1}^{1} = 1$, it starts the restoration process immediately. If the first crew is the only crew assigned to node $\bar{i}$, as shown in Fig. A1a, it completes the restoration process at time, $t_1^{'} = t_1 + p_{\bar{i}}^{1}$. One crew progresses through the restoration process of node $\bar{i} \in \overline{N}_{\diagup}/\overline{N}_D$ by $\lambda_{\bar{i}}^{1}$ percent of restoration work per time. Therefore, for node $\bar{i} \in \overline{N}_{\diagup}/\overline{N}_D$, there is $\lambda_{\bar{i}}^{1} p_{\bar{i}}^{1} = (u_{\bar{i}t_{\diagup}} - u_{\bar{i}t_d})$ amount of restoration work that must be accomplished until node $\bar{i}$ is fully operational. If another crew joins the restoration process, it must arrive before the completion of the restoration process, $g_{\bar{i}t_2}^{2} = 1$where $t_1 \le t_2 < t_1 + p_{\bar{i}}^{1}$. When the next crew ($k_1 \in K$) joins the process, shown in Fig. A1b, it accelerates the remaining restoration requirements, $\lambda_{\bar{i}}^{1}(t_1 + p_{\bar{i}}^{1} - t_2)$, by $\lambda_{\bar{i}}^{2}$ percent of restoration work per time, where $\lambda_{\bar{i}}^{2} > \lambda_{\bar{i}}^{1}$.

The required time for the remaining restoration task is $\frac{\lambda_{\bar{i}}^{1}}{\lambda_{\bar{i}}^{2}}\left(t_1 + p_{\bar{i}}^{1} - t_2\right)$, and the updated completion time of the restoration task to $t_2^{'} = t_2 + \frac{\lambda_{\bar{i}}^{1}}{\lambda_{\bar{i}}^{2}}\left(t_1 + p_{\bar{i}}^{1} - t_2\right)$. Again, if another crew joins the restoration process of node $\bar{i}$, as shown in Fig. A1c, it must arrive before the completion of the restoration process, $g_{\bar{i}t_3}^{3} = 1$where $t_2 \le t_3 < t_2 + \frac{\lambda_{\bar{i}}^{1}}{\lambda_{\bar{i}}^{2}}\left(t_1 + p_{\bar{i}}^{1} - t_2\right)$. When the next crew ($k_5$), joins the restoration process, it accelerates the remained restoration task, $\lambda_{\bar{i}}^{2}(t_2 + \frac{\lambda_{\bar{i}}^{1}}{\lambda_{\bar{i}}^{2}}\left(t_1 + p_{\bar{i}}^{1} - t_2\right) - t_3)$, by the rate of $\lambda_{\bar{i}}^{3}$, where $\lambda_{\bar{i}}^{3} > \lambda_{\bar{i}}^{2}$. The updated remained restoration time is $\frac{\lambda_{\bar{i}}^{2}}{\lambda_{\bar{i}}^{3}}\left(t_2 + \frac{\lambda_{\bar{i}}^{1}}{\lambda_{\bar{i}}^{2}}\left(t_1 + p_{\bar{i}}^{1} - t_2\right) - t_3\right)$, and its corresponding completion time is $t_3^{'} = t_3 + \frac{\lambda_{\bar{i}}^{2}}{\lambda_{\bar{i}}^{3}}\left(t_2 + \frac{\lambda_{\bar{i}}^{1}}{\lambda_{\bar{i}}^{2}}\left(t_1 + p_{\bar{i}}^{1} - t_2\right) - t_3\right)$. In general, the completion time of the

restoration process of node $\bar{i}$ when l crews are assigned to it is calculated with Eq. (A.1), where t_j is the arrival time of the jth crew to node $\bar{i}$, and therefore t_l is the arrival time of the last (lth) crew, and $\lambda_{\bar{i}}^{j}$ is the rate of restoration after the jth crew joins the restoration process.

$$\beta_{\bar{i}}^{l} = t_l + \frac{\lambda_{\bar{i}}^{1} p_{\bar{i}}^{1} + \sum_{j=1}^{l-1} \lambda_{\bar{i}}^{j} t_j - \sum_{j=2}^{l} \lambda_{\bar{i}}^{j-1} t_j}{\lambda_{\bar{i}}^{l}} \tag{A.1}$$

We implement the dynamic restoration process into the model by defining $\beta_{\bar{i}}^{l}$ in Eq. (A.2), where $\sum_{t=1}^{T} tg_{\bar{i}t}^{j}$ is the arrival time of jth crew to Node $\bar{i}$. Each restoration task on each link should be processed without interruption. All assigned crews work on a disrupted location until its restoration process is completed.

$$\beta_{\bar{i}}^{l} = \sum_{t=1}^{T} tg_{\bar{i}t}^{l} + \frac{\lambda_{\bar{i}}^{1} p_{\bar{i}}^{1} + \sum_{j=1}^{l-1} \lambda_{\bar{i}}^{j} \sum_{t=1}^{T} tg_{\bar{i}t}^{j} - \sum_{j=2}^{l} \lambda_{\bar{i}}^{j-1} \sum_{t=1}^{T} tg_{\bar{i}t}^{j}}{\lambda_{\bar{i}}^{l}} \tag{A.2}$$

Note two important conditions: First, the restoration time of a disrupted node might end before the arrival of some assigned restoration crews. The model eliminates the redundant routes to already recovered nodes by rescheduling the late crews to the nearest unrecovered or under recovery nodes. Second, the addition of more crews does not necessarily shorten the restoration time of a disrupted node. The model eliminates excessive crew assignment by distributing ineffective crews to the nodes capable being restored faster as the number of assigned crews increases. Both strategies are developed through the effort of the model to maximize the satisfied demand in each time period of restoration horizon by restoring the maximum number of disrupted nodes in the minimum time horizon. For the first condition, this requires minimizing the time when en route idle crews visiting already restored disrupted nodes. For the second condition, this requires assigning idle crews to ongoing restoration tasks.

A.3. Mathematical model

Morshedlou et al. (2019) presented a mixed integer formulation for the restoration routing problem. The objective is to maximize the cumulative enhancement in the infrastructure network resilience in each period by measuring the restoration progress of the network (i.e., the total flow reaching demand nodes in each time period after a disruption).

$$\max \sum_{t=1}^{T} Ð_{\varphi}(t|\not{\prec}) \tag{A.3}$$

$$\sum_{k\in K} \sum_{\bar{j}:\bar{i}\in \overline{N}_{\not{\prec}}/\overline{N}_D:(\bar{i},\bar{j})\in \overline{A}} x_{\bar{i}\bar{j}}^{k} = v_{\bar{i}} \quad \forall \bar{i} \in \overline{N}_D \tag{A.4}$$

$$\sum_{\bar{i}:\bar{i}\in \overline{N}_{\not{\prec}}/(n+1):(\bar{i},\bar{j})\in \overline{A}} x_{\bar{i}\bar{j}}^{k} \le 1 \quad \forall \bar{j} \in \overline{N}_{\not{\prec}}/\overline{N}_D \cup (n+1), k \in K \tag{A.5}$$

$$\sum_{\bar{j}:\bar{i}\in \overline{N}_{\not{\prec}}/\overline{N}_D:(\bar{i},\bar{j})\in \overline{A}} x_{\bar{i}\bar{j}}^{k} \le 1 \quad \forall \bar{i} \in \overline{N}_{\not{\prec}}/(n+1), k \in K \tag{A.6}$$

$$\sum_{\bar{i}:\bar{i}\in \overline{N}_{\not{\prec}}:(\bar{i},\bar{j})\in \overline{A}} x_{\bar{i}(n+1)}^{k} = 1 \quad k \in K \tag{A.7}$$

$$\sum_{\bar{i}:\bar{i}\in \overline{N}_{\not{\prec}}/(n+1):(\bar{i},\bar{j})\in \overline{A}} x_{\bar{i}\bar{j}}^{k} - \sum_{\bar{i}:\bar{i}\in \frac{\overline{N}_{\not{\prec}}}{\overline{N}_D}\cup(n+1):(i,j)\in \overline{A}} x_{\bar{j}\bar{i}}^{k} = 0 \quad \forall \bar{j} \in \overline{N}_{\not{\prec}}/\overline{N}_D, k \in K \tag{A.8}$$

Eqs. (A.4)–(A.8) are crew routing flow balance equations. Eq. (A.3) ensures that $v_{\bar{i}}$ crews are dispatched from depot $\bar{i} \in \overline{N}_D$. Eqs. (A.5)–(A.6) and guarantee than each crew $k \in K$ visits each disrupted link $(i,j) \in A$', or its counterpart node $\bar{i} \in \overline{N}_{\not{\prec}}/\overline{N}_D$, at most once. All crews finish their route by reaching to dummy sink node $(n+1)$, according to Eq. (A.7). Flow balance is incorporated into the model with Eq. (A.8), by which the number of arrivals to each node $\bar{i} \in \overline{N}_{A'}$ must equal the number of departures from it.

$$\sum_{\bar{i}:\bar{i}\in \overline{N}_{\not{\prec}}/(n+1):(\bar{i},\bar{j})\in \overline{A}} x_{\bar{i}\bar{j}}^{k} = \sum_{t=1}^{T} \tau_{\bar{j}t}^{k} \quad \forall \bar{j} \in \overline{N}_{\not{\prec}}/\overline{N}_D, k \in K \tag{A.9}$$

$$\sum_{k\in K} \sum_{t=1}^{T} \tau_{\bar{i}t}^{k} = \sum_{l=1}^{L} l z_{\bar{i}}^{l} \quad \forall \bar{i} \in \overline{N}_{\not{\prec}}/\overline{N}_D, l = 1, \cdots, L \tag{A.10}$$

$$\sum_{l=1}^{L} z_{\bar{i}}^{l} \le 1 \quad \forall \bar{i} \in \overline{N}_{\not{\prec}}/\overline{N}_D \tag{A.11}$$

In Eq. (A.9), only the crews that are scheduled to restore node $\bar{i} \in \overline{N}_{\not{\prec}}/\overline{N}_D$ visit that corresponding node. Eq. (A.10) ensures that if crews are assigned to node $\bar{i} \in \overline{N}_{\not{\prec}}/\overline{N}_D$ then they must visit that node during the restoration time horizon. Eq. (A.11) requires that the number of crews assigned to each disrupted node $\bar{i} \in \overline{N}_{\not{\prec}}/\overline{N}_D$ cannot be changed during the restoration process.

$$\sum_{t=1}^{T} t\tau_{\bar{j}t}^{k} \ge c_{\bar{i}\bar{j}} + \beta_{\bar{i}}^{l} - M(1 - x_{\bar{i}\bar{j}}^{k}) - M\left(1 - z_{\bar{i}}^{l}\right) \quad \forall \bar{i} \in \overline{N}_{\not{\prec}}/\overline{N}_D, l = 1, \cdots, L, k \in K \tag{A.12}$$

$$\sum_{k\in K} \tau_{\bar{i}t}^{k} = \sum_{l=1}^{L} g_{\bar{i}t}^{l} \quad \forall \bar{i} \in \overline{N}_{\not{\prec}}/\overline{N}_D, t = 1, \cdots, T \tag{A.13}$$

$$\sum_{t=1}^{T} tg_{\bar{i}t}^{l+1} \geq \sum_{t=1}^{T} tg_{\bar{i}t}^{l} - M\left(1 - \sum_{\bar{l}=l+1}^{L} z_{\bar{i}}^{\bar{l}}\right) \quad \forall \bar{i} \in \overline{N}_{\nearrow}/\overline{N}_D, l = 2, \cdots, L \tag{A.14}$$

$$\sum_{t=1}^{T} tg_{\bar{i}t}^{l+1} \leq \beta_{\bar{i}}^{l} + M\left(1 - \sum_{\bar{l}=l+1}^{L} z_{\bar{i}}^{\bar{l}}\right) \quad \forall \bar{i} \in \overline{N}_{\nearrow}/\overline{N}_D, l = 1, \cdots, L \tag{A.15}$$

Eqs. (A.12)–(A.15) calculate the dynamic restoration process. Eq. (A.12) calculates the arrival time of restoration crews scheduled to each node $\bar{i} \in \overline{N}_{\nearrow}/\overline{N}_D$ and eliminate the existence of infeasible routes including subtours. Eq. (A.13) sequences the assigned crews to each node $\bar{i} \in \overline{N}_{A^{'}}$. If crew $k \in K$ is visiting node $\bar{i} \in \overline{N}_{A^{'}}$, its arrival time is put in a set of sequenced arrival time of crews assigned to that node. Eq. (A.13) ensures that the total number of crews assigned to each node $\bar{i} \in \overline{N}_{\nearrow}/\overline{N}_D$ is equal to the total number of sequenced arrival times associated with that node. Eqs. (A.14)–(A.15) defines the time windows in which the second, third, and finally lth crew must arrive if l crews, $l = 1, \cdots, L$, are scheduled to node $\bar{i} \in \overline{N}_{\nearrow}/\overline{N}_D$.

$$\sum_{j:(i,j)\in A} f_{ijt} - \sum_{j:(j,i)\in A} f_{jit} \leq o_i \quad \forall i \in N_+, t = 1, \cdots, T \tag{A.16}$$

$$\sum_{j:(i,j)\in A} f_{ijt} - \sum_{j:(j,i)\in A} f_{jit} = 0 \quad \forall i \in N_=, t = 1, \cdots, T \tag{A.17}$$

$$\sum_{j:(i,j)\in A} f_{ijt} - \sum_{j:(i,j)\in A} f_{jit} = -\varphi_{it} \quad \forall i \in N_-, t = 1, \cdots, T \tag{A.18}$$

$$0 \leq \varphi_{it} \leq b_i \quad \forall i \in N_-, t = 1, \cdots, T \tag{A.19}$$

$$u_{ijt_d} \leq f_{ijt} \leq u_{ijt_0} \quad \forall (i,j) \in A, t = 1, \cdots, T \tag{A.20}$$

Eqs. (A.16)–(A.20) are network flow balance constraints that send flow from supply nodes, through transshipment nodes, to the demand nodes and control the flow over the entire network. Eq. (A.16) ensures that the flow sent from each supply node $i \in N_+$ does not exceed its capacity. When network flow enters a transshipment node, Eq. (A.17) requires that it must leave that node to reach demand nodes in subsequent steps. Eq. (A.18) calculates the amount of flow reaching each demand Node $i \in N_-$. Eq. (A.19) ensures that the flow reaching to each demand node will not exceed the capacity of that demand node. In Eq. (A.20), the flow on each Link $(i,j) \in A$ does not exceed the capacity of that link.

$$u_{ijt_d} \leq f_{ijt} \leq \sum_{s=1}^{t} \alpha_{ijs} u_{ijt_0} \quad \forall (i,j) \in A^{'}, t = 1, \cdots, T \tag{A.21}$$

$$\sum_{s=1}^{T} \alpha_{ijs} \leq 1 \quad \forall (i,j) \in A\text{'} \tag{A.22}$$

$$\sum_{s=1}^{T} s\alpha_{ijs} \geq \beta_{\bar{i}}^{l} - M\left(1 - z_{\bar{i}}^{l}\right) - M\left(1 - \theta_{\bar{i}\bar{j}}\right) \quad \forall h \in \overline{N}_{\nearrow}/\overline{N}_D \forall (i,j) \in A^{'}, k \in K \tag{A.23}$$

$$z_{\bar{i}}^{l} = \{0,1\} \forall l = 1, \cdots, L, \quad \forall \bar{i} \in \overline{N}_{\nearrow}/\overline{N}_D \tag{A.24}$$

$$g_{\bar{i}t}^{l} = \{0,1\} \quad \begin{matrix} \forall \bar{i} \in \overline{N}_{\nearrow}/\overline{N}_D, k \in K \\ l = 1, \cdots, Lt = 1, \cdots, T \end{matrix} \tag{A.25}$$

$$x_{\bar{i}\bar{j}}^{k} = \{0,1\} \quad (\bar{i},\bar{j}) \in \overline{A}, k \in K \tag{A.26}$$

$$\varphi_{it} > 0 \quad i \in N_-, t = 1, \cdots, T \tag{A.27}$$

$$\alpha_{kijt} = \{0,1\}, f_{ijt} > 0 \quad k \in K, (i,j) \in A, t = 1, \cdots, T \tag{A.28}$$

Similar to Eq. (A.20), in Eq. (A.21), the flow on each link $(i,j) \in A^{'}$, whether disrupted or restored, does not exceed the capacity of that link. Eq. (A.22) ensures that each disrupted link receives restoration services no more than once. Eq. (A.23) relates the routing network ($\overline{G}$) operations to the progress in the infrastructure network performance, G. This equation calculates the time when each disrupted link $(i,j) \in A$' becomes fully operational after the completion of its restoration process. The formulation has $O(|K| \times |N|^4 \times |T|)$ binary decision variables.

Appendix B

B.1. DBSCAN-based Mapping and Solution Algorithm

Algorithm 1. (*DBSCAN-based Mapping and Solution Algorithm*)

Input:

$l_v \in \mathscr{L}, v \in \overline{N}_{A^{'}_{\nearrow}}$ → Spatial location of disrupted components

$l_o \in \mathscr{L}, o \in \overline{N}_D$ → Spatial location of depots

f_{hijt_0} → Predisruption flow on component$(h,j) \in A^{'}_{\nearrow}$

u_{hijt_d} → Postdisruption capacity of component$(h,j) \in A^{'}_{\nearrow}$

(*continued on next page*)

(*continued*)

Solve DBSCAN ($\overline{G} = \left(\overline{N}, \overline{A}_{\swarrow}\right), \mathscr{L}, \varepsilon, MinPts$)

$N_\varepsilon(i) : \{j | d(i,j) \le \varepsilon\} \rightarrow$ the epsilon neighborhood is the disrupted locations within a radius of ε from an object

$d(i,j)$ is the distance between disrupted location $(i,j) \in \overline{A}_{\swarrow}$

MinPts→ the epsilon neighborhood of an object contains at least *MinPts* of disrupted locations

$\mathbb{C}_i = \{\ \} \rightarrow$ an empty cluster

for each $n \in \overline{N}$ **do**:

  mark n as visited

  $N_\varepsilon(n) = \{m | d(n,m) \le \varepsilon\} \cup \{n\}$

  if $|N_\varepsilon(n)| < MinPts$ **do**:

  mark n as NOISE

  else:

  Open $\mathbb{C}_i$as the next cluster

  $\mathbb{C}_i \leftarrow n$

  for each disrupted location $m \in \overline{N}$ in $N_\varepsilon(n)$ **do**:

  if m is not visited **do**:

  mark m as visited

  $N_\varepsilon(m) = \{h | d(m,h) \le \varepsilon\} \cup \{m\}$

  if $|N_\varepsilon(m)| \ge MinPts$ **do**:

  $N_\varepsilon(n) \leftarrow N_\varepsilon(n) \cup N_\varepsilon(m)$

  if m is not a member of any cluster **do**:

  $\mathbb{C}_i \leftarrow m$

for each cluster, $\mathbb{C}_i \in$ DBSCAN($\overline{G} = \left(\overline{N}, \overline{A}_{\swarrow}\right), \mathscr{L}$), **do**

$I_{\mathbb{C}_i} = \sum_{(h,j)\in\mathbb{C}_i} \frac{f_{hjt=0}}{u_{hjt_d}}$ (The aggregate importance measure of disrupted components in each cluster)

$D_{\mathbb{C}_i} = \frac{\sum_{v\in\overline{N}_{A^{\cdot}}, v\approx(h,j),(h,j)\in\mathbb{C}_i} \sum_{o\in\overline{N}_D} d(l_v, l_o)}{\left|\overline{N}_D\right| \times \left|\overline{N}_{A^{\cdot}_{\swarrow}}\right|}$ (The average distance of each cluster from depots $o \in \overline{N}_D$)

$I^{Total}_{\mathbb{C}_i} = \frac{I_{\mathbb{C}_i}}{D_{\mathbb{C}_i}}$ (The priority index associated with each cluster $\mathbb{C}_i$)

end for

Sort clusters by $D_{\mathbb{C}_i}$ in non-dicreasing order and put them in a list of listsD^{sort}

Add Eq. (12) to the formulation and solve the relaxed model

Represent the output of the relaxed model as:

$A = \left\{\tilde{i} : [\gamma_{k\tilde{i}t}, k, t] \middle| \gamma_{k\tilde{i}t} = \alpha_{k\tilde{i}t} - p_{k\tilde{i}} + 1, \alpha_{k\tilde{i}t} = 1, k = 1, \cdots, |K|, \tilde{i} \in N_{A^{\cdot}}, t = 1, \cdots, T\right\}$

A includes the disrupted locations, their corresponding No. of assigned crews, k, and the time their restoration time begins, t

$L = \{k : i | k = 1, \cdots, |K|, i \in \overline{N}_D\}$ where L is the last updated location of each crew

$$c = \begin{array}{c} \\ D_1 \\ \vdots \\ D_{\left|\overline{N}_D\right|} \\ 1 \\ \vdots \\ |\overline{N}_{A^{\cdot}}| \end{array} \begin{bmatrix} \cdots & c_{D_1 D}\big|_{\overline{N}_D}\big| & c_{D_1 1} & \cdots & c_{D_1|\overline{N}_{A^{\cdot}}|} \\ \ddots & \vdots & \vdots & \ddots & \vdots \\ \cdots & c_{D_1 D}\big|_{\overline{N}_D}\big| & c_{D\left|_{\overline{N}_D}\right|1} & \cdots & c_{D\left|_{\overline{N}_D}\right||\overline{N}_{A^{\cdot}}|} \\ \cdots & c_{1D}\big|_{\overline{N}_D}\big| & c_{11} & \cdots & c_{1|\overline{N}_{A^{\cdot}}|} \\ \ddots & \vdots & \vdots & \ddots & \vdots \\ \cdots & c_{|\overline{N}_{A^{\cdot}}|D}\big|_{\overline{N}_D}\big| & c_{|\overline{N}_{A^{\cdot}}|1} & \cdots & c_{|\overline{N}_{A^{\cdot}}|D}\big|_{\overline{N}_D}\big| \end{bmatrix}$$

$D_i i \in \overline{N}_D$ is the index of depots from where the crews start their routes

$|K|$ is the total number of crews

c is the traveling time between each pair of location in $\overline{N}_{A^{\cdot}} \cup \overline{N}_D$.

Min Distance-Earliest Release Procedure

$\overline{L}_{idle} \leftarrow \overline{L}_{idle} \subseteq L$ default dictionary of the location of idle crews

$\overline{L}_{busy} \leftarrow \overline{L}_{busy} \subseteq L$ default dictionary of the location of busy crews

$\overline{L}_{idle} \leftarrow \overline{L}_{idle} \subseteq L$ default dictionary of the location of idle crews

$\Lambda \leftarrow \{\tilde{i} : [\Im_{\tilde{i}}, \mathscr{B}_{\tilde{i}}, \mathscr{F}_{\tilde{i}}, \Re_{\tilde{i}}] | \tilde{i} \in \overline{N}_{A^{\cdot}}, the\ time\ the\ process\ \Im_{\tilde{i}} = starts, \mathscr{B}_{\tilde{i}} = proceeds, \mathscr{F}_{\tilde{i}} = ends, \Re_{\tilde{i}} = number\ of\ assigned\ crews\}$

$\mathscr{H} \leftarrow$ default list of lists with $1, \cdots, |K|$ lists each $\mathscr{H}_k = \{h_{kl} | the\ lth\ node\ in\ the\ schedule\ of\ crew\ k\}$

Temp ← empty dictionary

Sort γ_{kijt}based on t in non-increasing order and update array A

$\overline{L}_{idle} \leftarrow$ L

$\overline{L}_{busy} \leftarrow L - \overline{L}_{idle}$

$\overline{K} \leftarrow |\overline{L}_{idle}|$

while A $\neq \varnothing$ **do**

  while $\overline{L}_{idle} \neq \varnothing$ **do**

  Find the next$l$ where $\left|\overline{L}_{idle}\right| \ge A[l][1]$

  for $k = 1, \cdots, A[l][1]$ **do**

  Finds the next crew with $\min_{j\in\overline{L}_{idle}[k]} \mathscr{F}_j + c_{j,\text{list}(A.keys())[l]}$

  if $k = 1$ **do**

  $\Re_{\tilde{i}} \leftarrow 1$

(*continued on next page*)

(*continued*)

48: $\mathscr{H}_{\kappa} \leftarrow \bar{i}$
49: $\Im_{\bar{i}} \leftarrow \mathscr{T}_j + c_{j\bar{i}}$
50: $\mathscr{B}_{\bar{i}} \leftarrow p_{k\bar{i}}$
51: $\mathscr{T}_{\bar{i}} \leftarrow \Im_{\bar{i}} + p_{k\bar{i}}$
52: $\bar{L}_{busy} \leftarrow \bar{L}_{busy}$ + nextcrew: $\bar{i}$
53: $\bar{L}_{idle} \leftarrow \bar{L}_{idle}$ −nextcrew: $\bar{i}$
54: **else**
55: $\Im\_temp \leftarrow \mathscr{T}_j + c_{j\bar{i}}$

Elimination Procedure

56: **if** $\Im\_temp < \mathscr{T}_{\bar{i}}$ **do**
57: $\Re_{\bar{i}} \leftarrow 1 + \Re_{\bar{i}}$
58: $\mathscr{H}_{\kappa} \leftarrow \bar{i}$
59: $\mathscr{B}_{\bar{i}} \leftarrow \Im\_temp - \Im_{\bar{i}} + \left\lceil \frac{(\mathscr{T}_{\bar{i}} - \Im\_temp)\lambda_{\bar{i}\mathfrak{N}_{\bar{i}}-1}}{\lambda_{\bar{i}\mathfrak{N}_{\bar{i}}}} \right\rceil$
60: $\mathscr{T}_{\bar{i}} \leftarrow \Im\_temp + \left\lceil \frac{(\mathscr{T}_{\bar{i}} - \Im\_temp)\lambda_{\bar{i}\mathfrak{N}_{\bar{i}}-1}}{\lambda_{\bar{i}\mathfrak{N}_{\bar{i}}}} \right\rceil$
61: $\bar{L}_{busy} \leftarrow \bar{L}_{busy}$ −nextcrew: $\bar{i}$
62: $\bar{L}_{idle} \leftarrow \bar{L}_{idle}$ −nextcrew: $\bar{i}$
63: **end if**
64: **end if**
65: **end for**
66: $\Lambda\left[\bar{i}\right] \leftarrow [\Im_{\bar{i}}, \mathscr{B}_{\bar{i}}, \mathscr{T}_{\bar{i}}, \Re_{\bar{i}}]$
67: Temp ← Temp −Temp$[\bar{i}]$
68: A←A −A$[l]$
69: **end while**
70: $\bar{L}_{idle} \leftarrow \Lambda[\underset{\bar{i} \in \Lambda.keys(\ )}{\operatorname{argmin}}\ \Lambda\left[\bar{i}\right][2]][3]$
71: **Output** Completed dictionaries: Λ, and $\mathscr{H}$(Shown in Table 2)

Table C1
The original restoration routing problem solution for the power network in Shelby County, TN.

Ins.	⁄ (Number of disrupted links)	$\|K\|$	Original restoration routing problem		
			CPU	Gap(%)	T
1	M_6 (10)	2	8.1	0.07	57
2		3	9.77	0.03	49
3		4	354	0.01	153
4		5	3600	0.03	129
5	M_7 (15)	2	3600	0.3	78
6		3	3600	0.6	58
7		4	3600	3	120
8		5	3600	3	138
9	M_8 (28)	2	3600	3.5	137
10		3	3600	2.5	96
11		4	3600	3	87
12		5	3600	2.7	61
13		6	3600	4.14	166
14		7	3600	12.1	191
15	M_9 (30)	2	3600	12.03	184
16		3	3600	7.76	180
17		4	3600	18.31	196
18		5	3600	7.3	187
19		6	3600	3	160
20		7	3600	19.1	171

B.1. Heuristic Algorithm for Restoration Crew Routing Problem

Algorithm 2. (*Heuristic Algorithm for Restoration Crew Routing Problem*)

Input: $B, \mathscr{T}, P$, and the solution of **Algorithm 1**
1: Apply the *Shorten Routes* procedure $\forall k \in K$.
2: Apply the *Change Routes* procedure $\forall k \in K$.
3: Apply the *Add Crew* procedure for utmost $\forall k \in K$.

(*continued on next page*)

(*continued*)

4: Apply the *Remove Crew* procedure $\forall k \in K$
5: any of *Shorten Routes, Add Crew,* and *Remove Crew, Change Routes* is applicable for the current solution.
If No: the current solution is the output of the *Heuristic Algorithm*
If Yes: check the application of at least one of steps 1,2,3, and 4 to provide a better upper bound for the current solution
7: Repeat step 5.

Table C2
The original restoration routing problem solution for the water network in Shelby County, TN.

Ins.	∕ (Number of disrupted links)	$\|K\|$	Original model		
			CPU	Gap(%)	T
21	M_6 (10)	2	3600	2.45	220
22		3	3600	2.48	137
23		4	3600	1.2	166
24		5	3600	3.28	208/284
25		6	3600	2.42	89/101
26		7	3600	1.67	193/213
27	M_7 (15)	2	3600	10.7	247
28		3	3600	11.3	211
29		4	3600	9.6	203
30		5	3600	10	184
31	M_8 (28)	2	3600	9.5	252
32		3	3600	10.1	222/252
33		4	3600	–	–
34		5	3600	–	–
35		6	3600	13	390
36		7	3600	14.1	381
37	M_9 (30)	2	3600	10.9	255
38		3	3600	9.3	219
39		4	3600	–	–
40		5	3600	–	–
41		6	3600	–	–
42		7	3600	–	–

Table C3
The original restoration routing problem solution for the gas network in Shelby County, TN.

Ins.	∕ (Number of disrupted links)	$\|K\|$	Original model		
			CPU	Gap (%)	*T*
43	M_6 (3)	2	8.1	0.07	57
44		3	9.77	0.03	49
45	M_7 (5)	2	354	0.01	153
46		3	3600	0.03	129
47		4	3600	0.3	78
48		5	3600	0.6	58
49	M_8 (8)	2	3600	3	120
50		3	3600	3	138
51		4	3600	3.5	137
52		5	3600	2.5	96
53		6	3600	3	87
54		7	3600	2.7	61
55	M_9 (10)	2	3600	4.14	166
56		3	3600	12.1	191
57		4	3600	12.03	184
58		5	3600	7.76	180
59		6	3600	18.31	196
60		7	3600	7.3	187

Appendix C

See Tables C1-C3.

References

Iloglu, S., & McLay, L. A. (2018). An integrated network design and scheduling problem for network recovery and emergency response. *Operations Research Perspectives, 5*, 218–231.

American Society of Civil Engineers (ASCE), 2017. Report Card for America's Infrastructure. <http://www.infrastructurereportcard.org/oklahoma/oklahoma-overview/>.

Akbari, V., & Salman, F. S. (2017). Multi-vehicle synchronized arc routing problem to restore post-disaster network connectivity. *European Journal of Operational Research, 257*(2), 625–640.

Akbari, V., Sadati, M. E. H., & Kian, R. (2021). A decomposition-based heuristic for a multicrew coordinated road restoration problem. *Transportation Research Part D: Transport and Environment, 95*, Article 102854.

Aksu, D. T., & Özdamar, L. (2014). A mathematical model for post-disaster road restoration: Enabling accessibility and evacuation. *Transportation Research Part E: Logistics and Transportation Review, 61*, 56–67.

Arab, A., Khodaei, A., Han, Z., & Khator, S. K. (2015). Proactive recovery of electric power assets for resiliency enhancement. *IEEE Access, 3*, 99–109.

Atsiz, E., Balcik, B., Gunnec, D., & Sevindik, B. U. (2021). A coordinated repair routing problem for post-disaster recovery of interdependent infrastructure networks. *Annals of Operations Research*, 1–31.

Barker, K., Lambert, J. H., Zobel, C. W., Tapia, A. H., Ramirez-Marquez, J. E., Albert, L., & Caragea, C. (2017). Defining resilience analytics for interdependent cyber-physical-social networks. *Sustainable and Resilient Infrastructure, 2*(2), 59–67.

Behzadi, G., O'Sullivan, M. J., & Olsen, T. L. (2020). On metrics for supply chain resilience. *European Journal of Operational Research, 287*(1), 145–158.

Bienstock, D., & Mattia, S. (2007). Using mixed-integer programming to solve power grid blackout problems. *Discrete Optimization, 4*(1), 115–141.

Carvalho, H., Barroso, A. P., Machado, V. H., Azevedo, S., & Cruz-Machado, V. (2012). Supply chain redesign for resilience using simulation. *Computers & Industrial Engineering, 62*(1), 329–341.

Caunhye, A. M., Aydin, N. Y., & Duzgun, H. S. (2020). Robust post-disaster route restoration. *OR Spectrum, 42*(4), 1055–1087.

Çelik, M. (2017). Network restoration and recovery in humanitarian operations: Framework, literature review, and research directions. *Surveys in Operations Research and Management Science., 21*, 47–61.

Çelik, M., Ergun, Ö., & Keskinocak, P. (2015). The post-disaster debris clearance problem under incomplete information. *Operations Research, 63*(1), 65–85.

D'Ambrosio, V., & Leone, M. F. (2015). Climate change risks and environmental design for resilient urban regeneration. Napoli Est pilot case. *TECHNE-Journal of Technology for Architecture and Environment*, 130–140.

Davis, K. F., Downs, S., & Gephart, J. A. (2021). Towards food supply chain resilience to environmental shocks. *Nature Food, 2*(1), 54–65.

Dixit, V., Verma, P., & Tiwari, M. K. (2020). Assessment of pre and post-disaster supply chain resilience based on network structural parameters with CVaR as a risk measure. *International Journal of Production Economics, 227*, 107655. environmental shocks. Nature Food, 2(1), 54–65.

Duque, P. A. M., Dolinskaya, I. S., & Sörensen, K. (2016). Network repair crew scheduling and routing for emergency relief distribution problem. *European Journal of Operational Research, 248*(1), 272–285.

Faturechi, R., & Miller-Hooks, E. (2014). Measuring the performance of transportation infrastructure systems in disasters: A comprehensive review. *Journal of Infrastructure Systems, 21*(1), 4001–4025.

Ghorbani-Renani, N., González, A. D., Barker, K., & Morshedlou, N. (2020). Protection-interdiction-restoration: Tri-level optimization for enhancing interdependent network resilience. *Reliability Engineering & System Safety, 199*, 106907. https://doi.org/10.1016/j.ress.2020.106907

González, A. D., Chapman, A., Dueñas-Osorio, L., & Mesbahi, M. (2017). Efficient infrastructure restoration strategies using the recovery operator. *Computer-Aided Civil and Infrastructure Engineering, 32*(12), 991–1006.

González, A. D., Dueñas-Osorio, L., Sánchez-Silva, M., & Medaglia, A. L. (2016). The interdependent network design problem for optimal infrastructure system restoration. *Computer-Aided Civil and Infrastructure Engineering, 31*(5), 334–350.

Henry, D., & Ramirez-Marquez, J. E. (2012). Generic metrics and quantitative approaches for system resilience as a function of time. *Reliability Engineering and System Safety, 99*, 114–122.

Hishamuddin, H., Sarker, R. A., & Essam, D. (2013). A recovery model for a two-echelon serial supply chain with consideration of transportation disruption. *Computers & Industrial Engineering, 64*(2), 552–561.

Kasaei, M., & Salman, F. S. (2016). Arc routing problems to restore connectivity of a road network. *Transportation Research Part E: Logistics and Transportation Review, 95*, 177–206.

Kim, S., Shin, Y., Lee, G. M., & Moon, I. (2018). Network repair crew scheduling for short-term disasters. *Applied Mathematical Modelling, 64*, 510–523.

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. *Technological Forecasting and Social Change, 77*(6), 960–974.

Li, Y., Zobel, C. W., Seref, O., & Chatfield, D. (2020). Network characteristics and supply chain resilience under conditions of risk propagation. *International Journal of Production Economics, 223*, 107529. https://doi.org/10.1016/j.ijpe.2019.107529

Mao, X., Lou, X., Yuan, C., & Zhou, J. (2020). Resilience-based restoration model for supply chain networks. *Mathematics, 8*(2), 163.

Moreno, A., Alem, D., Gendreau, M., & Munari, P. (2020a). The heterogeneous multicrew scheduling and routing problem in road restoration. *Transportation Research Part B: Methodological, 141*, 24–58.

Morshedlou, N., González, A. D., & Barker, K. (2018). Work crew routing problem for infrastructure network restoration. *Transportation Research Part B: Methodological, 118*, 66–89.

Moreno, A., Munari, P., & Alem, D. (2020b). Decomposition-based algorithms for the crew scheduling and routing problem in road restoration. *Computers & Operations Research, 119*, 104935. https://doi.org/10.1016/j.cor.2020.104935

Morshedlou, N., Barker, K., & Sansavini, G. (2019). Restorative capacity optimization for complex infrastructure networks. *IEEE Systems Journal, 13*(3), 2559–2569.

Nan, C., & Sansavini, G. (2017). A quantitative method for assessing resilience of interdependent infrastructures. *Reliability Engineering and System Safety, 157*, 35–53.

Nurre, S. G., & Sharkey, T. C. (2014). Integrated network design and scheduling problems with parallel identical machines: Complexity results and dispatching rules. *Networks, 63*(4), 306–326.

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. *European Journal of Operational Research, 223*(3), 794–806.

Ouyang, M., & Fang, Y. (2017). A mathematical framework to optimize critical infrastructure resilience against intentional attacks. *Computer-Aided Civil and Infrastructure Engineering, 32*(11), 909–929.

Özdamar, L., Tüzün Aksu, D., & Ergüneş, B. (2014). Coordinating debris cleanup operations in post disaster road networks. *Socio-Economic Planning Sciences, 48*(4), 249–262.

Pamukcu, D., & Balcik, B. (2020). A multi-cover routing problem for planning rapid needs assessment under different information-sharing settings. *OR Spectrum, 42*(1), 1–42.

Sanci, E., & Daskin, M. S. (2019). Integrating location and network restoration decisions in relief networks under uncertainty. *European Journal of Operational Research, 279*(2), 335–350.

Sanci, E., & Daskin, M. S. (2021). An integer L-shaped algorithm for the integrated location and network restoration problem in disaster relief. *Transportation Research Part B: Methodological, 145*, 152–184.

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. *European Journal of Operational Research, 244*(1), 309–321.

Shin, Y., Kim, S., & Moon, I. (2019). Integrated optimal scheduling of repair crew and relief vehicle after disaster. *Computers & Operations Research, 105*, 237–247.

Su, Z., Jamshidi, A., Núñez, A., Baldi, S., & De Schutter, B. (2019). Integrated condition-based track maintenance planning and crew scheduling of railway networks. *Transportation Research Part C: Emerging Technologies, 105*, 359–384.

Tan, Y., Qiu, F., Das, A. K., Kirschen, D. S., Arabshahi, P., & Wang, J. (2019). Scheduling post-disaster repairs in electricity distribution networks. *IEEE Transactions on Power Systems, 34*(4), 2611–2621.

Vugrin, E. D., & Camphouse, R. C. (2011). Infrastructure resilience assessment through control design. *International Journal of Critical Infrastructure, 7*(3), 240–260.