Integrated Reinforcement and Repair of Interdependent Infrastructure Networks Under Disaster-Related Uncertainties

Tuğçe Canbilen[1], Sakine Batun

Department of Industrial Engineering, Middle East Technical University, Universiteler Mh, Dumlupinar Bl. No:1, 06800 Cankaya, Ankara, Turkey

Melih Çelik

School of Management, University of Bath, Claverton Down Rd, Combe Down, Bath BA2 7AY, United Kingdom

Abstract

Natural or human-inflicted disasters may cause large-scale disruptions in the services of infrastructure networks including power, water, and telecommunication. Restoring the services of these infrastructures is vital in the aftermath of the disaster, so that search-and-rescue activities, relief transportation, and restoration efforts can be efficiently facilitated. On the other hand, operations of these infrastructures may depend on receiving services from one another, resulting in an interdependent network structure. Consequently, addressing the decisions of network reinforcement before the disaster and the repairs in its aftermath needs to take into account this interdependent structure, as well as the uncertainties arising from the timing, location, and magnitude of the disaster.

This paper introduces the Stochastic Interdependent Infrastructure Reinforcement and Repair Problem, which considers the pre-disaster reinforcement of interdependent network components and post-disaster repair scheduling in an integrated manner. In making these decisions, the uncertainty on which network components will be disrupted is incorporated into the problem definition. The problem is modelled using scenario-based two-stage stochastic programming. A heuristic based on a genetic algorithm and partial optimization is proposed to solve realistically-sized instances of the problem. Computational experiments not only show that the heuristic is able to find near-optimal solutions within reasonable times, but also illustrate the ability of the approach to help derive managerial insights.

Keywords: OR in societal problem analysis, disaster management, interdependent infrastructures, stochastic programming, genetic algorithms

1. Introduction

Disasters are sudden catastrophic events disrupting the operations of daily life and nature. They can be human-inflicted (e.g., terrorist attacks), or natural (e.g., earthquakes, and floods). In recent decades, both natural and human-inflicted disasters have increased in number and in intensity (Guha-Sapir et al., 2016). In some cases, multiple cascading disasters have exacerbated the relief efforts. Due to the upward trends in population, urbanization, global warming, and economic activity, disasters have had increasing impacts on human life and

[1]Corresponding author, e-mail: canbilen@metu.edu.tr

well-being, the global economy, and the environment (Yaghmaei et al., 2019). In light of this information, any effort to prevent or reduce these impacts and to provide relief is very valuable.

Services of the critical infrastructures are vital for the society to maintain everyday activities. These critical infrastructures can be classified under the four groups of transportation (e.g., highways), energy (e.g., power networks), telecommunication (e.g., the Internet), and water (e.g., wastewater) (Lee et al., 2007). Each of these infrastructures operate on their own structural elements. However, their operations are often *interdependent*, i.e., the activities performed by one infrastructure network may require receiving the service of another. For example, a power plant generating electricity from wastewater needs the wastewater network to function. A wastewater plant, in turn, would need the power network for electricity. Another example is that a wastewater infrastructure should receive electrical power to be functional to discharge the flood in a flooded highway infrastructure.

Large-scale disasters may have profound impact on critical infrastructures, affecting many users and requiring significant effort to return to normalcy. According to the U.S. Department of Energy, Office of Electricity Delivery and Energy Reliability, one week after Hurricane Sandy hit the USA in 2012, more than 650,000 customers were still out of power in the affected states (Heath et al., 2016). Furthermore, the failure of these infrastructures also hampers disaster response activities, including search-and-rescue, evacuation, or relief distribution. As an example, disruption on a power network may prevent search-and-rescue equipment from operating, while damage on road segments may make timely delivery of relief more difficult. Consequently, these services should be restored (repaired) as quickly as possible in the immediate aftermath of a disaster. Given the importance of timeliness in this stage, scheduling of repairs plays an important role in determining the effectiveness of response activities. Therefore, extant literature has given considerable attention to address the scheduling of repairs (e.g., Coffrin et al., 2012; Cavdaroglu et al., 2013).

A further step in improving infrastructure network resilience is to involve the preventive measures in the mitigation stage, so that critical infrastructures are disrupted at the smallest extent possible. As has been observed in the literature, mitigation activities not only help restore the services of critical infrastructures more rapidly, but also reduce the potential impact of subsequent disasters (Heath et al., 2016). Within the context of network resilience, this would correspond to incorporating the pre-disaster reinforcement decisions into the planning process for post-disaster restoration scheduling. To the best of our knowledge, despite the importance of this approach, there is no study that integrates infrastructure reinforcement decisions in interdependent networks with those of network repair and service distribution. The main contribution of this paper is to bridge this gap by formally defining this novel problem, formulating a mathematical model, developing a solution approach, and deriving managerial insights for it.

From a modeling perspective, incorporation of infrastructure reinforcement decisions in the pre-disaster stage implies that uncertainties in the disaster type, intensity, and location need to be considered. To the best of our knowledge, this paper also makes an important contribution to the literature by providing the first stochastic modeling and solution approaches for reinforcement, repair, and service delivery on interdependent networks.

This paper defines and studies the Stochastic Interdependent Infrastructure Reinforcement and Repair (SIIRR) problem. In doing so, we incorporate the inherent uncertainty of which network segments will be disrupted. In the first (pre-disaster) stage of the problem, reinforcement decisions are considered. In the second (post-disaster) stage, network flow and repair scheduling decisions are focused on. The objective of the SIIRR problem is to maximize the overall performance of the infrastructures throughout the planning horizon. At the same time, operational interdependencies between the nodes of different infrastructures and potential disaster scenarios are considered. We present a two-stage stochastic mixed-integer linear programming model formulation for the SIIRR problem. The formulation includes constraints related to reinforcement, repair scheduling, interdependency, and demand satisfaction. As practical-sized instances of the problem cannot be solved within a reasonable amount of time, a novel heuristic making use of a genetic algorithm and partial optimization is devised. The model and the heuristics are employed to gather managerial inferences and to provide sensitivity analyses for the decision-makers of the critical infrastructures. Our computational experiments show the substantial impact of considering network reinforcement decisions prior to the disaster on improving the service restoration performance in the aftermath of it. This has important practical implications by showing the decision makers the impact of advance planning and complements the findings in the literature on disaster management by underlining the importance of the mitigation phase within the context of interdependent network restoration.

The remainder of this paper is organized as follows. In Section 2, relevant literature for the problem environment is reviewed. In Section 3, the SIIRR problem is defined and the two-stage stochastic mixed-integer programming formulation is presented. The proposed solution methodology is explained in detail in Section 4. Computational experiments are discussed in Section 5, and Section 6 concludes the paper with the contributions of this study and points out future research directions.

2. Literature Review

Over the last decade, there has been a substantial increase in the interest and number of studies employing optimization tools and techniques on infrastructure network reinforcement and restoration. In this section, we classify our review of this stream into two groups: (i) studies focusing on the reinforcement and/or restoration of a single-infrastructure network, and (ii) those that consider multiple interdependent infrastructures. For comprehensive reviews, the interested reader is referred to Sharkey et al. (2021) for network resilience and reinforcement, and Çelik (2016) for network restoration.

2.1. Single-Infrastructure Reinforcement and Restoration Problems

In the network optimization literature, design and reinforcement of a network for resilience have been considered in various ways. Chimani et al. (2010), Botton et al. (2013, 2015) and Diarrassouba et al. (2016) focus on increasing the connectivity of the network in general. By designing the network to ensure multiple paths between any two nodes of the network, connectivity may be increased. Peeta et al. (2010), Du & Peeta (2014), Fan et al. (2010), Chang et al. (2012), and Lu et al. (2018) work on the network reinforcement problem that mainly aims to decide on which segments in the network to reinforce. Our work extends these studies to involve the post-disaster decisions of network restoration and service delivery as well. Studies integrating pre-attack mitigation activities, the attack and the post-attack response using a defender-attacker-defender model are reviewed in Sharkey et al.

(2021). An example on a single network is given in Ouyang & Fang (2017). Unlike these studies, the SIIRR problem takes place in a collaborative environment and multiple interdependent networks.

A vast majority of the literature on infrastructure restoration is focused on the restoration of a single critical infrastructure. For brevity, considering our problem environment, we limit our review to studies that consider the sequencing or scheduling of these restoration decisions or model uncertainty in the problem environment. Network construction problems (e.g., Averbakh, 2012; Averbakh & Pereira, 2012; Kalinowski et al., 2013) aim to restore damaged arcs of a transportation network by determining the sequence in which arcs should be added to the network. The Integrated Network Design and Scheduling (INDS) problem is defined by Nurre et al. (2012) to determine which arcs of a damaged infrastructure to repair over a time horizon to maximize the weighted flow from the supply to demand nodes. Nurre & Sharkey (2014) show that the INDS problem is $\mathcal{NP}$-hard under various objectives. Kalinowski et al. (2013) and Baxter et al. (2014) examine the bounds and properties of the optimal solutions of the INDS problem. Iloglu & Albert (2018) model the INDS problem to minimize the weighted distance between the emergency responders and the demand points. A number of studies (e.g., Maya Duque et al., 2016; Moreno et al., 2018; Akbari & Salman, 2017a,b; Ajam et al., 2019, 2021) consider the routing of restoration vehicles in addition to restoration scheduling. Compared to these studies, the SIIRR problem also involves the additional considerations of reinforcement, and disaster-related uncertainties, and the subsequent flow of services. Examples of studies considering disaster-related uncertainties while addressing the pre-disaster preparedness decisions together with post-disaster network repair and flows in an integrated manner using a two-stage stochastic programming approach are Aslan & Çelik (2019), Sanci & Daskin (2019), Fang & Sansavini (2018), Álvarez Miranda & Pereira (2017), Fang et al. (2019), Heath et al. (2016), Álvarez Miranda & Pereira (2017), and Fang et al. (2019). The work in this paper differs from these studies by the involvement of multiple interdependent infrastructures and integrated consideration of multiple operational aspects of such networks.

2.2. Reinforcement and Restoration of Interdependent Infrastructures

To the best of our knowledge, the first framework on interdependent infrastructures is given by Lee et al. (2007), who define an *input interdependency (operational interdependency)* as a node not being able to function until its demand from the infrastructures it depends on is satisfied (e.g., a wastewater plant not operating until receiving sufficient power). He & Cha (2020) define system-to-facility interdependencies as one network element depending on another infrastructure. The SIIRR problem involves both system-to-facility and input interdependencies. Sharkey et al. (2015a) define restoration interdependencies, where the restoration of an infrastructure requires that of another one (e.g., restoring the electricity of a critical region before flood can be re-channeled to sewers with the help of pumps). Loggins et al. (2019) focus on interdependencies between the critical civil infrastructures and social infrastructures such as healthcare, police, fire-fighting and education services.

Part of the literature on interdependent infrastructures focuses on the restoration of the networks, ignoring the detailed repair schedules. Among these, González et al. (2016) aim to find the subset of nodes and arcs to be reconstructed with minimum cost subject to resource, operational and interdependency constraints. Taking into account the repair horizon, the lack of information among the repair crews during the restoration activities is modeled by Talebiyan & Dueñas-Osorio (2019, 2020) by comparing different assumption patterns about the restoration efforts of other infrastructures. An interdependent multi-layered network flow model deciding on the components that should be operational in a network and the service flow is proposed in Enayaty Ahangar et al. (2020). Coffrin et al. (2012) consider the last-mile restoration of interdependent power and gas infrastructures to determine the order of the repairs. The SIIRR problem further extends these studies to involve infrastructure reinforcement, disaster-related uncertainties, and a more detailed scheduling approach.

The Interdependent Integrated Network Design and Scheduling (IINDS) problem also considers the scheduling decisions during infrastructure repairs (Cavdaroglu et al., 2013). The objective in the IINDS problem is to minimize the unmet demand of the demand nodes as well as the restoration and operation costs. Sharkey et al. (2015b) extend the IINDS by incorporating restoration interdependency constraints and information sharing with the objective of maximizing the amount of infrastructure services relative to their disaster-free levels. Other studies on information sharing during the repair of interdependent infrastructures include Lee & Rao (2009) and Smith et al. (2020). The IINDS problem is also addressed under multiple objectives in Sharma et al. (2019, 2020), Gomez et al. (2019), Karakoc et al. (2019, 2020), and Almoghathawi & Barker (2019). Our work extends these studies by incorporating the reinforcement decisions and considering the uncertainty of the problem environment.

A number of studies model the inherent uncertainty in the problem environment of interdependent infrastructure restoration. Random arc disruptions are introduced and a two-stage stochastic program is proposed by Shen (2013). The first-stage problem decides on the arcs to be built, while the second-stage determines the network flows. A similar problem structure is considered in a defender-attacker model in Ouyang (2017), which is extended by Fang & Zio (2019) to incorporate the information on various disasters into the model, and by Ghorbani-Renani et al. (2020) to involve different objectives for the defender and the attacker. He & Cha (2018) build a dynamic model to assess the resilience of the facility-to-facility type interdependent infrastructures (e.g., water pumping station depending on electricity pole) after a disaster, considering the uncertainty in the damage assessment of the facilities. Lastly, He & Cha (2020) incorporate different types of interdependencies into the model, as well as the uncertainty on the arc repair times and demand levels after a disaster. The SIIRR problem belongs to this class of network reinforcement and restoration problems, but also involves the additional considerations of integrated reinforcement and repair in a collaborative setting, as well as detailed scheduling of network repair at the same time.

2.3. Our Contributions

The main contribution of the work in this paper mainly arises from the definition of a novel problem with important practical implications. The SIIRR problem brings together four important characteristics of pre-disaster network reinforcement and post-disaster restoration which, to the best of our knowledge, have been considered in an integrated way for the first time: (i) operational interdependencies among interdependent infrastructures, (ii) the inherent uncertainty caused by the disaster on the functionality of the network elements, (iii) integrated reinforcement and the repair activities to maximize the performance of the infrastructures, and (iv) scheduling of the repair activities for multiple repair teams. The aforementioned studies in our literature review consider only subsets of these settings.

From a practical standpoint, the SIIRR problem allows the relevant decision makers (particularly local governments and infrastructure network operators) to understand and quantify the importance of incorporating pre-disaster reinforcement decisions into the post-disaster network restoration and service delivery process. Furthermore, it helps detect the critical components of networks, and therefore plays an important role in preparing the disaster preparedness plans as well as budget allocations for local communities. Lastly, its focus on network operators working together to prepare for and respond to the effects of the disaster further promotes coordination and collaboration among the different actors in the infrastructure network.

From a theoretical perspective, while the SIIRR problem is a novel one that better incorporates real-life settings and decisions compared to its counterparts, it is also a computationally challenging problem that requires a novel heuristic to find near-optimal solutions. We provide such an approach in Section 4 of this paper. To our best knowledge, this is the first use of a representative scenario within a genetic algorithm for a stochastic programming model. The high quality of the resulting solutions implies that such an approach could be helpful in tackling similar problems that can be modeled by two-stage stochastic programs.

3. Problem Definition and Mathematical Model

Motivated by the need to coordinate the reinforcement and repair operations, as well as the operations of multiple infrastructure networks, we define the Stochastic Interdependent Infrastructure Reinforcement and Repair (SIIRR) problem in this section. We first describe the environment in which the SIIRR problem takes place, followed by the modeling assumptions, culminating in the two-stage stochastic programming model.

3.1. Problem Environment

The SIIRR problem considers multiple interdependent infrastructure networks prior to and in the aftermath of a disaster. These networks may be operated by local municipalities (e.g., road, water, or wastewater networks) or by private companies (e.g., power or telecommunications networks). Each of these operators is responsible for ensuring that the recipients can receive these services with as little interruption as possible, by maintaining connectivity in their corresponding infrastructure networks. Prior to the disaster, each operator may reinforce a subset of their network elements using their available budget. These reinforcement decisions are aimed at preventing the disruption to the elements once the disaster strikes. In the aftermath of the disaster, the operators need to decide on the scheduling of repairs to the disrupted elements, subject to the available number of repair teams. Upon determining the schedules of its teams, each infrastructure operator directs these teams to the assigned network segments (e.g., power lines, water/wastewater pipes, fiber-optic cables) at the scheduled times. These repairs are aimed at establishing the connectivity between the supply points (e.g., power plants, water pumps, or telecommunication hubs) and the demand points (e.g., households or supply points of networks whose service depend on other infrastructures) in a timely manner.

Due to the uncertainty about the effects of the disaster, the reinforcement decisions need to be made without exact knowledge on which network elements will be disrupted. However, estimates on potential effects of various disaster scenarios on the infrastructure can be obtained by means of structural engineering models (e.g., Faturechi

& Miller-Hooks, 2015). Due to the fragmented structure of how these networks are typically managed, their reinforcement efforts are generally funded by the individual operators' own budgets. Post-disaster repair activities are also conducted by the repair teams specific to the infrastructures, due to the need for specialized expertise.

A central decision maker (e.g., the local or national government) is responsible for coordinating the reinforcement and repair operations of the interdependent infrastructures. This decision maker has access to the topographic structures, interdependency relationships, and data on supply and demand for each network. In the aftermath of the disaster, the main aim of the central decision maker is to restore the infrastructure network to its pre-disaster flow levels by re-establishing the connectivity between the supply and demand points of the networks. In doing so, time is of the essence, as these networks are critical in supporting the disaster response activities such as search-and-rescue and relief distribution, as well as maintaining the needs of their users.

3.2. Modeling Assumptions

To make the SIIRR problem tractable and to incorporate the aspects of the problem where data and information are not immediately available, we make a number of simplifying assumptions.

Without loss of generality, we consider damages on the segments of the infrastructure networks. Damages on the supply and demand locations as well as intermediate junctions can be easily incorporated by including an artificial segment between two copies of the potentially damaged points. We also assume *binary damage*, where following the disaster, a network segment either retains its full capacity or cannot be traversed at all by the services of the infrastructure. Partial damages on the segments can be modeled by adjusting the capacity of the segment in accordance with the amount of damage. Due to the availability of rapid damage assessment by means of satellite images or unmanned air vehicles, we assume that the set of damaged segments become known in the aftermath of the disaster, before the repair operations start. To prevent intractability due to decision-dependent uncertainties, we also assume that a reinforced arc is not damaged in the aftermath of the disaster.

We assume input interdependencies between the infrastructures (Lee et al., 2007), which also correspond to system-to-facility type of interdependencies defined in He & Cha (2020). However, we note that our models can be easily extended to involve other interdependency types (e.g., effectiveness or options precedence) as well.

Since timely restoration of services is important, the SIIRR problem involves multiple discrete time periods. The length of a period can be determined based on how long it takes for a repair team to finish the repair of a segment, which is generally in the order of a few hours or a whole work shift (e.g., 8 hours). The length of the repair horizon depends on the estimated time to re-establish connectivity, and would be expected to last a few days to one week following the disaster.

Repair teams of each infrastructure network are assumed to be capable of repairing any segment in their corresponding infrastructure. Any restrictions on this can be easily incorporated into the modeling approach. We prohibit the repair of segments by multiple teams, to keep the repair time estimates simple. This is also in line with the assumptions in the extant literature. Following similar assumptions in the literature, we assume that the repair time (in number of periods) of each segment can be accurately estimated before the disaster, that repair teams can move from one repair to another in negligible time, and that service flows are instantaneous

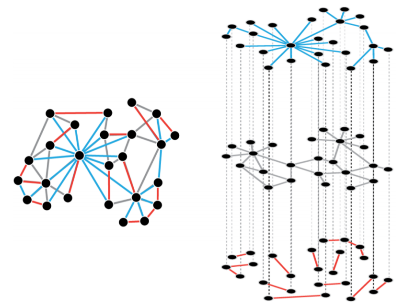

Figure 1: Layered network example (Krzywinski, 2010), where each layer represents a different infrastructure

once connection is established. The latter is also the case in reality for many infrastructures such as electricity and water, as the services are ready to be used in every household.

Each network segment has a certain flow capacity. To provide an uninterrupted service to the demand points once the connection is secured, we assume that once a flow is established in a period, this flow is maintained in the subsequent time periods. For a more flexible model, weights are introduced for the demand nodes to reflect their varying importance in the infrastructures. In the absence of relevant data, all weights can be assumed to be equal.

3.3. A Two-Stage Stochastic Program for the SIIRR Problem

We model the SIIRR problem on a network where the nodes correspond to the service supply, demand, or transshipment points in each infrastructure. Each network segment is represented by an arc. Following Lee et al. (2007), infrastructures are represented as directed layered networks. An example with three interdependent networks is given in Figure 1, in which the layers correspond to different infrastructures. As the figure also exemplifies, the same node may exist multiple infrastructures as different types of nodes. For example, a supply node in a water infrastructure may act as a demand node in the power infrastructure, since the water pump in that node may require power to be operated. Flows on the arcs represent the services of each infrastructure (e.g., arcs may be the pipelines for a water infrastructure). Interdependencies are represented by the connections across the infrastructures. Once the demand of an interdependent node is satisfied in one (higher-level) infrastructure, the node becomes active in the other (lower-level) infrastructure which depends on the higher-level one.

The SIIRR problem can be modeled as a two-stage stochastic mixed-integer program that maximizes the expected accumulated service provided over all periods. In the first-stage, reinforcement decisions are taken under budget limitations. In the second-stage, restoration activities are planned and a repair schedule is determined for repair teams for every possible disaster scenario. The service flows on each network for each period are determined based on the connectivity of these networks in the given period. These also determine the amounts of demand and interdependencies satisfied in that period.

Assuming a discrete set S of potential disaster scenarios, the two-stage stochastic programming model SIIRR-SP aims to maximize the expected performance of all infrastructures in set M over the planning horizon T. The network corresponding to infrastructure $m \in M$ consists of nodes N_m and arcs A_m. The node set is further classified into S_m, D_m and T_m, denoting the supply, demand and transshipment nodes, respectively. Repair crews in infrastructure are represented by K_m. Once the disaster hits, sets E_{ms} and $\bar{E}_{ms}$ denote the undamaged and damaged arcs in infrastructure $m \in M$ in scenario $s \in S$, respectively. We use set F_{mn} to include the pairs of nodes $i \in D_m$, $j \in N_n$, $m, n \in M$, such that node j has an input dependency on node i.

Each infrastructure $m \in M$ has an available reinforcement budget of B_m prior to the disaster. The supply capacity of node $i \in S_m$ and demand of node $i \in D_m$ are given by s_{im} and d_{im}, respectively. For infrastructure $m \in M$, cn_{im} and ca_{ijm} denote the capacities of transshipment node $i \in T_m$ and arc $(i, j) \in A_m$, respectively. Reinforcing arc $(i, j) \in A_m$ costs c_{ijm} monetary units, and repairing arc $(i, j) \in A_m$ takes p_{ijm} time periods. Each disaster scenario $s \in S$ has an associated probability of P_s.

The first-stage decisions of whether arc $(i, j) \in A_m$ in infrastructure $m \in M$ is reinforced are denoted by r_{ijm}. Among the second-stage decision variables, continuous variables x_{ijtms} denote the flow on arc $(i, j) \in A_m$ and v_{itms} represent the demand satisfied in node $i \in D_m$ in infrastructure $m \in M$ at period $t \in T$ in scenario $s \in S$. There are three sets of binary second-stage decision variables. Variables y_{ijmnts} store whether input dependency of node $j \in N_n$ on node $i \in D_m$ is satisfied at period $t \in T$ in scenario $s \in S$. Similarly, α_{ijtkms} denotes if repair of arc $(i, j) \in A_m$, $m \in M$ is finished at period $t \in T$ by repair crew $k \in K_m$ in scenario $s \in S$, and β_{ijtms} represent whether arc $(i, j) \in A_m$, $m \in M$ is operational at period $t \in T$ in scenario $s \in S$.

Before we present the mathematical model for the SIIRR problem, we discuss the structure of the objective function, which we base on the one defined by Sharkey et al. (2015b). The function needs to use an appropriate effectiveness measure for service delivery in each period, as well as to ensure that establishing connectivity in the earlier time periods is encouraged by the model. To measure the effectiveness of service delivery, we use the ratio of additional service provided by the reinforcement and restoration efforts resulting from SIIRR-SP over the maximum possible increase in the service level. More specifically, the parameter fNR_{ms} represents the amount of weighted service that can be supplied by infrastructure $m \in M$ in scenario $s \in S$ right after the disaster without any reinforcement and repair efforts, whereas fDF_m denotes the total weighted service level of infrastructure $m \in M$ before the disaster. Defining w_{im} as the weight of demand node $i \in D_m$ in infrastructure $m \in M$ and v_{itms} as the amount of demand met in demand node $i \in D_m$ in infrastructure $m \in M$ at period $t \in T$, the overall effectiveness of service in infrastructure $m \in M$ in period $t \in T$ under scenario $s \in S$ is then given by:

$$\frac{\sum_{i \in D_m} w_{im} v_{itms} - fNR_{ms}}{fDF_m - fNR_{ms}}.$$

To exemplify this, consider a power network where the total electricity service provided to the users prior to the disaster (fDF_m) is 200 GWh. Suppose also that without any prior reinforcements to the network, the service level that can be provided immediately after the disaster hits (fNR_{ms}) is 125 GWh. If all the users are equally weighted and after the first period of repairs the total amount of service provided is 140 GWh, the effectiveness of restoration for this infrastructure and time period is given by $(140 - 125)/(200 - 125) = 0.20$.

For the objective function of the SIIRR-SP model, this ratio is summed up over all infrastructures and the planning horizon so as to restore the services as soon as possible. Summing up the performance of the infrastructures as percentages is also used for scaling purposes in case the units of the services of the infrastructures are different from one another:

$$\text{Maximize} \sum_{s \in S} P_s \sum_{t \in T} \sum_{m \in M} \frac{\sum_{i \in D_m} w_{im} v_{itms} - fNR_{ms}}{fDF_m - fNR_{ms}} \tag{1}$$

The first set of constraints for the SIIRR-SP model stipulate the budget for the reinforcement efforts. These constitute the only constraints of the first stage:

$$\sum_{(i,j)\in A_m} c_{ijm}\, r_{ijm} \leq B_m \qquad \forall m \in M \quad (2)$$

Flow balance equations for supply, transshipment and demand nodes are given in constraints (3), (4) and (5), respectively. The demand satisfaction of a node is limited with the demand value via constraints (6) to prevent superfluous flows:

$$\sum_{j:(i,j)\in A_m} x_{ijtms} - \sum_{j:(j,i)\in A_m} x_{jitms} \leq s_{im} \qquad \forall m \in M,\ i \in S_m,\ t \in T,\ s \in S \quad (3)$$

$$\sum_{j:(i,j)\in A_m} x_{ijtms} - \sum_{j:(j,i)\in A_m} x_{jitms} = 0 \qquad \forall m \in M,\ i \in T_m,\ t \in T,\ s \in S \quad (4)$$

$$\sum_{j:(i,j)\in A_m} x_{ijtms} - \sum_{j:(j,i)\in A_m} x_{jitms} = -v_{itms} \qquad \forall m \in M,\ i \in D_m,\ t \in T,\ s \in S \quad (5)$$

$$v_{itms} \leq d_{im} \qquad \forall m \in M,\ i \in D_m,\ t \in T,\ s \in S \quad (6)$$

Node capacity of a transshipment node is enforced by constraints (7). Arc capacities are stipulated on the flow variables for undamaged arcs and repaired arcs with constraints (8) and (9), respectively:

$$\sum_{j:(j,i)\in A_m} x_{jitms} \leq cn_{im} \qquad \forall m \in M,\ i \in T_m,\ t \in T,\ s \in S \quad (7)$$

$$x_{ijtms} \leq ca_{ijm} \qquad \forall m \in M,\ t \in T,\ s \in S,\ (i,j) \in E_{ms} \quad (8)$$

$$x_{ijtms} \leq ca_{ijm}\ \beta_{ijtms} \qquad \forall m \in M,\ t \in T,\ s \in S,\ (i,j) \in \bar{E}_{ms} \quad (9)$$

An input interdependency between nodes i and j is satisfied only if demand of node i is satisfied completely and this condition is reflected to the model by constraints (10). Once an interdependency is satisfied at period t, it needs to be satisfied until the end of the horizon via constraints (11). Flow from node j is allowed only if its input interdependency is satisfied with the help of constraints (12)-(14) and the flow amount is limited by supply, demand, and node capacity, respectively, if interdependency is satisfied:

$$d_{im}\ y_{ijmnts} \leq v_{itms} \qquad \forall m, n \in M,\ (i,j) \in F_{mn},\ t \in T,\ s \in S \quad (10)$$

$$y_{ijmn(t-1)s} \leq y_{ijmnts} \qquad \forall m, n \in M,\ (i,j) \in F_{mn},\ t \in T - \{1\},\ s \in S \quad (11)$$

$$\sum_{h:(j,h)\in A_n} x_{jhtns} \leq s_{jn}\ y_{ijmnts} \qquad \forall m, n \in M,\ j \in S_n,\ (i,j) \in F_{mn},\ t \in T,\ s \in S \quad (12)$$

$$\sum_{h:(h,j)\in A_n} x_{hjtns} \leq d_{jn}\ y_{ijmnts} \qquad \forall m, n \in M,\ j \in D_n,\ (i,j) \in F_{mn},\ t \in T,\ s \in S \quad (13)$$

$$\sum_{h:(h,j)\in A_n} x_{hjtns} \leq cn_{jn}\ y_{ijmnts} \qquad \forall m, n \in M,\ j \in T_n,\ (i,j) \in F_{mn},\ t \in T,\ s \in S \quad (14)$$

The operational availability of an arc considering all repair actions is determined with constraints (15). If arc (i,j) is reinforced, constraints (16) allow it to be available starting from time period 1. Constraints (17) make sure that only one arc is repaired by a restoration team at a time and prevent repair of any other arcs during the repair. Constraints (18) prevent the repair of an arc to be completed before the minimum possible completion time, which is the repair duration:

$$\beta_{ijtms} - \beta_{ij(t-1)ms} = \sum_{k\in K_m} \alpha_{ijtkms} \qquad \forall m \in M,\ t \in T - \{1\}, \forall s \in S,\ (i,j) \in \bar{E}_{ms} \quad (15)$$

$$\beta_{ij1ms} \leq r_{ijm} \qquad \forall m \in M,\ s \in S,\ (i,j) \in \bar{E}_{ms} \quad (16)$$

$$\sum_{l=max\{1,t-p_{ijm}+1\}}^{t} \sum_{(h,b)\in\bar{E}_{ms}} \alpha_{hblkms} \leq \alpha_{ijtkms} + \mathcal{M}(1-\alpha_{ijtkms}) \qquad \forall m \in M,\, t \in T,\, k \in K_m,\, s \in S,\, (i,j) \in \bar{E}_{ms} \quad (17)$$

$$\alpha_{ijtkms} = 0 \qquad \forall m \in M,\, k \in K_m,\, s \in S,\; (i,j) \in \bar{E}_{ms},\, t \in T : \, t \leq p_{ijm} \quad (18)$$

The model SIIRR-SP is completed with the set and nonnegativity constraints in constraints (19)-(24):

$$r_{ijm} \in \{0,1\} \qquad \forall m \in M,\, (i,j) \in A_m \quad (19)$$

$$y_{ijmnts} \in \{0,1\} \qquad \forall m,n \in M,\, (i,j) \in F_{mn},\, t \in T,\, s \in S \quad (20)$$

$$\alpha_{ijtkms} \in \{0,1\} \qquad \forall m \in M,\, t \in T,\, k \in K_m,\, s \in S,\, (i,j) \in \bar{E}_{ms} \quad (21)$$

$$\beta_{ijtms} \in \{0,1\} \qquad \forall m \in M,\, t \in T,\, s \in S,\, (i,j) \in \bar{E}_{ms} \quad (22)$$

$$x_{ijtms} \geq 0 \qquad \forall m \in M,\, (i,j) \in A_m,\, t \in T,\, s \in S \quad (23)$$

$$v_{itms} \geq 0 \qquad \forall m \in M,\, i \in D_m,\, t \in T,\, s \in S \quad (24)$$

The fNR_{ms} values in the objective function can be calculated beforehand for every scenario $s \in S$ and $T = \{1\}$ by solving (3)-(14), (20), (23) and (24) after equating the right-hand side of equation (9) to zero with an objective of maximizing the weighted demand met. Similarly, fDF_m values can be computed by solving (3)-(14), (20), (23) and (24) with an objective of maximizing the weighted demand met using a single scenario s such that $E_{ms} = A_m$, $\bar{E}_{ms} = \emptyset$ and $T = \{1\}$.

Without any uncertainty on which network segments will be disrupted and any reinforcement decisions, the SIIRR problem reduces to the IINDS problem, which has been shown to be $\mathcal{NP}$-hard by reduction from the set cover problem (Nurre & Sharkey, 2014). Hence, unless $\mathcal{P} = \mathcal{NP}$, a polynomial-time algorithm that finds the optimal solution cannot be found for the SIIRR problem. Furthermore, the existence of binary variables in the second-stage of the problem deems using exact approaches, such as the L-shaped algorithm, impractical. Our preliminary experiments with a Benders decomposition algorithm show that even finding feasible solutions in reasonable times using this approach is not possible. Consequently, we develop a heuristic procedure that aims to find high-quality solutions for the SIIRR problem within acceptable time limits.

4. Solution Methodology

As the SIIRR problem cannot be solved within reasonable computational times even for small-sized instances, a heuristic method that makes use of a genetic algorithm (GA) and partial optimization has been developed for its solution. In the literature, GAs have been observed to produce high-quality results for infrastructure restoration problems (e.g., Sato & Ichii, 1996; Dudenhoeffer et al., 2006; Xu et al., 2007; Permann, 2007; Ouyang & Wang, 2015; Sharma et al., 2019) and infrastructure reinforcement problems (e.g., Najafi et al., 2018; He & Yuan, 2019; Ghaffarpour et al., 2020; Najafi Tari et al., 2021).

In the proposed solution method, the first-stage decisions of which arcs to reinforce are determined using the GA, whereas the second-stage decisions regarding repair scheduling and network flows are addressed by a heuristic. To reduce the computational burden on the GA, the scenario set S is reduced to a representative scenario that considers the worst possible disruption to the infrastructure network. Furthermore, instead of an exact calculation of the fitness value, the GA uses the Flow and Repair Scheduling Heuristic (FRESH) with this representative scenario. This is because the SIIRR-SP model cannot be solved to optimality even for a single scenario. At the end of the GA, the reinforcement decisions with the highest fitness value are fed to FRESH to determine the second-stage decisions for all scenarios. The expected objective value is also calculated over the whole scenario set S, using these heuristically-determined second-stage decisions. The following sections discuss the details of the GA and FRESH.

4.1. A Genetic Algorithm for the SIIRR Problem

The proposed GA for the SIIRR problem is used to determine the first-stage reinforcement decision variables r_{ijm} by utilizing a representative scenario of the scenario set S. This representative scenario is generated by ranking the arcs in descending order of the number of scenarios in which the arc is damaged and assuming the first $\delta_{max}\%$ (maximum percentage of arcs that might be disrupted) of the arcs as damaged. The GA utilizes this worst-case scenario to take as many reinforcement action opportunities into account as possible.

Algorithm 1 GA(Γ, SS) $\rightarrow$ (R)

Generate an initial population of size $PopSize$
    Insert $PopSize \times p_s\%$ individuals from SS
    Call RandS(A_m, c_{ijm}, B_m, p_s, $PopSize$)
Compute fitness of each individual using FRESH
for $Iter = 1$ to $NumIter$ **do**
    for $Steps = 1$ to $PopSize/2$ **do**
        Select two parents, p_1 and p_2 using the parent selection strategy
        Apply Uniform Crossover to parents with probability p_c to produce two offspring o_1 and o_2
        Apply single-gene mutation to each offspring with probability p_m to obtain mutated offspring mo_1 and mo_2
        Compute penalized fitness values of mo_1 and mo_2 using FRESH
        Sort the population and the mutated offspring in decreasing order of fitness value
        Form the population for the next generation by selecting the first $PopSize$ many individuals
        $Steps = Steps + 1$
    end for
    $Iter = Iter + 1$
end for

The pseudocode of the GA is given in Algorithm 1. Each individual in the GA corresponds to an R vector, where each gene represents the binary r_{ijm} values. The initial population composes of $p_s\%$ promising seed solutions from the seed solution set SS for rapid convergence, with the remaining members being random feasible solutions. Generation of SS and random feasible solutions $RandS$ is explained in detail in Appendix A of the Online Supplement. GA takes all parameters Γ of the SIIRR problem with the representative scenario and the seed solution set SS as inputs, and outputs the best individual R. In every iteration, two parents are either selected randomly or with probabilities proportional to the fitness value of the parent. Uniform crossover is applied to the parents with probability p_c and offspring are modified using single-gene mutation with probability p_m. Fitness values of the mutated offspring are calculated using FRESH with the representative scenario. Infeasible solutions in terms of budget are allowed throughout the iterations to increase exploration of the solution space. However, infeasible solutions are penalized by the percentage they exceed the budget using a penalty function as follows:

$$F' = max\left\{0, F - c_p \sum_{m \in M} \frac{\sum_{(i,j)\in A_m} r_{ijm}c_{ijm} - B_m}{B_m}\right\},$$

where F' is the penalized fitness value, c_p is the penalization coefficient and F is the fitness value calculated by FRESH. The value of c_p is increased towards the later iterations of the algorithm to increase the possibility of a feasible solution at the end. In every iteration, the next generation is formed by replacing two worst individuals of the current generation with the offspring if offspring have higher fitness values. In the end, the feasible r_{ijm} values that result in the best estimated fitness value is returned by the GA.

4.2. The Flow and Repair Scheduling Heuristic (FRESH)

FRESH is used both for the fitness function calculation during the GA and to determine the second-stage decisions of the two-stage SIIRR problem at the end of it. In the first case, the heuristic is called with the representative scenario. In the second case, it takes the first-stage decisions of the incumbent GA solution as fixed, and is applied to all scenarios in S to determine the second-stage decision variables and the objective value.

Algorithm 2 FRESH(Γ_s,R) $\rightarrow (X, Y, V, A, B)$

$\beta_{ijtms} = r_{ijm} \quad \forall t \in T,\ (i,j) \in A_m,\ m \in M$
$w_i = \mathcal{M} \quad \forall j \in N_n : (i,j) \in F_{mn}, i \in D_m,\ m \in M$
$RT_m = |T| \quad \forall m \in M$
t^{crew}_{km}, t^{infr}_m, α_{ijtkms}, $x_{ijtms}, v_{itms}, y_{ijmnts}, tabu_{ijm} = 0 \quad \forall m, n \in M,\ (i,j) \in A_m,\ t \in T,\ k \in K_m$
while $RT_m \geq \min_{(i,j)\in \bar{E}_{ms}} \{p_{ijm}\}$ for any $m \in M$ **and** $\sum_{i\in D_m} \sum_{t\in T} \sum_{m\in M} v_{itms} \neq \sum_{i\in D_m} \sum_{m\in M} d_{im}$ **do**
    Call FRSCH($\Gamma_s, Tabu, T^{crew}, T^{infr}, A, B, X, V, Y$)
    if there is no new repair **then**
        break while
    end if
    Call RSIH($\bar{E}_{ms}, T^{crew}, T^{infr}, A, B$)
    Call FO(A, B)
    $RT_m = \sum_{k\in K_m} (T - t^{crew}_{km}) \quad \forall m \in M$
end while
Fill the idle time periods at the end of the schedule with repairs according to the SPTF rule
Call FO(A, B)

The pseudocode of FRESH for a given scenario $s \in S$ is provided in Algorithm 2. FRESH takes all parameters of SIIRR problem for scenario s (Γ_s) and R found by GA, and outputs the values of the decision variables as vectors X, Y, V, A, and B. FRESH first initializes all reinforced arcs to be available for all $t \in T$. All remaining arcs are initialized as unavailable. Weights of dependent nodes are increased to a large number to reflect their importance and the remaining time in an infrastructure, RT_m, is initialized as $|T|$.

During FRESH, the repair schedule is constructed for every repair team $k \in K_m$ of every infrastructure $m \in M$. The time period in which the latest repair is completed by a repair team k in infrastructure m is stored as the crew time point t^{crew}_{km}. Similarly, the time period in which the latest repair is completed over all repair teams of infrastructure m is stored as the infrastructure time point t^{infr}_m. These time points are updated as new repairs are scheduled and are used to keep track of the crew schedules. At the beginning of FRESH, these are initialized as 0, as are decision vectors A, X, V, and Y. The $tabu_{ijm}$ variable keeps track of the time point until which arcs under repair cannot be used. This is also initialized as zero for all arcs.

As long as the remaining time in any of the infrastructures is sufficient to schedule a repair and total demand satisfaction in all infrastructures is not equal to the total demand, the main loop of FRESH finds a repair schedule (values of α_{ijtkms} and β_{ijtms} as vectors A and B) considering the interdependencies between the infrastructures using the Flow and Repair Schedule Construction Heuristic (FRSCH). If there are no new repairs suggested by FRSCH, the main loop is terminated. Otherwise, new repairs are scheduled. The resulting schedule may involve repair teams being idle for a significant amount of time, as FRSCH schedules a repair only when a flow through a damaged arc is desired. In that case, the repair schedule is modified using the Repair Schedule Improvement

Heuristic (RSIH), where the gaps in the schedule are removed by preserving the order of repairs. The new repair schedule is then fed into a flow optimization model (FO) to find the optimal flows on arcs, demand met in each node, and interdependency satisfaction (values of the x_{ijtms}, v_{itms} and y_{ijmnts} decision variables, respectively as vectors X, V and Y), given the repair schedule (A and B). The FO is a special case of the SIIRR-SP model for a single scenario $s \in S$ and fixed values of the r_{ijm}, α_{ijtkms}, and β_{ijtms} variables. The only modification is the following set of constraints:

$$d_{itms} - v_{itms} \geq 1 - y_{ijmnts} \quad \forall m, n \in M,\ (i, j) \in F_{mn},\ t \in T,\ s \in S,$$

which ensure that $y_{ijmnts} = 1$ if and only if interdependency is satisfied along with constraints (10). After the main loop is terminated, the resulting schedule may have idle periods at the end of the schedule. As infrastructures need to be completely repaired after a disaster for recovery, we allow repairs to continue even if there is no flow on the repaired arcs in these periods. For this purpose, the remaining idle time periods are filled with arc repairs according to the shortest processing time first (SPTF) rule. In the end, the final repair schedule is fed to the flow optimization model one last time to obtain the final X, Y and V vectors.

In what follows, we describe the subroutines of FRESH, namely the FRSCH and RSIH algorithms.

4.2.1. The Flow and Repair Schedule Construction Heuristic (FRSCH)

FRSCH, whose pseudocode is available in Algorithm 3, gradually fills the repair schedule considering the interdependencies between the infrastructures, arc, and node capacities, and repair times of the arcs. The algorithm starts with the first infrastructure ($m = 1$) and finds an augmenting path on it. When an augmenting path is found, a repair is scheduled for damaged arcs on this path. After path repair is completed, flow is sent through this path from the time path repair is completed until the end of the horizon. In the next iteration, m is incremented by one and an augmenting path is searched for this infrastructure. When all infrastructures finish their first iteration, the algorithm turns back to the first infrastructure. These steps are repeated until demand is completely satisfied or the schedule is full with repairs in all infrastructures.

As a node can require multiple dependencies to be satisfied to be operational, FRSCH first calculates a $ycheck_{imt}$ variable for all supply nodes at the infrastructure time point to store how many dependencies of a supply node are satisfied. Then, networks of infrastructures are extended by adding a supply and demand supernode via the Network Extension (NE) subroutine, whose pseudocode is given in Algorithm 4. In the extended network, all supply nodes are linked to a supply supernode $\mathcal{S}$. If all dependencies of a supply node are satisfied, the capacities of the new arcs are set as the supply amounts. Otherwise, these are set to zero. The repair times of the new arcs are set to zero except where interdependency is not satisfied, in which case they are set to infinity. Similarly, all demand nodes are linked to a demand supernode $\mathcal{D}$. The capacities of the new arcs are set as the demand amounts and the repair times of the new arcs are set to zero. For all the arcs that are repaired by the infrastructure time point, repair times are set to zero as well.

After the network extension, FRSCH goes through the main loop for each infrastructure. If there are idle time periods at the end of the repair schedule for any infrastructure, the availability of a dependent supply node is checked first. If all dependencies of a supply node are satisfied, arc capacity from the supply supernode to the

Algorithm 3 FRSCH$(\Gamma_s, Tabu, T^{crew}, T^{infr}, A, B, X, Y, V) \rightarrow (Tabu, T^{crew}, T^{infr}, A, B, X, Y, V)$

$ycheck_{imt_m^{infr}} = \sum_{n \in M} \sum_{j \in N_n} y_{jinmt_m^{infr}s} \quad \forall m \in M,\ i \in S_m$
Call NE$(\Gamma_s, T^{infr}, B, X, Ycheck)$
$m = 1$
while $\sum_{m \in M} t_m^{infr} \neq |T||M|$ **do**
  $ca_{\mathcal{S}im} = s_{im}$ and $p_{\mathcal{S}im} = 0 \quad \forall i \in S_m : ycheck_{imt_m^{infr}} = \sum_{j \in N_n} \sum_{n \in M} I_{(i,j) \in F_{mn}}$
  $rc_{ijm} = \min_{k \in K_m} R_{ijt_{km}^{crew}m} \quad \forall (i,j) \in A_m$, where R_{ijtm} is residual capacity of (i,j) at $t \in T$
  $rc_{ivm} = 0 \quad \forall (i,v) \in A_m : \sum_{j \in N_n} \sum_{n \in M} I_{(i,j) \in F_{mn}} > 0$ and $ycheck_{imt_m^{infr}} < \sum_{j \in N_n} \sum_{n \in M} I_{(i,j) \in F_{mn}}$
  Call AP$(\Gamma_{sm}, RC, B, T^{crew}, \min_{k \in K_m}\{t_{km}^{crew}\}, Tabu, X, V)$
  if $\delta_{P^*} > 0$ **then**
    Call SCH$(\Gamma_{sm}, T^{crew}, B, X, P^*, p_{P^*}, \delta_{P^*})$
    if $Fl = 1$ **then**
      Call FLOW$(\Gamma_{sm}, X, V, Y, Ycheck, T^{crew}, t_{path}, P^*, p_{P^*}, \delta_{P^*})$
    end if
  end if
  $t_m^{infr} = \max_{k \in K_m} t_{km}^{crew}$
  if $t_m^{infr} = |T| \quad \forall m \in M$ **then**
    break while
  end if
  if $\delta_{P^*} = 0 \quad \forall m \in M$ in their last iteration **then**
    if $\sum_{i \in D_m} v_{it_m^{infr}ms} = \sum_{i \in D_m} d_{im} \quad \forall m \in M$ **then**
      break while
    else
      if $\exists t : t > t_{m'}^{infr}, ycheck_{im't} = \sum_{j \in N_n} \sum_{n \in M} I_{(i,j) \in F(m',n)}$ for an $m' \in M$ **then**
        $t_{m'}^{infr} = t,\ t_{km'}^{crew} = t \quad \forall k \in K'_m$, and $m = m'$
      else
        $t_m^{infr} = |T| \quad \forall m \in M$ and $t_{km}^{crew} = |T| \quad \forall k \in K_m,\ \forall m \in M$
      end if
    end if
  else
    $m = m\ (mod\ |M|) + 1$
  end if
end while

Algorithm 4 NE$(\Gamma_s, T^{infr}, B, X, Ycheck) \rightarrow$ (CA, P,X)

for all $m \in M$ **do**
  for all $i \in S_m$ **do**
    $x_{\mathcal{S}itms} = \sum_{j:(i,j) \in A_m} x_{ijtms}$
    if $ycheck_{imt_m^{infr}} = \sum_{j \in N_n} \sum_{n \in M} I_{(i,j) \in F_{mn}}$ **then**
      $ca_{\mathcal{S}im} = s_{im}$ and $p_{\mathcal{S}im} = 0$
    else
      $ca_{\mathcal{S}im} = 0$ and $p_{\mathcal{S}im} = \infty$
    end if
  end for
  for all $i \in D_m$ **do**
    $x_{i\mathcal{D}tms} = \sum_{j:(j,i) \in A_m} x_{jitms}$
    $ca_{i\mathcal{D}m} = d_{im}$ and $p_{i\mathcal{D}m} = 0$
  end for
  for all $(i,j) \in A_m : \beta_{ijt_m^{infr}m} = 1$ **do**
    $p_{ijm} = 0$
  end for
end for

supply node is restored to the supply level and repair time is set to zero. A residual network is then extracted from the extended network. The residual capacities rc are calculated at $t^{crew}_{km} \quad \forall k \in K_m$ and the minimum of these residual capacities is taken to make sure the arc capacities are not exceeded until the end of the horizon. In addition, residual capacities of the arcs emanating from interdependent nodes whose dependencies are not yet satisfied are set to zero to prevent the algorithm from selecting those arcs.

FRSCH tries to find an augmenting path P^* in the resulting residual network that has the maximum flow amount, δ_{P^*}, and minimum path repair time, p_{P^*}, to satisfy the demand of the node by the largest amount as soon as possible until the end of the horizon using the Augmenting Path (AP) subroutine (see Algorithm 5). AP initializes δ_{P^*} and p_{P^*} to zero and recalculates the remaining time for infrastructure m. For all possible flow amounts r_P, arcs with less residual capacity than r_P are deleted from the network by setting their residual capacity to zero and weight to infinity. The weights of arcs with residual capacity of at least r_P are set to their repair times if they are not already repaired, and to zero otherwise. These weights are then fed to the Dijkstra algorithm as arc lengths to find the shortest paths to all demand points. Path repair times, p_{P_i} are calculated for all shortest paths and valid paths are selected. A path P_i is in $Invalid$ set if one arc on the path is under construction or path repair time is larger than the remaining time. For the valid paths, path capacity δ_{P_i} is calculated as the minimum of residual capacities of the arcs, remaining transshipment node capacities (if any), and amount of unmet demand at the demand node. The path maximizing $w_i\delta_{P_i}(T+1-p_{P_i}-t)$ over all r_P is selected as P^* where w_i is the weight of demand node. This formula is in line with the objective function of the SIIRR problem, as it also tries to accumulate satisfied demand values over as many time periods as possible to reach the demand nodes the earliest. The multiplication $r_P(T+1-p_P-t)$ denotes the estimation of addition to the objective function from path P assuming one team works on the repairs.

Algorithm 5 AP$(\Gamma_{sm}, RC, B, T^{crew}, t, Tabu, X, V) \rightarrow (P^*, p_{P^*}, \delta_{P^*})$

$p_{P^*} = 0$ and $\delta_{P^*} = 0$
$RT_m = \sum_{k\in K_m}(T - t^{crew}_{km})$
for $r_P = 1, 2, 3, ..., 2|A_m|$ **do**
  for all $(i,j) \in A_m$ **do**
    if $rc_{ijm} < r_P$ **then**
      $rc_{ijm} = 0$ and $weight_{ijm} = \infty$
    else
      $weight_{ijm} = p_{ijm}(1-\beta_{ijtms})$
    end if
  end for
  Call Dijkstra Algorithm with $weight_{ijm}$ to find shortest paths to all $i \in D_m$
  $p_{P_i} = \sum_{(u,v)\in P_i} weight_{uvm} \quad \forall i \in D_m$ where P_i is the SPT for $i \in D_m$
  $Valid = InValid = \{\ \}$
  for all $i \in D_m$ **do**
    if $\exists\ (u,v) \in P_i : tabu_{uvm} > t$ **or** $p_{P_i} > RT_m$ **then**
      $InValid = InValid \cup \{P_i\}$
    else
      $Valid = Valid \cup \{P_i\}$
      $\delta_{P_i} = \min\left\{\min_{(u,v)\in P_i}\{rc_{uvm}\},\ \min_{u\in T_m, u\in P_i, k\in K_m}\left\{cn_{um} - \sum_{v\in N_m} x_{uvt^{crew}_{km}ms}\right\},\ d_{im} - v_{itms}\right\}$
    end if
  end for
  $P_r = argmax_{P_i\in Valid, i\in D_m}\{w_i\delta_{P_i}(T+1-p_{P_i}-t)\}$
  $r = argmax_{i\in D_m:\ P_i\in Valid}\{w_i\delta_{P_i}(T+1-p_{P_i}-t)\}$
  if $w_r\delta_{P_r}(T+1-p_{P_r}-t) > w_i\delta_{P^*}(T+1-p_{P^*}-t)$ **then**
    $P^* = P_r$, $\delta_{P^*} = \delta_{P_r}$, and $p_{P^*} = p_{P_r}$
  end if
end for

If the P^* has nonzero path capacity, repairs for arcs needing repair on P^* are scheduled via the Scheduling (SCH) subroutine in Algorithm 6. SCH first initializes indicator variable Fl to 1. Then, starting from the demand node St at the end of the path, SCH finds the preceding node Pt of St, and the team that has the minimum crew time point k'. It then checks if the arc (Pt, St) needs repair. A damaged and not repaired arc (Pt, St) with zero flow on it and less than δ_{P^*} flow in the opposite direction can be repaired by crew k' if there is enough time for repair. If so, the crew time point is increased by the repair time of the arc, and repair and arc availability variables are updated. The *tabu* time of the arcs (Pt, St) and (St, Pt) are set to the crew time point to prevent their usage before the repair is completed. Repair time of arc (Pt, St) is set to 0 as it is already repaired, and path repair completion time t_{path} is set to the last repair completion. These steps are followed for every arc on P^* until the starting node of the path or until there is not enough time to schedule a repair.

Algorithm 6 SCH$(\Gamma_{sm}, T^{crew}, B, X, P^*, p_{P^*}, \delta_{P^*}) \rightarrow (A, Tabu, P, T^{crew}, t_{path}, Fl)$

$Fl = 1$
$St = i \in D_m$ at the end of P^*
while $St \neq \mathcal{S}$ **do**
    $Pt := j|\ (j, St) \in P^*$
    $k' = argmin_{k \in K_m} t^{crew}_{km}$
    if $(Pt, St) \in \bar{E}_{ms}$ and $\beta_{PtStt^{crew}_{k'm}ms} = 0$ and $x_{PtStt^{crew}_{k'm}ms} = 0$ and $\delta_{P^*} > x_{StPtt^{crew}_{k'm}ms}$ **then**
        if $t^{crew}_{k'm} + p_{PtStm} \leq T$ **then**
            $t^{crew}_{k'm} = t^{crew}_{k'm} + p_{PtStm}$ and $\alpha_{Pt,St,t^{crew}_{k'm},k'm,s} = 1$
            $\beta_{Pt,St,t,m,s} = 1 \quad for\ t^{crew}_{k'm} \leq t \leq T$
            $tabu_{Pt,St,m} = t^{crew}_{k'm}$ and $tabu_{St,Pt,m} = t^{crew}_{k'm}$
            $p_{Pt,St,m} = 0$ and $t_{path} = t^{crew}_{k'm}$
            $St = Pt$
        else
            $Fl = 0$ and terminate
        end if
    else
        $St = Pt$
    end if
end while

If SCH returns a Fl value of 1, flow can be set through the path P^* using the FLOW subroutine in Algorithm 7. Starting from the arc between St and Pt, the FLOW subroutine arranges the flow on the arcs on the path from time period t_{path} until T. If there is no flow in any direction between St and Pt, flow in the forward direction (Pt, St) is increased by δ_{P^*}. If there is flow only in the forward direction, the flow is increased by δ_{P^*}. If there is at least δ_{P^*} amount of flow only in the reverse direction, then this flow is decreased by δ_{P^*}. Lastly, if there is less than δ_{P^*} amount of flow only in the reverse direction, flow changes direction. The current flow amount in the reverse direction is stored in variable F and the flow of the reverse direction is set to zero. Flow on the forward direction (Pt, St) is increased by $\delta_{P^*} - F$. These steps are repeated for all arcs on the path P^* and the demand satisfaction value of the demand node i at the end of the path is increased by δ_{P^*} until the end of the horizon. If demand node i satisfies an interdependency and its demand is satisfied completely, the interdependency control variable is set to 1 and *ycheck* is increased by 1 from t_{path} until T. The weight of the interdependent supply node is increased to infinity to give this node the highest importance in the other infrastructures and to activate the dependent point as soon as possible.

After the flow arrangement, FRSCH recalculates the infrastructure time point and checks if all time points hit $|T|$. If so, the main loop of FRSCH is terminated and the repair schedule is returned to FRESH. Otherwise,

Algorithm 7 FLOW$(\Gamma_{sm}, X, V, Y, Ycheck, T^{crew}, t_{path}, P^*, p_{P^*}, \delta_{P^*}) \rightarrow (X, V, Y, Ycheck, W)$

$St = i \in D_m$ at the end of P^*
while $St \neq S$ **do**
    $Pt = \{j : (j, St) \in P^*\}$
    if $x_{Pt,St,t_{path},m,s} = 0$ and $x_{St,Pt,t_{path},m,s} = 0$ **then**
        $x_{Pt,St,t,m,s} = x_{Pt,St,t,m,s} + \delta_{P^*}$ for $t_{path} \leq t \leq T$
    else if $x_{Pt,St,t_{path},ms} > 0$ and $x_{St,Pt,t_{path},m,s} = 0$ **then**
        $x_{Pt,St,t,m,s} = x_{Pt,St,t,m,s} + \delta_{P^*}$ for $t_{path} \leq t \leq T$
    else if $x_{Pt,St,t_{path},ms} = 0$ and $x_{St,Pt,t_{path},m,s} > 0$ and $x_{St,Pt,t_{path},m,s} \geq \delta_{P^*}$ **then**
        $x_{St,Pt,t,m,s} = x_{St,Pt,t,m,s} - \delta_{P^*}$ for $t_{path} \leq t \leq T$
    else if $x_{Pt,St,t_{path},m,s} = 0$ and $x_{St,Pt,t_{path},m,s} > 0$ and $x_{St,Pt,t_{path},m,s} < \delta_{P^*}$ **then**
        $F = x_{St,Pt,t_{path},ms}$
        $x_{PtSttms} = x_{PtSttms} + \delta_{P^*} - F$ for $t_{path} \leq t \leq T$
        $x_{StPttms} = x_{StPttms} - F$ for $t_{path} \leq t \leq T$
    end if
    $St = Pt$
end while
Set $v_{itms} = v_{itms} + \delta_{P^*}$ for $t_{path} \leq t \leq T$ where i is the demand node at the end of P^*
if $\exists\, j \in N_n | (i,j) \in F_{mn}$ and $v_{it_{path}ms} = d_{im}$ **then**
    $y_{ijmnts} = 1$ for $t_{path} \leq t \leq T$
    $ycheck_{jnt} = ycheck_{jnt} + 1$ for $t_{path} \leq t \leq T$
    $w_j = \infty$
end if

FRSCH checks if any augmenting path with nonzero path capacity can be found in any of the infrastructures. If this is the case, FRSCH continues its iterations with the next infrastructure. Otherwise, if the total demand of all infrastructures is met, the main loop of FRSCH is terminated and the repair schedule is returned to FRESH. If there is at least one infrastructure with incomplete demand satisfaction, an interdependent supply node whose interdependency is satisfied in later time periods than t_m^{infr} is sought. Since repairs are independent of each other, t_m^{infr} are also different for every infrastructure. Thus, the interdependency of a supply node may be met in a later time period than t_m^{infr}. The earliest time point in which a new supply point becomes available is calculated and the parameters t_m^{infr} and t_{km}^{crew} $\forall k \in K_m$ are set to this time point in the infrastructure in which a new supply point becomes available. New augmenting paths are suggested iteratively for all infrastructures and more demand is satisfied in this manner, if possible. If there is no new active supply point in any of the infrastructures, t_m^{infr} is set to the last time point in every infrastructure, and FRSCH terminates in the next iteration.

4.2.2. The Repair Schedule Improvement Heuristic (RSIH)

FRSCH schedules a repair only when flow through a damaged arc is needed. As a result, there are gaps (idle time periods) in the repair schedule. As repairs are not normally dependent on flows, FRESH passes the repair schedule generated by FRSCH to RSIH, which takes the repair schedule generated by FRSCH and removes the gaps while preserving the order of the repairs via the RSIH in Algorithm 8. RSIH checks the schedule of every team in every infrastructure by ranking the repaired arcs in increasing order of their repair time period. Starting from the first arc, it checks if there are idle time periods ahead of the repair time period. If so, repair of the arc is shifted ahead to the earliest idle time period. Arc availability variable and the crew time point are also updated. Once the schedules of all teams in an infrastructure are updated, the infrastructure time point is recalculated as the maximum of the crew time points. FRESH feeds the resulting repair schedule from RSIH to the flow optimization model. The results are then fed to FRSCH again to schedule new repairs starting from updated t_{km}^{crew} and t_m^{infr} values.

Algorithm 8 RSIH($\bar{E}_{ms}, T^{crew}, T^{infr}, A, B$) $\rightarrow$ (T^{crew}, T^{infr}, A, B)

for all $m \in M$ **do**
  for all $k \in K_m$ **do**
    Rank arcs $(i,j) \in \bar{E}_{ms}$ in increasing order of $t' \; | \alpha_{ijt'kms} = 1$ such that arc (i,j) has rank $Rank_{ij}$
    Start with $(i,j) : Rank_{ij} = 1$
    while $Rank_{ij} \neq \max_{(i,j)\in\bar{E}_{ms}} Rank_{ij}$ **do**
      if $\exists t^* \; | t^* = min_{t<t'}\{\sum_{(i,j)\in\bar{E}_{ms}} \sum_{(i,j)\in\bar{E}_{ms}} \alpha_{ijtkms} = 0\}$ **then**
        $\alpha_{ijt^*kms} = 1$, $\alpha_{ijtkms} = 0$
        $\beta_{ijt''ms} = 1 \; \forall t^* \leq t'' < t'$
        $t^{crew}_{km} = t^*$
      end if
      $(i,j) = (u,v) : Rank_{uv} = Rank_{ij} + 1$
    end while
  end for
  $t^{infr}_m = \max_{k\in K_m} t^{crew}_{km}$
end for

An implementation of FRESH on an example instance can be found in Appendix B of the Online Supplement.

5. Computational Experiments

Our computational experiments serve four main purposes. First, we would like to assess the performance of the proposed heuristic algorithm in terms of how it compares to the optimal solutions. Second, we attempt to draw a number of managerial insights regarding the value of reinforcement, budget pooling, and considering interdependencies. Our third aim is to evaluate the sensitivity of the results to problem parameters, namely the budget and number of teams. Our last goal is to justify the use of a stochastic modeling approach by estimating the value of the stochastic solution and expected value of perfect information.

5.1. Test Instances

Our test instances are generated using the network of Sioux Falls City in South Dakota (LeBlanc et al., 1975), originally used for traffic equilibrium problems. The network, which is presented in Appendix C of the Online Supplement, includes 24 nodes and 76 arcs. We consider three infrastructures, namely electricity, wastewater, and water. All three infrastructures are assumed to use the same network. Two repair teams work in each infrastructure and the reinforcement budget of an infrastructure is set to 5% of the total reinforcement cost of all of its arcs. One time period represents 10 hours, and the planning horizon is 30 time periods (approximately one week), as in Sharkey et al. (2015b). We assume unit weights on the demand nodes.

The sets of supply, transshipment, and demand nodes on the network are listed in Table C.1 in Appendix C of the Online Supplement along with the probability distributions of the demand and supply amounts, arc and node capacities, repair times, and reinforcement costs. The generated demand values are modified to ensure that the total demand can be completely met before the disaster. In other words, fDF_m values are equal to the total weighted demand in all infrastructures.

There are a total of 15 interdependencies between pairs of infrastructures in the network. In the test instances, an interdependency occurs if a node is a demand node in one infrastructure and a supply node in another infrastructure. Thus, a node is not dependent on another specific node, but it is rather dependent on itself over different infrastructures. Such interdependencies are widely applicable in practice, as in the case of a water pumping station being dependent on the power from the electricity network into the same node. Interdependencies in the instances are given in Table C.2 in Appendix C of the Online Supplement.

The scenarios of the SIIRR problem instances are generated using 10%, 30%, 50%, 70%, and 90% (δ_{max}) damage rate on the arcs of the infrastructures during the disaster. In one scenario, each infrastructure is damaged with the same damage rate and all arcs are equally likely to be damaged. For example, to generate a scenario with a 10% damage rate, randomly selected 10% of the arcs of each infrastructure are damaged. The master set of scenarios MS consists of 200 scenarios where 40 scenarios are generated for each damage rate. For each instance in the experiments, 10 scenarios are randomly selected from each damage rate in MS, culminating in a total of 50 equally likely scenarios. A total of 10 instances are generated in this manner for our experiments.

5.2. Algorithm Settings

A number of parameters for the GA are fixed based on preliminary experiments. The mutation probability of each gene, p_m, is set to 1%, in order to increase the exploration power of the algorithm. In the preliminary runs, it is observed that the algorithm converges easily in 100 iterations. Thus, the number of iterations in GA is set to 100. The penalization coefficient, c_p, is increased in two steps towards the end of the iterations. Its value is set to 3 for the first 33 iterations, 10 for the iterations between 34 and 66, and 20 for the iterations between 67 and 100. Such an approach enables better exploration of the solution space in the initial iterations and enforces feasible solutions towards the end.

The population size, seeding percentage p_s%, crossover probability p_c%, seeding strategy, and parent selection strategy are determined as design parameters of an experimental design setting with two levels attempted for each. According to these preliminary experiments, whose details and results are provided in Appendix D of the Online Supplement, population size is set to 50 individuals, p_s is equal to 40%, and p_c is taken as 1. Parent selection strategy is an elitist parent selection, i.e., parents have a selection probability proportional to their fitness values. Seeding strategy is based on the expected objective value of the seed solutions. In other words, seeds are listed in decreasing order of expected objective value and selected from the top of the list.

5.3. Computational Results

In this section, we report the computational results to analyze the performance of the heuristic algorithm, draw managerial insights, perform sensitivity analysis, and assess the value of capturing stochasticity. During the experiments, the SIIRR-SP is solved using CPLEX version 12.8.0, which is implemented in IBM ILOG Optimization Studio. The heuristic algorithm is coded in Java programming language via NetBeans IDE 8.2. The test environment is an Intel(R) Core (TM) i7-477OS CPU @3.10GHz, 16GB RAM Windows 10 PC.

5.3.1. Results on Heuristic Performance

Our experiments in analyzing heuristic performance mainly serve three objectives: First, we would like to test the quality of the heuristic solutions by comparing how the proposed heuristic approach performs with respect to the solutions and bounds found by CPLEX. The second aim is to understand the effect of using a representative scenario within the GA in determining the first-stage decisions. Lastly, as the heuristic estimates the fitness of

Table 1: Summary of CPLEX and heuristic results (heuristic results are averages of five GA runs and CPU times are in seconds)

	CPLEX			**Heuristic**				
					% UB Gap			CPU
Instance	z_B	z_{BI}	% Gap	z_H	Avg	Min	Max	Time
1	79.00	73.61	7.33	72.50	8.24	6.81	10.15	2,599
2	80.74	62.75	28.68	73.93	8.44	7.00	9.48	2,265
3	77.66	-1551.21	105.01	71.38	8.08	7.00	8.53	2,692
4	79.46	57.31	38.64	73.34	7.70	7.26	8.41	3,068
5	76.13	-10097.05	100.75	69.35	8.90	8.50	9.56	2,444
6	78.31	71.30	9.84	71.41	8.81	7.87	9.40	3,365
7	80.22	-1847.50	104.34	74.01	7.75	6.63	8.56	2,800
8	78.90	59.80	31.92	71.20	9.76	9.14	10.97	2,436
9	77.78	-588.90	113.21	72.00	7.43	6.97	8.06	2,573
10	79.67	47.58	67.43	70.94	10.95	9.21	13.26	2,878
Average	78.79	-1371.23	60.72	72.01	8.61	7.64	9.64	2,712

each solution using the FRESH heuristic, as opposed to solving each scenario to optimality, the loss in solution quality due to this procedure is also assessed.

We use a time limit of three days for CPLEX on each of our 10 instances. However, CPLEX is unable to find the optimal solution in any of these instances within the time limit. Consequently, we use the best upper bound values reported by CPLEX (denoted as z_B) as the benchmark to measure the performance of the heuristic. We also report the objective value corresponding to the best feasible solution found by CPLEX (denoted as z_{BI}).

The differences between the z_B and z_{BI} values in Table 1 indicate that in most of the instances, CPLEX provides solutions with large optimality gaps. The percent gap is 7.33% in the best case and 60.72% on average. The conclusion from our results with CPLEX is that even for very long time limits, the performance of CPLEX varies substantially across instances, and the solution quality is quite poor, as also evidenced by the negative z_{BI} values for four instances. Negative values of z_{BI} indicate that within the given time limit, CPLEX is unable to find an integer solution such that demand of an infrastructure is met higher than the minimum level of fNR_{ms}.

To quantify the solution quality of the heuristic solutions, we calculate the *percent upper bound gap* (% UB Gap) of each heuristic solution, with respect to the best upper bound of the CPLEX run for the same instance. The % UB Gap is calculated from:

$$\% \ UB \ Gap = \frac{z_B - z_H}{z_B} \times 100\%.$$

As z_B constitutes an upper bound on the optimal objective value, the % UB gaps are also upper bounds on the percentage gaps of the heuristic solutions from the actual optimal values.

Within the proposed heuristic, we run the GA five times for each instance. The resulting first-stage decisions are then fed to FRESH for each scenario $s \in S$ to determine the second-stage decisions and the expected objective value. The average, minimum, and maximum % UB gap of each instance are reported in Table 1, as well as the average of the expected objective function values z_H, and average CPU times. The UB gap of the proposed heuristic approach is observed to be 8.61% on average. The minimum % UB gap is 7.64% and the maximum is 9.64% on average. The close distribution of the % UB gap values points to robust results. Furthermore, the heuristic is able to produce these results in approximately 1% of the time spent by CPLEX. The main conclusion from these experiments is that the proposed heuristic finds significantly better solutions than CPLEX does for almost all of the instances in an average time of approximately 45 minutes, as opposed to three days.

The proposed heuristic uses a representative scenario to obtain the first-stage solution using the GA, and also

Table 2: Summary of computational results regarding managerial insights

Instance	z_H	z_{rep}	% Value of Reinforcement	z_{pool}	% Benefit of Budget Pooling	z_{no_int}	% Price of Ignoring Interdependencies
1	72.50	60.71	19.41	72.15	-0.48	70.12	3.28
2	73.93	63.45	16.52	73.07	-1.16	70.84	4.18
3	71.38	59.10	20.79	71.35	-0.04	68.32	4.29
4	73.34	61.98	18.32	74.16	1.12	71.03	3.14
5	69.35	59.01	17.52	69.67	0.46	68.20	1.65
6	71.41	60.67	17.71	71.52	0.15	69.29	2.98
7	74.01	63.11	17.27	74.74	1.00	72.08	2.60
8	71.20	59.94	18.78	71.09	-0.16	69.69	2.11
9	72.00	59.69	20.63	72.09	0.13	69.52	3.44
10	70.94	62.30	13.87	71.82	1.24	70.35	0.84
Average	72.01	61.00	18.08	72.17	0.23	69.94	2.85

uses FRESH to solve the second-stage problem heuristically. By using two benchmark approaches, we estimate the loss in the objective function due to these simplifications. The details of and the results of these approaches are provided in Appendix E of the Online Supplement. As indicated by these results, the level of quality loss is acceptable considering the substantial savings in the solution times.

5.3.2. Managerial Insights

The proposed heuristic, which provides high-quality results in short computational times, allows us to draw a number of managerial insights from our instances. Three such insights are drawn in this section, where we consider the value of reinforcement and its effect on restoration, the benefit of pooling the budget across all infrastructures, and the price of ignoring the inherent interdependencies in the network. The results of these experiments are summarized in Table 2.

To measure the effect of reinforcement efforts on the heuristic results, we run our instances using FRESH without any reinforcement effort ($r_{ijm} = 0 \; \forall i,\; j,\; m$) hence with only repair, as a benchmark. The objective values, represented as z_{rep}, are given in Table 2 along with the value of reinforcement, which is computed as the percent difference between z_{rep} and z_H. It can be inferred from these results that by including the reinforcement decisions in the pre-disaster stage, the average objective value increases from 61.00 to 72.01, an 18.08% improvement. The improvement ranges from 13.87% to 20.79% over the 10 instances, underlining the importance of reinforcement.

In general, the reinforcement activities are funded independently for each infrastructure by their decision-makers. In the next set of our experiments, we aim to measure the benefit of employing a pooled budget (e.g., funded by a local government or a municipality) for all the infrastructures. Towards this aim, a pooled reinforcement budget B is defined and used instead of B_m in the penalized fitness value calculation, F', for the GA. For the new benchmark instances, B is set to 5% of the total cost of reinforcement of all arcs in all infrastructures. The new instances are run five times and the average expected objective values z_{pool} are compared with that of the baseline results in Table 2. If one were to compare the optimal solutions of these instances, one would observe an improvement (or at least the same performance) in all of the instances. However, a performance decline is observed in four of the instances. This could be attributed to the optimality gap of the heuristic approach. On average, the expected objective value increases by 0.23%, which is a considerably small improvement. This implies that even in a decentralized environment, the coordinated scheduling by the proposed approach virtually

Table 3: Budget distribution in independent and pooled budget cases

	Independent	**Pooled**
Electricity	7.9	11.4
Wastewater	8.5	4.2
Water	7.9	8.3

eliminates the need for a pooled budget, leading to an easier implementation.

On the other hand, if the reinforcement activities are planned with a pooled budget, there is a chance to re-allocate the budget to the infrastructure which needs the reinforcement activities the most. The results are investigated in terms of the distribution of the reinforcement budget over the infrastructures and the average amount of investment on the reinforcement activities for every infrastructure are represented in Table 3 for independent and pooled budget cases. If the reinforcement activities are planned with a pooled budget, the budget spent on the reinforcement activities of the wastewater infrastructure is shifted mainly to the reinforcement activities of the electricity infrastructure. Furthermore, the budget spent on the reinforcement activities of the water infrastructure increases slightly. A pooled budget clearly brings flexibility to the reinforcement activities in the GA since the results show that there are different solutions of the SIIRR problem that lead to similar results in terms of the objective value.

For the SIIRR problem, one of the main difficulties in the problem environment is the interdependencies between the infrastructures. The decision-makers of the infrastructures often plan their activities without considering the interdependencies, which may cause delays in service restoration. In the third set of our experiments on managerial insights, we consider this case as a benchmark by ignoring the interdependencies while finding the first-stage decisions using the GA, and feeding the resulting first-stage decisions to FRESH after including the interdependencies. This is achieved by setting $F_{mn} = \emptyset$ while using FRESH in GA and then restoring the original set of F_{mn} while determining the second-stage decisions. The new average expected objective function values, z_{no_int}, are presented in Table 2 together with decline percentages, which can be interpreted as the price of ignoring interdependencies. Table 2 shows that on average, the loss from ignoring the interdependencies in the network while making the decisions is around 2.85%. While this might appear to be a small change, it might correspond to substantial changes, particularly in larger networks.

5.3.3. Sensitivity Analysis

In this set of experiments, we assess the sensitivity of the results to the budget levels and the number of teams. The results of these experiments are given in Table 4.

To measure the effect of the reinforcement budget, we consider three levels with regard to the budget as percent reinforcement cost of all arcs in each infrastructure. The first level, where we consider a zero budget, corresponds to the case where there is no reinforcement. In the second level, we set this at 5%. This is equivalent to our baseline runs of the heuristic. In the third level, the reinforcement budget for each infrastructure is 10% of the total reinforcement cost of all arcs in that infrastructure. The objective values for these three levels, denoted as z_{bgt0}, z_H, and z_{bgt10}, are given in Table 4 with percentage improvement values.

From Table 4, we observe that increasing the reinforcement budget from 0 to 5% brings an improvement of

Table 4: Summary of computational results regarding sensitivity analysis

Instance	z_H	z_{bgt0}	% Improvement (0% to 5%)	z_{bgt10}	% Improvement (5% to 10%)	z_{1tm}	% Improvement (1 to 2 teams)	z_{4tm}	% Improvement (2 to 4 teams)
1	72.50	60.71	19.41	75.92	4.72	65.77	10.22	75.79	4.55
2	73.93	63.45	16.52	77.19	4.42	64.70	14.26	76.93	4.06
3	71.38	59.10	20.79	75.15	5.28	63.81	11.86	73.85	3.45
4	73.34	61.98	18.32	77.70	5.95	67.43	8.77	76.26	3.98
5	69.35	59.01	17.52	73.75	6.35	61.84	12.13	72.85	5.05
6	71.41	60.67	17.71	75.63	5.90	64.14	11.34	74.57	4.41
7	74.01	63.11	17.27	77.58	4.83	66.94	10.56	76.70	3.63
8	71.20	59.94	18.78	76.18	7.00	61.46	15.85	74.48	4.62
9	72.00	59.69	20.63	76.46	6.19	65.21	10.41	74.44	3.39
10	70.94	62.30	13.87	76.17	7.37	64.80	9.49	75.69	6.69
Average	72.01	61.00	18.08	76.17	5.80	64.61	11.49	75.16	4.38

Table 5: Infrastructure performance distribution with different number of repair teams

Infrastructure (M)	**1 team**	**2 teams**	**4 teams**	$\lvert F_{mn} : m \in M \rvert$	$\lvert F_{mn} : n \in M \rvert$
Electricity	20.62	22.77	23.74	6	3
Wastewater	21.12	23.66	24.59	3	10
Water	22.87	25.58	26.82	6	2

18.08%, while increasing it further to 10% brings an additional 5.80% improvement on average. Improvement in the objective value decreases by approximately threefold from 5% to 10% reinforcement budget, pointing to decreasing marginal returns, as expected. Indeed, a reinforcement budget of 40% is observed to deem the second-stage of the problem unnecessary. That is, if there is enough budget to reinforce most of the arcs, most of the demand can be already satisfied after the disaster and the repair efforts do not have any significant effect on demand satisfaction.

Our second set of experiments evaluates the effect of the number of repair teams. We assess this by setting the number of teams in every infrastructure to 1 and 4. The resulting objective values are presented as z_{1tm} and z_{4tm} in Table 4 with percentage improvement values. Comparing these to the baseline case with 2 teams (z_H), the expected objective value increases by 11.49% on average when the number of teams in every infrastructure is increased from 1 to 2. When the number of teams in every infrastructure is increased to 4, an additional 4.38% increase is observed on average. We observe decreasing marginal returns once more as the number of teams in every infrastructure increases. These results are then analyzed in depth to see if the improvements in the performances of the infrastructures are in line with the dependencies they create in the networks. For this end, the ratio in Equation (1) is calculated for each infrastructure separately and summed up over all periods and scenarios for an instance and presented in Table 5 as averages. From Table 5 we observe that, as expected, the performances of the infrastructures improve as the number of teams increases. On the other hand, the relative performances of these infrastructures are not solely determined by the interdependency structure. For example, the wastewater infrastructure depends heavily on both the electricity and water networks to operate. Whereas it performs better than the electricity network on average, the same does not hold in comparison to the water network. This points to the fact that in determining the number of teams to assign to each infrastructure, the decision-makers should also consider the network structure, in addition to that of interdependencies.

Table 6: Summary of experiments on the effect of stochasticity

Instance	VSS				EVPI			
	z_{EEV}	$z_{B,EEV}$	% VSS_{LB}	% VSS_{UB}	z_{PI}	$z_{B,PI}$	% $EVPI_{LB}$	% $EVPI_{UB}$
1	76.12	76.16	0.00	3.79	79.44	79.51	0.56	9.68
2	75.11	75.13	0.00	7.50	81.14	81.25	0.49	9.91
3	73.79	73.81	0.00	5.24	78.08	78.13	0.54	9.46
4	74.61	74.64	0.00	6.50	80.05	80.13	0.75	9.26
5	71.44	71.45	0.00	6.55	76.41	76.46	0.38	10.26
6	72.55	72.58	0.00	7.94	78.62	78.74	0.40	10.25
7	75.82	75.91	0.00	5.80	80.68	80.72	0.57	9.08
8	75.15	75.15	0.00	4.99	79.31	79.41	0.53	11.53
9	72.25	72.34	0.00	7.65	78.07	78.16	0.37	8.55
10	74.08	74.17	0.00	7.55	80.00	80.08	0.42	12.88
Average	74.09	74.13	0.00	6.35	79.18	79.26	0.50	10.08

5.3.4. Effect of Stochasticity

To measure the effect of stochasticity, we make use of the Value of the Stochastic Solution (VSS) and the Expected Value of Perfect Information (EVPI). For the SIIRR problem, an exact calculation of the VSS and EVPI is not possible. Hence, in this section, we discuss the mechanism we devise to estimate these two measures. A summary of the experiments in this section is given in Table 6.

To estimate the VSS calculation, we first need to define the Expected Value Problem (EVP). To create the scenario to be considered in the EVP, i.e., the expected value scenario, the damage rate is set to 50% for all infrastructures. For each infrastructure, arcs are ranked in descending order of the total number of scenarios in which they are damaged. The first 50% of all arcs for each infrastructure are assumed as damaged in the EVP. As it is computationally exhaustive to find the optimal objective function value of the SIIRR-SP and the optimal expected result of using the EVP solution, we use lower and upper bounds on the VSS values. The EVP for each instance is solved using the SIIRR-SP, with a 3-hour time limit and the expected value scenario, and the first-stage decisions are fed to the SIIRR-SP (with a 3-hour time limit for each scenario) with all scenarios $s \in S$ to calculate the expected result of using the EVP, denoted as z_{EEV}. The best bound on the expected result of using the EVP, denoted as $z_{B,EEV}$, which is reported by CPLEX, is also recorded. A lower bound and an upper bound on the VSS value are calculated by using z_H (the baseline heuristic results obtained by running the GA and FRESH) and z_B (the best bound of the SIIRR-SP after running it for three days) as:

$$\%VSS_{LB} = \frac{\max\{0, z_H - z_{B,EEV}\}}{z_{B,EEV}} \times 100\% \text{ and } \%\ VSS_{UB} = \frac{z_B - z_{EEV}}{z_{EEV}} \times 100\%.$$

The lower and upper bounds on the VSS values for each instance are given in Table 6, together with the z_{EEV} and $z_{B,EEV}$ values. The lower bounds on VSS are zero for all the instances, while the upper bounds range from 3.79% to 7.94%. The VSS is observed to be at most 6.35% on average over all instances. As the lower bounds on the VSS are all zero, capturing stochasticity in the problem structure through stochastic programming may be of little value, and solving EVP may provide a high-quality solution. However, as the VSS upper bound values are not negligibly small, it is also possible that using a stochastic model rather than a deterministic model may yield significantly better solutions.

The EVPI, which can be used as a measure to demonstrate the importance of the information being revealed after the disaster, is calculated as the difference between the case with perfect information about the second-stage of the problem before the disaster and the optimal two-stage SIIRR-SP solution. As the optimal solution of the

SIIRR-SP is not known, a percent lower bound on the EVPI is calculated by using z_B as:

$$\%\ EVPI_{LB} = \frac{\max\{0, z_{PI} - z_B\}}{z_B} \times 100\% = \frac{\max\left\{0, \sum_{s\in S} P_s z_s - z_B\right\}}{z_B} \times 100\%,$$

where z_s is the objective function value of scenario $s \in S$ in the perfect information case. The z_s values are calculated for each scenario by running the SIIRR-SP model with a 3-hour time limit for the corresponding scenario. Similarly, a percent upper bound on the EVPI is calculated by using z_H as:

$$\%\ EVPI_{UB} = \frac{z_{B,PI} - z_H}{z_H} \times 100\% = \frac{\sum_{s\in S} P_s z_{B,s} - z_H}{z_H} \times 100\%,$$

where the $z_{B,s}$ values are the best-bound values reported by CPLEX after running the SIIRR-SP with a 3-hour time limit model for the corresponding scenario.

The lower and upper bounds on the EVPI values can be seen in Table 6, together with the z_{PI} and $z_{B,PI}$ values. For all instances, EVPI lower bounds are lower than 1%, with an average of 0.5%. EVPI upper bounds range from 8.55% to 12.88%, with an average of 10.08%. These figures should be interpreted by a decision-maker to understand their significance in infrastructure services. However, one may claim that having more information on the course of the disaster may improve the results as the upper bound on the EVPI values are as high as 12% for some of the instances. On the other hand, attempting to fine-tune the available imperfect information may not be worth the effort considering the lower bounds on the EVPI values.

6. Conclusions and Further Research Directions

Interdependency is rooted in the infrastructures providing different services to the populations, as their operations are mutually dependent on one another. Over the recent decades, disasters have impacted the interdependent functioning of the infrastructures in affected areas. Furthermore, there are inherent uncertainties regarding the disasters, since their intensity, type, and location are not generally known beforehand. To cope with these challenges, the literature has focused on the operations and the post-disaster repair of interdependent infrastructures more heavily in recent years.

Motivated by the fact that reinforcement of infrastructure networks may mitigate the post-disaster damages and facilitate the subsequent flow of services, this paper has defined the Stochastic Interdependent Infrastructure Reinforcement and Repair (SIIRR) Problem by modeling the infrastructures as layered networks and incorporating the uncertainty of which infrastructure segments will be damaged. A two-stage stochastic mixed-integer program for its solution with the objective of maximizing the services provided by the infrastructures after the disaster over a fixed planning horizon has been presented.

Due to the exhaustive computational effort to solve the SIIRR problem to optimality, we have developed a heuristic approach to solve the SIIRR problem, which makes use of a genetic algorithm for the first-stage decisions and uses the Flow and Repair Scheduling Heuristic (FRESH) for the second-stage. In order to reduce the computational burden of the GA, a representative scenario is obtained from the possible disaster scenarios. The proposed heuristic approach is able to find high-quality solutions in reasonable amount of run times, whereas CPLEX is unable to find an optimal solution in multiple days. Furthermore, the optimality gaps of our heuristic are within acceptable ranges.

The devised heuristic can be easily applied to different infrastructures, even virtual ones, such as the internet. In that case, the informational necessities of the virtual infrastructures can be represented as interdependencies. Moreover, decision-makers may benefit from using the heuristic for valuable insights on the infrastructures and the delivery of their services in the aftermath of a disruption. Running the heuristic for different sets of scenarios, one can determine the critical elements of the infrastructures, which should be prioritized in the reinforcement or repair operations. Additionally, the interdependencies which are vital for the functioning of the infrastructures may be determined and potentially vulnerable services may be pre-positioned in the interdependent nodes.

In this paper, we have only accounted for operational interdependencies among the infrastructures. Although it may be challenging, one may include restorational interdependencies, i.e., those faced during the restoration efforts, in the problem environment. Similarly, partial damage on the arcs of the infrastructures can be integrated to the problem. In the problem environment of the SIIRR problem, repair times are independent of the disaster scenario. Particularly in cases where partial damage is involved, the repair time for damaged segments may depend on the scenario. In the devised heuristic, GA employs one representative scenario to reduce the computational burden. The number of representative scenarios may be increased for better representation considering the trade-off between the first-stage solution quality and computation time. Another promising research direction may be incorporating the devised second-stage heuristic, FRESH, to a decomposition algorithm to obtain better solutions. To aid decision makers, a more detailed exploration of how many repair teams to assign for each infrastructure to maximize the overall system performance would help increase the effectiveness of our proposed approaches. Lastly, uncertainty structure may be incorporated in different ways. In our problem environment, we have assumed that once an arc is reinforced, it will not be damaged during the disaster. One may relax this assumption and determine a probability according to which a reinforced arc is damaged. This relaxation introduces decision-dependent uncertainty into the problem structure and poses a valuable research direction.

References

Ajam, M., Akbari, V., & Salman, F. S. (2019). Minimizing latency in post-disaster road clearance operations. *European Journal of Operational Research*, *277*, 1098–1112.

Ajam, M., Akbari, V., & Salman, F. S. (2021). Routing multiple work teams to minimize latency in post-disaster road network restoration. *European Journal of Operational Research*, *to appear*.

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

Akbari, V., & Salman, F. S. (2017b). Multi-vehicle prize collecting arc routing for connectivity problem. *Computers & Operations Research*, *82*, 52–68.

Almoghathawi, Y., & Barker, K. (2019). Component importance measures for interdependent infrastructure network resilience. *Computers & Industrial Engineering*, *133*, 153 – 164.

Aslan, E., & Çelik, M. (2019). Pre-positioning of relief items under road/facility vulnerability with concurrent restoration and relief transportation. *IISE Transactions*, *51*, 847–868.

Averbakh, I. (2012). Emergency path restoration problems. *Discrete Optimization*, *9*, 58–64.

Averbakh, I., & Pereira, J. (2012). The flowtime network construction problem. *IIE Transactions*, *44*, 681–694.

Baxter, M., Elgindy, T., Ernst, A. T., Kalinowski, T., & Savelsbergh, M. W. (2014). Incremental network design with shortest paths. *European Journal of Operational Research*, *238*, 675–684.

Botton, Q., Fortz, B., & Gouveia, L. (2015). On the hop-constrained survivable network design problem with reliable edges. *Computers & Operations Research*, *64*, 159 – 167.

Botton, Q., Fortz, B., Gouveia, L., & Poss, M. (2013). Benders decomposition for the hop-constrained survivable network design problem. *INFORMS Journal on Computing*, *25*, 13–26.

Cavdaroglu, B., Hammel, E., Mitchell, J., Sharkey, T., & Wallace, W. (2013). Integrating restoration and scheduling decisions for disrupted interdependent infrastructure systems. *Annals of Operations Research*, *203*, 1–16.

Çelik, M. (2016). Network restoration and recovery in humanitarian operations: Framework, literature review, and research directions. *Surveys in Operations Research and Management Science*, *21*, 47–61.

Chang, L., Peng, F., Ouyang, Y., Elnashai, A. S., & Spencer, B. F. (2012). Bridge seismic retrofit program planning to maximize postearthquake transportation network capacity. *Journal of Infrastructure Systems*, *18*, 75–88.

Chimani, M., Kandyba, M., Ljubic, I., & Mutzel, P. (2010). Orientation-based models for 0,1,2-survivable network design: Theory and practice. *Mathematical Programming*, *124*, 413–439.

Coffrin, C., Van Hentenryck, P., & Bent, R. (2012). Last-mile restoration for multiple interdependent infrastructures. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence* (pp. 455–463). volume 1.

Diarrassouba, I., Gabrel, V., Mahjoub, A. R., Gouveia, L., & Pesneau, P. (2016). Integer programming formulations for the k-edge-connected 3-hop-constrained network design problem. *Networks*, *67*, 148–169.

Du, L., & Peeta, S. (2014). A stochastic optimization model to reduce expected post-disaster response time through pre-disaster investment decisions. *Networks and Spatial Economics*, *14*, 271–295.

Dudenhoeffer, D. D., Permann, M. R., & Manic, M. (2006). Cims: A framework for infrastructure interdependency modeling and analysis. In *Proceedings of the 2006 Winter Simulation Conference* (pp. 478–485).

Enayaty Ahangar, N., Sullivan, K. M., & Nurre, S. G. (2020). Modeling interdependencies in infrastructure systems using multi-layered network flows. *Computers & Operations Research*, *117*, 104883.

Fan, Y., Liu, C., Lee, R., & Kiremidjian, A. S. (2010). Highway network retrofit under seismic hazard. *Journal of Infrastructure Systems*, *16*, 181–187.

Fang, Y., Fang, C., Zio, E., & Xie, M. (2019). Resilient critical infrastructure planning under disruptions considering recovery scheduling. *IEEE Transactions on Engineering Management*, *68*, 452–466.

Fang, Y., & Sansavini, G. (2018). Optimum post-disruption restoration under uncertainty for enhancing critical infrastructure resilience. *Reliability Engineering & System Safety*, *185*, 1–11.

Fang, Y., & Zio, E. (2019). An adaptive robust framework for the optimization of the resilience of interdependent infrastructures under natural hazards. *European Journal of Operational Research*, *276*, 1119–1136.

Faturechi, R., & Miller-Hooks, E. (2015). Measuring the performance of transportation infrastructure systems in disasters: A comprehensive review. *Journal of Infrastructure Systems*, *21*, 04014025.

Ghaffarpour, R., Oskuee, M. R. J., & Ranjbar, A. M. (2020). Resilience-oriented distribution network optimal planning to improve the continuity of power supply. *International Journal of Ambient Energy*, *41*, 466–474.

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.

Gomez, C., González, A. D., Baroud, H., & Bedoya-Motta, C. D. (2019). Integrating operational and organizational aspects in interdependent infrastructure network recovery. *Risk Analysis*, *39*, 1913–1929.

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*, 334–350.

Guha-Sapir, D., Hoyois, P., Below, R., & Vanderveken, A. (2016). *Annual Disaster Statistical Review 2015: The numbers and trends*. Technical Report Centre for Research on the Epidemiology of Disasters.

He, X., & Cha, E. (2018). Modeling the damage and recovery of interdependent civil infrastructure network using dynamic integrated network model. *Sustainable and Resilient Infrastructure*, *5*, 1–16.

He, X., & Cha, E. J. (2020). Din ii: incorporation of multi-level interdependencies and uncertainties for infrastructure system recovery modeling. *Structure and Infrastructure Engineering*, *0*, 1–16.

He, X., & Yuan, Y. (2019). A framework of identifying critical water distribution pipelines from recovery resilience. *Water Resources Management*, *33*, 3691–3706.

Heath, E., Mitchell, J., & Sharkey, T. (2016). Applying ranking and selection procedures to long-term mitigation for improved network restoration. *EURO Journal on Computational Optimization*, *4*, 448.

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

Kalinowski, T., Matsypura, D., & Savelsbergh, M. (2013). Incremental network design with maximum flows. *European Journal of Operational Research*, *242*, 51–62.

Karakoc, D. B., Almoghathawi, Y., Barker, K., González, A. D., & Mohebbi, S. (2019). Community resilience-driven restoration model for interdependent infrastructure networks. *International Journal of Disaster Risk Reduction*, *38*, 101228.

Karakoc, D. B., Barker, K., Zobel, C. W., & Almoghathawi, Y. (2020). Social vulnerability and equity perspectives on interdependent infrastructure network component importance. *Sustainable Cities and Society*, *57*, 102072.

Krzywinski, M. (2010). Linear layout for visualization of networks. URL: `http://www.hiveplot.net/talks/hive-plot.pdf` accessed: August 2020.

LeBlanc, L. J., Morlok, E. K., & Pierskalla, W. P. (1975). An efficient approach to solving the road network equilibrium traffic assignment problem. *Transportation Research*, *9*, 309–318.

Lee, E. E., Mitchell, J. E., & Wallace, W. A. (2007). Restoration of services in interdependent infrastructure systems: A network flows approach. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *37*, 1303–1317.

Lee, J., & Rao, H. R. (2009). Exploring the causes and effects of inter-agency information sharing systems adoption in the anti/counter-terrorism and disaster management domains. In *The Proceedings of the 8th Annual International Digital Government Research Conference* (pp. 155–163). Philadelphia, USA: Digital Government Society of North America.

Loggins, R., Little, R., Mitchell, J., Sharkey, T., & Wallace, W. (2019). Crisis: Modeling the restoration of interdependent civil and social infrastructure systems following an extreme event. *Natural Hazards Review*, *20*, 04019004.

Lu, J., Gupte, A., & Huang, Y. (2018). A mean-risk mixed integer nonlinear program for transportation network protection. *European Journal of Operational Research*, *265*, 277 – 289.

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

Álvarez Miranda, E., & Pereira, J. (2017). Designing and constructing networks under uncertainty in the construction stage: Definition and exact algorithmic approach. *Computers & Operations Research*, *81*, 178–191.

Moreno, A., Munari, P., & Alem, D. (2018). A branch-and-benders-cut algorithm for the crew scheduling and routing problem in road restoration. *European Journal of Operational Research*, *275*, 16–34.

Najafi, J., Peiravi, A., & Guerrero, J. (2018). Power distribution system improvement planning under hurricanes based on a new resilience index. *Sustainable Cities and Society*, *39*, 592–604.

Najafi Tari, A., Sepasian, M. S., & Tourandaz Kenari, M. (2021). Resilience assessment and improvement of distribution networks against extreme weather events. *International Journal of Electrical Power & Energy Systems*, *125*, 106414.

Nurre, S., & Sharkey, T. (2014). Integrated network design and scheduling problems with parallel identical machines: Complexity results and dispatching rules. *Networks*, *63*, 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*, 794–806.

Ouyang, M. (2017). A mathematical framework to optimize resilience of interdependent critical infrastructure systems under spatially localized attacks. *European Journal of Operational Research*, *262*, 1072–1084.

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

Ouyang, M., & Wang, Z. (2015). Resilience assessment of interdependent infrastructure systems: With a focus on joint restoration modeling and analysis. *Reliability Engineering & System Safety*, *141*, 74–82.

Peeta, S., Salman, F. S., Gunnec, D., & Viswanath, K. (2010). Pre-disaster investment decisions for strengthening a highway network. *Computers & Operations Research*, *37*, 1708–1719.

Permann, M. R. (2007). Toward developing genetic algorithms to aid in critical infrastructure modeling. In *2007 IEEE Conference on Technologies for Homeland Security* (pp. 192–197).

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

Sato, T., & Ichii, K. (1996). Optimization of post-earthquake restoration of lifeline networks using genetic algorithms. *Doboku Gakkai Ronbunshu*, *1996*, 245–256.

Sharkey, T., Nurre, S., Nguyen, H., Chow, J., Mitchell, J., & Wallace, W. (2015a). Identification and classification of restoration interdependencies in the wake of hurricane sandy. *Journal of Infrastructure Systems*, *22*, 1–12.

Sharkey, T. C., Cavdaroglu, B., Nguyen, H., Holman, J., Mitchell, J. E., & Wallace, W. A. (2015b). Interdependent network restoration: On the value of information-sharing. *European Journal of Operational Research*, *244*, 309–321.

Sharkey, T. C., Nurre Pinkley, S. G., Eisenberg, D. A., & Alderson, D. L. (2021). In search of network resilience: An optimization-based view. *Networks*, *77*, 225–254.

Sharma, N., Tabandeh, A., & Gardoni, P. (2019). Recovery optimization of interdependent infrastructure: A multi-scale approach. In *The Proceedings of the 13th International Conference on Applications of Statistics and Probability in Civil Engineering(ICASP13)*. Seoul, South Korea.

Sharma, N., Tabandeh, A., & Gardoni, P. (2020). Regional resilience analysis: A multiscale approach to optimize the resilience of interdependent infrastructure. *Computer-Aided Civil and Infrastructure Engineering*, *35*, 1–16.

Shen, S. (2013). Optimizing designs and operations of a single network or multiple interdependent infrastructures under stochastic arc disruption. *Computers & Operations Research*, *40*, 2677–2688.

Smith, A. M., González, A. D., Dueñas-Osorio, L., & D'Souza, R. M. (2020). Interdependent network recovery games. *Risk Analysis*, *40*, 134–152.

Talebiyan, H., & Dueñas-Osorio, L. (2019). Probabilistic assessment of decentralized decision-making for interdependent network restoration. In *Proceedings of the 13th International Conference on Applications of Statistics and Probability in Civil Engineering (ICASP13)* (pp. 1–8). Seoul, South Korea.

Talebiyan, H., & Dueñas-Osorio, L. (2020). Decentralized decision making for the restoration of interdependent networks. *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering*, *6*, 04020012.

Xu, N., Guikema, S. D., Davidson, R. A., Nozick, L. K., Çağnan, Z., & Vaziri, K. (2007). Optimizing scheduling of post-earthquake electric power restoration tasks. *Earthquake Engineering & Structural Dynamics*, *36*, 265–284.

Yaghmaei, N., Below, R., & Wathelet, V. (2019). *2018 Annual Disaster Statistics Report*. Technical Report Centre for Research on the Epidemiology of Disasters.