MANAGEMENT OF INTERDEPENDENT INFRASTRUCTURE NETWORKS UNDER DISASTER-RELATED UNCERTAINTIES

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

TUĞÇE CANBİLEN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
INDUSTRIAL ENGINEERING

AUGUST 2020

Approval of the thesis:

MANAGEMENT OF INTERDEPENDENT INFRASTRUCTURE NETWORKS UNDER DISASTER-RELATED UNCERTAINTIES

submitted by **TUĞÇE CANBİLEN** in partial fulfillment of the requirements for the degree of **Master of Science in Industrial Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** ________________

Prof. Dr. Esra Karasakal
Head of Department, **Industrial Engineering** ________________

Assist. Prof. Dr. Sakine Batun
Supervisor, **Indsutrial Engineering, METU** ________________

Assoc. Prof. Dr. Melih Çelik
Co-supervisor, **School of Management, University of Bath** ________________

Examining Committee Members:

Prof. Dr. Meral Azizoğlu
Industrial Engineering, METU ________________

Assist. Prof. Dr. Sakine Batun
Industrial Engineering, METU ________________

Assoc. Prof. Dr. İsmail Serdar Bakal
Industrial Engineering, METU ________________

Assist. Prof. Dr. Özlem Çavuş İyigün
Industrial Engineering, Bilkent University ________________

Prof. Dr. Esra Karasakal
Industrial Engineering, METU ________________

Date: 14.08.2020

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Tuğçe Canbilen

Signature :

ABSTRACT

MANAGEMENT OF INTERDEPENDENT INFRASTRUCTURE NETWORKS UNDER DISASTER-RELATED UNCERTAINTIES

Canbilen, Tuğçe
M.S., Department of Industrial Engineering
Supervisor: Assist. Prof. Dr. Sakine Batun
Co-Supervisor: Assoc. Prof. Dr. Melih Çelik

August 2020, 101 pages

During a disaster, multiple infrastructures such as power, water, or telecommunications networks may face disruptions in their services. Services of these infrastructures are vital in the aftermath of a disaster to facilitate search-and-rescue activities, relief transportation, and restoration efforts. Furthermore, the operations of these infrastructures may depend on receiving services from others, resulting in an interdependent network structure. In the aftermath of a disaster, damages on the network segments are observed and repair activities are scheduled accordingly. Repair activities need to be planned efficiently and in limited amount of time, taking into account the interdependencies between these networks.

Reinforcement of network components prior to a disaster helps mitigate the amount of damage on the infrastructures and may play an important role in reducing the need for repairs in the aftermath of the disaster. On the other hand, reinforcement activities are costly and need to take into account uncertainties related to the effects of a potential disaster on the network. To integrate the pre-disaster reinforcement and post-disaster repair activities, the Stochastic Interdependent Infrastructure Reinforcement and Repair (SIIRR) problem is defined in this study. A scenario-based two-stage stochastic program is proposed to model the SIIRR problem, and a heuristic making use of the genetic algorithm and partial optimization is devised. The heuristic is then tested on realistic instances to observe its performance and make managerial inferences.

Keywords: humanitarian logistics, interdependent infrastructures, disaster management, stochastic programming, genetic algorithm

ÖZ

FELAKETE DAYALI BELİRSİZLİKLER ALTINDAKİ BİRBİRİNE BAĞIMLI ALTYAPI AĞLARININ YÖNETİMİ

Canbilen, Tuğçe
Yüksek Lisans, Endüstri Mühendisliği Bölümü
Tez Yöneticisi: Dr. Öğr. Üyesi. Sakine Batun
Ortak Tez Yöneticisi: Doç. Dr. Melih Çelik

Ağustos 2020 , 101 sayfa

Afetler, elektrik, su ve telekomünikasyon gibi altyapı ağlarının hizmetlerini kesintiye uğratabilir. Afet sonrası bu altyapı hizmetleri, arama-kurtarma, yardım malzemelerinin taşınması ve restorasyon gibi faaliyetlerin sürdürülebilmesi için hayati önem taşımaktadır. Öte yandan, her bir altyapı ağının hizmet verebilmesi için başka altyapılardan hizmet alması gerekebilir. Bu durum, birbirine bağımlı bir ağ yapısına neden olur. Afet sonrası onarım faaliyetlerinin kısıtlı zamanda verimli bir şekilde planlanabilmesi için, bu birbirine bağımlı yapının da göz önünde bulundurulması gerekir.

Afet öncesi bazı ağ parçalarının güçlendirilmesi, olası hasarı ve afet sonrası tamir ihtiyacını önemli ölçüde azaltmaya yardımcı olur. Bununla birlikte ağ güçlendirme hem yüksek maliyet, hem de afetin etkileriyle ilgili belirsizliklerin göz önünde bulundurulmasını gerektirir. Buradan yola çıkarak bu çalışmada afet öncesi ağ güçlendirme, afet sonrasında ise hasar gören ağ parçalarının onarımı ve ağ hizmetlerinin akışını bütünleşik olarak planlamak amacıyla Stokastik Birbirine Bağımlı Ağ Güçlendirme ve Onarım Problemi tanımlanmıştır. Problemin modellenmesinde senaryo temelli iki

aşamalı stokastik programlama yöntemi kullanılmış, modelin çözümü içinse genetik algoritma ve kısmi optimizasyondan yararlanan bir sezgisel geliştirilmiştir. Sezgisel yöntemin performansını gözlemlemek ve yönetimsel çıkarımlar yapmak amacıyla gerçekçi ağ örneklerinin kullanıldığı sayısal deneylere başvurulmuştur.

Anahtar Kelimeler: insani yardım lojistiği, bağımlı altyapılar, felaket yönetimi, stokastik programlama, genetik algoritma

To all disaster victims...

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my advisors Assist. Prof. Dr. Sakine Batun and Assoc. Prof. Dr. Melih Çelik for their endless support during this thesis study and believing in me. Every time I thought I hit a wall, they are the ones opening a door on it and encouraging me to go through. I could not ask for better advisors for this work. They are my role models and mentors in academia and always will be. I am also grateful to Assoc. Prof. Dr. Melih Çelik for introducing me to Humanitarian Logistics with an undergraduate course. If I believe I made a contribution, that is because I have made my research in this area.

I would like to present my special thanks to the examining committee members of this thesis Prof. Dr. Meral Azizoğlu, Prof. Dr. Esra Karasakal, Assoc. Prof. Dr. İsmail Serdar Bakal and Assist. Prof. Dr. Özlem Çavuş İyigün for their valuable contribution and feedback.

Moreover, I thank all my friends and family who supported me emotionally and showed me patience during this process. Without their support, I could not have come this far in my education. I appreciate their belief in me and finding them right beside me when I needed.

I am also very grateful to all professors of METU-IE Department for providing me with knowledge to pursue a master's program and helping me find my way in academia. Last but not least, I would like to thank to TÜBİTAK 2211 National Graduate Scholarship Program for supporting me to pursue a master's program.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

MIP	Mixed-Integer Programming
SIIRR	Stochastic Interdependent Infrastructure Reinforcement and Repair
GA	Genetic Algorithm
FRESH	Flow and Repair Scheduling Heuristic
FRSCH	Flow and Repair Schedule Construction Heuristic
RSIH	Repair Schedule Improvement Heuristic
ANOVA	Analysis of Variation
VSS	Value of the Stochastic Solution
EVP	Expected Value Problem
EVPI	Expected Value of Perfect Information

CHAPTER 1

INTRODUCTION

Disasters are sudden catastrophic events disrupting the operations of daily life and the nature. They can be human-inflicted, such as terrorist attacks, nuclear leaks and arson, or natural such as earthquakes, hurricanes, floods, tsunamis and wildfires. In recent decades, both natural and human-inflicted disasters have increased in number and in intensity (Guha-Sapir et al., 2016). Recent examples of large-scale disasters in the previous decade include the Haiti Earthquake in 2010, Hurricane Sandy in 2012, Typhoon Haiyan in 2013, the Nepal Earthquake in 2015, Hurricanes Harvey and Maria in 2017, and the wildfires in the Amazon, Indonesia, and Australia in 2019. In some cases, multiple cascading disasters have exacerbated the relief efforts. Two recent examples are the earthquake and tsunami in Southeast Asia in 2004 and the Tohoku Earthquake, followed by a tsunami and nuclear leak in the Fukushima Nuclear Plant in 2011. Due to the upward trends in population, urbanization, global warming and economic activity, disasters have had increasing impacts on human life and well-being, global economy, and the environment (Yaghmaei et al., 2019). In the light of this information, it is clear that any effort to prevent or reduce these impacts and to provide relief is very valuable.

Efforts to address the disaster effects are gathered under the roof of disaster management. Disaster management is divided into four stages: mitigation, preparedness, response and recovery, in accordance with the disaster timeline. Mitigation efforts are those trying to reduce the possibility of occurrence of a disaster or the effects of one on the people, economy, and the nature. Structural strengthening the highways is an example of mitigation. Preparedness activities are performed before the disaster so as to facilitate the response to the disaster efficiently. Pre-positioning relief items in critical locations is an example of these. The mitigation and preparedness stages include pro-active measures to a disaster and face the uncertainty of its type, intensity and location. As there are infinitely many possibilities of the course of a disaster and resources are limited, monetary concerns are in the foreground in these stages. On the other hand, response and recovery stages are re-active, and include measures taken after the disaster. Response activities, such as search-and-rescue, relief transportation, and repair efforts, are of the highest urgency, since these have the most profound and direct effect on saving human lives and minimizing the impact of the disaster. As there is a possibility of loss of life and critical damage to the living areas, urgency is in the foreground and monetary concerns are of secondary importance. Recovery efforts are performed later on to make sure that society returns to their normal life, or at least overcomes the long-term effects of the disaster as promptly as possible. These efforts generally last longer periods of time and monetary concerns are sometimes inevitably on the foreground.

Services of the critical infrastructures are vital for the society to maintain everyday activities. The critical infrastructures are classified by Lee et al. (2007) under four headings, namely transportation, energy, telecommunication and water. Some examples of each class of critical infrastructures are highways, power networks, the internet, and wastewater infrastructures, respectively. In general, these infrastructures have their own structural elements. However, their operations are often interdependent, i.e., their activities performed to provide the services are mutually dependent on one another. For instance, water treatment plants need electricity to function and wastewater infrastructure should be functional to discharge the flood in a flooded highway infrastructure.

In the aftermath of a disaster, services of the critical infrastructures are needed not only to return to normalcy, but also for the search and rescue activities and distribution of relief items. For instance, if road segments fail, we may not reach people in need or transport relief supplies; if the power network is disrupted, search-and-rescue equipment may not work. According to the U.S. Department of Energy, Office of Electricity Delivery and Energy Reliability, one week after Hurricane Sandy hit USA, 650,416 customers were still out of power in the affected states (Heath et al., 2016). That is why these services should be restored (repaired) as quickly as possible in the response stage once disrupted. Given the importance of timeliness in the response stage, scheduling of these repairs plays an important role in determining the effectiveness of response activities. The scheduling of repairs for different infrastructures should take into consideration the interdependency between these, as there will be delays in the restoration of these critical infrastructures otherwise, which means losing time, a precious resource, in the aftermath of a disaster. Given the importance of the post-disaster repair of interdependent infrastructures, a considerable number of studies have been conducted to address the scheduling of these repairs (e.g., Coffrin et al., 2012; Cavdaroglu et al., 2013).

Although infrastructure repair decisions in the response stage play a crucial role in supporting response activities, a more effective approach would be to take preventive measures in the mitigation stage to make sure that these infrastructures are disrupted to the smallest extent possible. As explained in Heath et al. (2016), with the help of the mitigation activities (e.g., infrastructure reinforcement), both the services of critical infrastructures can be restored rapidly and the impact of another disaster is reduced. In making the reinforcement decisions, uncertainties in the disaster type, intensity and location, as well as the interdependencies between the networks affect the mitigation and response activities. These activities need to be planned regarding the possible outcomes of the disaster. This points to the need to address the decisions of network reinforcement, repair scheduling, and network flow decisions in an integrated manner, under disaster-related uncertainties. To the best of our knowledge, no such study exists in the literature to aid these decisions for interdependent networks.

To bridge this gap in the literature, we define and study the Stochastic Interdependent Infrastructure Reinforcement and Repair (SIIRR) problem in this thesis, where we incorporate the inherent uncertainty of which network segments will be disrupted. In the first (mitigation) stage of the problem, where we address the operations during mitigation, reinforcement decisions are considered. In the second (response) stage, network flow and repair decisions are focused on. For the mitigation effort, monetary concerns are incorporated by means of a budget, whereas demand satisfaction is prioritized for the response efforts. Infrastructures are modeled as layered networks, and flows in the network correspond to the services provided by the infrastructures. The objective of the SIIRR problem is to maximize the overall performance of the infrastructures throughout the planning horizon. While doing so, operational interdependencies between the nodes of different infrastructures and potential disaster scenarios are considered.

We propose a two-stage stochastic mixed-integer linear program for the SIIRR problem. The aim of the proposed model is to maximize the services of the infrastructures after the disaster. The formulation of the problem includes constraints related to reinforcement, repair scheduling, interdependency, and demand satisfaction. The reinforcement efforts are limited with a reinforcement budget and the repair efforts are scheduled to the repair teams in order to reach to the demand nodes quickly, while considering the interdependencies between the infrastructures. A heuristic method making use of a genetic algorithm and partial optimization is devised for the solution of this model. The performance of the heuristic is tested on instances generated by modifying those from the literature, and evaluated by comparison to the upper bound on the stochastic program's optimal objective function value. The devised heuristic is then used to make useful managerial inferences, such as the effect of network reinforcement, independent vs. coordinated repair actions for the networks, and decoupling of the interdependencies between the networks, as well as sensitivity analysis for the decision makers of the problem.

The organization of this thesis is as follows. In Chapter 2, relevant literature for the problem environment is reviewed. In Chapter 3, research questions of this study are shared, the SIIRR problem is defined and the two-stage stochastic mixed-integer program formulation is presented. Solution methodology for this problem, which is a heuristic making use of a genetic algorithm and partial optimization, is explained in detail in Chapter 4. Computational experiments assessing the performance of the heuristic and answering the research questions of this thesis are shared in Chapter 5. Chapter 6 concludes the contributions of this study and points out future research directions.

CHAPTER 2

LITERATURE REVIEW

A substantial amount of studies exists on the restoration and repair of infrastructure networks. A comprehensive review of this literature is conducted in Çelik (2016). In this chapter, we classify these studies into four groups: (i) studies in which individual infrastructures are reinforced, (ii) problems where individual infrastructures are restored, (iii) studies where multiple interdependent infrastructures are considered, and (iv) applications of genetic algorithms for infrastructure restoration.

2.1 Infrastructure Reinforcement Problems

Infrastructure reinforcement problems are studied by different disciplines in the literature. In this section, studies that are the most relevant to our problem are shared. For other problems in this stream, one can refer to the literature review for the reinforcement activities in transportation networks under physical damage presented in Faturechi and Miller-Hooks (2015).

Peeta et al. (2010) consider the stochastic reinforcement problem. The authors attempt to strengthen a highway network before a disaster disrupts their services considering the random failures on the arcs during the disaster. The set of arcs to strengthen is limited with an available budget and the aim is to minimize the cost of reaching the demand nodes after the disaster. Connectivity of the network is focused on with uncapacitated arcs. The authors propose a path-based deterministic equivalent form of the two-stage stochastic programming model for the problem and provide structural results. Building on this work, Du and Peeta (2014) include the type of the disaster as a second uncertainty level into the problem structure. The authors propose a two-stage stochastic program determining the arcs to be reinforced before the disaster in order to minimize the expected response time to the demand nodes. Reinforcement investment on an arc decreases the probability of an arc to be damaged in a disaster. Du and Peeta (2014) consider three features of an arc to make the investment decision, namely the importance for network connectivity, for network traffic flow and the marginal increase in the survivability. The authors devise a two-stage heuristic algorithm to manage the complexity of the problem without compromising the solution quality.

Ouyang and Fang (2017) propose a three-level decision making model for intentional attacks to critical infrastructures. The authors model the problem as a defender-attacker-defender model with an aim of maximizing the system resilience. Resilience is quantified as the ratio of the actual performance to the target performance of the infrastructure. In the first level, the defender chooses the arcs and nodes to defend or build, and the attacker chooses the nodes and the arcs to destroy. After the attack, the defender decides on the arcs to repair and the sequencing of these repairs, together with the flow amounts. Ouyang and Fang (2017) devise a decomposition algorithm for this problem and provide managerial insights.

2.2 Infrastructure Restoration Problems

A vast majority of the literature on infrastructure restoration is focused on the restoration of critical infrastructures independently from each other. Many studies including Ang (2006) and Xu et al. (2007) propose solutions to efficiently restore the electricity infrastructure after a disaster. Averbakh (2012) devises polynomial-time algorithms for the restoration of the arcs of a transportation network whose services are disrupted by a disaster. The aim is to reach the demand nodes as soon as possible from pre-determined depots using multiple repair teams. Repair teams move to the demand points by repairing the paths they traverse, and repair time decreases when multiple teams work on the same repair. Averbakh and Pereira (2012) also try to install arcs to the disrupted network to minimize the recovery time of each node, which is defined as the first time period in which a path exists from a depot to a demand node. Authors present a MIP formulation for the problem and propose exact solution methods as well as heuristics, taking advantage of the combinatorial structure of the problem.

Maya Duque et al. (2016) work on the Network Repair Crew Scheduling and Routing Problem aiming to minimize the weighted accessibility time of the demand nodes in a one-depot and one-repair crew environment where the nodes may be damaged during a disaster. The authors propose to use dynamic programming and an iterated greedy randomized constructive procedure for solutions of small-sized and larger-sized instances respectively. Consequently, they estimate the number of nodes that should be repaired to make all the demand nodes accessible. Later, Moreno et al. (2018) develop a branch-and-Benders-cut algorithm for the same problem. The authors propose making the scheduling decisions in the master problem while the routing decisions are located in the sub-problems. With this decomposition, routing problems are equivalent to shortest path problems in the sub-problems. The authors also provide some valid inequalities for the master problem. Akbari and Salman (2017) define the Multi-Vehicle Synchronized Arc Routing Problem to Restore Network Connectivity in a post-disaster road network environment to generate synchronized work schedules for the road clearance teams aiming to restore network connectivity. Knowing which road segments are damaged, the authors also provide paths for the repair teams to traverse in order to reach the road segment to be repaired. Therefore, the paths should be synchronized considering the repairs completed by the other repair teams. The authors provide an exact MIP formulation for the problem and propose a relaxation-based heuristic making use of the local neighborhood search to improve the relaxed solution.

Ransikarbum and Mason (2016) present the Multi-Objective Integrated Response and Recovery Problem and formulate the problem as a goal programming model in a single time period environment. Response efforts include the relief item distribution, while the recovery efforts are node and arc repairing activities. Different objectives including equity-based and cost-based ones are considered and efficient frontiers are provided for the decision makers to analyze the trade-offs between these objectives.

Nurre et al. (2012) define the Integrated Network Design and Scheduling (INDS) problem. They propose a MIP model to determine which arcs to repair in an infrastructure during the planning horizon in order to maximize the weighted demand that can be sent from the supply to demand nodes over the planning horizon. After defining the arcs to repair, they use a heuristic dispatching rule for scheduling the repairs. They also propose some valid inequalities for the MIP to strengthen the formulation. Nurre and Sharkey (2014) show that the INDS problems are NP-hard with performance measures of maximum flow, minimum cost flow, shortest path and minimum spanning tree, and propose a dispatching rule algorithm framework for the problems. Kalinowski et al. (2013) also present complexity analysis and approximation algorithms for the INDS problem. Baxter et al. (2014) examine the structure of the optimal solutions of the INDS problem. They discover that the problem is relatively easier to solve if the network is unstructured instead of having special topological properties. Iloglu and Albert (2018) model the INDS problem as a p-median problem with the objective of minimizing the weighted distance between the emergency responders and the demand points. The authors also include the decision on the location of the emergency responders and the assignment of these responders to the demand points for each time period in the planning horizon. The authors provide lower and upper bounds on the optimal objective value by making use of the Lagrangian relaxation techniques.

Aslan and Çelik (2019) focus on the effect of performing the pre-positioning activities together with the road network repair and relief transportation activities under network vulnerability and relief item demand uncertainties. The authors formulate the problem as a two-stage stochastic program which decides on the relief item pre-positioning locations and amounts in the first stage, and the relief transportation and road network repair in the second stage. The objective of the model is to minimize the weighted arrival times to the demand nodes for all relief items. To solve the model, Aslan and Çelik (2019) propose using a sample average approximation scheme for a near-optimal solution and valid inequalities to enhance the performance. The authors also put forward different objective functions for the problem and compare them in the basis of efficacy, equity and robustness aspects. They suggest the usage of the equity-based objective to balance the trade-offs in the problem structure. Sanci and Daskin (2019) also propose to use a two-stage stochastic program aiming to minimize the total cost for a rather similar problem structure. The proposed model decides on the location and the capacity of the emergency response facilities, and location and number of the restoration equipment in the first stage. In the second stage, relief item distribution and restoration decisions are determined. In their problem environment, uncertainty is in the demand and supply amounts as well as in the network availability. Sanci and Daskin (2019) use a sample average approximation technique to decrease the number of scenarios considered in the model. The authors also make use of concentration sets to limit the possible values of the first-stage decision variables.

Fang and Sansavini (2018) work on the critical infrastructure restoration planning problem under uncertain resources and repair times. The authors propose a two-stage stochastic program with maximum system resilience objective. Their model includes the flexibility of multiple modes, i.e., repair times are inversely proportional to the number of teams working on the repair job. Fang and Sansavini (2018) also propose a Benders decomposition technique to tackle the complexity of the stochastic program. The authors conclude that the value of the stochastic solution is higher if the variance of the inherent uncertainty is high.

2.3 Interdependent Infrastructure Restoration Problems

Planning of interdependent infrastructure restoration after a disaster is a rather recent research topic for the literature. The term *interdependent* is used instead of dependent, because infrastructures may be mutually dependent on one another. More specifically, one node of infrastructure A may be dependent on one node of infrastructure B, while another node of infrastructure B is dependent on another node of infrastructure A.

One of the first studies on this topic is conducted by Lee et al. (2007), who define five categories for infrastructure interdependencies, namely input (operational), mutual, shared, exclusive and collocated dependencies. Of these, an input interdependency, which is the main type of interdependency considered in this thesis, is defined as a node not being able to function until its dependency from the other infrastructure is satisfied. An input interdependency creates a precedence relationship in the operations of the infrastructures. Lee et al. (2007) provide an interdependent layered network representation (ILN) for the interdependent infrastructures. In this representation, every infrastructure is represented as a network. For example, for the electricity infrastructure, supply nodes are the power plants, while demand nodes are the household or buildings of other critical infrastructures. Transshipment nodes correspond to substations and arcs are the power lines. Their model is used to determine which arcs to repair in order to restore the services of the infrastructures completely without any repair scheduling efforts. In Lee et al. (2009), the authors provide an application tool for the ILN and explain its capabilities. Lee and Rao (2009) assert the importance of information sharing in the infrastructure management which is generally performed independently from each other.

González et al. (2016) present the interdependent network design problem (INDP), which allows multiple commodities to flow on the infrastructures. Their interpretation does not fix time periods and is convenient to apply decomposition techniques. The authors propose a heuristic that includes solving the INDP iteratively and using a simulation-optimization framework to analyze the expected system behavior. Coffrin et al. (2012) consider the last-mile restoration of interdependent power and gas infrastructures and provide a three-stage decomposition algorithm to determine the order of the repairs. In Cavdaroglu et al. (2013), the authors define the Interdependent Integrated Network Design and Scheduling (IINDS) problem, which considers the scheduling decisions for the repairs as well as which arcs to repair. They propose a MIP model for the IINDS problem which includes operational (input) interdependencies. The objective of the model is to minimize the unmet demand of the demand nodes and the restoration and operation costs. They also provide a heuristic method based on Lee et al. (2007) for large problem instances, since the solution time of the MIP model is inadequate to make decisions quickly. By giving priority to the interdependent nodes, the flow on the damaged network is determined first. Second, the improvement of adding an arc or path to the network is calculated and used to determine which arcs to repair. These arcs are placed into a queue for repair teams to work on them.

Restoration interdependencies between infrastructures are first introduced by Sharkey et al. (2015a) as a result of their study on the restoration efforts of Hurricane Sandy in 2012. An example of restoration interdependency is to restore the electricity of a critical region before flood can be re-channeled to sewers with the help of pumps. These interdependencies are divided to four categories, namely traditional precedence, effectiveness precedence, options precedence and time sensitive options. Sharkey et al. (2015b) extend the mathematical model presented in Cavdaroglu et al. (2013) by incorporating restoration interdependency constraints. Restoration interdependencies are formulated for each category defined in Sharkey et al. (2015a), where the aim of the model is to maximize the amount of infrastructure services to their disaster-free levels. The authors use the developed model to evaluate different decision-making environments for the problem, namely centralized, decentralized and information sharing decision making environments. Two realistic data sets (based on New Hanover and CLARC Counties) are used for the computational tests. With the help of the computational tests, the authors show the significance of centralized decision-making and the value of information sharing.

Loggins et al. (2019) define the interdependencies between the critical civil infrastructures and social infrastructures such as healthcare, police, fire-fighting and education services. The authors devise a decision making tool to optimize the repair activities after a disaster. The main objective of the repair activities is to restore the network elements that are needed in the services of the social infrastructures. The decision making model enables the user to generate different damage scenarios and observe the difference in the restoration activities.

Shen (2013) introduces random arc disruptions to interdependent infrastructure networks and proposes a two-stage stochastic program. The first-stage problem decides on the arcs to be built, while the second-stage determines the flow decisions. The objective is to minimize the total cost of construction, flow generation, penalization cost of unmet demand and unsatisfied interdependencies. Cutting-plane algorithms and heuristics are proposed for the problem and tested on different network topologies. Ouyang (2017) works on a similar problem structure in a defender-attacker environment with an objective of maximizing the satisfied demand. The defender chooses the network components to defend prior to the attack, and the attacker decides on the center and the radius of the attack. The defender decides on the flow values and interdependency satisfaction after the attack without considering the repair efforts. Ouyang (2017) models this problem as a three-level defender-attacker-defender model and provides an exact solution algorithm based on decomposition. Building on this work, Fang and Zio (2019) incorporate the information on various disasters including the temporal damage probabilities on the network elements into the defender-attacker-defender model. Their aim is to find the best reinforcement investment before the disaster, and flow and interdependency satisfaction values after the disaster maximizing the performance of the infrastructures after the disaster, which is measured as the ratio of the real performance to the target performance of an infrastructure. The authors model the problem as a two-stage adaptive robust optimization-based mathematical program and devise a nested cutting plane decomposition algorithm for its solution.

2.4 Genetic Algorithms on Infrastructure Restoration

Genetic algorithms (GAs) were proposed by Holland (as cited in Beasley et al. 1993), inspired by the genetic processes of the living organisms. As in nature, genes (features of the solution) are transferred from parents (parent solutions) to children (offspring solutions) after a genetic crossover (combination of solutions) is performed. Through generations (iterations of the algorithm), good features of the individuals with high fitness (solutions that have a desirable objective function value) are combined and transferred to find the fittest individual (the best solution). From time to time, there happens to be a mutation on the genes (random changes in the solutions) to search for more fit individuals (solutions). While the parent selection in a GA helps to exploit the promising areas of the solution space, mutation and crossover operations help to explore new areas of it. Thanks to these features, GAs are able to find near-optimal or optimal solutions for many optimization problems.

Although usage of the GA on the infrastructure restoration problems is not very common, there are a few studies in the literature using this method. Sato and Ichii (1996) make use of the GA to determine the order of restoration of a highway network after an earthquake. Xu et al. (2007) model the restoration of electric infrastructure after an earthquake as a stochastic integer program and present a GA to solve it. The proposed GA tries to find the optimal order of the restorations and makes use of simulation to calculate the performance of each individual. Their aim is to minimize the time each customer is without electricity.

Dudenhoeffer et al. (2006) present a modelling and simulation framework to represent interdependent infrastructure networks and to analyze the interdependency structures among the infrastructures. Users can manually damage individual or multiple nodes and arcs to observe the effect of such a change on the functioning of the infrastructures. They propose to integrate a GA to the framework in order to determine which network elements are the most valuable ones to protect or repair in a disaster scenario. Later, Permann (2007) actualizes this proposition and integrates the GA into the simulation.

Ouyang and Wang (2015) propose to use the GA in assessing the resilience of the interdependent infrastructures. They aim to determine the optimal restoration order of the damaged parts of the infrastructures with the help of the GA. The authors provide and compare five different restoration strategies. The GA is tested with these different strategies using the electricity and gas infrastructures of Houston, USA. The authors conclude that random repair of the parts of the interdependent infrastructures offers the least resilience while the strategy considering the interdependencies offers the most.

2.5 Contributions of This Study

To the best of our knowledge, this study is the first one simultaneously considering the operational interdependencies among interdependent infrastructures, integration of the reinforcement and the repair activities with the aim of maximizing the performance of the infrastructures, incorporating the inherent uncertainty caused by the disaster on the functionality of the network elements, and scheduling of the repair activities for multiple repair teams. The studies presented in this chapter may combine only a subset of these aspects. We define the Stochastic Interdependent Infrastructure Reinforcement and Repair (SIIRR) problem and present a two-stage stochastic programming model formulation. Since the 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 insights and provide sensitivity analysis for the decision makers of the critical infrastructures.

CHAPTER 3

PROBLEM DEFINITION AND MATHEMATICAL MODEL

In this study, we focus on the pre- and post-disaster operations of a centralized decision maker that aims to coordinate the reinforcement and repair operations of multiple interdependent infrastructure networks. Within the context of Turkey, this coordinating entity may be the municipality for smaller-scale events, or the Disaster and Emergency Management Presidency (AFAD) for larger-scale disasters.

The individual networks may be operated by local municipalities (e.g., road, water or wastewater networks) or by private companies (e.g., power or telecommunications networks). Due to the fragmented structure of how these networks are managed, their reinforcement is funded by separate budgets allocated by the entity managing each network. Likewise, due to the need for specialized expertise, post-disaster repair of these networks is conducted by separate teams for each infrastructure.

An important aspect that needs to be taken into account when planning for the reinforcement and repair of critical infrastructure networks is the interdependency between the operations of these networks. Lee et al. (2007) define five categories for infrastructure interdependencies namely input (operational), mutual, shared, exclusive and collocated dependencies. For the sake of simplicity, our models and solution approaches include only the input interdependency, which covers cases where a node is not being able to function until its dependency from the other infrastructure is satisfied. This creates a precedence relationship in the operations of the infrastructures. Although our models incorporate only input interdependencies, they can be extended to involve other interdependency types as well.

The pre-disaster decisions of network reinforcement involve the uncertainties related to the disaster, which deem an exact estimation of the network conditions impossible. However, estimates on potential effects of various disaster scenarios on the infrastructure can be obtained by means of structural engineering models (e.g., Faturechi and Miller-Hooks, 2015). A coordinated planning of these reinforcement efforts by considering all networks in an integrated manner is likely to provide substantial advantages to maintaining or restoring the post-disaster service flows in a timely manner. Similarly, once the disaster hits, a coordinated effort towards planning the repair activities ensures that the interdependencies are taken into account and restoration is carried out more efficiently than the case where repairs in each network are made independently of the other networks.

Since scheduling of the repair teams is involved in the post-disaster response stage, multiple periods are involved in the decision making process. Upon determining the schedules of its teams, each infrastructure directs these teams to the assigned network components (e.g., power lines, water/wastewater pipes, fiber-optic cables) at the scheduled time periods, which may represent hours or multiple-hour shifts. 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.

Motivated by the need to coordinate the reinforcement and repair operations, as well as the operations of multiple infrastructure networks, we define the Stochastic Independent Infrastructure Reinforcement and Repair (SIIRR) problem in this chapter. The problem aims to restore the services of the infrastructures as early as possible by deciding on the reinforcement and restoration activities considering the interdependencies between the infrastructures and different potential outcomes of the disaster.

In this study, infrastructures are modeled as directed layered networks as in Lee et al. (2007), where the nodes correspond to the service supply, demand, or transshipment points in each infrastructure. An example of a layered network can be seen in Figure 3.1, in which the layers represent different infrastructures. Three infrastructures in the figure share the same nodes. In general, the same nodes may exist in more than one infrastructure in different types. For example, a supply node in a water infrastructure

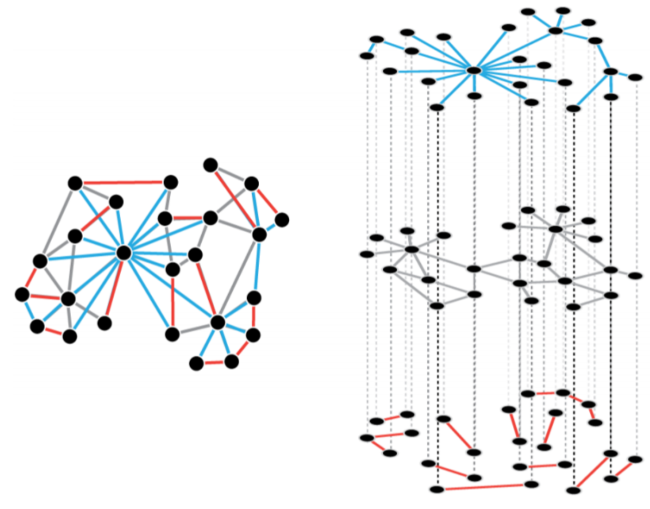

Figure 3.1: Layered network example (Krzywinski, 2010), where each color represents a different infrastructure

may act as a demand node in the power infrastructure, since the water pump in that node may require power to be operated. Flow on the arcs are the services of each infrastructure. For instance, arcs may be the pipelines for the water infrastructure. In Figure 3.1, the infrastructures have different arcs from one another. Interdependencies represent 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.

A two-stage stochastic mixed integer program is proposed for the SIIRR problem. In the first stage, reinforcement decisions are taken under budget limitations and by considering the expected second-stage objective value. 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. This also determines the amounts of demand and interdependencies satisfied in that period. The objective of the SIIRR problem is to maximize the expected accumulated service provided over all periods.

The proposed model and solution approaches are used to obtain managerial insights by investigating the following research questions for the considered problem environment:

- What is the value of reinforcement activities and their effect on the restoration activities?
- What is the benefit of planning reinforcement activities centrally?
- What is the price of ignoring the interdependencies when planning the restoration activities?
- What is the benefit of holding extra service capacities at the interdependent nodes?
- What is the effect of increasing the available reinforcement budget?
- What is the value of the stochastic solution and the expected value of perfect information?

The remainder of this chapter is organized as follows. In §3.1, assumptions of the problem are listed. In §3.2, mathematical model of SIIRR is shared.

3.1 Assumptions

To model the SIIRR problem, we make use of the following assumptions:

- Without loss of generality, it is assumed that the damage is on the arcs of the infrastructure. To model node damage, the damaged node can be converted into two undamaged nodes with two damaged arcs in between.
- When an arc is damaged, its capacity drops to zero. That is, damaged arcs cannot be traversed by the services of an infrastructure.
- Once an arc is reinforced before the disaster, it is assumed not to be damaged during the disaster. Although this is a strong assumption, it prevents the intractability of the model that arises from the decision-dependent uncertainties.

- There is perfect visibility of the network after the disaster. In other words, which arcs are damaged are known certainly after the disaster for every scenario. In practice, satellite images or unmanned air vehicles may be used to assess the damage to the infrastructures, or repair teams may have to traverse or check the arcs physically to observe the damage. We assume that we have the results of this first assessment at hand.

- Repair teams are specific to each infrastructure. Repair teams of a given infrastructure are capable of repairing every arc of the infrastructure.

- It is assumed that the repair times of the arcs are known in advance. The repair time of each arc in case of a possible damage is estimated before the disaster.

- Travel times on the network are negligible compared to the repair times. Therefore, it is assumed that repair teams can move from one repair to the other without losing any time.

- Service flows on the infrastructures take place in negligible time. This is the case in reality for many infrastructures such as electricity and water, since the services are ready to use in every household.

3.2 Mathematical Model

The two-stage stochastic programming model for the Stochastic Interdependent Infrastructure Reinforcement and Repair (SIIRR) Problem will be explained in this section. For this end, §3.2.1 gives the notation used throughout the model and solution approach, and §3.2.2 presents the model formulation.

3.2.1 Notation

The notation used throughout the model and heuristic is as follows:

	Sets
M	Infrastructures
N_m	Nodes in infrastructure $m \in M$

S_m Supply nodes in infrastructure $m \in M$

D_m Demand nodes in infrastructure $m \in M$

T_m Transshipment nodes in infrastructure $m \in M$

K_m Repair crews in infrastructure $m \in M$

A_m Arcs in infrastructure $m \in M$

S Scenarios

E_{ms} Undamaged arcs in infrastructure $m \in M$ in scenario $s \in S$

$\bar{E}_{ms}$ Damaged arcs in infrastructure $m \in M$ in scenario $s \in S$

T Time periods in the planning horizon

F_{mn} 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

Parameters

B_m Available reinforcement budget of infrastructure $m \in M$

s_{im} Amount of supply available at node $i \in S_m$ in infrastructure $m \in M$ per period

d_{im} Amount of demand at node $i \in D_m$ in infrastructure $m \in M$ per period

cn_{im} Capacity of transshipment node $i \in T_m$ in infrastructure $m \in M$ per period

ca_{ijm} Capacity of arc (i, j) in infrastructure $m \in M$ per period

c_{ijm} Cost of reinforcement for arc (i, j) in infrastructure $m \in M$

p_{ijm} Repair time of arc (i, j) in infrastructure $m \in M$

fDF_m Total demand that can be met per period in infrastructure $m \in M$ before the disaster

w_{im} Weight of demand node $i \in D_m$ in infrastructure $m \in M$

ω_t Weight of time period $t \in T$

$\mathcal{M}$ A large positive value

fNR_{ms} Total demand that can be met per period in infrastructure $m \in M$ in scenario $s \in S$ after the disaster without any repair or reinforcement

P_s Realization probability of scenario $s \in S$

Decision variables

$$r_{ijm} = \begin{cases} 1, & \text{if arc } (i,j) \in A_m,\ m \in M \text{ is reinforced before the disaster} \\ 0, & \text{otherwise} \end{cases}$$

x_{ijtms} Amount of flow on arc $(i,j) \in A_m$, $m \in M$ at time $t \in T$ in scenario $s \in S$

v_{itms} Amount of demand met in demand node $i \in D_m$ in infrastructure $m \in M$ at time $t \in T$ in scenario $s \in S$

$$y_{ijmnts} = \begin{cases} 1, & \text{if input dependency of node } j \in N_n \text{ on node } i \in D_m \text{ is satisfied} \\ & \text{at time } t \in T \text{ in scenario } s \in S \\ 0, & \text{otherwise} \end{cases}$$

$$\alpha_{ijtkms} = \begin{cases} 1, & \text{if repair of arc } (i,j) \in A_m,\ m \in M \text{ is finished at time } t \in T \text{ by} \\ & \text{repair crew } k \in K_m \text{ in scenario } s \in S \\ 0, & \text{otherwise} \end{cases}$$

$$\beta_{ijtms} = \begin{cases} 1, & \text{if arc } (i,j) \in A_m,\ m \in M \text{ is operational at time } t \in T \text{ in} \\ & \text{scenario } s \in S \\ 0, & \text{otherwise} \end{cases}$$

3.2.2 Formulation

Using the sets, parameters, and decision variables in the preceding section, the two-stage stochastic programming model SIRR-SP for the SIIRR problem is presented below.

(SIIRR-SP)

$$\text{Maximize} \sum_{s \in S} P_s \sum_{t \in T} \omega_t \sum_{m \in M} \frac{\sum_{i \in D_m} w_{im} v_{itms} - fNR_{ms}}{fDF_m - fNR_{ms}} \tag{3.1}$$

subject to

$$\sum_{(i,j) \in A_m} c_{ijm}\, r_{ijm} \leq B_m \qquad \forall m \in M \tag{3.2}$$

$$\sum_{j:(i,j)\in A_m} x_{ijtms} - \sum_{j:(j,i)\in A_m} x_{jitms} \leq s_{im} \qquad \forall m \in M,\ i \in S_m,\ t \in T,\ s \in S \quad (3.3)$$

$$\sum_{j:(i,j)\in A_m} x_{ijtms} - \sum_{j:(j,i)\in A_m} x_{jitms} = 0 \qquad \forall m \in M,\ i \in T_m,\ t \in T,\ s \in S \quad (3.4)$$

$$\sum_{j:(i,j)\in A_m} x_{ijtms} - \sum_{j:(j,i)\in A_m} x_{jitms} = -v_{itms} \qquad \forall m \in M,\ i \in D_m,\ t \in T,\ s \in S \quad (3.5)$$

$$v_{itms} \leq d_{im} \qquad \forall m \in M,\ i \in D_m,\ t \in T,\ s \in S \quad (3.6)$$

$$\sum_{j:(j,i)\in A_m} x_{jitms} \leq cn_{im} \qquad \forall m \in M,\ i \in T_m,\ t \in T,\ s \in S \quad (3.7)$$

$$x_{ijtms} \leq ca_{ijm} \qquad \forall m \in M,\ t \in T,\ s \in S,\ (i,j) \in E_{ms} \quad (3.8)$$

$$x_{ijtms} \leq ca_{ijm}\, \beta_{ijtms} \qquad \forall m \in M,\ t \in T,\ s \in S,\ (i,j) \in \bar{E}_{ms} \quad (3.9)$$

$$d_{im}\, y_{ijmnts} \leq v_{itms} \qquad \forall m,n \in M,\ (i,j) \in F_{mn},\ t \in T,\ s \in S \quad (3.10)$$

$$y_{ijmn(t-1)s} \leq y_{ijmnts} \qquad \forall m,n \in M,\ (i,j) \in F_{mn},\ t \in T - \{1\},\ s \in S \quad (3.11)$$

$$\sum_{h:(j,h)\in A_n} x_{jhtns} \leq s_{jn}\, y_{ijmnts} \qquad \forall m,n \in M,\ j \in S_n,\ (i,j) \in F_{mn},\ t \in T,\ s \in S \quad (3.12)$$

$$\sum_{h:(h,j)\in A_n} x_{hjtns} \leq d_{jn}\, y_{ijmnts} \qquad \forall m,n \in M,\ j \in D_n,\ (i,j) \in F_{mn},\ t \in T,\ s \in S \quad (3.13)$$

$$\sum_{h:(h,j)\in A_n} x_{hjtns} \leq cn_{jn}\, y_{ijmnts} \qquad \forall m,n \in M,\ j \in T_n,\ (i,j) \in F_{mn},\ t \in T,\ s \in S \quad (3.14)$$

$$\beta_{ijtms} - \beta_{ij(t-1)ms} = \sum_{k\in K_m} \alpha_{ijtkms} \qquad \forall m \in M,\ t \in T - \{1\}, \forall s \in S,\ (i,j) \in \bar{E}_{ms} \quad (3.15)$$

$$\beta_{ij1ms} \leq r_{ijm} \qquad \forall m \in M,\ s \in S,\ (i,j) \in \bar{E}_{ms} \quad (3.16)$$

$$\sum_{l=max\{1,t-p_{ijm}+1\}}^{t} \sum_{(h,b)\in \bar{E}_{ms}} \alpha_{hblkms} \leq \alpha_{ijtkms} + \mathcal{M}(1-\alpha_{ijtkms})$$

$$\forall m \in M,\ t \in T,\ k \in K_m,\ s \in S,\ (i,j) \in \bar{E}_{ms} \quad (3.17)$$

$$\alpha_{ijtkms} \leq 0 \qquad \forall m \in M,\ k \in K_m,\ s \in S,\ (i,j) \in \bar{E}_{ms},\ t \in T:\ t \leq p_{ijm} \quad (3.18)$$

$$r_{ijm} \in \{0,1\} \qquad \forall m \in M,\ (i,j) \in A_m \quad (3.19)$$

$$y_{ijmnts} \in \{0,1\} \qquad \forall m,n \in M,\ (i,j) \in F_{mn},\ t \in T,\ s \in S \quad (3.20)$$

$$\alpha_{ijtkms} \in \{0,1\} \qquad \forall m \in M,\ t \in T,\ k \in K_m,\ s \in S,\ (i,j) \in \bar{E}_{ms} \quad (3.21)$$

$$\beta_{ijtms} \in \{0,1\} \qquad \forall m \in M,\ t \in T,\ s \in S,\ (i,j) \in \bar{E}_{ms} \quad (3.22)$$

$$x_{ijtms} \geq 0 \qquad \forall m \in M,\ (i,j) \in A_m,\ t \in T,\ s \in S \quad (3.23)$$

$$v_{itms} \geq 0 \qquad \forall m \in M,\ i \in D_m,\ t \in T,\ s \in S \quad (3.24)$$

The model SIIRR-SP is the extensive form of the two-stage stochastic mixed integer program with recourse aiming to maximize the performance of all infrastructures over the planning horizon for all scenarios using objective function (3.1). The objective function is measured by 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. Here, fNR_{ms} represents the amount of service that can be supplied right after the disaster without any reinforcement and repair efforts in scenario s, and fDF_m denotes the total service level before the disaster. This ratio is summed up over all infrastructures and the planning horizon with time period weights 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. Every scenario $s \in S$ is represented in (3.1) with its probability P_s. Note that objective function does not include any first-stage components. Constraints (3.2) and (3.19) are first-stage constraints, whereas the remaining constraints are second-stage constraints.

Constraints (3.2) limit the reinforcement efforts with the given budget for all infrastructures. Constraints (3.3), (3.4) and (3.5) are flow balance equations for supply, transshipment and demand nodes, respectively. Constraints (3.6) limit the demand satisfaction of a node with the demand value. Constraints (3.7) enforce node capacity of a transshipment node. Constraints (3.8) and (3.9) force arc capacities on the flow variables for undamaged arcs and repaired arcs, respectively. Constraints (3.10) state that an input interdependency between nodes i and j is satisfied only if demand of node i is satisfied completely. Constraints (3.11) enforce the interdependency to be satisfied until the end of the horizon once it is satisfied at t. Constraints (3.12), (3.13) and (3.14) allow flow from node j only if its input interdependency is satisfied and limit the flow amount by supply, demand and node capacity, respectively, if interdependency is satisfied. Constraints (3.15) determine the operational availability of an arc considering repair actions of all restoration teams. Constraints (3.16) allow arc (i, j) to be available starting from time period 1 if it is reinforced. Constraints (3.17) make sure that only 1 arc is repaired by a restoration team at a time and prevents repair of any other arc during the repair. Constraints (3.18) prevent the repair of an arc to be completed before the minimum possible completion time, which is the repair duration. Constraints (3.19)-(3.22) are set constraints while (3.23) and (3.24) are nonnegativity constraints.

With two possibilities of post-disaster condition on the arcs of the infrastructures, there can be a total of $2^{\sum_{m \in M} |A_m|}$ possible scenarios in the SIIRR problem. Even though some of these scenarios have low realization probability, having exponentially many potential scenarios leads to the curse of dimensionality, which deems the problem to be intractable. As it is shown by Nurre and Sharkey (2014), even the deterministic version of the problem without any reinforcement activities is NP-hard. In all, SIIRR-SP is impossible to solve to optimality for reasonably-sized networks using commercial optimization solvers, even within multiple days. The use of Benders decomposition algorithm for an exact solution in our initial trials could not improve the solution time even for small-sized instances. Consequently, we propose a heuristic procedure in the next section that aims to find high-quality solutions within acceptable time limits.

CHAPTER 4

SOLUTION METHODOLOGY

The SIIRR problem cannot be solved within reasonable computational times, even for small-sized instances. To overcome this, a heuristic method that makes use of a genetic algorithm (GA) and partial optimization has been developed. GA is preferred over other heuristic methods as it provides a convenient framework to work with binary decision variables and crossover operation does not generate substantial infeasibilities that need to be repaired.

In the devised 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 procedure. To reduce the computational burden on the GA, the scenario set S is reduced to one 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), developed to determine the second-stage decisions with this representative scenario. At the end of the GA, the reinforcement decisions with the highest fitness value are fed to FRESH to determine the second-stage decision variables for all possible scenarios. The expected objective value is also calculated over the whole scenario set S, using these heuristically-determined second-stage decisions.

This chapter has been organized as follows. In §4.1, general steps of the genetic algorithm will be presented. In §4.2, the steps of FRESH will be given in detail.

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}. The pseudocode of the overall structure of the proposed GA is provided in Algorithm 1 and a flowchart of the GA can be seen in Figure 4.1. An individual in the algorithm corresponds to an $\overrightarrow{r}$ vector, where each gene represents the r_{ijm} values, resulting in $|A_m| \times |M|$ genes in an individual. Since r_{ijm} are binary variables; binary coding is used as the representation scheme. As there are constraints in the SIIRR problem, infeasible solutions should be penalized or repaired until the last generation. In the proposed approach, the initial generation consists only of feasible solutions, but infeasible solutions are allowed throughout the iterations to increase exploration of the solution space. Infeasible solutions are penalized, and penalization increases towards the later iterations of the algorithm. The feasible r_{ijm} values that result in the best estimated fitness value are returned by the GA. In the remainder of this section, the steps of the genetic algorithm are explained in detail.

4.1.1 Representative Scenario Generation

This procedure is performed to ensure that the GA is able to determine the first-stage decisions in reasonable time. The representative scenario $\bar{E}_m$ is generated from the set of damaged arcs $\bar{E}_{ms}$ for all $m \in M$, assuming that 90% of the arcs will be damaged in the second stage, in order to represent the worst possible second-stage scenario. By doing so, GA is urged to schedule as many repairs as it can in the first stage to help the recovery of even the worst possible second-stage scenario. The representative scenario generation procedure is shown in Algorithm 2. First, for each infrastructure, arcs are ranked in descending order of the total number of scenarios in which they are damaged. The first 90% of all arcs for each infrastructure are assumed as damaged in the representative scenario.

Algorithm 1 Genetic Algorithm for the SIIRR Problem

Input: All parameters of the SIIRR problem with the representative scenario ($\bar{E}_m$ and E_m) and seed solutions set, SS
Output: The r_{ijm} values that yield the highest fitness value
Generate an initial population of size $PopulationSize$
    Insert $PopulationSize \times p_s\%$ individuals from SS
    Generate $PopulationSize \times (1 - p_s)\%$ individuals using the Random Initial Solution Generation procedure
Compute fitness of each individual using FRESH
repeat
    repeat
        Select two parents, p_1 and p_2
        Recombine the parents with probability p_c to produce two offspring o_1 and o_2 using the Uniform Crossover Operator
        Apply mutation with probability p_m to each offspring using the Mutation Operator and obtain mutated offspring mo_1 and mo_2
        Compute penalized fitness values of mo_1 and mo_2 using FRESH
        Form the population for the next generation
    until $PopulationSize/2$ steps are complete
until $Iterations$ steps are complete

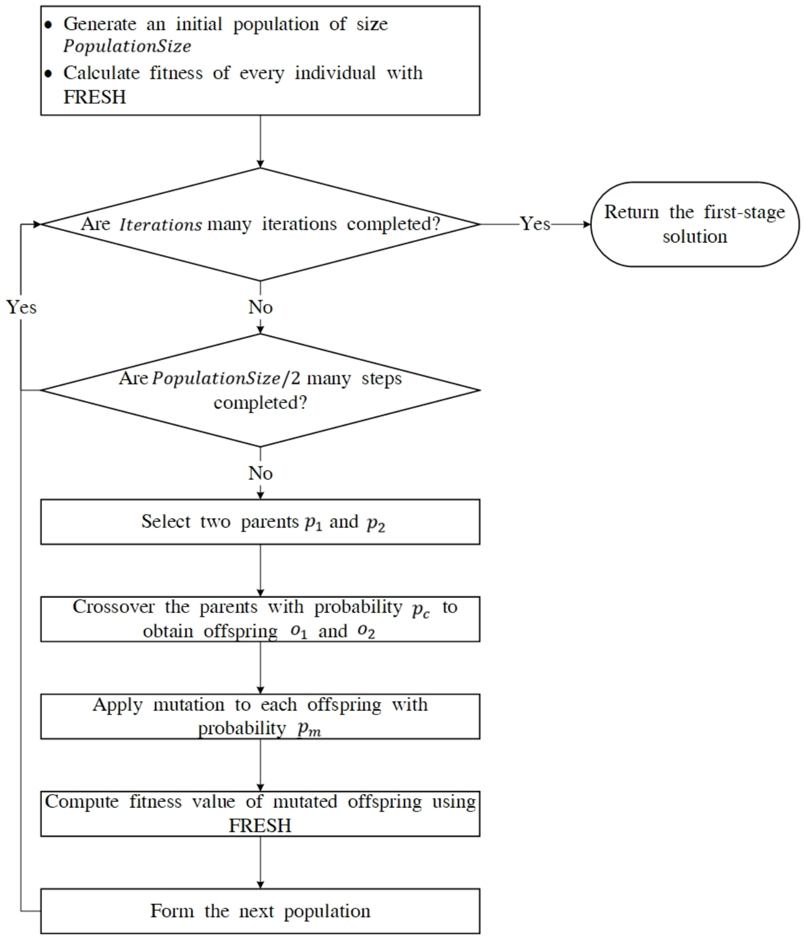

Figure 4.1: Flowchart of the genetic algorithm

Algorithm 2 Representative Scenario Generation

Input: $\bar{E}_{ms}\ \forall m \in M$ and $\forall s \in S,\ A_m\ \forall m \in M$

Output: $\bar{E}_m$ and E_m

for all $m \in M$ **do**

  Rank arcs $(i,j) \in A_m$ in descending order of $R_{ijm} := \sum_{s \in S} I_{(i,j) \in \bar{E}_{ms}}$, where I is a binary indicator variable

  Start with $\bar{E}_m = \emptyset$

  while $|\bar{E}_m| < 0.90|A_m|$ **do**

    $(i,j) = argmax_{(u,v) \notin \bar{E}_m}\{R_{uvm}\}$

    $\bar{E}_m \leftarrow \bar{E}_m \cup \{(i,j)\}$

  end while

  $E_m = A_m \setminus \bar{E}_m$

end for

4.1.2 Initial Population Generation

Population size of the algorithm, $PopulationSize$, is considered as a design parameter and is determined via an experimental design setting. The initial population of the proposed GA consists of a set of random solutions and promising seed solutions to aim for more rapid convergence. A seeding percentage, denoted by p_s, is also a design parameter, and is used to determine the number of solutions that will be seeded to the initial population. Seed solutions set is computed beforehand using the information gathered from second-stage solutions with no reinforcement in the first stage.

The seeding procedure is explained in Algorithm 3. One seed is generated from each $s \in MS$ where MS is the master set of scenarios which includes all possible second-stage scenarios. Using this procedure, $|MS|$ many feasible seeds are generated prior to the GA. Seeds are constructed so that the arcs which have flow in the largest number of periods in the second stage are reinforced in the first stage, that is, before the disaster. As many seeds as needed are inserted into the random initial population. These may be selected randomly or in decreasing expected second-stage performance from the seed solutions set. This strategy is also a design parameter. The remaining $(1 - p_s)\%$ of the initial solutions are generated randomly using Algorithm 4.

4.1.3 Fitness Value Calculation

In cases where the objective function value of a given solution is computationally expensive or difficult to calculate, an approximation is used. In the proposed approach, the expected objective function value of the given first-stage decision variables r_{ijm} is approximated using a representative scenario in the second stage. Fitness values of each individual and offspring are calculated using FRESH with this representative scenario. Fitness value, F, returned by FRESH is calculated using equation (4.1) for the representative scenario s.

$$F = \sum_{t \in T} \omega_t \sum_{m \in M} \frac{\sum_{i \in D^m} w_i v_{itms} - fNR_{ms}}{fDF_m - fNR_{ms}}. \tag{4.1}$$

Although r_{ijm} is not a part of this formula, it affects the v_{itms} values in the second

Algorithm 3 Seeding Procedure

Input: All parameters of the SIIRR problem instance, master scenario set MS

Output: Seed solutions set SS

for all $s \in MS$ **do**

  Call FRESH for scenario s with $r_{ijm} = 0 \, \forall (i,j) \in A_m, \; m \in M$

  $U_{ijms} := \sum_{t \in T} I_{x_{ijtms} > 0} \, \forall (i,j) \in A_m, \; m \in M$, where I is a binary indicator variable

  for all $m \in M$ **do**

    Rank arcs $(i,j) \in A_m$ in decreasing U_{ijms}; arc (i,j) has rank $Rank_{ij}$

    Start with $(i,j) : Rank_{ij} = 1$

    while $B_m - \sum_{(u,v) \in A_m} \sum_{(u,v) \in A_m} c_{uvm} r_{uvm} > 0$ **do**

      if $B_m - \sum_{(u,v) \in A_m} \sum_{(u,v) \in A_m} c_{uvm} r_{uvm} - c_{ijm} \geq 0$ **then**

        Set $r_{ijm} = 1$

      end if

      $(i,j) \leftarrow (u,v) : Rank_{uv} = Rank_{ij} + 1$

    end while

  end for

  $SS \leftarrow SS \cup \overrightarrow{r}$

end for

for all $ss \in SS$ **do**

  Call FRESH for all $s \in S$ to calculate expected second-stage objective

end for

Order the seeds in decreasing expected second-stage objective value or randomly

Algorithm 4 Random Initial Solution Generation

Input: $A_m, c_{ijm}, B_m, p_s, PopulationSize$
Output: Random initial solutions
repeat
  Set $r_{ijm} = 0 \quad \forall (i,j) \in A_m, \; \forall m \in M$
  for all $m \in M$ **do**
    while $B_m - \sum_{(i,j)\in A_m} \sum_{(i,j)\in A_m} c_{ijm} r_{ijm} > 0$ **do**
      Select an arc $(i,j) \in A_m$ randomly
      if $B_m - \sum_{(u,v)\in A_m} \sum_{(u,v)\in A_m} c_{uvm} r_{uvm} - c_{ijm} \geq 0$ **then**
        Set $r_{ijm} = 1$
      end if
    end while
  end for
  Add vector $\overrightarrow{r}$ to population
until $PopulationSize \times (1 - p_s)\%$ individuals are generated

stage through flow variables x_{ijtms}. If the given first-stage decision variable is infeasible with respect to the budget constraint, the fitness value is penalized using equation (4.2), where F' is the penalized fitness value and c_p is the penalization coefficient.

$$F' = max\left\{0, F - c_p \sum_{m\in M} \frac{\sum_{(i,j)\in A_m} r_{ijm} c_{ijm} - B_m}{B_m}\right\} \tag{4.2}$$

The penalization coefficient increases towards the end of the iterations to make sure that the algorithm explores the search space at the beginning of the iterations but still finds a feasible solution at the end.

4.1.4 Parent Selection

Parent selection is performed probabilistically in the proposed GA. The individuals to reproduce may be selected randomly or with probabilities based on fitness value (*elitist* parent selection). In the first case, diversification is prioritized and the solution space is explored more. In the second case, intensification is strengthened, and the

p_1:	**1**	**1**	**1**	**1**	**...**	**1**	**0**	**0**	**0**
p_2:	0	1	1	0	...	0	1	1	0
$mask$:	0	0	1	0	...	1	1	0	0
o_1:	0	1	**1**	0	...	**1**	**0**	1	0
o_2:	**1**	**1**	1	**1**	...	0	1	**0**	**0**

Figure 4.2: An example uniform crossover operator

promising regions of the solution space are exploited more. These beneficial features of the GA should be balanced considering the other steps of the algorithm which provide intensification or diversification. Thus, parent selection scheme is considered to be a design parameter of the GA and is determined using an experimental design setting. In the elitist parent selection, an individual is selected to reproduce with a probability of $\frac{F'-F_{min}}{F_{max}-F_{min}}$. Note that F_{min} is 0 and F_{max} is $\sum_{t\in T}\omega_t|M|$ since $|M|$ many percentages are summed up in the inner sum of equation (4.1). The penalized fitness value is used in the probability calculation to prevent an infeasible solution from dominating the population with unrealistic fitness values.

4.1.5 Crossover Operator

In every iteration, $PopulationSize/2$ many pairs (parents) are selected from the population and crossover is applied to each pair with probability p_c. Otherwise, parents are returned as offspring. Crossover probability is accepted as a design parameter and its value is determined using an experimental design setting. The selected parents are subject to a uniform crossover, where the genes of the parents are transferred to the offspring using a binary crossover mask, and this mask is generated randomly. An example for uniform crossover is shown in Figure 4.2 and the procedure is summarized in Algorithm 5. With the help of the crossover operation, good genes are transferred to the next generations and diversity in the solution set is achieved.

Algorithm 5 Uniform Crossover Operator

Input: Parents p_1 and p_2, p_c, A_m
Output: Offspring o_1 and o_2
$rand \leftarrow U(0,1)$
if $rand > p_c$ **then**
    $o_1 = p_1,\ \forall (i,j) \in A_m,\ \forall m \in M$
    $o_2 = p_2,\ \forall (i,j) \in A_m,\ \forall m \in M$
else
    Generate a binary crossover mask, $mask$
    for all $m \in M$ **do**
        for all $(i,j) \in A_m$ **do**
            if $mask_{ijm} = 1$ **then**
                $o_{1_{ijm}} = p_{1_{ijm}}$
                $o_{2_{ijm}} = p_{2_{ijm}}$
            else
                $o_{1_{ijm}} = p_{2_{ijm}}$
                $o_{2_{ijm}} = p_{1_{ijm}}$
            end if
        end for
    end for
end if

4.1.6 Mutation Operator

It is common practice to apply mutation to every gene with a small mutation probability. In this study, each gene of an individual is mutated with probability p_m using Algorithm 6. Population diversity is increased with the help of the mutation operation and thus different areas of solution space are explored.

Algorithm 6 Mutation Operator

```
Input: Offspring o_1, p_m, A_m
Output: Mutated offspring mo_1
for all m ∈ M do
    for all (i, j) ∈ A_m do
        rand ← U(0, 1)
        if rand > p_m then
            mo_{1_{ijm}} = o_{1_{ijm}}
        else
            mo_{1_{ijm}} = 1 − o_{1_{ijm}}
        end if
    end for
end for
```

4.1.7 Forming the Next Generation

In this algorithm, the offspring replace the worst two individuals in the population if the former have higher penalized fitness function value than the worst two individuals in the population. The population and offspring are sorted together according to their penalized fitness function value and the best $PopulationSize$ many of them are selected as the new generation. Good solutions are promoted in the population in this manner to increase intensification.

4.2 The Flow and Repair Scheduling Heuristic (FRESH)

The Flow and Repair Scheduling Heuristic (FRESH) is used both for the fitness function calculation in the GA and to determine the second-stage decisions of the two-stage SIIRR problem. In the first case, the heuristic is called with one representative scenario, whereas in the second case, it is applied on all scenarios in the set S, where the second-stage decision variables are calculated for each scenario and the expected objective value is calculated over the scenarios in S.

Flowchart of FRESH is provided in Figure 4.3 and Algorithm 7 gives the steps of FRESH for a scenario $s \in S$. This algorithm first finds a repair schedule (values of α_{ijtkms} and β_{ijtms}) considering the interdependencies between the infrastructures using the Flow and Repair Schedule Construction Heuristic (FRSCH). Since FRSCH schedules a repair only when a flow through a damaged arc is desired, the resulting repair schedule may involve repair teams being idle for a significant amount of time. This repair schedule is then modified using the Repair Schedule Improvement Heuristic (RSIH), where the order of the repairs is preserved and gaps in the repair schedule are eliminated. Following this, the new repair schedule is fed into a flow optimization model to find the optimal flow values on arcs, demand met in each node, and interdependency satisfaction (values of the x_{ijtms}, v_{itms} and y_{ijmnts} decision variables, respectively), given the repair schedule (α_{ijtkms} and β_{ijtms} values). The flow optimization model 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 addition of constraints (4.3), which ensures that y_{ijmnts} is equal to 1 if and only if the interdependency is satisfied together with constraints (3.10).

$$d_{itms} - v_{itms} \geq 1 - y_{ijmnts} \quad \forall m, n \in M,\ (i,j) \in F_{mn},\ t \in T,\ s \in S \tag{4.3}$$

Constraints (4.3) are needed because if an interdependency is satisfied and y_{ijmnts} is not equal to 1, FRESH cannot realize the interdependency satisfaction.

The scheduling steps are repeated until schedule is full or demand is satisfied completely. The resulting schedule may have idle periods at the end of the schedule. As infrastructures are desired to be completely repaired after a disaster for recovery purposes, we allow repairs to continue even if there is no flow on the repaired arcs in

Algorithm 7 Flow and Repair Scheduling Heuristic (FRESH) for scenario s

Input: SIIRR parameters for scenario $s \in S$, r_{ijm} values

Output: Values of the second-stage decision variables $x_{ijtms}, y_{ijmnts}, v_{itms}, \alpha_{ijtkms}, \beta_{ijtms}$

for all $m \in M$ **do**
  for all $(i,j) \in A_m$ **do**
    if $r_{ijm} = 1$ **then**
      $\beta_{ijtms} = 1 \quad \forall t \in T$
    end if
  end for
end for
for all $m \in M$ **do**
  for all $i \in D_m$ **do**
    if $\exists\, j \in N_n : (i,j) \in F_{mn}$ **then**
      $w_i = bigM$
    end if
  end for
end for
$RT_m = |T| \quad \forall m \in M$
$t^{crew}_{km}, t^{infr}_{m}, \alpha_{ijtkms}, x_{ijtms}, v_{itms}, y_{ijmnts} \leftarrow 0 \quad \forall m, n \in M,\ (i,j) \in A_m,\ t \in T,\ k \in K_m$
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, with SIIRR parameters for scenario s, $tabu_{ijm}$, t^{crew}_{km}, t^{infr}_{m}, $\alpha_{ijtkms}, \beta_{ijtms}, x_{ijtms}, v_{itms}, y_{ijmnts}$ values to obtain the α_{ijtkms} and β_{ijtms} values
  if there is no new repair **then**
    break while
  end if
  Call RSIH with $\bar{E}_{ms}$, t^{crew}_{km}, t^{infr}_{m}, α_{ijtkms} and β_{ijtms} to re-organize the repair schedule
  Call the flow optimization model with current α_{ijtkms} and β_{ijtms} to determine the optimal x_{ijtms}, v_{itms}, and y_{ijmnts} values
  $RT_m \leftarrow \sum_{k\in K_m} T - t^{crew}_{km} \quad \forall m \in M$
end while
Fill the idle time periods at the end of the schedule with repairs according to the SPTF rule
Call the flow optimization mathematical model with the final α_{ijtkms} and β_{ijtms} values to determine the optimal x_{ijtms}, v_{itms}, and y_{ijmnts} values

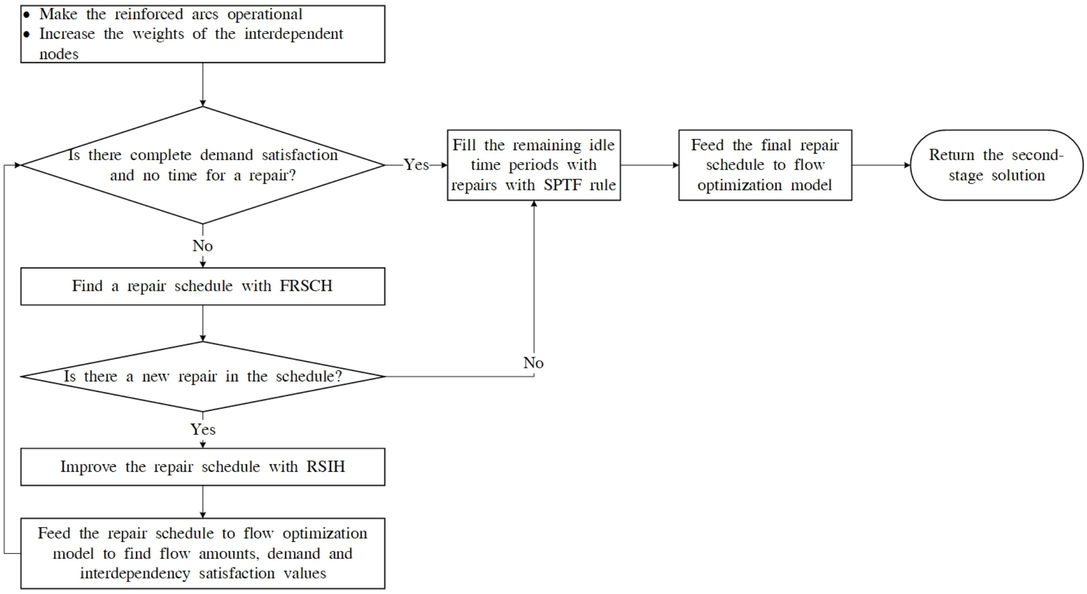

Figure 4.3: Flowchart of FRESH

these periods. For this purpose, the remaining idle time periods are filled with arc repairs according to the shortest processing time first (SPTF) rule. At the end, final repair schedule is fed to the flow optimization mathematical model one last time to obtain the final x_{ijtms}, y_{ijmnts} and v_{itms} values.

The remainder of this chapter is organized as follows. In §4.2.1, FRSCH will be explained in detail. In §4.2.2, RSIH will be presented. An example execution of FRESH can be seen in §4.2.3.

4.2.1 Flow and Repair Schedule Construction Heuristic (FRSCH)

FRSCH, given in Algorithm 8, 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. It makes sense to start with the infrastructure which can satisfy the highest number of interdependencies of the other infrastructures and in this case it is the first infrastructure. An augmenting path is a path from a supply node to a demand node through which an additional flow (conforming the network capacities) can be sent in a network. When an augmenting path is found, a repair is scheduled for every damaged arc 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.

The repair schedule formed by FRSCH 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_{km}^{crew}. 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_m^{infr}. These time points are updated as new repairs are scheduled and they are used throughout the heuristics to keep track of the schedule of the repair crews.

In any one of the iterations of Algorithm 8, one augmenting path with path capacity and repair time is proposed by giving priority to interdependent nodes. This is done by extracting a residual network from the original network of infrastructure m after adding a supply and demand supernode. We call this procedure the Network Extension Subroutine, whose detailed steps can be seen in Algorithm 9.

In the extended network, all supply nodes are linked to the supply supernode and all demand nodes are linked to the demand supernode. The capacity of arcs between the demand supernode and the demand nodes is set as the demand amount of the latter to make sure demand nodes do not receive more flow than their demanded amounts. Similarly, the capacity of arcs between the supply supernode and the supply nodes is set to the supply amount if the supply node is independent from any other infrastructure. Otherwise, the capacity of arcs is set to zero, which prevents the interdependent supply nodes from functioning until their interdependency is satisfied. Once the interdependency is satisfied, their capacity is increased to the supply amount. The repair time of the arcs between the demand supernode and the demand nodes, and the supply supernode and independent supply nodes are set to zero. On the other hand, the repair time of the arcs between the supply supernode and interdependent supply nodes is set to infinity before their interdependency is satisfied, to prevent their use. After the

Algorithm 8 Flow and Repair Schedule Construction Heuristic (FRSCH)

Input: SIIRR parameters for scenario $s \in S$, $tabu_{ijm}$, t^{crew}_{km}, t^{infr}_{m}, α_{ijtkms}, β_{ijtms}, x_{ijtms}, v_{itms}, y_{ijmnts}

Output: Updated $tabu_{ijm}, t^{crew}_{km}, t^{infr}_{m}, \alpha_{ijtkms}$,$\beta_{ijtms}, x_{ijtms}, v_{itms}, y_{ijmnts}$

$ycheck_{it^{infr}_{m}m} := \sum_{n \in M} \sum_{j \in N_n} y_{ijmnt^{infr}_{m}s} \quad \forall m \in M,\, i \in D_m$

Call Network Extension Subroutine

$m \leftarrow 1$

while $\sum_{m \in M} t^{infr}_{m} \neq |T||M|$ **do**

  for $i \in S_m$ **do**

    if $ycheck_{it^{infr}_{m}m} = \sum_{j \in N_n} \sum_{n \in M} I_{(i,j) \in F_{mn}}$ **then**

      $ca_{Sim} \leftarrow s_{im}$ and $p_{Sim} \leftarrow 0$

    end if

  end for

  $rc_{ijm} := \min_{k \in K_m} R_{ijt^{crew}_{km}m} \, \forall (i,j) \in A_m$, 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 $ycheck_{it^{infr}_{m}m} < \sum_{j \in N_n} \sum_{n \in M} I_{(i,j) \in F_{mn}}$

  Call Augmenting Path Subroutine to find P^* with $t = \min_{k \in K_m} t^{crew}_{km}$

  if $\delta_{P^*} > 0$ **then**

    Call Scheduling Subroutine

    if $Flow = 1$ **then**

      Call Flow Subroutine

    end if

  end if

  if $t^{infr}_{m} = |T| \quad \forall m \in M$ **then**

    break while

  end if

  if $\delta_{P^*} = 0 \quad \forall m \in M$ in their last iteration **then**

    if $\sum_{i \in D_m} v_{it^{infr}_{m}ms} = \sum_{i \in D_m} d_{im} \quad \forall m \in M$ **then**

      break while

    else

      if $\exists t : t > t^{infr}_{m'}, ycheck_{itm'} = \sum_{j \in N_n} \sum_{n \in M} I_{(i,j) \in F(m',n)}$ for an $m' \in M$ **then**

        $t^{infr}_{m'} \leftarrow t, t^{crew}_{km'} \leftarrow t \quad \forall k \in K'_m$, and $m = m'$

      else

        $t^{infr}_{m} \leftarrow |T| \quad \forall m \in M$ and $t^{crew}_{km} \leftarrow |T| \quad \forall k \in K_m, \forall m \in M$

      end if

    end if

  else

    $m \leftarrow m + 1(mode\ |M|)$

  end if

end while

Algorithm 9 Network Extension Subroutine

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

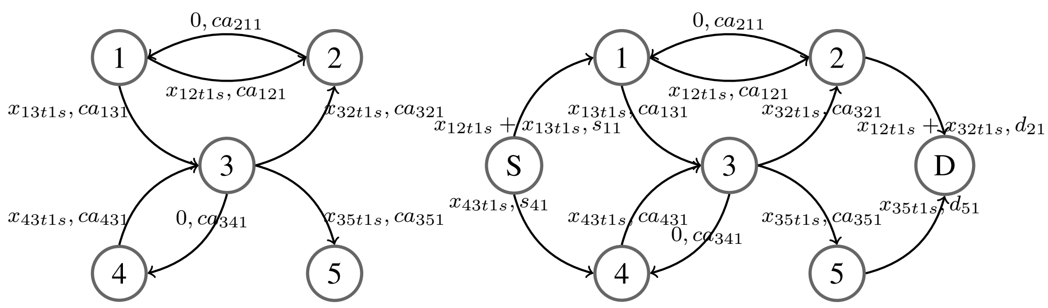

Figure 4.4: A 5-node network and its extended network

interdependency is satisfied, their repair time is set to zero. An example of a 5-node network and its extended network can be seen in Figure 4.4. Here, ca_{ijm} denotes the capacity of arc (i, j) and x_{ijtms} denotes the flow amount on arc (i, j) in infrastructure m at time t in scenario s. Nodes 1 and 4 are supply nodes, while nodes 2 and 5 are demand nodes, and 3 is a transshipment node.

A residual network is extracted from the extended network as described in Schroeder et al. (2004). A residual network consists of residual capacities of arcs and is used to obtain an augmenting path. The residual capacity of the two arcs between nodes i and j in infrastructure m is calculated as in Figure 4.5. Since there can be a flow on only one direction, there are three possible cases. In case 1, there is flow on arc (i, j). In the residual network, this flow should be subtracted from the capacity of arc (i, j), and it should be added to the capacity of arc (j, i) to find the residual capacities. If the augmenting path suggests to increase flow on arc (j, i) by δ, either flow on (i, j) will be decreased by δ, or the flow will change direction. If $\delta > x_{ijtms}$, the flow will change direction and there will be flow on arc (j, i) by an amount of $\delta - x_{ijtms}$. Case 2 is the case where the flow is on the reverse direction, but the calculations are similar to case 1. In case 3, there is no flow between nodes i and j and arc capacities are directly the residual capacities of the arcs. Residual network for the example network can be seen in Figure 4.6 along with an example augmenting path. Only the residual capacities along the augmenting path $S - 1 - 3 - 5 - D$ and on arcs $(1, 2)$ and $(2, 1)$ are denoted for simplicity. The maximum flow that can go through the augmenting path can be calculated by taking the minimum of the residual capacities along the path.

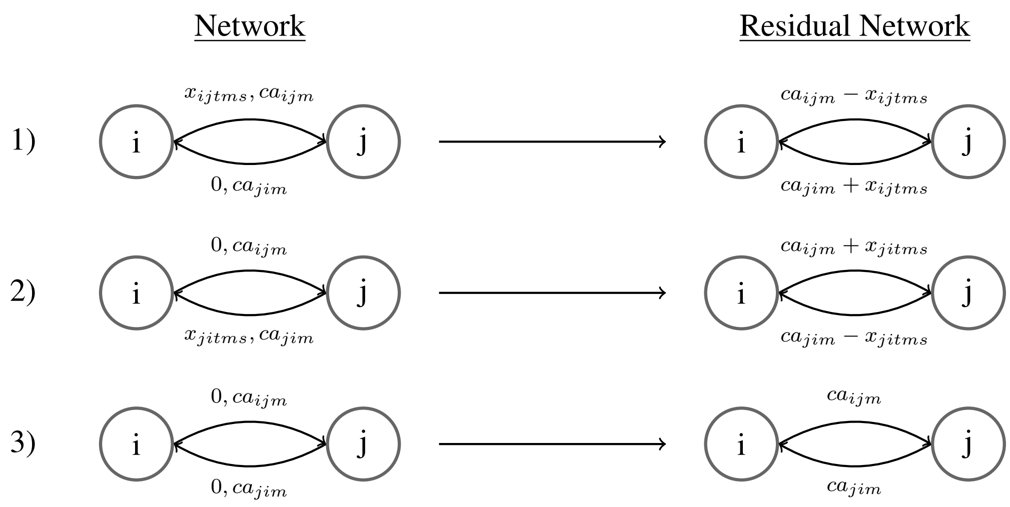

Figure 4.5: Residual capacity calculation for the three possible cases of flow between two nodes, where the arc labels in the residual network denote residual capacities

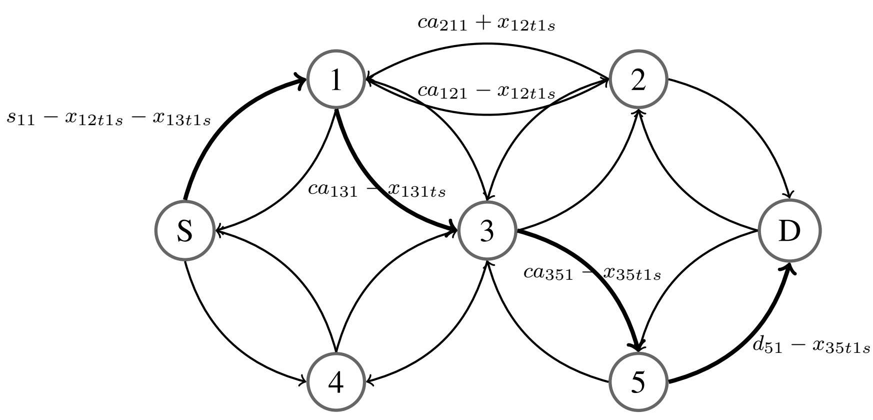

Figure 4.6: Residual network and the augmenting path for the example in Figure 4.4

Residual capacity calculation is carried out with flows of a given time period. However, changes in the repair schedules by the algorithm in different time periods may retrospectively affect the flows on the same arcs. To make sure that capacity of an arc is not exceeded in this way, residual capacities of each arc should be calculated in different time periods and the minimum of the residual capacities should be taken over multiple time periods. First, residual capacities are calculated with flows at t_m^{infr} time point to consider the latest flow values. Later, residual capacities should be updated for the teams that have $t_{km}^{crew} < t_m^{infr}$.

There are 10 possible cases of flow change on an arcs (i, j) and (j, i) between time periods t_{km}^{crew} and t_m^{infr}, which are depicted in Figure 4.7. The two labels on each arc denote the flow amount and the residual capacity of the arc, respectively. Variables F', F, f and f are used to denote the flow amounts on arcs, where $F' > F > f > 0$ and $f' > 0$. For instance, in case 1.a, flow on arc (i, j) is increased from time period t_{km}^{crew} to t_m^{infr}, and thus arc (j, i) has a larger residual capacity at t_m^{infr} than at t_{km}^{crew}. However, if a flow starting from t_{km}^{crew} is scheduled, the capacity of arc (j, i) may be exceeded between t_{km}^{crew} and t_m^{infr}. In this case, the residual capacity of arc (j, i) should be reduced to its residual capacity at time t_{km}^{crew}. Furthermore, the residual capacity of arc (i, j) should be decreased for time periods t_{km}^{crew} through t_m^{infr}. Since it is already the minimum of two capacities, an update is not needed in the residual capacity of (i, j). Problematic cases may occur when residual capacity of an arc increases for time periods t_{km}^{crew} through t_m^{infr}, and those residual capacities should be decreased to their values at t_{km}^{crew}. To overcome this, in the case of multiple teams in an infrastructure, residual capacities should be calculated at t_{km}^{crew} $\forall k \in K_m$ and the minimum of these residual capacities should be taken. This is due to the fact that the flow on an arc can change only when a repair is finished in the infrastructure. On the other hand, when the optimal solution of the flow optimization model is fed to the heuristic, flow on an arc can change at any time period. Checking the residual capacity at every time period would be cumbersome for the heuristic. That is why exceeding the capacity is allowed in the intermediate steps of the heuristic. In addition, residual capacities of the arcs emanating from interdependent nodes whose dependency is not yet satisfied are set to zero to prevent the algorithm from selecting those arcs.

The algorithm tries to find an augmenting path P^* in the resulting residual network

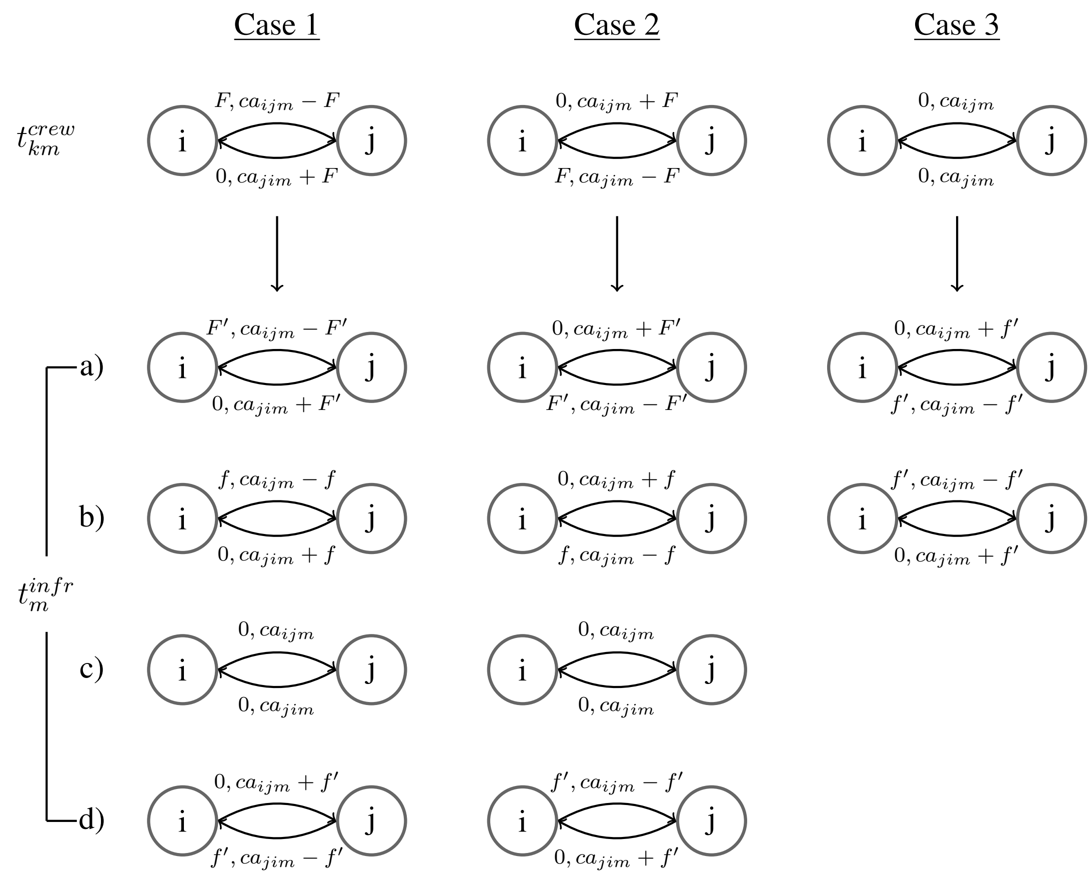

Figure 4.7: Residual capacity correction for $F' > F > f > 0$ and $f' > 0$

that has the maximum flow amount and minimum path repair time in order to satisfy the demand of the node by the largest amount as soon as possible, using the Augmenting Path Subroutine given in Algorithm 10. Towards this aim, equation (4.4) is used, where w_i is the weight of demand node at the end of the path P, r_P is the intended flow increase amount, p_P is the repair time of path P, T is length of the planning horizon, and t is the current time period.

$$P^* = argmax_P\{w_i r_P(T + 1 - p_P - t)\} \tag{4.4}$$

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. As there are multiple teams, repair will be finished sooner. In this sense, $r_P(T + 1 - p_P - t)$ is a lower bound on the objective function and equation (4.4) tries to find a path P^* maximizing this lower bound. The weights w_i are accepted as the same for a node over all infrastructures and used to give priority to interdependent nodes in the infrastructure by setting these to a large value at the beginning of FRESH. When a node has multiple interdependencies and one of them is satisfied during the iterations, the weight of the node is increased to infinity to satisfy its other dependencies right away. This is because interdependent demand nodes are generally supply nodes in another infrastructure and their supply is needed as soon as possible.

In order to find P^*, every possible r_P value is tried and p_P value is aimed to be minimized for that r_P value, as in Nurre et al. (2012). First, the residual network is reduced by deleting the arcs having residual capacity less than the chosen r_P value. Afterwards, this reduced network is fed to the Dijkstra Algorithm to find the shortest path from the supply supernode to the demand supernode. Arc lengths in the Dijkstra Algorithm are equal to the repair time for the damaged arcs and zero for undamaged or repaired arcs. If there is an interdependent supply node whose interdependency is not yet satisfied, the length of the arc between supply supernode and the interdependent node is set to infinity to avoid choosing that arc. Due to its properties, the Dijkstra Algorithm also reports shortest paths to all demand nodes from the supply supernode. Lengths of the shortest paths are the number of periods needed to repair the path

Algorithm 10 Augmenting Path Subroutine

Input: SIIRR parameters for $s \in S$ and $m \in M$, $rescap_{ijm}$, $weight_{ijm}$, t_{km}^{crew}, t, $tabu_{ijm}$, v_{itms}
Output: P^*, p_{P^*} and δ_{P^*}
$p_{P^*} \leftarrow 0$ and $\delta_{P^*} \leftarrow 0$
$RT_m = \sum_{k \in K_m} T - t_{km}^{crew} \quad \forall m \in M$
for $r_P = 1, 2, 3, ..., 2|A_m|$ **do**
  for all $(i, j) \in A_m$ **do**
    if $rc_{ijm} < r_P$ **then**
      $rc_{ijm} \leftarrow 0$ and $weight_{ijm} \leftarrow \infty$
    else
      $weight_{ijm} \leftarrow p_{ijm}$
    end if
  end for
  Call Dijkstra Algorithm with $weight_{ijm}$ to find shortest paths to all $i \in D_m$
  $Valid = \{ \}$ and $InValid = \{ \}$
  for all $i \in D_m$ **do**
    if $\exists\ (i, j) \in P_i : tabu_{ijm} > t$ **or** $p_{p_i} > RT_m$ where P_i is the SPT for $i \in D_m$ **then**
      $InValid \leftarrow InValid \cup \{P_i\}$
    else
      $Valid \leftarrow Valid \cup \{P_i\}$
      $\delta_i \leftarrow \min\{\min_{(i,j) \in P_i} rc_{ijm},\ \min_{i \in T_m, i \in P_i} cn_{im},\ d_{im} - v_{itms}\}$
    end if
  end for
  $P_r := argmax_{P_i\ \forall i \in D_m\ |P_i \in Valid}\ w_i \delta_{P_i}(T + 1 - p_{P_i} - t)$
  $r := argmax_{i\ \forall i \in D_m\ |P_i \in Valid}\ w_i \delta_{P_i}(T + 1 - p_{P_i} - t)$
  if $w_r \delta_{P_r}(T + 1 - p_{P_r} - t) > w_i \delta_{P^*}(T + 1 - p_{P^*} - t)$ **then**
    $P^* \leftarrow P_r$, $\delta_{P^*} \leftarrow \delta_{P_r}$, and $p_{P^*} \leftarrow p_{P_r}$
  end if
end for

completely. The maximum flows that can go through from these shortest paths, δ_P, are calculated by taking the minimum of the amount of unmet demand at the demand node, the residual capacities of the arcs and capacity of the transshipment nodes (if there is any) along the shortest paths. Some of the shortest paths found may include arcs that are currently under repair, i.e., their repair will be finished in a later time period than the current time t. These arcs and their reverse arcs are considered as forbidden to use, denoted as *tabu*, and shortest paths including these *tabu* arcs are eliminated. Reverse of the arcs is also considered *tabu*, because there is a possibility of incorrectly increasing the residual capacity of the forward arc. Likewise, paths that need longer time than the total remaining time are eliminated, since they cannot be repaired within the given time horizon. Valid shortest paths are found in this manner for all r_P values and demand nodes. These paths are compared using equation (4.4) by taking δ_P in the place of r_P to find the best path, P^*.

Now that P^* is at hand, it is easy to extract damaged arcs of P^* and schedule repairs for them using the Scheduling Subroutine described in Algorithm 11. From the demand node to the supply node, every arc is checked to see if it needs repair. Some arcs may be damaged but may not need repair. This special case occurs if augmenting path P^* tries to increase flow on damaged arc (i, j) in infrastructure m at time t in scenario s by δ_P and there is already flow on arc (j, i) in m at time t in scenario s which is greater than or equal to δ_P. In this case, flow on arc (j, i) in m is decreased by δ_P and no repair will be scheduled for arc (i, j) in m since arc (i, j) will not be used in the flow. To schedule a repair for an arc (i, j) in m, all of the following conditions should hold:

1. $(i, j) \in \bar{E}_{ms}$ and $\beta_{ijtms} = 0$
2. $x_{ijtms} = 0$
3. $t + p_{ijm} \leq T$
4. If $x_{jitms} > 0$ then $\delta_P > x_{jitms}$

If all of these conditions hold, repair of arc (i, j) is scheduled to the crew which has the minimum t^{crew}_{km} over all $k \in K_m$. These steps are followed for every damaged arc on P^* while updating t^{crew}_{km} after each repair. If a repair cannot be scheduled at some time point because of condition 3, that means there is not enough time to schedule that repair even for the crew which has the most idle time. In this case, scheduling efforts for P^* are stopped and the algorithm moves to the next infrastructure.

In the case where all needed repairs are scheduled, the latest repair time point is stored as the path repair time, t_{path}. Flow amounts on the arcs (i, j) of path P^* (and possibly on their reverse arcs (j, i)) are updated using the Flow Subroutine in Algorithm 12 starting from t_{path} using one of the 4 possible cases below:

1. If $x_{ij(t_{path}-1)ms} = 0$ and $x_{ji(t_{path}-1)ms} = 0$, then $x_{ijt_{path}ms} = x_{ij(t_{path}-1)ms} + \delta_{P^*}$ and $x_{jit_{path}ms} = x_{ji(t_{path}-1)ms}$
2. If $x_{ij(t_{path}-1)ms} > 0$ and $x_{ji(t_{path}-1)ms} = 0$, then $x_{ijt_{path}ms} = x_{ij(t_{path}-1)ms} + \delta_{P^*}$ and $x_{jit_{path}ms} = x_{ji(t_{path}-1)ms}$
3. If $x_{ij(t_{path}-1)ms} = 0$ and $x_{ji(t_{path}-1)ms} > 0$ and $x_{ji(t_{path}-1)ms} \geq \delta_{P^*}$, then $x_{ijt_{path}ms} = x_{ij(t_{path}-1)ms}$ and $x_{jit_{path}ms} = x_{ji(t_{path}-1)ms} - \delta_{P^*}$
4. If $x_{ij(t_{path}-1)ms} = 0$ and $x_{ji(t_{path}-1)ms} > 0$ and $x_{ji(t_{path}-1)ms} < \delta_{P^*}$, then $x_{ijt_{path}ms} = \delta_{P^*} - x_{ji(t_{path}-1)ms}$ and $x_{jit_{path}ms} = 0$

In case 1, flow is zero in both ways, thus flow on (i, j) is increased by δ_{P^*}. In case 2, there is already flow on the arc and flow on (i, j) is increased by δ_{P^*}. In case 3, flow on (i, j) is zero but there is flow in the reverse arc (j, i). If flow on (j, i) is greater than or equal to δ_{P^*}, flow on the reverse arc (j, i) is decreased by δ_{P^*}. If flow on (j, i) is less than δ_{P^*} as in case 4, the flow changes direction. Flow on (j, i) becomes zero and flow on (i, j) becomes $\delta_{P^*} - x_{ij(t_{path}-1)ms}$. In the algorithm, flow values are changed from path repair time, t_{path}, until the last time period, T, to calculate residual capacities easily in every time period. In other words, flows and demand satisfaction values are accepted as the same until the end of the schedule if a new P^* is not suggested at later iterations.

As the amount of demand at the end node of P^* is satisfied with an update on the flow values, demand satisfaction value of the demand node, $v_{it_{path}ms}$, is increased by δ_{P^*}. If the demand node is an interdependent node, it is possible that its interdependency may be satisfied. If so, the interdependency control variable $y_{ijmnt_{path}s}$ is set to 1 and

Algorithm 11 Scheduling Subroutine

Input: SIIRR parameters for $s \in S$ and $m \in M$, t^{crew}_{km}, β_{ijtms}, x_{ijtms}, P^*, p_{P^*} and δ_{P^*}

Output: $\alpha_{ijtkms}, tabu_{ijm}, p_{ijm}, t^{crew}_{km}, t_{path}, Flow$

$Flow \leftarrow 1$

$St \leftarrow i \in D_m$ at the end of P^*

while $St \neq S$ **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**

      $\alpha_{PtStt^{crew}_{k'm}k'ms} \leftarrow 1$

      $t^{crew}_{k'm} \leftarrow t^{crew}_{k'm} + p_{PtStm}$

      $tabu_{PtStm} \leftarrow t^{crew}_{k'm}$

      $tabu_{StPtm} \leftarrow t^{crew}_{k'm}$

      $p_{PtStm} \leftarrow 0$

      $\beta_{PtSttms} \leftarrow 1 \quad for \; t^{crew}_{k'm} \leq t \leq T$

      $t_{path} \leftarrow t^{crew}_{k'm}$

      $St \leftarrow Pt$

    else

      $Flow \leftarrow 0$

      Terminate Subroutine

    end if

  else

    $St \leftarrow Pt$

  end if

end while

Algorithm 12 Flow Subroutine

Input: SIIRR parameters for $s \in S$ and $m \in M$, x_{ijtms}, v_{itms}, y_{ijmnts} , $ycheck_{itm}$, t^{crew}_{km}, P^*, p_{P^*} and δ_{P^*}, t_{path}

Output: $x_{ijtms}, v_{itms}, y_{ijmnts}$, $ycheck_{itm}, w_i$

$St \leftarrow i \in D_m$ at the end of P^*

while $St \neq S$ **do**

  $Pt \leftarrow j|(j, St) \in P^*$

  if $x_{PtStt_{path}ms} = 0$ and $x_{StPtt_{path}ms} = 0$ **then**

    $x_{PtSttms} \leftarrow x_{PtSttms} + \delta_{P^*}$ for $t_{path} \leq t \leq T$

  else if $x_{PtStt_{path}ms} > 0$ and $x_{StPtt_{path}ms} = 0$ **then**

    $x_{PtSttms} \leftarrow x_{PtSttms} + \delta_{P^*}$ for $t_{path} \leq t \leq T$

  else if $x_{PtStt_{path}ms} = 0$ and $x_{StPtt_{path}ms} > 0$ and $x_{StPtt_{path}ms} \geq \delta_{P^*}$ **then**

    $x_{StPttms} \leftarrow x_{StPttms} - \delta_{P^*}$ for $t_{path} \leq t \leq T$

  else if $x_{PtStt_{path}ms} = 0$ and $x_{StPtt_{path}ms} > 0$ and $x_{StPtt_{path}ms} < \delta_{P^*}$ **then**

    $F \leftarrow x_{StPtt_{path}ms}$

    $x_{PtSttms} \leftarrow x_{PtSttms} + \delta_{P^*} - F$ for $t_{path} \leq t \leq T$

    $x_{StPttms} \leftarrow x_{StPttms} - F$ for $t_{path} \leq t \leq T$

  end if

  $St \leftarrow Pt$

end while

Set $v_{itms} \leftarrow v_{itms} + \delta_{P^*}$ 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**

  $y_{ijmnts} \leftarrow 1$ for $t_{path} \leq t \leq T$

  $ycheck_{jtn} \leftarrow ycheck_{jtn} + 1$ for $t_{path} \leq t \leq T$

  $w_j \leftarrow \infty$

end if

the weight of this node is increased to infinity to give this node the highest importance in the other infrastructures and to activate the supply point as soon as possible.

A new P^* is suggested in every iteration for the infrastructure considered in that iteration until an augmenting path cannot be found ($\delta_{P^*} = 0$) or no augmenting path fits into the repair schedule in any of the infrastructures. In that case, if all demand is satisfied, the schedule is complete, and the solution is returned to FRESH. Otherwise, the possibility of a new active supply point, i.e., an interdependent supply node whose interdependency is satisfied in later time periods than t_m^{infr}, is sought. Since repairs are independent from each other, t_m^{infr} are also different for every infrastructure. Thus, 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 if no augmenting path can be found in any of the infrastructures, but there is still unsatisfied demand. 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 the algorithm terminates. The objective function value of the current solution is calculated and returned with the current solution.

Note that 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. Repairs are not normally dependent on flows and can be made beforehand in the SIIRR problem. Due to this reason, the repair schedule generated by FRSCH is passed to RSIH, which will remove the gaps and compress the schedule.

4.2.2 Repair Schedule Improvement Heuristic (RSIH)

The Repair Schedule Improvement Heuristic (RSIH) in Algorithm 13 takes the repair schedule generated by FRSCH and removes the gaps from the schedule. For every team in every infrastructure, the schedule is checked from time period 0 to T. Every repair is shifted to the earliest time period possible. By this way, the schedule is compressed and returned to the FRESH. The last repair times of the crews, t_{km}^{crew}, and the infrastructures, t_m^{infr}, are also changed with this function. FRESH feeds this new repair schedule to the flow optimization mathematical 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 13 Repair Schedule Improvement Heuristic (RSIH)

Input: $\bar{E}_{ms}$, t_{km}^{crew}, t_m^{infr}, α_{ijtkms} and β_{ijtms}
Output: Updated t_{km}^{crew}, t_m^{infr}, α_{ijtkms} and β_{ijtms}
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**
        $\alpha_{ijt^*kms} \leftarrow 1$, $\alpha_{ijtkms} \leftarrow 0$
        $\beta_{ijt''ms} \leftarrow 1 \ \forall t^* \leq t'' < t'$
        $t_{km}^{crew} \leftarrow t^*$
      end if
      $(i,j) \leftarrow (u,v) : \ Rank_{uv} = Rank_{ij} + 1$
    end while
  end for
  $t_m^{infr} \leftarrow \max_{k\in K_m} t_{km}^{crew}$
end for

4.2.3 An Example Execution of FRESH

In this section, an example execution of FRESH is presented. In the example instance, there are three infrastructures and two repair teams work in each infrastructure. Each infrastructure has 24 nodes and 76 arcs in its network. In the considered scenario, 90% of all arcs are damaged in each infrastructure. The budget is set to zero for all infrastructures, and thus reinforcement efforts are not involved. The time horizon consists of 30 periods. Repair schedules generated by the heuristic in intermediate steps are presented as Gantt charts for better understanding.

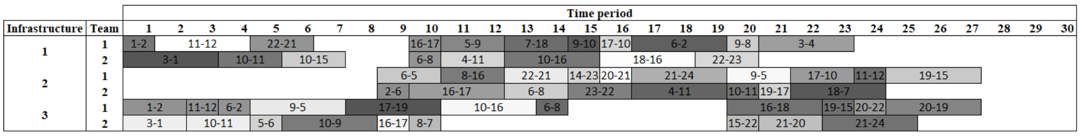

Figure 4.8: FRESH iteration 1 - FRSCH results

In Figure 4.8, repair schedule generated by FRSCH at iteration 1 of FRESH is presented. In the figure, labels inside the boxes refer to the arcs and box sizes are in accordance with the repair times of the arcs. For example, the first repair team of infrastructure 1 starts repairing arc $(1, 2)$ and its repair is finished at the beginning of time period 2. Repair activities start from time period 1 in infrastructures 1 and 3, while infrastructure 2 waits until time period 8. This is because all of the supply nodes of infrastructure 2 are interdependent on other infrastructures. At time period 8, one of the supply nodes becomes active after its interdependencies are satisfied. After time period 8, supply can flow through the network and new repairs are scheduled. Similarly, repair efforts stop at time period 14 in infrastructure 3. The reason behind this is that all available supply is already sent through the network and a new supply point will be available at time period 19 after its interdependencies are satisfied by other infrastructures. Since FRSCH schedules a repair only when a flow through a damaged arc is desired, repair teams wait until time period 19 for the new supply point to be available. At the beginning of time period 28, the last repairs are finished and FRSCH terminates since an augmenting path cannot be found in any of the infrastructures.

The repair schedule in Figure 4.8 is then fed to RSIH to eliminate the gaps in the schedule. The resulting repair schedule of the RSIH can be seen in Figure 4.9. The order of the repairs from the first call of FRSCH is preserved and schedule is arranged in a way that repair teams will have no idle time. After the modification of RSIH on the repair schedule, it is observed that by the time period 8, repair crews of infrastructure 2 have already repaired the damaged arcs that will be used once the interdependencies of its supply nodes are satisfied. Thus, infrastructure 2 can send its services to the prioritized demand nodes without possibly losing more time on repairs. The last repair is finished at the beginning of time period 23 after the modification and the repair schedule has idle time periods towards the end of the schedule,

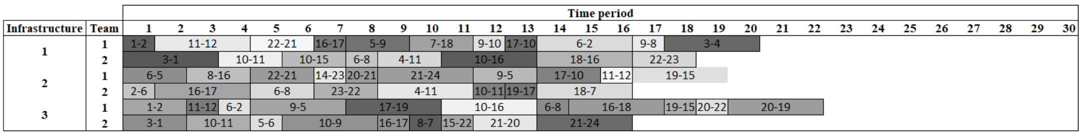

Figure 4.9: FRESH iteration 1 - RSIH results

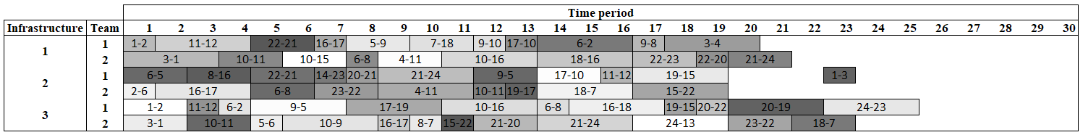

Figure 4.10: FRESH iteration 2 - FRSCH results

where new repairs can be scheduled.

The repair schedule modified by RSIH is fed to the flow optimization model to decide on the optimal flows, demand and interdependency satisfaction values for the proposed repair schedule. In the optimal solution of the model, demands of the nodes are not satisfied completely, and thus FRESH moves on to the second iteration. The repair schedule and the optimal flow, demand and interdependency satisfaction values are fed to the FRSCH, which in turn schedules new repairs to meet more demand as damaged arcs and possibly interdependent supply nodes are available sooner and different flow values suggested by the flow optimization model cause different residual capacities on the arcs. The repair schedule generated by FRSCH at iteration 2 is given in Figure 4.10. As can be seen in this figure, 8 new arc repairs are scheduled by the FRSCH.

In the second iteration of FRESH, these new repairs are shifted to the earliest idle time periods by the RSIH. The resulting repair schedule can be seen in Figure 4.11. The only modification is performed on arc $(1, 3)$ in infrastructure 2 and this repair is shifted from time period 22 to 19.

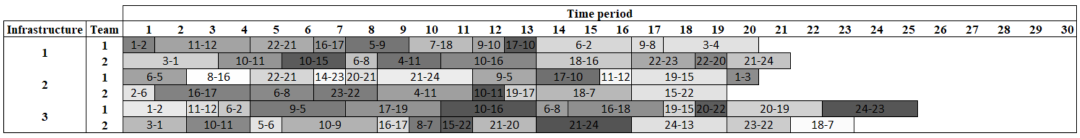

Figure 4.11: FRESH iteration 2 - RSIH results

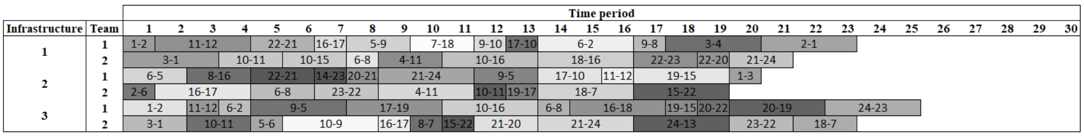

Figure 4.12: FRESH iteration 3 - FRSCH results

As in iteration 1, the repair schedule modified by the RSIH is fed to the flow optimization model. Again, the flow optimization model is unable to meet all of the demand of the infrastructures with the given repair schedule. Thus, FRESH moves on to the third iteration to search for new repairs. In iteration 3, only one more repair is suggested by the FRSCH, which can be seen in Figure 4.12. Arc $(2, 1)$ is repaired between time periods 21 and 23 and FRSCH is unable to suggest new augmenting paths. This new repair schedule is fed to the RSIH, but the repair schedule remains the same in Figure 4.12 since there are no gaps in the schedule.

Flows, demand and interdependency satisfaction values are decided by the flow optimization model at iteration 3 with the current repair schedule. In the results, it is observed that infrastructures 1 and 2 can satisfy their demand nodes completely after the last repair, while infrastructure 3 cannot. Thus, one more iteration of the FRESH is needed. In the fourth iteration, FRSCH is not able to schedule any repairs in any of the infrastructures. In infrastructures 1 and 2 there is no demand to satisfy, however, in infrastructure 3 there is demand to satisfy but no augmenting path can be found. This can be due to the *tabu* arcs or the lack of residual capacities on the arcs. Since there is no new repair suggested by the FRSCH, iterations of FRESH are terminated.

Notice that the repair schedule in Figure 4.12 still has idle time periods at the end of the schedule. In order to repair the infrastructures as much as possible, the remaining time periods are filled with arc repairs according to the shortest processing time first rule. The final repair schedule generated in this manner can be seen in Figure 4.13. This final repair schedule is fed to the flow optimization model one last time to obtain the final values of flow, demand and interdependency satisfaction values.

Demand satisfaction values determined at the end of the heuristic are depicted in Figure 4.14 as the percentage of the demand covered for each infrastructure. It is seen that none of the demand is covered in the first few time periods since the heuristic is

<table>
<tr><th rowspan="2">Infrastructure</th><th rowspan="2">Team</th><th colspan="30">Time period</th></tr>
<tr><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th><th>11</th><th>12</th><th>13</th><th>14</th><th>15</th><th>16</th><th>17</th><th>18</th><th>19</th><th>20</th><th>21</th><th>22</th><th>23</th><th>24</th><th>25</th><th>26</th><th>27</th><th>28</th><th>29</th><th>30</th></tr>
<tr><td rowspan="2">1</td><td>1</td><td>1-2</td><td colspan="3">11-12</td><td colspan="2">22-21</td><td>16-17</td><td colspan="2">5-9</td><td colspan="2">7-18</td><td>9-10</td><td>17-10</td><td colspan="3">6-2</td><td>9-8</td><td colspan="3">3-4</td><td colspan="3">2-1</td><td>5-6</td><td>8-6</td><td>9-5</td><td>12-11</td><td>13-12</td><td>16-18</td><td></td></tr>
<tr><td>2</td><td colspan="3">3-1</td><td colspan="2">10-11</td><td colspan="2">10-15</td><td>6-8</td><td colspan="2">4-11</td><td colspan="3">10-16</td><td colspan="3">18-16</td><td colspan="2">22-23</td><td>22-20</td><td colspan="2">21-24</td><td>4-3</td><td>5-4</td><td>6-5</td><td>8-7</td><td>11-10</td><td>12-13</td><td>16-8</td><td>20-18</td><td></td></tr>
<tr><td rowspan="2">2</td><td>1</td><td colspan="2">6-5</td><td colspan="2">8-16</td><td colspan="2">22-21</td><td>14-23</td><td>20-21</td><td colspan="3">21-24</td><td colspan="2">9-5</td><td colspan="2">17-10</td><td>11-12</td><td colspan="3">19-15</td><td>1-3</td><td>10-17</td><td>13-12</td><td>16-8</td><td>18-20</td><td>20-18</td><td>21-20</td><td colspan="2">1-2</td><td></td><td></td></tr>
<tr><td>2</td><td>2-6</td><td colspan="3">16-17</td><td colspan="2">6-8</td><td colspan="2">23-22</td><td colspan="3">4-11</td><td>10-11</td><td>19-17</td><td colspan="3">18-7</td><td colspan="3">15-22</td><td>3-12</td><td>11-10</td><td>14-11</td><td>16-10</td><td>19-20</td><td>20-19</td><td>22-15</td><td colspan="2">2-1</td><td></td><td></td></tr>
<tr><td rowspan="2">3</td><td>1</td><td colspan="2">1-2</td><td>11-12</td><td>6-2</td><td colspan="3">9-5</td><td colspan="3">17-19</td><td colspan="3">10-16</td><td>6-8</td><td colspan="3">16-18</td><td>19-15</td><td>20-22</td><td colspan="3">20-19</td><td colspan="3">24-23</td><td>6-5</td><td>11-4</td><td>15-19</td><td>19-17</td><td></td></tr>
<tr><td>2</td><td colspan="2">3-1</td><td colspan="2">10-11</td><td>5-6</td><td colspan="3">10-9</td><td>16-17</td><td>8-7</td><td>15-22</td><td colspan="2">21-20</td><td colspan="3">21-24</td><td colspan="3">24-13</td><td colspan="2">23-22</td><td colspan="2">18-7</td><td>4-11</td><td>5-9</td><td>10-17</td><td>15-10</td><td>17-16</td><td>23-14</td><td></td></tr>
</table>

Figure 4.13: FRESH - The final repair schedule

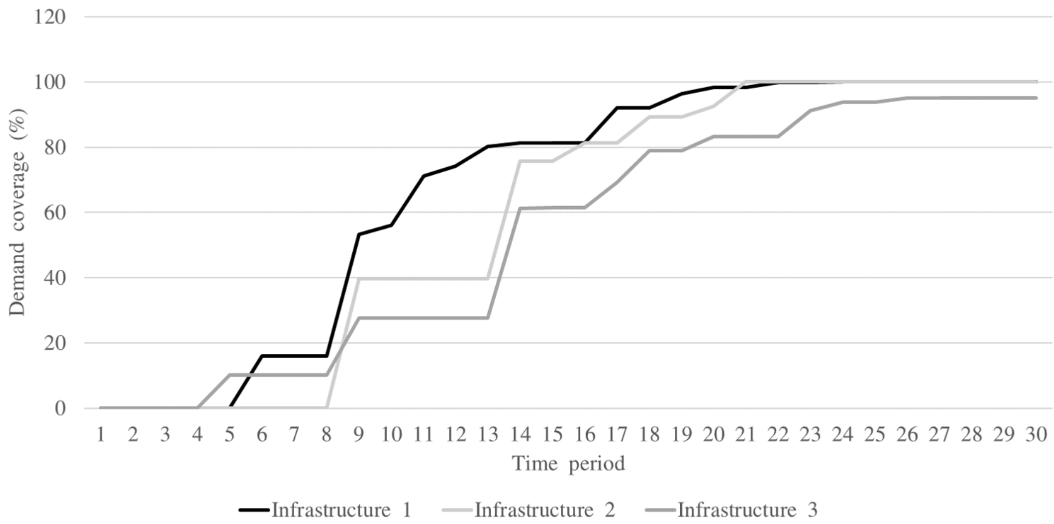

Figure 4.14: Demand satisfaction progress over the periods

repairing the arcs of the paths reaching to the prioritized nodes. Between time periods 6 and 8, demand satisfaction stays the same until a new supply point is available at infrastructure 3 at time period 8. A similar stagnation is observed in infrastructures 2 and 3 between time periods 9 and 13. After infrastructure 1 satisfies the dependencies of the supply points of infrastructures 2 and 3, demand satisfaction increases again. At the end of the planning horizon, infrastructures 1 and 2 cover all of their demands while infrastructure 3 can satisfy 95% of its demand.

CHAPTER 5

COMPUTATIONAL EXPERIMENTS

The performance of our proposed solution method presented in Chapter 4 is tested and reported in this chapter by means of computational experiments. Our numerical results are also used to answer several relevant research questions and to obtain important managerial insights. In §5.1, generation of the test instances are explained. §5.2 gives the algorithm settings of the genetic algorithm. Computational results are described in detail in §5.3.

5.1 Test Instances

Our test instances are generated using the network of Sioux Falls City in South Dakota, which is first presented in LeBlanc et al. (1975) for traffic equilibrium problems. The Sioux Falls network, which is presented in Figure 5.1, includes 24 nodes and 76 arcs. In our experiments, this network is assumed to be the network structure of all infrastructures. We consider three infrastructures, namely electricity, wastewater, and water. The number of demand, supply and transshipment nodes are determined in proportion with the New Hanover County data set in Sharkey et al. (2015b) for all infrastructures. The weights of the time periods ω_t and demand nodes w_{im} are accepted as a single unit. One time period is considered to represent 10 hours and the planning horizon lasts for 30 time periods (close to one week), as in Sharkey et al. (2015b). 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. Demand and supply amounts, node and arc capacities, repair time and costs of arc reinforcement are generated randomly using a uniform distribution. The generated

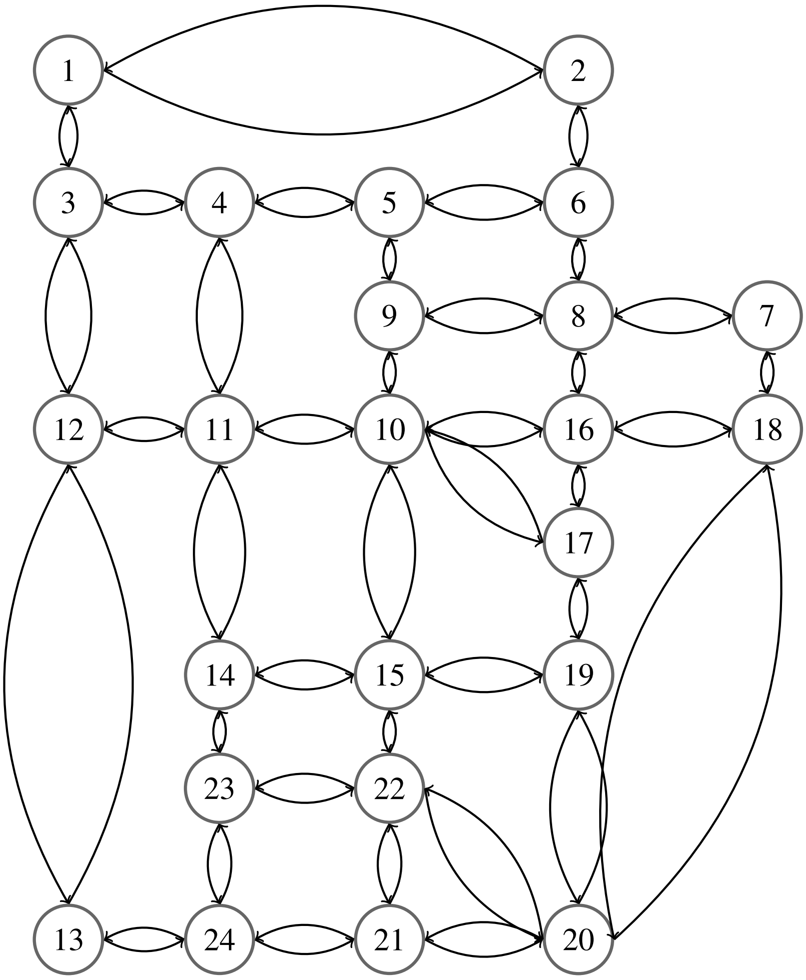

Figure 5.1: Sioux Falls City network

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 demand in all infrastructures. A summary of the sets and distribution of random parameters can be seen in Table 5.1. Costs of reinforcement, repair time and flow capacities of arcs are given in Table A.1 in Appendix A. Supply and demand values of the nodes and transshipment node capacities are provided in Table A.2 in Appendix A.

15 interdependencies are generated between pairs of infrastructures in coherence with the demand and supply amounts of the nodes. 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

Table 5.1: Summary of sets and probability distributions of random parameters

	Infrastructure		
	Electricity	**Water**	**Wastewater**
Supply nodes	{5,6,10}	{1,2,9,18,19}	{10,21}
Transshipment nodes	{13,14,17}	{8,10,13,23}	{3,4,6,8,11,14,15,23}
Demand nodes	{1,2,3,4,7,8,9,11,12, 15,16,18,19,20,21,22, 23,24}	{3,4,5,6,7,11,12,14, 15,16,17,20,21,22,24}	{1,2,5,7,9,12,13,16, 17,18,19,20,22,24}
Demand	Discrete U[1,100]	Discrete U[1,100]	Discrete U[1,100]
Supply	Discrete U[1,400]	Discrete U[1,400]	Discrete U[1,400]
Arc capacity	Discrete U[1,400]	Discrete U[1,400]	Discrete U[1,400]
Node capacity	Discrete U[1,100]	Discrete U[1,100]	Discrete U[1,100]
Repair time	Discrete U[1,3]	Discrete U[1,3]	Discrete U[1,3]
Reinforcement cost	Discrete U[1,3]	Discrete U[1,3]	Discrete U[1,3]

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 can be seen in Table 5.2. The second column of the table indicates service demanded by the dependent node, whereas the third column indicates the service it supplies. For instance, node 21 is a water treatment plant demanding electricity and wastewater in order to supply clean water. If electricity and wastewater demand of node 21 is not satisfied, water cannot be supplied from this node. Similarly, node 6 is a power plant generating electricity from wastewater. If wastewater demand of node 6 is not satisfied, electricity cannot be generated.

The scenarios of the SIIRR problem instances are created using 10%, 30%, 50%, 70% and 90% 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, 10% of the arcs of each infrastructure are randomly picked as damaged arcs. The master set of scenarios MS consists of 200 scenarios where each damage rate has a corresponding set of 40 scenarios. The scenarios in instances are generated from this master set, choosing 10 scenarios randomly for each damage rate. Accordingly, the instances consist of 50 scenarios in total and scenarios are equally likely

Table 5.2: Interdependencies between infrastructures

Node	Interdependency	
1	Electricity&Water	Wastewater
2	Electricity&Water	Wastewater
5	Water&Wastewater	Electricity
6	Wastewater	Electricity
9	Electricity&Water	Wastewater
18	Electricity&Water	Wastewater
19	Electricity&Water	Wastewater
21	Electricity&Wastewater	Water

to happen. The fNR_{ms} values are calculated for the disrupted infrastructures using the constraints of the SIIRR-SP model with an objective of maximizing the weighted demand met right after the disaster. 10 instances are generated in this manner to test the performance of the proposed heuristic method.

5.2 Algorithm Settings

The design parameters of the GA are set with the help of the literature as much as possible to decrease the computational burden of experimental design runs. 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.

The population size, seeding percentage, crossover probability, seeding strategy, and parent selection strategy are determined as design parameters and two levels are tested for each. The levels of these design parameters can be seen in Table 5.3.

A full factorial design would require 5×2^5 runs in total, assuming five runs per

Table 5.3: Levels of the design parameters of experimental design

Design Parameter	**Level 1**	**Level 2**
Population Size	20	50
Seeding Percentage, p_s	10%	40%
Crossover Probability, p_c	0.6	1.0
Parent Selection	R: Random	E: Based on fitness value
Seeding Strategy	R: Random	P: Based on expected objective value

configuration. Instead of a full factorial design, Taguchi Orthogonal Array Design is used to avoid an extensive amount of preliminary experiments. For five factors with two levels, Taguchi Design suggests eight configurations corresponding to orthogonal array L8. One test instance is selected for the experimental design and the GA is run five times with each configuration. Percent upper bound gap from the best upper bound on the objective function at the root node of Branch and Bound algorithm (z_B) is reported as the performance measure and calculated using equation (5.1), where z_H denotes average of the expected objective function values calculated by running the heuristic 5 times. Configurations and their average performance can be seen in Table 5.4.

$$\%\ UB\ Gap = \frac{z_B - z_H}{z_B} \tag{5.1}$$

The results of the experimental design runs are tested with Analysis of Variance (ANOVA) to measure significance of each design parameter. The ANOVA results can be seen in Figure 5.2. As can be seen, only the seeding percentage is insignificant in terms of variance with confidence level of 85%. However, its interaction with the crossover probability is significant. The model accurately captures the variance in data with an R-square adjusted value of 97.7%. The residuals are checked for normality, constant variation and independence with graphs in Figure 5.3 and model assumptions are observed to be satisfied.

The main effect plot and interaction plot for Signal to Noise (SN) ratio in Figure 5.4 are examined to suggest a promising new configuration, labeled as configuration 9 in

Table 5.4: Paramater configurations for experimental design and their average performances (E: elitist, R: random, P: performance-based)

Design parameters	**Configurations**								
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Pop. Size	20	20	20	50	20	50	50	50	50
p_s	40%	10%	40%	10%	10%	10%	40%	40%	40%
p_c	0.6	0.6	1	1	1	0.6	0.6	1	1
Parent Sel.	E	R	E	E	R	E	R	R	E
Seeding Str.	R	R	P	R	P	P	P	R	P
z_H	80.52	80.36	81.15	81.06	80.61	81.12	80.72	80.92	81.16
% Gap	7.27%	7.46%	6.55%	6.64%	7.17%	6.58%	7.03%	6.80%	6.53%

```
          Estimated Model Coefficients for SN ratios

              Term       Coef   SE Coef          T      P
          Constant    38.1484  0.001710  22307.026  0.000
   Population Size    -0.0162  0.001710     -9.445  0.067
               p_s    -0.0024  0.001710     -1.375  0.400
               p_c    -0.0137  0.001710     -7.982  0.079
  Parent Selection     0.0168  0.001710      9.813  0.065
  Seeding Strategy     0.0099  0.001710      5.798  0.109
           p_s*p_c     0.0085  0.001710      4.964  0.127

       S = 0.004837   R-Sq = 99.7%   R-Sq(adj) = 97.7%

                  Analysis of Variance for SN ratios

           Source  DF    Seq SS    Adj SS    Adj MS      F      P
  Population Size   1  0.002087  0.002087  0.002087  89.20  0.067
              p_s   1  0.000044  0.000044  0.000044   1.89  0.400
              p_c   1  0.001491  0.001491  0.001491  63.71  0.079
 Parent Selection   1  0.002253  0.002253  0.002253  96.29  0.065
 Seeding Strategy   1  0.000786  0.000786  0.000786  33.62  0.109
          p_s*p_c   1  0.000576  0.000576  0.000576  24.64  0.127
   Residual Error                1  0.000023  0.000023  0.000023
                       Total                           7  0.007261
```

Figure 5.2: ANOVA results

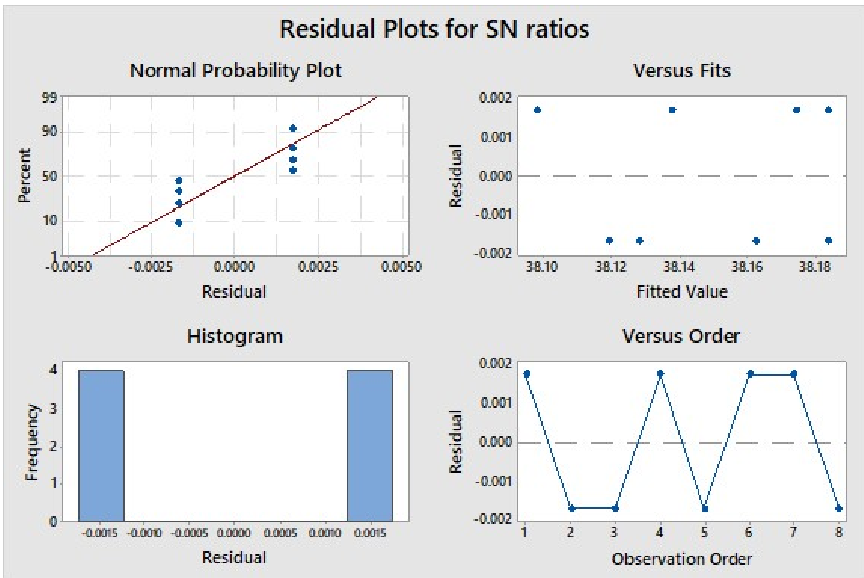

Figure 5.3: Residual plots

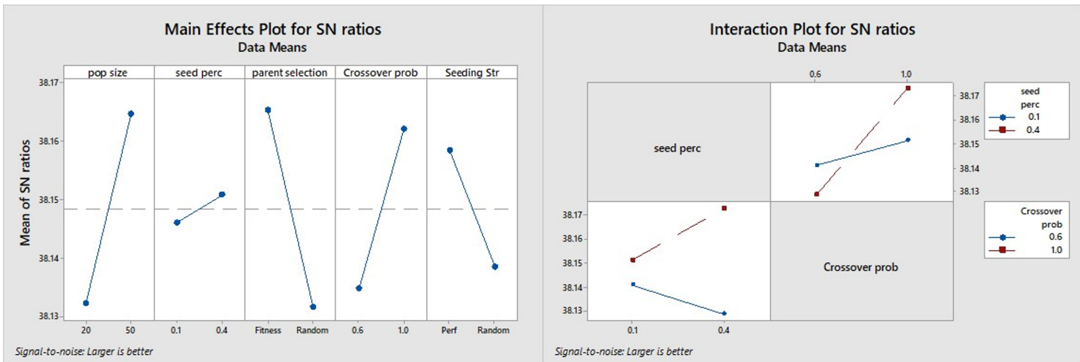

Figure 5.4: Main effect and interaction plots

Table 5.4. The GA is run five times with configuration 9 and its average performance can be seen in Table 5.4. Since configuration 9 is the best performing configuration, it is selected to use in further computational experiments.

Convergence of the algorithm is depicted for one run of the test instance in Figure 5.5. Convergence behavior for the remaining runs is similar. The population average approaches the population best as convergence is reached. Sharp drops in both lines are due to the fact that as the penalization coefficient increases, infeasible solutions are eliminated from the population and the population average and population best values first drop, and then increase as better solutions are found. 100 iterations are clearly satisfactory for the GA to converge to a good solution for our problem.

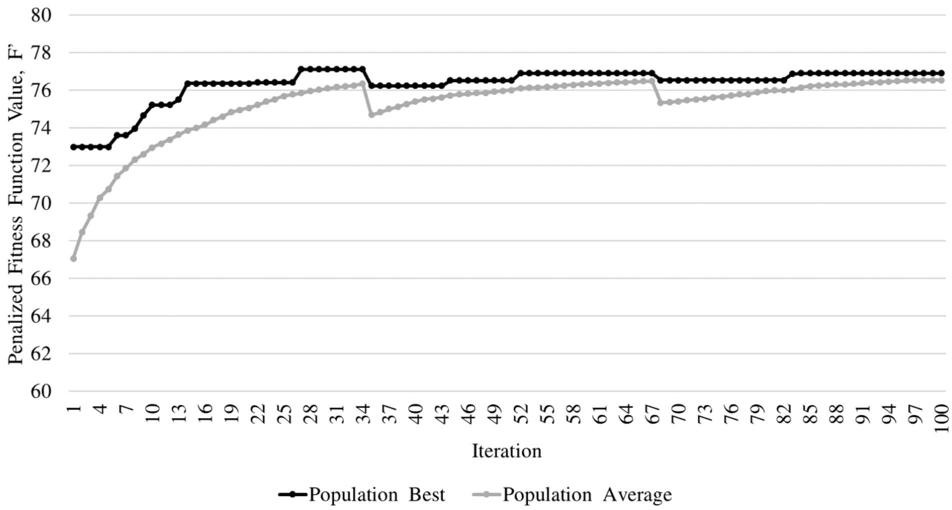

Figure 5.5: Penalized fitness value of population best and average of population through the iterations of the GA

5.3 Computational Results

The mathematical model 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. Multi-dimensional arrays in the two-stage SIIRR problem are flattened, i.e., transformed to vector representation, in order to reduce the run times. Test environment is an Intel(R) Core (TM) i7-477OS CPU @3.10GHz, 16GB RAM Windows 10 PC.

Computational results of this study aim to answer four types of research questions regarding the performance of the heuristic algorithm, managerial insights, sensitivity analysis, and the value of capturing stochasticity. In the following sections, answers of each such question are presented.

5.3.1 Algorithmic Questions

For our instances, CPLEX is unable to find the optimal solutions within the given time limit of 3 days. As a result, we use the best upper bound values reported by CPLEX as the benchmark to measure the performance of the algorithm. The best

Table 5.5: The SIIRR-SP results with CPLEX solver

Instance	z_B	z_{BI}	**CPU (sec)**	**Optimality Gap (%)**
1	79.00	73.61	259,234	7.33
2	80.74	62.75	259,241	28.68
3	77.66	-1551.21	259,233	105.01
4	79.46	57.31	259,233	38.64
5	76.13	-10097.05	259,232	100.75
6	78.31	71.30	259,232	9.84
7	80.22	-1847.50	259,231	104.34
8	78.90	59.80	259,235	31.92
9	77.78	-588.90	259,232	113.21
10	79.67	47.58	259,236	67.43
Avg.	78.79	-1371.23	259,234	60.72

bound on the objective function z_B, the objective function value found by CPLEX within the given time limit z_{BI}, CPU time and the optimality gap of each instance are presented in Table 5.5. The optimality gap values are calculated by CPLEX using equation (5.2). The minimum optimality gap is 7.33% while the average optimality gap is 60.72% for the instances.

$$Optimality\ Gap = \frac{|z_B - z_{BI}|}{|z_{BI}|} \tag{5.2}$$

The percent upper bound gaps (% UB Gap) that are presented in §5.3.1.1 and §5.3.1.2 for the heuristic solutions are calculated with z_B values in Table 5.5 using equation (5.1) for every instance. As the optimal value of the SIIPP-SP is not known and z_B is used to approximate it, the % UB gaps presented are the upper bounds on the percentage gaps from the actual optimal values.

5.3.1.1 *What is the price of using the heuristic approach, as opposed to an optimal solution?*

In order to evaluate the solution quality of the proposed heuristic approach, we make use of the % UB gap values. For this purpose, each one of the 10 instances is run

Table 5.6: The objective values, % UB gaps and CPU times of the heuristic solutions

Instance	z_H	**Avg. % UB Gap**	**Min. % UB Gap**	**Max. % UB Gap**	**Avg. CPU (sec)**
1	72.50	8.24	6.81	10.15	2,599
2	73.93	8.44	7.00	9.48	2,265
3	71.38	8.08	7.00	8.53	2,692
4	73.34	7.70	7.26	8.41	3,068
5	69.35	8.90	8.50	9.56	2,444
6	71.41	8.81	7.87	9.40	3,365
7	74.01	7.75	6.63	8.56	2,800
8	71.20	9.76	9.14	10.97	2,436
9	72.00	7.43	6.97	8.06	2,573
10	70.94	10.95	9.21	13.26	2,878
Avg.	72.01	8.61	7.64	9.64	2,712

5 times using the GA and the resulting first-stage decisions are fed to the FRESH for each scenario $s \in S$ to determine the second-stage decisions and the expected objective value. Average, minimum, and maximum % UB gap of each instance can be seen in Table 5.6, as well as the average of the expected objective function values z_H, and average CPU times. The expected objective function values for each instance in each individual run are presented in Table B.1 in Appendix B. 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. Looking at the closely distributed % UB gap values, it can be said that the heuristic generates robust results. Furthermore, the heuristic method is able to produce these results in approximately 1% of the time spent by CPLEX. Comparing the z_{BI} values in Table 5.5 with z_H values, one can conclude that the proposed heuristic approach finds significantly better results than CPLEX does in 3 days for almost all of the instances in an average time of approximately 45 minutes. Only for instance 1, CPLEX finds a better result than the proposed heuristic approach.

Table 5.7: Summary of objective values, gaps, and CPU times to evaluate the price of using a representative scenario

			CPU (sec)			Optimality Gap (%)	
Instance	z_{SPR+SP}	**% UB Gap**	**1st Stage**	**2nd Stage**	**Total**	**1st Stage**	**2nd Stage**
1	76.77	2.82	10,801	10,699	21,501	0.00	0.00
2	78.10	3.28	10,801	29,356	40,157	0.01	0.00
3	76.12	1.98	10,801	32,533	43,334	0.01	0.01
4	77.17	2.88	10,801	73,125	83,926	0.01	0.00
5	74.32	2.37	10,801	28,650	39,451	0.00	0.02
6	75.44	3.67	10,801	34,099	44,901	0.00	0.00
7	78.24	2.47	10,801	80,580	91,381	0.00	0.00
8	76.28	3.32	6,364	29,863	36,227	0.00	0.00
9	75.71	2.67	10,801	35,688	46,489	0.00	0.01
10	76.76	3.65	10,801	8,759	19,560	0.01	0.02
Avg.	76.49	2.91	10,357	36,335	46,693	0.01	0.01

5.3.1.2 *What is the price of using a representative scenario?*

To calculate the loss in the objective from using a representative scenario, as opposed to considering all 50 scenarios when obtaining a first-stage solution, we first solve the SIIRR-SP model (with a 3-hour time limit) with only the representative scenario to determine the first-stage decisions. Afterwards, we solve the SIIRR-SP model with all scenarios $s \in S$ (with a 3-hour time limit for each scenario) by fixing the first-stage decisions to obtain the second-stage decisions and the expected objective value, z_{SPR+SP}. The % UB gap of this solution is used to evaluate the price of using a representative scenario. The z_{SPR+SP} values, the % UB gaps, CPU times, and the optimality gaps (as reported by CPLEX) of each instance are presented in Table 5.7. From these values, we conclude that for our instances, the price of using a representative scenario is an average loss of 2.91% in solution quality, with a savings of 82% in the run time on average.

5.3.1.3 *What is the price of using a heuristic second-stage solution?*

To evaluate the loss of solution quality due to solving the second-stage problem heuristically using FRESH, as opposed to an optimal second-stage solution, SIIRR-SP is first solved (with a 3-hour time limit) with only the representative scenario to determine the first-stage decisions. Then, the first-stage decisions are fed to the second-stage heuristic FRESH for each scenario $s \in S$ to obtain the second-stage decisions and the expected objective value, $z_{SPR+FRESH}$. We then use *percent second-stage gap (% SS gap)*, which is the relative gap between $z_{SPR+FRESH}$ and z_{SPR+SP}, to quantify the price of a heuristic second-stage solution. % SS gap can be calculated by:

$$\%\ SS\ gap = \frac{z_{SPR+SP} - z_{SPR+FRESH}}{z_{SPR+SP}} \tag{5.3}$$

The $z_{SPR+FRESH}$ values, CPU times, and the % SS gaps of the instances are provided in Table 5.8.

From Table 5.8, the average solution quality loss due to using the FRESH for the second stage is observed as 6.11%, with average savings of 78% in the CPU time. Moreover, a comparison to the results in Table 5.6 reveals that using the GA in the first stage and the FRESH in the second stage brings slightly better results than solving the representative scenario to optimality and using the FRESH for the second stage, both in terms of objective value and CPU time.

5.3.1.4 *What is the performance of the genetic algorithm for a single scenario?*

As the near-optimal solution of the SIIRR-SP model with the representative scenario is known for each instance, these can be used to measure the performance of the GA. For this purpose, we calculate *percent GA gap* as in equation (5.4), as the relative gap between the objective function values obtained by solving the representative scenario using GA z_{GA}, and using the SIIRR-SP (with a 3-hour time limit) z_{SPR}.

$$\%\ GA\ gap = \frac{z_{SPR} - z_{GA}}{z_{SPR}} \tag{5.4}$$

Table 5.8: Summary of objective values, gaps, and CPU times to evaluate the price of using a heuristic for the second stage

		CPU (sec)			% Opt. Gap	
Instance	$z_{SPR+FRESH}$	1^{st} **Stage**	2^{nd} **Stage**	**Total**	1^{st} **Stage**	**% SS Gap**
1	70.96	10,801	36	10,837	0.00	7.57
2	73.29	10,801	33	10,834	0.01	6.16
3	71.72	10,801	31	10,832	0.01	5.78
4	72.98	10,801	30	10,831	0.01	5.43
5	70.41	10,801	29	10,830	0.00	5.26
6	69.73	10,801	32	10,833	0.00	7.56
7	73.77	10,801	31	10,832	0.00	5.72
8	71.29	6,364	31	6,395	0.00	6.53
9	71.57	10,801	31	10,832	0.00	5.46
10	72.45	10,801	34	10,835	0.01	5.61
Avg.	71.82	10,357	32	10,389	0.01	6.11

In Table 5.9, z_{GA} and z_{SPR} values are shared, together with the CPU times and the % GA gap values. It is seen that on average, the GA deviates from the near-optimal solution of the representative scenario by 7.52% in return of a 74% decrease in the run time.

5.3.1.5 *What is the performance of FRESH for the second-stage problem?*

In order to observe the performance of the second-stage heuristic FRESH, the first stage of the SIIRR problem is eliminated by equating the reinforcement budget to zero for all infrastructures. The results without any reinforcement efforts (with zero reinforcement budget) are obtained by solving the FRESH for all scenarios $s \in S$ by taking r_{ijm} values equal to zero. As the first stage of SIIRR-SP is eliminated, near-optimal solutions of the instances can also be found. In Table 5.10, heuristic solutions obtained by FRESH (z_{FRESH}), the near-optimal solutions of SIIRR-SP (with a 3-hour time limit for each scenario) (z_{SP}), and the percentage gap from the near-optimal solutions are presented, as well as the CPU times and the optimality gaps of the near-optimal solutions. It is observed that the FRESH deviates from the near-optimal

Table 5.9: Summary of objective values, gaps, and CPU times to evaluate the performance of the genetic algorithm

Instance	z_{GA}	**CPU (sec)**	z_{SPR}	**CPU (sec)**	**Opt. Gap (%)**	**% GA Gap**
1	76.60	2,563	83.61	10,801	0.00	8.38
2	77.43	2,233	82.46	10,801	0.01	6.10
3	76.46	2,663	81.39	10,801	0.01	6.06
4	76.37	3,038	83.09	10,801	0.01	8.08
5	76.03	2,415	82.77	10,801	0.00	8.14
6	74.61	3,337	83.09	10,801	0.00	10.21
7	76.79	2,771	82.83	10,801	0.00	7.29
8	77.73	2,405	82.84	6,364	0.00	6.17
9	76.39	2,542	82.05	10,801	0.00	6.89
10	76.58	2,847	83.15	10,801	0.01	7.89
Avg.	76.50	2,681	82.73	10,357	0.01	7.52

solution by 12.40% on average. On the other hand, FRESH spends approximately half a minute to find a solution, while the SIIRR-SP spends 196,805 seconds on average.

5.3.2 Managerial Insights

In this section, we answer a number of managerial research questions by changing the input data or the functioning of the heuristic algorithm and comparing the results with the baseline results presented in Table 5.6 of §5.3.1.1. The modified instances are run five times using the heuristic approach and the averages of the expected objective function values are compared to those of the baseline instances. The percentage improvement values over the baseline instances (denoted as % Improvement) are calculated for every instance using equation (5.5).

$$\% \, Improvement = \frac{z_{new} - z_{base}}{z_{base}} \tag{5.5}$$

In cases where equation (5.5) returns a negative value, this implies a decline in the objective value.

Table 5.10: Summary of objective values, gaps, and CPU times to evaluate the performance of FRESH for the second-stage problem

Instance	z_{FRESH}	**CPU (sec)**	z_{SP}	**CPU (sec)**	**Opt. Gap(%)**	**% Gap**
1	60.71	27	69.70	199,860	0.01	12.90
2	63.45	27	71.38	202,217	0.01	11.12
3	59.10	35	68.31	198,106	0.02	13.50
4	61.98	26	70.73	213,730	0.01	12.37
5	59.01	24	66.84	199,936	0.02	11.72
6	60.67	21	69.31	197,473	0.01	12.46
7	63.11	30	70.96	182,134	0.01	11.06
8	59.94	32	69.75	180,803	0.01	14.06
9	59.69	28	68.65	192,492	0.02	13.05
10	62.30	27	70.57	201,298	0.02	11.72
Avg.	61.00	28	69.62	196,805	0.01	12.40

5.3.2.1 *What is the value of reinforcement and its effect on restoration?*

To measure the effect of reinforcement efforts on the heuristic results, the heuristic results without the reinforcement efforts z_{FRESH} in Table 5.10 are compared with the results in Tables 5.6 and 5.5. Table 5.11 demonstrates the percentage improvement over the heuristic results if reinforcement activities are performed. GA+FRESH denotes the method of using the GA for the first stage and the FRESH for the second stage of the problem. Percentage improvement of using $z_{GA+FRESH}$ over z_{FRESH} values are presented in this column. In the column labeled SIIRR-SP, z_B and z_{FRESH} values are compared. Since z_B denotes the best bound obtained by solving the SIIRR-SP with a 3-day time limit, the value reported for each instance in this column denotes an upper bound on the percentage improvement of using SIIRR-SP with all scenarios $s \in S$ over the FRESH solution without any reinforcement. It can be inferred from Table 5.11 that if SIIRR-SP is used instead of the FRESH without any reinforcement, a maximum improvement of 29.20% could be achieved on average. The results of FRESH without any reinforcement can be improved by 18.08% on average if the GA is used for the first-stage (reinforcement) decisions. Comparing the results of FRESH

Table 5.11: The percentage improvement over the heuristic results when reinforcement efforts are performed

Instance	GA+FRESH	SIIRR-SP
1	19.41	30.13
2	16.52	27.27
3	20.79	31.42
4	18.32	28.19
5	17.52	29.01
6	17.71	29.08
7	17.27	27.12
8	18.78	31.62
9	20.63	30.31
10	13.87	27.87
Avg.	18.08	29.20

to that of GA+FRESH, the value of reinforcing the arcs before the disaster is estimated as 18.08% on average. This substantial increase emphasizes the importance of the reinforcement activities before the disaster.

5.3.2.2 *What is the benefit of planning reinforcement activities centrally?*

In general, the reinforcement activities are planned independently for each infrastructure by their decision makers. In this part, we aim to measure the benefit of planning the reinforcement activities centrally (e.g., by a local government or a municipality) for all the infrastructures. Towards this aim, the penalized fitness value calculation equation (4.2) for the GA is changed as in equation (5.6), where B is the cumulative reinforcement budget.

$$F' = max \left\{ 0, F - c_p \frac{\sum_{m \in M} \sum_{(i,j) \in A_m} r_{ijm} c_{ijm} - B}{B} \right\} \tag{5.6}$$

For the new instances, B is set to 5% of the total cost of reinforcement of all arcs

Table 5.12: The benefit of planning reinforcement activities centrally

Instance	z_{base}	z_{new}	**% Improvement**
1	72.50	72.15	-0.48
2	73.93	73.07	-1.16
3	71.38	71.35	-0.04
4	73.34	74.16	1.12
5	69.35	69.67	0.46
6	71.41	71.52	0.15
7	74.01	74.74	1.00
8	71.20	71.09	-0.16
9	72.00	72.09	0.13
10	70.94	71.82	1.24
Avg.	72.01	72.17	0.23

in all infrastructures. The new instances are run 5 times and the average expected objective values z_{new} are compared with that of the baseline results. In Table 5.12, the percentage improvement values are represented with z_{base} and z_{new} values. If we were to compare the optimal solutions of these instances, we would observe an improvement (or at least the same performance) in all of the instances. However, a performance decline is observed in some of the instances which could be attributed to the optimality gap of the heuristic approach. On average, the expected objective value increases by 0.23%, which is not a very significant increase. This implies that the proposed approach does not require a centralized planning of the budget, which eliminates the need for additional coordination efforts for this end.

On the other hand, if the reinforcement activities are planned centrally, 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 5.13 for independent and central decision making cases. If the reinforcement activities are planned centrally, budget spend on the reinforcement activities of the wastewater

Table 5.13: Budget distribution in independent and central cases

	Independent			Central		
Instances	**Electricity**	**Wastewater**	**Water**	**Electricity**	**Wastewater**	**Water**
1	8	8.2	8	10.8	3.4	9.8
2	8	8.8	8	12.2	3	8.2
3	8	9	8	11.6	5	7.4
4	7.6	8.4	7.4	13.2	3.8	7
5	8	8.2	8	10.8	3.4	9.8
6	7.8	8.4	7.8	12	3	8.8
7	8	9	7.8	11	4.2	8.8
8	8	8.6	8	10.4	6.6	7
9	8	9	7.8	10.4	5.4	8
10	7.6	7.8	7.8	11.6	4	8.2
Avg.	7.9	8.54	7.86	11.4	4.18	8.3

infrastructure is shifted mainly to the reinforcement activities of the electricity infrastructure. Furthermore, budget spent on the reinforcement activities of the water infrastructure increases slightly. Deciding on the reinforcement activities centrally clearly brings a 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.

5.3.2.3 *What is the price of ignoring interdependencies?*

For the SIIRR problem, one of the main difficulties 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. We aim to consider this case as a benchmark by ignoring the interdependencies while finding the first-stage decisions using the GA and then feeding the resulting first-stage decisions to FRESH after including the interdependencies. The new average expected objective function values, z_{new1} and the percentage decline values over the baseline results are presented in Table 5.14, together with the baseline results z_{base}.

Table 5.14: The price of ignoring the interdependencies when planning the restoration activities

Instance	z_{base}	z_{new1}	**% Decline**	z_{new2}	**% Overestimation**
1	72.50	70.12	3.28	74.12	5.71
2	73.93	70.84	4.18	74.41	5.05
3	71.38	68.32	4.29	72.18	5.65
4	73.34	71.03	3.14	72.05	1.43
5	69.35	68.20	1.65	71.40	4.69
6	71.41	69.29	2.98	73.59	6.21
7	74.01	72.08	2.60	74.36	3.17
8	71.20	69.69	2.11	73.07	4.84
9	72.00	69.52	3.44	71.73	3.18
10	70.94	70.35	0.84	74.01	5.20
Avg.	72.01	69.94	2.85	73.09	4.51

Furthermore, we aim to estimate the magnitude of overestimation if the interdependencies are ignored by solving the instances without considering the interdependencies in GA or FRESH. In Table 5.14, one can see the perceived average expected objective function values, z_{new2} and the corresponding percentage overestimation of the z_{new1} values for these solutions as well. We conclude that for our instances, the decision makers are actually reducing the overall infrastructure performance by 2.85% while they are overestimating the performance by 4.51% on average, if the interdependencies between the infrastructures are ignored.

5.3.2.4 *What is the benefit of holding extra service capacities in the interdependent nodes?*

Keeping in mind the effect of ignoring the interdependencies when planning the restoration activities, we also aim to investigate the effect of holding extra capacities in the dependent nodes. We assume that there are generators in the electricity-dependent supply nodes of water and wastewater infrastructures and their electricity dependency can be satisfied with these generators after the disaster. In this manner,

Table 5.15: The percentage improvement values when electricity-related dependencies are eliminated

Instance	z_{base}	z_{new}	**% Improvement**
1	72.50	73.60	1.52
2	73.93	73.90	-0.03
3	71.38	71.86	0.66
4	73.34	74.29	1.29
5	69.35	70.12	1.11
6	71.41	71.15	-0.37
7	74.01	74.03	0.03
8	71.20	71.29	0.13
9	72.00	72.05	0.06
10	70.94	71.23	0.40
Avg.	72.01	72.35	0.48

electricity related interdependencies are eliminated in the new instances and the results are presented in Table 5.15. In two of the 10 instances, there is a slight decrease in the expected objective values, which is likely due to the heuristic performance. In the remaining instances, expected objective values increase up to 1.52%. On average, a 0.48% increase is observed when electricity related interdependencies are eliminated. The reason for this minor increase may be due to the fact that other interdependencies are still binding and as long as all the interdependencies of a node are not satisfied, the dependent nodes remain inactive. Even so, the decision makers should compare this performance increase with the investment on the generators and make a decision accordingly.

5.3.3 Sensitivity-Related Questions

In this section, we answer the sensitivity-related research questions of this study. Sensitivity changes are performed one at a time on the baseline instances described in Table 5.6 of §5.3.1. Instances are run five times with the GA for the first-stage decisions and with FRESH for the second-stage decisions and the average of the expected objective values are compared with that of baseline instances using the equation (5.5). From time to time, the penalization coefficient of the GA falls short of eliminating the infeasible solutions at the end of the run for the new instances. In such a case, the run is repeated until a feasible solution is obtained.

5.3.3.1 *What is the effect of increasing the available reinforcement budget?*

In order to measure the effect of reinforcement budget, it is increased to 10% of the reinforcement cost of all arcs in the infrastructure for all infrastructures. New heuristic results, z_{new}, are compared with the results of 0 and 5% reinforcement budgets introduced in Tables 5.10 and 5.6 respectively. The percentage improvement observed while increasing the budget is presented in Table 5.16. It can be concluded that increasing the reinforcement budget from 0 to 5% brings an improvement of 18.08% while increasing it further to 10% brings an additional 5.80% improvement on average. Improvement on the results decreases by approximately threefold from 5% to 10% reinforcement budget. Marginal contribution of additional budget is expected to decrease as we increase the reinforcement budget further. A reinforcement budget of 40% is observed to make the second stage of the problem purposeless in the initial runs. 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.

5.3.3.2 *What is the effect of decreasing the number of teams?*

The effect of changing the number of repair teams in every infrastructure is measured by setting the number of teams in every infrastructure from two to one. The new average expected objective function values and percentage decline values from the baseline instances are presented in Table 5.17 along with the baseline results. It is observed that the expected objective function value decreases by up to 13.68% in this case. On average, 10.28% decrease is expected if the repair teams are reduced from two teams to one team per infrastructure. This substantial value underlines the importance of the repair teams. In general, increasing the number of teams may

Table 5.16: z_{new} and the percentage improvement values when increasing the reinforcement budget

Instance	z_{new}	**0% to 5%**	**5% to 10%**
1	75.92	19.41	4.72
2	77.19	16.52	4.42
3	75.15	20.79	5.28
4	77.70	18.32	5.95
5	73.75	17.52	6.35
6	75.63	17.71	5.90
7	77.58	17.27	4.83
8	76.18	18.78	7.00
9	76.46	20.63	6.19
10	76.17	13.87	7.37
Avg.	76.17	18.08	5.80

increase the performance of the repair efforts even if the marginal benefit is expected to decrease for every additional team.

5.3.3.3 *What is the effect of increasing the repair times?*

The repair times of the arcs are doubled to measure the effect of increasing the repair times. The heuristic results are compared with the baseline instances and the percentage decline values are reported in Table 5.18 together with the average expected objective value z_{new}, and the baseline results z_{base}. A 12.16% decline is observed on average when the repair times are doubled. That is because significantly less number of valid augmenting paths can be proposed by the heuristic when the repair times are longer.

Table 5.17: The percentage decline values when number of teams in every infrastructure is decreased to one

Instance	z_{base}	z_{new}	**% Decline**
1	72.50	65.77	9.28
2	73.93	64.70	12.48
3	71.38	63.81	10.60
4	73.34	67.43	8.06
5	69.35	61.84	10.82
6	71.41	64.14	10.19
7	74.01	66.94	9.55
8	71.20	61.46	13.68
9	72.00	65.21	9.43
10	70.94	64.80	8.66
Avg.	72.01	64.61	10.28

Table 5.18: The percentage decline values when repair times are doubled

Instance	z_{base}	z_{new}	**% Decline**
1	72.50	63.76	12.06
2	73.93	63.70	13.84
3	71.38	62.51	12.43
4	73.34	64.93	11.47
5	69.35	61.63	11.13
6	71.41	62.45	12.56
7	74.01	65.46	11.55
8	71.20	62.24	12.58
9	72.00	63.44	11.90
10	70.94	62.34	12.13
Avg.	72.01	63.24	12.16

Table 5.19: The percentage improvement values when node capacities are increased by 10%

Instance	z_{base}	z_{new}	**% Improvement**
1	72.50	72.65	0.21
2	73.93	71.15	-3.76
3	71.38	71.14	-0.35
4	73.34	73.21	-0.18
5	69.35	68.19	-1.67
6	71.41	70.89	-0.73
7	74.01	73.91	-0.14
8	71.20	70.28	-1.28
9	72.00	71.51	-0.69
10	70.94	71.80	1.22
Avg.	72.01	71.47	-0.74

5.3.3.4 *What is the effect of increasing the capacities of the transshipment nodes?*

In order to measure the effect of increasing the capacities of the transshipment nodes, node capacities are increased by 10%. z_{base}, z_{new} and percentage improvement and decline values are reported in Table 5.19 for every instance. We see that a capacity increase behaves differently in every instance. In instances 1 and 10, the capacity increase slightly increases the performance as expected, whereas in the remaining instances, the performance decreases. These unexpected results may be due to the convergence of the GA. New instances may need more iterations to properly converge. Overall performance decreases by 0.74% on average when transshipment capacities are increased by 10%. Further experiments with different capacity increase or decrease amounts are needed to see the effect clearly.

5.3.3.5 *What is the effect of increasing demand amounts?*

The effect of increasing the demand amounts is observed by increasing the demand of every demand node by 10%. With the new demand values, infrastructures 1 and

Table 5.20: The percentage improvement values when demands are increased by 10%

Instance	z_{base}	z_{new}	% Improvement
1	72.50	75.39	3.99
2	73.93	74.19	0.35
3	71.38	74.21	3.95
4	73.34	75.15	2.46
5	69.35	73.09	5.40
6	71.41	72.83	1.99
7	74.01	76.41	3.25
8	71.20	73.94	3.85
9	72.00	72.24	0.33
10	70.94	74.46	4.96
Avg.	72.01	74.19	3.05

2 can satisfy their total demand before the disaster while infrastructure 3 can satisfy 98% of its total demand. That is, $fDF_m = \sum_{i \in D_m} d_{im}$ for $m = 1, 2$ and $fDF_3 = 0.98 \sum_{i \in D_3} d_{i3}$ and the heuristic aims to reach to these service levels after the disaster. The baseline results z_{base}, the average expected objective values, z_{new}, and percentage improvement over the baseline instances are reported in Table 5.20. It is observed that the expected objective values increase as there is more demand to satisfy and the heuristic is able to propose good solutions when demand is increased. With a 10% increase in the demand values, demand satisfaction increases by 3.05% on average.

5.3.4 Stochasticity-Related Questions

In this section, the Value of the Stochastic Solution (VSS) and the Expected Value of Perfect Information (EVPI) measures are shared.

5.3.4.1 *What is the value of the stochastic solution?*

For the Value of the Stochastic Solution (VSS) calculation, the Expected Value Problem (EVP) should be defined first. When creating 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. VSS is normally calculated by using equation (5.7) as a percentage where z^*_{SP} is the optimal objective function value of the SIIRR-SP and z^*_{EEV} is the optimal expected result of using the EVP solution.

$$VSS = \frac{z^*_{SP} - z^*_{EEV}}{z^*_{EEV}} \tag{5.7}$$

We present the lower and upper bounds on the VSS values in this study, as it is computationally exhaustive to find the z^*_{SP} and z^*_{EEV} values. EVP of the instances are solved using the SIIRR-SP (with a 3-hour time limit) with the expected value scenario only, 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 on the VSS value is calculated employing equation (5.8) where z_H denotes the baseline heuristic results obtained by running the GA and FRESH. Since the optimal solution of the SIIRR-SP is underestimated and z^*_{EEV} is overestimated by its best bound, the following equation represents an lower bound on the VSS value.

$$VSS_{LB} = \frac{\max\{0, z_H - z_{B,EEV}\}}{z_{B,EEV}} \tag{5.8}$$

Likewise, an upper bound on the actual VSS is calculated as a percentage using equation (5.9) where z_B is the SIIRR-SP best bound calculated after 3 days of run time. As the exact value of the SIIRR-SP is not known and it is overestimated by z_B, and z^*_{EEV} is underestimated by z_{EEV}, equation (5.9) represents an upper bound on the

Table 5.21: Lower and upper bounds on the value of the stochastic solution

Instance	z_{EEV}	$z_{B,EEV}$	**Optimality Gap (%)**	$VSS_{LB}(\%)$	$VSS_{UB}(\%)$
1	76.12	76.16	0.00	0.00	3.79
2	75.11	75.13	0.00	0.00	7.50
3	73.79	73.81	0.01	0.00	5.24
4	74.61	74.64	0.00	0.00	6.50
5	71.44	71.45	0.02	0.00	6.55
6	72.55	72.58	0.00	0.00	7.94
7	75.82	75.91	0.00	0.00	5.80
8	75.15	75.15	0.00	0.00	4.99
9	72.25	72.34	0.01	0.00	7.65
10	74.08	74.17	0.02	0.00	7.55
Avg.	74.09	74.13	0.01	0.00	6.35

actual VSS value.

$$VSS_{UB} = \frac{z_B - z_{EEV}}{z_{EEV}} \tag{5.9}$$

The lower and upper bounds on the VSS values for every instance are given in Table 5.21, together with the z_{EEV} and $z_{B,EEV}$ values and the optimality gaps. 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 the 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.

5.3.4.2 *What is the expected value of perfect information?*

EVPI is the measure demonstrating the importance of the information revealing after the disaster. It is calculated as the difference between the case where we have perfect information about the second stage of the problem before the disaster, and the optimal two-stage SIIRR-SP solution z^*_{SP}, using equation (5.10) where z^*_s is the optimal objective function value of scenario s in the perfect information case.

$$EVPI = \frac{z^*_{PI} - z^*_{SP}}{z^*_{SP}} = \frac{\sum_{s \in S} P_s z^*_s - z^*_{SP}}{z^*_{SP}} \tag{5.10}$$

As the optimal solution of the SIIRR-SP is not known, a lower bound on the EVPI is calculated here as a percentage by using the best bound of the SIIRR-SP after running for 3 days (denoted by z_B), by employing equation (5.11), where z_s is the objective function value of scenario s in the perfect information case. The z_s values are calculated for each scenario by running the SIIRR-SP (with a 3-hour time limit) model with the corresponding scenario. Since z^*_s values are underestimated and the optimal solution of the SIIRR-SP is overestimated by z_B, the EVPI value calculated by equation (5.11) is a lower bound on the actual EVPI value.

$$EVPI_{LB} = \frac{\max\{0, z_{PI} - z_B\}}{z_B} = \frac{\max\left\{0, \sum_{s \in S} P_s z_s - z_B\right\}}{z_B} \tag{5.11}$$

Similarly, an upper bound on the EVPI is calculated as a percentage by employing equation (5.12), 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 with the corresponding scenario and z_H values are the baseline heuristic results obtained by running the GA and FRESH. As z^*_{PI} is overestimated and the optimal solution of the SIIRR-SP is underestimated by z_H, equation (5.12) represents an upper bound on the EVPI value.

$$EVPI_{UB} = \frac{z_{B,PI} - z_H}{z_H} = \frac{\sum_{s \in S} P_s z_{B,s} - z_H}{z_H} \tag{5.12}$$

Table 5.22: Lower and upper bounds on the expected value of perfect information

Instance	z_{PI}	$z_{B,PI}$	**Opt. Gap (%)**	$EVPI_{LB}$ **(%)**	$EVPI_{UB}$ **(%)**
1	79.44	79.51	0.00	0.56	9.68
2	81.14	81.25	0.00	0.49	9.91
3	78.08	78.13	0.01	0.54	9.46
4	80.05	80.13	0.00	0.75	9.26
5	76.41	76.46	0.02	0.38	10.26
6	78.62	78.74	0.00	0.40	10.25
7	80.68	80.72	0.00	0.57	9.08
8	79.31	79.41	0.00	0.53	11.53
9	78.07	78.16	0.01	0.37	8.55
10	80.00	80.08	0.02	0.42	12.88
Avg.	79.18	79.26	0.01	0.50	10.08

The lower and upper bounds on the EVPI values can be seen in Table 5.22 with the expected optimality gaps for every instance. For all of the instances, EVPI lower bounds are lower than 1% and the average is 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 the 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 worth the effort considering the lower bounds on the EVPI values.

CHAPTER 6

CONCLUSIONS

In recent decades, humanity has experienced numerous disasters, increasing in number and intensity. These disasters have impacted the interdependent functioning of the infrastructures in the affected areas. Interdependency is rooted in the infrastructures as their operations are mutually dependent on one another. Furthermore, there are 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, we have defined the Stochastic Interdependent Infrastructure Reinforcement and Repair (SIIRR) Problem in this thesis. In the SIIRR problem, infrastructures are modeled as layered networks and the uncertainty arises from which segments of the infrastructures will be damaged by the disaster. We have formulated a two-stage stochastic integer program for the solution of the SIIRR problem. In this model, the subset of arcs that should be reinforced before the disaster are determined in the first stage, while in the second stage, repair schedules on each network and the service flow on each arc are determined. The objective of the SIIRR problem is to maximize the services provided by the infrastructures after the disaster over a fixed planning horizon.

Solving the two-stage stochastic program for the SIIRR problem is computationally exhaustive. For the instances in our computational experiments, even the root node of the branch-and-bound tree cannot be solved under one week of computation time. To overcome this challenge, 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. FRESH iteratively fills and improves the repair schedule, and optimizes the flow on the infrastructures until the repair schedule can no longer be improved.

The proposed heuristic approach is able to find high-quality solutions in a reasonable amount of run times. Our experiments show that the combination of GA and FRESH is able to solve realistically-sized instances within minutes, whereas CPLEX is unable to find an optimal solution in multiple days. Furthermore, the optimality gap of our heuristic solutions are within acceptable ranges. Many valuable research questions can be answered with the help of the proposed heuristic. For instance, in our computational experiments, we have observed substantial improvements in service delivery when reinforcement is involved. We were also able to quantify the impact of changing the budget levels and repair times on the timeliness of service delivery. Moreover, we have approximated the value of the stochastic solution and the expected value of perfect information measures.

The devised heuristic can be easily applied on different infrastructures, even virtual ones, such as the internet. In that case, 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. Moreover, 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 study, 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. Partial damage may be exemplified as a road segment having one out of two lanes functioning after the disaster. 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. To incorporate this, scenario-dependent repair times may be estimated and used in the SIIRR-SP model as well as the heuristic method. In the devised heuristic, GA employs one representative scenario to reduce the computational burden. The number of representative scenarios may be increased in number 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.

From a different perspective, reinforcement budget may be considered as a second objective to the SIIRR problem. Although it may be computationally long, a Pareto frontier may be drawn for the efficient solutions considering both objectives. In that case, decision makers can decide on the performance level increase in the infrastructures and the reinforcement budget at the same time.

Moreover, aftershocks in the disasters may be incorporated into the problem. The resulting mathematical model would be a multi-stage stochastic program. Likewise, 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

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

C. C. Ang. Optimized recovery of damaged electrical power grids. Master's thesis, Naval Postgraduate School, Monterey, California, 2006.

E. Aslan and M. Çelik. Pre-positioning of relief items under road/facility vulnerability with concurrent restoration and relief transportation. *IISE Transactions*, 51(8):847–868, 2019.

I. Averbakh. Emergency path restoration problems. *Discrete Optimization*, 9(1): 58–64, 2012.

I. Averbakh and J. Pereira. The flowtime network construction problem. *IIE Transactions*, 44:681–694, 2012.

M. Baxter, T. Elgindy, A. T. Ernst, T. Kalinowski, and M. W. Savelsbergh. Incremental network design with shortest paths. *European Journal of Operational Research*, 238(3):675–684, 2014.

D. Beasley, D. Bull, and R. Martin. An overview of genetic algorithms: Part 1, fundamentals. *University Computing*, 15(2):58–69, 1993.

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

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

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

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

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

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

Y. Fang and E. Zio. An adaptive robust framework for the optimization of the resilience of interdependent infrastructures under natural hazards. *European Journal of Operational Research*, 2019.

R. Faturechi and E. Miller-Hooks. Measuring the performance of transportation infrastructure systems in disasters: A comprehensive review. *Journal of Infrastructure Systems*, 21(1):04014025, 2015.

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

D. Guha-Sapir, P. Hoyois, R. Below, and A. Vanderveken. Annual disaster statistical review 2015: The numbers and trends. Technical report, Centre for Research on the Epidemiology of Disasters, 2016.

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

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

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

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

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

E. E. Lee, J. E. Mitchell, and W. A. Wallace. Restoration of services in interdependent infrastructure systems: A network flows approach. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 37(6):1303–1317, 2007.

E. E. Lee, J. E. Mitchell, and W. A. Wallace. *Network Flow Approaches for Analyzing and Managing Disruptions to Interdependent Infrastructure Systems*, pages 1–9. American Cancer Society, 2009.

J. Lee and H. R. Rao. 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*, pages 155–163, Philadelphia, USA, 2009. Digital Government Society of North America.

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

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

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

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

S. G. Nurre, B. Cavdaroglu, J. E. Mitchell, T. C. Sharkey, and W. A. Wallace. Restoring infrastructure systems: An integrated network design and scheduling (INDS) problem. *European Journal of Operational Research*, 223:794–806, 2012.

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

M. Ouyang and Y. Fang. A mathematical framework to optimize critical infrastructure resilience against intentional attacks. *Computer-Aided Civil and Infrastructure Engineering*, 2017.

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

S. Peeta, F. S. Salman, D. Gunnec, and K. Viswanath. Pre-disaster investment decisions for strengthening a highway network. *Computers & Operations Research*, 37 (10):1708–1719, 2010.

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

K. Ransikarbum and S. J. Mason. Goal programming-based post-disaster decision making for integrated relief distribution and early-stage network restoration. *International Journal of Production Economics*, 182:324 – 341, 2016.

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

T. Sato and K. Ichii. Optimization of post-earthquake restoration of lifeline networks using genetic algorithms. *Doboku Gakkai Ronbunshu*, 1996(537):245–256, 1996.

J. Schroeder, A. P. Guedes, and E. P. DuarteJr. Computing the minimum cut and maximum flow of undirected graphs. Technical Report 3, Federal University of Parana Department of Informatics, 2004.

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

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

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

N. Xu, S. D. Guikema, R. A. Davidson, L. K. Nozick, Z. Çağnan, and K. Vaziri. Optimizing scheduling of post-earthquake electric power restoration tasks. *Earthquake Engineering & Structural Dynamics*, 36(2):265–284, 2007.

N. Yaghmaei, R. Below, and V. Wathelet. 2018 annual disaster statistics report. Technical report, Centre for Research on the Epidemiology of Disasters, 2019.

Appendix A

PARAMETERS OF THE SIIRR PROBLEM INSTANCES USED IN COMPUTATIONAL EXPERIMENTS

Table A.1: Cost of reinforcement, repair time and flow capacity of arcs

(i, j)	c_{ij1}	p_{ij1}	ca_{ij1}	c_{ij2}	p_{ij2}	ca_{ij2}	c_{ij3}	p_{ij3}	ca_{ij3}
$(1, 2)$	2	1	138	3	2	97	3	2	97
$(1, 3)$	3	2	109	3	1	326	2	3	166
$(2, 1)$	1	3	178	1	2	257	1	2	259
$(2, 6)$	3	2	19	3	1	250	3	2	207
$(3, 1)$	1	3	133	3	3	222	1	2	337
$(3, 4)$	2	3	143	2	2	60	2	3	19
$(3, 12)$	3	3	68	2	1	18	1	3	260
$(4, 3)$	3	1	253	2	2	104	1	2	176
$(4, 5)$	1	2	309	2	2	261	2	3	19
$(4, 11)$	2	2	96	3	3	169	2	1	83
$(5, 4)$	1	1	99	3	3	283	1	3	163
$(5, 6)$	1	1	11	3	3	47	2	1	337
$(5, 9)$	2	2	376	3	3	397	3	1	126
$(6, 2)$	2	3	307	2	2	383	3	1	53
$(6, 5)$	2	1	198	3	2	354	3	1	206
$(6, 8)$	1	1	153	3	2	338	2	1	160
$(7, 8)$	3	2	386	1	3	45	3	3	123
$(7, 18)$	3	2	182	2	3	176	2	3	184
$(8, 6)$	3	1	268	2	3	76	1	3	25
$(8, 7)$	2	1	400	2	3	179	3	1	98

Table A.1: Cost of reinforcement, repair time and flow capacity of arcs - Continued

(i,j)	c_{ij1}	p_{ij1}	ca_{ij1}	c_{ij2}	p_{ij2}	ca_{ij2}	c_{ij3}	p_{ij3}	ca_{ij3}
$(8,9)$	3	2	170	2	3	304	1	3	166
$(8,16)$	2	3	200	1	2	36	2	2	26
$(9,5)$	1	1	312	1	2	116	1	3	299
$(9,8)$	2	1	247	3	3	27	1	3	184
$(9,10)$	1	1	242	2	3	337	3	3	37
$(10,9)$	2	2	280	1	3	288	2	3	234
$(10,11)$	3	2	380	2	1	181	1	2	276
$(10,15)$	1	2	356	2	3	178	3	3	297
$(10,16)$	2	3	106	1	2	287	1	3	329
$(10,17)$	2	3	129	3	1	274	3	1	12
$(11,4)$	1	3	31	3	2	294	3	1	329
$(11,10)$	3	1	136	1	1	345	2	2	16
$(11,12)$	2	3	368	2	1	361	2	1	171
$(11,14)$	1	3	102	3	3	46	2	3	180
$(12,3)$	2	2	178	2	2	113	1	2	276
$(12,11)$	3	1	77	3	3	182	3	3	187
$(12,13)$	3	1	148	3	2	372	3	3	340
$(13,12)$	1	1	47	3	1	23	1	3	328
$(13,24)$	2	2	230	2	2	347	2	3	83
$(14,11)$	1	3	281	1	1	141	2	3	285
$(14,15)$	3	3	7	2	3	385	3	3	2
$(14,23)$	1	3	146	1	1	390	2	3	181
$(15,10)$	3	3	349	1	2	56	1	1	189
$(15,14)$	2	2	238	2	2	224	1	2	94
$(15,19)$	1	3	230	3	2	347	1	1	75
$(15,22)$	2	3	124	3	3	306	3	1	111
$(16,8)$	3	1	298	3	1	350	3	2	67
$(16,10)$	2	3	221	3	1	28	3	2	98
$(16,17)$	3	1	230	3	3	246	2	1	319

Table A.1: Cost of reinforcement, repair time and flow capacity of arcs - Continued

(i,j)	c_{ij1}	p_{ij1}	ca_{ij1}	c_{ij2}	p_{ij2}	ca_{ij2}	c_{ij3}	p_{ij3}	ca_{ij3}
$(16,18)$	3	1	265	2	3	325	3	3	240
$(17,10)$	2	1	316	1	2	315	2	3	174
$(17,16)$	2	3	247	3	3	25	1	1	325
$(17,19)$	2	2	49	2	3	60	3	3	261
$(18,7)$	3	3	151	2	3	108	2	2	118
$(18,16)$	2	3	398	2	3	272	1	2	179
$(18,20)$	2	3	38	3	1	53	3	2	178
$(19,15)$	1	3	337	3	3	208	1	1	356
$(19,17)$	1	3	368	3	1	323	2	1	268
$(19,20)$	1	3	203	3	1	7	2	2	273
$(20,18)$	1	1	283	2	1	257	3	3	304
$(20,19)$	3	3	216	3	1	254	1	3	197
$(20,21)$	3	2	37	1	1	274	3	3	104
$(20,22)$	3	1	244	1	3	75	1	1	15
$(21,20)$	1	1	65	3	1	373	1	2	276
$(21,22)$	3	2	381	3	3	92	1	3	43
$(21,24)$	1	2	219	1	3	95	3	3	386
$(22,15)$	3	1	344	2	1	296	1	3	196
$(22,20)$	1	1	165	2	3	209	2	3	398
$(22,21)$	3	2	302	3	2	385	2	3	253
$(22,23)$	2	2	353	2	2	22	1	2	78
$(23,14)$	1	1	256	3	2	119	3	1	12
$(23,22)$	1	1	369	3	2	164	3	2	97
$(23,24)$	3	3	340	2	2	21	3	1	83
$(24,13)$	3	2	255	3	2	251	3	3	8
$(24,21)$	2	3	143	1	2	152	3	1	73
$(24,23)$	2	3	66	2	3	380	2	3	178

Table A.2: Supply and demand values of nodes and capacity of transshipment nodes

i	s_{i1}	d_{i1}	cn_{i1}	s_{i2}	d_{i2}	cn_{i2}	s_{i3}	d_{i3}	cn_{i3}
1	-	40	-	174	-	-	-	33	-
2	-	52	-	265	-	-	-	52	-
3	-	20	-	-	39	-	-	-	49
4	-	84	-	-	50	-	-	-	27
5	312	-	-	-	45	-	-	21	-
6	398	-	-	-	92	-	-	-	53
7	-	38	-	-	12	-	-	15	-
8	-	46	-	-	-	55	-	-	87
9	-	15	-	31	-	-	-	19	-
10	84	-	-	-	-	58	335	-	-
11	-	12	-	-	17	-	-	-	54
12	-	11	-	-	14	-	-	43	-
13	-	-	27	-	-	31	-	12	-
14	-	-	64	-	25	-	-	-	49
15	-	14	-	-	11	-	-	-	40
16	-	62	-	-	10	-	-	21	-
17	-	-	79	-	33	-	-	50	-
18	-	15	-	191	-	-	-	53	-
19	-	26	-	185	-	-	-	66	-
20	-	10	-	-	19	-	-	39	-
21	-	35	-	-	66	-	390	-	-
22	-	11	-	-	45	-	-	61	-
23	-	22	-	-	-	48	-	-	25
24	-	9	-	-	42	-	-	41	-

Appendix B

OBJECTIVE VALUES FOR THE INDIVIDUAL RUNS OF THE PROPOSED HEURISTIC

Table B.1: The expected objective function values of the instances for five runs

Instance	**Run 1**	**Run 2**	**Run 3**	**Run 4**	**Run 5**	z_H
1	72.43	73.63	73.16	70.99	72.28	72.50
2	73.75	75.09	73.50	74.21	73.09	73.93
3	71.31	71.03	71.23	72.22	71.12	71.38
4	73.23	73.65	72.78	73.34	73.69	73.34
5	69.49	69.66	68.84	69.45	69.29	69.35
6	71.72	70.95	70.97	71.29	72.15	71.41
7	74.91	73.60	73.58	74.59	73.35	74.01
8	70.24	71.68	71.13	71.43	71.50	71.20
9	71.84	72.18	72.36	72.12	71.52	72.00
10	72.03	72.33	69.10	71.35	69.91	70.94