A SYNCRONIZED ROUTING PROBLEM FOR RESTORING INTER-DEPENDENT INFRASTRUCTURE NETWORKS

A Thesis

by

Büşra Sevindik

Submitted to the
Graduate School of Sciences and Engineering
In Partial Fulfillment of the Requirements for
the Degree of

Master of Science

in the
Department of Industrial Engineering

Özyeğin University
May 2018

A SYNCRONIZED ROUTING PROBLEM FOR RESTORING INTER-DEPENDENT INFRASTRUCTURE NETWORKS

Approved by:

Assistant Professor Dilek Günneç,
Advisor,
Department of Industrial Engineering,
Özyeğin University

Associate Professor Ali Ekici,
Department of Industrial Engineering,
Özyeğin University

Associate Professor Burcu Balçık,
Advisor,
Department of Industrial Engineering,
Özyeğin University

Assistant Professor İhsan Yanıkoğlu,
Department of Industrial Engineering,
Özyeğin University

Date Approved: 15 May 2018

Assistant Professor Burak Çavdaroğlu,
Department of Industrial Engineering,
Kadir Has University

To my family

ABSTRACT

Disasters may cause significant damages in lifeline infrastructure systems (such as gas, power, water) and lead to long-lasting failures. It is important to repair the damaged components and restore the affected infrastructures quickly. Since different lifeline infrastructure systems depend on each other, considering the inter-dependencies among different networks during repair planning can speed up the recovery process. In this thesis, we focus on developing practical methods to support planning repair operations for two inter-dependent infrastructure networks by considering the inter-dependencies within and between these networks. Specifically, we assume that repairing a damaged component may not be sufficient for making the component functional due to network dependencies. We consider multiple repair teams, each of which can repair the damaged components of one type of infrastructure, and formulate a coordinated repair scheduling problem, which determines a repair schedule for each repair team to minimize the total time for making all nodes functional. To solve this problem, we propose two alternative constructive heuristics, which employ different strategies to prioritize the visit of the damaged nodes based on their dependency status. We also apply local search procedures to improve the solutions attained by the constructive heuristics. We present computational results to evaluate the performance of the proposed heuristics. The results show that our heuristics lead to high quality solutions and can be used to make repair plans quickly in the post-disaster environment.

ÖZETÇE

Doğal felaketler gaz, elektrik, su ve telekomünikasyon gibi temel altyapı sistemlerine önemli hasarlar verebilir ve uzun süre işlevselliklerini kaybetmelerine yol açabilir. Zarar görmüş altyapı bileşenlerinin tamir edilmesi ve etkilenen altyapı sistemlerinin bir an önce işlevsel hale gelmesi mühimdir. Çeşitli temel altyapı sistemlerinin birbirine bağımlı olmalarından dolayı, tamirat süreci esnasında çeşitli ağların arasında mevcut olan birbirine bağımlılığın göz önünde bulundurulması hizmetlerin yeniden tesisini hızlandırmaktadır. Bu çalışmada, kendi içlerinde ve aralarında mevcut bulunan birbirine bağımlılıklarını göz önünde bulundurmak kaydı ile, iki adet birbirine bağımlı altyapı ağlarının tamirat süreci planlamalarına destek olacak pratik metotlar geliştirme üzerine yoğunlaşılmaktadır. Özel olarak, bileşenlerin tamiratının tamamlanmasının, ağlar arasındaki bağımlılıklar nedeniyle bileşenlerin fonksiyonel olmasını sağlamaya yetmeyeceği varsayılmaktadır. Her biri sadece bir tip altyapı sistemini tamir etmeye muktedir birden fazla tamirat ekiplerini düşünerek, tüm düğümleri fonksiyonel kılacak toplam süreyi en kısa hale getirecek tamirat ekiplerinin tamirat programlarına karar verecek koordineli bir tamirat planlama problemi üretilmektedir. Bu problemi çözmek için, bağımlılık durumlarına dayanarak ziyaret edilecek düğümlerin önceliklendirilmesini sağlayacak stratejiler türetecek iki adet alternatif yapıcı sezgisel metot önerilmektedir. İlave olarak, yapıcı sezgisel metotlardan elde edilen sonuçları iyileştirmek için lokal arama prosedürleri uygulanmaktadır. Tanıtılan sezgisel metotların performanslarını değerlendirmek maksadıyla elde edilen işlemsel sonuçlar sunulmaktadır. Elde edilen sonuçlar, arz edilen sezgisel metotların yüksek kalitede sonuçlar verdiğini ve afet sonrası ortamda ivedi tamirat planları oluşturulmasında kullanılabileceğini göstermektedir.

ACKNOWLEDGEMENTS

In the graduate school, I have had the privilege and honor of working with my research advisors, Assoc. Prof. Burcu Balçık and Asst. Prof. Dilek Günneç. I have learned so much from their impressive technical knowledge, research philosophy, and incredible humility, generosity, and kindness. I thank them for patiently listening to me whether the subject matter was technical or not and sharing their insight. I consider myself extremely lucky to have known such technical pioneers and wonderful people.

I would like to thank my office friends, Cem Deniz Cağlar Bozkır, Görkem Emirhüseyinoğlu, Hamed Shourabizadeh and Tonguç Yavuz, for their valuable friendships.

Special thanks to my best friend and my husband, Ömer Hikmet Sevindik for always being there for me and to my mother Didem, my father Levent and my lovely sister Beyza Uydaşoğlu for their invaluable support. I am so grateful for their encouragements and sacrifices during my educational life.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Devastating effects of a disaster last for a long period of time in the affected regions. It can take months before people can reach a steady access to functional infrastructure systems such as power, water, gas or telecommunication. For instance, hurricane Maria that occurred in September 2017 knocked out the power to Puerto Rico; although power and water were restored in the majority of the island's urban areas within a few weeks, more than 100,000 thousand residents in the rural areas remained in the dark for months [3].

An important factor, which may affect the length of recovery of the lifeline infrastructure systems, is the inter-dependencies among the affected networks; that is, the components of different infrastructures may depend on each other to function properly. Therefore, a damaged component in an infrastructure network may lead to failures in other infrastructures which are not necessarily damaged themselves. The inter-dependencies among infrastructure systems can appear in different ways. For instance, most of the lifeline infrastructure systems (such as gas, water and telecommunications) count on a working power system. Moreover, water is critical for cooling electricity, telecommunication and oil systems (see Figure 1) and gas is required to provide fuel with power generators and heat with telecommunication systems. In power systems, gas-fired generating units are commonly used in the last few decades because they are cheaper and more environment-friendly than fossil generating units [4].

The dependencies between two infrastructure networks can be bidirectional. For example, compressors, storage and control systems in gas networks require power to

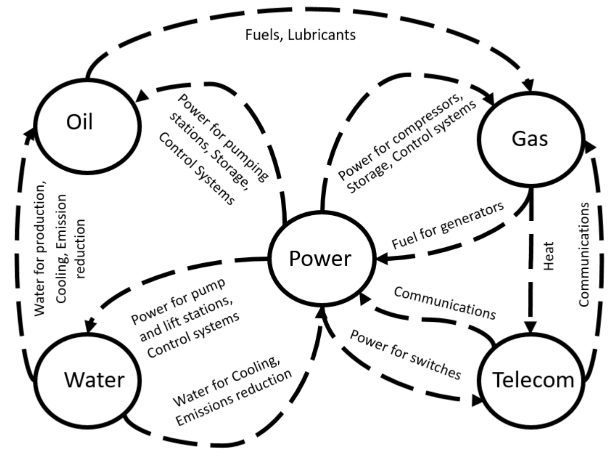

Figure 1: Inter-dependencies of Infrastructures (Adapted from Rinaldi et al. [1])

operate, while a combined-cycle power plant uses gas to produce electricity [1]. An example of a cumulative failure due to such dependencies is the Blackout in Italy in 2003 [5]; in this case, large loss in power systems led to failure in Internet communication network, causing further break down in power stations due to problems in telecommunication systems.

Besides, components in gas network are illustrated in Figure 2 in order to present the dependency in infrastructures. There are 7 different components in the gas infrastructure: gas wells, trams miss pipelines, underground storages, valves, distribution pipelines, compressors, gas fired units. For instance, damage in a underground storage may affect a compressor or a gas-fired unit directly.

In the aftermath of a disaster, it is important to restore lifeline infrastructure systems quickly. In this study, we consider a setting in which repair teams are dispatched to repair/restore damaged (i.e., non-functional) components of the lifeline networks affected by a disaster. We assume that when a component in a lifeline network is damaged, it is not operational and further it affects nodes which are directly connected to it. They become non-operational as well. However, nodes that are indirectly connected to this node over the network are not affected. In other words,

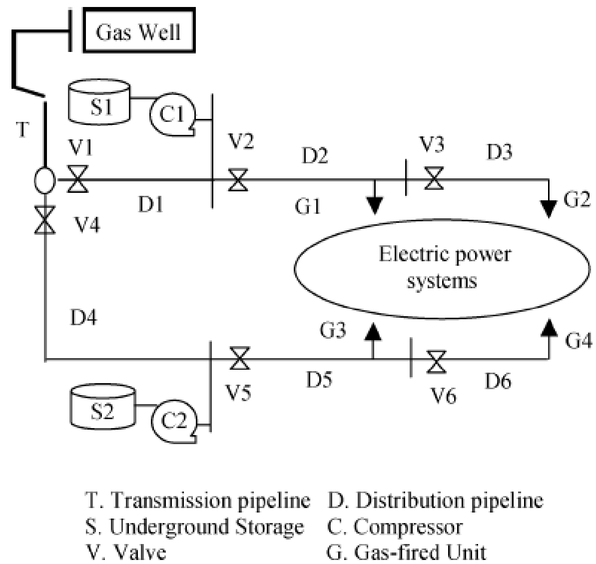

Figure 2: Example of Gas Flow Infrastructure (from Shahidehpour et al. [2])

we assume that two-stage cascading failure [21] occurs when restoring infrastructures. Also, one-to-many inter-dependency type is assumed, in which a damaged node causes failures of all the nodes that it affects. Furthermore, if this damaged component has dependent components in the same network or other networks, those dependent components, which may not be damaged themselves, remain in non-operational status until the repair process of that damaged components is completed (i.e., the damaged component becomes functional). Therefore, identifying the critical damaged nodes in the network, whose functionality affects the completion time of the restoration process, and developing repair schedules by prioritizing such critical nodes is essential to prevent delays in recovery operations.

Given the inter-dependencies in the lifeline networks, scheduling repairs for the damaged components may significantly affect the total duration of the restoration process. Therefore, recovery operations must be planned carefully while considering existing inter-dependencies and available time and resources. However, in practice, restoration operations of different lifeline infrastructures may be carried out independently; repair teams may not be coordinated. For instance, AYKOME, which is the infrastructure coordination agency in Turkey, has different departments that are responsible for restoring power and gas networks, and these departments do not perform joint planning after a disaster [6].

If repair schedules for different lifeline networks are not planned in a coordinated way, the effects of inter-dependencies can be overlooked, which may lead to delays in the operational time of the overall system. In this study, we consider multiple lifeline networks and explore the benefits of a centralized decision making process for scheduling repair operations. We define this problem as Syncronized Routing Problem for Restoring Inter-dependent Infrastructure Networks (SRPRIIN). Specifically, we address two inter-dependent networks (such as power and gas), both of which are affected by a disaster. For each network, there are multiple repair teams, which visit the nodes with damaged components and make the repairs. We assume that repair schedules are developed at the beginning of planning horizon based on estimated repair and travel times in the networks. Each team originates from a depot and returns to the same depot once all repairs are completed. The depots associated with different networks are different. We develop a mathematical model, which determines a repair schedule for each team to minimize the sum of the completion times of all nodes in both network, while completion time is defined as the time that a node becomes functional.

To solve the SRPRIIN model, we present alternative constructive heuristics, which differ in terms of prioritizing visiting damaged components based on their dependency status. We apply local search methods to improve the solutions developed by constructive methods. Our solution approach provides rapid and high-quality solutions and is easy-to-implement in the post-disaster environment. We present computational results, which compare alternative heuristic methods. Moreover, we present results, which illustrate the benefits of coordinated planning.

In Section 2, we review the related literature. In Section 3, we describe the problem in detail, and present our mathematical model. In Section 4, we introduce our solution methods. In Section 5, we present our numerical results and analysis. We present our conclusions and future work in Section 6.

CHAPTER II

LITERATURE REVIEW

SRPRIIN studied in this paper is related with two major streams of literature. The first stream is related with the recovery of inter-dependent infrastructure systems in humanitarian operations and the second stream is related with the synchronized vehicle routing problems.

2.1 Inter-dependent Infrastructure Restoration in Humanitarian Operations

For the reasons we mention in the previous chapter, restoration of infrastructure systems is a critical issue after disaster. Therefore, it has been studied to a large extent in the past decade. For a recent review on network restoration and repair problems in post disaster stage, we refer the reader to Celik's review paper [7] which categorized related studies into five parts; network restorations as road restoration and rehabilitation, infrastructure restoration, network construction, snow removal problems, debris clearance and removal.

Celik [7] presents objective functions and solution methods of papers in this study area. Approximately 40% of the papers in this review aims to minimize related costs. Approximately 20% of the papers aims to minimize completion time similar to our study. Considering all restoration categories, most of the problems are solved using exact methods and meta heuristics. Approximately 30% of the related literature use constructive heuristics similar to our study.

Although there are many studies on infrastructures and their restoration, most of the literature focus on individual systems and neglect their dependency on each other as in Nurre et al. [8] and Hentenryck et al. [9]. While in Hentenryck et al. [9], joint assessment and restoration in power systems based on potential damage scenarios are studied. They use three different algorithms to solve the joint damage assessment and restoration problem. *i)* an online stochastic combinatorial optimization (OSCO) algorithm, *ii)* two-stage approach, which consists of decision on real damage and online optimization to reflect reality, *iii)* simultaneous exploration and restoration.

Models of infrastructure restorations are introduced before the onset of natural disasters while considering uncertainties of natural disasters. Xu et al. [10] and Corin et al. [11] present stochastic models to include related uncertainties in the models, in contrast to Çağnan et al. [12] and Çağnan and Davidson [13]. [10] aims to schedule repairing processes and flow decisions and resource allocation are the main decisions in [11]. Decisions of the [12] and [13] include both flow and facility location for single infrastructure.

Since early 2000s, inter-dependencies have been addressed by a group of studies. In this respect, one of the first studies is the article by Shahidehpour et al. [2] which draw attention to the dependency relationship between power and natural gas systems with a focus on integrated approach of operations and planning, however, not focusing on restoration.

In a review paper, Ouyang and Min [14] emphasize that critical infrastructure systems are inter-dependent and inter-dependency is the key performance in planning, maintenance and emergency decision making processes. They present current models and simulation approaches with proofs from different studies in this context. Different inter-dependency types are emphasized in the definition of inter-dependency. Rinaldi et al. [1] define physical, cyber, geographical and logical inter-dependency types. A physical inter-dependency happens when material output of one infrastructure affects other infrastructures. Cyber inter-dependency uses information transmission between infrastructures. Zimmerman and Rae [15] define functional and spatial inter-dependency types. A functional inter-dependency means necessity of operational state of one infrastructure in order for others to be also operational. Dudenboegger et al. [16], Wallace et al. [17], Zhang et al. [18] present studies that categorizes inter-dependency types.

Furthermore, there are studies that examine the cascading failures of inter-dependent networks. Buldyrev et al. [19] highlight a specific problem of failure in a two inter-dependent network setting in events of correlated degrees. Besides, stage by stage cascading failure is assumed to exist in coupled networks, which have equal number of nodes and bidirectional inter-dependencies. The motivation behind such an assumption is the fact that cascading failures do not advance rapidly as mentioned in Bienstock and Daniel [20]. In Veremyev et al. [21] which is another study on cascading failures, there are two different inter-dependency definitions. First one is defined as one-to-many, in which a damaged node causes failures all the nodes that it affects. Second one inter-dependency is defined as many-to-one, in which a node becomes inoperative provided that all the nodes affecting this node must be damaged. Cascade stage is defined in a similar fashion as our study. In the first stage, there exist a damaged node after a disaster. As the number of stages increases, the number of affected nodes also increases in accordance with the dependency among nodes. In our study, we assume one-to-many inter-dependency definition and two-stage cascade failures. In other words, damaged nodes may only affect nodes that they are connected directly.

In a pioneer work, Lee et al. [22] use an inter-dependent layered network model for restoration of independent systems. In case of a disruption, they consider five types of dependencies (input dependence, mutual dependence, shared dependence, exclusive or dependence, and co-location dependence). They develop a mathematical model with a network flow approach. Considering the repair of only the power system, they can measure service insufficiency during a failure and determine the necessary parts to restore all services. However this model does not provide the information of how and when the infrastructures may be restored. In order to contribute to this model, Çavdaroğlu et al. [23] form a model that involves integrated installation and assignment decisions. A first step in this process is to define the nodes of power and telecommunication infrastructure systems and 2 other infrastructures that are to be restored. In the second step, work groups are assigned to repair these nodes and repair schedules for work groups are created, but this study also focuses on damage and restoration of only the power grid. Coffrin et al. [24] study a similar problem but focuses on creating a realistic representation of a linearized decentralized model for a power system. They include modeling cyclic inter-dependencies and form a good fundamental for inter-dependency studies. More recently, Sharkey et al. [25] consider a centralized planning for the association of restoration decisions as an extension to [23], using damage scenarios. Repair scheduling, flow decisions and resource allocation decisions are studied in both papers. Gonzalez et al. [26] highlight the importance of costs of restoration and study exploiting efficiencies from joint restoration of systems.

In addition, inter-dependent power and natural gas system schedules are optimized from the joint coordinator point of view in Liu et al. [27]. They use Lagrangian relaxation to divide the problem in two sub problems and solve it with many individual constraints of power and gas networks. They aim to minimize the social cost. Social cost refers to total operating cost during power and gas infrastructure scheduling. The aim of this study is to coordinate hourly schedules in order to supply both natural gas and generate power by a single operator. But it does not cover repair scheduling and resource allocation decisions. Necessary schedules are created for a single coordinator to supply inter-dependent infrastructures.

Furthermore, network structure is considered for the inter-dependent infrastructure systems in Gutfraind et al. [28]. They aim to minimize the installation or recovery cost of networks by modeling the effect of inter-dependency on recovery steps. Network structure is taken into account in cost determination. The installation cost of a given node is related to the number of functional neighborhood nodes that are recovered previously. They assume decreasing and convex cost functions. This model is a new discrete optimization problem, which is called Neighbor-Aided Network Installation Problem. The difference of the study is that this paper does not include routing or scheduling and inter-dependency is considered only in the response step.

Complexities of these types of infrastructure restoration problems are discussed in the literature. Hentenryck and Pascal [29] emphasize the necessity of strategic and complex decisions on optimization in times of disasters. Managers who carry out decision making processes in disaster management face numerous problems such as stochastic aspects, complex infrastructures, multiple objectives. In this study, these complex problems and some cases studies are introduced. Based on the work, we also employ specific assumptions. We desire to limit the effect of these complexities in our model which we may encounter during infrastructure restoration processes. For instance, in real life gas and power infrastructures may include more than one type of components such as gas-fired units, compressors, and so on. In this study, we regard all the components as of one same type.

In this study, we use functional inter-dependency definition presented in Zimmerman and Rae [15] and [18]. In other words, infrastructures are necessary for other infrastructures to be functional. We assume an inter-dependency to be divided in two categories: one directional and bidirectional. In one directional -dependency, only one of the infrastructures is dependent on the other one. However, in bidirectional inter-dependency, both infrastructures are dependent on each other to be functional.

Our study differs from literature that is explained above in terms of restoration decisions and problem type. In this study, we examine the inter-dependent infrastructure problem and model the problem from a different perspective. We do not consider the demand of infrastructure components as in a flow network. We define demand as restoration and the state of functionality of the infrastructure components. In the case of a disaster, time is critical and available resources (especially in terms of vehicles and undamaged roads) are limited. For a realistic modeling of restoration efforts, we aim to determine the routes of repair teams in their networks as well as the order of repair that these teams carry out.

2.2 Synchronized Routing Problems

The inter-dependency structure in SRPRIIN resembles a synchronization type of constraint for a vehicle routing problem. Drexl [30] provides a recent literature survey on VRP with multiple synchronizations, i.e., when more than one vehicle may or must be used to fulfill a task. They classify synchronization types as task, operation, movement, load and resource synchronization. Almost all of these synchronizations are based on a restriction that affects the routing problem and that a change in a route has an effect in all other routes. Forest operations can be an example of this synchronization. The process of cutting down trees are implemented using two types of vehicles: harvesters and forwarders. Harvesters cut down the trees, create logs and place in the piles. After harvesters finish their task, forwarders pick up small piles and move them to larger piles near roads where trucks pick them up for transportation to mills or terminals. Both types of vehicles have their own capacity and forwarders are not able to pick up logs if they are not yet created from trees by harvesters. This process is a kind of pick up and delivery problem. However, in SRPRIIN, functional status of another component is required to minimize the objective function, but there is no restriction on the order in which the components can be visited. Therefore, a dependent node may be visited before the node it is dependent on, and wait for an idle period of time in between.

In [30], synchronization types are classified in six categories: *i)* load synchronization *ii)* resource synchronization *iii)* pure spatial operation synchronization *iv)* operation synchronization with precedence *v)* exact operation synchronization *vi)* movement synchronization. Load synchronization is referred to as split delivery VRP in literature. Multiple vehicles visit customers and meet their demand in this problem. Aim of the second synchronization type is meeting all demand in systems with limited time and resource. Scheduling of automated guided vehicles (AGVs) at airports is an example to this synchronization type. AGVs are capable to perform one task at a time, so a request should be delivered accordingly. There are a couple of limited resources such as unloading docks, parking lots and cargo storage areas. The logic behind pure spatial operation synchronization is to supply demand points via n stage distribution networks, not via a one main depot. Task and support vehicle problems are applications of this type of synchronization. Dial-a-ride and pickup and delivery problems are examples of operation synchronization with precedence. Although none of classified synchronization types is an exact match for SRPRIIN, we consider the operations synchronization with soft precedence to be the closest one. SRPRIIN problem is a type of operation synchronization. However, precedence is not necessarily a constraint when considering synchronization. Therefore, we classify the SRPRIIN problem as operation synchronization problem with soft precedence. Because in this type of synchronization, jobs cannot start without the finish of other jobs. In this study, there is no precedence to start restoration. However, when restoration processes are complete, there is precedence if there exists an inter-dependency to restore functionality. Allocation of workers with different qualifications is an example to exact operation synchronization. This type of synchronization may be explained by multiple vehicles being at the same location at the same time period. Finally, movement synchronization type can be illustrated such that scheduling of trucks and drivers depart from the same depot and return to this depot.

In combined VRP and scheduling problems that are mentioned in Bredström and Mikael [31], there are time window and temporal precedence constraints. As in real-life applications such as homecare staff operations; the problem is simply to create daily routes for staff members. Additional constraints in this problem are skills of the personnel, working hours, time window of the customers, work durations and precedence of the tasks. Precedence defines the state of members presence in a given order upon the request of the customer. If we think this system as a network, it is possible to think the house of the customers as the nodes and the staff as vehicles. In addition, while creating a route, the same demand node may have precedence or dependency according to the visit of the vehicles.

Another problem type of synchronized routing problems is the manpower allocation problem. Li et al. [32] discuss the assignment of different team jobs to demand points on scattered areas with specific time periods. In contrast to our study, there are slight differences in terms of inter-dependencies and constraints. We assume consecutive jobs similar to debris clearance. Manpower allocation is very challenging in many staff types such as technical service providers, medical attendants and janitorial service providers. In this model, authors aim to minimize weighted number of workers and total travel time.

The focus of the paper is to adapt synchronized routing concept into a new application. The studies in this field are known as network restoration and response in humanitarian operations and the general aim in these problems is to finish repairing the damaged components in the affected networks in the quickest way possible. In this paper, inter-dependent infrastructure network design is adapted into a synchronized routing problem which is a sub-topic of repairing inter-dependent infrastructure concept. In contrast with the well-known dial-a-ride or pick-up and delivery problems, operational synchronization with node precedence becomes more important in this paper.

In this study, up to our knowledge, we introduce the first VRP problem in a restoration setting with such inter-dependency relationship among infrastructures. The dependency relationship is not a hard constraint for the problem, but affects the objective function value of the mathematical model. Therefore, the inter-dependency is implicitly enforced, meaning that the idle time between the time a node is repaired and the time it becomes functional, would potentially worsen the objective function value.

CHAPTER III

PROBLEM DEFINITION & MATHEMATICAL MODEL

Disasters may affect multiple infrastructure systems simultaneously. For instance, after four months from the hurricane in Puerto Rico in 2017, more than 450,000 residents still do not have access to power [33]. Clean water and food resources are very limited. Full recovery and supply of all resources are estimated to happen over one year period [34].

In this study, we focus on two infrastructure systems; power and gas networks. They are both prone to failure and subject to maintenance as part of post-disaster restoration efforts. The problem setting is as follows: We define the nodes as damaged, inoperative components and functional components. In Figure 3, we have two networks (power and gas) where arrows represent dependencies among nodes. Nodes *p1*, *p2*, *p5* and *g1* are directly affected from the disaster, i.e., will be called "damaged" and require a repair. In the aftermath of a disaster, nodes from both networks are either functional or inoperative. Obviously, the functional nodes do not need any repair as nodes *p4* and *g4* in Figure 3. However, there are some nodes which are inoperative but do not need any repair as nodes *p3*, *g2* and *g3* in Figure 3. This is because their failure is due to failure of another node (from the same or the other network). This is due to inter-dependency of networks. In other words, damaged nodes in the power infrastructure may affect other nodes in the power network. Similarly, damaged nodes in the gas infrastructure may affect others in gas infrastructure. In addition, gas and power networks may affect each other. Necessity of gas-fired units for power generators is an example for mutual inter-dependency. We consider dependency in two different ways: one directional dependency and bidirectional inter-dependency.

In one directional dependency only one infrastructure is affected by the other infrastructure (see Figure 3, dependency between *p5* and *g3*). If both infrastructures are affected by each other, then two infrastructures have bidirectional inter-dependency.

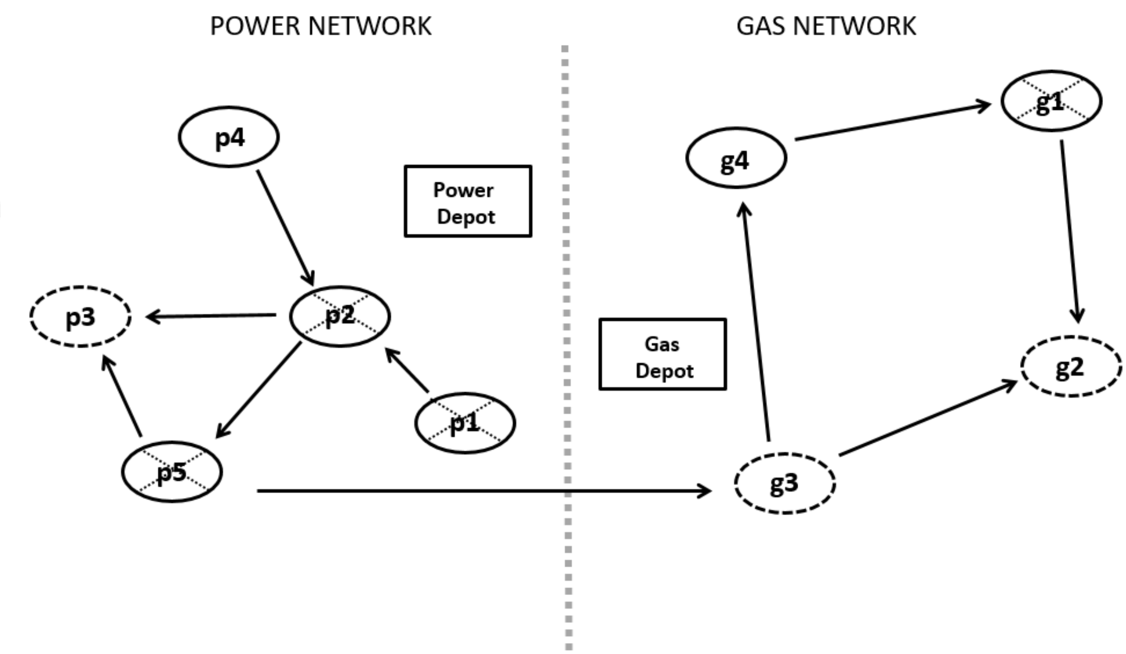

Figure 3: Illustrative example for node types

Our study is motivated by the need for effective recovery of inter-dependent networks. After a natural disaster, each network may carry out restoration by considering only its own system and then may have to wait for other dependent nodes to be functional. However, when we consider this overall state, neglecting inter-dependency may cause time and efficiency problems. For these reasons, our study suggests planning by taking into account the inter-dependencies between networks in the restoration processes. In addition, we define dependencies between nodes as two-stage cascading failures. In other words, a damaged node does not affect a node that is connected to it indirectly, but affects a node with a direct connection. If there are nodes connected to undamaged nodes, these undamaged nodes do not have any impact on the completion time of the nodes that they affect. For instance, node *p5* is damaged node and affects undamaged node *g3* directly. However, it does not affect node *g4*.

This study aims to restore the damaged nodes in inter-dependent power and gas networks after disasters as soon as possible by creating the best route. We assume multiple vehicles for each damaged network, and these vehicles may only restore their own networks. Besides, to focus on a general setting, we assume multiple depots with one depot for each network.

The main goal of the problem is to make all nodes functional again as soon as possible. Completion time of a node is the time that it becomes functional; that is, the node and all the nodes that affect this node are repaired. We define the time a node becomes functional as the completion time of a node. If a node is damaged and there is no dependency to other nodes, the completion time of that node is the total of arrival time of a vehicle to that node and the repair time. If a node is damaged and also dependent to another node, completion time of that node is the maximum of the sum of own arrival and repair time and the sum of arrival and repair time of the node that it depends on. If a node is not damaged but dependent on another node, completion time of that undamaged node is the sum of arrival and repair time of the damaged node that it depends on. To illustrate that, this is a non-trivial problem, we provide the following example in Figure 4.

In Figure 4, there are five power and four gas nodes and one of the gas nodes, *g3* is dependent on one node from power network, *p5*, to function. Nodes *p1*, *p2*, *p5*, *g3* and *g1* are directly affected from the disaster, they are damaged. Nodes *p3* and *g2* do not function only because of their dependencies on nodes *p5*, *g1* and *g3*. Moreover, node *g3* on the gas network is dependent on node *p5* from the power network. Repair times are also assumed to be one hour for each node, and travel times between nodes are assumed to be one hour. There is a depot for each system where the repair teams

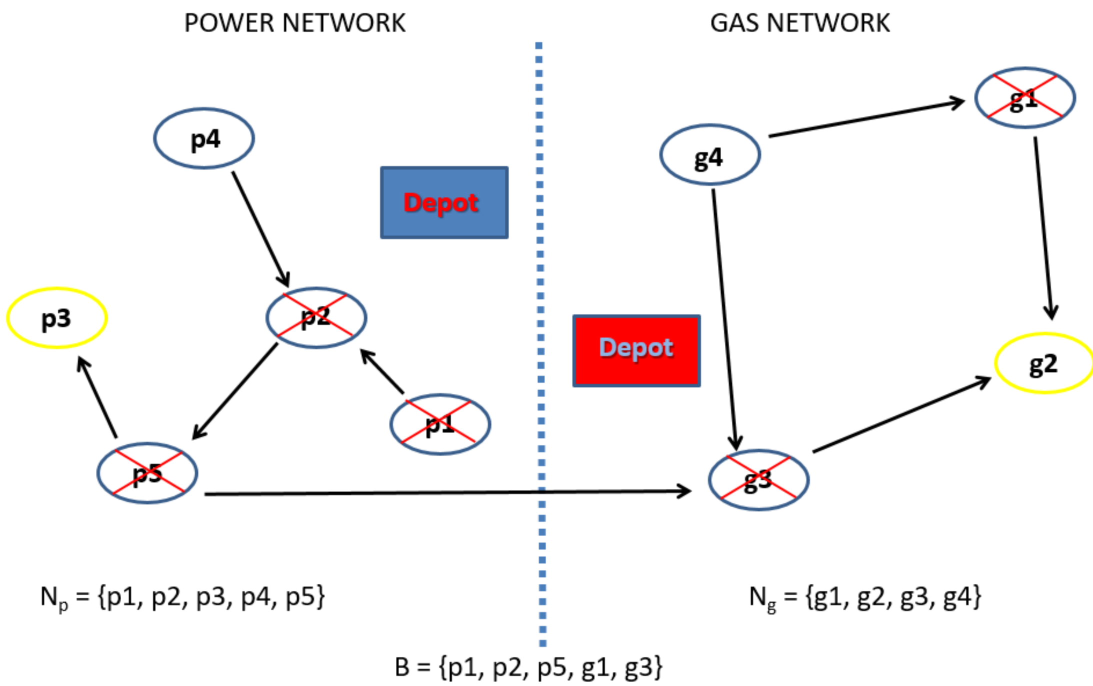

Figure 4: Illustrative example for inter-dependent networks

will depart. Travel times to corresponding depots are assumed to be one and a half hours for nodes *p2*, *p3* and for *g1* and *g3*, and one hour for other nodes. We consider one vehicle for each network.

In Figure 5, we show how the completion time changes when dependency among systems is taken into account. R represents the time node is repaired and c_i is the time node starts service again, i.e., completion time. If infrastructure restoration after a disaster is carried out without paying attention to inter-dependencies, then each separate network would focus on its own damaged nodes and minimize completion time of their nodes. The power network has three nodes to be visited. The optimal route that minimizes the total completion time would be $\{depot,\ p1,\ p2,\ p5\}$ with a total completion time of 18 hours (completion times of power nodes are c_{p1}= 2, c_{p2}= 4, c_{p3}= 6, c_{p4}= 0, c_{p5}= 6 hours respectively). For the gas network, the optimal route would be {*depot*, *g3* and *g1*}. However, since *g3* is dependent on *p5*, the completion time realized would be 6 hours rather than 2 hours. Therefore the actual completion time for node *g3* would be 6 and the total completion time would be 14 hours (completion times of gas nodes are c_{g1}= 4, c_{g2}= 4, c_{g3}= 6, c_{g4}= 0 hours respectively). If the gas network was to take the network dependency into account, it would visit g1 before *g3* leading to a total completion time of 13 hours (completion times of gas nodes are c_{g1}= 2.5, c_{g2}= 4.5, c_{g3}= 6, c_{g4}= 0 hours respectively).

Campell et al. [35] define the starting point of our study. In this paper, a basic model is developed which is necessary for Vehicle Routing Problem (VRP) parts of all processes for post disaster humanitarian relief operations. In the objective function they aim to minimize the maximum arrival time or minimize total arrival time. They model separate objective functions for VRP and Travelling Salesman Problem (TSP) and introduce versions with additional constraints such as capacity. Based on this model, we aim to minimize completion time which we define in the objective function of our problem and add inter-dependency constraints.

Since we are minimizing total completion time, the route visits the closest nodes in order to make them functional as soon as possible when dependency is not considered. However when we have the information on the dependency, and know that we will have to wait for the other network components to function, then it is better to use the idle time to visit further customers and utilize that idle time. For this reason, objective function in our mode is not *minmax* completion time. In *minmax* problems, all nodes are completed close to each other when only the completion time of damaged node that is expected to be completed the latest is minimized. We choose *minsum* objective function in order to consider inter-dependency as soon as possible and minimize the total completion time.

Next, we present our mathematical model for this problem. Let N^a represent the set of all nodes in the overall system. In addition, let N represent the set of all but

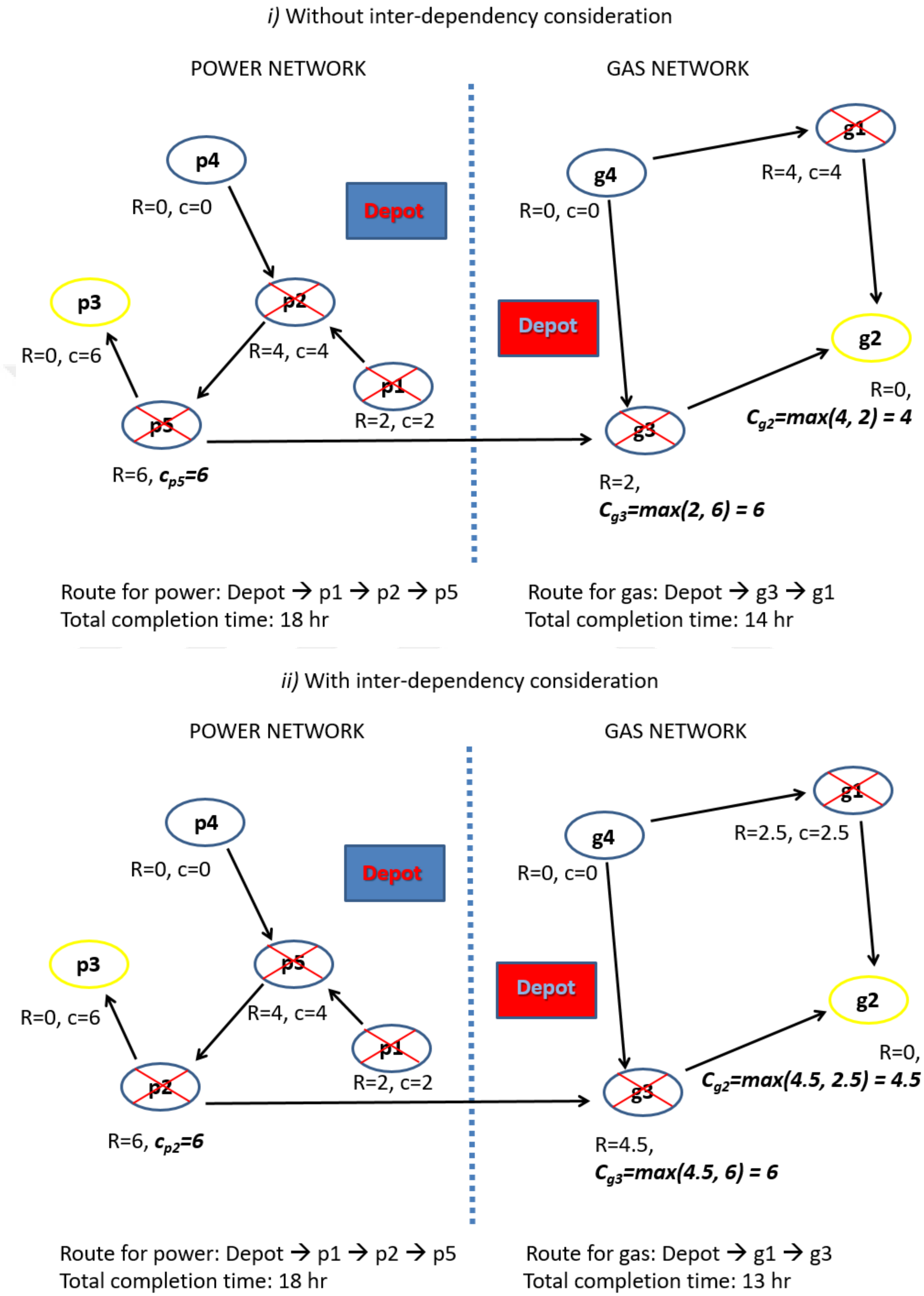

Figure 5: Comparison of solutions *i)* without inter-dependency consideration, and *ii)* with inter-dependency consideration

the depot nodes. N consists of two parts: let N_p represent the set of power network nodes and N_g represent the set of gas network nodes. Let nodes $D = D_p \cup D_g$ be the depots for the power and gas networks, respectively, i.e., $N^a = N_p \cup N_g \cup D$. Similarly, let K represent the set of all repair teams (vehicles). There are k_p vehicles for the power network and k_g vehicles for the gas network. Let B represent the set of all damaged power and gas nodes, where B_p is the damaged power nodes and B_g is the damaged gas nodes, $B_p \cup B_g$. Dependency between nodes i and j is represented by $d_{ij} \ \ \forall i, j \in N$. d_{ij} take the value of 1 if node $i \in N$ depends on $j \in N$, and 0 otherwise. We assume that nodes depends on itself, $d_{ii} = 1$, $\forall i \in N$. Travel time between nodes i and j is represented by $t_{ij} \ \ \forall i, j \in N^a$. Repair time of node i is denoted by $r_i \ \ \forall i \in N$. M is assumed to be a large number.

There are three sets of decision variables. x_{ij} take the value of 1 if node $j \in N^a$ is visited immediately after node $i \in N^a$, and 0 otherwise. Arrival time at a node is represented by a_i and the completion time of a node, i.e., time that node i is functional again is represented by $c_i \ \ \forall i \in N$, respectively. Our formulation can model multiple vehicles without vehicle (k) index thanks to relation between arrival time (a_i) and visiting nodes (x_{ij}) decision variables. The notation and the mathematical formulation of the SRPRIIN are as follows:

Sets:

N^a: Set of all nodes on both networks, where N_p is the power nodes and N_g is the gas nodes, $N^a = N_p \cup N_g$.

$D = \{0, 1\}$: Set of all depots, where D_p is the depot for power and D_g is the depot for gas, $D = D_p \cup D_g$.

N: Set of all non-depot nodes, $N = N^a \setminus D$.

B: Set of damaged power and gas nodes, where B_p is the damaged power nodes and B_g is the damaged gas nodes, $B = B_p \cup B_g$.

K: Set of vehicles of power and gas networks, where K_p is the number of vehicles for power and K_g is the number of vehicles for gas, $K = K_p \cup K_g$.

Parameters:

$d_{ij} = 1$, if i depends on j, $\forall i, j \in N$ and 0, otherwise.

t_{ij}: travel time between i and j, $\forall i, j \in N^a$.

r_i: repair time for i, $\forall i \in B$.

Decision Variables:

$x_{ij} = 1$, if node j is visited after node i, and 0 otherwise, $\forall i, j \in B \cup D$.

c_i: completion time for $i \in N$.

a_i: arrival time at $i \in B$.

$$\text{IP:} \quad \text{Minimize} \sum_{i \in N} c_i$$

$$
\begin{aligned}
s.t. \quad & c_i \geq d_{ij}(a_j + r_j) && \forall i \in N, \forall j \in B && (1)\\
& a_i \geq \sum_{d \in D} t_{di} x_{di} && \forall i \in B, && (2)\\
& a_i + r_i + t_{ij} \leq a_j + M(1 - x_{ij}) && \forall i, j \in B, && (3)\\
& \sum_{j \in B \cup D} x_{ij} = 1 && \forall i \in B && (4)\\
& \sum_{j \in B \cup D} x_{ji} = 1 && \forall i \in B && (5)\\
& x_{ij} = 0 && \forall i \in B_p \cup D_p, \forall j \in B_g \cup D_g && (6)\\
& x_{ji} = 0 && \forall i \in B_g \cup D_g, \forall j \in B_p \cup D_p && (7)\\
& \sum_{i \in D_p} \sum_{j \in B_p} x_{ij} \leq k_p && && (8)\\
& \sum_{i \in D_g} \sum_{j \in B_g} x_{ij} \leq k_g && && (9)\\
& c_i \geq 0 && \forall i \in N && (10)\\
& a_i \geq 0 && \forall i \in B && (11)\\
& x_{ij} \in \{0, 1\} && \forall i, j \in B \cup D && (12)
\end{aligned}
$$

The objective of the problem is to minimize total completion time of all nodes. Constraint set (1) defines the completion time of a node. If i does not depend on any other node, then completion time of i is equal to the arrival time plus the repair time for that node. If there are other nodes $j(i \neq j)$ for which $d_{ij} = 1$, then c_i is equal to the maximum of arrival time plus the repair time for node i and nodes j. Arrival time is calculated using constraint sets (2) (for the first nodes visited after the depot) and (3) (for all other nodes). In addition, Constraint set (3) provides sub-tour elimination. With constraint sets (4) and (5) we make sure that one vehicle arrives at and leaves each visited node. Constraint sets (6) and (7) prevent visits between networks, i.e., vehicles for power and gas networks may only repair their own networks. Constraint sets (8) and (9) limit the number of vehicles leaving and returning to depot which also represent the repair teams. Lastly, Constraint sets (10) and (11) are the non-negativity and Constraint (12) is the binary constraint.

As a synchronized vehicle routing problem, SRPRIIN is an NP-hard problem. Problem sizes that may be solved with this mathematical model, are very limited. Using an optimization tool, almost one day is required to solve a problem instance that has networks with more than 20 nodes. In response processes after disasters, usage of optimization tools by authorities in disaster coordination centers is not a very efficient way due to the run times. These run times prevent rapid decision making and implementation of these decisions. To overcome these challenges, we provide two constructive heuristics and local search methods. We present two easy-to-implement constructive heuristics (with two different versions of each heuristic) and local search methods. These solution methods would help infrastructure restoration teams with quick decision making and restoring damaged networks in a fast and effective manner.

CHAPTER IV

SOLUTION METHODOLOGY

4.1 Constructive Heuristics

In this section, we present two constructive heuristics (CH) to solve the SRPRIIN. Our heuristics prioritize visiting and repairing the damaged nodes with respect to their dependencies on other nodes. First heuristic is named *peer heuristic* (CH^P) and the second heuristic is named *leader-follower heuristic* (CH^{LF}). The difference between two solution approaches is in designating the priorities of the broken nodes to be restored. In CH^P, power and gas infrastructures designate their own restoration plans for the sake of the performance of the whole system. In other words, we assume the importance of the infrastructures to be equal in the decision making process of infrastructure restoration plans. In the CH^{LF}, gas infrastructure, which we set as in the follower position, designates its own restoration plan according to the route of power infrastructure. Furthermore, we develop two versions of each heuristic in order to determine node weights by considering *weight strategy 1* (WS^1) and *weight strategy 2* (WS^2). WS^1 and WS^2 of CH^P are denoted by CH^{P1} and CH^{P2}, respectively. WS^1 and WS^2 of CH^{LF} are denoted by CH^{LF1} and CH^{LF2}, respectively. In the WS^2, inter-dependencies between systems are weighted more than the ones within a system. For example, power network would prefer repairing a node which affects two other nodes in the gas network, rather than a node which affects two other nodes in the power network. In the WS^1, weights are equal. More specifically, to calculate the weight of node i, we count the number of nodes that are dependent on node i. Depending on the system, nodes from the other network may or may not be of more importance.

There are two major steps of our constructive heuristics. First one is the initialization part which determines the order of nodes precedence. The second is assigning nodes to vehicles in a way that minimizes completion time. Then, these steps are followed by a local search improvement. The structure of the constructive heuristics are represented in Figure 6. Steps of heuristics steps are explained in detail. In addition, Figure 7 presents an example network on which we illustrate some heuristics steps.

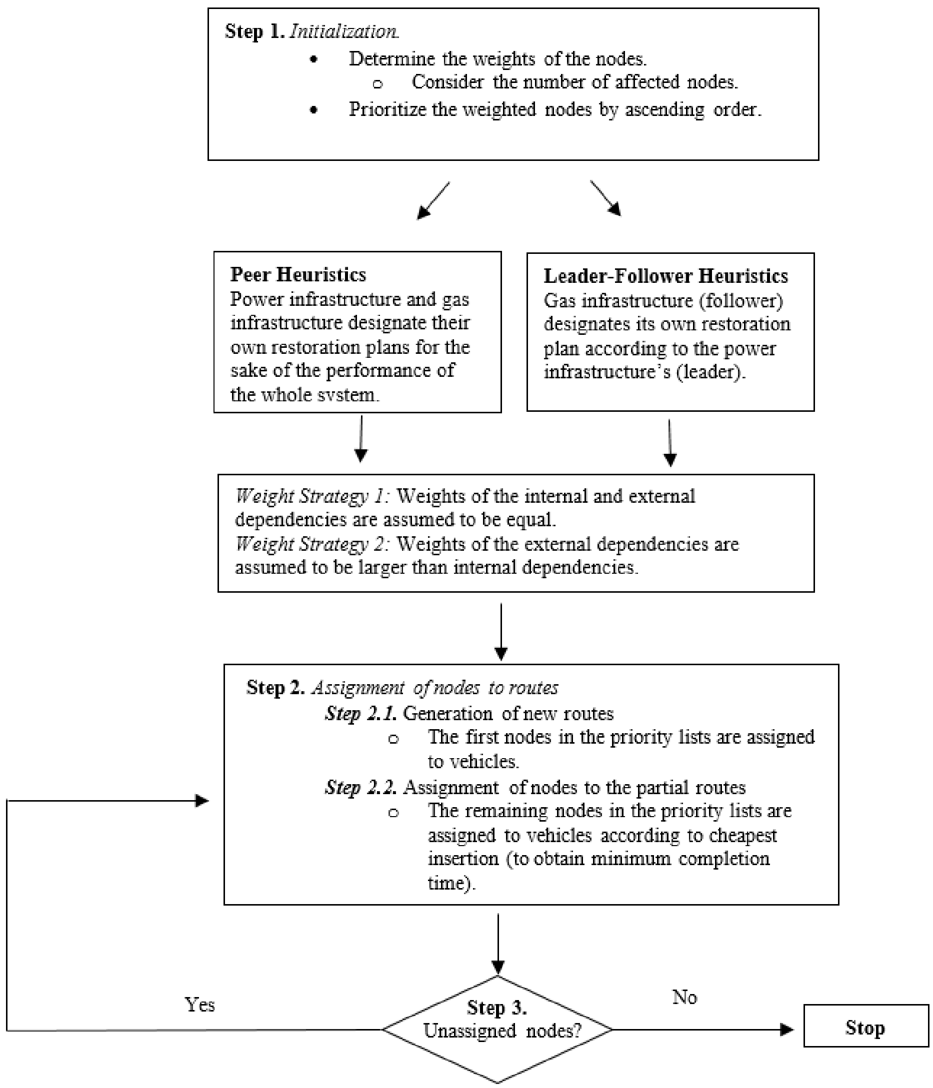

Figure 6: Flowchart of the constructive heuristic approaches

Step 1. Initialization. Weights of the damaged nodes for each infrastructure is calculated. We assign weights to each node with respect to the number of other nodes that are dependent on them. Basically, weight calculation depends on the number of affected nodes. In cases of CH^{P1} and CH^{P2}, power infrastructure and gas infrastructure designate their own restoration plans for the sake of the performance of the whole system. In cases of CH^{LF1} and CH^{LF2} gas infrastructure (follower) designates its own restoration plan according to the power infrastructures (leader). Two different versions are developed for each constructive heuristic; in WS^1, weights of the internal and external dependencies are assumed to be equal, while in the WS^2, weights of the external dependencies are assumed to be larger than internal dependencies. In the process of employing WS^1, the model assigns a weight of 1 to a affected node, if this affected node is within or from a different network. Whereas in WS^2, the model assigns a weight of 1.25 to a node from another network due to the intuition of giving high priority to other networks. After the other network is functional, it would aid the functionality efforts of the WS^2.

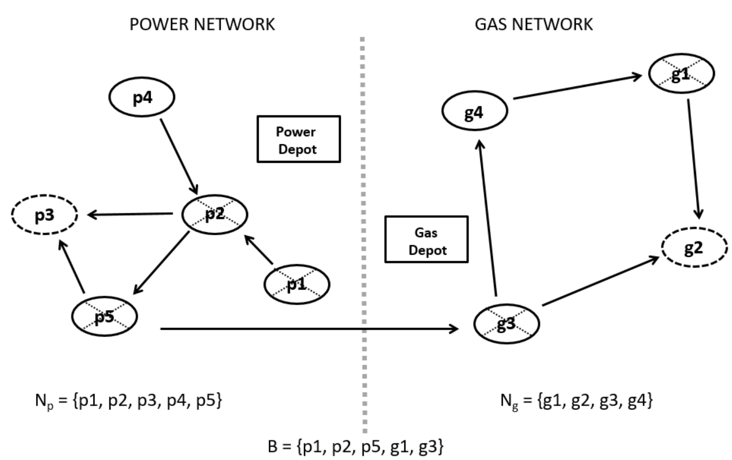

Figure 7: Illustrative example for heuristics steps

In Figure 7, we have two networks where arrows represent dependencies among nodes as illustrated and explained in section 3. The difference between two examples lies in the dependencies among nodes. In this figure, *p3* also depends on *p2* and *g4* depends on *g3* in order to function. For the node prioritization step, following table demonstrates the weights of the nodes according to heuristics.

CH	**Weights**					**Priority List**	
	p1	**p2**	**p5**	**g1**	**g3**	**Power**	**Gas**
CH^{P1}	1	2	2	1	1	p2-p5-p1	g1-g3
CH^{P2}	1	2	2.5	1	1	p5-p2-p1	g1-g3
CH^{LF1}	1	2	2	1	1	p2-p5-p1	g1-g3
CH^{LF2}	1	2	2.5	1	1	p5-p2-p1	g1-g3

Table 1: Node prioritization of illustrative example in Figure 7

WS^1 and WS^2 differentiate in weight calculation. In both versions, the number of nodes that depends on a specific node is important. In WS^1, every node, in both power and gas networks, has equal importance. However in WS^2, affected nodes in the other infrastructure are more important. For instance, let us examine node *p5*. In WS^1, the weight is 2 because the number of nodes that node *p5* affects in total is 2. The associated weight in the WS^2 is calculated as 2.5 since node *p5* affects node *g3* with weight 1.5 which is in the gas network. Furthermore, we create a priority list which provides the information about the order of the visits to damaged nodes, after assigning weights to damaged nodes. In cases of nodes with equal weight, such nodes are ordered numerically in the priority list.

Step 2. Assignment of nodes to routes. Node prioritizations are determined on the first step, our goals to assign them to vehicle routes. Node assignment in peer heuristic is carried out as follows: Each infrastructure route is created within networks. First of all, prioritized nodes are assigned to vehicles as first in the sequence. In other words, the number of nodes that are equal to the number of vehicles on the top of priority list are selected. Then we assign these nodes to vehicles one by one as the first visited node. The other unassigned nodes in priority list are included in the route using cheapest insertion method in order to minimize completion time. In this step, nodes to be assigned are selected in order according to priority list. The vehicle and the route order to be assigned is determined by the minimum increase in the total completion time, i.e. cheapest insertion heuristic. To avoid overlooking good solutions due to assignment of the first prioritized node, we use the following rule. The distance of the first node to be assigned to a depot should be shorter than the one thirds of the distance between the depot and the farthest node. If there is no such node, the next node in the priority list assigned.

Node assignment in leader-follower heuristic is carried out in two steps; node assignment of leader infrastructure and follower infrastructure. Nodes of leader infrastructure are assigned like in peer heuristic's. Node assignment in follower infrastructure is carried out as follows: First of all, gas nodes that are affected by power infrastructure are defined. Then we calculate the repair time in hours if the gas repair vehicles visit these affected nodes first. If calculated repair time exceeds the completion time of the node which affects the nodes in the other networks, affected gas node is assigned to new priority list. According to new list, nodes are assigned to the vehicles as in peer heuristic. Given this setup, this gas node becomes functional as soon as it is repaired. If this calculated repair time is shorter than the completion time of the node which affects the nodes in the other networks, this gas node may be assigned according to the old priority list. This is because it has to wait for the completion time of the electric node to be functional, even if it is repaired first. If the number of affected nodes are larger than the number of available repair vehicles, routes for damaged nodes are generated using cheapest insertion method. After the first assignment of the affected gas nodes, node assignment rules from peer heuristic and leader infrastructure in leader-follower heuristic, are applied for routing of the remaining nodes.

When we examine the example in Figure 7, it is possible to observe that all constructive heuristics differentiate in the node assignment step. Routes for the CH^P

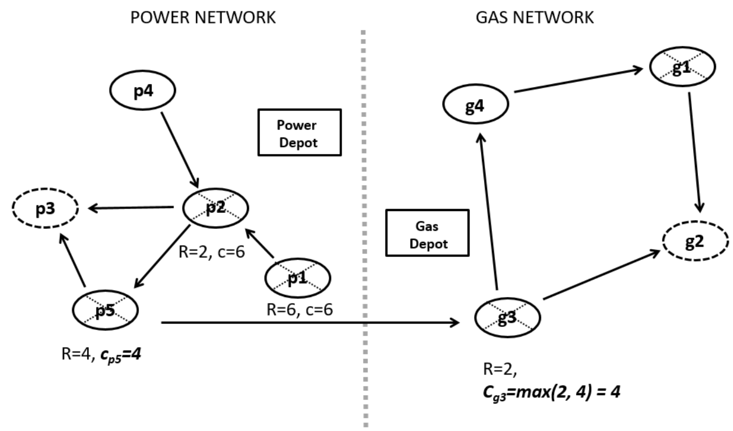

Figure 8: Illustrative example for CH^{LF1}

heuristic are generated directly according to the priority list. Routes for the power infrastructure in CH^{LF} heuristic is similar to the priority list. However, there is a difference in the gas network. In CH^{LF1}, since the repair time of *p5* is 4 hours, repair of *g3* may be finished until then. If the gas route visits *g3* first, *g3* is repaired after 2 hours, however it is functional after 4 hours. For this reason repair vehicle visits damaged node *g1* first (see Figure 8). On the other hand, in CH^{LF2},since the first visit is to *p5* in power infrastructure, route for the gas repair team is *g3*, *g1*. When we examine these 4 different cases in Table 2, we can note that CH^{LF2} heuristic yields the shortest completion time.

CH	**Route Assignment**		**Completion Times**		
	Power	**Gas**	**Power**	**Gas**	**Total**
CH^{P1}	p2-p5-p1	g1-g3	20	14	34
CH^{P2}	p5-p2-p1	g1-g3	18	14	32
CH^{LF1}	p2-p5-p1	g1-g3	20	14	34
CH^{LF2}	p5-p2-p1	g3-g1	18	12	30

Table 2: Node assignment of illustrative example in Figure 7

4.2 Improvement Heuristics

Heuristics continue to operate unless all the nodes are assigned. Routes computed from these heuristics are improved using local search methods: *swap*, *exchange* and *remove* & *insert* methods. In the swap method we test whether there are routes that have less completion times by changing the places of the nodes in routes. In the exchange improvement method, we test if there exist a better solution by swapping each node pairs from different routes. Finally, in remove and insert method, we test if we may have additional gains when we take a node out of a route and insert it into another route. These algorithms are executed in the mentioned order. Stopping criteria of the improvement algorithm is the case of no improvement in a given step, which is the absence of smaller total completion time. Implementation of each neighborhood search method is explained in detail in the following section and in Figure 13. We provide an example (see Figure 9) to present improvement methods. In this example, we assume that there are two vehicles for power network and there is only one vehicle for gas network.

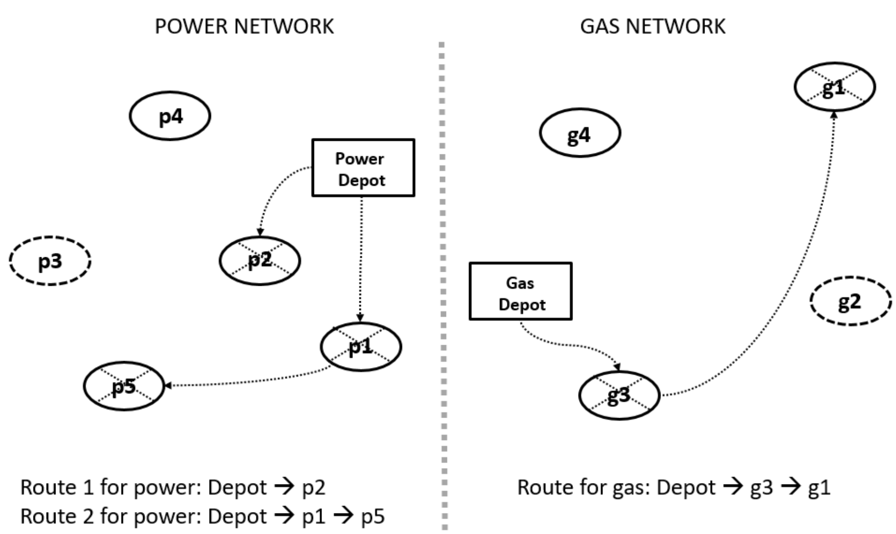

Figure 9: Example routes of initial solution generated by constructive heuristic

1. ***Swap Method:*** In this method, one swap procedure for each infrastructure is executed. Nodes are swapped in sequence in each route and infrastructure. All of the routes are examined one by one for each infrastructure as illustrated in Figure 10, intra-route move is applied. Pairs of nodes are examined in all routes. In the cases of these pairs of nodes are swapped, associated completion times are computed. All moves are applied in both infrastructures and the most improved one solution per iteration is chosen and swap procedure is executed.

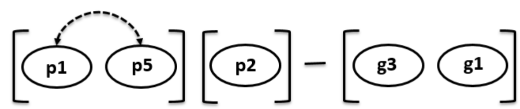

Figure 10: Illustrative example of swap method

2. ***Exchange Method:*** In this method, one exchange procedure for each infrastructure is executed. Nodes between two routes are swapped for each infrastructure as in Figure 11. In this method, routes are not examined individually, inter-route move is applied. For instance, let us consider two routes for each power and gas infrastructure. Nodes in the first power route are swapped with the nodes in the second power route. And similarly, nodes in the first gas route are swapped with the nodes in the second gas route. Then, we compute completion time in cases where nodes are exchanged between routes. All moves are applied in both infrastructures and the most improved one solution per iteration is chosen and exchange procedure is executed.

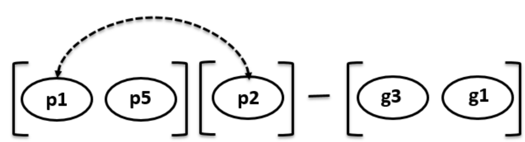

Figure 11: Illustrative example of exchange method

3. ***Remove*** & ***Insert Method:*** In this method, one remove and one insert for each infrastructure are executed. Nodes are removed from their routes in sequence and inserted in the other routes for each infrastructure as in Figure 12. All moves are applied in both infrastructures and the most improved one solution per iteration is chosen. Route that results with the minimum completion time is assigned as the current solution for the next step.

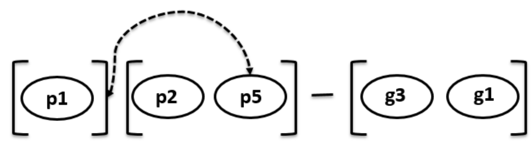

Figure 12: Illustrative example of remove and insert method

In a nutshell, two constructive heuristics, which are peer ve leader-follower constructive heuristics, and two different versions (WS^1 and WS^2) for these heuristics are developed for SRPRIIN. These heuristics differ according to the designation of priorities given to the inter-dependencies between infrastructures. Three neighborhood search methods (swap, exchange and remove and insert methods) are developed to improve the results from constructive heuristics. In the following section, performance of these heuristics and their versions are studied and compared.

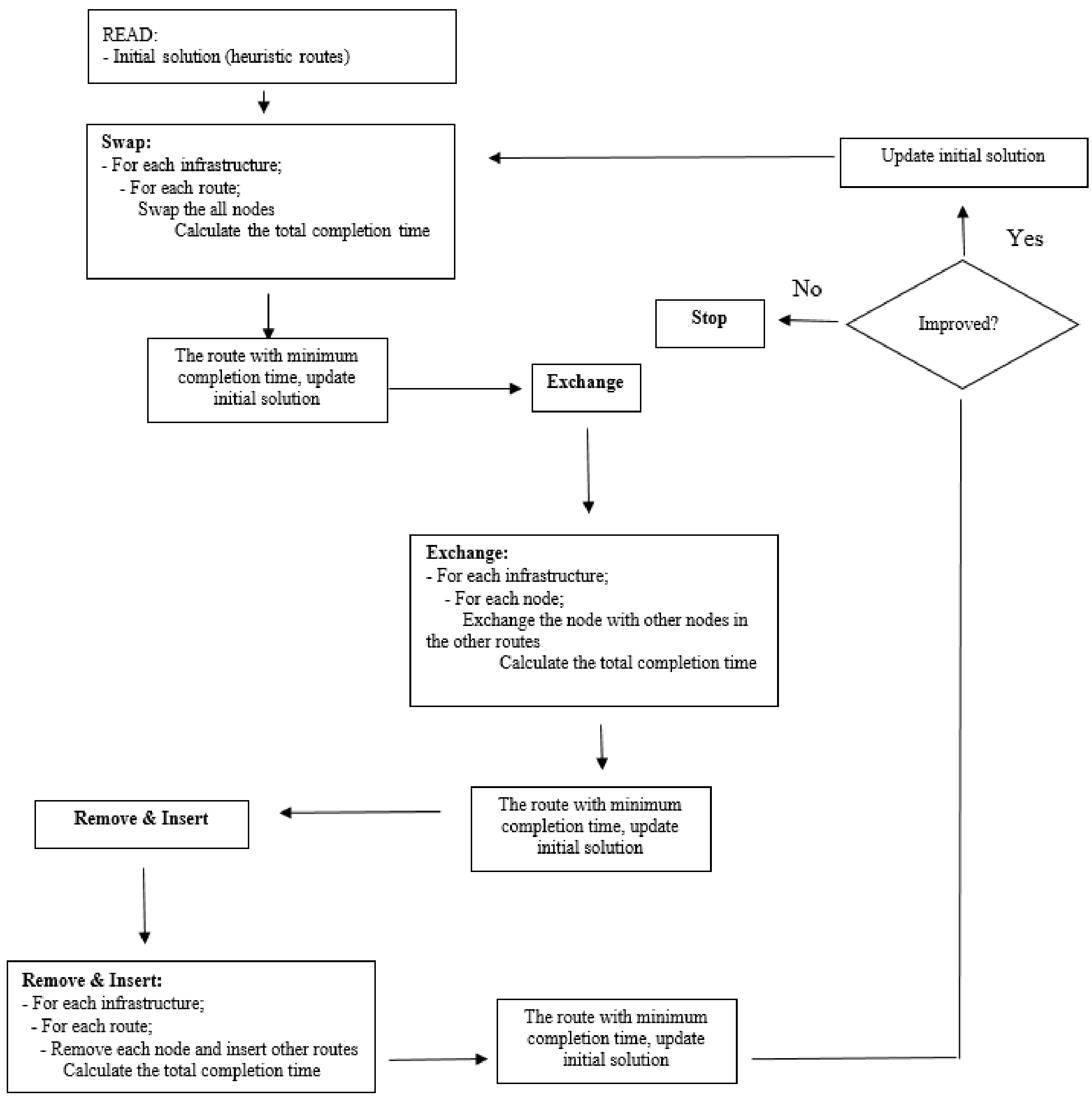

Figure 13: Flowchart of the local search improvement approaches

CHAPTER V

NUMERICAL STUDY

In this section we describe our test instances, present solutions for the SRPRIIN and evaluate the performance of the proposed heuristics.

5.1 Test instances

We develop sets of test instances with 20, 50 and 100 nodes. Each instance includes information on number of power and gas nodes, number of non-functional nodes, node locations, inter-dependency relationship between nodes within and across networks and number of vehicles.

Characteristics of Instances:

- *Number of power and gas nodes.* Number of power nodes are selected to be 50%, 60% and 75% of the total number of nodes. In this paper, $\#N$ refers to number of total nodes of both networks. $\#N_p$ and $\#N_g$ refer to number of total nodes of power and gas networks respectively.

- *Location of nodes.* Power and gas nodes are located randomly across a grid of size 100x100. Power and gas depots are located on points (45,50) and (50,45), respectively. Euclidean distance is used for distance calculation between nodes. Later, these distances are divided by 60 to convert the time unit to hours. All the computation is carried out over time unit.

- *Number of damaged nodes.* Randomly selected 20% and 60% of the number of nodes in each network are assumed to be damaged nodes. Number of damaged nodes of power and gas networks are represented by $\#B_p$ and $\#B_g$.

- *Inter-dependency matrix.* Two different inter-dependency types are developed for each instance. First type is when only gas nodes are dependent on power nodes, i.e., one directional inter-dependency (example d_{ij} matrix can be seen in Table 3). Second type is when both network nodes may be dependent on each other, i.e., bidirectional inter-dependency (example d_{ij} matrix can be seen in Table 4). Moreover, closeness of nodes is taken into account in the process of inter-dependency decision making both inside and outside the infrastructure. Specifically, we first compute the distance of all nodes to each other is computed. Later, a point which is farthest from a node (e.g. node a) is chosen when determining the inter-dependent nodes of node a. One third of the farthest distance is considered a threshold. Inter-dependency is assumed to exist if a node is closer to node a than the threshold distance. Then, random pairs are chosen and assumed to be inter-dependent among pairs of nodes which satisfy the threshold criterion. We assume that 20% of the nodes are selected for inter-dependency connection between networks, 60% of the nodes are selected for inter-dependency connection within networks. Besides, *Direction* refers to one directional and bidirectional inter-dependencies of networks in the tables.

- *Number of vehicles.* We assume two vehicles (teams) for each infrastructure.

- *Node repair times.* The repair time of all nodes are assumed to be one hour.

We first create three different instances of 20 nodes; *i)* 10 power and 10 gas nodes, *ii)* 12 power and 8 gas nodes and *iii)* 15 power and 5 gas nodes, three different instances of 50 nodes; *i)* 25 power and 25 gas nodes, *ii)* 30 power and 20 gas nodes and *iii)* 38 power and 12 gas nodes, three different instances of 100 nodes; *i)* 50 power and 50 gas nodes, *ii)* 60 power and 40 gas nodes and *iii)* 75 power and 25 gas nodes. For each data set, we consider four instances with different number of damaged nodes. In other words, if we examine the instance of 10 power and 10 gas node, there may be

Node	Dependencies																			
	1	2	3	4	5	6	7	8	9	10	12	13	14	15	16	17	18	19	20	21
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1
16	0	1	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
20	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1

Table 3: One directional node dependencies (d_{ij} values) for the 20 node (10 power - 10 gas) network

2 power - 2 gas, 2 power - 6 gas, 6 power - 2 gas and 6 power - 6 gas many damaged nodes, respectively. This way, number of instances that we developed increases to 36. Allowing for two types of inter-dependency, one directional and bidirectional inter-dependencies, we obtain 72 test instances. An example of a 20 node network may be found in Figure 14. For details on instances, you can see Tables 5, 6 and 7. SRPRIIN solutions and heuristics performances are studied in section 5.2.

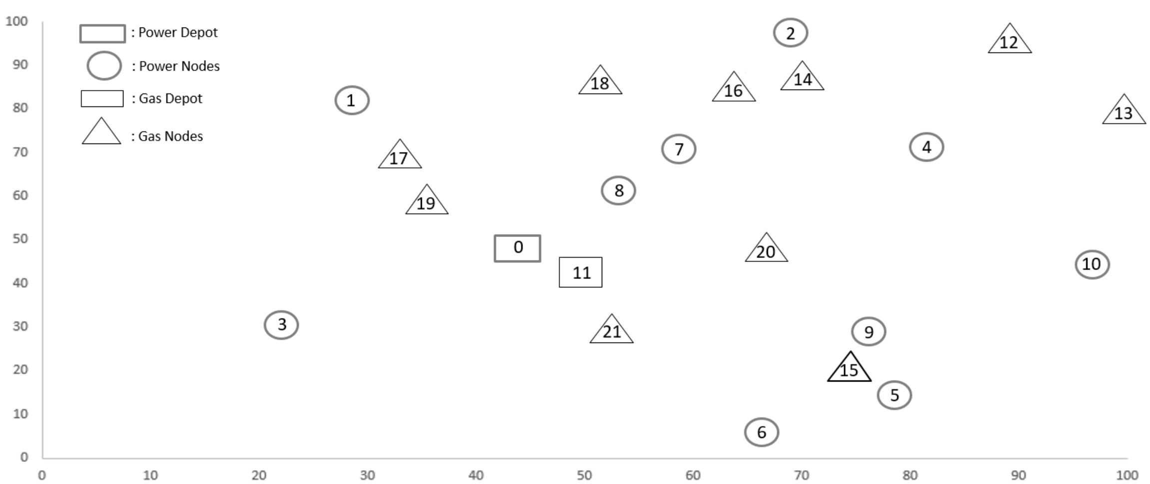

Figure 14: 10 power-10 gas type network

	Dependencies																			
Node	1	2	3	4	5	6	7	8	9	10	12	13	14	15	16	17	18	19	20	21
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
2	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	1	0	0	1	0	0	0	0	0	1	0	0	0	0	0
7	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0
8	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1
16	0	1	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
20	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1

Table 4: Bi-directional node inter-dependencies (d_{ij} values) for the 20 node (10 power - 10 gas) network

Instance	**#N**	**#Np**	**#Ng**	**#Bp**	**#Bg**	**Direction**
1	20	10	10	2	2	1
2	20	10	10	2	2	2
3	20	10	10	2	6	1
4	20	10	10	2	6	2
5	20	10	10	6	2	1
6	20	10	10	6	2	2
7	20	10	10	6	6	1
8	20	10	10	6	6	2
9	20	12	8	3	2	1
10	20	12	8	3	2	2
11	20	12	8	3	5	1
12	20	12	8	3	5	2
13	20	12	8	8	2	1
14	20	12	8	8	2	2
15	20	12	8	8	5	1
16	20	12	8	8	5	2
17	20	15	5	3	1	1
18	20	15	5	3	1	2
19	20	15	5	3	3	1
20	20	15	5	3	3	2
21	20	15	5	6	1	1
22	20	15	5	6	1	2
23	20	15	5	6	3	1
24	20	15	5	6	3	2

Table 5: Instance details of instances with 20 nodes

Instance	#N	#Np	#Ng	#Bp	#Bg	Direction
25	50	25	25	5	5	1
26	50	25	25	5	5	2
27	50	25	25	5	15	1
28	50	25	25	5	15	2
29	50	25	25	15	5	1
30	50	25	25	15	5	2
31	50	25	25	15	15	1
32	50	25	25	15	15	2
33	50	30	20	6	4	1
34	50	30	20	6	4	2
35	50	30	20	6	12	1
36	50	30	20	6	12	2
37	50	30	20	18	4	1
38	50	30	20	18	4	2
39	50	30	20	18	12	1
40	50	30	20	18	12	2
41	50	38	12	8	3	1
42	50	38	12	8	3	2
43	50	38	12	8	8	1
44	50	38	12	8	8	2
45	50	38	12	23	3	1
46	50	38	12	23	3	2
47	50	38	12	23	8	1
48	50	38	12	23	8	2

Table 6: Instance details of instances with 50 nodes

Instance	#N	#Np	#Ng	#Bp	#Bg	Direction
49	100	50	50	10	10	1
50	100	50	50	10	10	2
51	100	50	50	10	30	1
52	100	50	50	10	30	2
53	100	50	50	30	10	1
54	100	50	50	30	10	2
55	100	50	50	30	30	1
56	100	50	50	30	30	2
57	100	60	40	12	8	1
58	100	60	40	12	8	2
59	100	60	40	12	24	1
60	100	60	40	12	24	2
61	100	60	40	36	8	1
62	100	60	40	36	8	2
63	100	60	40	36	24	1
64	100	60	40	36	24	2
65	100	75	25	15	5	1
66	100	75	25	15	5	2
67	100	75	25	15	15	1
68	100	75	25	15	15	2
69	100	75	25	45	5	1
70	100	75	25	45	5	2
71	100	75	25	45	15	1
72	100	75	25	45	15	2

Table 7: Instance details of instances with 100 nodes

5.2 Heuristic Performances

In this section, we will compare initial and final solutions with CPLEX and compare constructive heuristics improvement heuristics.

5.2.1 Comparison with CPLEX

Our model is solved for the 72 instances using IBM ILOG OPL/CPLEX on a 64-bit Windows Server with two 2.0 GHz Intel Xeon CPU's and 32 GB RAM. Optimal solutions obtained by CPLEX (Z^{CPLEX}), optimality gaps (O.Gap (%), limiting the solution time to one hour) and solution times (S.Time (s)) computed with CPLEX, solutions of two constructive heuristic and two versions and local search improvement solutions are summarized between Table 8 and Table 12 as $Z(CH^{P1})$, $Z(CH^{P2})$ and $Z(CH^{LF1})$, $Z(CH^{LF2})$ and Z^{CPLEX}. More specifically, $Z(CH^{P1})$ and $Z(CH^{P2})$ terms represent the objective function values of peer heuristics for two versions (WS^1 and WS^2), $Z(CH^{LF1})$ and $Z(CH^{LF2})$ terms represent the objective function leader-follower heuristics for two versions. We refer to the best solution obtained by the heuristics as the "best solution" and its objective value as the $Z(CH^{best})$. The last column of tables presents the percentage gap between the $Z(CH^{best})$ and the Z^{CPLEX}.

In Table 8, we present the CPLEX results of instances with 20 nodes which are obtained from two different constructive heuristics and associated WS^1 and WS^2. In the first part of Table 8 optimality gaps, the run times and the results are presented for CPLEX runs. Second part represents the results of heuristics. The last part represents the gap between the optimal solution and best heuristic solution. In Table 9, we present the initial solutions from constructive heuristics which are improved using neighborhood search methods. Tables 10, 11, 12 and 13 represent the results of instances with 50 and 100 nodes with the same structure. On average, run-time of constructive heuristics is approximately one minute and the heuristics involving local search find a solution in at most three minutes. For this reason the run times are not

				Initial Solutions				
Instance	**O.Gap (%)**	**S.Time (s)**	$\mathbf{Z}^{CPLEX}$	**Z(CH^{P1})**	**Z(CH^{P2})**	**Z(CH^{LF1})**	**Z(CH^{LF2})**	$\frac{Z(CH^{best})-Z^{CPLEX}}{Z^{CPLEX}} * 100(\%)$
1	0.00	0.16	6.77	6.77	6.77	6.77	6.77	0.00%
2	0.00	0.21	6.77	6.77	6.77	6.77	6.77	0.00%
3	0.00	21.45	23.04	25.62	25.62	24.12	24.12	4.70%
4	0.00	10.05	20.82	24.00	23.92	22.97	21.51	3.33%
5	0.00	13.07	24.23	24.23	26.59	24.23	26.59	0.00%
6	0.00	18.00	24.23	24.23	26.59	24.23	26.59	0.00%
7	0.00	1435.65	38.03	43.09	45.45	43.09	45.45	13.31%
8	0.00	1226.30	39.74	41.46	43.74	41.46	43.74	4.33%
9	0.00	0.08	9.87	9.87	9.87	9.87	9.87	0.00%
10	0.00	0.11	8.21	8.21	8.21	8.21	8.21	0.00%
11	0.00	2.81	21.82	21.90	21.90	21.90	21.90	0.35%
12	0.00	3.90	22.78	24.00	24.00	24.00	24.00	5.36%
13	0.00	970.55	33.26	38.67	40.10	38.67	40.10	16.27%
14	0.00	257.60	32.97	38.67	40.10	38.67	40.10	17.29%
15	14.97	3605.30	44.67	50.70	52.13	50.70	52.13	13.49%
16	18.33	3602.45	44.67	52.81	54.24	52.81	54.24	18.20%
17	0.00	0.58	7.66	7.66	7.66	7.66	7.66	0.00%
18	0.00	1.02	7.66	7.66	7.66	7.66	7.66	0.00%
19	0.00	3.10	14.00	14.16	14.16	14.16	14.16	1.15%
20	0.00	3.17	15.88	15.88	15.88	15.88	15.88	0.00%
21	0.00	789.80	22.69	23.49	23.49	23.49	23.49	3.54%
22	0.00	1118.65	23.08	23.49	23.49	23.49	23.49	1.80%
23	0.00	840.38	29.19	30.00	30.00	30.00	30.00	2.75%
24	0.00	885.32	30.91	31.71	31.71	31.78	31.78	2.60%

Table 8: Initial solutions for 20 nodes

				Final solutions (with Local Search)				
Instance	**O.Gap (%)**	**S.Time (s)**	$\mathbf{Z}^{CPLEX}$	$\mathbf{Z(CH}^{P1}\mathbf{)}$	$\mathbf{Z(CH}^{P2}\mathbf{)}$	$\mathbf{Z(CH}^{LF1}\mathbf{)}$	$\mathbf{Z(CH}^{LF2}\mathbf{)}$	$\frac{Z(CH^{best})-Z^{CPLEX}}{Z^{CPLEX}} * 100(\%)$
1	0.00	0.16	6.77	6.77	6.77	6.77	6.77	0.00%
2	0.00	0.21	6.77	6.77	6.77	6.77	6.77	0.00%
3	0.00	21.45	23.04	23.04	23.04	23.04	23.04	0.00%
4	0.00	10.05	20.82	20.82	20.82	20.82	20.82	0.00%
5	0.00	13.07	24.23	24.23	24.23	24.23	24.23	0.00%
6	0.00	18.00	24.23	24.23	24.23	24.23	24.23	0.00%
7	0.00	1435.65	38.03	38.03	38.03	38.03	38.03	0.00%
8	0.00	1226.30	39.74	39.74	39.74	39.74	39.74	0.00%
9	0.00	0.08	9.87	9.87	9.87	9.87	9.87	0.00%
10	0.00	0.11	8.21	8.21	8.21	8.21	8.21	0.00%
11	0.00	2.81	21.82	21.82	21.82	21.82	21.82	0.00%
12	0.00	3.90	22.78	22.78	22.78	22.78	22.78	0.00%
13	0.00	970.55	33.26	33.26	33.26	33.26	33.26	0.00%
14	0.00	257.60	32.97	32.97	32.97	32.97	32.97	0.00%
15	14.97	3605.30	44.67	44.67	44.67	44.67	44.67	0.00%
16	18.33	3602.45	44.67	44.67	44.67	44.67	44.67	0.00%
17	0.00	0.58	7.66	7.66	7.66	7.66	7.66	0.00%
18	0.00	1.02	7.66	7.66	7.66	7.66	7.66	0.00%
19	0.00	3.10	14.00	14.00	14.00	14.00	14.00	0.00%
20	0.00	3.17	15.88	15.88	15.88	15.88	15.88	0.00%
21	0.00	789.80	22.69	22.69	22.69	22.69	22.69	0.00%
22	0.00	1118.65	23.08	23.08	23.08	23.08	23.08	0.00%
23	0.00	840.38	29.19	29.19	29.19	29.19	29.19	0.00%
24	0.00	885.32	30.91	30.91	30.91	30.91	30.91	0.00%

Table 9: Final solutions for 20 nodes

				Initial Solutions				
Instance	**O.Gap (%)**	**S.Time (s)**	$\mathbf{Z}^{CPLEX}$	**Z(CH^{P1})**	**Z(CH^{P2})**	**Z(CH^{LF1})**	**Z(CH^{LF2})**	$\frac{Z(CH^{best})-Z^{CPLEX}}{Z^{CPLEX}} * 100(\%)$
25	27.70	3605.56	29.68	31.57	31.57	31.57	31.57	6.36%
26	26.50	3607.95	29.12	29.50	31.54	29.50	31.54	1.30%
27	62.85	3602.16	126.90	131.10	131.10	131.10	131.10	3.31%
28	66.35	3605.83	124.70	129.52	129.50	129.52	129.50	3.84%
29	66.00	3602.22	110.84	123.28	123.85	123.28	123.85	11.22%
30	65.33	3602.59	114.17	123.06	124.85	123.06	124.85	7.79%
31	72.18	3606.37	212.40	232.71	239.17	232.71	239.17	9.56%
32	74.05	3601.07	220.76	242.84	248.16	242.84	248.16	10.00%
33	21.20	3607.71	27.92	33.68	33.14	33.68	33.14	18.68%
34	31.18	3603.03	29.99	34.31	34.28	34.31	34.28	14.29%
35	54.98	3609.84	96.80	97.89	98.28	97.89	98.28	1.13%
36	55.90	3603.28	87.97	98.76	92.38	98.76	92.38	5.01%
37	69.90	3604.74	147.90	160.98	161.01	160.98	161.01	8.84%
38	73.30	3603.16	154.13	160.98	161.01	160.98	161.01	4.45%
39	73.34	3606.47	210.67	241.98	231.47	241.98	231.47	9.88%
40	76.11	3604.38	220.08	244.04	227.54	244.04	227.54	3.39%
41	18.45	3605.06	38.74	40.02	40.84	40.02	40.84	3.30%
42	20.03	3605.93	40.02	40.84	41.34	40.84	41.34	2.05%
43	31.73	3606.27	66.59	70.67	70.22	70.67	70.22	5.45%
44	35.85	3604.39	64.70	65.15	67.68	65.15	67.68	0.70%
45	70.75	3608.86	254.72	256.15	268.87	256.15	268.87	0.56%
46	72.98	3603.06	245.92	256.15	268.87	256.15	268.87	4.16%
47	68.15	3605.24	279.54	309.71	311.29	309.71	311.29	10.79%
48	69.61	3606.08	267.82	311.70	313.27	311.70	313.27	16.38%

Table 10: Initial solutions for 50 nodes

				Final solutions (with Local Search)				
Instance	**O.Gap (%)**	**S.Time (s)**	$\mathbf{Z}^{CPLEX}$	**Z(CH^{P1})**	**Z(CH^{P2})**	**Z(CH^{LF1})**	**Z(CH^{LF2})**	$\frac{Z(CH^{best})-Z^{CPLEX}}{Z^{CPLEX}} * 100(\%)$
25	27.70	3605.56	29.68	29.12	29.12	29.12	29.12	-1.90%
26	26.50	3607.95	29.12	28.29	28.30	28.29	28.30	-2.86%
27	62.85	3602.16	126.90	105.68	105.68	105.68	105.68	-16.72%
28	66.35	3605.83	124.70	115.20	106.36	115.20	106.36	-14.71%
29	66.00	3602.22	110.84	100.80	100.80	100.80	100.80	-9.06%
30	65.33	3602.59	114.17	102.34	102.34	102.34	102.34	-10.36%
31	72.18	3606.37	212.40	195.42	195.42	195.42	195.42	-8.00%
32	74.05	3601.07	220.76	201.92	202.17	201.92	202.17	-8.54%
33	21.20	3607.71	27.92	27.25	27.25	27.25	27.25	-2.39%
34	31.18	3603.03	29.99	27.53	27.53	27.53	27.53	-8.21%
35	54.98	3609.84	96.80	80.54	80.54	80.54	80.54	-16.80%
36	55.90	3603.28	87.97	82.08	82.08	82.08	82.08	-6.70%
37	69.90	3604.74	147.90	137.17	133.17	137.17	133.17	-9.96%
38	73.30	3603.16	154.13	133.17	137.17	133.17	137.17	-13.60%
39	73.34	3606.47	210.67	193.36	193.36	193.36	193.36	-8.22%
40	76.11	3604.38	220.08	189.99	189.99	189.99	189.99	-13.67%
41	18.45	3605.06	38.74	33.93	33.93	33.93	33.93	-12.42%
42	20.03	3605.93	40.02	33.93	33.93	33.93	33.93	-15.23%
43	31.73	3606.27	66.59	63.50	63.50	63.50	63.50	-4.64%
44	35.85	3604.39	64.70	59.41	59.41	59.41	59.41	-8.17%
45	70.75	3608.86	254.72	205.58	205.58	205.58	205.58	-19.29%
46	72.98	3603.06	245.92	205.58	205.58	205.58	205.58	-16.40%
47	68.15	3605.24	279.54	219.14	219.14	219.14	219.14	-21.60%
48	69.61	3606.08	267.82	219.14	219.14	219.14	219.14	-18.18%

Table 11: Final solutions for 50 nodes

				Initial Solutions				
Instance	**O.Gap (%)**	**S.Time (s)**	$\mathbf{Z}^{CPLEX}$	$\mathbf{Z(CH}^{P1}\mathbf{)}$	$\mathbf{Z(CH}^{P2}\mathbf{)}$	$\mathbf{Z(CH}^{LF1}\mathbf{)}$	$\mathbf{Z(CH}^{LF2}\mathbf{)}$	$\frac{Z(CH^{best})-Z^{CPLEX}}{Z^{CPLEX}} * 100(\%)$
49	63.01	3609.46	105.38	106.39	108.54	106.39	108.54	0.95%
50	57.55	3605.86	104.49	105.01	107.17	105.01	107.17	0.50%
51	82.83	3606.23	502.64	504.92	505.08	504.92	505.08	0.45%
52	82.46	3603.05	448.89	547.08	537.59	547.08	537.59	19.76%
53	82.90	3609.37	485.30	501.80	495.36	501.80	495.36	2.07%
54	88.38	3605.93	508.65	546.10	532.47	546.10	532.47	4.68%
55	87.78	3605.84	778.69	886.40	888.43	886.40	888.43	13.83%
56	90.56	3607.03	849.22	886.78	886.84	886.78	886.84	4.42%
57	58.90	3604.86	95.38	101.83	101.23	101.83	101.23	6.14%
58	65.36	3603.91	96.05	101.92	101.33	101.92	101.33	5.49%
59	72.48	3615.08	329.66	334.45	335.60	334.45	335.60	1.45%
60	77.39	3605.57	325.41	381.39	379.42	381.39	379.42	16.60%
61	84.03	3604.49	483.49	539.73	549.90	539.73	549.90	11.63%
62	77.49	3605.21	485.16	538.84	548.61	538.84	548.61	11.06%
63	82.65	3608.64	751.74	776.36	775.93	776.36	775.93	3.22%
64	80.72	3602.71	743.36	752.10	770.94	752.10	770.94	1.18%
65	45.73	3613.66	119.84	129.62	126.12	129.62	126.12	5.24%
66	46.85	3618.35	123.94	129.62	126.12	129.62	126.12	1.76%
67	62.95	3616.44	217.76	233.66	231.30	233.66	231.30	6.22%
68	59.34	3610.20	227.90	235.64	233.28	235.64	233.28	2.36%
69	82.90	3608.17	948.47	995.04	1078.72	995.04	1078.72	4.91%
70	81.68	3609.81	949.65	996.22	956.88	996.22	956.88	0.76%
71	86.88	3615.26	1107.75	1167.07	1126.07	1175.88	1134.89	1.65%
72	82.26	3613.58	1106.07	1163.19	1122.19	1167.97	1126.98	1.46%

Table 12: Initial solutions for 100 nodes

				Final solutions (with Local Search)				
Instance	**O.Gap (%)**	**S.Time (s)**	$\mathbf{Z}^{CPLEX}$	**Z(CH^{P1})**	**Z(CH^{P2})**	**Z(CH^{LF1})**	**Z(CH^{LF2})**	$\frac{Z(CH^{best})-Z^{CPLEX}}{Z^{CPLEX}} * 100(\%)$
49	63.01	3609.46	105.38	88.80	88.80	88.80	88.80	-15.74%
50	57.55	3605.86	104.49	88.49	88.49	88.49	88.49	-15.31%
51	82.83	3606.23	502.64	402.64	402.64	402.64	402.64	-19.90%
52	82.46	3603.05	448.89	422.33	422.33	422.33	422.33	-5.92%
53	82.90	3609.37	485.30	409.50	409.50	409.50	409.50	-15.62%
54	88.38	3605.93	508.65	487.19	487.19	487.19	487.19	-4.22%
55	87.78	3605.84	778.69	822.73	822.73	822.73	822.73	5.66%
56	90.56	3607.03	849.22	862.07	862.07	862.07	862.07	1.51%
57	58.90	3604.86	95.38	87.83	82.33	87.83	82.33	-13.67%
58	65.36	3603.91	96.05	89.75	99.29	89.75	99.29	-6.56%
59	72.48	3615.08	329.66	325.84	328.62	325.84	328.62	-1.16%
60	77.39	3605.57	325.41	316.36	339.99	316.36	339.99	-2.78%
61	84.03	3604.49	483.49	510.07	486.51	510.07	486.51	0.62%
62	77.49	3605.21	485.16	483.88	493.16	483.88	493.16	-0.26%
63	82.65	3608.64	751.74	648.11	648.11	648.11	648.11	-13.79%
64	80.72	3602.71	743.36	652.90	652.90	652.90	652.90	-12.17%
65	45.73	3613.66	119.84	109.35	109.35	109.35	109.35	-8.76%
66	46.85	3618.35	123.94	109.35	109.35	109.35	109.35	-11.78%
67	62.95	3616.44	217.76	175.35	175.35	175.35	175.35	-19.48%
68	59.34	3610.20	227.90	173.91	173.91	173.91	173.91	-23.69%
69	82.90	3608.17	948.47	845.99	845.99	845.99	845.99	-10.80%
70	81.68	3609.81	949.65	845.99	845.99	845.99	845.99	-10.92%
71	86.88	3615.26	1107.75	865.19	865.19	865.19	865.19	-21.90%
72	82.26	3613.58	1106.07	903.75	918.02	903.75	918.02	-18.29%

Table 13: Final solutions for 100 nodes

specified in the result tables.

For 20 node instances, optimal solutions for 22 of the 24 instances are obtained by CPLEX within one hour. Optimal solutions cannot be obtained for the remaining two instances (instance numbers are 15 and 16) due to large number of damaged nodes with high inter-dependencies. These instances yield optimal results after approximately 2 hours. Peer heuristics, leader-follower heuristics and versions yield results that are close to optimal solution by at least 20%. After obtaining initial solutions, they are improved by applying local search algorithms. All instances with 20 node instances obtain optimal solutions.

Instances with 50 and 100 nodes, all instances obtain large optimality gaps after one-hour runs in CPLEX. Results from optimization tool for 50 node instances have 54% optimality gap on average, with a maximum of 76%. CPLEX results for 100 node instances have 74% optimality gap on average, with a maximum of 91%. Peer and leader-follower heuristics perform equally in 90% of the instances with 50 nodes. In addition, WS^1 performs better than WS^2 in 35% of the instances. Furthermore, WS^1 performs better in 50% of the instances with 100 nodes. Initial solutions with 50 and 100 nodes are on average 7% and 4%, close to results from the optimization tool, respectively. Initial solutions are obtained within approximately 3 minutes, whereas optimization tool yields results within approximately one hour. When we improve initial solutions from these instances with local search algorithms, we observe that improved final solutions perform better when compared to results that are obtained from CPLEX as reported in Tables 11 and 13.

Furthermore, three different neighborhood search methods are applied to improve the initial solution performances: *i)* swap, *ii)* exchange, *iii)* remove and insert method. We can note that swaps and replacements improve the solutions for both heuristics and their versions, making them equal to the optimal solutions for all instances with 20 nodes. Because, most of the initial solutions are feasible solutions that are near optimal, and we are able to obtain optimal solutions in many instances just by local search methods. In instances with 50 and 100 nodes, 90% of the solutions that are obtained after local search improvement methods yield the same results as all heuristic cases. Although there are no optimal solutions in instances with 50 and 100 nodes, there are solutions with a minimum of 21% and a maximum of 89% optimality gaps. When we compare local search improvements solutions with benchmark solutions, local search improvement solutions perform better in 48 instances. We obtain shorter total completion times with improvement methods even though we are not able to do so with initial solutions. The worst solution obtained from local search improvement methods performs 1.9% better than the solution we obtain from CPLEX in a 1 hour set time. The most improved instance yields 23.69% shorter completion time when compared to the benchmark solutions.

This problem is a real life problem that may occur after a disaster. Our constructive heuristics are tested and observed to deliver near optimal solutions. Assuming that disaster and infrastructure coordination centers, such as gas and power, do not possess advanced optimization tools, our constructive heuristics may be helpful. In addition, they may be adapted easily to such larger scale networks in short periods of time.

5.2.2 Comparison of Heuristics

In this subsection, we compare the performances of constructive and improvement heuristics. To begin with, we compare the peer heuristic and leader-follower heuristic. After the comparison of peer heuristic and leader-follower heuristic, we present specific information regarding the performances of WS^1 and WS^2. Finally, we compare the performances of results of the improvement heuristics.

The performance of constructive heuristics depend on the instance structure. In some instances, peer heuristic yields better results. However in some instances leader-follower heuristic yields better results. In most instances with 20 nodes both heuristics yield same results, but in two different instances (see instances Number 3 and 4 in Table 8), leader-follower heuristic performs better. However, since each instance has different structure, it is not possible to make a generalization regarding the performance of constructive heuristics. We are able to state that CH^{LF2} performs better than CH^{P2} when we consider initial solutions with 20 nodes (see Table 14). CH^{P1} and CH^{LF1} heuristics yield optimal results in 22 over 24 instances, whereas CH^{LF2} heuristic yields optimal results in 15 over 24 instances. Besides, after implementing local search methods, all instances reach optimal solutions. In initial solutions with 50 nodes, WS^1 performs better by 71% (17 over 24 instances). In addition, after the initial solutions are improved, all heuristics perform equal to each other and best solution is reached in 21 over 24 instances. The performance of instances with 100 nodes is different. While WS^2 performs better in initial solutions, WS^1 obtains best solution with 92% with improved final solutions. For this reason, it is recommended to decide which heuristic to be used based on the structure of disaster areas in after disaster improvement processes. In addition, thanks to the short run times of both heuristics, results of the these two heuristics may be compared before decision making. However, in order to carry out more effective decision making during disasters, it is required to apply any of the constructive heuristics to smaller size instances then apply improvement methods (see Tables 14 and 15). It is possible to state that peer heuristic performs slightly better when only considering initial solutions of instances with 50 and 100 nodes. When we consider improved solutions, any of the proposed heuristic may be applied, since performances of the heuristics are equal to each other.

Besides, when we compare two different versions; WS^1 and WS^2, we observe that they both obtain same solutions in few instances. We observe different routes in almost 80% of the instances. In such cases, there are inter-dependencies between two

#N	#B		Constructive Heuristics			
	Min	Max	$\mathbf{CH}^{P1}$	$\mathbf{CH}^{P2}$	$\mathbf{CH}^{LF1}$	$\mathbf{CH}^{LF2}$
20	4	12	22	14	22	15
50	10	31	17	9	17	9
100	20	60	10	14	10	12

Table 14: Comparison of heuristic performances for initial solutions

#N	#B		Improvement Methods			
	Min	Max	$\mathbf{CH}^{P1}$	$\mathbf{CH}^{P2}$	$\mathbf{CH}^{LF1}$	$\mathbf{CH}^{LF2}$
20	4	12	24	24	24	24
50	10	31	22	21	22	21
100	20	60	22	19	22	19

Table 15: Comparison of heuristic performances for final solutions

networks and more number of nodes making it more important for WS^2 resulting in partial chances to the routes. This is because, the completion time of the whole system,not just own network completion time, is taken into account, leading to solutions close to the best solution. 58% of the instances with 20 nodes in bidirectional instances, WS^1 and WS^2 yield the same results. 66% of the instances in one directional instances, two version yield the same results. In 93% of the instances with 50 ve 100 nodes, these versions yield different results. In other words, results begin to differ as the number of nodes in constructive heuristic versions increases. Especially in instances of bidirectional instances with 50 and 100 nodes, WS^2 performs better. Total completion time is shorter in such instances when compared to other instances (see Tables 10 and 12).

Results of WS^2 heuristics are promising for large scale instances. Executing any sort of greedy heuristic takes a short time, less than a second. Given this information, executing both versions of our heuristic at the same time and picking the better solution may be more awarding.

Instance	Lower Bound			$\mathbf{Z}(\mathbf{CH}^{best})$	$\mathbf{Z}^{CPLEX}$
	Power	**Gas**	**Total**		
1	2.97	3.80	6.77	6.77	6.77
2	2.97	3.80	6.77	6.77	6.77
3	2.97	18.87	21.84	23.04	23.04
4	2.97	17.63	20.60	20.82	20.82
5	20.43	3.80	24.23	24.23	24.23
6	20.43	3.80	24.23	24.23	24.23
7	20.17	18.03	38.20	38.03	38.03
8	20.17	17.63	37.80	39.74	39.74
9	6.86	3.00	9.87	9.87	9.87
10	5.78	2.43	8.21	8.21	8.21
11	5.78	13.59	19.37	21.82	21.82
12	5.78	13.06	18.84	22.78	22.78
13	24.69	3.80	28.49	33.26	33.26
14	24.82	3.80	28.62	32.97	32.97
15	28.74	13.59	42.33	44.67	44.67
16	28.74	13.59	42.33	44.67	44.67
17	6.40	1.20	7.60	7.66	7.66
18	6.40	1.20	7.60	7.66	7.66
19	6.40	6.53	12.93	14.00	14.00
20	6.40	6.53	12.93	15.88	15.88
21	20.74	1.20	21.94	22.69	22.69
22	21.31	1.20	22.51	23.08	23.08
23	20.89	7.26	28.15	29.19	29.19
24	20.89	6.53	27.42	30.91	30.91

Table 16: Lower bounds of 20 nodes instances without inter-dependency consideration

Instance	Lower Bound Power	Lower Bound Gas	Lower Bound Total	Z(CH^{best})	Z^{CPLEX}
25	13.56	12.75	26.31	29.12	29.68
26	13.56	12.75	26.31	28.29	29.12
27	17.01	83.06	100.07	105.68	126.90
28	17.01	83.06	100.07	106.36	124.70
29	79.24	13.94	93.18	100.80	110.84
30	79.24	13.94	93.18	102.34	114.17
31	88.29	91.43	179.72	195.42	212.40
32	88.29	91.43	179.72	201.92	220.76
33	17.65	7.91	25.56	27.25	27.92
34	17.65	7.91	25.56	27.53	29.99
35	15.10	54.87	69.97	80.54	96.80
36	15.10	54.87	69.97	82.08	87.97
37	118.39	10.02	128.41	133.17	147.90
38	118.39	10.02	128.41	133.17	154.13
39	123.40	57.94	181.34	193.36	210.67
40	123.40	57.94	181.34	189.99	220.08
41	26.03	5.93	31.96	33.93	38.74
42	26.03	5.93	31.96	33.93	40.02
43	28.43	25.69	54.12	63.50	66.59
44	28.43	25.69	54.12	59.41	64.70
45	193.85	5.97	199.82	205.58	254.72
46	193.85	5.97	199.82	205.58	245.92
47	188.52	25.26	213.78	219.14	279.54
48	188.52	25.26	213.78	219.14	267.82

Table 17: Lower bounds of 50 nodes instances without inter-dependency consideration

Instance	Lower Bound Power	Lower Bound Gas	Lower Bound Total	Z(CH^{best})	Z^{CPLEX}
49	43.37	37.02	80.39	88.80	105.38
50	43.37	37.02	80.39	88.49	104.49
51	42.64	293.94	336.58	402.64	502.64
52	42.64	293.94	336.58	422.33	448.89
53	289.43	47.20	336.63	409.50	485.30
54	289.43	47.20	336.63	487.19	508.65
55	300.56	391.05	691.61	778.69	822.73
56	300.56	391.05	691.61	862.07	849.22
57	46.71	24.38	71.09	82.33	95.38
58	46.71	24.38	71.09	89.75	96.05
59	66.28	234.95	301.23	325.84	329.66
60	66.28	234.95	301.23	316.36	325.41
61	420.99	29.51	450.50	486.51	483.49
62	420.99	29.51	450.50	483.88	485.16
63	345.82	270.52	616.34	648.11	751.74
64	345.82	270.52	616.34	652.90	743.36
65	89.34	13.62	102.96	109.35	119.84
66	89.34	13.62	102.96	109.35	123.94
67	77.39	85.29	162.68	175.35	217.76
68	77.39	85.29	162.68	173.91	227.90
69	781.35	10.20	791.55	845.99	948.47
70	781.35	10.20	791.55	845.99	949.65
71	686.74	92.41	779.15	865.19	1107.75
72	686.74	92.41	779.15	903.75	1106.07

Table 18: Lower bounds of 100 nodes instances without inter-dependency consideration

To obtain the lower bound we assumed the following: if we consider the dependency between nodes in a infrastructure, in other words if do not consider any inter-dependency between infrastructures and carry out completion time computation, this value is lower bound. For each instance infrastructure inter-dependency is not considered in the process. As can be seen in Tables 16, 17 and 18, completion times are on average; 22 hours for the instances with 20 nodes, 110 hours for the instances with 50 nodes and 390 hours for instances with 100 nodes, without considering infrastructure inter-dependencies. Whereas in the best constructive heuristics, completion times are on average; 23 hours for the instances with 20 nodes, 116 hours for the instances with 50 nodes and 445 hours for instances with 100 nodes.

	Power		**Gas**				
	Route 1	**Route 2**	**Route 1**	**Route 2**	**Power**	**Gas**	**Total**
Coord.	12,	3, 8	14, 20	19, 17, 16	7.69	14.13	21.82
Uncoord.	12, 8	3	14, 20	19, 16, 17	9.53	15.55	25.08

Table 19: Comparison of routes with dependency consideration and without dependency consideration

In addition, some instances are run by considering separate power and gas infrastructures. The mathematical model of individual infrastructure is provided in Appendix. In other words, results of inter-dependent infrastructures are obtained in the case of independent actions. The completion time of power and gas infrastructures is calculated given that each infrastructure holds the information of damaged nodes of own infrastructure only (see Table 20). In Table 19 of instance number 11, we present the completion time of the system of both infrastructures as 26 hours, provided that power and gas infrastructures run the model separately. However, with SRPRIIN model, completion time is approximately 21 hours, when both infrastructures also consider each other (see Table 19). Completion times are calculated according to the generated routes, both in coordinated and uncoordinated cases. For instance, in a specific uncoordinated case for a power infrastructure, one of the vehicles visit nodes

12 and 8, whereas the other vehicle visits only node 3. The change in routes affects the arrival time of vehicles and the completion time. Time for node 8 is 3 hours when it is a coordinated case, however it is 4 hours in uncoordinated case. In times of natural disasters, time assessment and rapid decision making is of great importance. As can be seen from this instance, completion time of infrastructures can be shortened when there is coordination between infrastructures.

Instance	Uncoordinated			Coordinated		
	Power	Gas	Total	Power	Gas	Total
1	2.97	3.80	6.77	2.97	3.80	6.77
2	2.97	3.80	6.77	2.97	3.80	6.77
3	2.97	23.94	26.91	2.97	20.07	23.04
4	2.97	19.07	22.04	2.97	17.85	20.82
5	24.81	3.80	28.61	20.43	3.80	24.23
6	24.81	3.80	28.61	20.43	3.80	24.23
7	22.15	16.26	38.41	20.17	17.85	38.03
8	23.98	20.37	44.35	20.17	19.57	39.74
9	9.53	3.00	12.53	6.86	3.00	9.87
10	9.53	2.43	11.96	5.78	2.43	8.21
11	9.53	15.55	25.08	7.69	14.13	21.82
12	9.53	17.23	26.76	7.69	15.09	22.78
13	34.28	3.00	37.28	30.25	3.00	33.26
14	31.74	3.00	34.74	29.97	3.00	32.97
15	33.02	13.99	47.01	30.55	14.13	44.67
16	33.02	13.99	47.01	30.55	14.13	44.67
17	7.25	1.20	8.45	6.46	1.20	7.66
18	7.25	1.20	8.45	6.46	1.20	7.66
19	7.25	8.63	15.88	6.46	7.54	14.00
20	9.04	8.63	17.67	8.34	7.54	15.88
21	22.76	1.20	23.96	21.49	1.20	22.69
22	22.04	1.20	23.24	21.88	1.20	23.08
23	22.45	8.84	31.29	21.49	7.71	29.19
24	24.86	8.37	33.23	23.37	7.54	30.91

Table 20: Comparison of completion times with dependency consideration and without dependency consideration for 20 nodes instances

5.3 Solution Structures and Observations

In this section, we aim to study the structure of constructive heuristic solutions that are obtained from peer and leader-follower heuristics. We illustrate the solutions and present specific observations regarding the solutions.

Instances with 20 nodes are presented and observations regarding the solutions are supplied. Instances number 3 and 4 are one directional and bidirectional instances, respectively, and both include 10 power and 10 gas nodes. In Tables 3 and 4, we present one directional dependency matrix and bi-directional interdependency matrix for these two instances.

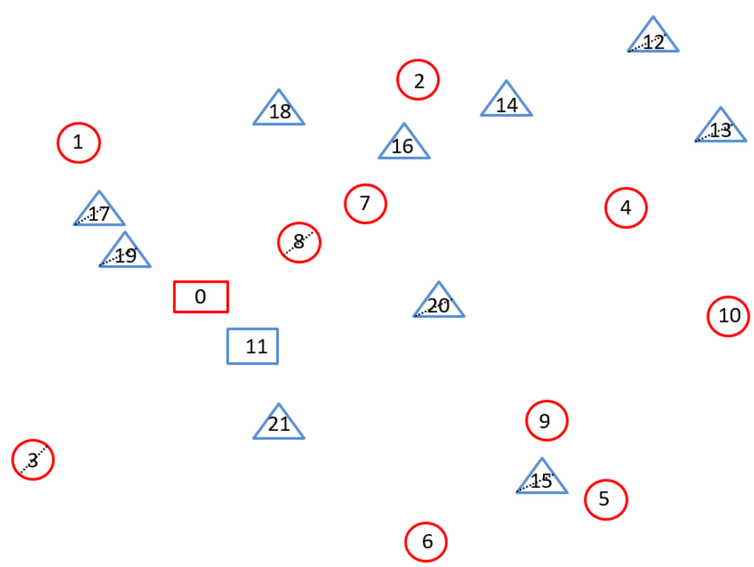

Figure 15: Illustrative example of instance 4

Routes change according to the dependency of instances. In most instances included in this study, total completion times are longer in cases that have bi-directional inter-dependency. Main cause of prolonged total completion times is the increased interaction among networks. In addition, total completion times increase, as the number of damaged nodes in a network increases since completion of repairs increase accordingly. Under same circumstances, in WS^1 and WS^2, routes begin to differ.

For instance, in Figures 15, nodes 3, 8, 12, 13, 15, 17, 19 and 20 are damaged nodes. In bi-directional inter-dependency, node 2 affects nodes 4 and 16. There is a two way dependency between nodes 5 and 9. Node 7 affects nodes 2 and 4. Damaged node 7 affects damaged node 20. Damaged node 12 affects node 14. Node 14 affects node 16. Damaged node 15 is affected by node 21 and affects node 6. Node 16 affects nodes 7, 18 and 14. Finally, damaged node 19 affects nodes 1 and 18. According to this dependency, priority list in WS^1 is 19, 12, 15, 13, 17 and 20, whereas in WS^2 it is 19, 15, 12, 13, 17 and 20. Because in WS^1, the affected infrastructure is not important and has a resulting weight of 1. In this instance, nodes 12 and 15 have weights of 1 since they effect a single node. In contrast, in WS^2 affecting another infrastructure is weighted more. While node 12 has a weight of 1, node 15 has a weight of 1.25. For this reason, gas routes in WS^1 are [19, 15, 17], [12, 13, 20],whereas in WS^2 they are [19, 12, 17], [15,13, 20]. Total completion time in WS^1 is 24 hours, while in WS^2 it is 23 hours. Priority lists that are designated in the first step of constructive heuristics are the main reason of this state of routes. Since external interaction in WS^2 is regarded more important, order of the nodes change and created routes change accordingly.

Moreover, although nodes 17 and 19 are very close to each other after initial solutions of WS^1 and WS^2, they can not be visited after each other. This is caused by the weight when considering dependency effect of both heuristic structure. Since node 17 does not affect any other node, it falls behind in the priority list. To prevent such routing inefficiencies, initial solutions are improved by employing local search methods. After the improvement process, optimal solutions are obtained even in smaller size instances such as 20 nodes.

CHAPTER VI

CONCLUSION

In this study, main focus is on the improvement of inter-dependent infrastructures that are damaged after disasters. The aim is to restore damaged nodes as soon as possible and enable the whole system back to a functioning state. Minimizing total completion time is the aim of this problem, which represents the time that it takes for all nodes in networks to be functional again. Three different assumptions are implemented in calculations for completion times of nodes in infrastructures: if a node is not damaged but dependent on another node, completion time of such node is the summation of arrival and repair time of the node that is depended on. If a node is damaged but not dependent completion time is the summation of arrival time of repair vehicles and the repair time. Finally, if a node is damaged and inter-dependent on another node, completion time is the maximum value of summation of own arrival and repair time or the summation of the arrival and repair time of the node that is dependent on. A mathematical model is developed with respect to the inter-dependency and the functionality of the infrastructures, but it is challenging and complex. After the formation of mathematical model, we develop two constructive heuristics, specifically peer and leader-follower heuristics. After development of these heuristics, we introduce two different versions of each heuristic, which are WS^1 and WS^2. Initial solutions from constructive heuristics are improved using local search improvement methods: swap, exchange and remove and insert methods. In order to test these constructive heuristics, different types of instances are created and represented. Finally, constructive and improvement heuristics solutions are compared with the solutions obtained from CPLEX.

The type and time of constructive heuristic to be used depend on the structure of nodes in the disaster area. As the number of damaged nodes and dependency between networks increase, total completion times increase due to prolonged repair times. Although there is no difference in WS^1 and WS^2 instances with 20 nodes, WS^2 performs better in instances with 50 and 100 nodes. All of the solutions from constructive heuristics are improved using the iterative implementation of neighborhood search algorithms. For instances with 20 nodes optimal solution is obtained. For instances with 50 and 100 nodes, we manage to obtain better results than CPLEX results, which are considered as benchmark solutions with optimality gaps. The worst performance among the results from local search improvement methods, actually performs 1.9% better than the solutions that are obtained from CPLEX. On the other hand, the most improved instance performs 23.69% better when compared to the solutions from CPLEX. Thanks to this better performance, we observe that local search improvement methods are very important in the improvement of these constructive heuristics. The run times of both constructive heuristics and improvement heuristics are three minutes on average. Relatively short run times of these heuristics make them very advantageous in the event of disasters, when each second matters in the process of decision making.

Given the fact that restoration of inter-dependent infrastructures after disasters is a developing research area, biggest contribution of this work is a routing model with bidirectional inter-dependencies. In the previous studies in the literature, even if inter-dependency is considered, it has been only studied as one directional inter-dependency, examining the effects of just one network on another. In addition, approaching this problem with our heuristic, which is accessible and executable in real life, may contribute to disaster coordination centers to a large extent in means of managerial insights. Testing this model in uncertain post disaster scenarios may be possible future research topic, assuming that restoration subject is a new and developing research area.

CHAPTER VII

APPENDIX

Mathematical Model for Individual Infrastructure:

Sets:

N^a: Set of all nodes on power network.

D: Set of depots.

N: Set of all non-depot nodes, $N = N^a \setminus D$.

B: Set of damaged power nodes.

K: Set of vehicles of power network.

Parameters:

$d_{ij} = 1$, if i depends on j, $\forall i, j \in N$ and 0, otherwise.

t_{ij}: travel time between i and j, $\forall i, j \in N^a$.

r_i: repair time for i, $\forall i \in B$.

Decision Variables:

$x_{ij} = 1$, if node j is visited after node i, and 0 otherwise, $\forall i, j \in B \cup D$.

c_i: completion time for $i \in N$.

a_i: arrival time at $i \in B$.

$$\text{IP:} \quad \text{Minimize} \sum_{i \in N} c_i$$

$$
\begin{aligned}
s.t. \quad & c_i \geq d_{ij}(a_j + r_j) && \forall i \in N, \forall j \in B && (13)\\
& a_i \geq \sum_{d \in D} t_{di} x_{di} && \forall i \in B, && (14)\\
& a_i + r_i + t_{ij} \leq a_j + M(1 - x_{ij}) && \forall i, j \in B, && (15)\\
& \sum_{j \in B \cup D} x_{ij} = 1 && \forall i \in B && (16)\\
& \sum_{j \in B \cup D} x_{ji} = 1 && \forall i \in B && (17)\\
& \sum_{i \in D} \sum_{j \in B} x_{ij} \leq k_p && && (18)\\
& c_i \geq 0 && \forall i \in N && (19)\\
& a_i \geq 0 && \forall i \in B && (20)\\
& x_{ij} \in \{0, 1\} && \forall i, j \in B \cup D && (21)
\end{aligned}
$$

Bibliography

[1] S. M. Rinaldi, J. P. Peerenboom, and T. K. Kelly, "Identifying, understanding, and analyzing critical infrastructure interdependencies," *IEEE Control Systems*, vol. 21, no. 6, pp. 11–25, 2001.

[2] M. Shahidehpour, Y. Fu, and T. Wiedman, "Impact of natural gas infrastructure on electric power systems," *Proceedings of the IEEE*, vol. 93, no. 5, pp. 1042–1056, 2005.

[3] H. M.-O. Ian J. Seda-Irizarry, "Puerto rico's not-so-natural disaster."

[4] B. Lu and M. Shahidehpour, "Short-term scheduling of combined cycle units," *IEEE Transactions on Power Systems*, vol. 19, no. 3, pp. 1616–1625, 2004.

[5] S. V. Buldyrev, R. Parshani, G. Paul, H. E. Stanley, and S. Havlin, "Catastrophic cascade of failures in interdependent networks," *Nature*, vol. 464, no. 7291, pp. 1025–1028, 2010.

[6] T.C. Sayıştay Başkanlığı, "Büyükşehir belediyelerinde altyapı faaliyetlerinin koordinasyonu," 2008.

[7] M. Çelik, "Network restoration and recovery in humanitarian operations: framework, literature review, and research directions," *Surveys in Operations Research and Management Science*, vol. 21, no. 2, pp. 47–61, 2016.

[8] 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*, vol. 223, no. 3, pp. 794–806, 2012.

[9] P. V. Hentenryck, N. Gillani, and C. Coffrin, “Joint assessment and restoration of power systems,” in *Proceedings of the 20th European Conference on Artificial Intelligence*, pp. 792–797, IOS Press, 2012.

[10] 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*, vol. 36, no. 2, pp. 265–284, 2007.

[11] C. Coffrin, P. Van Hentenryck, and R. Bent, “Strategic planning for power system restoration,” in *Vulnerability, Uncertainty, and Risk: Analysis, Modeling, and Management*, pp. 180–187, 2011.

[12] Z. Çağnan, R. A. Davidson, and S. D. Guikema, “Post-earthquake restoration planning for los angeles electric power,” *Earthquake Spectra*, vol. 22, no. 3, pp. 589–608, 2006.

[13] Z. Cagnan and R. A. Davidson, “Discrete event simulation of the post-earthquake restoration process for electric power systems,” *International Journal of Risk Assessment and Management*, vol. 7, no. 8, pp. 1138–1156, 2007.

[14] M. Ouyang, “Review on modeling and simulation of interdependent critical infrastructure systems,” *Reliability engineering & System safety*, vol. 121, pp. 43–60, 2014.

[15] R. Zimmerman, “Social implications of infrastructure network interactions,” *Journal of Urban Technology*, vol. 8, no. 3, pp. 97–119, 2001.

[16] D. D. Dudenhoeffer, M. R. Permann, and M. Manic, “Cims: A framework for infrastructure interdependency modeling and analysis,” in *Proceedings of the 38th conference on Winter simulation*, pp. 478–485, Winter Simulation Conference, 2006.

[17] W. A. Wallace, D. Mendonça, E. Lee, J. Mitchell, and J. Chow, "Managing disruptions to critical interdependent infrastructures in the context of the 2001 world trade center attack," *Impacts of and Human Response to the September 11, 2001 Disasters: What Research Tells Us*, 2001.

[18] P. Zhang and S. Peeta, "A generalized modeling framework to analyze interdependencies among infrastructure systems," *Transportation Research Part B: Methodological*, vol. 45, no. 3, pp. 553–579, 2011.

[19] S. V. Buldyrev, N. W. Shere, and G. A. Cwilich, "Interdependent networks with identical degrees of mutually dependent nodes," *Physical Review E*, vol. 83, no. 1, p. 016112, 2011.

[20] D. Bienstock, "Optimal control of cascading power grid failures," in *Decision and control and European control conference (CDC-ECC), 2011 50th IEEE conference on*, pp. 2166–2173, IEEE, 2011.

[21] A. Veremyev, A. Sorokin, V. Boginski, and E. L. Pasiliao, "Minimum vertex cover problem for coupled interdependent networks with cascading failures," *European Journal of Operational Research*, vol. 232, no. 3, pp. 499–511, 2014.

[22] L. Earl E., M. John E., and W. William A., "Restoration of services in interdependent infrastructure systems: A network flows approach," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 37, no. 6, pp. 1303–1317, 2007.

[23] B. Cavdaroglu, E. Hammel, J. E. Mitchell, T. C. Sharkey, and W. A. Wallace, "Integrating restoration and scheduling decisions for disrupted interdependent infrastructure systems," *Annals of Operations Research*, pp. 1–16, 2013.

[24] C. Coffrin, P. Van Hentenryck, and R. Bent, "Last-mile restoration for multiple interdependent infrastructures.," in *AAAI*, vol. 12, pp. 455–463, 2012.

[25] 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*, vol. 244, no. 1, pp. 309–321, 2015.

[26] 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*, vol. 31, no. 5, pp. 334–350, 2016.

[27] C. Liu, M. Shahidehpour, and J. Wang, "Application of augmented lagrangian relaxation to coordinated scheduling of interdependent hydrothermal power and natural gas systems," *IET generation, transmission & distribution*, vol. 4, no. 12, pp. 1314–1325, 2010.

[28] A. Gutfraind, M. Bradonjić, and T. Novikoff, "Optimal recovery of damaged infrastructure network," *arXiv preprint arXiv:1207.2799*, 2012.

[29] P. Van Hentenryck, "Computational disaster management.," in *IJCAI*, pp. 12–19, 2013.

[30] M. Drexl, "Synchronization in vehicle routinga survey of vrps with multiple synchronization constraints," *Transportation Science*, vol. 46, no. 3, pp. 297–316, 2012.

[31] D. Bredström and M. Rönnqvist, "Combined vehicle routing and scheduling with temporal precedence and synchronization constraints," *European journal of operational research*, vol. 191, no. 1, pp. 19–31, 2008.

[32] Y. Li, A. Lim, and B. Rodrigues, "Manpower allocation with time windows and job-teaming constraints," *Naval Research Logistics (NRL)*, vol. 52, no. 4, pp. 302–311, 2005.

[33] M. L. Nestel, “Electricity restored to 75 percent of customers in puerto rico: Utility.”

[34] R. Jervis, “’5 months without power’: Blackout is latest snag in puerto rico’s long recovery from hurricane maria.”

[35] A. M. Campbell, D. Vandenbussche, and W. Hermann, “Routing for relief efforts,” *Transportation Science*, vol. 42, no. 2, pp. 127–145, 2008.

VITA

Büşra Sevindik received her Bachelor of Science degree in Industrial Engineering from Özyeğin University, İstanbul, Turkey in 2015. After graduation, she joined Master of Science Program in Industrial Engineering at Özyegin University and currently studies under supervision of Assoc. Prof. Burcu Balçık and Asst. Prof. Dilek Günneç. She has worked as teaching assistant and assisted in courses: Operations Research 2, Applied Statistics, Agile Management, Engineering Economics. Her research interest mainly revolves around humanitarian relief operations. She currently works as a supply chain modeler in Solvoyo, İstanbul, Turkey.