UNIVERSIDADE FEDERAL DE SÃO CARLOS
CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO-PPGEP

THE CREW SCHEDULING AND ROUTING PROBLEM IN ROAD RESTORATION

Alfredo Daniel Moreno Arteaga

São Carlos
May 2020

UNIVERSIDADE FEDERAL DE SÃO CARLOS
CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO-PPGEP

THE CREW SCHEDULING AND ROUTING PROBLEM IN ROAD RESTORATION [1]

Alfredo Daniel Moreno Arteaga

Advisor: Prof. Dr. Pedro Munari
Co-advisor: Prof. Dr. Douglas Alem

Texto para defesa de tese apresentado ao Programa de Pós-Graduação em Engenharia de Produção da Universidade Federal de São Carlos como requisito para a obtenção do título de Doutor em Engenharia de Produção.

São Carlos
May 2020

[1]This research was supported by the State of São Paulo Research Foundation (FAPESP) through grants 2016/15966-6 and 2017/22094-8, the National Council for Scientific and Technological Development (CNPq) through grant 141973/2016-1 and CAPES-DS.

UNIVERSIDADE FEDERAL DE SÃO CARLOS

Centro de Ciências Exatas e de Tecnologia
Programa de Pós-Graduação em Engenharia de Produção

Folha de Aprovação

Assinaturas dos membros da comissão examinadora que avaliou e aprovou a Defesa de Tese de Doutorado do candidato Alfredo Daniel Moreno Arteaga, realizada em 27/03/2020:

Pedro Augusto Munari Junior

Prof. Dr. Pedro Augusto Munari Junior
UFSCar

Prof. Dr. Reinaldo Morabito Neto
UFSCar

Prof. Dr. Douglas José Alem Junior
Edin

Prof. Dr. Fabrício Oliveira
AU

Profa. Dra. Franklina Maria Bragion de Toledo
USP

Profa. Dra. Deisemara Ferreira
UFSCar

Certifico que a defesa realizou-se com a participação à distância do(s) membro(s) Reinaldo Morabito Neto, Douglas José Alem Junior, Fabrício Oliveira, Franklina Maria Bragion de Toledo, Deisemara Ferreira e, depois das arguições e deliberações realizadas, o(s) participante(s) à distância está(ao) de acordo com o conteúdo do parecer da banca examinadora redigido neste relatório de defesa.

Pedro Augusto Munari Junior

Prof. Dr. Pedro Augusto Munari Junior

Dedicated to my wife Aura Jalal
and to my mother Almiris Arteaga.

Acknowledgments

First and foremost, praises and thanks to God.
I am very much thankful to my beloved wife Aura Jalal, for her love and constant support.
I am very much thankful to my mother Almiris Arteaga, the person I admire most in this world.
I express my thanks to my brother Argemiro Moreno, for his support and help.
I wish to thank all my family and friends in Colombia, for their support and help from afar. I extend my sincere gratitude to my friends in Brazil.
I am extremely grateful to my research supervisors, Professor Pedro Munari and Professor Douglas Alem, for providing me invaluable guidance and support throughout this research work. I am also grateful to professor Michel Gendreau, for his guidance and assistance during my research internship at CIRRELT.
I wish to thank all my friends and collages from the Operations Research Group in DEP/UFSCar, Aldair Alvarez, Alyne Toscano, Amelia Stanzani, Cesar Cruz, Jonathan De la Vega, Karim Perez, Mateus Pereira, Pedro Miranda, and professor Reinaldo Morabito.
I wish to thank the post-graduate program in Production Engineering and all the people from DEP/UFSCar whose assistance was a milestone in the completion of this work.
Finally, I am so thankful to the State of São Paulo Research Foundation (FAPESP), the National Council for Scientific and Technological Development (CNPq) and CAPES-DS, for the financial support.
Many thanks to all of you.

Abstract

Extreme events as large-scale disasters can cause partial or total disruption of basic services such as water, energy, communication and transportation. In particular, recovering the transportation infrastructure is of ultimate importance in post-disaster situations, to enable the evacuation of victims and the distribution of supplies to affected areas. Road restoration, one of the main activities in this context, is a complex activity due to its inherent decisions that must be taken quickly and under uncertainty, such as the allocation of resources and the scheduling/routing of the crews that perform the restoration activities. In this thesis, we address road restoration by means of the Crew Scheduling and Routing Problem (CSRP), which integrates scheduling and routing decisions. The problem also involves the design of relief paths to connect a supply depot to demand nodes that become accessible only after the damaged nodes in these paths are repaired. We start addressing the basic variant of the CSRP, which considers a single crew available to perform the repair operations and minimizes the accessibility time of the demand nodes. Then, we extend the problem to consider multiple heterogeneous crews and uncertainties in the repair times via robust optimization. Also, we introduce the minimization of the latency of the demand nodes, where the latency of a node is defined as the accessibility time plus the travel time from the depot to that node. To solve the CSRP and the proposed extensions, effective solution methods based on Benders decomposition are proposed. We propose three types of solution approaches: branch-and-Benders-cut algorithms (BBC), metaheuristics based on simulated annealing and genetic algorithm, and hybrid branch-and-Benders-cut algorithms (HBBC). We develop two BBC algorithms. The first BBC has a master problem with scheduling decisions while the crew routing and the design of relief paths are considered in the subproblems. The second BBC considers the crew scheduling and relief path decisions in the master problem and a subproblem with the routing decisions. The metaheuristics operate on a subproblem representing the scheduling decisions and call for specialized algorithms to optimize the crew routing and the relief path decisions as well as to determine the feasibility and cost of the proposed schedule in the original CSRP. The HBBC combines the metaheuristics with the BBC algorithm. Computational experiments using instances from the literature are performed to verify the performance of the solution methods. The experiments show that the solution approaches developed so far improve the results of other exact and heuristic methods from the literature for the single crew CSRP. Computational experiments with real-world instances based on real disaster situations, such as floods and mass movements in the state of Rio de Janeiro - Brazil, are performed to validate the new proposed multicrew and robust extensions of the problem and they show that the proposed approaches are able to find good quality solutions for practical-sized instances.

Keywords: road restoration; network repair; crew scheduling and routing; robust optimization; Benders decomposition; branch-and-Benders-cut; hybrid methods; metaheuristics.

Resumo

Eventos extremos como desastres em grande escala podem causar interrupção total ou parcial de serviços básicos como água, energia, comunicação e transporte. Particularmente, a restauração da infraestrutura de transporte é de grande importância em situações pós-desastre, para permitir a evacuação das vítimas e a distribuição de suprimentos para as áreas afetadas. A restauração de estradas, uma das principais atividades nesse contexto, é uma atividade complexa devido às decisões inerentes que devem ser tomadas rapidamente e sob incerteza, como a alocação de recursos e a programação/roteamento das equipes de trabalho que devem executar as atividades de restauração. Nessa tese, é abordado o problema de programação e roteamento de equipes de trabalhos (CSRP) na restauração de estradas, o qual integra decisões de programação e roteamento. O problema considera também a definição de caminhos para conectar um depósito central com os nós de demanda, os quais tornam-se accessíveis somente após a reparação dos nós danificados nesses caminhos. Nesse trabalho é abordada inicialmente a variante básica do CSRP, a qual considera uma única equipe de trabalho disponível para executar as operações de reparo e a minimização do tempo de acessibilidade. Em seguida, o problema é estendido para considerar múltiplas equipes de trabalho heterogêneas, incertezas no tempo de reparo via otimização robusta e a minimização do *latency*, definido como o tempo de accessibilidade mais o tempo de viagem entre o deposito e os nós de demanda. Para resolver o CSRP e as extensões propostas, são desenvolvidos métodos de solução baseados em decomposição de Benders. São propostos três tipos de métodos de solução: algoritmos *branch-and-Benders-cut* (BBC), metaheurísticas baseadas em recozimento simulado e algoritmo genético e BBC híbridos (HBBC). São desenvolvidos dois algoritmos BBC. O primeiro BBC utiliza um problema mestre com decisões de programação, enquanto o roteamento das equipes de trabalho e a definição dos caminhos entre o depósito e os nós de demanda são considerados nos subproblemas. O segundo BBC considera as decisões de programação e a definição de caminhos no problema mestre e um subproblema com decisões de roteamento. As metaheurísticas operam em um subproblema que representa as decisões de programação e utilizam algoritmos especializados para otimizar o roteamento e a definição de caminhos, e assim determinar a viabilidade e o custo da programação no CSRP. O HBBC combina as metaheurísticas com os algoritmos BBC. Testes computacionais usando instâncias da literatura são realizados para verificar o desempenho dos métodos de solução. Os resultados mostram que as abordagens de solução desenvolvidas melhoram as soluções de outros métodos exatos e heurísticos propostos na literatura para o CSRP básico. Experimentos computacionais com instâncias baseadas nas inundações e movimentos de massas no estado do Rio de Janeiro - Brasil são realizados para validar as novas versões do problema e mostram que os métodos de solução desenvolvidos encontram soluções de boa qualidade para instâncias de tamanho prático.

Palavras-chave: restauração de estradas; restauração de redes; programação e roteamento de equipes de trabalho; otimização robusta; decomposição de Benders; branch-and-cut; métodos híbridos; meta-heurísticas.

List of Figures

List of Tables

Contents

Chapter 1

Introduction

Hurricanes, floods, landslides and earthquakes are examples of natural hazards that affect millions of people every year (EM-DAT, 2019). Specifically, these types of extreme events cause disruptions in the transportation infrastructure composed of roads, bridges, tunnels, etc., impeding access to affected areas. For instance, the 2010 Haiti earthquake generated more than 30 million cubic yards of debris (Booth, 2010) from damaged infrastructure, which includes the airport, seaport and roads within the country, constraining the access of the victims to relief aid (Van Wassenhove et al., 2010). Other examples of extreme events that have significantly affected road networks, thus compromising the accessibility to affected areas, are hurricanes in the southeastern region of the United States (Rawls and Turnquist, 2010), earthquakes in China (Hu et al., 2019), and floods and landslides in Rio de Janeiro State in Brazil (Moreno et al., 2018). Inaccessible affected areas result in a lack of commodities and delays in evacuation, rescue and medical assistance activities, thus causing victim suffering and loss of life. In an attempt to provide an effective emergency response in disaster aftermath, it is essential to restate the accessibility of the affected areas, which is popularly known in humanitarian logistics as road restoration (Tuzun-Aksu and Ozdamar, 2014).

Road restoration is complex due to its inherent decisions that must be taken quickly and under uncertainty, such as the allocation of resources (crews, equipment and vehicles) and the scheduling/routing of the crews that must perform the repair operations. Basically, the allocation decisions are focused on determining which crews must be used to perform the restoration of the damaged roads as well as on the selection of the damaged roads that need to be repaired. We consider that a damaged road can have one or more damaged points (damaged nodes), as may occur in real cases, especially on long highways. Scheduling decisions are important to define the sequence in which the damaged nodes in the network should be repaired by the crews, and involve also the assignment of crews to damaged nodes. The routing decisions determine the routes to be used by the crews to visit the damaged nodes and then return to the depot at the end. We are particularly interested in a problem that explicitly considers the complex interdependence between scheduling and routing decisions, hereafter called the Crew Scheduling and Routing Problem (CSRP). The objective is to restore the damaged nodes in the network as soon as possible, as they are necessary to connect a source node (depot) with demand nodes where exists some demand of humanitarian assistance. Thus, the CSRP also defines relief paths to connect the central depot with the demand nodes.

The design of crew routes is challenging because damaged nodes can obstruct the access to other nodes of the network and they are not traversable unless completely repaired first. The traversable nodes include those that were not damaged and the repaired ones. Then, the feasible routes in a specific moment depend on which nodes are damaged at that moment, which in turn depends on the scheduling decisions. Thus, the routes available for the crews change dynamically during the restoration according to the schedule. In addition, without considering routing decisions simultaneously, the damaged nodes that are not accessible at a given moment might be selected first in the schedule, making the schedule infeasible in practice.

For the mentioned reasons, crew scheduling and routing decisions on road restoration have been studied from an integrated perspective in recent studies (Maya-Duque et al., 2016; Kim et al., 2018; Shin et al., 2019). Particularly, the CSRP has been tackled recently via the proposition of mathematical programming and dynamic programing models (Maya-Duque et al., 2016). However, such models have proven to be intractable and fail to solve even some small problem instances. Hence, the authors in this field have devised heuristic methods to solve practical instances of the CSRP without an optimality guarantee or any information on the quality of the solutions. Furthermore, the solution methods proposed in literature address the CSRP considering a single crew available to perform the restoration.

In a practical context, multiple crews associated with various agencies, such as civil defense, armed forces, and firefighters, may be available to perform the repair operations. The crews consist of workforce teams equipped with heavy machinery, dozers, excavators, light vehicles, etc., and they may not have the same equipment. For example, one crew may have dozers and excavators to remove heavy debris from a blocked road, while another may have only workers using shovels. Some crews may not have enough resources (machinery, workforce, etc.) to repair some damaged nodes. Furthermore, a crew with heavy machinery may take more time to reach the damaged nodes than a crew with only light vehicles, although the former may perform a faster restoration with the help of heavy machinery. Consequently, the crews differ in the time required to repair the damaged nodes, in the travel time between nodes, and in the set of damaged nodes that they can repair. However, the consideration of multiple heterogeneous crews in the problem has been neglected in the literature because of the complexity involved in such consideration. In fact, the CSRP with a single crew is already very challenging due to the scheduling and routing decisions that must be integrated (Maya-Duque et al., 2016). In the multicrew version of the problem, an additional complexity factor is the synchronization of the crews at the damaged nodes (Akbari and Salman, 2017a,b) because these nodes cannot be traversed unless they are completely repaired, and a crew may have to wait at some damaged nodes, while another crew performs the restoration of such nodes.

The uncertainties inherent to extreme events make the aforementioned decisions even more challenging. Generally, repair times are unknown in those situations (Çelik et al., 2015) and they are critical in road restoration, mainly when the short-term response operations must be performed as soon as possible. Stochastic programming and robust optimization models can be devised to handle such uncertainties. In the stochastic programing models, it is assumed that the probability distribution of the random variables are known or can be well approximated by a finite set of realizations or scenarios. However, due to the unpredictability of extreme events, the derivation of such scenarios is particularly difficult. Furthermore, the deterministic equivalent formulations resulting from using a finite set of scenarios can lead to intractable models if the number of scenarios considered is large. For the robust optimization models, on the other hand, it is not necessary to estimate the probability distribution functions or to use scenarios to approximate such functions. In addition, these models may have similar computational tractability when compared with the deterministic version (Bertsimas and Sim, 2004; Alem et al., 2018; Munari et al., 2019). No robust model has been developed in the literature to deal with uncertainties in the CSRP or any variant of the road restoration problem.

One of the most common objective of road restoration problems is the minimization of the accessibility time, which is defined as the total time that demand nodes remain inaccessible from the central depot. However, the accessibility time in the CSRP neglect the travel time on the relief paths. This travel time can significantly impact the actual time at which the demand nodes are reached from the central depot. Recently, a new objective called "latency" have been proposed for road restoration problems (Ajam et al., 2019), which is defined as the time at which demand nodes are reached from the central depot. Thus, latency incorporates accessibility and travel times simultaneously. Although the latency has been recently proposed for other variant of road restoration problems, this objective has not yet been considered in the CSRP.

1.1 Objectives

The main objective of this thesis is to develop mathematical formulations and solution methods for the basic variant of the CSRP as well as for more realistic variants with multiple heterogeneous crews, uncertain repair times and latency objective function. In order to achieve this general objective, the following specific objectives must be attained:

- Study the CSRP with a single crew (SCSRP) and develop effective exact and heuristic solution approaches based on Benders decomposition to solve it.
- Extend the SCSRP models and solution approaches to consider multiple heterogeneous crews (MCSRP).
- Extend the MCSRP formulations and solution methods to consider the latency objective function and uncertain repair times by robust optimization approaches.
- Perform computational experiments and validate the optimization models and solution methods using benchmark instances from the literature and instances based on a real disaster event.

1.2 Methodology

According to the classification proposed by Bertrand and Fransoo (2002) and Morabito and Pureza (2010), this research can be characterized as *normative axiomatic quantitative research.* The research is called *axiomatic quantitative* because it is oriented to develop models and methods of idealized problems and is called *normative* because the primary objective is to develop mathematical models and methods that represent adequately the problem and support the involved decisions. To achieve the proposed objective, the next steps are followed:

- Problem definition. This thesis focuses on the crew scheduling and routing problem in road restoration. The idea is to extend the deterministic single crew version of the problem available in the literature to consider multiple crews and uncertain repair times.
- Literature review. A literature review of studies focused on the CSRP and related road restoration problems is conducted to properly identify the research gaps.
- Formulation of the mathematical models and solution methods. Based on the gaps found in the literature review, we propose solution methods as well as new mathematical formulations considering new extension of the problem.
- Computational experiments. The proposed models and solution methods are validated using benchmark instances from the literature and instances based on a real disaster event. Computational experiments are conducted to compare the performance of the proposed solution methods and models.
- Analysis and discussion of results. The results of computational experiments are analyzed and discussed in order to present useful insights to the practice and theory.

1.3 Contributions

In this thesis, the CSRP and some extensions are addressed using solution methods based on Benders decomposition. The contributions of this thesis are described as follows:

1. We develop for the first time Branch-and-Benders-cut (BBC) methods to solve the basic variant of the CSRP. The BBC algorithms exploit the fact that when the scheduling decisions are fixed, the crew routing and relief path decisions can be easily solved. We develop two BBC algorithms. The first BBC has a master problem with scheduling decisions while the crew routing and the design of relief paths are considered in the subproblems. The second BBC considers the crew scheduling and relief path decisions in the master problem and a subproblem with the routing decisions. The resulting master problem obtained from the Benders decomposition is solved by a single search tree, exploring the generation of cuts inside the tree. Due to the discrete subproblems, standard duality theory cannot be applied to derive cuts. Therefore, different types of feasibility and optimality Benders cuts

based on particular characteristics of the problem are proposed. A total of 11 variants of the BBC algorithms were tested using different types of feasibility and optimilaty cuts, valid inequalities, and warm-start strategies.

2. We develop a graph reduction strategy to speed up the proposed solution methods. The graph reduction consists of solving the problem over a graph with a reduced number of nodes, thus deriving lower bounds for the variables of the original problem. It relies on the elimination of intersection nodes and arcs that are not directly connected to either damaged or demand nodes.

3. We propose metaheuristics based on genetic algorithm (GA) and simulated annealing (SA) to solve the basic variant of the CSRP. These metaheuristics operate on a subproblem representing the scheduling decisions and call for specialized algorithms to optimize the crew routing and the relief path decisions as well as to determine the feasibility and cost of the proposed schedule in the original CSRP. The proposed metaheuristics do not explicitly consider all the possible crew routes and relief paths but only the best ones of a given scheduling solution, which significantly reduces the space of solutions explored by them. They use a construction heuristic that is able to find feasible solutions for all the instances of the problem and five local search operators to diversify the search in the solution space.

4. We propose the first hybrid branch-and-Benders-cut method (HBBC) that combines the metaheuristics and the BBC to solve the basic variant of the CSRP. We use the metaheuristics not only to improve the master problem solutions but to derive Benders cuts from their neighborhood. Hybridizing Benders decomposition methods with heuristics or metaheuristics can simultaneously improve both the lower and upper bounds.

5. We introduce the heterogeneous multicrew scheduling and routing problem (MCSRP) for road restoration and develop for the first time mixed integer programming models to represent the problem. In addition, we study particular properties of the MCSRP and derive valid inequalities based on these properties. Three different formulations and their corresponding valid inequalities were developed for the MCSRP.

6. We introduce the robust multicrew scheduling and routing problem (RCSRP) in road restoration, which considers uncertain repair times. A compact formulation based on recursive equations is proposed for the problem. Additionally, we introduce a new objective called "latency" for the problem, which is based on the time at which a demand node is reached from the central depot. Therefore, the latency takes into account, for a given demand node, the time at which the demand node becomes accessible from the depot plus the travel time on the relief paths.

7. The BBC algorithms, the metaheuristics, and the proposed valid inequalities are adapted to solve the RCSRP.

8. We carried out extensive computational experiments with the developed formulations and solution methods using benchmark instances and instances based on the so-called megadisaster of the Serrana Region in Rio de Janeiro, Brazil.

1.4 Organization of the thesis

The remainder of this thesis is organized as follows. Chapter 2 describes the different variants of the CSRP. Chapter 3 reviews the relevant background literature on road restoration problems. Chapter 4 presents the first BBC approach for the single crew scheduling and routing problem. Chapter 5 develops the second BBC method that enhances the approach presented in Chapter 4. Additionally, Chapter 5 proposes genetic algorithm and simulated annealing metaheuristics and an exact hybrid BBC method that effectively combines the metaheuristics with the BBC. Chapter 6 introduces the heterogeneous multicrew scheduling and routing problem in road restoration. The main contributions of the chapter include three novel mathematical formulations and the development of valid inequalities based on some particular properties of the problem. Chapter 7 introduces the Robust Crew Scheduling and Routing Problem (RCSRP) in road restoration, which considers uncertain repair time and a new objective function based on latency. A mathematical formulation and a Benders decomposition based algorithm are developed for the RCSRP. Finally, Chapter 8 discusses the conclusion and perspectives of future researches.

Chapter 2

Problem description

In this chapter, we describe the Crew Scheduling and Routing Problem (CSRP) in road restoration. First, in Section 2.1 we describe the basic variant of the problem. Then, Section 2.2 presents a extension of the problem considering multiple heterogeneous crews while Section 2.3 describes the robust CSRP considering uncertain repair times and the latency objective function.

2.1 Single Crew Scheduling and Routing Problem (SCSRP)

After extreme events, some components of the transportation infrastructure such as roads, bridges, and tunnels can be damaged or blocked interrupting the distribution and/or evacuation operations to some affected cities. Generally, such activities are performed from or to a central depot that is a supply point previously located to respond to such situations. For instance, consider the damaged network of Figure 2.1 and assume that there is a single depot in Teresópolis. Assume also that the highways RJ-148, RJ-130 and RJ-116 are damaged or blocked. In this case, the distribution/evacuation from the depot to Nova Friburgo city cannot be performed by road transportation. Thus, we say that the city of Nova Friburgo is inaccessible from the depot. To make this city accessible, one of the highways (RJ-148, RJ-130, RJ-116) must be restored or unblocked. The goal is to restore the damaged highways as soon as possible as they are necessary to make the affected areas accessible from the depot.

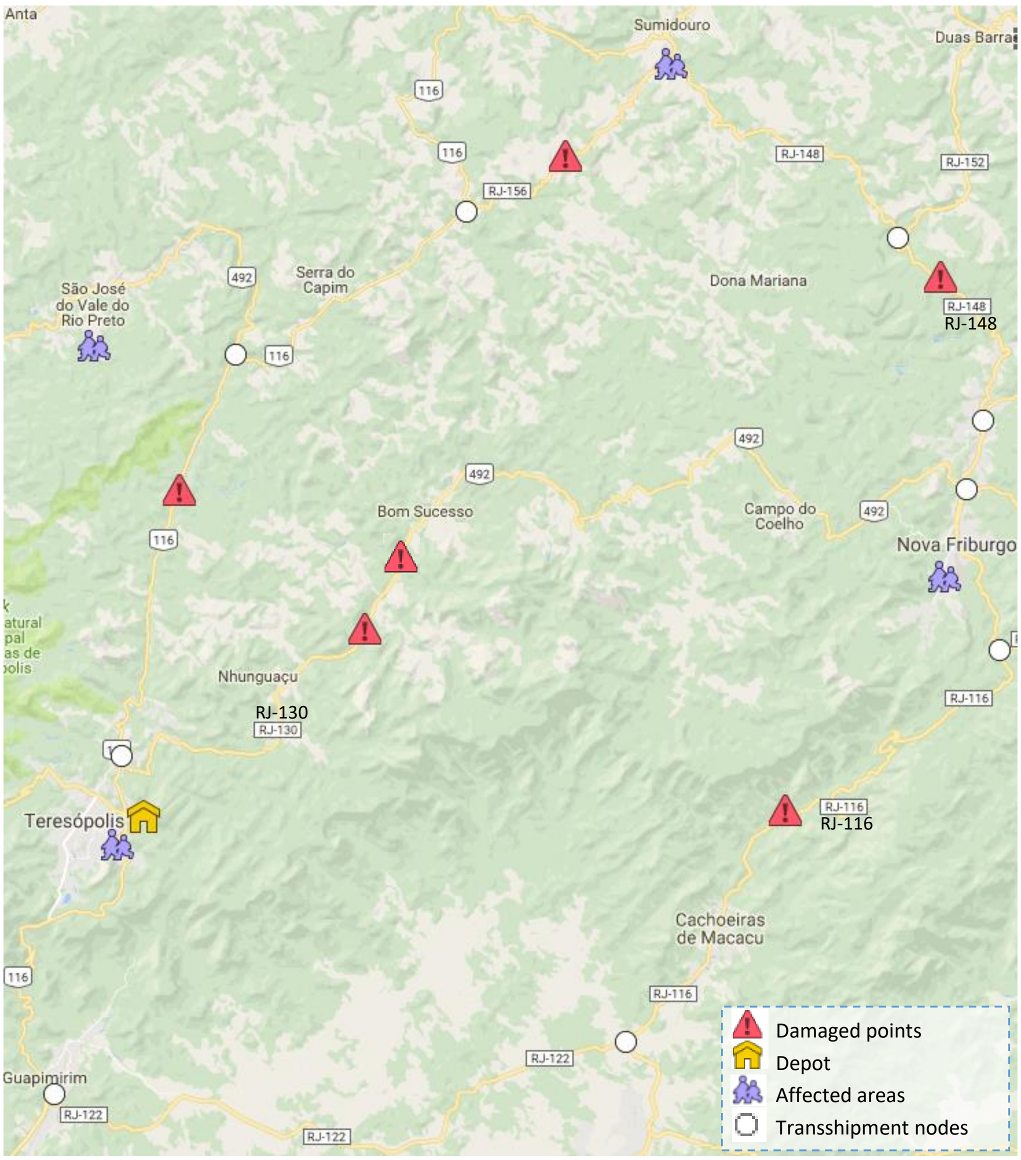


Figure 2.1: Example of a damaged network.

Formally, the SCSRP can be defined on an undirected and connected graph $G = (\mathcal{V}, \mathcal{E})$, in which $\mathcal{V}$ is the set of nodes and $\mathcal{E}$ is the set of arcs. There are demand nodes ($\mathcal{V}^d \subset \mathcal{V}$) representing the affected cities and damaged nodes ($\mathcal{V}^r \subset \mathcal{V}$) representing the damaged points in the network. In the demand nodes $i \in \mathcal{V}^d$ there is a demand d_i for humanitarian assistance. Furthermore, there may be transshipment (intersection) nodes, which represent the intersection of two or more arcs. Figure 2.2 shows a graph representation of the damaged network of Figure 2.1. The demand node 5 represents the Nova Friburgo city while the damaged nodes 14, 15, 18 and 19 represent the damaged points in the highways RJ-148, RJ-130 and RJ-116. For each node $i \in \mathcal{V}$, there is a set $\mathcal{E}_i \subseteq \mathcal{E}$ representing the arcs incident to node i. A damaged node $j \in \mathcal{V}^r$ has a repair time δ_j that represents the time the crew spends to repair the node j. A travel time τ_e and a length (distance) ℓ_e are defined for each arc $e \in \mathcal{E}$.

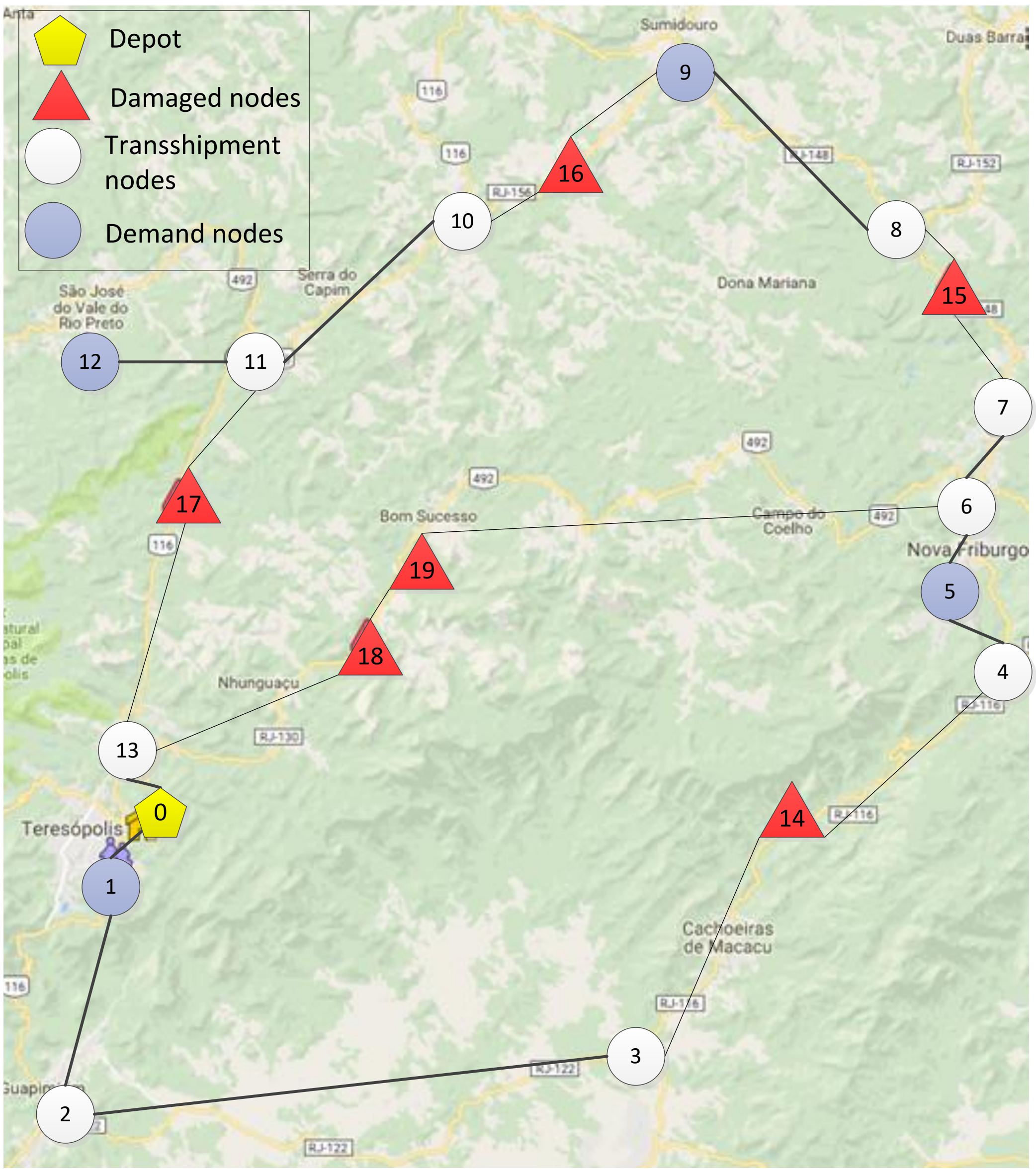


Figure 2.2: Example of a graph representing a damaged network.

The goal of the SCSRP is to minimize the time that the demand nodes remain inaccessible from the depot, weighted by their corresponding demands. The accessibility of the demand nodes influences the delivery of commodities and the evacuation of affected people. The SCSRP considers a single crew available to perform the restoration activities. The problem consists of determining (i) the optimal crew scheduling to repair the damaged nodes (crew scheduling decisions), (ii) the paths that must be followed by the crew between two successive damaged nodes in the schedule (crew routing decisions), and (iii) the paths between the depot and the demand nodes (relief path decisions). Figure 2.3 shows an example of the main decisions considered in the SCSRP highlighing the crew schedule (red dashed lines), the path followed by the crew between the damaged nodes 14 and 15 (black arrows) and the path from the depot to demand node 12 (blue arrows).

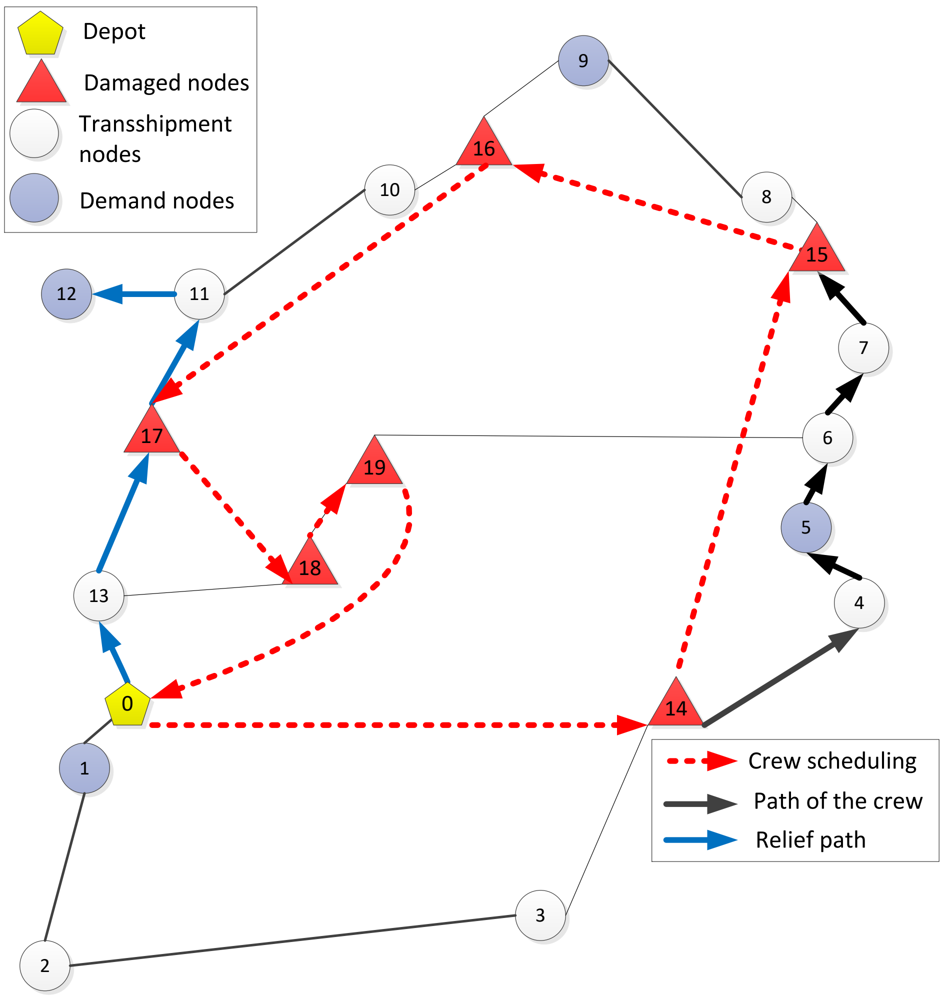


Figure 2.3: Example of the main decisions in the SCSRP.

The scheduling decisions define the sequence in which the damaged nodes in the network will be repaired by the crew. In Figure 2.3, for example, the first node to be repaired is the damaged node 14 while the last node in the sequence is the damaged node 19. The routing decisions determine the paths/routes to be used by the crew to visit and repair the damaged nodes. In this variant, a path of the crew is usually a sequence of nodes and arcs used by the crew to travel from one damaged node to another, while a route is a sequence of paths that ends at the depot after repairing all the damaged nodes. The damaged nodes must be repaired the first time they are visited by the crew, incurring in the repair time. In subsequent visits, the crew can use the already repaired damaged nodes without incurring in repair time. Some damaged nodes cannot be repaired before the restoration of other damaged nodes. For example, node 15 of the Figure 2.3 cannot be repaired directly from the depot because any path from the depot to damaged node 15 uses at least one of the other damaged nodes. Thus, a crew schedule considering node 15 as the first node to be repaired is infeasible. More than one path can be available for the crew to travel from one damaged node to the next in the sequence. In Figure 2.4, three examples of paths from damaged node 17 to damaged node 18 are presented. Notice we are assuming that the damaged nodes 14, 15 and 16 have been already repaired. The crew can travel from damaged node 17 to damaged node 18 using paths 17-13-18 (path 1), 17-11-10-16-9-8-15-7-6-19-18 (path 2), 17-11-10-16-9-8-15-7-6-5-4-14-3-2-1-0-13-18 (path 3), among others. However, path 2 is infeasible because the node 19 is not repaired yet. Hence, feasible paths between damaged nodes must include only nodes that were not damaged and/or the repaired ones.

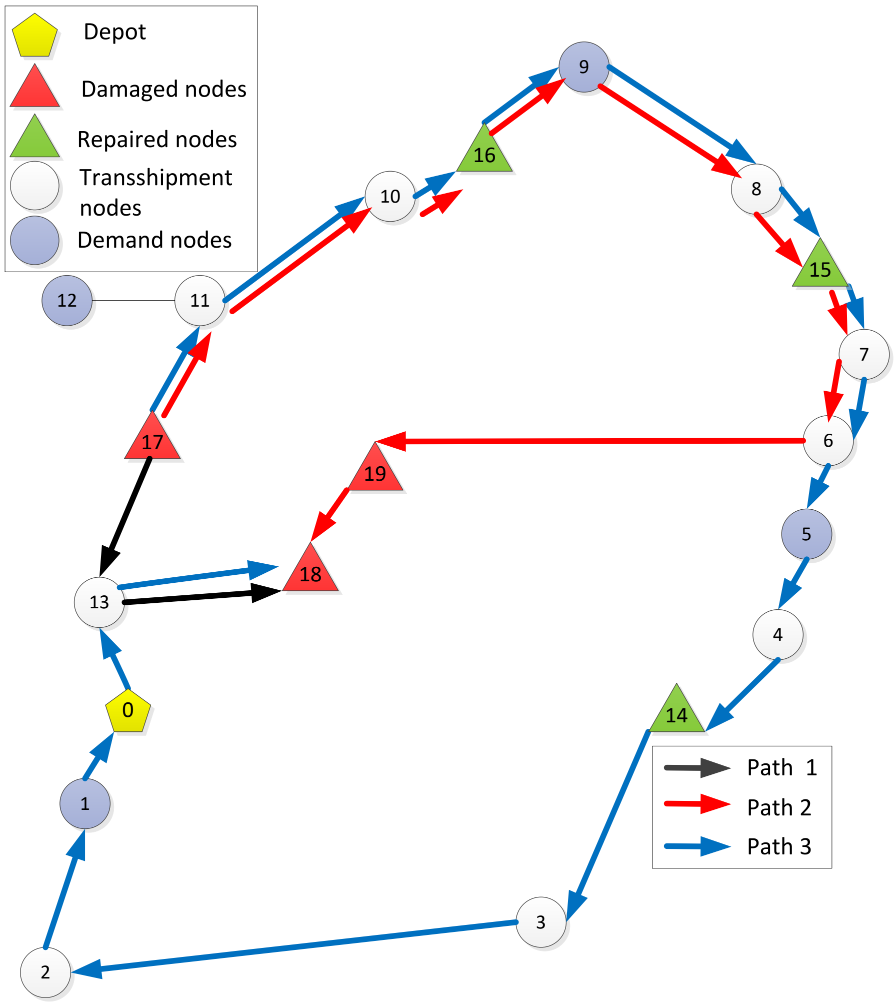


Figure 2.4: Example of paths of the crew.

The relief path decisions correspond to determine the paths that make the demand nodes accessible from the depot. A relief path is a sequence of nodes and arcs used to connect the depot with a demand node. Figure 2.5 presents an example of two relief paths between the depot and the demand node 12. A demand node $i \in \mathcal{V}^d$ is called accessible if there exists a relief path that connects this node to the depot using only undamaged and/or repaired nodes and that is not longer than a maximum distance l_i. The maximum distance l_i is based on pre-disaster conditions and has to be greater than or equal to the shortest distance between the depot and the demand node i. In Figure 2.5, for example, assuming a distance $\ell = 1$ in all the arcs of the graph, the shortest distance from the depot to demand node 12 is 4. Then, $l_{12} \geq 4$. If $l_{12} = 4$, only relief path 1 (0-13-17-11-12) can be used to connect the depot with demand node 12. In this case, relief path 2 (0-1-2-3-14-4-5-6-7-15-8-9-16-10-11-12) is infeasible. On the other hand, if $l_{12} = 15$, relief paths 2 can be used to connect the depot with demand node 12. Relief paths connecting the depot with the demand nodes can require the restoration of damaged nodes. In Figure 2.5, for example, if relief path 1 is defined for connecting node 12 with the depot, damaged node 17 must be repaired to make node 12 accessible. On the other hand, if relief path 2 is defined for connecting node 12 with the depot, then damaged nodes 14, 15 and 16 must be repaired for demand node 12 to become accessible. The time that a demand node i remains inaccessible depend on the path defined to connect it with the depot. For example, the time that demand node 12 remains inaccessible from the depot is equal to the restoration time of node 17 if the relief path 1 is used; or equal to the maximum of the restoration times of damaged nodes 14, 15 and 16 if relief path 2 is used. Assuming that $l_{12} \geq 15$, relief path 2 is better than relief path 1 to connect the depot with the demand node 12 because the damaged nodes 14, 15, 16 (used in the relief path 2) are repaired before damaged node 17 (used in the relief path 1). Notice that some demand nodes can be accessible without the restoration of damaged nodes. In Figure 2.5, for example, node 1 can be connected with the depot with path 0-1. In this case, the time that demand node 1 remain inaccessible is null. Notice also that some damaged nodes might not need to be repaired for connecting the depot with the demand nodes. However, the SCSRP considers the restoration of all the damaged nodes.

2.2 Heterogeneous Multicrew Scheduling and Routing Problem (MCSRP)

Different from the SCSRP described in the previous section, in the MCSRP a set $\mathcal{K}$ of multiple heterogeneous crews is available to perform the restoration activities. The crews are initially located in the central depot and differ in the time required to repair the damaged nodes (δ_{ki}), in the travel time on the arcs (τ_{ke}), and in the set of damaged nodes that each crew can repair. Basically, the MCSRP consists of determining (i) the paths to connect the depot to the demand nodes (relief path decisions), (ii) the assignment of crews to the damaged nodes (assignment decisions), (iii) the schedule of crews to repair the damaged nodes (scheduling decisions), and (iv) the routes of crews to repair the damaged nodes and return to the depot (routing decisions).

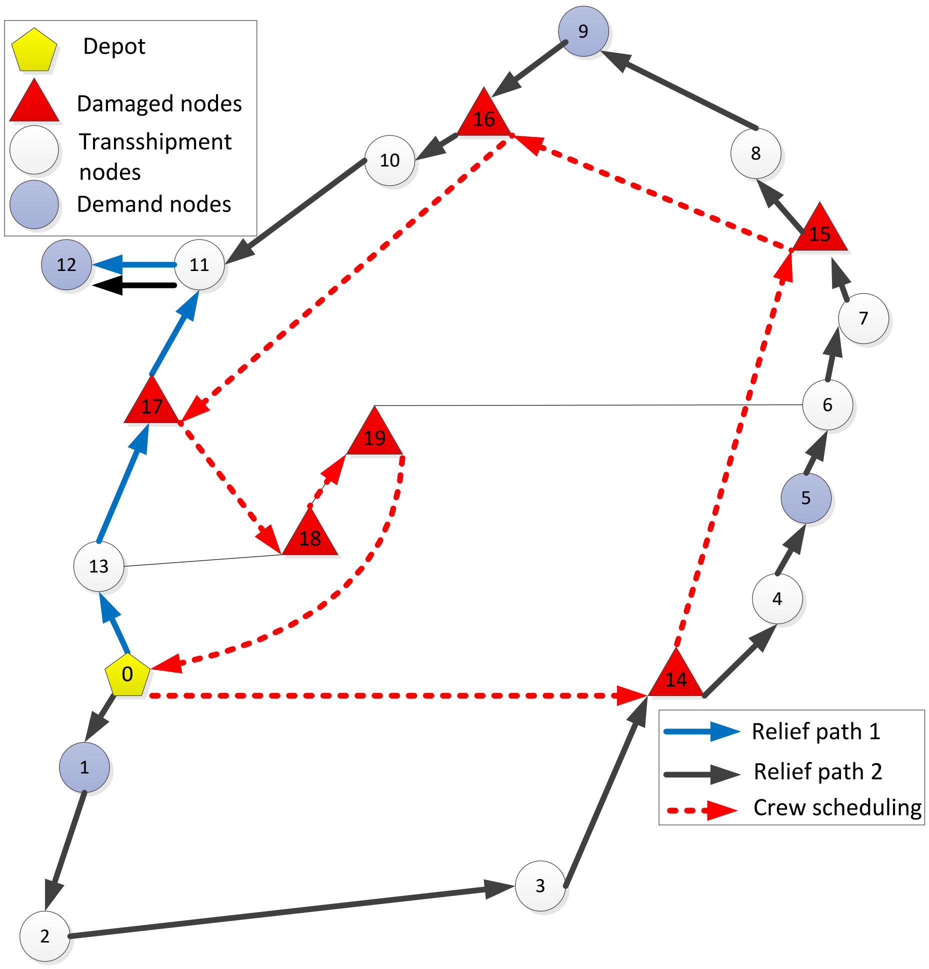


Figure 2.5: Example of relief paths.

Figure 2.6 illustrates the main decisions attributed to the MCSRP, highlighting the schedule of two crews (red arrows), the route of a crew (black arrows), and one relief path (green arrows). The scheduling decision includes the assignment of crews to damaged nodes.

The relief path decision defines the sequence of nodes and arcs used to connect the depot with the demand nodes to perform the distribution, evacuation and/or rescue operations. A given relief path connecting the depot with the demand node i is called a relief path $0-i$. For instance, Figure 2.6 shows an example of a relief path $0-2$ (green arrows), which is defined by the sequence of nodes $0 \rightarrow 1 \rightarrow 8 \rightarrow 2$. Multiple paths may be available to reach a given demand node i. For example, the sequence $0 \rightarrow 6 \rightarrow 4 \rightarrow 5 \rightarrow 2$ is an alternative relief path $0-2$. The total distance of a relief path $0-i$ must be less than or equal to a predefined maximum distance l_i^{d}. The damaged nodes used in the relief paths must be repaired by the available crews as soon as possible to minimize the time that the demand nodes remain inaccessible from the depot (accessibility time). The accessibility time of demand node i depends on the time at which the damaged nodes used in relief path $0-i$ are repaired. In Figure 2.6, for example, demand node 2 becomes accessible after the restoration of damaged node 8.

The assignment decision determines the damaged nodes that need to be repaired, and the

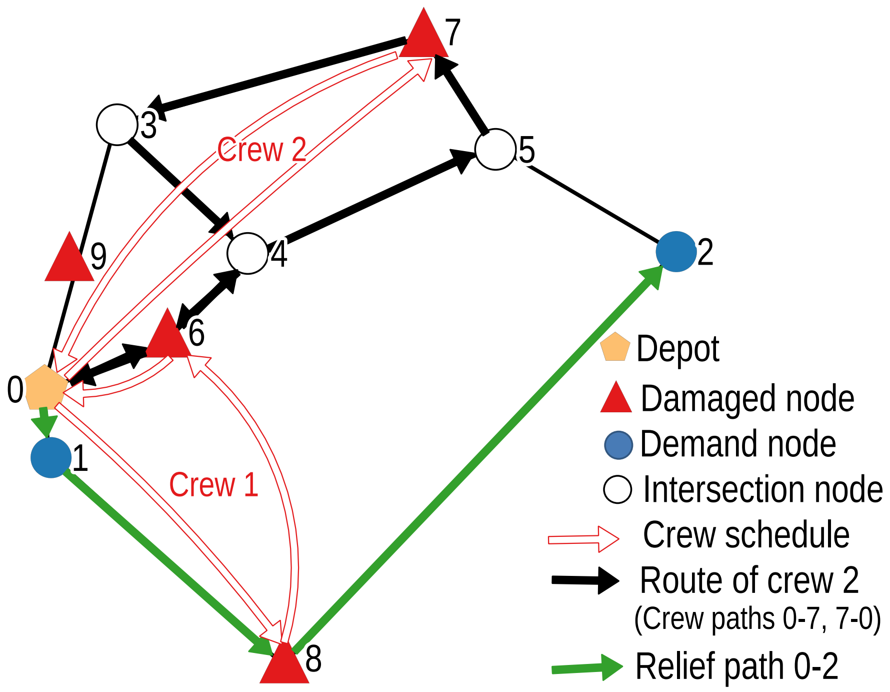


Figure 2.6: Decisions attributed to the MCSRP.

crew that must perform their restoration. The scheduling decisions define, for each crew, the repair order of the damaged nodes. Figure 2.6 shows the assignment and scheduling decisions for two crews (red arrows). Crew 1 is assigned to repair damaged nodes 6 and 8, while crew 2 must perform the restoration of damaged node 7. The schedule for the first crew is defined by the ordered set of nodes (0, 8, 6, 0). Thus, node 8 is repaired before node 6. Since the crews must depart and return to the depot, we include node 0 at the beginning and at the end of each schedule. The assignments and schedules defined for the crews may not need to include all the damaged nodes. A subset of damaged nodes may be enough to make the demand nodes accessible. For example, only damaged node 8 needs to be repaired to enable relief path $0-2$. However, solutions repairing more than the needed damaged nodes are feasible for the problem.

The routing decisions determine the paths/routes to be used by the crews to repair the damaged nodes and return to the depot. A path associated with a given crew is a sequence of nodes and arcs used by this crew to travel between two consecutive damaged nodes in its schedule. A path used by a crew to travel from node i to node j is called crew path $i-j$. Crew path $0-7$ in Figure 2.6 is defined by the sequence of nodes $0 \to 6 \to 4 \to 5 \to 7$, while crew path $7-0$ is defined by the sequence of nodes $7 \to 3 \to 4 \to 6 \to 0$. More than one path can be available for a crew to travel from one damaged node to the next in its schedule. For example, the path defined by nodes $0 \to 1 \to 8 \to 2 \to 5 \to 7$ is an alternative crew path $0-7$. For a given crew, a route is a sequence of paths that ends at the depot after repairing all the damaged nodes in its schedule. The route for crew 2 consists of crew paths $0-7$ and $7-0$. The time spent by the crews to return to the depot after repairing the last damaged node in their schedules does not affect the accessibility time of the demand nodes. Therefore, any feasible path composed of repaired damaged nodes and/or undamaged nodes can be used by the crews to return to the depot without affecting the accessibility time of the demand nodes. Nodes and edges can be traversed multiple times by the crews. A damaged node i is repaired when it is visited for the first time by crew k assigned to its restoration. In this case, crew k incurs in the repair time δ_{ki}. The crews can use the already repaired damaged nodes multiple times after their restoration without incurring extra repair time. Some damaged nodes cannot be accessed directly from the depot without the restoration of other intermediate damaged nodes. This is the case for node 7 in Figure 2.6, for example.

It is assumed that the same crew cannot restore more than one damaged node simultaneously. Similarly, more than one crew cannot repair the same damaged node. Thus, if a crew arrives at a damaged node while another crew is performing its restoration, then it has to wait until the damaged node is totally repaired. For example, crew 2 may have to wait to cross damaged node 6, which is repaired by crew 1. In this case, the time at which crew 2 can cross node 6 must be synchronized with the time at which crew 1 completes the restoration of this node.

Different from the single crew version of the problem, the synchronization of crews in the MCSRP requires the consideration of both the arrival and waiting times at each damaged node crossed in the paths of the crews. This increases the difficulty of the problem in terms of tractability of the MIP model representing the MCSRP with respect to the SCSRP given the number of additional variables and constraints that must be considered. Nevertheless, neglecting the synchronization in the MCSRP can significantly deteriorate the solutions to the problem. For instance, consider the schedules presented in Figure 2.6 and assume that crew 1 completes the restoration of nodes 8 and 6 after 2 and 4 hours, respectively. Additionally, assume that the travel time of crew 2 is 1 hour for all arcs. Figure 2.7 shows two possible paths for crew 2 to travel from node 0 to damaged node 7 with and without considering the synchronization of the crews. Path 1 is defined by nodes $0 \rightarrow 6 \rightarrow 4 \rightarrow 5 \rightarrow 7$, while path 2 is defined by nodes $0 \rightarrow 1 \rightarrow 8 \rightarrow 2 \rightarrow 5 \rightarrow 7$. When the synchronization is neglected, we assume that the damaged nodes visited in paths 1 and 2 can be used without incurring waiting time. In this case, the best path for arriving at damaged node 7 seems to be path 1, and crew 2 arrives at node 7 after 4 hours. However, when we consider the synchronization of the crews, crew 2 has a waiting time of 3 hours using path 1 because damaged node 6 can be crossed only after 4 hours. Then, using path 1 implies crew 2 arrives at node 7 after 7 hours and not after 4 hours as was wrongly determined when no waiting time was considered. In contrast, path 2 has no waiting time since damaged node 8 is already repaired when crew 2 arrives. Therefore, ignoring the synchronization implies neglecting the waiting time, which in turn leads to the selection of path 1. This strategy delays the restoration of damaged node 7 by 2 hours with respect to the selection of path 2.

2.3 Robust Crew Scheduling and Routing Problem (RCSRP)

In the RCSRP, we introduce a new latency objective function and consider uncertain repair times in the problem. We consider a new parameter that represents the travel time t_e associated to the

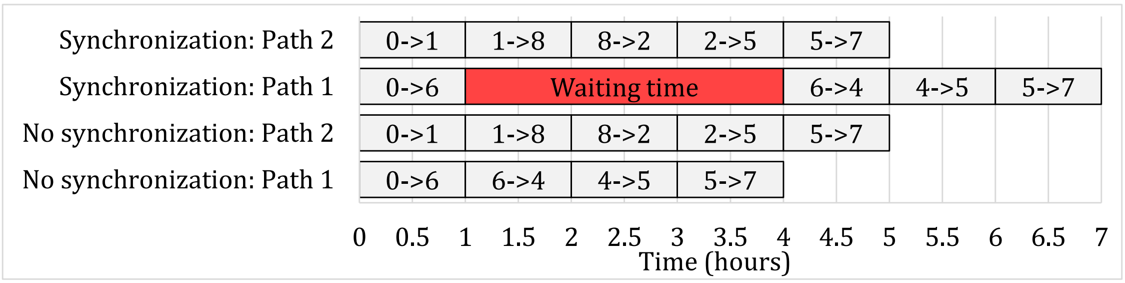
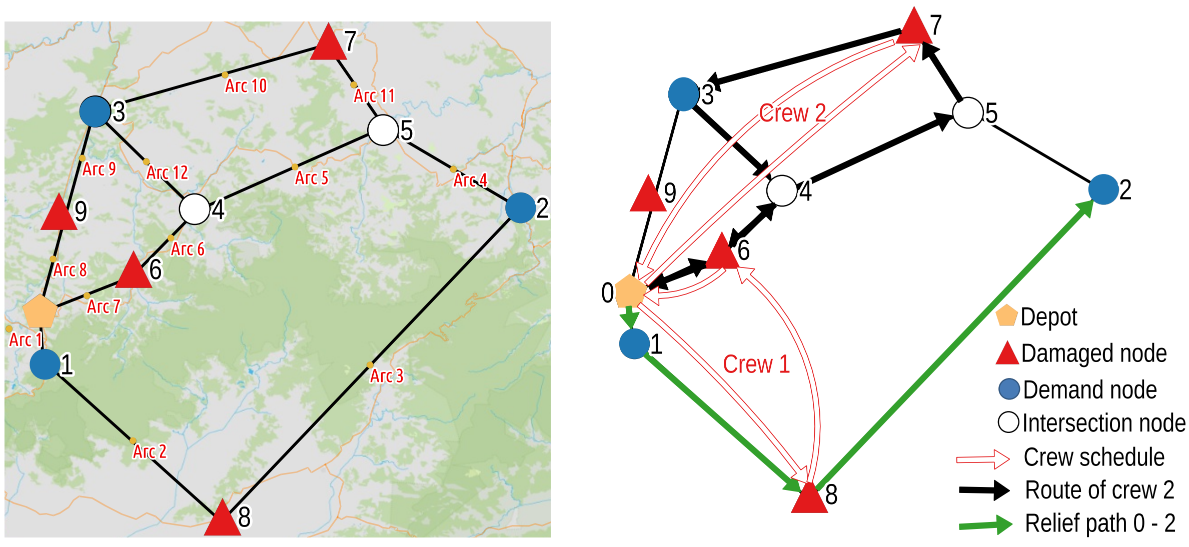


Figure 2.7: Example of the impact of the synchronization in the routing decisions.

arcs used in the relief paths. The decisions attributed to the RCSRP are the same considered for the MCSRP, as illustrated in Figure 2.8(b). As in the MCSRP, the accessibility time Z_i^{d} of demand node i depends on the time at which the damaged nodes used in relief path $0-i$ are repaired. The latency LT_i of a demand node i is defined as the sum of the travel time t_e of all the arcs in relief path $0-i$ and the accessibility time of the demand node i.



(a) Graph G.

(b) Decisions attributed to the RCSRP.

Figure 2.8: Graph G and main decisions in the RCSRP.

Multiple relief paths $0-i$ may be available to connect the depot with a demand node i. Consider the damaged network G presented in Figure 2.8(a) and the schedule of the crews presented in Figure 2.8(b). Let Z_j^{r} be the restoration time of damaged node j, i.e., time at which damaged node j is repaired and assume $t_e = 3$ for arcs 2 and 3 in Figure 2.8(a) and $t_e = 1$ for all the other arcs. Table 2.1 shows five possible paths to connect the depot with the demand node 3 and the corresponding accessibility time and latency of demand node 3 using these paths. Note that path p_1 is infeasible because it uses damaged node 9, which is not repaired by any crew. Path p_2 used damaged node 6 while path p_3 uses damaged node 6 and 7. Path p_2 is evidently better than path p_3 while path p_4 is better than path p_5 regarding the accessibility time and latency of demand node 3. Then, in our example, either p_2 or p_4 is the optimal relief path $0-3$.

Table 2.1: Example of possible relief paths $0-3$.

Relief path $0-3$	Accessibility time	Travel time	Latency
$p_1: 0 \to \mathbf{9} \to 3$	$+\infty$	$t_8 + t_9 = 2$	$+\infty$
$p_2: 0 \to \mathbf{6} \to 4 \to 3$	$Z_3^{\mathfrak{d}} = Z_6^{\mathfrak{r}}$	$t_7 + t_6 + t_{12} = 3$	$Z_6^{\mathfrak{r}} + 3$
$p_3: 0 \to \mathbf{6} \to 4 \to 5 \to \mathbf{7} \to 3$	$Z_3^{\mathfrak{d}} = \max\{Z_6^{\mathfrak{r}}, Z_7^{\mathfrak{r}}\}$	$t_7 + t_6 + t_5 + t_{11} + t_{10} = 5$	$\max\{Z_6^{\mathfrak{r}}, Z_7^{\mathfrak{r}}\} + 5$
$p_4: 0 \to 1 \to \mathbf{8} \to 2 \to 5 \to 4 \to 3$	$Z_3^{\mathfrak{d}} = Z_8^{\mathfrak{r}}$	$t_1 + t_2 + t_3 + t_4 + t_5 + t_{12} = 10$	$Z_8^{\mathfrak{r}} + 10$
$p_5: 0 \to 1 \to \mathbf{8} \to 2 \to 5 \to \mathbf{7} \to 3$	$Z_3^{\mathfrak{d}} = \max\{Z_8^{\mathfrak{r}}, Z_7^{\mathfrak{r}}\}$	$t_1 + t_2 + t_3 + t_4 + t_{11} + t_{10} = 10$	$\max\{Z_8^{\mathfrak{r}}, Z_7^{\mathfrak{r}}\} + 10$

* Damaged nodes in **bold**. The value $+\infty$ indicates that path p_1 is infeasible.

In the robust CSRP (RCSRP), the repair time $\widetilde{\delta}_{ki}$ is considered as an uncorrelated uncertain value modeled as an independent random variable that fall within the symmetric and bounded range $\widetilde{\delta}_{ki} \in [\delta_{ki} - \hat{\delta}_{ki}, \delta_{ki} + \hat{\delta}_{ki}]$, where $\hat{\delta}_{ki}$ is a positive deviation of the random variable from its corresponding nominal value δ_{ki}. This uncertainty can significantly affect the latency of the demand nodes. For instance, consider the example presented in Figure 2.8 and assume the repair times $\widetilde{\delta}_{kj}$, travel times τ_{ke} and route for crew 1 as shown in Figure 2.9. Therefore, the restoration time of damaged nodes 8 and 6 is calculated as $Z_8^{\mathfrak{r}} = \tau_{11} + \tau_{12} + \widetilde{\delta}_{18}$ and $Z_6^{\mathfrak{r}} = Z_8^{\mathfrak{r}} + \tau_{12} + \tau_{11} + \tau_{17} + \widetilde{\delta}_{16}$, respectively. Since the exact repair times are unknown, consider the three cases presented in Table 2.1 assuming different values for the repair times $\widetilde{\delta}_{18}$ and $\widetilde{\delta}_{16}$ within the ranges $[3,7]$ and $[2,6]$, respectively. The table shows the restoration time of the damaged nodes 6 and 8 in such cases and the latency for demand node 3 when using either the path p_2 ($LT_3 = Z_6^{\mathfrak{r}} + 3$) or p_4 ($LT_3 = Z_8^{\mathfrak{r}} + 10$) defined in Table 2.1.

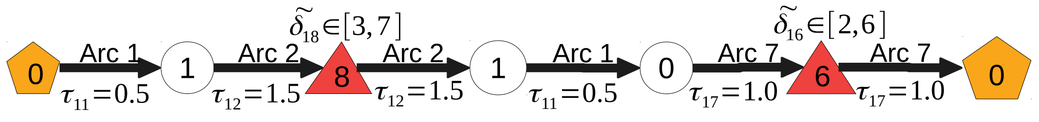


Figure 2.9: Travel times, repair times and route for crew 1.

Table 2.2: Restoration times $Z_6^{\mathfrak{r}}$ and $Z_8^{\mathfrak{r}}$ and latency for demand node 3 when considered different $\widetilde{\delta}_{kj}$ values.

			LT_3	
	$\widetilde{\delta}_j$ values	$Z_j^{\mathfrak{r}}$ values	p_2	p_4
Case 1:	$\widetilde{\delta}_{18} = 3, \widetilde{\delta}_{16} = 2;$	$Z_8^{\mathfrak{r}} = 5, Z_6^{\mathfrak{r}} = 10;$	13*	15
Case 2:	$\widetilde{\delta}_{18} = 7, \widetilde{\delta}_{16} = 6;$	$Z_8^{\mathfrak{r}} = 9, Z_6^{\mathfrak{r}} = 18;$	21	19*
Case 3:	$\widetilde{\delta}_{18} = 5, \widetilde{\delta}_{16} = 4;$	$Z_8^{\mathfrak{r}} = 7, Z_6^{\mathfrak{r}} = 14;$	17*	17*

* Optimal latency of demand node 3.

Note that the optimal relief path $0-3$ depends on the values attributed to the repair times. In case 1, the optimal relief path $0-3$ is p_2 while that for case 2 the optimal relief path $0-3$ is p_4. In case 3, on the other hand, the selection of either p_2 or p_4 results in the same latency for demand node 3. In the RCSRP, the uncertainty in the repair time does not affect the feasibility of the relief paths, i.e., the relief paths are feasible/infeasible independently of the value assumed by the repair times of the crews. However, the uncertainty in the repair times can lead to the selection of suboptimal relief paths. For example, we could select path p_2 as the optimal relief path $0-3$ aiming to have a latency equal to 13 (case 1). However, the realization of the repair times could be $\widetilde{\delta}_{18} = 7$ and $\widetilde{\delta}_{16} = 6$ leading to a latency equal to 21 (case 2).

Chapter 3

Literature review

In this chapter, we review the pertinent literature related to the CSRP. In Section 3.1, we introduce and define the scope of the literature review. In Section 3.2, we review the related problems considering a single crew. In Section 3.3, we extend the review to related road restoration problems that consider multiple crews. Finally, in Section 3.4, we summarize the review and highlight the main research gaps.

3.1 Scope of the literature review

The number of works addressing humanitarian logistics and disaster management has grown in the last years (Özdamar and Ertem, 2015; Goldschmidt and Kumar, 2016). Disaster operations in both, pre-disaster and post-disaster phases, have been widely addressed in literature. Evacuation of victims, relief distribution, road restoration, debris clearance and debris collection are some of the main post-disaster operations tackled by the authors. Road restoration is defined as the restoration of transportation infrastructure after extreme events, while debris clearance is a special application of the road restoration that consists of unblocking debris-blocked roads by pushing the debris to road sides (Çelik, 2016). Road restoration is particularly important to guarantee good service levels in the evacuation of victims and the distribution of relief aid. In fact, recent works have planned the distribution of relief supplies considering different scenarios of road destruction (Rawls and Turnquist, 2010; Noyan, 2012; Ahmadi et al., 2015; Moreno et al., 2016, 2018). Not surprisingly, scenarios with more damaged roads present lower service levels and/or higher logistic costs. Furthermore, the damaged roads can make it impossible to reach all the centers of demand (Liberatore et al., 2014).

Despite the importance of road restoration, recent surveys have pointed out the lack of studies in this area. In this direction, Altay and Green (2006) and Galindo and Batta (2013) highlighted a lack of research in problems related to recovery operations such as recovery of lifeline services, disaster debris cleanup, and restoration of roads. They pointed out the importance of such decisions to return to the normal functioning of the affected areas. Özdamar and Ertem (2015) concluded that there is a need for developing solvable models including debris transportation, road repair, relief delivery and evacuation. They claimed that the existing models were either oversimplified or too complex and unsolvable. Finally, Çelik (2016) emphasized the lack of works addressing uncertainties. The authors concluded that the uncertainty on the demand or resource requirements for restoration/clearance must be properly considered. The surveys show gaps in the consideration of uncertainties and in the development of solvable realistic models with integrated decisions for the road restoration problem.

3.2 Variants of the problem considering a single crew

The integration of crew scheduling and routing has been recently studied in the literature under the assumption of a single crew available to perform the repair operations. Sahin et al. (2016) developed a model to determine the order and route to visit critical nodes. The roads to be restored are those in the defined routes. The objective of the model is to minimize the total time spent to reach all the critical nodes. A construction heuristic based on Dijkstra's shortest-path algorithm is proposed to find the visiting order. Then, to improve the solution quality, the authors applied the 2-opt algorithm. In Berktaş et al. (2016), two mathematical models are considered with different objectives. The first model is a reformulation of the one proposed by Sahin et al. (2016). In the second model, the authors define a new objective function that consists of minimizing the weighted sum of visiting times using priorities for the critical nodes. Heuristic algorithms are proposed to obtain solutions quickly. The heuristics determine critical paths between critical nodes using Dijkstra's shortest-path algorithm and solve a simple version of the model by fixing these paths. Similarly, Kasaei and Salman (2016) developed two mathematical models as well as heuristic methods to find the schedule and route of a crew. The first model minimizes the total time to restore the connectivity of disconnected components of the network and the second one maximizes the total components connected in a given time limit. Ajam et al. (2019) adapted the models proposed by Kasaei and Salman (2016) to minimize the latency of critical nodes, where the latency of a node is defined as the travel time from the depot to that node including the repair time of the damaged roads. The authors developed a metaheuristic based on a combination of GRASP and variable neighborhood search (VNS).

The single crew scheduling and routing problem (SCSRP) integrating crew scheduling and routing with the definition of relief path has been tackled recently in the literature using exact methods and heuristics. Maya-Duque et al. (2016) developed a dynamic programming (DP) algorithm to optimally solve the SCSRP. This approach is based on the gradual addition of damaged nodes to a schedule that starts in the depot, keeping a list of states with information about the elapsed time and current location of the crew, the unrepaired damaged nodes, and the inaccessible demand nodes. However, the DP algorithm was able to solve to optimality only a few (small) instances of the problem. A mathematical formulation was also developed by the same authors, but they claimed that a direct implementation of the model in a commercial solver resulted in an intractable solution method even for small instances. Hence, they did not report computational results of using the model to solve the problem. Finally, because of the limitations regarding their exact approaches, the authors developed a metaheuristic based on GRASP to solve medium and large instances. Due to the heuristic nature of the method and the lack of lower bounds, the analysis of the quality of the solutions is compromised. Kim et al. (2018) defined a golden period for the repair operations. Then, they penalized the accessibility after the golden period at a higher rate. Additionally, they considered the minimization of the completion time of the repair operations. To solve this problem, the authors developed an ant colony algorithm. Shin et al. (2019) solved the problem with the same type of algorithm, considering additional relief goods distribution decisions in the SCSRP and minimizing the time of the relief distribution.

3.3 Variants of the problem considering multiple crews

Variants of the problem have been addressed in the literature considering multiple crews. Tzeng et al. (2000) addressed the assignment and schedule of heterogeneous crews within a fuzzy multi-objective framework. The authors considered three objectives that aim to minimize the completion time, the work-load difference between any two crews and the maximum work load of the crews. Feng and Wang (2003) integrated additional routing decisions into the problem, but considering homogeneous crews. They developed a multi-objective model to maximize the total kilometers of roads repaired, maximize the total number of lives saved, and minimize the risk of the restoration operations. They did not deal with the relief paths decisions. Furthermore, they did not consider the dynamic changes in the accessibility of the nodes in the network, i.e., some nodes cannot be used before the restoration of other nodes. To incorporate the network dynamics, Yan and Shih (2007) devised a time-space network MIP model. This formulation considers copies i' of an original node i to represent the state of this node over the time horizon. The model minimizes the completion time of the restoration. To find feasible solutions for the problem, the authors proposed a heuristic algorithm that divides the originally damaged network into several smaller networks. Each subnetwork was then solved using a commercial solver. However, even the subproblems remain unsolvable in practical time. Therefore, in another study (Yan and Shih, 2012), the same authors developed an ant colony system-based metaheuristic to solve practical instances of the problem.

Yan and Shih (2009) integrated crew scheduling and routing with relief distribution in a bi-objective model to minimize the completion time of the restoration and the time due to the relief distribution to all demand nodes. The bi-objective model was reduced to a single objective via the evaluation of a weighted objective function and then solved by a heuristic analogous to Yan and Shih (2007). Similarly, Yan et al. (2014) incorporated rescheduling repair decisions into the problem proposed by Yan and Shih (2007). Basically, they considered that backup repair crews can be dispatched to support the regular crews when subsequent events after the primary extreme event cause new damage points over the time horizon. The authors used an ant colony system-based metaheuristic to solve the problem.

Tang et al. (2009) used the idea of time-space networks to model a stochastic version of the problem presented in Yan and Shih (2007). They incorporated both stochastic travel and repair times into the problem using a two-stage stochastic programming model. The first stage refer to the scheduling and routing decisions, whereas the second stage considers alternative routing decisions for each scenario. The model aims at minimizing the travel and repair times plus an expected penalty value. Small instances of the problem were solved by a commercial optimization solver. Chang and Li (2010) considered uncertainty repair times in two-stage stochastic programming models to minimize the expected value of the total arrival time at all damaged points. They also consider that new damaged points can randomly arise in the planning horizon. The two-stage model is combined with a rolling-horizon modeling technique. Therefore, the problem is dynamically solved for different periods. In a given period, the first stage decision is the definition of a scheduling plan in accordance with the current damaged points. In the second stage, a modified schedule is determined according to the actual repair times and damaged points. To solve the problem, they proposed a integrated online algorithm with a sampling-based approximation method.

Pramudita et al. (2012) and Pramudita and Taniguchi (2014) integrated location decisions with crew scheduling and routing decisions. They considered the problem as a variant of the undirected capacitated arc routing problem (CARP), in which there exists a set of blocked arcs that need to be unblocked. Additional constraints were added to the classical CARP to limit access to some section of the network as a result of debris-blocked arcs. The objective is to minimize the cost of collecting the debris in all the damaged arcs. Pramudita and Taniguchi (2014) studied the same problem by transforming the CARP into the capacitated vehicle routing problem (CVRP). For the transformation, blocked arcs were associated with two nodes that must be visited in sequence. Pramudita et al. (2012) and Pramudita and Taniguchi (2014) used the tabu search metaheuristic to solve the problem.

Özdamar et al. (2014) proposed a multi-objective non-linear recursive model to minimize the network inaccessibility and to minimize the completion time. In this model, schedule decisions are generated for a fleet of dozers that perform the task of debris cleanup from blocked arcs. The authors developed heuristics based on priority selection rules to solve the problem. Tuzun-Aksu and Ozdamar (2014) and Çelik et al. (2015) defined models to optimally identify the schedule of blocked (damaged) arcs to be repaired. Tuzun-Aksu and Ozdamar (2014) proposed the schedule of arcs that need to be repaired by using a heuristic algorithm to divide the damaged network in smaller ones and attempting to solve more efficiently those simple networks. Çelik et al. (2015) considered additional decisions of flow of supplies and uncertainty in the debris amount of each blocked arc and, as a consequence, in the time necessary for clearance or removal operations. To model those uncertainties, a partially observable Markov decision process model is used. The authors considered arc capacity and a multi-period context. However, they relaxed this features and solve the model using an approximation heuristic. They also used specialized heuristic for partially observable Markov models, such as heuristic pruning (Ross et al., 2008). Xu and Song (2015) proposed optimizing crew scheduling and routing with relief distribution but focused on minimizing the time in which relief goods arrive at the demand nodes. The resulting problem was solved by an ant colony system-based metaheuristic.

Akbari and Salman (2017b) introduced the multi-vehicle synchronized arc routing problem. The model optimally determines the set of debris-blocked roads that need to be repaired and the synchronized routes for the crews (vehicles) to clear these roads in the shortest completion time. They proposed an MIP formulation and a relaxation-based heuristic in which the routes of the crew might not be synchronized. Additionally, they developed a constructive heuristic to obtain a feasible solution from the unsynchronized solution and a neighborhood search algorithm to improve the feasible solutions. Finally, the same problem and its solution method were addressed in Akbari and Salman (2017a) with a different objective function consisting of maximizing the network components connected to the depot node.

3.4 Summary of the review and main gaps in literature

Table 3.1 summarizes the main approaches that have been developed to model and solve the CSRP. Table 3.2 summarizes the main decisions, characteristic and objective functions of the most related problems considered in the literature.

Regarding solution methods, a few authors rely on commercial solvers to solve small instances of the problem or to solve subproblems within heuristic algorithms. Most studies use heuristic/metaheuristic algorithms, such as Tabu search, ant colony, GRASP, and specialized heuristic depending on particular characteristic of the problem. On the other hand, the literature on exact methods is still scarce. Exact methods as dynamic programming (Maya-Duque et al., 2016) and partially observable Markov decision process model (Çelik et al., 2015) were used in the literature. However, these methods fail to solve even small instances of the problem. Note that no decomposition-based exact algorithm has been proposed for the CSRP or its variants. We help to fill this gap by proposing state-of-the-art exact approaches based on Benders decomposition. We propose two exact branch-and-Benders-cut (BBC) methods and a exact hybrid BBC (HBBC) method. The HBBC combines a BBC algorithm with genetic algorithm and simulated annealing metaheuristics developed to solve the problem. The components of the proposed method are specialized and sharpened to take advantage of the mathematical structure of the CSRP. For example, the method relies on specialized algorithm to solve the problem and on particular feasibility and optimality cuts based on particular characteristic of the problem. As the proposed BBC and HBBC approaches are exact algorithms, the solution quality can be assessed, which is relevant not only from the theoretical perspective, but also in practice, as it can help decision-makers to rely on solutions that are known to be optimal or near-optimal. We show that reasonable solutions are obtained even for very large-scale instances that have never been tackled before by exact methods.

For variants of the problem addressing multiple crews, we notice that there is a lack of studies considering heterogeneous crews in road restoration problems. Furthermore, the works related to homogeneous crews lack taking into account synchronization constraints (Akbari and Salman, 2017b,a) that are inherent to the problem and/or decisions related to the definition of the relief paths connecting the source node with the demand nodes. As mentioned before, it is crucial to address such paths because they define the critically damaged nodes that must be immediately repaired to perform emergency response. In this thesis, we help to fill this gap by proposing mathematical formulations and valid inequalities for the CSRP with synchronized multiple heterogeneous crews.

Regarding the inherent uncertainty of the problem, only three of the reviewed works have focused on it (Tang et al., 2009; Chang and Li, 2010; Çelik et al., 2015), but without integrating the decisions attributed to the CSRP. Tang et al. (2009); Chang and Li (2010) and Çelik et al. (2015) did not consider relief routing or distribution decisions while Chang and Li (2010) and Çelik et al. (2015) did not consider routing decisions for the crews. Furthermore, none of these studies consider multiples heterogeneous crews available to perform the repair operations. Thus, there is a lack of studies considering uncertainty in the CSRP, even though it can strongly affect the decisions in post-disaster situations (Çelik et al., 2015; Alem et al., 2016). We help to fill this gap of the literature by considering uncertain repair times via robust optimization. In this regard, we propose a compact formulation for the problem based on recursive equations and develop a Benders decomposition based algorithm and a simulated annealing metaheuristic.

The most used objectives in the literature are the minimization of the completion time and the accessibility of the nodes with central depots. These are also the objectives addressed in the most recent works (Maya-Duque et al., 2016; Berktaş et al., 2016; Sahin et al., 2016; Akbari and Salman, 2017b,a). However, accessibility is addressed in different ways by the authors. Maya-Duque et al. (2016) minimize the time that the demand nodes remain unconnected from a central depot. Similarly, Sahin et al. (2016) and Berktaş et al. (2016) minimize the total time to visit all demand nodes from a supply node. Akbari and Salman (2017a) maximize the network components connected to the depot node. However, in the CSRP, the accessibility time neglects the travel time on the relief paths to connect the depot with the demand nodes. We help to fill this gap by incorporating the latency objective function into the problem, which minimizes both, the accessibility time and the travel time, simultaneously. The latency objective has been considered in other variant of road restoration problems (Ajam et al., 2019). However, in this case, no decision related to the definition of relief paths is integrated with the crew scheduling and routing and a single crew is considered.

Table 3.1: Solution methods and uncertainty in the CSRP and related problems.

	Commercial solver	Particular heuristic	Metaheuristic	Other methods	Uncertainty
Sahin et al. (2016)	*	*			
Berktaş et al. (2016)	*	*			
Kasaei and Salman (2016)	*	*			
Maya-Duque et al. (2016)			GRASP	Dynamic programming	
Kim et al. (2018)	*		Ant Colony		
Shin et al. (2019)	*		Ant Colony		
Ajam et al. (2019)			GRASP; VNS		
Tzeng et al. (2000)			Genetic algorithm		
Feng and Wang (2003)	*				
Yan and Shih (2007)	*	*			
Yan and Shih (2009)	*	*			
Tang et al. (2009)	*				Stochastic travel and repair time; two stage stochastic model.
Chang and Li (2010)				Sampling-based method	Stochastic repair time; two stage stochastic model.
Yan and Shih (2012)			Ant Colony		
Pramudita et al. (2012)			Tabu search		
Pramudita et al. (2014)			Tabu search		
Yan et al. (2014)	*		Ant Colony		
Özdamar et al. (2014)		*			
Tuzun-Aksu and Ozdamar (2014)	*	*			
Çelik et al. (2015)		*			Stochastic repair time; markov decision process model.
Xu and Song (2015)	*		Ant Colony		
Akbari and Salman (2017b)	*	*			
Akbari and Salman (2017a)	*	*			
This thesis	*	*	Genetic algorithm; simulated annealing	BBC; HBBC	Stochastic repair time; robust optimization.

Table 3.2: Main objective functions and decisions of the CSRP and related problems.

			Objective functions						Main decisions					
Reference	Multi-crew[1]	Synchro-nization	Accessibility	Comple-tion time	La-tency	Time distribution	Logistic cost	Other[2]	Assign-ment	Sche-duling	Rou-ting	Relief path	Distri-bution	other[3]
Sahin et al. (2016)	S		*	*						*	*			
Berktaş et al. (2016)	S		*	*				*		*	*			
Kasaei and Salman (2016)	S							*		*	*			
Maya-Duque et al. (2016)	S		*							*	*	*		
Kim et al. (2018)	S		*	*						*	*	*		
Shin et al. (2019)	S					*				*	*	*	*	
Ajam et al. (2019)	S				*					*	*			
Tzeng et al. (2000)	M-Ht			*				*	*	*				
Feng and Wang (2003)	M-Hm							*	*	*	*			
Yan and Shih (2007)	M-Hm			*					*	*	*			
Yan and Shih (2009)	M-Hm			*		*			*	*	*	*	*	
Tang et al. (2009)	M-Hm			*				*	*	*	*			
Chang and Li (2010)	M-Hm			*					*	*				
Yan and Shih (2012)	M-Hm			*					*	*	*			
Pramudita et al. (2012)	M-Hm						*		*	*				*
Pramudita et al. (2014)	M-Hm						*		*	*				*
Yan et al. (2014)	M-Hm			*					*	*	*			*
Özdamar et al. (2014)	M-Hm			*				*	*	*	*			
Tuzun-Aksu and Ozdamar (2014)	M-Hm							*	*	*				
Çelik et al. (2015)	M-Hm							*	*	*				
Xu and Song (2015)	M-Hm					*			*	*	*	*	*	
Akbari and Salman (2017b)	M-Hm	*	*	*					*	*	*			
Akbari and Salman (2017a)	M-Hm	*	*					*	*	*	*			
This thesis	S, M-Ht	*	*		*				*	*	*	*		

[1] S: single crew; M-Hm: homogeneous multi-crew; M-Ht: heterogeneous multi-crew.
[2] Work-load difference between work-troops; length of road open; risks of working in sensitive areas; unmet demand; distance.
[3] Location; Re-scheduling decisions.

Chapter 4

Branch-and-Benders-cut algorithm for the SCSRP in road restoration

This chapter presents a Benders decomposition approach for the single crew scheduling and routing problem (SCSRP). The proposed algorithm decomposes the SCSRP into a master problem (MP) with scheduling decisions and subproblems with routing and relief path decisions. This chapter is organized as follows. Section 4.1 introduces and motivates the development of the Benders decomposition algorithm for the SCSRP. In Section 4.2, we develop the Benders decomposition algorithm. Section 4.3 discusses the computational results. Finally, Section 4.4 presents final remarks and areas of future research.

*A paper based on this chapter was published in the European Journal of Operational Research (EJOR).

(Moreno et al., 2019) Moreno, Alfredo; Munari, Pedro; Alem, Douglas. A branch-and-Benders-cut algorithm for the Crew Scheduling and Routing Problem in road restoration. European Journal of Operational Research, v. 275, p. 16-34, 2019. https://doi.org/10.1016/j.ejor.2018.11.004

4.1 Introduction

Road restoration involves certain decisions that must be taken quickly, such as the selection of the roads to restore and the scheduling and routing of the crews that will perform the repair activities. We are particularly interested in a variant studied in Maya-Duque et al. (2016) that explicitly considers the complex interdependence between scheduling and routing decisions for a single crew, hereafter called the Single Crew Scheduling and Routing Problem (SCSRP) in road restoration. We consider that a damaged road can have one or more damaged points (damaged nodes), as may occur in real cases, especially on long highways. The scheduling decisions define the sequence in which the damaged nodes in the network will be visited by the crew. The routing decisions determine the paths/routes to be used by the crew to visit and repair the damaged nodes. In this variant, a path is usually a sequence of nodes and arcs used by the crew to travel from one damaged point to another, while a route is a sequence of paths that ends at the depot after repairing all the damaged nodes. The objective is to restore the damaged nodes in the network as soon as possible, because they are necessary to define paths connecting a source node to demand nodes that require humanitarian assistance.

The design of crew routes is challenging because damaged nodes can obstruct access to other nodes of the network and also damaged roads are not traversable unless they are completely repaired first. The traversable roads include those that were not damaged and the repaired ones. Then, the number of paths that are feasible at a specific moment depends on which nodes are damaged at that moment, which in turn depends on the scheduling decisions. In addition, without considering routing decisions simultaneously, the damaged nodes that are not accessible at a given moment might be selected first in the schedule, making the schedule infeasible in practice. Furthermore, the shortest paths between damaged nodes, if they exist, change dynamically during the restoration according to the schedule.

The integration of the main decisions that emerge in road restoration has been addressed by other authors in the literature (Çelik, 2016). Particularly, the SCSRP has been tackled recently via the proposition of MIP and dynamic programming models (Maya-Duque et al., 2016). However, such models have proven to be intractable and failed to solve even small instances. Hence, the authors have devised heuristic methods (Maya-Duque et al., 2016) to obtain feasible solutions for the instances of the SCSRP. As usual, the main drawback of heuristic approaches is that they do not provide optimality guarantees or any information on the quality of the solutions. Furthermore, a heuristic can stagnate in locally sub-optimal solutions.

We develop an exact algorithm based on Benders decomposition for the SCSRP. The algorithm exploits the fact that when the scheduling decisions are fixed, the routing decisions become a set of shortest-path subproblems. To solve the subproblems, we propose specialized algorithms based on Dijkstra's shortest-path algorithm. Hence, we consider a master problem (MP) with scheduling decisions and subproblems with the remaining routing decisions. The resulting MP obtained from the Benders decomposition is solved by a single search tree, exploring the generation of cuts inside the tree. This strategy has been recently referred to as

Branch-and-Benders-Cut (BBC) (Gendron et al., 2016; Errico et al., 2017) and has been shown to be more effective than the standard Benders approach, which solves a mixed-integer MP at each iteration. We are not aware of any other decomposition-based exact algorithm proposed for the SCSRP or related variants.

Due to the discrete subproblems, standard duality theory cannot be applied to derive cuts; therefore, we propose different types of lower-bounding functions and combinatorial Benders cuts (Laporte and Louveaux, 1993) based on particular characteristics of the SCSRP. Combinatorial Benders cuts avoid infeasible solutions in the MP, while lower-bounding functions set lower bounds for the feasible solutions in the MP. We empirically compare different BBC approaches based on combinations of feasibility and optimality cuts. In addition, we add valid inequalities to the MP, which helps to transfer information from the subproblems that is lost due to the decomposition. Construction and local search heuristics are also used to provide good initial solutions for the BBC.

4.2 Solution approach

In this section, we present a mathematical formulation and propose the BBC algorithm. Basically, this algorithm has three main components: an MIP master problem defined in Subsection 4.2.2, optimality and feasibility cuts defined in Subsection 4.2.3 and separation routines defined in Subsection 4.2.4. The MP considers only scheduling decisions for the crew, while subproblems determine the paths between pairs of damaged nodes and between the depot and the demand nodes. The solutions of a MP are used to generate feasibility and optimality cuts that cut off solutions corresponding to infeasible schedules. A flowchart showing the interaction between the main components of the proposed BBC algorithm is presented in Subsection 4.2.5. Additionally, in Subsection 4.2.6, we derive valid inequalities to have stronger LP relaxations, and in Subsection 4.2.8, we develop construction and local search heuristics to find good feasible solutions.

4.2.1 Mathematical modeling

To formulate the SCSRP, we closely follow the mathematical model mentioned in Maya-Duque et al. (2016). The notation used to describe the model is as follows.

Sets	
$\mathcal{V}$	Set of nodes.
$\mathcal{V}^d \subset \mathcal{V}$	Set of demand nodes.
$\mathcal{V}^r \subset \mathcal{V}$	Set of damaged nodes.
$\mathcal{E}$	Set of arcs.
$\mathcal{E}_i \subseteq \mathcal{E}$	Set of arcs incident to node $i \in \mathcal{V}$.

Parameters

d_i	Demand of node $i \in \mathcal{V}^d$.
δ_i	Repair time of node $i \in \mathcal{V}^r$.
τ_e	Travel time on arc $e \in \mathcal{E}$.
ℓ_e	Length (distance) of arc $e \in \mathcal{E}$.
l_i	Maximum distance allowed between the depot and the demand node $i \in \mathcal{V}^d$.
M	A sufficiently large number.

Decision variables

$$X_{ij} = \begin{cases} 1, \text{ if node } j \in \mathcal{V}^r \cup \{0\} \text{ is repaired immediately after node } i \in \mathcal{V}^r \cup \{0\}. \\ 0, \text{ otherwise.} \end{cases}$$

$$P_{eij} = \begin{cases} 1, \text{ if arc } e \in \mathcal{E} \text{ is used on the path from node } i \in \mathcal{V}^r \cup \{0\} \text{ to node } j \in \mathcal{V}^r \cup \{0\}. \\ 0, \text{ otherwise.} \end{cases}$$

$$N_{kij} = \begin{cases} 1, \text{ if node } k \in \mathcal{V} \text{ is used on the path from node } i \in \mathcal{V}^r \cup \{0\} \text{ to node } j \in \mathcal{V}^r \cup \{0\}. \\ 0, \text{ otherwise.} \end{cases}$$

$$Y_{ej} = \begin{cases} 1, \text{ if arc } e \in \mathcal{E} \text{ is used on the path from supply node 0 to node } j \in \mathcal{V}^d. \\ 0, \text{ otherwise.} \end{cases}$$

$$V_{kj} = \begin{cases} 1, \text{ if node } k \in \mathcal{V} \text{ is used on the path from supply node 0 to node } j \in \mathcal{V}^d. \\ 0, \text{ otherwise.} \end{cases}$$

Z_i^r	Exact time at which the damaged node $i \in \mathcal{V}^r$ is repaired.
Z_i^d	Exact time at which the demand node $i \in \mathcal{V}^d$ becomes accessible. If node i is accessible at time zero, this variable takes value zero.

Note that the variables X_{ij} define the schedule of the crew, i.e., the sequence of damaged nodes to be repaired. They do not provide the route of the crew, as they are defined for damaged nodes only. The full route is obtained from variables P_{eij} and N_{kij}, which determine the arcs and nodes, respectively, to be visited in a path between each two consecutive damaged nodes $i - j$ in the schedule of the crew. On the other hand, variables Y_{ej} and V_{kj} define the arcs and nodes, respectively, to be visited in the paths between the depot and each demand node j. These two types of variables are not related to the crew.

The model is formulated as follows:

$$\min \sum_{i \in \mathcal{V}^d} d_i \cdot Z_i^d. \tag{4.1}$$

$$\text{s.t.} \sum_{j \in \mathcal{V}^r \cup \{0\}} X_{ij} = 1, \forall\, i \in \mathcal{V}^r \cup \{0\}, \tag{4.2}$$

$$\sum_{i \in \mathcal{V}^r \cup \{0\}} X_{ij} = 1, \forall\, j \in \mathcal{V}^r \cup \{0\}, \tag{4.3}$$

$$\sum_{e \in \mathcal{E}_i} P_{eij} = X_{ij}, \forall\, i \in \mathcal{V}^r \cup \{0\},\ j \in \mathcal{V}^r, \tag{4.4}$$

$$\sum_{e \in \mathcal{E}_j} P_{eij} = X_{ij}, \forall\, i \in \mathcal{V}^r \cup \{0\},\ j \in \mathcal{V}^r, \tag{4.5}$$

$$\sum_{e \in \mathcal{E}_k} P_{eij} = 2N_{kij}, \forall\ i \in \mathcal{V}^r \cup \{0\},\ j \in \mathcal{V}^r,\ k \in \mathcal{V} \setminus \{i,\ j\}, \tag{4.6}$$

$$\sum_{e \in \mathcal{E}_0} Y_{ej} = 1, \forall\ j \in \mathcal{V}^d, \tag{4.7}$$

$$\sum_{e \in \mathcal{E}_j} Y_{ej} = 1, \forall\ j \in \mathcal{V}^d, \tag{4.8}$$

$$\sum_{e \in \mathcal{E}_k} Y_{ej} = 2V_{kj}, \forall\ j \in \mathcal{V}^d,\ k \in \mathcal{V} \setminus \{0,\ j\}, \tag{4.9}$$

$$\sum_{e \in \mathcal{E}} Y_{ej} \cdot \ell_e \leq l_j, \forall\ j \in \mathcal{V}^d, \tag{4.10}$$

$$Z_j^r \geq Z_i^r + \sum_{e \in \mathcal{E}} P_{eij} \cdot \tau_e + \delta_j - (1 - X_{ij}) \cdot M, \forall\ i \in \mathcal{V}^r \cup \{0\},\ j \in \mathcal{V}^r, \tag{4.11}$$

$$Z_j^r \geq Z_k^r + (N_{kij} - 1) \cdot M, \forall\ i \in \mathcal{V}^r \cup \{0\},\ j \in \mathcal{V}^r,\ k \in \mathcal{V}^r, \tag{4.12}$$

$$Z_i^d \geq Z_j^r + (V_{ji} - 1) \cdot M, \forall\ i \in \mathcal{V}^d,\ j \in \mathcal{V}^r, \tag{4.13}$$

$$X_{ij} \in \{0,\ 1\}, \forall\ i \in \mathcal{V}^r \cup \{0\},\ j \in \mathcal{V}^r \cup \{0\}, \tag{4.14}$$

$$P_{eij}, N_{kij} \in \{0,\ 1\}, \forall\ i \in \mathcal{V}^r \cup \{0\},\ j \in \mathcal{V}^r,\ k \in \mathcal{V},\ e \in \mathcal{E}, \tag{4.15}$$

$$Y_{ei}, V_{ki} \in \{0,\ 1\}, \forall\ i \in \mathcal{V}^d,\ k \in \mathcal{V},\ e \in \mathcal{E}, \tag{4.16}$$

$$Z_i^r \geq 0, \forall\ i \in \mathcal{V}^r \cup \{0\}, \tag{4.17}$$

$$Z_i^d \geq 0, \forall\ i \in \mathcal{V}^d. \tag{4.18}$$

The objective function (4.1) consists of minimizing the time that the demand nodes remain inaccessible from the depot, weighted by their corresponding demands. A demand node $j \in \mathcal{V}^d$ is called accessible if there exists a path that connects this node to the depot using only undamaged and/or repaired nodes and that is not longer than a maximum distance l_j – see constraints (4.10). Thus, the accessibility time of a demand node depends on the damaged nodes in its path from the depot and is computed in constraints (4.13). In Figure 4.4(a), for example, paths 0-6-2 and 0-7-5 can be defined for connecting nodes 2 and 5 with the depot, respectively. Thus, the times that the demand nodes 2 and 5 remain inaccessible from the depot are equal to the exact times at which nodes 6 and 7 are repaired, respectively. Constraints (4.2) and (4.3) specify that each damaged node must be visited once during the schedule of the crew. Constraints (4.4), (4.5) and (4.6) ensure the flow conservation in the path of the crew between damaged nodes i and j. If there is a path between damaged nodes i and j ($X_{ij} = 1$), constraints (4.4) force the use of an arc incident to node i in the path, while constraints (4.5) force the use of an arc incident to node j in the path. Furthermore, for each node k in the path from i to j ($N_{kij} = 1$), there is one arc leaving and one arc arriving at node k considered in the path, as imposed by constraints (4.6). Similarly, constraints (4.7), (4.8) and (4.9) ensure the flow conservation in the paths from the depot to the demand nodes. Constraints (4.10) prohibit the use of paths with a distance greater than the maximum distance allowed between the depot and the demand nodes. Notice that l_j considers the distances only, not travel or repair times. Constraints (4.11) define the exact time at which the damaged nodes are repaired. For a given node j, this is the result of adding the time at which the predecessor node i is repaired plus the travel time of the path from node i to node j plus the time it takes to repair node j. These constraints also act as subtour elimination constraints and are based on the Miller-Tucker-Zemlin (MTZ) formulation of the traveling salesman problem (TSP) (Miller et al., 1960), which has a number of constraints that depends polynomially on the number of nodes. They are different from the subtour elimination constraints originally used in the model cited by Maya-Duque et al. (2016), which are based on the Dantzig-Fulkerson-Johnson (DFJ) formulation of the TSP (Dantzig et al., 1954) and lead to a number of constraints that is exponential in terms of the number of nodes. To keep the model polynomial-sized, we decided to use the MTZ-based constraints. Constraints (4.12) ensure that a node k in the path from node i to node j must be repaired before node j; i.e., damaged unrepaired nodes cannot be used in a path from node i to node j. Constraints (4.13) define the exact time at which each demand node i become accessible, which is based on the time when damaged nodes in the path connecting i to the depot are repaired. Finally, constraints (4.14)-(4.18) impose the domain of the decision variables. It is worth mentioning that variables P_{eij} and Y_{ej} do not need to be defined as binary variables in the computational implementation because they naturally assume binary values if variables N_{kij} and V_{kj} are defined as binaries, respectively.

4.2.2 Benders decomposition

Benders decomposition is a variable partitioning technique whose goal is to tackle problems with complicating variables (Benders, 1962; Costa, 2005; Martins de Sá et al., 2013). Usually, a master problem considering only the complicating variables is solved, then the complicating variables are temporarily fixed, and one or more subproblems are solved. For the SCSRP, we identified as complicating variables the X_{ij} variables, which define the schedule of the crew. When the scheduling decisions (X_{ij}) are fixed, the remaining problem becomes a set of shortest-path problems, which can be efficiently solved by using specialized algorithms based on the well-known Dijkstra's shortest-path algorithm (Dijkstra, 1959). The master problem is defined as follows:

$$(MP) \qquad \min \quad \Theta, \tag{4.19}$$

$$\text{s.t.} \quad \text{Constraints (4.2), (4.3), (4.14)}, \tag{4.20}$$

$$R_j \geq R_i + 1 - |\mathcal{V}^r \cup \{0\}| \cdot (1 - X_{ij}), \forall\, i \in \mathcal{V}^r \cup \{0\},\ j \in \mathcal{V}^r, \tag{4.21}$$

$$\Theta \geq \sum_{i \in \mathcal{V}^d} d_i \cdot \theta_i, \tag{4.22}$$

$$\Theta, \theta_i, R_j \geq 0, \forall\, i \in \mathcal{V}^d,\ j \in \mathcal{V}^r \cup \{0\}. \tag{4.23}$$

Model (4.19)-(4.23) still lacks the feasibility and optimality cuts to be defined in Subsection 4.2.3. Notice that constraints (4.11), which act also as subtour elimination constraints in model (4.1)-(4.18), do not remain in the MP (they go to the subproblems because of variables Z_j^r and

P_{eij}). Thus, we add the new subtour elimination constraints (4.21) to the MP, together with the auxiliary variables R_j. R_j defines the position of damaged node j in the schedule of the crew. Variable θ_i computes the exact time at which the demand node $i \in \mathcal{V}^d$ becomes accessible, and Θ computes the value of the objective function. Initially, the lower bound for the Θ and θ_i variables is zero. When a solution is found for the MP, feasibility or optimality cuts are added, and they are likely to increase the lower bound of the Θ and/or θ_i variables. We can set a lower bound for variable Θ directly or by using the θ_i variables. Constraint (4.22) guarantees that the addition of optimality cuts setting a lower bound for the variables θ_i also sets a lower bound for the variable Θ.

The MP determines a schedule for the crew. The feasibility of this schedule for the original model (4.1)-(4.18) is verified in subproblem SP1, which obtains a set of shortest paths between consecutive nodes in the schedule of the crew:

$$(SP1)\ \min \sum_{i \in V^r} Z_i^r, \tag{4.24}$$

$$\text{s.t.}\quad \text{Constraints } (4.6), (4.12), (4.15), (4.17), \tag{4.25}$$

$$\sum_{e \in \mathcal{E}_i} P_{eij} = \widehat{X}_{ij}, \ \forall\, i \in \mathcal{V}^r \cup \{0\},\ j \in \mathcal{V}^r, \tag{4.26}$$

$$\sum_{e \in \mathcal{E}_j} P_{eij} = \widehat{X}_{ij}, \ \forall\, i \in \mathcal{V}^r \cup \{0\},\ j \in \mathcal{V}^r, \tag{4.27}$$

$$Z_j^r \geq Z_i^r + \sum_{e \in \mathcal{E}} P_{eij} \cdot \tau_e + (\widehat{X}_{ij} - 1) \cdot M + \delta_j, \ \forall\, i \in \mathcal{V}^r \cup \{0\},\ j \in \mathcal{V}^r, \tag{4.28}$$

in which $\widehat{X}_{ij}$ is a solution for the MP. For each pair of consecutive nodes $i - j$ with $\widehat{X}_{ij} = 1$ in the schedule defined by the MP, SP1 determines the shortest path with arcs and nodes defined by variables P_{eij} and N_{kij}, respectively. Indeed, for this pair $i - j$, constraints (4.28) become

$$Z_j^r \geq Z_i^r + \sum_{e \in \mathcal{E}} P_{eij} \cdot \tau_e + \delta_j$$

and hence, the objective function becomes a summation of the repair times and travel times on the traversed arcs.

SP1 may be infeasible if there is no path between two nodes $i - j$ that uses only undamaged and/or repaired nodes. In such a case, the schedule $\widehat{X}_{ij}$ provided by the MP is infeasible in the original problem (4.1)-(4.18), and feasibility cuts must be added to the MP (see Subsection 4.2.3). Otherwise, the values of the variables Z_i^r are used to calculate the total cost of the schedule in subproblem SP2, which determines the shortest paths between the depot and the demand nodes. It can be defined as follows:

$$(SP2)\qquad \min \sum_{i \in \mathcal{V}^d} d_i \cdot Z_i^d, \tag{4.29}$$

$$\text{s.t.}\quad \text{Constraints } (4.7), (4.8), (4.9), (4.10), (4.16), (4.18), \tag{4.30}$$

$$Z_i^d \geq \widehat{Z}_k^r + (V_{ki} - 1) \cdot M, \forall\, i \in \mathcal{V}^d,\ k \in \mathcal{V}^r, \tag{4.31}$$

where parameter $\widehat{Z}_k^r$ is obtained from a solution of subproblem SP1. Subproblem SP2 determines the shortest paths from the depot to each demand node $i \in \mathcal{V}^d$ with a distance length less than or equal to the maximum distance l_i. Each path is composed of arcs and nodes defined by variables Y_{ej} and V_{kj}, respectively. The exact time at which the demand node $i \in \mathcal{V}^d$ becomes accessible is used to generate optimality cuts for the MP, as defined in the next section. From subproblem SP2, we derive only optimality cuts. If subproblem SP2 is infeasible, then the original problem (4.1)-(4.18) is also infeasible because there is no path between the depot and some demand node i with a distance length less than or equal to the maximum distance l_i (it considers only the distance of each arc, not the travel times or the repair times).

Therefore, we have decomposed the decisions of the SCSRP into three parts: the MP, which determines the crew schedule; SP1, which checks whether this schedule is feasible and, if it is, obtains crew paths between each pair of damaged nodes; and SP2, which determines the paths between the depot and each demand node and the corresponding objective costs. Note that we could have defined a single subproblem by gathering subproblems SP1 and SP2 and hence evaluated both the feasibility and cost of the MP solutions simultaneously. However, having separate subproblems allows us to design efficient specialized algorithms, as presented in Subsection 4.2.4.

4.2.3 Combinatorial Benders cuts and lower-bounding functions

Every time an integer solution is found by the BBC algorithm, the separation procedures based on specialized solution methods for subproblems SP1 and SP2 seek violated feasibility or optimality cuts, and the corresponding combinatorial Benders cuts (feasibility cuts) or lower-bounding functions (optimality cuts) are added to the MP. We rely on feasibility and optimality cuts based on particular characteristics of the problem and on inequalities proposed for related problems in the literature (Hjorring and Holt, 1999; Laporte et al., 2014). Proposition 1 states feasibility cuts for the MP.

Proposition 1. *Let $K = (v_0, v_1, ..., v_{(h-1)}, v_h, ..., v_p, ..., v_{|\mathcal{V}^r|})$ be an infeasible schedule for the crew, where v_i is the ith damaged node to be repaired and $v_0 = 0$. Assume that K is obtained by solving the MP and corresponds to the solution $\widehat{X}_{v_{(i-1)}v_i} = 1$, $\forall i = 1, ..., |\mathcal{V}^r|$. For a given index $h > 0$, let $S_h = \{v_0, v_1, ..., v_{(h-1)}, v_h\}$, and assume that K is infeasible because there exists no path from node $v_{(h-1)}$ to node v_h without using at least one damaged node not yet repaired v_p, with $p > h$. Hence, the following feasibility cuts are violated and can be added to the MP:*

$$\sum_{i \in S_h \setminus \{v_h\}} \sum_{\substack{j \in S_h \setminus \{v_0\}: \\ i \neq j}} X_{ij} \leq |S_h| - 2, \tag{4.32}$$

$$\sum_{i \in S_h} \sum_{\substack{j \in S_h: \\ \widehat{X}_{ij} = 1}} X_{ij} \leq |S_h| - 2. \tag{4.33}$$

Proof. Assume that there is no feasible path from node $v_{(h-1)}$ to node v_h. Hence, there is at least one damaged node v_p, with $p > h$, that must be repaired before node v_h (otherwise, v_h cannot be reached). Let $\bar{S}_h$ be any permutation of elements of set $S_h \setminus \{v_0, v_h\}$. Every schedule containing any partial sequence $\bar{K} = (v_0, \bar{S}_h, v_h)$ is infeasible because the node v_p is not repaired before node v_h. Then, all the schedules that contain any partial sequence $\bar{K}$ must be avoided. Every partial sequence $\bar{K}$ can be represented in the MP by binary variables in the left-hand side of (4.32), where $|S_h| - 1$ of them takes a value of 1. Therefore, to avoid any sequence $\bar{K}$, it is necessary to restrict the left-hand side of (4.32) to be strictly smaller than $|S_h| - 1$. The cut defined in (4.33) is a particular case of cut (4.32) to avoid any schedule with the sequence $\bar{K} = S_h$. $\square$

To illustrate Proposition 1, consider the schedule $K = \{v_0, v_1, v_2, v_3, v_4, v_5\} = \{0, 3, 1, 2, 4, 5\}$ that is assumed to be infeasible because there is no path from node 1 to node 2 without using node 5. Then, $v_h = v_3 = 2$ and $S_h = S_3 = \{0, 3, 1, 2\}$. The possible permutations of set $S_h \setminus \{v_0, v_h\}$ are $\bar{S}_h^1 = \{3, 1\}$ and $\bar{S}_h^2 = \{1, 3\}$. Thus, all the schedules that contain the partial sequences $\bar{K}^1 = \{0, 3, 1, 2\}$ and $\bar{K}^2 = \{0, 1, 3, 2\}$ must be avoided. The feasibility cut (4.32) is $X_{03} + X_{01} + X_{02} + X_{13} + X_{12} + X_{23} + X_{21} + X_{31} + X_{32} \leq 2$, where sequence $\bar{K}^1$ is represented by variables X_{03}, X_{31}, and X_{12} and sequence $\bar{K}^2$ is represented by variables X_{01}, X_{13}, and X_{12}. Each sequence is represented by three binary variables taking a value of 1, so to avoid the schedules with the infeasible sequences $\bar{K}^1$ and $\bar{K}^2$, we need to force these variables to sum to less than 3. The feasibility cut (4.33) considering only the sequence S_h is $X_{03} + X_{31} + X_{12} \leq 2$.

Note that the cut defined in (4.32) avoids all schedules with a partial sequence starting at node v_0, ending at node v_h, and containing nodes from set $S_h \setminus \{v_0, v_h\}$ (in any order). Thus, it cuts off every schedule with any partial sequence $\bar{K} = (v_0, \bar{S}_h, v_h)$. Equation (4.33) is a cut to avoid every schedule with a partial sequence starting at node v_0, ending at node v_h, and containing nodes from set S_h (in the original order), cutting off every schedule with a partial sequence $\bar{K} = S_h$. Only one of them, (4.32) or (4.33), is necessary to cut off the solution corresponding to K. However, the number of solutions cut off by (4.32) is greater than or equal to the number of solutions cut off by (4.33).

When a solution of the MP is feasible for the original model (4.1)-(4.18), optimality cuts must be added to properly set the corresponding cost. Proposition 2 defines optimality cuts for the variable Θ of the MP.

Proposition 2. *Let $L = (v_0, v_1, ... v_{(h-1)}, v_h)$ be a feasible partial sequence of damaged nodes repaired by the crew corresponding to the MP solution $\widehat{X}_{v_{(i-1)} v_i} = 1, \forall i = 1, ..., h$, where $v_0 = 0$ and v_h is the last node to be repaired to make all the demand nodes in the set $\mathcal{V}^d$ accessible. An optimality cut to be added to the MP is:*

$$\Theta \geq \widehat{\Theta} \cdot (\sum_{i=1}^{h} X_{v_{(i-1)} v_i} - (h-1)), \tag{4.34}$$

where $\widehat{\Theta}$ is the total cost computed in subproblem SP2.

Proof. All the demand nodes become accessible when node v_h in the partial sequence L is repaired, with a corresponding total cost $\widehat{\Theta}$. Hence, every schedule containing the sequence L must have a cost $\widehat{\Theta}$. The sequence L is represented by binary variables in the right-hand side of (4.34) when those h binary variables take value 1. Then, if the partial sequence L is considered in the schedule, the summation is equal to h, and we have the lower bound $\widehat{\Theta}$ for variable Θ activated in the MP, as $\Theta \geq \widehat{\Theta} \cdot (h - (h-1))$. Otherwise, if the partial sequence L is not considered in the schedule, there are $p < h$ variables taking a value of 1 in the right-hand side of (4.34), and the lower bound $\widehat{\Theta}$ cannot be activated in the MP, as we have $\Theta \geq \widehat{\Theta}(p - (h-1))$ with $p < h$. □

Cut (4.34) sets a lower bound for variable Θ only. Proposition 3 defines optimality cuts based on variables θ_i, $\forall i \in \mathcal{V}^d$.

Proposition 3. *Let $L^k = (v_0^k, v_1^k, ..., v_{(h-1)}^k, v_h^k)$ be a feasible partial sequence of damaged nodes repaired by the crew corresponding to the MP solution $\widehat{X}_{v_{(i-1)}^k v_i^k} = 1$, $\forall i = 1, \dots, h$, where $v_0^k = 0$ and v_h^k is the last node repaired to make the demand node $k \in \mathcal{V}^d$ accessible. Let $P_h^k = \{v_1^k, ..., v_{(h-1)}^k\}$. Let $\bar{P}_h^k$ be any permutation of elements of set P_h^k and $\bar{L}^k = (v_0^k, \bar{P}_h^k, v_h^k)$. Then, the following optimality multi-cuts can be added to the MP:*

$$\theta_k \geq \widetilde{\theta}_k \cdot \left(\sum_{i \in P_h^k} (X_{0i} + X_{i(v_h^k)}) + \sum_{i \in P_h^k} \sum_{\substack{j \in P_h^k: \\ i \neq j}} X_{ij} - |P_h^k| \right), \; \forall \; k \in \mathcal{V}^d, \tag{4.35}$$

$$\theta_k \geq \widehat{\theta}_k \cdot \left(X_{0v_1} + X_{(v_{h-1}^k)(v_h^k)} + \sum_{i \in P_h^k} \sum_{\substack{j \in P_h^k: \\ \widehat{X}_{ij}=1}} X_{ij} - |P_h^k| \right), \; \forall \; k \in \mathcal{V}^d, \tag{4.36}$$

where $\widehat{\theta}_k = \widehat{Z}_k^d$, $\forall \; k \in \mathcal{V}^d$, is computed in subproblem SP2 and $\widetilde{\theta}_k$ is a lower bound for variable θ_k when any partial sequence $\bar{L}^k$ is considered in the schedule. $\widetilde{\theta}_k$ can be computed as:

$$\widetilde{\theta}_k = \sum_{j \in P_h^k \cup \{v_h^k\}} \delta_j + \sum_{j \in P_h^k \cup \{v_h^k\}} t_j^*, \tag{4.37}$$

$$t_j^* = \min_{\substack{i \in P_h^k \cup \{v_0\}: \\ i \neq j}} \{t_{ij}\}, \; \forall \; j \in P_h^k \cup \{v_h^k\}, \tag{4.38}$$

where δ_j is the repair time of node j and t_{ij} is the cost of the shortest path from node i to node j considering that all nodes are repaired.

Proof. Cut (4.36) is a particular case of (4.35) to set the cost $\widehat{\theta}_k$ for variables θ_k corresponding to the original schedule L^k. For every partial sequence $\bar{L}^k = (v_0^k, \bar{P}_h^k, v_h^k)$, the demand node k becomes accessible when node v_h^k is repaired. For a given $\bar{L}^k$, we do not have the actual cost for variables θ_k. Instead, we have a valid lower bound $\widetilde{\theta}_k$. In the calculation of $\widetilde{\theta}_k$, we consider that in any sequence $\bar{L}^k$, all the nodes of set P_h^k must be repaired, and then the total repair time ($\sum_{j \in P_h^k \cup \{v_h^k\}} \delta_j$) must be computed. Additionally, the crew must arrive at all the damaged nodes in the sequence $\bar{L}^k$, and then we compute a lower bound using the minimum travel time to arrive at each node $j \in \bar{L}^k$ from any other node $i \in \bar{L}^k$ ($\sum_{j \in P_h^k \cup \{v_h^k\}} t_j^*$). As a result, $\widetilde{\theta}_k$ must be less than or equal to the actual accessibility time $\widehat{\theta}_k$. $\square$

The optimality cut (4.34) sets a lower bound (actual cost) for the total cost Θ for any schedule with the partial sequence L. Cut (4.36) is similar to (4.34) but sets a lower bound (actual cost) for variables θ_k, $\forall k \in \mathcal{V}^d$, which in turn sets a lower bound for the total cost Θ. Only one of them, either (4.34) or (4.36), is necessary to set the actual total cost for every schedule with partial sequence L. Cut (4.35) sets a valid (underestimated) lower bound for any schedule with any partial sequences $\bar{L}^k$, $\forall k \in \mathcal{V}^d$, so it sets a lower bound for more solutions in the MP than cuts (4.34) and (4.36).

4.2.4 Separation procedures

Both subproblems SP1 and SP2 can be efficiently solved via specialized methods based on Dijkstra's shortest-path algorithm (Dijkstra, 1959) instead of using the models (4.24)-(4.28) and (4.29)-(4.31), respectively. A pseudo-code of the method proposed to solve SP1 is outlined in Algorithm 1. The graph $G = (\mathcal{V}, \mathcal{E})$, a schedule $K = (v_0, v_1, ..., v_i, ..., v_{|\mathcal{V}^r|})$, the repair times δ_j, $\forall j \in \mathcal{V}^r$, and the travel times τ_e, $\forall e \in \mathcal{E}$ are used as the input of the algorithm. If SP1 is feasible for the schedule K, then the output of the algorithm is given by the optimal values for variables Z_i^r, $\forall i \in \mathcal{V}^r$. Otherwise, the algorithm indicates that the subproblem is infeasible.

Algorithm 1 Algorithm for solving SP1.

```
Input:
Graph $G = (\mathcal{V}, \mathcal{E})$;
Scheduling solution $K = (v_0, v_1, ..., v_j, ..., v_{|\mathcal{V}^r|})$;
Parameters $\delta_j$, $\forall j \in \mathcal{V}^r$, and $\tau_e$, $\forall e \in \mathcal{E}$;
Output:
If SP1 is feasible, return "Feasible SP1" and save optimal values $\widehat{Z}_j^r$, $\forall j \in \mathcal{V}^r$;
If SP1 is infeasible, return "Infeasible SP1";
 1: $C_e := \tau_e$, $\forall e \in \mathcal{E}$;
 2: $C_e := \infty$, $\forall e \in \mathcal{E}_j, j \in \mathcal{V}^r$;
 3: $\widehat{Z}_j^r := 0$, $\forall j \in \mathcal{V}^r$;
 4: for $j = 1$ to $|\mathcal{V}^r|$ do
 5:     $C_e := \tau_e + \delta_{v_j}$, $\forall e \in \mathcal{E}_{v_j}$;
 6:     Find the cost $\mathcal{C}$ of the shortest path from node $v_{j-1}$ to $v_j$;
 7:     if $\mathcal{C} < \infty$ then
 8:         $\widehat{Z}_{v_j}^r := \widehat{Z}_{v_{j-1}}^r + \mathcal{C}$;
 9:         $C_e := \tau_e$, $\forall e \in \mathcal{E}_{v_j} : e \notin \bigcup_{i=j+1}^{|\mathcal{V}^r|} \mathcal{E}_{v_i}$ ;
10:         $C_e := \infty$, $\forall e \in \mathcal{E}_{v_j} : e \in \bigcup_{i=j+1}^{|\mathcal{V}^r|} \mathcal{E}_{v_i}$ ;
11:     else
12:         return "Infeasible SP1";
13:     end if
14: end for
15: return "Feasible SP1";
```

Algorithm 1 starts by setting the cost C_e of each arc in the network as ∞ if the arc e is incident to a damaged node; otherwise, this cost is set as τ_e (lines 1 and 2). Then, iteratively and for each damaged node $v_j \in K \setminus \{v_0\}$, the cost C_e of each arc $e \in \mathcal{E}_{v_j}$ (i.e., incident to v_j) is reset as $\tau_e + \delta_{v_j}$ (line 5), and Dijkstra's algorithm is used to find the shortest path between nodes v_{j-1} and v_j (line 6). If a path between nodes v_{j-1} and v_j exists without using a damaged node (that was not repaired yet), the cost $\mathcal{C}$ of the path must be less than ∞, and the value of variable $Z^r_{v_j}$ is updated (line 8). The cost C_e of each arc incident to the damaged node v_j is also updated, as this node has been repaired (line 9).

It is important to emphasize that the arcs incident to damaged nodes not yet repaired have cost ∞. Thus, if an arc incident to node v_j is also incident to another damaged node not yet repaired, then that arc must continue with cost ∞ (line 10). If there is no path between nodes v_{j-1} and v_j without using a not yet repaired damaged node (i.e. $\mathcal{C} = \infty$), then the algorithm terminates and returns that SP1 is infeasible (line 12).

Figure 4.1 shows an example of the variation in the cost C_e in Algorithm 1 for a network with two damaged nodes, crew schedule $K = (v_0, v_1, v_2)$, $\tau_e = 1$ and $\delta_j = 2$, for all $e \in \mathcal{E}$ and $i \in \mathcal{V}^r$. Initially, the cost C_e of each arc incident to damaged nodes v_1 and v_2 is equal to ∞. In the first iteration, the cost of each arc incident to node v_1 is then reset to $C_e = \tau_e + \delta_{v_1} = 3$ and the Dijkstra's algorithm finds the shortest path between nodes v_0 and v_1. Once node v_1 is repaired, the cost of each arc incident to node v_1 is updated to $C_e = \tau_e = 1$. In the second iteration, the cost of each arc incident to node v_2 becomes $C_e = \tau_e + \delta_{v_j} = 3$ and Dijkstra's algorithm is now used to find the shortest path between nodes v_1 and v_2. Finally, after all nodes have been repaired, the cost of each arc becomes $C_e = \tau_e = 1$.

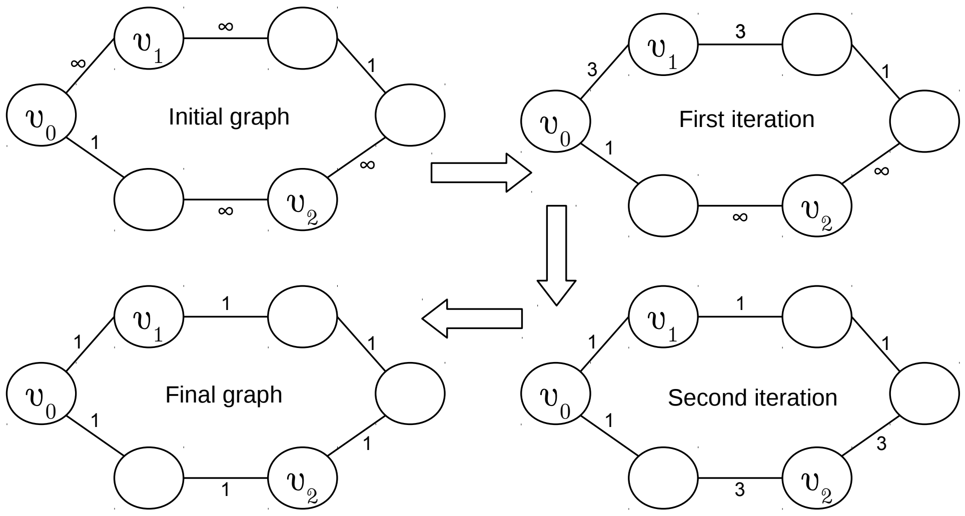


Figure 4.1: Example of the variation in arc costs in Algorithm 1.

An iterative solution approach based on solving a sequence of shortest-path problems can be used for solving subproblem SP2 as well. Recall that the goal of this subproblem is to determine the time at which affected areas become accessible. A node is accessible if there is a path from the depot to this node using only undamaged and/or repaired nodes and if the length of this path is no longer than a maximum distance l_i. If it is possible to access a demand node $i \in \mathcal{V}^d$

without using only undamaged nodes, then it becomes accessible at time $\widehat{Z}_i^d = 0$. Otherwise, let $j \in \mathcal{V}^r$ be the last damaged node that was repaired before i becomes accessible at exact time $\widehat{Z}_j^r$. Then, $\widehat{Z}_i^d = \widehat{Z}_j^r$. Notice that in subproblem SP2, given any two demand nodes i_1, i_2, the shortest path determined from the depot to i_1 is independent of the path determined from the depot to i_2. Hence, SP2 can be decomposed into $|\mathcal{V}^d|$ independent subproblems.

Algorithm 2 presents the pseudo-code of the proposed approach. The graph $G = (\mathcal{V}, \mathcal{E})$, a schedule $K = (v_0, v_1, ..., v_i, ..., v_{|\mathcal{V}^r|})$, the corresponding values of variables Z_i^r provided by the SP1, and the parameters ℓ_e, $\forall e \in \mathcal{E}$; l_i, $\forall i \in \mathcal{V}^d$; and d_i, $\forall i \in \mathcal{V}^d$ are considered the input of the algorithm. Initially, the cost C_e of each arc in the network is set as ℓ_e (line 1), the actual length (distance) of the arc. Iteratively, for each damaged node $v_j \in K \setminus \{v_0\}$, the algorithm sets the cost of each arc incident to v_j as ∞ (line 4) starting with the last damaged node ($v_{|\mathcal{V}^r|}$) in the schedule K. Then, for each demand node $i \in \mathcal{V}^d$ (line 5), Dijkstra's algorithm is used to find the shortest path from the depot to this node (line 6). If the cost $\mathcal{C}_i$ to reach this demand node is larger than the maximum allowed distance l_i (line 7), then the node v_j is necessary to find a path with cost smaller than the maximum distance l_i, and hence, the time instant $\widehat{Z}_i^d$ in which the demand node i becomes accessible is set as $\widehat{Z}_{v_j}^r$ (line 8). Note that we update $\widehat{Z}_i^d$ only if it was not updated in previous iterations; thus, $\widehat{Z}_i^d$ is equal to the largest repair time of the damaged nodes visited in the path from the depot to node i. Finally, the total cost $\widehat{\Theta}$ is computed (line 12). Recall that subproblem SP2 is always feasible if the original problem (4.1)-(4.18) is feasible.

Algorithm 2 Algorithm for solving the SP2.

```
Input:
Graph $G = (\mathcal{V}, \mathcal{E})$;
Scheduling solution $K = (v_0, v_1, ..., v_j, ..., v_{|\mathcal{V}^r|})$;
Time $\widehat{Z}_i^r$ at which damaged node $i \in \mathcal{V}^r$ is repaired;
Parameters $\ell_e$, $\forall e \in \mathcal{E}$, $l_i$, $\forall i \in \mathcal{V}^d$ and $d_i, \forall i \in \mathcal{V}^d$;
Output:
Time $\widehat{Z}_i^d$ at which the demand node $i \in \mathcal{V}^d$ becomes accessible;
Total cost $\widehat{\Theta}$;
 1: $C_e := \ell_e$, $\forall e \in \mathcal{E}$;
 2: $\widehat{Z}_i^d := 0$, $\forall i \in \mathcal{V}^d$;
 3: for $j = |\mathcal{V}^r|$ to 1 do
 4:     $C_e := \infty$, $\forall e \in \mathcal{E}_{v_j}$;
 5:     for $i = 1$ to $|\mathcal{V}^d|$ do
 6:         Find the cost $\mathcal{C}_i$ of the shortest path from the depot to the demand node $i$;
 7:         if $\mathcal{C}_i > l_i$ and $\widehat{Z}_i^d = 0$ then
 8:             $\widehat{Z}_i^d := \widehat{Z}_{v_j}^r$;
 9:         end if
10:     end for
11: end for
12: Compute total cost $\widehat{\Theta} := \sum_{i \in \mathcal{V}^d} d_i \cdot \widehat{Z}_i^d$;
```

Figure 4.2 shows an example of the variation in cost C_e in Algorithm 2 for a network with two damaged nodes, $K = (v_0, v_1, v_2)$ and $\ell_e = 1$, $\forall e \in \mathcal{E}$. In the first iteration, the cost of each arc incident to the last damaged node v_2 in the schedule is set to $C_e = \infty$, and Dijkstra's algorithm finds the shortest paths between node v_0 and each demand node $i \in \mathcal{V}^d$. If the cost $\mathcal{C}_i$ of the path between v_0 and the demand node i is larger than l_i, then the node v_2 is necessary to find a path with length smaller than l_i, and $\widehat{Z}_i^d = \widehat{Z}_{v_2}^r$. In the second iteration, the cost of each arc incident to the damaged node v_1 is updated with $C_e = \infty$, and the shortest paths between node v_0 and demand nodes $i \in \mathcal{V}^d$ are found again. In this case, if it is not possible to find a path for a demand node i with a cost $\mathcal{C}_i$ less than l_i, it needs either the node v_1 or v_2 in the path. Then, the time of accessibility $\widehat{Z}_i^d$ is equal to $\widehat{Z}_{v_1}^r$ if this was not updated in the past iteration with $\widehat{Z}_{v_2}^r$.

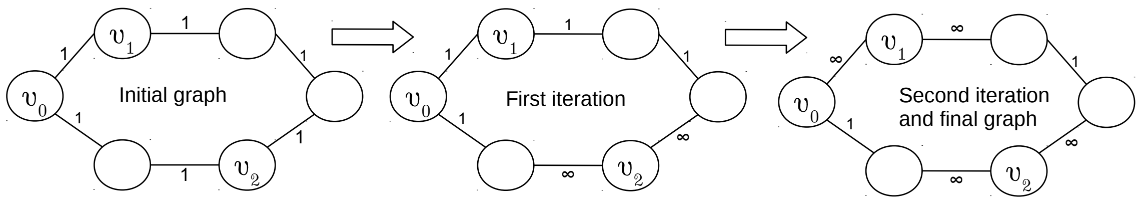
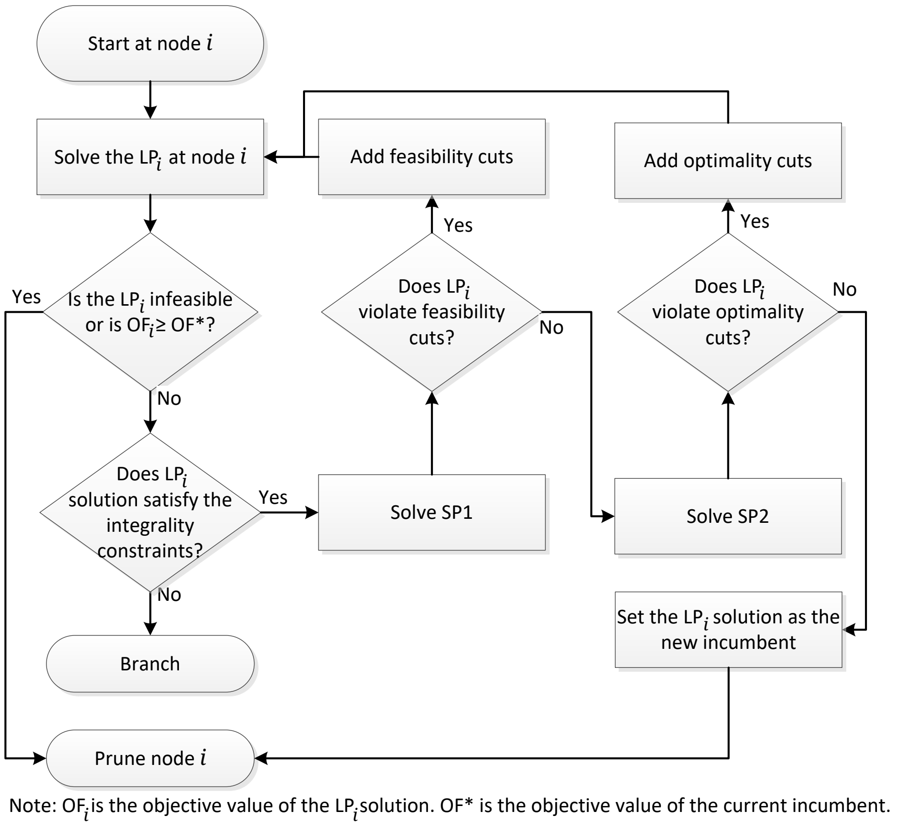


Figure 4.2: Example of the variation in arc costs in Algorithm 2.

4.2.5 Branch-and-Benders-cut

In the classical Benders decomposition, the MP and the subproblems are solved iteratively in an alternating sequence. At each iteration, the MP is solved to optimality by an MIP solver, and a considerable time may be spent revisiting candidate solutions that have been eliminated in previous iterations (Rahmaniani et al., 2017). On the other hand, in the BBC algorithm, a single search tree is built instead, and the cuts are generated inside the tree using separation routines that seek violated feasibility or optimality cuts (Errico et al., 2017).

Figure 4.3 shows a flowchart of the BBC method focusing on how the separation routines are used at each node of the branch-and-bound tree. At each node i, we solve the linear relaxation of the current MP, denoted by LP_i. If the LP_i is infeasible or the objective value of the LP_i solution (OF_i) is higher than or equal to the objective value of the current incumbent solution, then node i is pruned. Otherwise, integrality constraints are checked, and if the LP_i solution is not integer feasible, then branching is performed. Every time the LP_i solution is integer feasible, we call the separation routines of the subproblem. First, we solve SP1 and, if SP1 is infeasible, add new feasibility cuts to the MP. If no feasibility cut is obtained, then we solve SP2 to obtain an optimality cut for the MP. If no feasibility or optimality cuts are obtained, then the LP_i solution is feasible for the original problem (4.1)-(4.18) and is set as the new incumbent solution. Otherwise, the MP has been modified, LP_i must be resolved, and the described steps are applied again. It is worth mentioning that automatized cuts (for example, Gomory's cuts) and/or heuristics (for example, the relaxation induced neighborhood search (RINS) heuristic) available in commercial solvers can be used at each node of the branch-and-bound tree as well, although they are not included in Figure 4.3.



Start at node i
Solve the LP$_i$ at node i
Add feasibility cuts
Add optimality cuts
Yes
Is the LP$_i$ infeasible or is OF$_i$ ≥ OF*?
Does LP$_i$ violate feasibility cuts?
No
Yes
Does LP$_i$ violate optimality cuts?
No
No
Does LP$_i$ solution satisfy the integrality constraints?
Yes
Solve SP1
Solve SP2
No
Set the LP$_i$ solution as the new incumbent
Branch
Prune node i
Note: OF$_i$ is the objective value of the LP$_i$ solution. OF* is the objective value of the current incumbent.

Figure 4.3: Flowchart illustrating how the separation routines are used in a given node i of the BBC method.

4.2.6 Valid inequalities

The proposed BBC method also relies on valid inequalities, which are added to the MP and help improve the lower bounds provided by its linear relaxation. To define the valid inequalities, we first determine the shortest path between the depot and each demand node $i \in \mathcal{V}^d$. For each demand node i, we identify the damaged nodes that are used in a shortest path from the depot to this node. Then, we forbid the use of such damaged nodes in the paths and look for a new shortest path from the depot to the same node i again. If a damaged node j is forbidden and a path with distance less than l_i cannot be found for node i, then we have identified that the damaged node j is necessary to connect the demand node i with the depot. In such case, the accessibility time of node i depends on the repair time of node j and hence we obtain the following valid inequality:

$$\theta_i \geq R'_j, \forall\ i \in \mathcal{V}^d,\ j \in \mathcal{Q}_i, \tag{4.39}$$

in which $\mathcal{Q}_i$ is the set of damaged nodes that must be used to access the demand node i with a distance less than l_i. R'_j is defined as a decision variable of the MP and denotes a lower bound for the exact time at which the node j is repaired in the schedule defined by the variables X_{ij}.

Hence, we replace constraints (4.21) with:

$$R'_j \geq R'_i + t_{ij} + \delta_j - M \cdot (1 - X_{ij}), \forall\, i \in \mathcal{V}^r \cup \{0\},\ j \in \mathcal{V}^r, \tag{4.40}$$

where t_{ij} is the minimum time to travel from node i to node j when no nodes are damaged, which is easily computed using Dijkstra's algorithm.

Let $\mathcal{P} \subset \mathcal{V}^r$ be the subset of damaged nodes that cannot be repaired directly from the depot because they are not accessible without the restoration of other nodes. Using this set, we can define the following valid inequality:

$$\sum_{j \in \mathcal{P}} X_{0j} \leq 0. \tag{4.41}$$

It is also possible to identify the demand nodes that need the repair of at least one damaged node to become accessible. For each demand node, the lower bound for the time instant that it becomes accessible is the travel time plus the repair time of the first node repaired by the crew from the depot. Then, the valid inequality is given by:

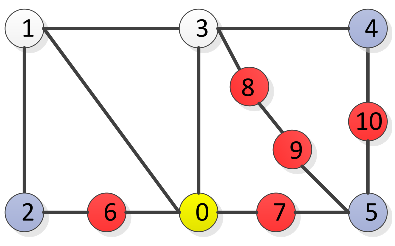
$$\theta_i \geq \sum_{j \in \mathcal{V}^r} (t_{0j} + \delta_j) \cdot X_{0j}, \forall\, i \in \mathcal{S}, \tag{4.42}$$

where $\mathcal{S}$ is the subset of demand nodes that require the restoration of at least one damaged node to guarantee they become accessible.

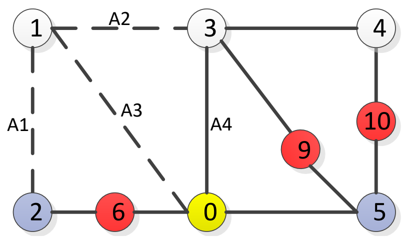
We also propose valid inequalities based on the reduction of the original damaged network of the problem. Let $L \subseteq \mathcal{V}^r$ be a subset of the damaged nodes and $F \subseteq \mathcal{V}^d$ be a subset of the demand nodes in the original graph G. We define G^{LF} as the subgraph obtained from G by deleting all the damaged nodes that are not in L and transforming all the demand nodes that do not belong to F into transshipment nodes . For instance, consider the graph G represented in Figure 4.4(a) with $\mathcal{V}^r = \{6, 7, 8, 9, 10\}$ and $\mathcal{V}^d = \{2, 4, 5\}$. For $L = \{6, 9, 10\}$ and $F = \{2, 5\}$, the graph G^{LF} is represented in Figure 4.4(b). To obtain G^{LF}, we removed all the damaged nodes in $\mathcal{V}^r \setminus L$ from G and transformed the demand nodes in $\mathcal{V}^d \setminus F$ into transshipment nodes.

We can further reduce the number of nodes in G^{LF} by removing transshipment nodes that are not directly connected to damaged nodes. For each node i removed from G^{LF}, we delete the arcs adjacent to this node and create new arcs connecting each pair of nodes j and k that were neighbors of i in G^{LF}, such that $j \neq k$. The cost c_{jk} of the new arc $j - k$ is set as $c_{jk} = c_{ji} + c_{ik}$. The resulting graph, denoted by $\bar{G}^{LF}$, is hereafter called as the LF-reduction of G. Figure 4.4(c) illustrates the graph $\bar{G}^{LF}$ obtained from the LF-reduction of graph G given in Figure 4.4(a). After obtaining the subgraph G^{LF} presented in Figure 4.4(b), we obtain $\bar{G}^{LF}$ by deleting node 1 from G^{LF}, as it was not directly connected to any damaged node. Then, we deleted arcs A1, A2 and A3, as they were adjacent to node 1, and created arcs A5, A6, and A7. Notice that either arc A4 or A7 is redundant and hence we can delete the one with the largest cost.

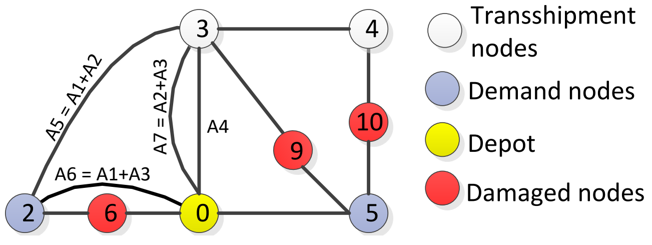
From a feasible solution of the SCSRP defined using $\bar{G}^{LF}$, we can derive valid inequalities



(a) Damaged network.



(b) Graph G^{LF}.



(c) Reduced graph $\bar{G}^{LF}$.

Figure 4.4: Example of a reduction of a damaged network.

for the original problem, as pointed out in Proposition 4.

Proposition 4. *Given $L \subseteq \mathcal{V}^r$ and $F \subseteq \mathcal{V}^d$, let $K^{\bar{G}^{LF}}$ be an optimal solution of the SCSRP defined using the LF-reduction $\bar{G}^{LF}$ of the original graph G. Let $\widehat{\Theta}^{\bar{G}^{LF}}$ be the optimal value and $\widehat{\theta}_i^{\bar{G}^{LF}}$ be the value of the variable Z_i^d in the optimal solution $K^{\bar{G}^{LF}}$, for all $i \in F$. Then, the following inequalities are valid for the MP of the original SCSRP defined using the graph G:*

$$\sum_{i \in F:\, d_i \cdot \widehat{\theta}_i^{\bar{G}^{LF}} > 0} d_i \cdot \theta_i \geq \widehat{\Theta}^{\bar{G}^{LF}}, \tag{4.43}$$

$$\Theta \geq \widehat{\Theta}^{\bar{G}^{LF}} + \sum_{i \in F:\, d_i \cdot \widehat{\theta}_i^{\bar{G}^{LF}} = 0} d_i \cdot \theta_i + \sum_{i \in \mathcal{V}^d \setminus F} d_i \cdot \theta_i. \tag{4.44}$$

Proof. Valid inequality (4.43) is proved by contradiction. Assume that there is a solution K^G of the original SCSRP (*i.e.*, using the original graph G) such that

$$\sum_{i \in F:\, d_i \cdot \widehat{\theta}_i^{\bar{G}^{LF}} > 0} d_i \cdot \widehat{\theta}_i^{G} < \widehat{\Theta}^{\bar{G}^{LF}},$$

where $\widehat{\theta}_i^G$ is the value of variable Z_i^d in the solution K^G. Since $L \subseteq \mathcal{V}^r$, graph $\bar{G}^{LF}$ have the same or less damaged nodes than graph G. Hence, a solution for the SCSRP defined using $\bar{G}^{LF}$ exists with $\widehat{\theta}_i^{\bar{G}^{LF}} \leq \widehat{\theta}_i^G$, $\forall\, i \in F$. Then,

$$\sum_{i \in F:\, d_i \cdot \widehat{\theta}_i^{\bar{G}^{LF}} > 0} d_i \cdot \widehat{\theta}_i^{\bar{G}^{LF}} \leq \sum_{i \in F:\, d_i \cdot \widehat{\theta}_i^{\bar{G}^{LF}} > 0} d_i \cdot \widehat{\theta}_i^{G} < \widehat{\Theta}^{\bar{G}^{LF}},$$

which is a contradiction because $\widehat{\Theta}^{\bar{G}^{LF}}$ is the value of an optimal solution of the SCSRP defined using $\bar{G}^{LF}$. Notice that valid inequality (4.43) remains valid if $\widehat{\Theta}^{\bar{G}^{LF}}$ is a lower bound for the value of the optimal solution related to graph $\bar{G}^{LF}$.

For inequality (4.44), notice that from constraint (4.22) of the MP, we have

$$\Theta \geq \sum_{i \in \mathcal{V}^d} d_i \cdot \theta_i = \sum_{i \in F} d_i \cdot \theta_i + \sum_{i \in \mathcal{V}^d \setminus F} d_i \cdot \theta_i = \sum_{i \in F:\, d_i \cdot \widehat{\theta}_i^{G^{LF}} > 0} d_i \cdot \theta_i + \sum_{i \in F:\, d_i \cdot \widehat{\theta}_i^{G^{LF}} = 0} d_i \cdot \theta_i + \sum_{i \in \mathcal{V}^d \setminus F} d_i \cdot \theta_i.$$

Then, using valid inequality (4.43), we obtain

$$\Theta \geq \sum_{i \in \mathcal{V}^d} d_i \cdot \theta_i \geq \widehat{\Theta}^{G^{LF}} + \sum_{i \in F:\, d_i \cdot \widehat{\theta}_i^{G^{LF}} = 0} d_i \cdot \theta_i + \sum_{i \in \mathcal{V}^d \setminus F} d_i \cdot \theta_i,$$

which proves the valid inequality (4.44). □

Valid inequalities (4.43) and (4.44) can be useful when the value (or a lower bound for the value) of the optimal solution of the SCSRP defined using the LF-reduction $\bar{G}^{LF}$ is trivial or can be easily derived. The separation procedure of these valid inequalities is detailed in the following subsection.

4.2.7 Graph reduction (GR) strategy

Based on Proposition 4, we propose a Graph Reduction (GR) strategy to obtain the LF-reduction of a graph G, as outlined in Algorithm 3. Basically, we create subgraphs $\bar{G}^{LF}$ using a feasible solution of the original SCSRP (*i.e.*, using graph G). This feasible solution can be quickly obtained using a heuristic, for example (see Section 4.2.8). Then, we determine the LF-reduction $\bar{G}^{LF}$, solve the SCSRP defined using this subgraph and check if there are violated valid inequalities of type (4.43) and (4.44) to be added to the MP of the original SCSRP. The idea is to generate sufficiently small subgraphs $\bar{G}^{LF}$, so that the corresponding (reduced) SCSRP can be quickly solved.

Let $L = (v_0, \ldots, v_i, \ldots, v_h)$ be a feasible partial sequence of damaged nodes repaired by the crew, where v_i is the ith damaged node repaired by the crew and v_h is the last damaged node repaired in order to make all the demand nodes in the set $\mathcal{V}^d$ accessible. For a given positive integer number $n_1 \leq h$, we partition the set of nodes in the partial schedule L into $P_1 = 1 + \left\lfloor \frac{h-1}{n_1} \right\rfloor$ sets. The sets are labeled from 0 to $P_1 - 1$, where $L_p = \{v_{(1+n_1 \cdot p)}, \ldots, v_{(n_1+n_1 \cdot p)}\}, \forall\, p = 0, \ldots, P_1 - 2$, and $L_{P_1-1} = \{v_{(1+n_1 \cdot (P_1-1))}, \ldots, v_h\}$. Similarly, let $F = (u_0, \ldots, u_i, \ldots, u_{|\mathcal{V}^d|})$ be the sequence of demand nodes connected to the depot when the damaged nodes are repaired according to sequence L, where u_i is the ith demand node that becomes accessible. Given a positive integer number $n_2 \leq |\mathcal{V}^d|$, we create a partition of the nodes in F given by $P_2 = 1 + \left\lfloor \frac{|\mathcal{V}^d|-1}{n_2} \right\rfloor$ sets. The sets are labeled from 0 to $P_2 - 1$, where $F_f = \{u_{(1+n_2 \cdot f)}, \ldots, u_{(n_2+n_2 \cdot f)}\}, \forall\, f = 0, \ldots, P_2 - 2$, and $F_{P_2-1} = \{v_{(1+n_2 \cdot (P_2-1))}, \ldots, v_{|\mathcal{V}^d|}\}$. Then, for each $p = 0, \ldots, P_1 - 1$ and $f = 0, \ldots, P_2 - 1$ we use sets L_p and L_f to obtain the LF-reduction $\bar{G}^{L_p F_f}$ and solve the corresponding (reduced) SCSRP to generate valid inequalities (4.43) and (4.44).

Notice that we are not using an arbitrary selection of damaged nodes to generate the sub-

Algorithm 3 Graph reduction strategy to derive valid inequalities (4.43) and (4.44).

```
Input:
Graph $G = (\mathcal{V}, \mathcal{E})$;
Sequences $L = (v_0, \dots, v_i, \dots, v_h)$ and $F = (u_0, \dots, u_i, \dots, u_{|\mathcal{V}^d|})$;
Positive integer numbers $n_1 \leq h$ and $n_2 \leq |\mathcal{V}^d|$;
Output:
Valid inequalities of type (4.43), (4.44);
1: for $p = 0$ to $P_1 - 1$ do
2:     for $f = 0$ to $P_2 - 1$ do
3:         Generate subgraph $\bar{G}^{L_p F_f}$;
4:         Solve the SCSRP defined using subgraph $\bar{G}^{L_p F_f}$;
5:         Derive the valid inequalities (4.43), (4.44) from the solution obtained using $\bar{G}^{L_p F_f}$;
6:     end for
7: end for
8: Add the valid inequalities (4.43),(4.44) to the MP.
```

graphs $\bar{G}^{L_p F_f}$, but a feasible sequence L. This sequence can be obtained, for example, by using a heuristic able to quickly define a good sequence of damaged nodes to be repaired. This way, we group in a same subgraph, the damaged nodes that are likely to be repaired sequentially in the solution of the original problem (associated with graph G). Also, notice that we do not consider all the damaged nodes but only those enough to make the demand nodes accessible. Similarly, we group in a same subgraph, the demand nodes that are likely to require the restoration of common damaged nodes to become connected to the depot.

4.2.8 Construction and local search heuristics

In this section, we use a construction heuristic and two local search heuristics with the aim of finding good feasible solutions of the SCSRP. The feasible solutions are used as initial incumbent solutions in the BBC algorithm.

4.2.8.1 Construction heuristic

The crew scheduling decision can be modeled as a traveling salesman problem (TSP) in which the cities to be visited are the damaged nodes. A simple construction heuristic for this problem is a greedy algorithm that makes a locally optimal choice at each iteration in an attempt to find a global optimum. The proposed method starts at the depot and, at each iteration, inserts at the end of the schedule a node that is not in the schedule yet and has the minimum travel time (when no nodes are damaged) to the last inserted node. A node insertion is feasible if this node can be visited without using a node that was not already repaired. Only feasible insertions can be selected at each iteration, and as a consequence, it always generates a feasible schedule. The construction heuristic can also generate feasible random solutions if we insert at the end of the schedule a randomly selected node and not the one with the minimum travel time to the last inserted node.

4.2.8.2 Local search heuristics

We propose two local search operators with the aim of improving a feasible schedule generated by the construction heuristic. The first local search operator (*swap*) exchanges the positions of two damaged nodes in the schedule. The second local search operator is a pairwise exchange (2-*opt*) that involves removing two edges and replacing them with two different edges that reconnect the fragments created. Let W_K^n be the set of all possible solutions (neighbors) obtained by applying the operator n in the schedule K, where $n \in \{swap, 2\text{-}opt\}$. Let Θ^K be the cost of the schedule K. Let $\widehat{K}_i$ be the ith element of set W_K^n. The local search heuristic based on the two operators is outlined in Algorithm 4. We have a feasible schedule as the input and a locally optimal solution provided by the heuristic as the output. We use subproblems SP1 and SP2 to evaluate the feasibility (line 10) and cost (line 12) of the schedule created when the operators are applied. When a solution better than the current one is found, the local search process is restarted. Furthermore, when a set W_K^n is fully explored, we restart the algorithm from a random solution (line 28). The algorithm terminates when no improvement is found for the last randomly generated solution.

4.3 Computational experiments

In this section, we evaluate the performance of the proposed solution approaches using instances from the literature. All the algorithms were implemented in C++ programming language. The BBC method was implemented on top of the IBM CPLEX Optimization Solver 12.7 using the Concert Technology library. We implemented the specialized algorithms to solve subproblems SP1 and SP2 and the heuristics according to their descriptions in Sections 4.2.4 and 4.2.8. All cuts and valid inequalities are added to problem using the Callback procedures available in the Concert Technology library. The experiments were run on a Linux PC with a CPU Intel Core i7 3.4 GHz and 16.0 GB of memory using a single thread. The stopping criteria was either the elapsed time exceeding the time limit of 3,600 seconds or the optimality gap being smaller than 10^{-4}. All the remaining parameters of CPLEX were kept at their default values.

4.3.1 Instance description

We carried out computational experiments using two types of theoretical instances: S1, which is composed of small instances, and S2, which is composed of medium and large instances, as presented by Maya-Duque et al. (2016). As described by the authors, they generated networks with different numbers of nodes and arcs based on the instance generator proposed by Klingman et al. (1974). Table 4.1 shows the characteristics of the set of instances. The type (S1 or S2), network name (class), number of demand nodes, and the total number of nodes and arcs in the original network can be seen in columns 1 to 5 of Table 4.1, respectively. For each original network, one class of instances was generated by varying two parameters, namely, α and β. Parameter α defines the percentage of damaged arcs in the network. Parameter β specifies the

Algorithm 4 Local search heuristic using the operator $n \in \{swap, 2\text{-}opt\}$.

```
Input:
Schedule $K = (v_0, v_1, ..., v_j, ..., v_{|\mathcal{V}^r|})$; Cost $\Theta^K$ of scheduling $K$;
Output:
Schedule $K^* = (v_0^*, v_1^*, ..., v_j^*, ..., v_{|\mathcal{V}^r|}^*)$;
1:  $\Theta^{K^*} := \Theta^{\widehat{K}_i}$; $K^* := K$;
2:  Determine set $W_K^n$;
3:  improvement_global := 1;
4:  while improvement_global = 1 do
5:      improvement_global := 0; improvement_local := 1;
6:      while improvement_local = 1 do
7:          $i := 1$;
8:          improvement_local := 0;
9:          while $i \leq |W_K^n|$ do
10:             Evaluate feasibility of schedule $\widehat{K}_i$ by solving subproblem SP1;
11:             if schedule $\widehat{K}_i$ is feasible then
12:                 Calculate cost $\Theta^{\widehat{K}_i}$ of schedule $\widehat{K}_i$ by solving subproblem SP2;
13:                 if $\Theta^{\widehat{K}_i} < \Theta^K$ then
14:                     $\Theta^K := \Theta^{\widehat{K}_i}$; $K := \widehat{K}_i$;
15:                     $i := |W_K^n| + 1$;
16:                     Determine new set $W_K^n$;
17:                     improvement_local := 1;
18:                     if $\Theta^{\widehat{K}} < \Theta^{K^*}$ then
19:                         $\Theta^{K^*} := \Theta^{\widehat{K}}$; $K^* := K$;
20:                         improvement_global := 1;
21:                     end if
22:                 end if
23:             end if
24:             $i := i + 1$;
25:         end while
26:     end while
27:     if improvement_global = 1 then
28:         Find a new random solution $K$ with the construction heuristic;
29:         Determine the new set $W_K^n$;
30:     end if
31: end while
```

maximum tolerable percentage by which the paths connecting demand nodes to the depot can increase in relation to the shortest paths in the network when no damaged node exists. For example, $\alpha = 50\%$ indicates that half of the arcs of the original network were damaged, and $\beta = 50\%$ indicates that the maximum distance l_i for the paths between the depot and the demand node i is 1.5 times the length of the shortest path between the depot and the node i when no damaged node exists. Columns 6 and 7 of Table 4.1 show the values of α and β, respectively.

For each damaged arc in the original network, one or more damaged nodes are added in the middle of the arc. Therefore, the total numbers of nodes and arcs in the instance depend on the parameter α. In the table, original network 1 with 25 nodes and 40 arcs is transformed into a damaged network with 27 nodes and 42 arcs when $\alpha = 5\%$ (the 2 damaged arcs are converted into 2 damaged nodes) and into a damaged network with 45 nodes and 60 arcs when $\alpha = 50\%$ (20 new damaged nodes). Thus, damaged networks generated from original network

1 have 27 to 45 nodes and 40 to 60 arcs. Columns 8 and 9 show the total number of nodes and arcs in the damaged networks. By combining the values of α and β for original network 1, for example, 20 instances were generated. For original network 16, the values of $\alpha = 5\%, 25\%, 50\%$ were combined with $\beta = 5\%, 10\%$ to form 6 instances, while the values of $\alpha = 10\%, 30\%$ were combined with $\beta = 25\%, 50\%$ to form 4 instances. For networks 1-15, the number of instances generated is 20. For networks 16-39, the number of instances generated is 10.

It is worth mentioning that some of the large instances in group S2 are actually much larger than the practical instances we typically find in real-world situations. Feng and Wang (2003), for example, considered a real network with 10 damaged points and less than 100 total nodes. Yan and Shih (2007) and Yan and Shih (2009) also considered real networks with less than 100 nodes but with 24 damaged points. Similarly, Xu and Song (2015) considered a real case with 36 damaged nodes and not more than 100 total nodes. Pramudita and Taniguchi (2014) considered a larger real damaged network with 98 damaged points (blocked arcs) and 198 total nodes. Finally, Akbari and Salman (2017a) considered one of the largest practical cases in the literature, involving networks with 240 damaged points, 349 nodes and 689 arcs. Note that we are considering instances with up to 312 damaged nodes, 712 total nodes and 937 total arcs.

Table 4.1: Set of instances.

	Network	Demand	Original network				Damaged network		Total
Type	(class)	nodes	nodes	arcs	Values for α (%)	Values for β (%)	nodes	arcs	instances
S1	1	19	25	40	5, 10, 25, 30, 50	5, 10, 25, 50	27 to 45	42 to 60	20
S1	2	19	25	37	5, 10, 25, 30, 50	5, 10, 25, 50	26 to 43	38 to 55	20
S1	3	19	25	39	5, 10, 25, 30, 50	5, 10, 25, 50	26 to 44	40 to 58	20
S1	4	24	30	83	5, 10, 25, 30, 50	5, 10, 25, 50	34 to 71	87 to 124	20
S1	5	24	30	89	5, 10, 25, 30, 50	5, 10, 25, 50	34 to 74	93 to 133	20
S1	6	24	30	84	5, 10, 25, 30, 50	5, 10, 25, 50	34 to 72	88 to 126	20
S1	7	28	35	118	5, 10, 25, 30, 50	5, 10, 25, 50	40 to 94	123 to 177	20
S1	8	28	35	115	5, 10, 25, 30, 50	5, 10, 25, 50	40 to 92	120 to 172	20
S1	9	28	35	113	5, 10, 25, 30, 50	5, 10, 25, 50	40 to 91	118 to 169	20
S1	10	15	20	39	5, 10, 25, 30, 50	5, 10, 25, 50	21 to 39	40 to 58	20
S1	11	15	20	37	5, 10, 25, 30, 50	5, 10, 25, 50	21 to 38	38 to 55	20
S1	12	15	20	37	5, 10, 25, 30, 50	5, 10, 25, 50	21 to 38	38 to 55	20
S1	13	35	40	146	5, 10, 25, 30, 50	5, 10, 25, 50	47 to 113	153 to 219	20
S1	14	35	40	143	5, 10, 25, 30, 50	5, 10, 25, 50	47 to 111	150 to 214	20
S1	15	35	40	143	5, 10, 25, 30, 50	5, 10, 25, 50	47 to 111	150 to 214	20
S2	16	50	60	191	5, 25, 50 | 10, 30	05, 10 | 25, 50	69 to 155	200 to 286	6 | 4
S2	17	50	60	197	5, 25, 50 | 10, 30	25, 50 | 05, 10	69 to 158	206 to 295	6 | 4
S2	18	50	60	196	5, 25, 50 | 10, 30	05, 10 | 25, 50	69 to 158	205 to 294	6 | 4
S2	19	70	80	247	5, 25, 50 | 10, 30	25, 50 | 05, 10	92 to 203	259 to 370	6 | 4
S2	20	70	80	245	5, 25, 50 | 10, 30	05, 10 | 25, 50	92 to 202	257 to 367	6 | 4
S2	21	70	80	248	5, 25, 50 | 10, 30	25, 50 | 05, 10	92 to 204	260 to 372	6 | 4
S2	22	90	100	274	5, 25, 50 | 10, 30	05, 10 | 25, 50	113 to 237	287 to 411	6 | 4
S2	23	90	100	271	5, 25, 50 | 10, 30	25, 50 | 05, 10	113 to 235	284 to 406	6 | 4
S2	24	90	100	273	5, 25, 50 | 10, 30	05, 10 | 25, 50	113 to 236	286 to 409	6 | 4
S2	25	125	140	324	5, 25, 50 | 10, 30	25, 50 | 05, 10	156 to 302	340 to 486	6 | 4
S2	26	125	140	323	5, 25, 50 | 10, 30	05, 10 | 25, 50	156 to 301	339 to 484	6 | 4
S2	27	125	140	322	5, 25, 50 | 10, 30	25, 50 | 05, 10	156 to 301	338 to 483	6 | 4
S2	28	140	170	398	5, 25, 50 | 10, 30	05, 10 | 25, 50	189 to 369	417 to 597	6 | 4
S2	29	140	170	399	5, 25, 50 | 10, 30	25, 50 | 05, 10	189 to 369	418 to 598	6 | 4
S2	30	140	170	396	5, 25, 50 | 10, 30	05, 10 | 25, 50	189 to 368	415 to 594	6 | 4
S2	31	200	200	447	5, 25, 50 | 10, 30	25, 50 | 05, 10	222 to 423	469 to 670	6 | 4
S2	32	200	200	449	5, 25, 50 | 10, 30	05, 10 | 25, 50	222 to 424	471 to 673	6 | 4
S2	33	200	200	449	5, 25, 50 | 10, 30	25, 50 | 05, 10	222 to 424	471 to 673	6 | 4
S2	34	300	300	524	5, 25, 50 | 10, 30	05, 10 | 25, 50	326 to 562	550 to 786	6 | 4
S2	35	300	300	525	5, 25, 50 | 10, 30	25, 50 | 05, 10	326 to 562	551 to 787	6 | 4
S2	36	300	300	525	5, 25, 50 | 10, 30	05, 10 | 25, 50	326 to 562	551 to 787	6 | 4
S2	37	400	400	625	5, 25, 50 | 10, 30	25, 50 | 05, 10	431 to 712	656 to 937	6 | 4
S2	38	400	400	625	5, 25, 50 | 10, 30	05, 10 | 25, 50	431 to 712	656 to 937	6 | 4
S2	39	400	400	625	5, 25, 50 | 10, 30	25, 50 | 05, 10	431 to 712	656 to 937	6 | 4
Total									540

4.3.2 Description of experiments

In this section, we present a description of the computational experiments. Table 4.2 presents the combination and stopping criteria of eleven proposed solution strategies, in which ET refers to elapsed time and TL to the total time limit. For instance, H3 is the heuristic strategy that uses first the construction heuristic, then the local search $2opt$, and finally the local search $swap$, either with stopping criteria given by time limit or a locally optimal solution found. The first four solution strategies (H1-H4) use only the construction and local search heuristics presented in Section 4.2.8. The following five strategies (BBC1-BBC5) are variants of the Branch-and-Benders-Cut (BBC) algorithm that use different combinations of the cuts presented in Section 4.2.3 and some of the valid inequalities presented in Section 4.2.6. The same separation algorithms are used in all the BBC strategies to identify the feasibility and cost of a scheduling solution of the MP. Then, we enumerate and add inequalities to the MP according to the type of cuts used in each BBC strategy. The algorithm BBC6 relies on the best heuristic method to provide a good feasible initial solution to the best BBC method. The best solution found with the heuristic is set as the incumbent solution of the MP. In the GR-BBC6 method, the algorithm BBC6 is combined with the Graph Reduction (GR) strategy presented in Section 4.2.7 to derive valid inequalities (4.39)-(4.44). BBC6 is used to solve the reduced SCSRP defined using the subgraphs $\bar{G}^{L_pF_f}$, which are generated using solutions provided by the heuristic H3, considering sets of $n_1 = 20$ damaged nodes and $n_2 = |\mathcal{V}^d|$ demand nodes (see Algorithm 4). The time limit to solve the reduced SCSRP is 60 seconds. If the reduced SCSRP for a given subgraph $\bar{G}^{L_pF_f}$ is not solved to optimality within this time limit, we reduce this subgraph by removing only the first five nodes of sets L_p and F_f and then we solve the corresponding reduced SCSRP again. Finally, the MIP model presented in Section 4.2.1 is used to solve the problem.

Table 4.2: Characteristic of the solution methods.

Solution strategy	Combination (stopping criteria)[1]
H1	Construction heuristic + $2opt$ ($ET > TL$ or locally optimal).
H2	Construction heuristic + $swap$ ($ET > TL$ or locally optimal).
H3	Construction heuristic + $2opt$ ($ET > \frac{1}{2}TL$ or locally optimal) + $swap$ ($ET > TL$ or locally optimal).
H4	Construction heuristic + $swap$ ($ET > \frac{1}{2}TL$ or locally optimal) + $2opt$ ($ET > TL$ or locally optimal).
BBC1	BBC algorithm with valid inequalities (4.39) - (4.42), feasibility cut (4.33) and optimality multi-cuts (4.35) and (4.36) ($ET > TL$ or gap = 0).
BBC2	BBC algorithm with valid inequalities (4.39) - (4.42), feasibility cut (4.32) and optimality cut (4.34) ($ET > TL$ or gap = 0).
BBC3	BBC algorithm with valid inequalities (4.39) - (4.42), feasibility cut (4.32) and optimality multi-cuts (4.36) ($ET > TL$ or gap = 0).
BBC4	BBC algorithm with valid inequalities (4.39) - (4.42), feasibility cut (4.32) and optimality multi-cuts (4.35) and (4.36) ($ET > TL$ or gap = 0).
BBC5	BBC algorithm <u>without</u> valid inequalities (4.39) - (4.42), feasibility cut (4.32) and optimality multi-cuts (4.35) and (4.36) ($ET > TL$ or gap = 0).
BBC6	H3 ($ET > \frac{1}{6}TL$ or locally optimal) + BBC4 ($ET > TL$ or gap = 0).
GR-BBC6	H3 ($ET > \frac{1}{6}TL$ or locally optimal) + GR ($ET > \frac{1}{60}TL$ each subproblem or optimality of the subproblems) + BBC4 with additional valid inequalities (4.43) - (4.44) ($ET > TL$ or gap = 0).
MIP model	Model (4.1)-(4.18) ($ET > TL$ or gap = 0)

[1] Let TL be the total time limit and ET be the elapsed time.

The solution methods were evaluated using performance profiles as proposed by Dolan and Moré (2002). Given a set $\mathcal{P}$ of instances and a set $\mathcal{F}$ of solution methods, performance profiles are based on the cumulative distribution function $P(f, q)$, which indicates the probability of a strategy f with a $\log_2$ performance ratio being within a factor $q \in R$ of the best possible ratio. The function $P(f, q)$ is defined as:

$$P(f,q) = \frac{|\{p \in \mathcal{P} : \log_2(v(p,f)) \leq q\}|}{|\mathcal{P}|} \;,\; q \geq 0, \tag{4.45}$$

$$\text{with } v(p,f) = \frac{TC_{pf}}{min\{TC_{pf} : f \in F\}} \,, \tag{4.46}$$

where $|\mathcal{P}|$ is the total number of instances and TC_{pf} is the performance measure (objective function cost, gap or elapsed time) of problem p when solved by method f. Values of $P(f, q)$ when $q = 0$ indicate the fraction of instances for which the strategy reached the best solution. For $q > 0$, $P(f, q)$ is the fraction of instances for which strategy f obtained solutions with a quality within a factor of 2^q of the best solutions. Values of q when $P(f, q) = 1$ indicate that quality of the solutions obtained by strategy f for all instances are within a factor of 2^q of the best solutions.

4.3.3 Computational performance of the proposed approaches

To evaluate the performance of the heuristic approaches, we use the objective value of the solutions found within a time limit of 3,600 seconds. The heuristic algorithms do not provide a lower bound for the objective value, so we cannot calculate the optimality gap. On the other hand, the BBC approaches provide upper- and lower-bound values, so we use the optimality gap provided by the algorithms within a time limit of 3,600 seconds as well to compare the BBC approaches. The optimality gap is computed as:

$$gap = \frac{Z^U - Z^L}{Z^U}, \tag{4.47}$$

in which Z^U is the upper bound or best integer solution and Z^L is the lower bound. The optimality gap is a good indicator of the quality of the methods because it considers simultaneously the upper and lower bound of the solutions. However, we also compared the upper bounds of the BBC strategies, and the overall results were similar to those obtained with the optimality gaps.

Figure 4.5 shows the performance profiles for the heuristic strategies (H1-H4) using the objective value. The results indicate that the two strategies that combine the local search heuristics *swap* and 2*opt*, H3 and H4, yield a more stable performance than the others. Strategy H3 (H4) found the smallest objective function cost for 80.18% (74.44%) of the instances, and in the remaining instances, H3 (H4) provides a solution with cost within a factor of $2^{0.61} \approx 1.53$ ($2^{0.78} \approx 1.72$) of the lowest cost found. Due to this behavior, H3 was selected as the best heuristic strategy.

Figure 4.6 presents the performance profiles for the BBC algorithms (BBC1-BBC5) based

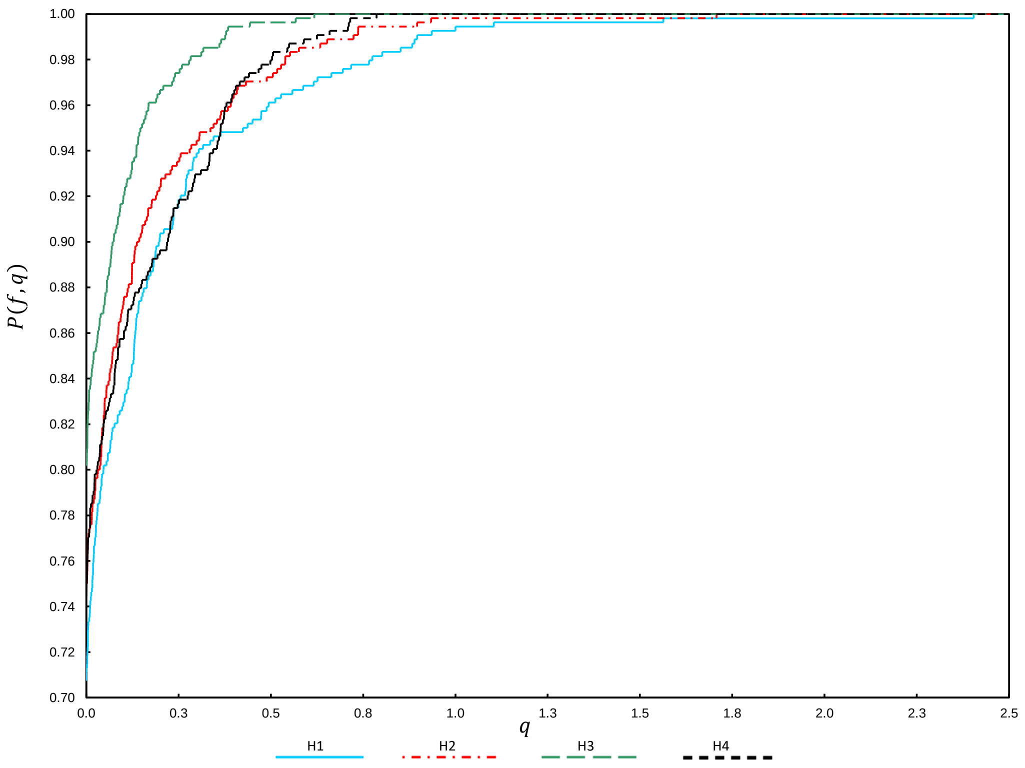


Figure 4.5: Performance profiles of the heuristic methods based on the objective value.

on the optimality gap. Table 4.3 shows the extreme values of the performance profiles for the BBC strategies. The performance profiles reveal that the majority of the strategies have similar results in 90% ($P(f,q) = 0.9$) of the instances. As expected, strategy BBC5, which does not use any valid inequalities, presents the worst performance. In most instances, the valid inequalities improved the lower bound, thus accelerating the convergence of the algorithms. In fact, while BBC5 found the best gap only for 35.5% of the instances, BBC4 found the best gap for 64.8% of the instances, which represents a substantial improvement of 82%. Furthermore, for instances in which the best gap is not achieved, BBC4 provides solutions with a gap within a factor of $2^{3.45} \approx 11$ of the best gap, while for the BBC5 algorithm, the factor is $2^{7.66} \approx 202$.

Table 4.3: Extreme values of the performance profiles for the BBC strategies.

BBC strategy	$P(f,q)$[1]	q[2]
BBC1	0.5842	7.6582
BBC2	0.6237	5.7664
BBC3	0.6411	4.0752
BBC4	0.6474	3.4558
BBC5	0.3553	7.6582

[1] Values of $P(f,q)$ when $q = 0$.
[2] Values of q when $P(f,q) = 1$.

By comparing the performance profiles of algorithms BBC1 and BBC4, it is possible to see that using feasibility cut (4.32) is better than using feasibility cut (4.33). This result was also expected because equation (4.32) cuts off a larger number of infeasible solutions when it is used. Using multiple lower bound functions as optimality cuts appears to be more efficient than

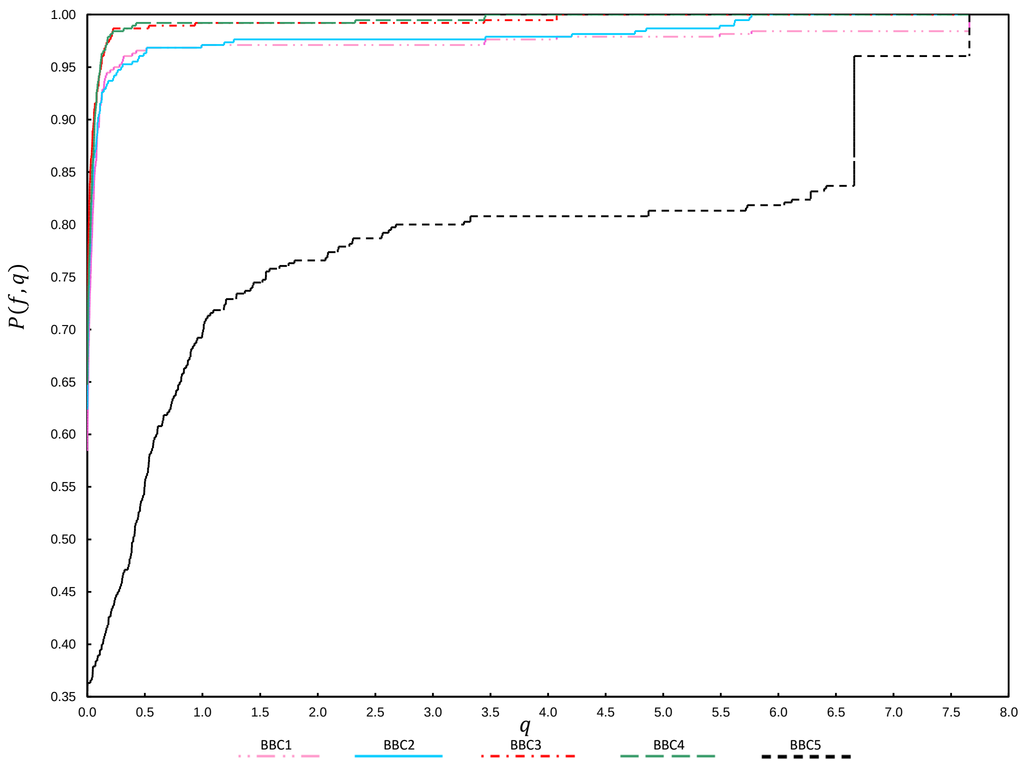


Figure 4.6: Performance profiles of the BBC algorithms based on the optimality gap.

using single cuts, which can be deduced from the comparison among algorithms BBC2, BBC3 and BBC4. Notice that the optimality multi-cut approaches (BBC3 and BBC4) are faster and more stable than the optimality single-cut approach (BBC2). Finally, from the comparison of algorithms BBC3 and BBC4, we can conclude that the use of cut (4.35) improves the convergence of the method. Optimality multi-cut (4.35) helps set a lower bound for a greater number of solutions than multi-cut (4.36) individually. Therefore, the algorithm BBC4 is selected as the best strategy, which provides the smallest gap for 64.73% of the instances and, in the remaining instances, provides solutions with a gap within a factor of $2^{3.45} \approx 11$ of the best gap obtained.

Approach BBC6 combines the best heuristic and BBC strategies, H3 and BBC4, respectively. GR-BBC6 combines BBC6 with the GR strategy. We build performance profiles based on the gap provided by the algorithms within the time limit of 3,600 seconds to compare BBC4, BBC6, GR-BBC6, and the MIP model. As we can see in Figure 4.7, not surprisingly, the BBC algorithms outperform the MIP model. In fact, the mathematical model found feasible solutions for only 45.8% of the instances. BBC6 shows a more stable performance than the BBC4 algorithm. Thus, starting the BBC with an initial solution provided by heuristic H3 improves the performance of the BBC algorithm. By comparing GR-BBC6 and BBC6, we can infer that the valid inequalities (4.43)−(4.44) derived by the GR strategy are effective to improve the convergence of the method. GR-BBC6 (BBC6) achieved the best gap in 96.1% (50.78%) of the instances and, for the instances it was not achieved, the solution gap was within a factor of $2^{4.68} \approx 25.63(2^{6.12} \approx 69.55)$ of the best gap obtained.

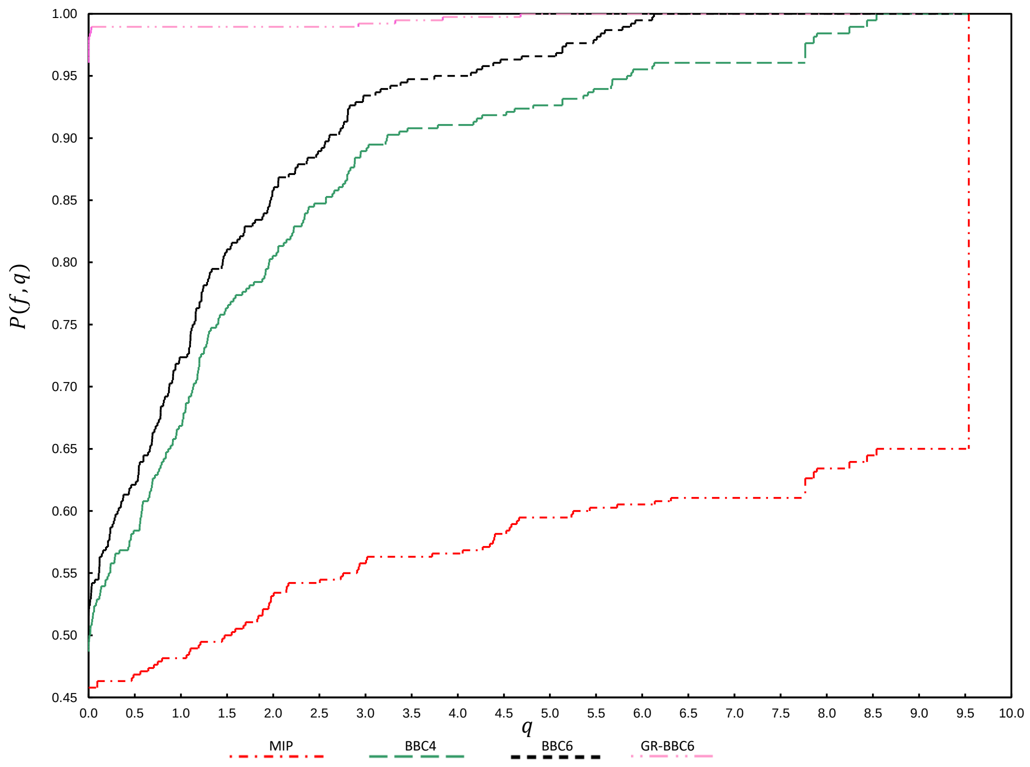


Figure 4.7: Performance profiles of the best BBC strategies and the MIP model based on the optimality gap.

4.3.4 Performance of the best strategy

Table 4.4 shows the average upper bound, gap and elapsed time of the best strategy proposed in this chapter for solving the instances of group S1 and S2. For all instances, the GR-BBC6 method provided feasible solutions within 3,600 seconds. The average total elapsed time was 2,361 seconds, and the average time that GR-BBC6 spent to find the best upper bound was 797.3 seconds, 66.24% smaller than the elapsed time. Thus, GR-BBC6 finds feasible solutions relatively quickly, and most of the elapsed time is consumed to improve the lower-bound values. The average gap considering all instances was 38.82%. For instances S1, the average gap was 9.32%, while for instances S2, the average gap was 57.26%. As expected, worse gaps are obtained for instances with a large number of nodes and arcs. If we consider only practical size instances (according to most of the real-world cases presented in the literature) we could limit our experiments to the first 24 class of instances, obtaining an average gap of 18.17%. Therefore, the GR-BBC6 algorithm is effective to solve practical size instances with good quality solutions.

We also performed additional experiments increasing the time limit of the GR-BBC6 algorithm to 24 hours. We considered the classes of instances 12 and 38, which present the smallest and the largest optimality gaps according to Table 4.4. The results reveal that the gap is reduced from 1.76% to 1.21% (31.25%) in class 12, and from 91.87% to 86.89% (5.42%) in class 38. Thus, our approach can be more effective for longer computational times, although the gap improvement can be rather negligible from the practical point of view.

Table 4.5 shows the average gap of the GR-BBC6 method according to different values of

Table 4.4: Average results of the GR-BBC6 strategy.

Type	Network (Class)	Avg. upper bound	Avg. gap (%)	Avg. elapsed time (sec.)	Avg. best[1] time (sec.)
S1	1	9,744.98	2.36	720.02	0.06
S1	2	34,088.95	5.04	1,039.29	0.05
S1	3	49,862.45	2.47	844.10	427.62
S1	4	18,037.43	8.24	1,440.08	2.51
S1	5	18,484.94	8.76	1,440.36	518.32
S1	6	20,917.01	12.51	1,618.59	260.78
S1	7	36,511.03	6.71	2,177.40	14.25
S1	8	26,048.79	13.94	1,956.36	66.38
S1	9	33,953.25	21.84	1,580.71	21.32
S1	10	48,459.97	2.42	725.84	487.55
S1	11	38,538.07	3.61	798.72	0.09
S1	12	28,036.81	1.76	723.84	6.21
S1	13	23,566.08	9.64	2,139.00	19.63
S1	14	81,031.99	20.28	2,119.10	351.12
S1	15	52,200.77	20.21	2,138.85	8.85
S2	16	38,737.05	18.09	2,074.49	628.29
S2	17	30,448.01	16.56	2,259.61	391.54
S2	18	97,476.41	21.39	2,092.61	569.09
S2	19	65,092.84	33.02	2,160.52	1,186.22
S2	20	71,172.82	38.95	2,550.34	697.30
S2	21	75,602.30	40.20	2,520.80	1,190.12
S2	22	211,486.51	41.05	2,883.30	1,522.30
S2	23	98,827.21	41.49	2,880.07	1,129.43
S2	24	209,434.98	45.56	2,880.88	1,975.86
S2	25	155,240.79	49.76	2,880.24	1,127.12
S2	26	274,847.22	62.94	2,961.89	1,756.10
S2	27	163,429.50	53.77	2,880.10	1,267.61
S2	28	435,199.83	58.23	2,881.21	1,495.73
S2	29	247,954.95	65.94	2,880.62	1,311.75
S2	30	468,494.13	57.76	3,252.19	1,643.45
S2	31	314,909.80	70.55	2,880.80	1,232.92
S2	32	337,827.97	61.12	2,881.92	1,224.98
S2	33	275,998.35	65.94	3,240.33	912.47
S2	34	481,122.55	88.49	3,600.03	1,097.02
S2	35	472,126.95	84.17	3,600.05	1,704.58
S2	36	497,125.76	86.16	3,600.03	1,311.96
S2	37	532,181.82	91.50	3,600.16	988.37
S2	38	697,781.81	91.87	3,600.05	1,433.22
S2	39	553,372.93	89.66	3,600.05	1,112.55
Avg. All		187,830.13	38.82	2,361.66	797.30
Avg. S1		34,632.17	9.32	1,430.82	145.65
Avg. S2		283,578.85	57.26	2,943.43	1,204.58

[1] Time that GR-BBC6 spent to find the best upper bound.

α and β. For example, the value 16.65 in bold in the table indicates the average gap for all instances with $\alpha = 10\%$ and $\beta = 5\%$. Note that the instances become more challenging when the percentage of damage (α) increases, as expected. In fact, more damaged nodes lead to (possibly) more crew schedules to be evaluated in the MP, slowing down the convergence of the method. More nodes in the network also makes the resolution of the subproblems even harder. The GR-BBC6 strategy found solutions with an average gap of 7.76% for the instances with $\alpha = 5\%$, and an average gap of 55.55% for instances with $\alpha = 50\%$. Similarly, the difficulty of the instances decreases (on average) when the maximum tolerable percentage (β) increases. Higher values of β make it easier for subproblem SP2 to find a feasible path between the depot and the demand nodes. The average gap for instances with $\beta = 5\%$ is 31.92%, while the average gap for instances with $\beta = 50\%$ is 29.10%. It is worth mentioning that, in most of the practical situations, no more than $\alpha = 30\%$ of the roads are considered as damaged roads. For instance, Akbari and Salman (2017a), which addressed one of the largest practical cases in the literature, considered 33% of the roads as damaged.

Table 4.5: Average gap for each value of α and β.

		α (%)					
		5	10	25	30	50	Avg.
β (%)	5	8.51	**16.65**	38.10	39.72	56.64	31.92
	10	7.59	16.33	34.11	40.85	53.85	30.55
	25	7.58	15.63	35.92	39.44	55.84	30.88
	50	7.34	12.32	32.64	37.36	55.83	29.10
	Avg.	7.76	15.23	35.19	39.34	55.55	30.61

4.3.5 Comparison with other results from the literature

This section compares the results obtained by the GR-BBC6 strategy with the results of other approaches available in the literature, namely, the dynamic programming (DP) algorithm and the iterated greedy-randomized constructive procedure (IGRCP) metaheuristic, both of which were proposed by Maya-Duque et al. (2016). While the DP approach is also an exact method analogous to our GR-BBC6 method, the IGCRP is a metaheuristic and hence has no guarantee of optimality. This metaheuristic is based on the greedy randomized adaptive search procedure (GRASP) and consists of two phases: the construction of a feasible solution and an improvement in the constructed solution, including multiple runs of the construction phase after the improvement routine. Our BBC algorithm is the first exact method proposed in literature able to find a lower bound for all the considered instances. Thus, it is not possible to perform any comparison of lower bounds using other approaches from the literature. This way, we only compare the solution costs (upper bounds) provided by the BBC with the costs delivered by the other approaches. We emphasize that the purpose is not to compare the methods, but to verify the quality of the solutions provided by the GR-BBC6 method. We show the results only for instances in group S1 (small instances) because the DP strategy proposed in Maya-Duque et al. (2016) is not able to solve medium and large instances. The IGRCP metaheuristic, on the other hand, was used to solve S2 instances in Maya-Duque et al. (2016), but we did not have access to those solutions.

Table 4.6 shows the average upper bound and elapsed time of the three approaches for instances in group S1. The character "–" indicates that no solution was obtained for one or more instances of the class. The last column "ratio" shows the ratio of the upper bound of IGRCP in relation to the upper bound of GR-BBC6. Ratios smaller than 1 indicate that the GR-BBC6 strategy improves the upper bound found by the IGRCP metaheuristic. The columns "# optimal" show the number of optimal solutions found by each exact method. The DP algorithm solved all the instances to optimality for classes of instances corresponding to networks 1, 2 and 10. For the other classes, the DP algorithm did not solve some of the instances within a time limit of 24 hours, especially those with $\alpha = 50$. For instances of classes corresponding to networks 1, 2 and 10, the solutions of the GR-BBC6 method were equal to the solutions of the DP strategy, indicating that these solutions are optimal, although there is a nonzero gap related to the lower bound computed with the GR-BBC6 method.

The BBC is able to prove optimality in (181) 60.33% of the small instances, while the best exact approach available so far in the literature proved optimality for (160) 53.33% of the small instances. Furthermore, our BBC algorithm solves to optimality 18.33% of the medium and large instances, while medium and large instances were not solved with the DP. In terms of computational times, the DP strategy was slower than the GR-BBC6 strategy for instances in classes 1, 2 and 10.

Table 4.6: Average result of the solution methods for small instances.

	Avg. upper bound			# optimal		Avg. elapsed time (sec.)			Avg. best[4]	
Class	IGRCP[1]	DP[2]	GR-BBC6[3]	DP	GR-BBC6	IGRCP	DP	GR-BBC6	time (sec.)	Ratio
1	9,744.98	9,744.98	9,744.98	20	16	1.09	2,945.31	720.02	0.06	1.000
2	34,088.95	34,092.54	34,088.95	20	16	1.43	8,733.50	1,039.29	0.05	1.000
3	49,987.07	–	49,862.45	17	16	1.60	–	844.10	427.62	0.998
4	18,246.59	–	18,037.43	16	11	7.04	–	1,440.08	2.51	0.989
5	18,151.81	–	18,484.94	12	11	14.10	–	1,440.36	518.32	1.018
6	21,253.39	–	20,917.01	1	11	17.98	–	1,618.59	260.78	0.984
7	36,873.30	–	36,511.03	0	8	11.77	–	2,177.40	14.25	0.990
8	26,382.27	–	26,048.79	0	9	38.37	–	1,956.36	66.38	0.987
9	35,223.52	–	33,953.25	0	11	20.26	–	1,580.71	21.32	0.964
10	48,545.84	48,460.28	48,459.97	20	16	0.91	16,663.44	725.84	487.55	0.998
11	39,212.63	–	38,538.07	17	16	1.59	–	798.72	0.09	0.983
12	28,876.04	–	28,036.81	16	16	0.87	–	723.84	6.21	0.971
13	23,535.63	–	23,566.08	8	8	7.83	–	2,139.00	19.63	1.001
14	87,163.33	–	81,031.99	8	8	91.47	–	2,119.10	351.12	0.930
15	52,085.29	–	52,200.77	5	8	53.51	–	2,138.85	8.85	1.002
Average	35,291.38	–	34,632.17	10.67	12.07	17.99	–	1,430.82	145.65	0.988

[1] Metaheuristic based on GRASP proposed in Maya-Duque et al. (2016)
[2] Exact dynamic programming algorithm proposed in Maya-Duque et al. (2016)
[3] Best BBC strategy (1 hour time limit).
[4] Time that GR-BBC6 spent to find the best upper bound.

On average, the solutions provided by the BBC approach GR-BBC6 are better than the solutions provided by the IGRCP heuristic. Additionally, the BBC6 method provided a lower bound and an optimality gap for all the solutions within a time limit of 3,600 seconds. Thus, the BBC can obtain a valid lower bound for all the instances without deteriorate the cost (upper bound) of the solutions or even at improving the upper bound of the solutions. As expected, the IGRCP metaheuristic was the fastest method but gave no guarantee of optimality, as it corresponds to a heuristic method. Note that most of the time spent by the BBC strategy is to improve the lower bound. In fact, the average time spent by GGR-BBC6 to find the best upper bound is 145.65 seconds, 10 times smaller than the average elapsed time.

4.4 Final remarks of the chapter

This chapter explored branch-and-Benders-cut (BBC) approaches to solve the crew scheduling and routing problem (CSRP), in the context of road restoration. As a key contribution, it developed the first exact solution approach that is able to obtain feasible solutions and lower bounds for all instances from the literature, including very large-scale instances. The addressed problem is typically found in post-disaster situations where the damaged network must be repaired as quickly as possible to promote an effective response. The joint presence of scheduling and routing decisions explains the complexity of solving such problems, for which commercial solvers cannot be efficiently used. Thus, we have devised approaches based on the Benders decomposition, applied to an MIP formulation that determines a fair and efficient road restoration plan. We employed feasibility cuts, multiple optimality cuts, and specialized valid inequalities, which have enhanced the performance of the BBC approaches. The use of simple heuristics to provide initial incumbent solutions for the master problem was also an important strategy to accelerate the convergence of the methods. The proposed BBC strategies have improved the results of exact and heuristic methods proposed so far in the literature. In fact, our best approach has proven the optimality of 41.67% of the instances, and for 100% of the instances, it obtained valid lower bounds for the first time. It is worth noting that we have not found any other computational study that considers so many nodes and arcs for any variant of the SCSRP in road restoration. The major remaining obstacle, though, is to provide the optimality certificate for some large-scale instances. In this sense, in the next chapter, we investigate particular properties and characteristics of the problem to derive new valid inequalities and different ways for decomposing the MIP formulation. Additionally, we develop hybrid methods combining exact and metaheuristic strategies to obtain tighter bounds and improved solutions.

Chapter 5

Metaheuristics and hybrid methods for the SCSRP

This chapter presents: (i) a branch-and-Benders-cut (BBC) method that enhances the approach presented in Chapter 4 by using a different variable partitioning scheme and new valid inequalities that strengthen the linear relaxation of the master problem; (ii) genetic algorithm (GA) and simulated annealing (SA) metaheuristics; and (iii) an exact hybrid BBC (HBBC) method that effectively combines the first two approaches. This chapter is organized as follows. Section 5.1 motivates the development of the new decomposition-based algorithms and presents the related works. The new Benders reformulation of the SCSRP is described in Section 5.2. Section 5.3 describes the proposed SA and GA metaheuristics. Section 5.4 presents the HBBC algorithm that combines the metaheuristics with the BBC algorithm. Finally, the computational results and conclusions are presented in Sections 5.5 and 5.6, respectively.

*A paper based on this chapter is accepted in Computers & Operations Research (COR). *(Moreno et al., 2020) Moreno, Alfredo; Munari, Pedro; Alem, Douglas. Decomposition-based algorithms for the crew scheduling and routing problem in road restoration. Computers & Operations Research, 2020.* `https://doi.org/10.1016/j.cor.2020.104935`

5.1 Introduction

As described in the previous chapter, the BBC algorithm is a branch-and-cut algorithm in which the master problem obtained from a Benders decomposition is solved by a single search tree, and the cuts are generated within the tree using the subproblems as separation routines. This strategy has yielded satisfactory results in many applications, particularly when used together with other acceleration strategies to improve the performance of the method (Rahmaniani et al., 2017). For instance, Taşkin and Cevik (2013) and Gendron et al. (2014) proposed heuristics to find initial solutions to warm-start the BBC and valid inequalities to improve the lower bound of the master problem. Additionally, Taşkin and Cevik (2013) applied a local search algorithm to the master problem solutions during the execution of the BBC. Salazar-González and Santos-Hernández (2015) also proposed valid inequalities that are added in a BBC scheme together with Benders cuts and a heuristic approach to find good upper bounds. Furthermore, the authors transformed the subproblem into a classical max-flow problem instead of using a linear programming formulation in the separation procedure. Adulyasak et al. (2015) used lifting inequalities, multiple Pareto-optimal cuts generation, and an exponential-sized set of subtour elimination constraints (SECs) dynamically added together with the Benders cuts to accelerate the BBC. Arslan and Karaşan (2016) increased the efficiency of a BBC method by using multiple Pareto-optimal cuts generation and by using a construction heuristic to derive solutions for the subproblems. Gendron et al. (2016) devised a BBC method in which integer variables of the subproblem are relaxed and included in the master problem.

More recently, Shao et al. (2017) adopted a tabu list of solutions in the master problem to eliminate solutions that would invoke repetitive Benders subproblems. Fischetti et al. (2017) applied a stabilization procedure at the root node and heuristics along the nodes of the branch-and-cut tree. Errico et al. (2017) enhanced a BBC by adding SECs and other inequalities together with the Benders cuts. Additionally, the authors proposed heuristics to generate initial cuts and solutions. They also applied a local search operator during the BBC to improve the master problem solutions and to generate multiple cuts from the neighborhood of the current solution. Li et al. (2018) developed a BBC algorithm involving Pareto-optimal cuts and valid inequalities to restrict the feasible space of the master problem.

Most of these studies used valid inequalities to improve the lower bound of the master problem, heuristics to provide a solution to warm-start the algorithm, and strategies to efficiently solve the subproblems derived from the decomposition. Other effective strategies such as heuristics along the branch-and-cut tree to generate cuts in the neighborhood of the master problem solutions have been seldom considered in the corresponding literature. We call this strategy *hybrid branch-and-Benders-cut* (HBBC). Hybridizing Benders decomposition methods with heuristics or metaheuristics can simultaneously improve both the lower and upper bounds (Rei et al., 2009; Raidl, 2015). The upper bound can be enhanced when new best solutions are found during the evaluation of the neighborhood of the current master problem solution, while the lower bound can be boosted via the generation of feasibility or optimality cuts from the solutions found by the heuristics. Heuristics can also reduce the size of the branch-and-cut tree by providing good solutions in early stages of the algorithm (Errico et al., 2017) and can be used as simple stabilization tools to handle the instability of the classical Benders decomposition (Rahmaniani et al., 2017).

To the best of our knowledge, no HBBC approaches have been proposed for the SCSRP or any variant of the problem. The use of heuristics or metaheuristics in the nodes of the branch-and-cut tree and other acceleration strategies proposed in the literature can improve the performance of the BBC for the CSRP. Thus, in this chapter we speed up a BBC approach for the SCSRP by using metaheuristics and other advanced acceleration strategies. The main contributions can be summarized as follows: (1) We develop a BBC method that enhances the BBC method proposed in the previous chapter using a different variable partitioning scheme that leads to a stronger relaxed master problem. This scheme keeps in the master problem the original objective function and the variables related to relief paths and scheduling decisions. In the previous BBC method, only variables related to scheduling decisions were kept in the master problem, with no information on the original cost. With a stronger master problem, the enhanced BBC method requires fewer calls to the subproblem and hence generates fewer cuts; (ii) We propose new valid inequalities to improve the performance of the BBC algorithm. These valid inequalities explore the special structure of the master problem and take advantage of the variables related to relief paths to further strengthen its linear relaxation; (iii) We devise two new metaheuristics based on a genetic algorithm (GA) and simulated annealing (SA) to solve the SCSRP. Our metaheuristics have a random search component that explores the space of scheduling decisions and an optimization component to find the best crew route and relief paths given a schedule. These are the first GA and SA approaches proposed for the SCSRP. We show that both metaheuristics outperform the existing GRASP-based metaheuristic developed to solve the same problem; (iv) We hybridize our BBC approach by embedding metaheuristics in the branch-and-cut tree that solves the master problem. We use the metaheuristics not only to improve the master problem solutions but also to derive Benders cuts from the neighborhood of the solutions.

The efficiency of the developed solution methods are compared using benchmark instances. As a result of the development of these techniques, the methods managed to obtain feasible solutions for the 390 benchmark instances, proving optimality for the first time on 30 of them. Moreover, they significantly improve the best known lower and upper bounds, optimality gaps, and execution times by 15.21%, 8.15%, 26.17%, and 71.14%, respectively, on average.

5.2 A new Branch-and-Benders-Cut algorithm

The BBC method proposed in the previous chapter considered only the scheduling decisions (variables X_{ij}) as complicating variables. There were two subproblems, one to check the feasibility of the RMP solutions using only the variables related to crew routing (Z_i^r, P_{eij} and N_{kij}); and another to check the cost of the RMP solutions using only the variables related to relief paths (Z_i^d, Y_{ej} and V_{kj}). This decomposition leads to a weak RMP formulation since the objective function of the original problem, which minimizes the weighted sum of the variables Z_i^d, is considered in the second subproblem and no information of the original costs was kept in the RMP.

In the new BBC algorithm, we rely on a different variable partitioning scheme in which the variables defining the paths that connect the depot with the demand nodes (Z_i^d, Y_{ej} and V_{kj}) are kept in the RMP, in addition to the scheduling decision variables. As a consequence, the master problem becomes stronger, and the resulting BBC method is likely to require fewer calls to the subproblem, hence decreasing the number of generated cuts. In the proposed scheme, the RMP is defined as follows:

$$(RMP) \qquad \min \sum_{i \in \mathcal{V}^d} d_i \cdot Z_i^d, \tag{5.1}$$

$$\text{s.t.}(4.2), (4.3), (4.7) - (4.10), (4.14), (4.16), (4.18), \tag{5.2}$$

$$Z_i^d \geq \theta_j + (V_{ji} - 1) \cdot M, \forall\ i \in \mathcal{V}^d,\ j \in \mathcal{V}^r, \tag{5.3}$$

$$\theta_j \geq \theta_i + \delta_j - M \cdot (1 - X_{ij}), \forall\ i \in \mathcal{V}_0^r,\ j \in \mathcal{V}^r, \tag{5.4}$$

$$\theta_j\ \geq 0, \forall\ j \in \mathcal{V}^r. \tag{5.5}$$

Constraints (5.3) define the time at which each demand node i becomes accessible. Constraints (5.4) are introduced to set a lower bound for the time at which the damaged nodes are repaired. These constraints also act as subtour elimination constraints. Constraints (5.5) impose the domain of the decision variables θ_j. The value of variable θ_j is underestimated because the RMP does not consider the routing decisions of the crew. The full route of the crew consists of paths connecting the consecutive damaged nodes in the schedule defined by variables X_{ij}. Since damaged nodes can obstruct the access to other damaged nodes of the network, the paths available for the crew at a specific moment depend on which nodes are still damaged at that moment, which, in turn, depends on the scheduling decisions. Thus, the paths available for the crew change dynamically during the restoration according to the schedule. Without considering routing decisions in the RMP, the damaged points that are not accessible at a given moment might be selected first in the schedule, making the schedule infeasible in practice. Therefore, the RMP lacks feasibility cuts to avoid infeasible schedules and optimality cuts to set the real values of the variables θ_j.

To derive feasibility and optimality cuts for the RMP, we solve a subproblem that defines the route of the crew from a scheduling solution. The subproblem can be stated as follows:

$$(SP) \ \min \sum_{i \in V^r} Z_i^r, \tag{5.6}$$

$$\text{s.t.} \quad (4.6), (4.12), (4.15), (4.17), \tag{5.7}$$

$$\sum_{e \in \mathcal{E}_i} P_{eij} = \widehat{X}_{ij}, \forall\ i \in \mathcal{V}^r \cup \{0\},\ j \in \mathcal{V}^r, \tag{5.8}$$

$$\sum_{e\in\mathcal{E}_j} P_{eij} = \widehat{X}_{ij},\ \forall\ i \in \mathcal{V}^r \cup \{0\},\ \ j \in \mathcal{V}^r, \tag{5.9}$$

$$Z_j^r \geq Z_i^r + \sum_{e\in\mathcal{E}} P_{eij} \cdot \tau_e + (\widehat{X}_{ij} - 1) \cdot M + \delta_j,\ \forall\ i \in \mathcal{V}^r \cup \{0\},\ \ j \in \mathcal{V}^r. \tag{5.10}$$

in which parameter $\widehat{X}_{ij}$ is a solution for the RMP. For each pair of consecutive nodes $i-j$ with $\widehat{X}_{ij} = 1$ in the schedule defined by the RMP, SP determines the shortest path with arcs and nodes defined by variables P_{eij} and N_{kij}, respectively. The actual value of the variable θ_j in the RMP is given by variable Z_j^r in the SP. Hence, if subproblem SP is feasible, we need to add optimality cuts to the RMP to update the cost of variable θ_j. Otherwise, if subproblem SP is infeasible, we need to add feasibility cuts to the RMP to cut off infeasible scheduling solutions. The feasibility and optimality cuts are defined by Propositions 1 (Section 4.2.3) and Proposition 5, respectively. In Proposition 5, $K = (v_0, v_1, ..., v_{(h-1)}, v_h, ..., v_p, ..., v_{|\mathcal{V}^r|})$ is a given schedule for the crew, where v_i is the ith damaged node to be repaired and $v_0 = 0$. This schedule is obtained by solving the RMP and corresponds to the solution $\widehat{X}_{v_{(i-1)}v_i} = 1$, $\forall i = 1, ..., |\mathcal{V}^r|$.

Proposition 5. *Assume that K is a feasible schedule, with values $\widehat{Z}_{v_j}^r$ computed in subproblem SP, for each $j \in \mathcal{V}^r$. Then, the following optimality multicuts are valid inequalities of the RMP:*

$$\theta_{v_j} \geq \widehat{Z}_{v_j}^r \cdot (\sum_{i=1}^{j} X_{v_{(i-1)}v_i} - (j-1)),\ \forall\ j \in \mathcal{V}^r, \tag{5.11}$$

$$\theta_{v_l} \geq (\widehat{Z}_{v_j}^r + t_{v_j v_l} + \delta_{v_l}) \cdot (\sum_{i=1}^{j} X_{v_{(i-1)}v_i} - (j-1)),\ \forall\ j \in \mathcal{V}^r,\ \ l \in \mathcal{V}^r, l > j, \tag{5.12}$$

in which t_{ij} is the minimum travel time from node $i \in \mathcal{V}$ to node $j \in \mathcal{V}$ and δ_j is the repair time of damaged node j.

Proof. In cuts (5.11), the term $\sum_{i=1}^{j} X_{v_{(i-1)}v_i} - (j-1)$ is equal to 1 if the partial schedule $K' = (v_0, v_1, ..., v_j)$ for a given j is part of the solution of the RMP. Therefore, the cuts become $\theta_{v_j} \geq \widehat{Z}_{v_j}^r$, and thus the lower bound $\widehat{Z}_{v_j}^r$ is activated for variable θ_{v_j}. If the partial schedule K' is not considered in the solution of the RMP, the term $\sum_{i=1}^{j} X_{v_{(i-1)}v_i} - (j-1)$ is smaller than 1 and the cuts become deactivated. Similarly, in cuts (5.12), if the partial schedule K' is considered in the RMP solution, the lower bound $\widehat{Z}_{v_j}^r + t_{v_j v_l} + \delta_{v_l}$ is activated for all nodes v_l with $l > j$. The lower bound $\widehat{Z}_{v_j}^r + t_{v_j v_l} + \delta_{v_l}$ is valid because it is known that nodes v_l need to be repaired at some moment after node v_j and the crew must spend at least $t_{v_j v_l}$ time units to reach node v_l from node v_j and δ_{v_l} time units to repair it. $\square$

Note that cuts (5.11) are sufficient for the variables θ_{v_j} assuming their actual values $\widehat{Z}_{v_j}^r$ in the RMP. However, the additional cuts (5.12) can be added to speed up the performance of the method. To illustrate Proposition 5, consider the schedule $K = (v_0, v_1, v_2) = (0, 2, 1)$ that is assumed to be feasible. Cuts (5.11) lead to $\theta_2 \geq \widehat{Z}_2^r \cdot (X_{02})$ and $\theta_1 \geq \widehat{Z}_1^r \cdot (X_{02} + X_{21})$. Additionally, cuts (5.12) result in $\theta_1 \geq (\widehat{Z}_2^r + t_{21} + \delta_1) \cdot (X_{02})$.

Together with the Benders cuts (4.32), (5.11) and (5.12), we propose the use of subtour elimination constraints (SECs) based on the Dantzig–Fulkerson–Johnson (DFJ) formulation (Dantzig et al., 1954). SECs based on DFJ can lead to stronger linear relaxations than those based on constraints (5.4) of the RMP. They are stated as follows:

$$\sum_{i\in S}\sum_{j\in \bar{S}} X_{ij} + \sum_{i\in \bar{S}}\sum_{j\in S} X_{ij} \geq 2, \forall\ S \subset \mathcal{V}^r : 2 \leq |S| \leq |\mathcal{V}^r| - 1, \tag{5.13}$$

in which $\bar{S} = \mathcal{V}_0^r \setminus S$. Because the formulation based on DFJ contains an exponentially large number of SECs, a typical strategy is to gradually add the constraints to the formulation through a branch-and-cut scheme (Adulyasak et al., 2015). In practice, relatively few SECs are needed (Öncan et al., 2009). In our BBC, we add them after solving the LP problem at each node of the branch-and-cut tree. To detect the violated subtour constraints in a fractional solution $\widehat{X}$, we solve a series of minimum $s-t$ cut problems in a support graph $G^* = (N^*, E^*)$, in which $N^* = \mathcal{V}_0^r$ and $E^* = \{(i,j)|\widehat{X}_{ij} > 0\}$. We set the depot as the source node s and the damaged node $i \in \mathcal{V}^r$ as the sink node t. A violated subtour constraint is identified every time the value of the resulting minimum cut is less than 2. Then, we separate the corresponding subtour constraint in the form of inequalities (5.13). In the implementation, we use the minimum $s-t$ cut algorithm of the Concorde Callable Library (Applegate et al., 2018). With constraints (5.13), we do not need constraints (5.4) in the RMP anymore. Nevertheless, we keep constraints (5.4) because they help to improve the lower bound value of the variables $\theta_j, \forall\ j \in \mathcal{V}^r$.

5.2.1 Valid inequalities

We derive valid inequalities to strengthen the LP relaxation of the RMP model and improve the convergence of the branch-and-cut algorithm. Valid inequalities can be of great importance in Benders-based methods because the decomposition causes the master problem to lose information of the variables considered in the subproblems.

If some damaged nodes are used in the path from the depot to the demand node j, all these damaged nodes need to be repaired before the demand node j becomes accessible. Then, the time at which the demand node j becomes accessible is higher than or equal to the sum of the repair times of the damaged nodes used in the path plus the minimum travel time to arrive at these damaged nodes. Based on this, we can define the following valid inequalities:

$$Z_j^d \geq \sum_{k\in\mathcal{V}^r} V_{kj} \cdot (\delta_k + t_k^*),\ \forall\ j \in \mathcal{V}^d, \tag{5.14}$$

in which $t_k^* = \min_{i\in\mathcal{V}_0^r : i\neq k}\{t_{ik}\},\ \forall\ k \in \mathcal{V}^r$.

The next set of valid inequalities is based on the maximum distance l_i allowed between the depot and the demand nodes. If the shortest distance from the depot to a node k plus the shortest distance from node k to the demand node i is greater than the maximum distance l_i, then node k cannot be used in the path from the depot to the demand node i. The valid inequalities are defined as follows:

$$V_{ki} \leq 0, \ \forall \ i \in \mathcal{V}^d, k \in \mathcal{V} : dist_{0k} + dist_{ki} > l_i, \tag{5.15}$$

in which $dist_{ki}$ is the shortest distance from node k to node i, and it is evaluated using Dijkstra's algorithm. In addition, we use all the valid inequalities proposed in Section 4.2.6.

5.3 Metaheuristic algorithms

In this section, we present a genetic algorithm (GA) and a simulated annealing (SA) tailored for the SCSRP. These metaheuristics operate on a TSP subproblem representing the scheduling decisions and call for specialized algorithms to optimize the crew routing and the relief path decisions as well as to determine the feasibility and cost of the proposed schedule in the original SCSRP. Therefore, the proposed metaheuristics do not explicitly consider all the possible crew routes and relief paths but only the best crew route and relief paths of a given scheduling solution, which significantly reduces the space of solutions explored by them. Additionally, they use five local search operators to diversify the search in the solution space. The way the operators are applied varies from one metaheuristic to the other, yet in both cases, they act as essential steps to escape from local optimal solutions. Such operators work by finding a neighbor of a current solution, and they can thus be seen as low-level heuristics that are selected by a higher-level algorithm (metaheuristic) that guides the search. This type of strategy has been referred to as hyper-heuristics (Burke et al., 2003; Drake et al., 2019), and successful applications of hyper-heuristics based on GA and SA metaheuristics can be found in the literature (Han and Kendall, 2003; Dowsland et al., 2007; Bai et al., 2012). Subsections 5.3.1 and 5.3.2 describe the proposed GA and SA methods, respectively.

5.3.1 Genetic algorithm

GA is a metaheuristic method based on three basic principles of the biological evolution process: reproduction, natural selection, and diversity of individuals. GAs have been used together with exact algorithms in a few applications (Lin et al., 2004; Poojari and Beasley, 2009) and are widely used in the context of routing problems (Karakatič and Podgorelec, 2015).

Algorithm 5 presents a basic scheme of the genetic algorithm developed in this work. GA starts with a population composed of a set of initial solutions (individuals) generated with a construction heuristic. The individuals are evaluated using a fitness function, and they evolve through a series of iterations (generations) by applying operators of selection of parents (natural selection), crossover (reproduction), and mutation (diversity of individuals). The procedure is repeated until it reaches some stopping criterion (e.g., maximum number of iterations or maximum computational time). The solution representation, construction heuristic, fitness function, selection, crossover and mutation of individuals are described in the following subsections.

Algorithm 5 Basic scheme of the GA metaheuristic.

```
1: Generate initial feasible individuals using the construction heuristic (see Section 5.3.1.3)
2: while stopping criteria are not reached do
3:     Evaluate the individuals with the feasibility and optimality check algorithms (see Section 5.3.1.2);
4:     Update the best solution;
5:     Perform selection (see Section 5.3.1.4) and crossover (see Section 5.3.1.5);
6:     Evaluate the individuals with the feasibility and optimality check algorithms (see Section 5.3.1.2);
7:     Perform mutation (see Section 5.3.1.6);
8: end while
```

5.3.1.1 Solution representation

Let $K = (v_0, v_1, \ldots, v_i, \ldots, v_{|\mathcal{V}^r|})$ be a schedule for the crew, in which v_i is the ith damaged node to be repaired and $v_0 = 0$ is the depot node. To represent the solution corresponding to schedule K, we use a vector with $|\mathcal{V}^r| + 2$ positions. The first $|\mathcal{V}^r| + 1$ positions indicate the order of the damaged nodes in the schedule of the crew, whereas the last position indicates the objective value of the solution in the original problem. Figure 5.1 shows an example of a vector representing a solution with 5 damaged nodes and an objective value of 100. Infeasible solutions are represented with an objective value of ∞ in the last position of the vector.

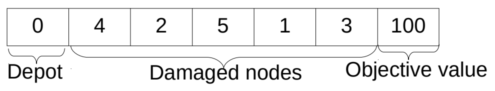


Figure 5.1: Vector representing a solution of the SCSRP problem.

Note that only scheduling decisions are represented in a solution, as depicted in Figure 5.1. For each fixed scheduling decision, we can optimally define the paths between damaged nodes and between the depot and the demand nodes using Dijkstra-based algorithms. Thus, the other decisions of the SCSRP can be straightforwardly derived from a given crew schedule.

5.3.1.2 Evaluation of individuals

A fitness function in GA takes a candidate solution of the problem and returns a value that represents the quality of the solution. In our GA algorithm, the quality of a given solution is evaluated based on the cost of the corresponding schedule K in the original SCSRP formulation. To evaluate this cost, we use two algorithms proposed in Section 4.2.4, which we call the feasibility check algorithm and the optimality check algorithm. The feasibility check algorithm (Algorithm 1) takes a schedule K and verifies whether it is feasible or not for the original SCSRP. If the solution is infeasible, we set a cost equal to ∞ in the last position of the solution vector. Otherwise, the algorithm returns the exact time at which the damaged nodes in schedule K are repaired by the crew. Subsequently, we call the optimality check algorithm (Algorithm 2) that evaluates the exact time at which the demand nodes become accessible and the total cost for the solution in the original SCSRP.

5.3.1.3 Initial population

Our initial population is created by using a construction heuristic that generates feasible solutions for the SCSRP. The heuristic consists of generating a feasible solution by sequentially adding damaged nodes to the schedule in an iterative process. The construction heuristic is outlined in Algorithm 6. Let $K = (v_0, v_1, \dots, v_l)$ be a partial schedule having the depot node $v_0 = 0$ and l nodes from set $\mathcal{V}^r$, for a given $l > 0$. Let $\mathcal{F}_K$ be the set of the nodes that can be reached from v_l, the last node added to this partial schedule. A node $j \in \mathcal{V}^r$ can be reached from v_l if there exists a path from v_l to j without using a damaged node that has not been repaired yet, and node j is not in the partial schedule K. In such a case, we say that j is a feasible node.

In the construction heuristic, we initially have $K = (v_0)$. At the end of each iteration, we randomly select a feasible node from the set $\mathcal{F}_K$, which is added to the end of the partial schedule K (line 12 of Algorithm 6). To update the set $\mathcal{F}_K$, we need to remove from it the last node v_l added to K (line 3), and then we execute Dijkstra's algorithm to find paths between node v_l and the nodes in $\mathcal{V}^r \setminus \mathcal{F}_K$ that are not in the partial schedule yet (line 6). If there is a path from v_l to some node $j \in \mathcal{V}^r \setminus \mathcal{F}_K$ that is not in K, then node j is added to the set $\mathcal{F}_K$ (line 8). Finally, when schedule K is completed, the solution is evaluated using the feasibility and optimality check algorithms (line 14). Notice that only feasible nodes can be selected at each iteration, and, as a consequence, feasible solutions are always generated.

Algorithm 6 construction heuristic for the SCSRP.

```
1: Set node 0 as the first node in schedule $K$, i.e., $v_0 := 0$, $K = (v_0)$, $\mathcal{F}_K = \{v_0\}$;
2: for $l = 0$ to $|\mathcal{V}^r| - 1$ do
3:     Remove node $v_l$ from set $\mathcal{F}_K$;
4:     for $j = 1$ to $|\mathcal{V}^r|$ do
5:         if node $j$ is in $\mathcal{V}^r \setminus \mathcal{F}_K$ but not in $K$ then
6:             Find a path between node $v_l$ and node $j$ without using damaged nodes not repaired yet;
7:             if a path between node $v_l$ and node $j$ exists then
8:                 Add node $j$ to set $\mathcal{F}_K$;
9:             end if
10:        end if
11:    end for
12:    Randomly select a node $i \in \mathcal{F}_K$ and add it to schedule $K$, i.e., set $v_{(l+1)} := i$;
13: end for
14: Compute the cost of schedule $K$ using the feasibility and optimality check algorithms;
```

5.3.1.4 Parent selection

The selection of individuals to evolve from one generation to the next one is based on a tournament with k competitors, a so-called k-tournament, in which k individuals are randomly compared and the best (the one with the smallest cost) is selected for the reproduction step. A tournament with multiple competitors may yield good solutions quickly but has a higher chance of settling at local optima (Karakatič and Podgorelec, 2015). To avoid this issue, the k-tournament selection can be combined with aggressive mutation strategies. The selected best individuals can be crossed over by exchanging pieces with others and can either mutate or remain unaltered until the next generation. Additionally, we apply an elitist strategy that consists of always passing the best individual of the population from one generation to the next one.

5.3.1.5 Crossover

We implemented three popular crossover operators for permutation representation: partial mapped crossover (PMX), ordered crossover (OX), and cycle crossover (CX) (Larranaga et al., 1999; Kumar et al., 2012). When the individuals are selected for reproduction, one of the three strategies is randomly applied to generate the offspring.

5.3.1.6 Mutation

Mutation is usually used as an operation to prevent the GA from getting stuck on local optimal solutions (Balin, 2011). In our GA, the individuals are randomly selected for mutation if they represent feasible solutions. On the other hand, when an individual represents an infeasible solution, we force its mutation in an attempt to make it feasible. We use five local search operators as mutation strategies: swap, 2-opt, or-opt-1, or-opt-2, and or-opt-3. When the individuals are selected for mutation, one of these five strategies is randomly applied to generate the offspring. The first local search operator is an exchange (swap) of position of two damaged nodes in the schedule. The second local search operator is a pairwise exchange (2-opt) that involves removing two edges and replacing them with two different edges that reconnect the fragments created. The three last local search operators are repositioning operators (or-opt-k) in which k adjacent nodes are removed from the schedule and reinserted at a different location in the schedule.

5.3.1.7 Parameters of the GA metaheuristic

In the GA proposed in this work, we have to adjust the following parameters to guarantee a better performance of the metaheuristic: maximum number of iterations, size of the population, selection probability, mutation probability, and parameter k for the k-tournament. The values selected for the parameters are described in Section 5.5.1.

5.3.2 Simulated annealing algorithm

SA is a randomized search method that exploits an analogy with the thermodynamic process of the cooling of metals, gradually adjusting a parameter called “temperature”. At high temperatures, the method searches in a large space of solutions, while at low temperatures, solutions with worsening objective values are less likely to be accepted. SA has the advantage of usually being easier to implement and less time consuming than more sophisticated metaheuristics while still providing good overall results (Galvão et al., 2005), in particular for related problems such as the TSP (Ohlmann and Thomas, 2007) and the inventory routing problem (Alvarez et al., 2018). Additionally, as pointed out by Gogna and Tayal (2013) and enforced by our computational experiments, SA is well suited to problems with a large number of local optima, such as the SCSRP.

A basic scheme of the SA implemented in this study is presented in Algorithm 7. The representation of the solution in our SA is the same as that used in our GA, i.e., a vector representing the scheduling decisions and the objective value of the solution. SA starts with a set of multiple initial solutions generated with the construction heuristic (Algorithm 6). At each iteration, one neighbor of each solution is randomly derived using one of the five local search operators (swap, 2-opt, or-opt-1, or-opt-2, and or-opt-3) defined in Subsection 5.3.1.6. The neighbors are evaluated by using the feasibility and optimality check algorithms presented in Subsection 5.3.1.2. We perform a few inner iterations before updating the temperature value T ($T > 0$). At the end of the inner iterations, we update the temperature value, and all the solutions are replaced by the best solution found.

Algorithm 7 Basic scheme of the SA metaheuristic.

```
1:  Generate multiple initial feasible solutions using the construction heuristic (see Algorithm 6);
2:  while stopping criteria is not reached do
3:      while inner iterations are not completed do
4:          Apply local search operators to derive the neighbors of the solutions (see Section 5.3.1.6);
5:          Evaluate the solutions with the feasibility and optimality check algorithms (see Section 5.3.1.2);
6:          Check the acceptance criteria and update the solutions (see Section 5.3.2.1);
7:      end while
8:      Update the best solution;
9:      Update temperature;
10:     Replace all solutions by the best solution;
11: end while
```

5.3.2.1 Acceptance criteria

In the inner iterations of the proposed SA algorithm, a solution corresponding to a schedule K can be replaced by a neighbor solution corresponding to a different schedule K' according to an acceptance criterion probability given by $\min\{1, e^{\Delta/T}\}$, where $\Delta = z^K - z^{K'}$ and z^K ($z^{K'}$) denotes the objective value of schedule K (K') in the SCSRP. If $z^K \geq z^{K'}$, then $\min\{1, e^{\Delta/T}\} = 1$. In this case, as K' is better than or equal to K, it is accepted and replaces K. Otherwise, K' can still be accepted to replace K with probability $\min\{1, e^{\Delta/T}\} < 1$. If K' is infeasible, then it is not accepted because $\Delta \to -\infty$ and $\min\{1, e^{\Delta/T}\} \to 0$. Note that higher values of T at the beginning of the method imply higher probabilities of acceptance. On the other hand, smaller values of T in the last iterations of the SA imply smaller probabilities of accepting worse solutions.

5.3.2.2 Parameters of the SA metaheuristic

The SA proposed in this study has the following parameters that need to be adjusted by the user: maximum number of iterations, number of inner iterations per temperature, number of multiple initial solutions, initial temperature, minimum temperature, and cooling rate. The values selected for the parameters are described in Section 5.5.1.

5.4 Hybrid branch-and-Benders-cut algorithm (HBBC)

The hybrid approach combines the BBC algorithm developed in Section 5.2 and the metaheuristic algorithms proposed in Section 5.3. More specifically, we call a metaheuristic inside the branch-and-cut tree to explore the neighborhood of the current incumbent solution. The solutions found in the neighborhood of the current incumbent are used to add feasibility and optimality cuts to the RMP. The cuts added from the metaheuristic solutions are as defined in (4.32), (5.11), (5.12), and although they are not required to guarantee the optimality or feasibility of the solutions in the RMP, they can accelerate the convergence of the BBC method. The cuts required to guarantee the feasibility and optimality of the solutions in the RMP are still generated by solving subproblem SP. Therefore, the HBBC is an exact method.

Figure 5.2 depicts a basic scheme of the HBBC algorithm at each node of the branch-and-cut tree. At each node i, we solve the linear relaxation of the current RMP, denoted by LP_i. If LP_i is infeasible or if the objective value of the LP_i solution (OF_i) is greater than or equal to the objective value of the current incumbent solution, then node i is pruned. Otherwise, we solve the minimum cut problem to identify violated SECs of type (5.13). If necessary, we add the violated SECs, and LP_i is solved again. Otherwise, the integrality constraints are checked, and if any component of the binary variables is fractional in the solution of LP_i, then the branching is performed. Before performing the branching, we call a metaheuristic at the corresponding node i with the fractional solution. In this case, the initial metaheuristic solutions (initial population for GA or initial multistart for SA) are composed of (i) an integer solution obtained from a rounding heuristic; (ii) the incumbent solution; and (iii) solutions randomly generated. The rounding heuristic works as follows. Starting in the depot, let i be the last node added to schedule K, and select the damaged node j with the highest value $\widehat{X}_{ij}$ that has not been included in K so far. Then, include j at the end of K, and repeat the process iteratively until schedule K is completed.

Every time the LP_i solution is integer feasible, we call the subproblem SP to verify the violation of feasibility or optimality cuts. Since we keep constraints (5.4) in the RMP, there is no need to verify violated SECs for the integer solutions. If no feasibility or optimality cuts are obtained, then the LP_i solution is feasible for the original problem and the solution is set as the new incumbent solution. Otherwise, LP_i must be resolved, and the previous steps are applied again. If a feasible integer solution of the SCSRP is found, we call the metaheuristic to improve this solution and generate additional cuts. In this case, the initial metaheuristic solutions are composed of the last found integer solution, the incumbent solution, and solutions randomly generated. If the metaheuristics find a better solution, the current incumbent solution is updated.

Note that the metaheuristics are called at most once per node to avoid stagnation in infinite cycles. Since calling a metaheuristic at every node is inefficient because many unnecessary cuts may be added and perhaps much time spent, we call the metaheuristics only at nodes with an integer solution and at some predefined nodes with a fractional solution (for example,

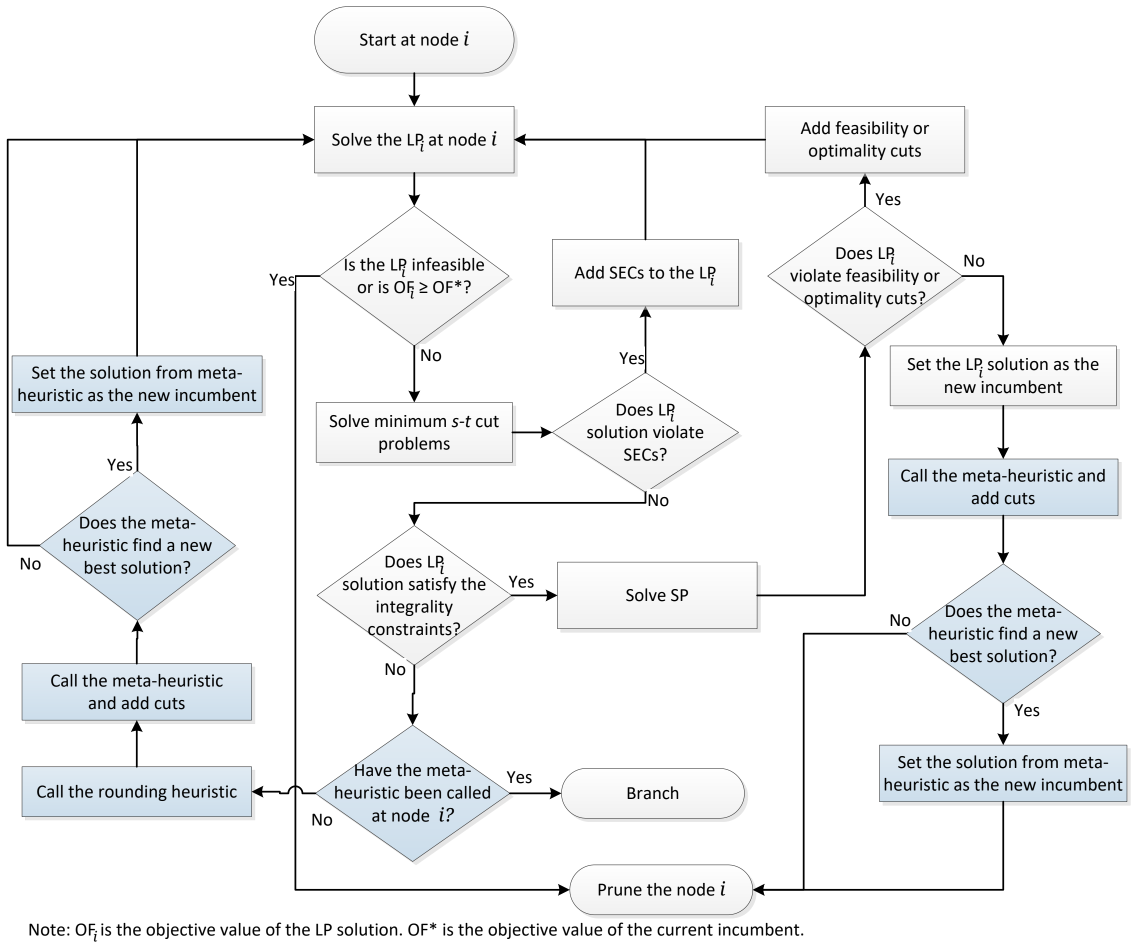

Start at node i
Solve the LP$_i$ at node i
Add feasibility or optimality cuts
Yes
Is the LP$_i$ infeasible or is OF$_i$ ≥ OF*?
Yes
No
Add SECs to the LP$_i$
Does LP$_i$ violate feasibility or optimality cuts?
No
Set the solution from meta-heuristic as the new incumbent
Yes
Solve minimum s-t cut problems
Does LP$_i$ solution violate SECs?
Set the LP$_i$ solution as the new incumbent
Yes
Call the meta-heuristic and add cuts
No
Does the meta-heuristic find a new best solution?
No
Does LP$_i$ solution satisfy the integrality constraints?
Yes
Solve SP
No
Does the meta-heuristic find a new best solution?
Call the meta-heuristic and add cuts
No
Yes
Call the rounding heuristic
Have the meta-heuristic been called at node i?
Yes
Branch
No
Set the solution from meta-heuristic as the new incumbent
Prune the node i
Note: OF$_i$ is the objective value of the LP solution. OF* is the objective value of the current incumbent.

Figure 5.2: Flowchart illustrating how the metaheuristics are combined with the BBC method.

at the first 100 nodes of the tree). Similarly, it could be inefficient to call the separation procedure to add violated SECs at each node of the tree. Therefore, we call the separation procedure to detect violated SECs at the same nodes with a fractional solution for which the metaheuristics are called. The cuts generated by the metaheuristic in a node i with a fractional solution are immediately added to the LP$_i$ subproblem, and the subproblem is solved again. The cuts generated by the metaheuristic in a node i with an integer solution are added to a pool of constraints checked later in the tree. Some general-purpose optimization software can use automated cuts and/or heuristics that are not included in Figure 5.2.

5.5 Computational experiments

In this section, we present experimentation campaigns conducted with the scope of comparing the performance of the proposed solution methods. All the methods were coded in C++ programming language and run on a Linux PC with an Intel Core i7 CPU at 3.4 GHz and 16 GB of RAM using a single thread. SECs and Benders cuts are added using the Callback classes available in the Concert Technology Library. The RMP is solved by CPLEX Optimization Solver 12.7. The SP problem is solved by the specialized Algorithm 1 proposed in Section 4.2.4

instead of using the CPLEX solver. To avoid running out of memory, we allow CPLEX to store the branch-and-bound tree in a file. As we use lazy constraints Callback, CPLEX automatically turns off nonlinear reductions and dual reductions. The stopping criterion on CPLEX was either the elapsed time exceeding the time limit of 3,600 seconds or the optimality gap being smaller than 10^{-4}. All the remaining parameters of CPLEX were kept at their default values for most of the computational experiments. We also conduct a few experiments varying the default value of some CPLEX parameters, as reported in Section 5.5.3. For the metaheuristics GA and SA, the stopping criterion was given by either the elapsed time exceeding the time limit of 3,600 seconds or by performing 500 iterations without improving the best solution. As the metaheuristics SA and GA presented very similar overall results (as shown in Table 5.2), we performed the computational experiments of the HBBC method using only SA, which obtained solutions with a smaller average cost. The time limit of SA inside the nodes of the branch-and-cut tree was set as 600 seconds in the root node and 60 seconds in the other nodes. The algorithms were tested using instances generated from original networks with up to 100 nodes, i.e., instances from the first 24 instances classes presented in Table 4.1 in Section 4.3.1.

5.5.1 Parameter tuning

As in most metaheuristics, the satisfactory performance of both the GA and the SA depends on the configuration choices for a set of key parameters. To appropriately calibrate such parameters, we use the ParamILS algorithm (Hutter et al., 2009). It is an automated tuning method that has shown very good performance in many applications (Montero et al., 2014). ParamILS iteratively improves the performance of a set of parameter configurations by searching in its neighborhood for another configuration with better quality. For this purpose, an initial configuration and discrete ranges for the set of parameters must be provided by the user. We also tested different strategies for the generation of cuts in the HBBC approach. The parameter tuning was carried out with 48 instances of different sizes, two from each class. The instances were solved many times using different random seeds.

Table 5.1 shows the parameters of the metaheuristics GA and SA, the discrete ranges defined according to preliminary experiments for each parameter and their values in the best configuration found by ParamILS. Note that, for both the metaheuristics, the size of the initial solution set is relatively small (10 solutions). This behavior was expected since our evaluation of the solutions (feasibility and optimality check algorithms) is very expensive to be performed over a large set of solutions at each iteration. The other expected result was the high mutation probability for the GA. The SCSRP is a degenerate problem (we can find several solutions with the same cost in the same neighborhood), which causes stagnation in local optimal solutions. The frequent application of mutation operators was shown to be a good strategy to escape from local optima. Recall that the mutation probability is defined only for feasible solutions, as the infeasible solutions are always forced to mutate.

For the HBBC, we varied the frequency of application of the metaheuristics in the branch-

Table 5.1: Ranges and best configurations for the parameters of the GA, SA and HBBC methods.

Solution method	Parameter	Tested values	Final value
GA	Size of the population	{5, 10, 15, 20, 25}	10
	Selection probability	{0.90, 0.92, 0.94, 0.96, 0.98}	0.98
	Mutation probability	{0.05, 0.1, 0.2, 0.3, 0.4, 0.5}	0.5
	Parameter k	{2, 3, 4, 5}	3
SA	Size of the set of initial solutions	{5, 10, 15, 20, 25}	10
	Inner iterations per temperature	{2, 3, 4, 5}	2
	Initial temperature	{100, 300, 500, 1000, 1500}	500
	Minimum temperature	{1E-6, 1E-4, 0.01, 0.1}	1E-6
	Cooling rate	{0.95, 0.96, 0.97, 0.98, 0.99}	0.99
HBBC	Frequency	(1) Only at root node (600 seconds)[1] and at nodes with integer solutions (30 seconds)[1] (2) (1) + every 100 nodes (30 seconds)[1] (3) (1) + at the first 100 nodes (30 seconds)[1] (4) (1) + decreasing frequency[2]	(4)
	Maximum number of solutions	{50, 100, 500, $5 \cdot \mathcal{V}^r$, $10 \cdot \mathcal{V}^r$, $20 \cdot \mathcal{V}^r$}	$10 \cdot \mathcal{V}^r$
	Type of cuts	(1) Optimality + feasibility cuts. (2) Only optimality cuts. (3) Only feasibility cuts.	(2)

[1] Time limit set for the metaheuristics at the nodes.
[2] From node 0 to node 100, we call the metaheuristics at every 10 nodes; from node 100 to node 1000, we call the metaheuristics at every 100 nodes; and from node 1000 onward, we call the metaheuristics at every 1000 nodes.

and-cut tree and the quantity and type of cuts that are added to the master problem from the metaheuristic solutions. Table 5.1 also shows the parameters tested for the HBBC method. For instance, the parameter frequency for the HBBC method defines how often the metaheuristics are called in the branch-and-cut tree. Four options were evaluated for this parameter: (1) apply the metaheuristics only at the first three nodes and at nodes with an integer solution; (2) apply the metaheuristics at the nodes of option (1) and additionally at every 100 nodes with a fractional solution; (3) apply the metaheuristics at the nodes of option (1) and additionally at the first 100 nodes with a fractional solution; and (4) apply the metaheuristics at the nodes of option (1) and at nodes with a fractional solution in a decreasing frequency. The decreasing frequency consists of calling the metaheuristics at every 10 nodes for nodes 0 to 100; at every 100 nodes for nodes 100 to 1000; and at every 1000 nodes for node 1000 onward. The last strategy showed the best performance, as we observe that it is more likely to find new best solutions with the metaheuristics at the first nodes of the tree when the incumbent solution is farther from the optimal solution than in the last nodes of the tree. Additionally, it is more likely to generate useful cuts in the first nodes because fewer cuts have been added to the RMP. Regarding the type of cuts derived from the metaheuristic solutions, it is more useful to add only optimality cuts in the HBBC, as they have a greater impact on the lower bound than feasibility cuts. Finally, we limited the number of solutions of the metaheuristics from which we can add cuts to the RMP since the generation of a large number of cuts can slow down the linear subproblems at the nodes. The computational experiments presented in the next sections were conducted using the best configuration found by the ParamILS algorithm. The separation routines for the SECs at nodes with fractional solutions are called at the same nodes as the metaheuristics.

5.5.2 Computational performance of the metaheuristic approaches

This section presents the results of the metaheuristics GA and SA. Before running the computational experiments with all the instances, we solved each one of the 48 instances used in the parameter tuning 10 times with different random seeds. The objective of this experiment was to analyze the differences in the quality of the solutions provided by the metaheuristics. The tests were performed setting a time limit of 3,600 seconds for each run. The average, maximum and minimum objective values over the ten repetitions for each class are shown in Table 5.2. We also evaluated the relative difference of the maximum value with respect to the minimum value ($\frac{\text{Max}-\text{Min}}{\text{Min}}$). Note that, on average, the maximum value over the ten repetitions was only 1.57% and 1.17% higher than the minimum value for metaheuristics GA and SA, respectively. In fact, for some instances, the result over the ten repetitions was the same, while for the other instances, the difference of the maximum value in relation to the minimum value was never higher than 8.12% for the GA and 8.28% for SA. Furthermore, such difference was higher than 4% in only 5 out of the 24 instance classes in the metaheuristic GA and in only 1 out of the 24 classes in the metaheuristic SA. Basically, in the instances with a smaller number of damaged nodes, the space of solutions to be explored is smaller than in the other instances, and the metaheuristics are able to find similar near-optimal solutions over the ten repetitions.

Table 5.2: Average results of the SA and GA metaheuristics for the instances used in the parameter tuning.

	GA					SA				
Class	Avg. cost	Min. cost	Max. cost	$\frac{\text{Max}-\text{Min}}{\text{Min}}$ (%)	Avg. time (sec.)	Avg. cost	Min. cost	Max. cost	$\frac{\text{Max}-\text{Min}}{\text{Min}}$ (%)	Avg. time (sec.)
1	22,675	22,675	22,675	0.00	7.71	22,675	22,675	22,675	0.00	2.04
2	76,367	76,367	76,367	0.00	5.12	76,367	76,367	76,367	0.00	1.44
3	60,645	60,645	60,645	0.00	6.72	60,645	60,645	60,645	0.00	1.99
4	50,479	50,479	50,479	0.00	23.91	50,479	50,479	50,479	0.00	6.94
5	36,944	36,944	36,944	0.00	29.18	36,944	36,944	36,944	0.00	9.12
6	60,841	60,841	60,841	0.00	16.57	60,841	60,841	60,841	0.00	6.35
7	50,416	50,416	50,416	0.00	36.38	50,416	50,416	50,416	0.00	13.82
8	69,196	69,196	69,196	0.00	60.77	69,196	69,196	69,196	0.00	19.62
9	64,564	64,564	64,564	0.00	53.16	64,564	64,564	64,564	0.00	20.57
10	68,605	68,605	68,605	0.00	5.65	68,605	68,605	68,605	0.00	1.52
11	76,268	76,268	76,268	0.00	5.44	76,268	76,268	76,268	0.00	1.48
12	64,619	64,619	64,619	0.00	3.37	64,619	64,619	64,619	0.00	0.95
13	46,179	46,179	46,179	0.00	68.70	46,179	46,179	46,179	0.00	19.37
14	129,183	129,108	129,407	0.23	95.22	129,108	129,108	129,108	0.00	36.43
15	74,357	74,357	74,357	0.00	99.06	74,376	74,357	74,431	0.10	37.77
16	89,097	88,956	89,338	0.43	256.58	89,051	88,956	89,298	0.38	110.53
17	44,169	43,971	44,333	0.82	172.76	44,040	43,878	44,452	1.31	74.38
18	134,518	134,024	135,526	1.12	629.64	134,042	132,332	136,241	2.95	209.90
19	64,070	62,389	66,253	6.19	942.34	61,808	61,057	62,626	2.57	382.37
20	85,896	81,169	87,756	8.12	682.38	76,732	75,390	78,039	3.51	261.72
21	77,520	74,540	79,548	6.72	1,352.30	69,449	68,696	70,790	3.05	529.08
22	246,686	245,172	247,895	1.11	2,129.45	238,805	235,724	244,200	3.60	854.04
23	100,642	97,625	105,440	8.01	2,154.62	95,131	92,772	100,453	8.28	677.67
24	188,774	184,512	193,558	4.90	2,511.13	182,371	180,337	184,546	2.33	1,055.83
Avg.	82,613	81,817	83,384	1.57	472.84	80,946	80,433	81,749	1.17	180.62

Note in Table 5.2 that the average solution cost is slightly smaller when the problem is solved by SA, which provided solutions with an average objective value 2.02% better than the solutions of the GA for an execution time of 3,600 seconds. Figure 5.3 shows the improvement of the average objective function value of the 48 instances solved by the metaheuristics SA and

GA along 3,600 seconds of execution in relation to the average objective function value of the initial solutions. As expected, we observed that for both the metaheuristics, the reduction in the objective function value is faster in the first iterations. For the GA (SA), the average cost of the initial solutions is 272,302 (291,194), which decreases to 106,813 (109,055) in 600 seconds and to 82,613 (80,946) in 3,600 seconds. Thus, for the instances solved by the GA (SA), the average objective function value decreases by 60.77% (62.55%) in the first 600 seconds and by 69.66% (72.20%) in 3,600 seconds. The SA presented a slightly better performance in terms of the objective function values since the problem typically has many local optimal solutions with the same objective value in a neighborhood and SA has the ability to avoid becoming trapped in local optimal solutions. Moreover, the average computational time of SA was 61.8% smaller than that of the GA. It is worth mentioning that we also ran additional experiments with a longer time limit (3 hours), but the overall improvement was less than 1% on average.

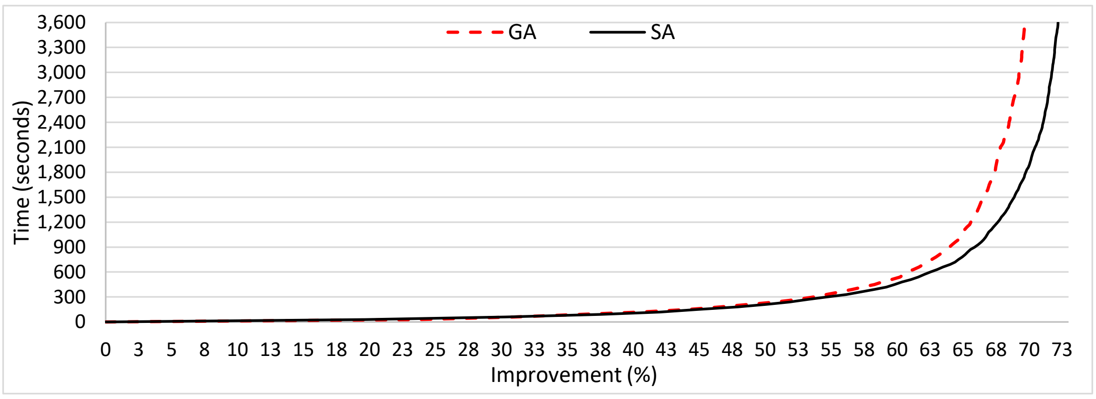


Figure 5.3: Average cost of the 48 instances solved by SA and the GA along 3,600 seconds of execution.

Table 5.3 shows the average results of the metaheuristic SA considering all instances of the classes presented in the first column. Since we observed a robust behavior over the 10 runs of our SA in the results reported in Table 5.2, we carried out the experiment reported in Table 5.3 with a single run of this method. We compare the results of our SA with the results of the GRASP metaheuristic of Maya-Duque et al. (2016), which was the only metaheuristic so far developed to solve the same variant of the problem addressed in this work. We present the results of GRASP for classes 1-15 only because we did not have access to the solutions of the other classes. We also compare the results of our SA with the best variant of the exact BBC proposed in Chapter 4, referred to as GR-BBC0.

Columns 6 and 7 of Table 5.3 show the relative reduction in the objective value of the solutions found with the SA with respect to the solutions found with GRASP and GR-BBC0, respectively. On average, for classes 1-15, the cost of the solutions of the metaheuristic SA decreases by 1.46% and 0.18% in relation to the cost of the solutions of GRASP and GR-BBC0, respectively. The improvements with respect to GRASP were up to 7.02% in classes 1-15. In relation to GR-BBC0, the higher improvement in classes 1-15 was only 2.22% since the solutions obtained with GR-BBC0 for these instances are close to the optimal solutions. For the other classes, the improvements were up to 55%. The improvement for the larger instance classes comes from the effectiveness of the local search operators that helps SA to escape from local optimal solutions. The GR-BBC0 approach proposed in Chapter 4 stagnates in local optima for larger instances of the problem.

Table 5.3: Average results of the metaheuristic SA for the different instance classes (time limit of 3,600 seconds).

Class	Objective function value			$\frac{\text{GRASP}-\text{SA}}{\text{GRASP}}$ (%)	$\frac{\text{GR-BBC0}-\text{SA}}{\text{GR-BBC0}}$ (%)
	GRASP[1]	GR-BBC0[1]	SA		
1	9,745	9,745	9,745	0.00	0.00
2	34,089	34,089	34,089	0.00	0.00
3	49,987	49,862	49,862	0.25	0.00
4	18,247	18,037	18,122	0.68	−0.47
5	18,152	18,485	18,074	0.43	2.22
6	21,253	20,917	20,917	1.58	0.00
7	36,873	36,511	36,511	0.98	0.00
8	26,382	26,049	26,146	0.90	−0.37
9	35,224	33,953	33,903	3.75	0.15
10	48,546	48,460	48,460	0.18	0.00
11	39,213	38,538	38,765	1.14	−0.59
12	28,876	28,037	28,037	2.91	0.00
13	23,536	23,566	23,528	0.03	0.16
14	87,163	81,032	81,042	7.02	−0.01
15	52,085	52,201	51,385	1.34	1.56
16	–	38,737	38,851	–	−0.30
17	–	30,448	30,537	–	−0.29
18	–	97,476	95,516	–	2.01
19	–	65,093	46,048	–	29.26
20	–	71,173	42,037	–	40.94
21	–	75,602	59,025	–	21.93
22	–	211,487	146,526	–	30.72
23	–	98,827	58,012	–	41.30
24	–	209,435	93,868	–	55.18
Avg. 1-15	35,291	34,632	34,572	1.41	0.18
Avg. 16-24	–	99,809	67,824	–	24.53
Avg. All	–	59,073	47,042	–	9.31

The character "–" indicates no available value.
[1] GRASP proposed in Maya-Duque et al. (2016) and BBC proposed in Chapter 4.

5.5.3 Computational performance of the exact approaches

In this section, we analyze the performance of our BBC and HBBC methods. Table 5.4 summarizes the different solution strategies that we tested. The first two strategies BBC1 and BBC2 compare the new Benders reformulation of the problem with and without using the valid inequalities (VIs) defined in Section 5.2.1. The BBC3 strategy shows the impact of adding the SECs dynamically together with the Benders cuts. All the HBBC strategies are based on BBC3. In the HBBC1 strategy, the SA metaheuristic is used in the root node to find good-quality initial solutions for the problem but without generating feasibility and/or optimality cuts. In the HBBC2 method, on the other hand, the metaheuristic is additionally called in the nodes with integer solutions and in some nodes with fractional solutions to improve the incumbent solution and to generate Benders cuts.

The modification of some default parameters of the solver can positively influence the performance of the branch-and-cut method (Baz et al., 2009; Moreno et al., 2016, 2018). Therefore, the BBC3 and HBBC2 approaches were also tested varying the default configuration of the solver

CPLEX to solve the RMP, leading to BBC3* and HBBC2*. For both, we changed parameters that could lead to improvements in the lower bound of the solutions. We modified the emphasis of the branch-and-cut algorithm to optimality rather than feasibility, setting the CPLEX parameter `MIPEmphasis = 3`. This configuration increases the lower bound faster but possibly with a poor detection of feasible solutions along the optimization. We also set the selection of nodes to be processed according to the node with the smallest objective function for the associated LP relaxation, setting the CPLEX parameter `NodeSel = 1`. This strategy looks at the nodes with smaller bounds to improve them first. Finally, we modified the order of separation of the different types of cuts at nodes with fractional solutions, using method *isAfterCutLoop()* of the callback procedures. In the root node, we keep the default settings of CPLEX, while in the remaining nodes, we call the metaheuristic and add our Benders cuts only after all automatized cuts of CPLEX have been generated (i.e., if *isAfterCutLoop()* returns `True`). This way, we avoid generating Benders cuts on fractional solutions that may be cut off by subsequent automatized cuts of CPLEX.

In the last strategy of Table 5.4, GR-HBBC2*, we apply the graph reduction (GR) strategy described in Section 4.2.7 to speed up the convergence of the solution method. Basically, the idea of the GR is to solve the problem over different subgraphs with a reduced number of demand and damaged nodes and derive lower bounds for the variables of the original problem based on the solution of the reduced subgraphs. The subgraphs are usually generated from an initial feasible solution of the problem, and the performance of the GR strategy highly depends on this initial solution. Since we do not have a trivial initial solution for the BBC strategies, we consider the GR only with the best HBBC approach.

Table 5.4: Characteristics of the solution methods.

Solution method	Description
BBC1	New Benders reformulation of the problem.
BBC2	BBC1 + VIs.
BBC3	BBC2 + SECs.
BBC3*	BBC3 varying the default configurations of some CPLEX parameters.
HBBC1	BBC3 + SA in the root node.
HBBC2	HBBC1 + SA in nodes with integer solution and in some nodes with fractional solutions.
HBBC2*	HBBC2 varying the default configurations of some CPLEX parameters.
GR-HBBC2*	HBBC2* + Graph reduction strategy.

Figure 5.4 presents the performance profiles (Dolan and Moré, 2002) for the proposed approaches. The performance is based on the optimality gap, computed as $gap = \frac{Z^U - Z^L}{Z^U}$, in which Z^U is the upper bound or cost of the best integer solution and Z^L is the lower bound. The value $P(f, q)$ (y-axis) when $q > 0$ (x-axis) indicates the fraction of instances for which a strategy f provides solutions with a gap within a factor of 2^q of the best obtained gap. The value of $P(f, q)$ when $q = 0$ represents the fraction of instances for which the strategy f reached the best gap. For a given instance, the best gap is the lowest gap found considering all the approaches. Clearly, the hybrid strategies outperform the standalone BBC strategies. Also, we can observe that the GR strategy significantly improves the performance of the HBBC approaches.

Table 5.5 shows the number of optimal solutions (#opt), the proportion of optimal solutions

(%opt), the average bounds and gap, and the average elapsed time of the different solution strategies. For the sake of comparison, the table also shows the results of the best BBC method developed in Chapter 4, referred to as methods BBC0 and GR-BBC0. BBC0 uses a simple heuristic to provide an initial solution for the BBC algorithm, while GR-BBC0 additionally relies on graph reduction. The average results of the exact solution methods for different instance classes grouped according to the size of the network are presented in the Apendix C.

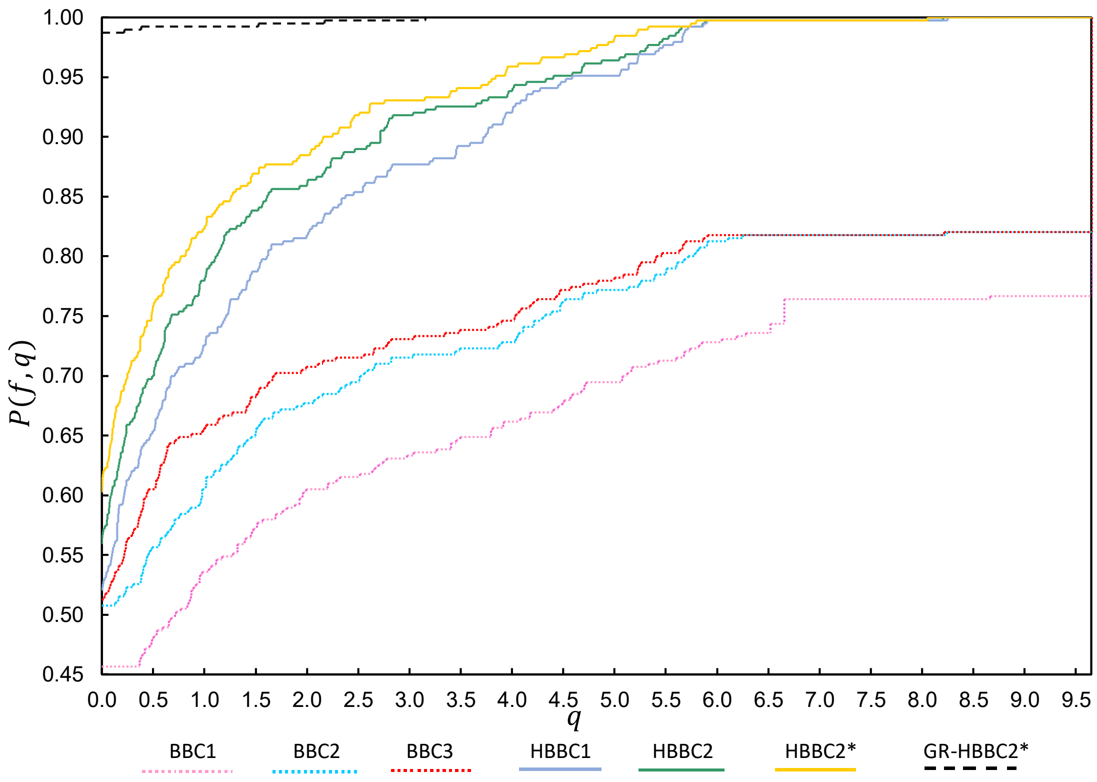


Figure 5.4: Performance profiles based on gap for the proposed solution methods.

Table 5.5: Comparison of the exact BBC and HBBC approaches.

Solution method	#ins	#opt	%opt	Avg. upper bound	Avg. lower bound	Avg. gap (%)	Avg. time (sec.)	Avg. best time[1](sec.)
BBC1	390	169	43.33	113,048	17,427	28.73	2,238.42	2,058.13
BBC2	390	171	43.85	110,763	20,494	25.03	2,201.55	1,889.79
BBC3	390	171	43.85	110,291	20,499	24.99	2,203.59	1,891.82
BBC3*	390	171	43.85	117,355	21,157	24.67	2,215.35	1,910.44
HBBC1	390	201	51.54	42,377	23,625	21.01	1,762.74	59.23
HBBC2	390	215	55.13	42,187	24,089	19.75	1,646.56	92.48
HBBC2*	390	215	55.13	42,158	24,855	18.15	1,651.39	83.35
BBC0[2]	390	186	47.69	53,342	16,042	31.87	1,906.38	450.57
GR-HBBC2*	390	239	61.28	42,118	29,836	10.90	1,473.87	101.07
GR-BBC0[2]	390	209	53.59	49,673	27,521	14.90	1,672.49	350.25

[1] Time spent to find the best upper bound.
[2] BBC developed in Chapter 4 with (GR-BBC0) and without (BBC0) the GR strategy.

Note that the use of the valid inequalities improves the performance of the solution method, mainly in relation to the average lower bound and average gap. The average lower bound increases by 17.60%, from 17,427 in BBC1 to 20,494 in BBC2. The average gap is reduced from 28.73% to 25.03%, a reduction of 12.87%. The average cost of the solutions and the average computational time are not significantly affected by the addition of the valid inequalities, while

BBC2 proves optimality for two additional instances with respect to BBC1. BBC2 and BBC3 present a similar performance, with a slight improvement with the use of the SECs in the BBC3 algorithm. Differently from the traditional TSP and VRP problems, in which the SECs have yielded good results, the changes caused by the SECs in the solutions of the LP problems to eliminate the subtours do not appear to directly affect the objective function of such LP problems.

The use of the metaheuristic SA in the root node (HBBC1) to warm-start the BBC algorithm significantly improves the result of the BBC3 strategy, mainly with respect to the upper bound and the time spent to find the best solution. The average cost of the solutions is reduced from 110,291 to 42,376, a 61.58% reduction. The average time spent to find the best solution is 32 times smaller in HBBC1, being reduced from 1,891 to 59 seconds, a 96.87% reduction. The average lower bound and the average gap are also improved by the HBBC1 strategy. The average lower bound increases by 15.25%, from 20,499 to 23,625, while the average gap is reduced by 15.95%, from 24.99 to 21.01. Furthermore, HBBC1 proves optimality for 30 additional instances in relation to BBC3. In the HBBC2 algorithm, the use of the SA metaheuristic to derive cuts and tighten the linear relaxation improves the convergence of the method. The average gap decreases from 21.01% to 19.75%. Although the reduction in the gap appears insignificant, we can observe that HBBC2 proves the optimality for 14 additional instances in relation to the HBBC1 method. In fact, for some instance classes, the reduction in the average gap was up to 26%. Note also that HBBC2 increases the average lower bound of the solutions, while it reduces the average cost with respect to the solutions obtained with the HBBC1 approach.

Regarding the impact of varying the default configuration of CPLEX in strategies BBC3 and HBBC2, Table 5.5 shows that BBC3* obtained better lower bounds and gaps but worse upper bounds on average since the new configuration emphasizes the improvement of the lower bound rather than having a good-quality solution. For HBBC2*, the negative effect of prioritizing the lower bound is neutralized by the use of the metaheuristic to improve the upper bound. The average gap was reduced from 19.75 to 18.15, a reduction of 8%, with respect to HBBC2. The reduction was up to 21% for some instances. When compared with BBC0, HBBC2* reduces the average upper bound and gap by 20.97% and 43.05%, respectively, and increases the lower bound by 54.94%.

Finally, the graph reduction strategy significantly improved the results of the HBBC2* approach. The GR-HBBC2* obtained the optimal solutions for 61.28% of the instances and reduced the average gap to 10.9%, a reduction of 39.94% with respect to the HBBC2* strategy. Table 5.6 shows the average results of GR-HBBC2* compared with the results of the GR-BBC0 strategy. Table 5.6 also presents the ratio of the GR-HBBC2* solutions in relation to the GR-BBC0 solutions evaluated as $\frac{\text{Value in GR-BBC0}-\text{Value in GR-HBBC2*}}{\text{Value in GR-BBC0}}$. A ratio higher than zero indicates a reduction in the value of the GR-HBBC2* method, whereas a ratio smaller than zero indicates an increase in the value.

Evidently, GR-HBBC2* outperforms GR-BBC0. For example, the average reduction in the upper bound considering all the instances is 15.21%. For some instances, the reduction is up

Table 5.6: Comparison of the GR-HBBC2* algorithm with the BBC approach from the literature.

Solution method	Instance classes	#ins	#opt	%opt	Avg. upper bound	Avg. lower bound	Avg. gap (%)	Avg. time (sec.)	Avg. best time[1](sec.)
	1, 2, 3	60	48	80.00	31,232	28,237	3.29	867.81	142.58
	4, 5, 6	60	33	55.00	19,146	14,865	9.84	1,499.68	260.53
	7, 8, 9	60	28	46.67	32,171	23,223	14.16	1,904.82	33.98
	10, 11, 12	60	48	80.00	38,345	35,623	2.60	749.47	164.62
GR-BBC0[2]	13, 14, 15	60	24	40.00	52,266	33,991	16.71	2,132.32	126.54
	16, 17, 18	30	12	40.00	55,554	34,985	18.68	2,142.24	529.64
	19, 20, 21	30	10	33.33	70,623	18,801	37.39	2,410.55	1,024.54
	22, 23, 24	30	6	20.00	173,250	32,110	42.70	2,881.42	1,542.53
	All instances	390	209	53.59	49,673	27,521	14.77	1,672.49	350.25
	1, 2, 3	60	54	90.00	31,232	30,076	1.02	713.16	0.14
	4, 5, 6	60	42	70.00	19,006	15,667	5.27	1,086.71	2.20
	7, 8, 9	60	32	53.33	32,145	25,431	9.49	1,744.60	5.51
	10, 11, 12	60	49	81.67	38,345	36,122	2.10	737.94	2.88
GR-HBBC2*	13, 14, 15	60	31	51.67	51,963	36,521	12.72	1,746.30	74.48
	16, 17, 18	30	13	43.33	54,588	36,713	14.88	2,047.63	301.98
	19, 20, 21	30	12	40.00	48,757	19,810	31.94	2,173.21	290.58
	22, 23, 24	30	6	20.00	98,812	43,716	33.71	2,881.99	550.99
	All instances	390	239	61.28	42,118	29,836	10.90	1,473.87	101.07
	1, 2, 3		12.50	12.50	0.00	6.51	−69.11	−17.82	−99.90
	4, 5, 6		27.27	27.27	−0.73	5.40	−46.38	−27.54	−99.15
	7, 8, 9		14.29	14.29	−0.08	9.51	−33.02	−8.41	−83.77
	10, 11, 12		2.08	2.08	0.00	1.40	−19.19	−1.54	−98.25
Ratio	13, 14, 15		29.17	29.17	−0.58	7.44	−23.87	−18.10	−41.14
(%)	16, 17, 18		8.33	8.33	−1.74	4.94	−20.34	−4.42	−42.98
	19, 20, 21		20.00	20.00	−30.96	5.37	−14.59	−9.85	−71.64
	22, 23, 24		0.00	0.00	−42.97	36.15	−21.05	0.02	−64.28
	All instances		14.35	14.35	−15.21	8.41	−26.17	−11.88	−71.14

[1] Time spent to find the best upper bound.
[2] BBC proposed in Chapter 4 that also uses GR.

to 42.97%. Similarly, the improvement of the lower bound in some instance classes is up to 36.15%. The average gap was reduced by 26.17% with GR-HBBC2*, whereas the time spent to find the best solution was reduced by 71.14% on average, considering all the instances. To confirm whether the performances of GR-HBBC2* and GR-BBC0 are statistically different in terms of the upper bound, lower bound, gap and time, we carried out Friedman statistical tests (Conover, 1999) for the instance classes presented in Table 5.6. The null hypothesis is that there is no significant performance difference between GR-BBC0 and GR-HBBC2*. Table 5.7 gives the corresponding p values for the Friedman tests. Regarding the lower bound, gap and best time, we observe that the null hypothesis is rejected for every instance class at confidence levels ranging between 0.000 and 0.045. Therefore, the performances of GR-BBC0 and GR-HBBC2* are significantly different with respect to the lower bound, gap and best time in each one of the considered instance classes. For the upper bound and total computational time, although the differences are not statistically significant for some instance classes, the strategy GR-HBBC2* obtained always the same or better average results than GR-BBC0*.

5.6 Final remarks of the chapter

In this chapter, we proposed two metaheuristics; a branch-and-Benders-cut (BBC) algorithm; and a hybrid approach (HBBC), to solve the SCSRP. The metaheuristics are the first genetic algorithm and simulated annealing proposed for the SCSRP. They are based on the decomposi-

Table 5.7: The statistic values (p values) of the Friedman test for GR-BBC0 vs GR-HBBC2*.

Instance Class	Upper bound	Lower bound	Gap (%)	Time (sec.)	Best time[1](sec.)
1, 2, 3	**0.366**	0.007	0.039	**0.606**	0.000
4, 5, 6	**0.245**	0.020	0.005	0.039	0.000
7, 8, 9	**0.606**	0.001	0.000	**0.053**	0.001
10, 11, 12	**0.121**	0.039	0.039	**0.606**	0.000
13, 14, 15	**0.197**	0.007	0.007	**0.897**	0.007
16, 17, 18	**0.465**	0.045	0.028	**0.361**	0.005
19, 20, 21	0.028	0.018	0.003	**0.197**	0.001
22, 23, 24	0.003	0.000	0.000	**0.051**	0.001

[1] Time spent to find the best upper bound.
p values > 0.05 are highlighted in **bold**.

tion of the problem into smaller subproblems and the use of specialized algorithms to evaluate the candidate solutions. The BBC is based on an improved Benders reformulation of the problem and enhances previous approaches by using a different variable partitioning scheme. Valid inequalities for the problem have been proposed as well. The HBBC is an exact hybrid method that uses a metaheuristic to obtain good-quality solutions at early stages of the search tree as well as to improve the performance of solving the master problem by exploring the neighborhood of the incumbent solutions to generate more effective Benders cuts. The results of extensive computational experiments with 390 benchmark instances showed that both metaheuristics outperformed the only metaheuristic available in the literature for the SCSRP. For the BBC approach, the new variable partitioning scheme and the proposed valid inequalities were shown to be effective to increase the lower bound of the master problem. The computational results also provide evidence that the combination of the metaheuristic with the BBC, resulting in the hybrid algorithm HBBC, significantly reduces the cost of the solutions and the time spent to find good-quality solutions. The lower bound and gap of the solutions were also improved with the use of the metaheuristic within the BBC, especially when additional cuts from the neighborhood of the master problem solutions are generated.

We have observed that, while the BBC presented a solution with a higher cost when varying the parameters of the solver, the HBBC is able to take advantage of the new parameter configuration to improve the lower bound, the gap and the cost of the solutions. Basically, the HBBC counteracts the elevation of the cost in the BBC by exploring the neighborhood of the master problem solutions. By incorporating the graph reduction technique in the HBBC, we observed a significant reduction in the average gap, mainly because the graph reduction helps to increase the lower bound of the solutions. With the GR-HBBC, we effectively reduce the cost, gap and computational time of most of the instances with respect to the best exact approach proposed in the literature. In addition to their theoretical relevance, the improvements obtained with the proposed approaches may also have great value to aid decision making in practice. The reduction in the cost of the solutions directly impacts the time at which the demand nodes are accessible from the supply node, thus reducing the time that victims in the affected areas wait for supplies, evacuation, rescue and medical assistance. In the next chapter, we extend the CSRP to consider multiple crews, a relevant characteristic in practical settings.

Chapter 6

The multicrew scheduling and routing problem in road restoration

This chapter introduces the heterogeneous multicrew scheduling and routing problem in road restoration. The main contributions of the chapter include three novel mathematical formulations that differ in the way of modeling the scheduling decisions and the synchronization of the crews, and the development of valid inequalities based on some particular properties of the problem. This chapter is organized as follows. An introduction and motivation of the problem is presented in Section 6.1. Section 2.2 describes the MCSRP. Section 6.2 presents the MIP models, while Section 6.3 defines the properties and valid inequalities. Section 6.4 describes the instances and discusses the computational results. We close with concluding remarks in Section 6.5.

*A paper based on this chapter is under review in Transportation Research Part B: Methodological (TRB).
Moreno, Alfredo; Alem, Douglas; Gendreau, Michel; Munari, Pedro. The heterogeneous multicrew scheduling and routing problem in road restoration. Under review in Transportation Research Part B: Methodological (revised version submitted), 2020.

6.1 Introduction

The multicrew scheduling and routing problem (MCSRP) primarily focuses on the restoration of the critical subset of damaged nodes that are essential to emergency response operations. Multiple crews, associated with various agencies, such as civil defense, armed forces, and firefighters, are available to perform the repair operations. The crews must be assigned to repair the damaged nodes. Additionally, for each crew, the sequence in which the damaged nodes must be repaired and the route used to reach them and return to the depot must be determined.

The crews consist of workforce teams equipped with heavy machinery, dozers, excavators, light vehicles, etc., and they may not have the same equipment. For example, one crew may have dozers and excavators to remove heavy debris from a blocked road, while another may have only workers using shovels. Some crews may not have enough resources (machinery, workforce, etc.) to repair some damaged nodes. For instance, during the removal of downed trees and debris after a flood, there are potential hazards of electrocution from contact with downed power lines or tree limbs in contact with power lines (OSHA, 2019). Only crews with the appropriate knowledge and protective equipment against electrical hazards should remove such debris. Furthermore, a crew with heavy machinery may take more time to reach the damaged nodes than a crew with only light vehicles, although the former may perform a faster restoration with the help of heavy machinery. Consequently, the crews differ in the time required to repair the damaged nodes, in the travel time between nodes, and in the set of damaged nodes that they can repair. However, the consideration of multiple heterogeneous crews in the problem has been neglected in the literature because of the complexity involved in such consideration. In fact, as observed in the previous chapters, the basic variant with a single crew is already very challenging due to the scheduling and routing decisions that must be integrated. In the multicrew version of the problem, an additional complexity factor is the synchronization of the crews at the damaged nodes (Akbari and Salman, 2017a,b) because these nodes cannot be traversed unless they are completely repaired, and a crew may have to wait at some damaged nodes, while another crew performs the restoration of such nodes.

The contributions of this chapter are thus fourfold: (1) we define for the first time the MCSRP for road restoration and develop three mixed integer programming (MIP) models that differ in the way of modeling the scheduling decisions and the synchronization of the crews; (2) we study some particular properties of the problem and derive valid inequalities based on these properties; (3) we carry out computational experiments based on a real case and randomly generated instances to compare the performance of the proposed formulations and the effectiveness of the valid inequalities; (4) we apply our proposed approaches to the case of floods and landslides in Brazil, showing that the proposed approaches can provide useful suggestions to decision-makers.

6.2 Mathematical formulations

In this section, we present three mixed integer programming formulations for the MCSRP and two families of valid inequalities to strengthen them. The first and second formulations differ in the way of modeling the scheduling decisions and the synchronization of the crews. The third formulation eliminates symmetry related to the routing decisions by dropping certain variables and imposing new types of constraints.

6.2.1 First MCSRP formulation (MCSRP1)

The first MCSRP formulation is based on the three-index vehicle flow formulation of the vehicle routing problem (VRP) (Irnich et al., 2014) to define the schedule of the crews, while the synchronization of the crews is controlled with a four-index variable. The mathematical notation is as follows.

Sets

$\mathcal{V}$	All nodes.
$\mathcal{V}^{\mathrm{r}} \subset \mathcal{V}$	Damaged nodes.
$\mathcal{V}^{\mathrm{r}}_0 = \mathcal{V}^{\mathrm{r}} \cup \{0\}$	Damaged nodes including the source node 0 (depot).
$\mathcal{V}^{\mathrm{u}} \subset \mathcal{V}$	Undamaged nodes ($\mathcal{V}^{\mathrm{u}} = \mathcal{V}/\mathcal{V}^{\mathrm{r}}$).
$\mathcal{V}^{\mathrm{d}} \subset \mathcal{V}^{\mathrm{u}}$	Demand nodes.
$\mathcal{E}$	Arcs.
$\mathcal{E}_i \subseteq \mathcal{E}$	Arcs incident to node $i \in \mathcal{V}$.
$\mathcal{R} = \{1, \cdots, \lvert\mathcal{V}^{\mathrm{r}}\rvert\}$	Positions at which an already repaired damaged node can be visited in a path between two damaged nodes.
$\mathcal{K}$	Available crews.
$\mathcal{K}_i \subseteq \mathcal{K}$	Crews able to repair the damaged node $i \in \mathcal{V}^{\mathrm{r}}$.

Parameters

d_i	Demand of node $i \in \mathcal{V}^{\mathrm{d}}$.
δ_{ki}	Repair time of crew $k \in \mathcal{K}_i$ at node $i \in \mathcal{V}^{\mathrm{r}}$.
τ_{ke}	Travel time of crew $k \in \mathcal{K}$ on arc $e \in \mathcal{E}$.
ρ_{kij}	Shortest travel time of crew k between nodes $i \in \mathcal{V}$ and $j \in \mathcal{V}$ without using damaged nodes.
ℓ_e	Length of arc $e \in \mathcal{E}$.
l^{d}_i	Maximum distance allowed between node 0 and demand node $i \in \mathcal{V}^{\mathrm{d}}$.
M	Sufficiently large number.

Decision variables

W_i	Binary variable that assumes the value of 1 if and only if node $i \in \mathcal{V}^{\mathrm{r}}$ is repaired.
X_{kij}	Binary variable that assumes the value of 1 if and only if crew $k \in \mathcal{K}$ repairs node $j \in \mathcal{V}_0^{\mathrm{r}}$ immediately after node $i \in \mathcal{V}_0^{\mathrm{r}}$.
P_{eij}	Binary variable that assumes the value of 1 if and only if arc $e \in \mathcal{E}$ is used in the path from node $i \in \mathcal{V}_0^{\mathrm{r}}$ to node $j \in \mathcal{V}^{\mathrm{r}}$.
N_{lij}^{u}	Binary variable that assumes the value of 1 if and only if node $l \in \mathcal{V}^{\mathrm{u}}$ is used in the path from node $i \in \mathcal{V}_0^{\mathrm{r}}$ to node $j \in \mathcal{V}^{\mathrm{r}}$.
N_{lhij}^{r}	Binary variable that assumes the value of 1 if and only if node $l \in \mathcal{V}^{\mathrm{r}}$ is the hth damaged node visited in the path from node $i \in \mathcal{V}_0^{\mathrm{r}}$ to node $j \in \mathcal{V}^{\mathrm{r}}$.
Y_{ej}	Binary variable that assumes the value of 1 if and only if arc $e \in \mathcal{E}$ is used in the path from node 0 to node $j \in \mathcal{V}^{\mathrm{d}}$.
V_{lj}	Binary variable that assumes the value of 1 if and only if node $l \in \mathcal{V}$ is used in the path from node 0 to node $j \in \mathcal{V}^{\mathrm{d}}$.
T_{lhj}^{s}	Time at which the damaged node $l \in \mathcal{V}^{\mathrm{r}}$ in the position $h \in \mathcal{R}$ is visited in the path to node $j \in \mathcal{V}^{\mathrm{r}}$ (arrival time).
T_{lhj}^{w}	Waiting time at the damaged node $l \in \mathcal{V}^{\mathrm{r}}$ visited in the position $h \in \mathcal{R}$ in the path to node $j \in \mathcal{V}^{\mathrm{r}}$.
Z_i^{r}	Restoration time of damaged node $i \in \mathcal{V}_0^{\mathrm{r}}$.
Z_i^{d}	Accessibility time of demand node $i \in \mathcal{V}^{\mathrm{d}}$.

The parameter ρ_{kij} is computed by solving multiple shortest path problems over a graph in which the damaged nodes and arcs incident to the damaged nodes have been removed. In some cases, the removal of the damaged nodes can result in multiple unconnected graph components in the graph. In these cases, there are no paths between some pair of nodes $i - j$ without using at least one damaged node, and ρ_{kij} is assumed to be a sufficiently large number.

The variables X_{kij} define the schedule of the crews, while their route is defined by variables P_{eij}, N_{lij}^{u} and N_{lhij}^{r}, which determine the arcs and nodes to be visited in a crew path $i - j$. Variable N_{lhij}^{r} controls the position h of the damaged node l visited in such a path. The position is used to synchronize the arrival and departure of the crews at the damaged nodes. Since no synchronization is necessary for the undamaged nodes, the position at which a node $l \in \mathcal{V}^{\mathrm{u}}$ is visited by the crews is not relevant. Variables P_{eij}, N_{lij}^{d} and N_{lhij}^{r} are not defined for $j = 0$ since we assume that the crews return to the depot by the same paths by which they arrived at the last damaged node on their schedule. Finally, variables Y_{ej} and V_{lj} define the arcs and nodes, respectively, to be visited in relief path $0 - j$. The MIP model is formulated as follows.

Objective function. The objective function (6.1) consists of minimizing the weighted sum of the accessibility time.

$$\min \sum_{i \in \mathcal{V}^{\mathrm{d}}} d_i \cdot Z_i^{\mathrm{d}}. \tag{6.1}$$

Accessibility time evaluation. The accessibility time is defined by constraints (6.2). A demand node i is accessible if there exists a relief path $0-i$ using undamaged and/or repaired nodes. Thus, the accessibility time Z_i^{d} associated with demand node $i \in \mathcal{V}^{\mathrm{d}}$ depends on the time Z_j^{r} when damaged nodes $j \in \mathcal{V}^{\mathrm{r}}$ in relief path $0-i$ are repaired.

$$Z_i^{\mathrm{d}} \geq Z_j^{\mathrm{r}} - M \cdot (1 - V_{ji}), \ \forall \ i \in \mathcal{V}^{\mathrm{d}}, \ j \in \mathcal{V}^{\mathrm{r}}. \tag{6.2}$$

Restoration time constraints. Constraints (6.3) define the restoration time when no damaged nodes are visited in crew path $i-j$ or when there is no waiting time associated with the visited damaged nodes. In this case, for a given node j repaired by crew k, the restoration time Z_j^{r} is the sum of three components: the restoration time of the predecessor node i (Z_i^{r}); the travel time in the path $i-j$ ($\sum_{e\in\mathcal{E}} \tau_{ke} \cdot P_{eij}$); and the repair time of node j (δ_{kj}). These constraints also prevent subtours. Constraints (6.4) define the restoration time when there is waiting time associated with the damaged nodes visited in a given path $i-j$. In this case, for a given node j repaired by crew k, the restoration time Z_j^{r} is the sum of the next components: the time when the crew departs from the last damaged node l visited in the path ($T_{lhj}^{\mathrm{w}} + T_{lhj}^{\mathrm{s}}$); the shortest travel time from node l to node j without using damaged nodes ($\sum_{i\in\mathcal{V}_0^{\mathrm{r}}} \rho_{klj} \cdot N_{lhij}^{\mathrm{r}}$); and the repair time of node j (δ_{kj}). Constraints (6.4) are activated only for the last occupied position h, i.e., when there is no node visited in the position $h+1$ ($\sum_{i\in\mathcal{V}_0^{\mathrm{r}}} \sum_{l\in\mathcal{V}^r} N_{l(h+1)ij}^{\mathrm{r}} = 0$).

$$Z_j^{\mathrm{r}} \geq Z_i^{\mathrm{r}} + \sum_{e\in\mathcal{E}} \tau_{ke} \cdot P_{eij} + \delta_{kj} - M \cdot (1 - X_{kij}), \ \forall \ i \in \mathcal{V}_0^{\mathrm{r}}, \ j \in \mathcal{V}^{\mathrm{r}}, \ k \in \mathcal{K}, \tag{6.3}$$

$$Z_j^{\mathrm{r}} \geq \sum_{l\in\mathcal{V}^{\mathrm{r}}} (T_{lhj}^{\mathrm{w}} + T_{lhj}^{\mathrm{s}} + \sum_{i\in\mathcal{V}_0^{\mathrm{r}}} \rho_{klj} \cdot N_{lhij}^{\mathrm{r}}) + \delta_{kj} - M \cdot (1 - \sum_{i\in\mathcal{V}_0^{\mathrm{r}}} X_{kij} + \sum_{i\in\mathcal{V}_0^r} \sum_{l\in\mathcal{V}^{\mathrm{r}}} N_{l(h+1)ij}^{\mathrm{r}}),$$
$$\forall \ j \in \mathcal{V}^{\mathrm{r}}, \ h \in \mathcal{R} \setminus \{|\mathcal{R}|\}, \ k \in \mathcal{K}. \tag{6.4}$$

Relief path constraints. For a given relief path $0-i$, constraints (6.5) force the use of an arc incident to node 0, while constraints (6.6) force the use of an arc incident to node i. Furthermore, for each node l in the middle of this path ($V_{li} = 1$), there must be one arc leaving and one arc arriving at node l, as imposed by constraints (6.7). Constraints (6.8) prohibit the use of relief paths whose distance between the depot and demand nodes is greater than the maximum distance allowed.

$$\sum_{e\in\mathcal{E}_0} Y_{ei} = 1, \ \forall \ i \in \mathcal{V}^{\mathrm{d}}, \tag{6.5}$$

$$\sum_{e\in\mathcal{E}_i} Y_{ei} = 1, \ \forall \ i \in \mathcal{V}^{\mathrm{d}}, \tag{6.6}$$

$$\sum_{e\in\mathcal{E}_l} Y_{ei} = 2V_{li}, \ \forall \ j \in \mathcal{V}^{\mathrm{d}}, \ l \in \mathcal{V} \setminus \{0, \ i\}, \tag{6.7}$$

$$\sum_{e\in\mathcal{E}} Y_{ei} \cdot \ell_e \leq l_i^{\mathrm{d}}, \ \forall \ i \in \mathcal{V}^{\mathrm{d}}. \tag{6.8}$$

Crew routing constraints. If there is a crew path $i-j$ ($\sum_{k\in\mathcal{K}} X_{kij} = 1$), constraints (6.9) force the use of an arc incident to node i in this path, while constraints (6.10) force the use of an arc incident to node j. Given a node l in crew path $i-j$, constraints (6.11) and (6.12) ensure that path $i-j$ contains one arc leaving node l and one arc arriving at node l. Constraints (6.11) are associated with undamaged nodes $l \in \mathcal{V}^{\mathrm{u}}$, while constraints (6.12) are associated with damaged nodes $l \in \mathcal{V}^{\mathrm{r}}$.

$$\sum_{e \in \mathcal{E}_i} P_{eij} = \sum_{k \in \mathcal{K}} X_{kij},\ \forall\ i \in \mathcal{V}_0^{\mathrm{r}},\ j \in \mathcal{V}^{\mathrm{r}}, \tag{6.9}$$

$$\sum_{e \in \mathcal{E}_j} P_{eij} = \sum_{k \in \mathcal{K}} X_{kij},\ \forall\ i \in \mathcal{V}_0^{\mathrm{r}},\ j \in \mathcal{V}^{\mathrm{r}}, \tag{6.10}$$

$$\sum_{e \in \mathcal{E}_l} P_{eij} = 2N^{\mathrm{u}}_{lij},\ \forall\ i \in \mathcal{V}_0^{\mathrm{r}},\ j \in \mathcal{V}^{\mathrm{r}},\ l \in \mathcal{V}^{\mathrm{u}} : l \neq i, \tag{6.11}$$

$$\sum_{e \in \mathcal{E}_l} P_{eij} = 2\sum_{h \in \mathcal{R}} N^{\mathrm{r}}_{lhij},\ \forall\ i \in \mathcal{V}_0^{\mathrm{r}},\ j \in \mathcal{V}^{\mathrm{r}},\ l \in \mathcal{V}^{\mathrm{r}} \setminus \{i,\ j\}. \tag{6.12}$$

Crew scheduling constraints. If node j is repaired ($W_j = 1$), constraints (6.13) state that there will be exactly one crew $k \in \mathcal{K}_j$ designated to repair this node. Constraints (6.14) represent the flow conservation. Constraints (6.15) establish that each crew k must perform at most one schedule.

$$\sum_{k \in \mathcal{K}_j} \sum_{\substack{i \in \mathcal{V}_0^{\mathrm{r}}: \\ i \neq l}} X_{kij} = W_j,\ \forall\, j \in \mathcal{V}^{\mathrm{r}}, \tag{6.13}$$

$$\sum_{\substack{i \in \mathcal{V}_0^{\mathrm{r}}: \\ i \neq l}} X_{kil} - \sum_{\substack{j \in \mathcal{V}_0^{\mathrm{r}}: \\ j \neq l}} X_{klj} = 0,\ \forall\ l \in \mathcal{V}_0^{\mathrm{r}},\ k \in \mathcal{K}, \tag{6.14}$$

$$\sum_{j \in \mathcal{V}^{\mathrm{r}}} X_{k0j} \leq 1,\ \forall\, k \in \mathcal{K}. \tag{6.15}$$

Assignment constraints. Constraints (6.16) and (6.17) state that node $l \in \mathcal{V}^{\mathrm{r}}$ must be repaired if it is used in either a relief path ($\sum_{i \in \mathcal{V}^{\mathrm{d}}} V_{li} > 1$) or a crew path ($\sum_{h \in \mathcal{R}} \sum_{i \in \mathcal{V}_0^{\mathrm{r}}} \sum_{j \in \mathcal{V}^{\mathrm{r}}} N^{\mathrm{r}}_{lhij} > 1$).

$$|\mathcal{V}^{\mathrm{d}}| \cdot W_l \geq \sum_{i \in \mathcal{V}^{\mathrm{d}}} V_{li},\ \forall\, l \in \mathcal{V}^{\mathrm{r}}, \tag{6.16}$$

$$|\mathcal{V}^{\mathrm{r}}| \cdot W_l \geq \sum_{h \in \mathcal{R}} \sum_{i \in \mathcal{V}_0^{\mathrm{r}}} \sum_{j \in \mathcal{V}^{\mathrm{r}}} N^{\mathrm{r}}_{lhij},\ \forall\, l \in \mathcal{V}^{\mathrm{r}}. \tag{6.17}$$

Synchronization constraints. Constraints (6.18)-(6.23) synchronize the arrival of crew k in damaged node l visited in crew path $i-j$. Here, i and j are damaged nodes repaired by crew k, while l is a damaged node used in path $i-j$. Thus, when crew k arrives at this node l, it either waits for node l to be repaired by another crew if this node is still damaged, or it can cross without waiting if l has already been repaired. Constraints (6.18) guarantee that a damaged node l cannot be visited more than once in the path to node j. Since all the travel times on the arcs are nonnegative values, the optimal crew path $i-j$ does not need to consider a node l

more than once. Constraints (6.19) establish that a given crew cannot visit different damaged nodes simultaneously, i.e., a position h can be occupied for at most one damaged node l in the path to node j. Constraints (6.20) ensure that damaged nodes in path $i-j$ must be visited in consecutive positions. Then, a damaged node cannot be visited in a position h if no damaged node was already visited in position $h-1$. Constraints (6.21) evaluate the arrival time of crew k at the first damaged node l visited in path $i-j$. Similarly, constraints (6.22) define the arrival time of crew k at the damaged node l visited in position $h>1$ based on the departure time of node v visited in position $h-1$. Finally, constraints (6.23) compute the waiting time of the crew in damaged node l visited in position h. The waiting time is calculated as the difference between the restoration time of the damaged node l and the arrival time of crew k at damaged node l.

$$\sum_{h\in\mathcal{R}}\sum_{i\in\mathcal{V}_0^{\mathrm{r}}} N^{\mathrm{r}}_{lhij} \leq 1,\ \forall\ l\in\mathcal{V}^{\mathrm{r}},\ j\in\mathcal{V}^{\mathrm{r}}, \tag{6.18}$$

$$\sum_{l\in\mathcal{V}^{\mathrm{r}}}\sum_{i\in\mathcal{V}_0^{\mathrm{r}}} N^{\mathrm{r}}_{lhij} \leq 1,\ \forall\ h\in\mathcal{R},\ j\in\mathcal{V}^{\mathrm{r}}, \tag{6.19}$$

$$\sum_{l\in\mathcal{V}^{\mathrm{r}}} N^{\mathrm{r}}_{lhij} \leq \sum_{l\in\mathcal{V}^{\mathrm{r}}} N^{\mathrm{r}}_{l(h-1)ij},\ \forall\ i\in\mathcal{V}_0^{\mathrm{r}},\ j\in\mathcal{V}^{\mathrm{r}},\ h\in\mathcal{R}\setminus\{1\}, \tag{6.20}$$

$$T^{\mathrm{s}}_{l1j} \geq Z^{\mathrm{r}}_i + \sum_{k\in\mathcal{K}} \rho_{kil}\cdot X_{kij} - M\cdot(1-N^{\mathrm{r}}_{l1ij}),\ \forall\ i\in\mathcal{V}_0^{\mathrm{r}},\ j\in\mathcal{V}^{\mathrm{r}},\ l\in\mathcal{V}^{\mathrm{r}}, \tag{6.21}$$

$$T^{\mathrm{s}}_{lhj} \geq \sum_{v\in\mathcal{V}^{\mathrm{r}}}(T^{\mathrm{w}}_{v(h-1)j} + T^{\mathrm{s}}_{v(h-1)j} + \sum_{i\in\mathcal{V}_0^{\mathrm{r}}} N^{\mathrm{r}}_{v(h-1)ij}\cdot\rho_{kvl}) - M\cdot(2-\sum_{i\in\mathcal{V}_0^{\mathrm{r}}}(N^{\mathrm{r}}_{lhij}+X_{kij})),$$
$$\forall\ k\in\mathcal{K}, l\in\mathcal{V}^{\mathrm{r}}, j\in\mathcal{V}^{\mathrm{r}},\ h\in\mathcal{R}\setminus\{1\}, \tag{6.22}$$

$$T^{\mathrm{w}}_{lhj} \geq Z^{\mathrm{r}}_l - T^{\mathrm{s}}_{lhj} - M\cdot(1-\sum_{i\in\mathcal{V}_0^{\mathrm{r}}} N^{\mathrm{r}}_{lhij}),\ \forall\ l\in\mathcal{V}^{\mathrm{r}},\ j\in\mathcal{V}^{\mathrm{r}},\ h\in\mathcal{R}. \tag{6.23}$$

Domain of the decision variables. Constraints (6.24)-(6.31) impose the domain of the decision variables. It is worth mentioning that variables P_{eij} and Y_{ej} do not need to be defined as binary variables in the computational implementation because they naturally assume binary values if variables N^{u}_{lij}, N^{r}_{lhij} and V_{kj} are binaries.

$$X_{kij}, W_j \in \{0,\ 1\},\ \forall\ i\in\mathcal{V}_0^{\mathrm{r}},\ j\in\mathcal{V}_0^{\mathrm{r}},\ k\in\mathcal{K}, \tag{6.24}$$

$$N^{\mathrm{r}}_{lhij} \in \{0,\ 1\},\ \forall\ i\in\mathcal{V}_0^{\mathrm{r}},\ j\in\mathcal{V}^{\mathrm{r}},\ l\in\mathcal{V}^{\mathrm{r}},\ h\in\mathcal{R}, \tag{6.25}$$

$$N^{\mathrm{u}}_{lij} \in \{0,\ 1\},\ \forall\ i\in\mathcal{V}_0^{\mathrm{r}},\ j\in\mathcal{V}^{\mathrm{r}},\ l\in\mathcal{V}^{\mathrm{u}}, \tag{6.26}$$

$$P_{eij} \geq 0,\ \forall\ i\in\mathcal{V}_0^{\mathrm{r}},\ j\in\mathcal{V}^{\mathrm{r}},\ e\in\mathcal{E}, \tag{6.27}$$

$$V_{li} \in \{0,\ 1\},\ \forall\ i\in\mathcal{V}^{\mathrm{d}},\ l\in\mathcal{V}, \tag{6.28}$$

$$Y_{el} \geq 0,\ \forall\ l\in\mathcal{V}^{\mathrm{d}},\ e\in\mathcal{E}, \tag{6.29}$$

$$T^{\mathrm{s}}_{lhj}, T^{\mathrm{w}}_{lhj} \geq 0,\ \forall\ l\in\mathcal{V}^{\mathrm{r}}, j\in\mathcal{V}^{\mathrm{r}},\ h\in\mathcal{R}, \tag{6.30}$$

$$Z^{\mathrm{r}}_i, Z^{\mathrm{d}}_j \geq 0,\ \forall\ i\in\mathcal{V}_0^{\mathrm{r}},\ j\in\mathcal{V}^{\mathrm{d}}. \tag{6.31}$$

6.2.2 Second MCSRP formulation (MCSRP2)

The second formulation for the MCSRP is based on the two-index vehicle flow formulation of the VRP to define the crew scheduling, while the synchronization of the crews is controlled with a new three-index variable. For this formulation, consider the following notation:

Decision variables

W'_{ik} Binary variable that assumes the value of 1 if and only if node $i \in \mathcal{V}^{\mathrm{r}}$ is repaired by crew k.

X'_{ij} Binary variable that assumes the value of 1 if and only if node $j \in \mathcal{V}_0^{\mathrm{r}}$ is repaired immediately after node $i \in \mathcal{V}_0^{\mathrm{r}}$.

N_{lij} Binary variable that assumes the value of 1 if and only if node $l \in \mathcal{V}$ is visited in the path from node $i \in \mathcal{V}_0^{\mathrm{r}}$ to node $j \in \mathcal{V}^{\mathrm{r}}$.

R_{lhj} Binary variable that assumes the value of 1 if and only if node $l \in \mathcal{V}^{\mathrm{r}}$ is the hth damaged node visited in the path to node $j \in \mathcal{V}^{\mathrm{r}}$.

The two-index variable X'_{ij} defines the restoration order of the damaged nodes, independent of the crew that performs this activity. The assignment of the crews to the damaged nodes is achieved with variable W'_{ik}. Furthermore, the position of the damaged nodes in the crew paths is controlled with the new variable R_{lhj}. The objective function (6.1), the accessibility time evaluation (6.2), and the relief paths constraints (6.5)-(6.8) are the same as in MCSRP1. The other group of constraints is modified as follows.

Restoration time and crew routing and scheduling constraints. In constraints (6.32)-(6.38), we use variables X'_{ij} and/or W'_{kj} instead of X_{kij} and/or W_j. Additionally, in constraints (6.33), variable R_{lhj} is used instead of N^{r}_{lhij}, and in constraints (6.39), variable N_{lij} is used instead of N^{r}_{lhij}. Constraints (6.39) control the arcs incident to any node $l \in \mathcal{V}$ in a given path $i-j$ instead of constraints (6.11) and (6.12) that are associated with undamaged nodes $l \in \mathcal{V}^{\mathrm{u}}$ and damaged nodes $l \in \mathcal{V}^{\mathrm{r}}$ separately. Furthermore, unlike MCSRP1, the flow conservation constraints (6.34) are not defined for each crew.

$$Z^{\mathrm{r}}_j \geq Z^{\mathrm{r}}_i + \sum_{e \in \mathcal{E}} \tau_{ke} \cdot P_{eij} + \delta_{kj} - M \cdot (2 - X'_{ij} - W'_{kj}),\ \forall\ i \in \mathcal{V}^{\mathrm{r}},\ j \in \mathcal{V}^{\mathrm{r}},\ k \in \mathcal{K}, \tag{6.32}$$

$$Z^{\mathrm{r}}_j \geq \sum_{l \in \mathcal{V}^{\mathrm{r}}} (T^{\mathrm{w}}_{lhj} + T^{\mathrm{s}}_{lhj} + \rho_{klj} \cdot R_{lhj}) + \delta_{kj} - M \cdot (1 - W'_{kj} + \sum_{l \in \mathcal{V}^{\mathrm{r}}} R_{(h+1)lj}),$$
$$\forall\ j \in \mathcal{V}^{\mathrm{r}},\ h \in \mathcal{R} \setminus \{|\mathcal{R}|\},\ k \in \mathcal{K}, \tag{6.33}$$

$$\sum_{\substack{i \in \mathcal{V}_0^{\mathrm{r}}: \\ i \neq j}} X'_{ij} = \sum_{k \in \mathcal{K}} W'_{kj},\ \forall\ j \in \mathcal{V}^{\mathrm{r}}, \tag{6.34}$$

$$\sum_{\substack{i \in \mathcal{V}_0^{\mathrm{r}}: \\ i \neq l}} X'_{il} - \sum_{\substack{j \in \mathcal{V}_0^{\mathrm{r}}: \\ j \neq l}} X'_{lj} = 0,\ \forall\ l \in \mathcal{V}_0^{\mathrm{r}}, \tag{6.35}$$

$$\sum_{j \in \mathcal{V}^{\mathrm{r}}} X'_{0j} \leq |\mathcal{K}|, \tag{6.36}$$

$$\sum_{e\in\mathcal{E}_i} P_{eij} = X'_{ij},\ \forall\ i \in \mathcal{V}_0^{\mathrm{r}},\ j \in \mathcal{V}^{\mathrm{r}}, \tag{6.37}$$

$$\sum_{e\in\mathcal{E}_j} P_{eij} = X'_{ij},\ \forall\ i \in \mathcal{V}_0^{\mathrm{r}},\ j \in \mathcal{V}^{\mathrm{r}}, \tag{6.38}$$

$$\sum_{e\in\mathcal{E}_l} P_{eij} = 2N_{lij},\ \forall\ i \in \mathcal{V}_0^{\mathrm{r}},\ j \in \mathcal{V}^{\mathrm{r}},\ l \in \mathcal{V} \setminus \{i,\ j\}. \tag{6.39}$$

Assignment constraints. Constraints (6.40) and (6.41) define which nodes must be repaired. Additionally, constraints (6.42) are introduced to guarantee that if nodes i and j are considered in the same schedule ($X'_{ij} = 1$), both are repaired by the same crew. Constraints (6.43) force the consideration of different crews for different schedules. If $X_{0i} = 1$ and $X_{0j} = 1$, i and j are the first nodes of two different schedules. Thus, if crew k repairs node i ($W'_{ki} = 1$), a different crew k' must repair node j, i.e., $\sum_{\substack{k'\in\mathcal{K}:\\ k'\neq k}} W_{k'j} \geq 1$.

$$|\mathcal{V}^{\mathrm{r}}| \cdot \sum_{k\in\mathcal{K}_l} W'_{kl} \geq \sum_{i\in\mathcal{V}_0^{\mathrm{r}}} \sum_{j\in\mathcal{V}^{\mathrm{r}}} N_{lij},\ \forall\ l \in \mathcal{V}^{\mathrm{r}}, \tag{6.40}$$

$$|\mathcal{V}^{\mathrm{d}}| \cdot \sum_{k\in\mathcal{K}_l} W'_{kl} \geq \sum_{i\in\mathcal{V}^{\mathrm{d}}} V_{li},\ \forall\ l \in \mathcal{V}^{\mathrm{r}}, \tag{6.41}$$

$$W'_{kj} \geq W'_{ki} + X'_{ij} - 1,\ \forall\ i \in \mathcal{V}^{\mathrm{r}},\ j \in \mathcal{V}^{\mathrm{r}},\ k \in \mathcal{K}, \tag{6.42}$$

$$\sum_{\substack{k'\in\mathcal{K}:\\ k'\neq k}} W_{k'j} \geq W_{ki} + X_{0i} + X_{0j} - 2,\ \forall\ i \in \mathcal{V}^{\mathrm{r}},\ j \in \mathcal{V}^{\mathrm{r}} : i \neq j,\ k \in \mathcal{K}. \tag{6.43}$$

Synchronization constraints. The new set of constraints (6.44) is introduced to enforce the allocation of damaged nodes l considered in path $i - j$ ($N_{lij} = 1$) to some position h defined by variable R_{lhj}. Constraints (6.45)-(6.47) define the position of a damaged node l in the path to node j. Constraints (6.48)-(6.50) define the arrival and waiting time of the crews at the damaged node l visited in the path from node i to node j.

$$\sum_{i\in\mathcal{V}_0^{\mathrm{r}}} N_{lij} = \sum_{h\in\mathcal{R}} R_{lhj},\ \forall\ l \in \mathcal{V}^{\mathrm{r}},\ j \in \mathcal{V}^{\mathrm{r}}, \tag{6.44}$$

$$\sum_{h\in\mathcal{R}} R_{lhj} \leq 1,\ \forall\ j \in \mathcal{V}^{\mathrm{r}},\ l \in \mathcal{V}^{\mathrm{r}}, \tag{6.45}$$

$$\sum_{l\in\mathcal{V}^{\mathrm{r}}} R_{lhj} \leq 1,\ \forall\ j \in \mathcal{V}^{\mathrm{r}},\ h \in \mathcal{R}, \tag{6.46}$$

$$\sum_{l\in\mathcal{V}^{\mathrm{r}}} R_{lhj} \leq \sum_{l\in\mathcal{V}^{\mathrm{r}}} R_{(h-1)lj},\ \forall\ j \in \mathcal{V}^{\mathrm{r}},\ h \in \mathcal{R} \setminus \{1\}, \tag{6.47}$$

$$T^{\mathrm{s}}_{1lj} \geq Z^{\mathrm{r}}_i + \sum_{k\in\mathcal{K}} W'_{kj} \cdot \rho_{kil} - (2 - X'_{ij} - R_{1lj}) \cdot M,\ \forall\ i \in \mathcal{V}_0^{\mathrm{r}},\ j \in \mathcal{V}^{\mathrm{r}},\ l \in \mathcal{V}^{\mathrm{r}}, \tag{6.48}$$

$$T^{\mathrm{s}}_{lhj} \geq \sum_{p\in\mathcal{V}^{\mathrm{r}}} (T^{\mathrm{w}}_{(h-1)pj} + T^{\mathrm{s}}_{(h-1)pj} + R_{(h-1)pj} \cdot \rho_{kpl}) - (2 - R_{lhj} - W'_{kj}) \cdot M,$$

$$\forall\ k \in \mathcal{K}, l \in \mathcal{V}^{\mathrm{r}}, j \in \mathcal{V}^{\mathrm{r}},\ h \in \mathcal{R} \setminus \{1\}, \tag{6.49}$$

$$T^{\mathrm{w}}_{lhj} \geq Z^{\mathrm{r}}_l - T^{\mathrm{s}}_{lhj} - M \cdot (1 - R_{lhj}),\ \forall\ l \in \mathcal{V}^{\mathrm{r}},\ j \in \mathcal{V}^{\mathrm{r}},\ h \in \mathcal{R}. \tag{6.50}$$

Domain of the decision variables. Constraints (6.51) and (6.53) impose the domain of the decision variables.

$$X'_{ij}, W'_{kj} \in \{0,\, 1\},\, \forall\, i \in \mathcal{V}_0^{\mathtt{r}},\; j \in \mathcal{V}_0^{\mathtt{r}},\; k \in \mathcal{K}, \tag{6.51}$$

$$R_{hij} \in \{0,\, 1\},\, \forall\, i \in \mathcal{V}^{\mathtt{r}},\; j \in \mathcal{V}^{\mathtt{r}},\; h \in \mathcal{R}, \tag{6.52}$$

$$N_{lij} \in \{0,\, 1\},\, \forall\, i \in \mathcal{V}_0^{\mathtt{r}},\; j \in \mathcal{V}^{\mathtt{r}},\; l \in \mathcal{V}. \tag{6.53}$$

6.2.3 Third MCSRP formulation (MCSRP3)

The third formulation is a modified version of MCSRP2 with the elimination of some variables related to the routing decisions. Furthermore, some additional constraints are introduced to prohibit the restoration of the damaged nodes that do not affect the accessibility of the demand nodes. For instance, consider the schedule (0, 1, 2, 0) for one crew and assume that damaged node 2 is not considered either in the relief paths or in the crew paths. Then, the restoration time of node 2 does not affect the accessibility of the demand nodes. In this case, several solutions considering different crew paths from node 1 to node 2 have the same cost as the solution considering the schedule (0, 1, 0). We say that such solutions are symmetric and can be eliminated by prohibiting the restoration of unnecessary damaged nodes. For the MCSRP3, consider the additional notation as follows.

Decision variables

P'_{eij} Binary variable that assumes the value of 1 if and only if arc $e \in \mathcal{E}$ is used either in the path from node $i \in \mathcal{V}_0^{\mathtt{r}}$ to node $j \in \mathcal{V}^{\mathtt{r}}$ or from node $j \in \mathcal{V}^{\mathtt{r}}$ to node $i \in \mathcal{V}^{\mathtt{r}}$ with $i < j$.

N'_{lij} Binary variable that assumes the value of 1 if and only if node $l \in \mathcal{V}^{\mathtt{r}}$ is used either in the path from node $i \in \mathcal{V}_0^{\mathtt{r}}$ to node $j \in \mathcal{V}^{\mathtt{r}}$ or from node $j \in \mathcal{V}^{\mathtt{r}}$ to node $i \in \mathcal{V}^{\mathtt{r}}$ with $i < j$.

Variables P'_{eij} and N'_{lij} are defined only for path $i - j$, where $i < j$. The objective function (6.1), the accessibility time evaluation (6.2), the relief paths constraints (6.5)-(6.8), and the scheduling constraints (6.34)-(6.36) are the same as in MCSRP2. The other constraints are posed as follows.

Restoration time constraints. Constraints (6.54) and (6.55) define the restoration time at the damaged nodes when $i < j$ and $i > j$, respectively. Constraints (6.33) are also included in MCSRP3.

$$Z_j^{\mathtt{r}} \geq Z_i^{\mathtt{r}} + \sum_{e \in \mathcal{E}} \tau_{ke} \cdot P'_{eij} + \delta_{kj} - (2 - X'_{ij} - W'_{kj}) \cdot M,\, \forall\, k \in \mathcal{K},\, i \in \mathcal{V}_0^{\mathtt{r}},\; j \in \mathcal{V}^{\mathtt{r}} : i < j, \tag{6.54}$$

$$Z_j^{\mathtt{r}} \geq Z_i^{\mathtt{r}} + \sum_{e \in \mathcal{E}} \tau_{ke} \cdot P'_{eji} + \delta_{kj} - (2 - X'_{ij} - W'_{kj}) \cdot M,\, \forall\, k \in \mathcal{K},\, i \in \mathcal{V}^{\mathtt{r}},\; j \in \mathcal{V}^{\mathtt{r}} : i > j. \tag{6.55}$$

Crew routing constraints. Constraints (6.56)-(6.60) define the paths of the crews for $i < j$ only.

$$\sum_{e \in \mathcal{E}_0} P'_{e0j} = X'_{0j},\, \forall\, j \in \mathcal{V}^{\mathtt{r}}, \tag{6.56}$$

$$\sum_{e \in \mathcal{E}_j} P'_{e0j} = X'_{0j}, \; \forall \; j \in \mathcal{V}^{\mathrm{r}}, \tag{6.57}$$

$$\sum_{e \in \mathcal{E}_i} P'_{eij} = X'_{ij} + X'_{ji}, \; \forall \; i \in \mathcal{V}^{\mathrm{r}}, \; j \in \mathcal{V}^{\mathrm{r}} : i < j, \tag{6.58}$$

$$\sum_{e \in \mathcal{E}_j} P'_{eij} = X'_{ij} + X'_{ji}, \; \forall \; i \in \mathcal{V}^{\mathrm{r}}, \; j \in \mathcal{V}^{\mathrm{r}} : i < j, \tag{6.59}$$

$$\sum_{e \in \mathcal{E}_l} P'_{eij} = 2N'_{lij}, \; \forall \; i \in \mathcal{V}^{\mathrm{r}}_0, \; j \in \mathcal{V}^{\mathrm{r}}, \; l \in \mathcal{V} \setminus \{i, \; j\} : i < j. \tag{6.60}$$

Assignment constraints. Constraints (6.61) replace constraints (6.40) to force the restoration of nodes used in crew paths $i - j$ with $i < j$. Constraints (6.62) are introduced to prohibit the restoration of some unnecessary damaged nodes. Thus, the last damaged node l repaired by a crew ($X'_{l0} = 1$) must be used either in a relief path ($\sum_{i \in \mathcal{V}^{\mathrm{d}}} V_{li} > 1$) or in a crew path ($\sum_{i \in \mathcal{V}^{\mathrm{r}}_0} \sum_{\substack{j \in \mathcal{V}^{\mathrm{r}}: \\ i<j}} N'_{lij} > 1$). Constraints (6.41)-(6.43) are also included in MCSRP3.

$$|\mathcal{V}^{\mathrm{r}}| \cdot \sum_{k \in \mathcal{K}_l} W'_{kl} \geq \sum_{i \in \mathcal{V}^{\mathrm{r}}_0} \sum_{\substack{j \in \mathcal{V}^{\mathrm{r}}: \\ i<j}} N'_{lij}, \; \forall \; l \in \mathcal{V}^{\mathrm{r}}, \tag{6.61}$$

$$\sum_{k \in \mathcal{K}_l} W'_{kl} \leq \sum_{i \in \mathcal{V}^{\mathrm{d}}} V_{li} + \sum_{i \in \mathcal{V}^{\mathrm{r}}_0} \sum_{\substack{j \in \mathcal{V}^{\mathrm{r}}: \\ i<j}} N'_{lij} - X'_{l0} + 1, \; \forall \; l \in \mathcal{V}^{\mathrm{r}}. \tag{6.62}$$

Synchronization constraints. Constraints (6.63)-(6.65) enforce the allocation of damaged nodes l considered in crew path $i - j$ to some position h defined by variable R_{lhj}. Constraints (6.45)-(6.50) are also included in MCSRP3.

$$\sum_{h \in \mathcal{R}} R_{lhj} \geq N'_{lij} + X'_{ij} - 1, \; \forall \; i \in \mathcal{V}^{\mathrm{r}}, \; l \in \mathcal{V}^{\mathrm{r}}, \; j \in \mathcal{V}^{\mathrm{r}} : i < j, \tag{6.63}$$

$$\sum_{h \in \mathcal{R}} R_{lhi} \geq N'_{lij} + X'_{ji} - 1, \; \forall \; i \in \mathcal{V}^{\mathrm{r}}, \; l \in \mathcal{V}^{\mathrm{r}}, \; j \in \mathcal{V}^{\mathrm{r}} : i < j, \tag{6.64}$$

$$\sum_{h \in \mathcal{R}} \sum_{j \in \mathcal{V}^{\mathrm{r}}} R_{lhj} = \sum_{i \in \mathcal{V}^{\mathrm{r}}_0} \sum_{j \in \mathcal{V}^{\mathrm{r}} : i<j} N'_{lij}, \; \forall \; l \in \mathcal{V}^{\mathrm{r}}. \tag{6.65}$$

Domain of the decision variables. Finally, constraints (6.66) and (6.67) impose the domain of the decision variables introduced in MCSRP3.

$$N'_{lij} \in \{0, \; 1\}, \; \forall \; l \in \mathcal{V}, \; i \in \mathcal{V}^{\mathrm{r}}_0, \; j \in \mathcal{V}^{\mathrm{r}} : i < j, \tag{6.66}$$

$$P'_{eij} \geq 0, \; \forall \; e \in \mathcal{E}, i \in \mathcal{V}^{\mathrm{r}}_0, \; j \in \mathcal{V}^{\mathrm{r}} : i < j. \tag{6.67}$$

Table 6.1 summarizes the variables and constraints considered in the three MIP models and shows examples of the number of binary variables and constraints in two arbitrary instances of different sizes. Note that the numbers of binary variables and constraints are strongly influenced by the number of damaged nodes $|\mathcal{V}^{\mathrm{r}}|$. Thus, small changes in the number of damaged nodes can have a significant impact on the size of the problem, and thus, on the difficulty of solving it. The number of demand nodes $|\mathcal{V}^{\mathrm{d}}|$ and crews $|\mathcal{K}|$ seem to have a smaller impact on the number of variables and constraints when compared with $|\mathcal{V}^{\mathrm{r}}|$. As can be observed in Table 6.1, the main shortcoming of MCSRP1 relies on the high number of binary variables, which is greatly influenced by the three-index variable X_{kij} and by the four-index variable N^{r}_{lhij}. In an effort to reduce the number of binary variables, we eliminate one index from these variables to come out with model MCSRP2. Furthermore, in model MCSRP2, we observe that variables N_{lij} and P_{eij} do not need to be defined for all the pairs of damaged nodes $i-j$. Thus, in model MCSRP3, we define these variables only for pair of damaged nodes $i-j$ such that $i<j$. In addition, we eliminate some symmetric solutions by prohibiting the restoration of unnecessary damaged nodes in model MCSRP3. In general, the number of variables considered in the models is reduced from MCSRP1 to MCSRP2 and from MCSRP2 to MCSRP3. Regarding the number of constraints, from MCSRP2 to MCSRP3, several constraints defined for $i>j$ were eliminated, but new constraints were also added. In general, a smaller number of constraints in MCSRP3 is expected compared to both MCSRP2 and MCSRP1. The computational results in Section 6.4.2 show that there is no model unrestrictedly recommended for all situations, e.g., some models are better to quickly return optimal solutions, whereas others are better to always find feasible solutions.

Table 6.1: Variables and constraints of the proposed MIP formulations.

	MCSRP1	MCSRP2	MCSRP3
Binary variables	$W_i, X_{kij}, N^{\mathrm{u}}_{lij}, N^{\mathrm{r}}_{lhij}, V_{lj}$	$W'_{ik}, X'_{ij}, N_{lij}, R_{lhj}, V_{lj}$	$W'_{ik}, X'_{ij}, N'_{lij}, R_{lhj}, V_{lj}$
Continuous variables	$P_{eij}, Y_{ej}, T^{\mathrm{s}}_{lhj}, T^{\mathrm{w}}_{lhj}, Z^{\mathrm{r}}_i, Z^{\mathrm{d}}_i$	$P_{eij}, Y_{ej}, T^{\mathrm{s}}_{lhj}, T^{\mathrm{w}}_{lhj}, Z^{\mathrm{r}}_i, Z^{\mathrm{d}}_i$	$P'_{eij}, Y_{ej}, T^{\mathrm{s}}_{lhj}, T^{\mathrm{w}}_{lhj}, Z^{\mathrm{r}}_i, Z^{\mathrm{d}}_i$
Objective function	(6.1)	(6.1)	(6.1)
Constraints	(6.2)-(6.31)	(6.2), (6.5)-(6.8), (6.27)-(6.53)	(6.2), (6.5)-(6.8), (6.28)-(6.31), (6.33)-(6.36), (6.41)-(6.43), (6.45)-(6.52), (6.54)-(6.67)
# of binary variables[1]	$\|\mathcal{V}^{\mathrm{r}}\|+\|\mathcal{V}^{\mathrm{r}}\|\|\mathcal{V}^{\mathrm{d}}\|+\|\mathcal{K}\|\|\mathcal{V}^{\mathrm{r}}\|^2+\|\mathcal{V}^{\mathrm{u}}\|\|\mathcal{V}^{\mathrm{r}}\|^2+\|\mathcal{R}\|\|\mathcal{V}^{\mathrm{r}}\|^3$	$\|\mathcal{V}^{\mathrm{r}}\|\|\mathcal{K}\|+\|\mathcal{V}^{\mathrm{r}}\|\|\mathcal{V}^{\mathrm{d}}\|+\|\mathcal{V}^{\mathrm{r}}\|^2+\|\mathcal{V}\|\|\mathcal{V}^{\mathrm{r}}\|^2+\|\mathcal{R}\|\|\mathcal{V}^{\mathrm{r}}\|^2$	$\|\mathcal{V}^{\mathrm{r}}\|\|\mathcal{K}\|+\|\mathcal{V}^{\mathrm{r}}\|\|\mathcal{V}^{\mathrm{d}}\|+\|\mathcal{V}^{\mathrm{r}}\|^2+(\|\mathcal{V}\|/2)(\|\mathcal{V}^{\mathrm{r}}\|^2-\|\mathcal{V}^{\mathrm{r}}\|)+\|\mathcal{R}\|\|\mathcal{V}^{\mathrm{r}}\|^2$
# of continuous variables[1]	$\|\mathcal{V}^{\mathrm{d}}\|+\|\mathcal{V}^{\mathrm{r}}\|+\|\mathcal{E}\|\|\mathcal{V}^{\mathrm{d}}\|+\|\mathcal{E}\|\|\mathcal{V}^{\mathrm{r}}\|^2+2\|\mathcal{R}\|\|\mathcal{V}^{\mathrm{r}}\|^2$	$\|\mathcal{V}^{\mathrm{d}}\|+\|\mathcal{V}^{\mathrm{r}}\|+\|\mathcal{E}\|\|\mathcal{V}^{\mathrm{d}}\|+\|\mathcal{E}\|\|\mathcal{V}^{\mathrm{r}}\|^2+2\|\mathcal{R}\|\|\mathcal{V}^{\mathrm{r}}\|^2$	$\|\mathcal{V}^{\mathrm{r}}\|+\|\mathcal{V}^{\mathrm{d}}\|+\|\mathcal{E}\|\|\mathcal{V}^{\mathrm{d}}\|+(\|\mathcal{E}\|/2)(\|\mathcal{V}^{\mathrm{r}}\|^2-\|\mathcal{V}^{\mathrm{r}}\|)+2\|\mathcal{R}\|\|\mathcal{V}^{\mathrm{r}}\|^2$
# of constraints[1]	$\|\mathcal{K}\|+3\|\mathcal{V}^{\mathrm{r}}\|+3\|\mathcal{V}^{\mathrm{d}}\|+\|\mathcal{R}\|\|\mathcal{V}^{\mathrm{r}}\|+\|\mathcal{V}^{\mathrm{r}}\|\|\mathcal{K}\|+\|\mathcal{V}^{\mathrm{d}}\|\|\mathcal{V}^{\mathrm{r}}\|+\|\mathcal{V}^{\mathrm{d}}\|\|\mathcal{V}\|+\|\mathcal{V}^{\mathrm{r}}\|\|\mathcal{R}\|\|\mathcal{K}\|+\|\mathcal{V}^{\mathrm{r}}\|^2+\|\mathcal{V}\|\|\mathcal{V}^{\mathrm{r}}\|^2+2\|\mathcal{R}\|\|\mathcal{V}^{\mathrm{r}}\|^2+\|\mathcal{K}\|\|\mathcal{V}^{\mathrm{r}}\|^2+\|\mathcal{K}\|\|\mathcal{R}\|\|\mathcal{V}^{\mathrm{r}}\|^2+\|\mathcal{V}^{\mathrm{r}}\|^3$	$3\|\mathcal{V}^{\mathrm{d}}\|+4\|\mathcal{V}^{\mathrm{r}}\|+\|\mathcal{V}^{\mathrm{d}}\|\|\mathcal{V}\|+2\|\mathcal{V}^{\mathrm{r}}\|\|\mathcal{R}\|+\|\mathcal{V}^{\mathrm{d}}\|\|\mathcal{V}^{\mathrm{r}}\|+\|\mathcal{V}^{\mathrm{r}}\|\|\mathcal{R}\|\|\mathcal{K}\|+4\|\mathcal{V}^{\mathrm{r}}\|^2+\|\mathcal{V}\|\|\mathcal{V}^{\mathrm{r}}\|^2+3\|\mathcal{K}\|\|\mathcal{V}^{\mathrm{r}}\|^2+\|\mathcal{R}\|\|\mathcal{V}^{\mathrm{r}}\|^2+\|\mathcal{K}\|\|\mathcal{R}\|\|\mathcal{V}^{\mathrm{r}}\|^2+\|\mathcal{V}^{\mathrm{r}}\|^3$	$3\|\mathcal{V}^{\mathrm{d}}\|+7\|\mathcal{V}^{\mathrm{r}}\|+\|\mathcal{V}^{\mathrm{d}}\|\|\mathcal{V}^{\mathrm{r}}\|+\|\mathcal{V}^{\mathrm{d}}\|\|\mathcal{V}\|+2\|\mathcal{V}^{\mathrm{r}}\|\|\mathcal{R}\|+\|\mathcal{V}^{\mathrm{r}}\|\|\mathcal{R}\|\|\mathcal{K}\|+(\|\mathcal{V}\|/2)(\|\mathcal{V}^{\mathrm{r}}\|^2-\|\mathcal{V}^{\mathrm{r}}\|)+\|\mathcal{K}\|\|\mathcal{R}\|\|\mathcal{V}^{\mathrm{r}}\|^2+2\|\mathcal{V}^{\mathrm{r}}\|^2+3\|\mathcal{K}\|\|\mathcal{V}^{\mathrm{r}}\|^2+\|\mathcal{R}\|\|\mathcal{V}^{\mathrm{r}}\|^2+2\|\mathcal{V}^{\mathrm{r}}\|^3$
# binary variables (and constraints) in an instance with $\|\mathcal{V}^{\mathrm{r}}\|=1, \|\mathcal{V}^{\mathrm{d}}\|=15, \|\mathcal{K}\|=1, \|\mathcal{V}\|=21$	38 (409)	39 (413)	18 (394)
# binary variables (and constraints) in an instance with $\|\mathcal{V}^{\mathrm{r}}\|=29, \|\mathcal{V}^{\mathrm{d}}\|=28, \|\mathcal{K}\|=5, \|\mathcal{V}\|=64$	741,762 (261,953)	80,011 (249,217)	52,171 (244,171)

[1] We approximate $|\mathcal{V}^{\mathrm{r}}_0|$ as $|\mathcal{V}^{\mathrm{r}}|$ and $|\mathcal{R}|-1$ as $|\mathcal{R}|$ in the calculation of the number of variables and constraints.

6.3 Properties and valid inequalities

In this section, we state a few properties of the problem and derive valid inequalities (VIs) based on them. We divide the VIs into two groups. In Section 6.3.1, we show the VIs related to the relief path decisions, while in Section 6.3.2, we present the VIs related to the crew scheduling and routing decisions. The VIs related to the relief path decisions are the same for the three models, while those related to routing decisions are specific for each model. We detail only the VIs for MCSRP1 in this section. The VIs related to the routing decisions for the second and third formulations are presented in Appendix A.

6.3.1 VIs related to the relief path decisions

Multiple relief paths $0-i$ may be available to reach a demand node i. Let $\mathcal{P}_i^{\mathtt{d}}$ be the set of possible $0-i$ relief paths. We call $\mathcal{E}_p$ and $\mathcal{V}_p$ as the set of arcs and nodes used in path $p \in \mathcal{P}_i^{\mathtt{d}}$. Similarly, $\mathcal{V}_p^{\mathtt{r}}$ and $\mathcal{V}_p^{\mathtt{u}}$ are the set of damaged and undamaged nodes used in path $p \in \mathcal{P}_i^{\mathtt{d}}$. We define w_p as the sum of the length of the arcs used in path p, i.e., $w_p = \sum_{e\in\mathcal{E}_p} \ell_e$. Since a relief path p connecting the depot with a demand node i must fall within a predefined maximum distance $l_i^{\mathtt{d}}$, p is a feasible path if $w_p \leq l_i^{\mathtt{d}}$. We also define $\theta_{pi}^{\mathtt{d}}$ as the accessibility time of the demand node i if path p is selected to connect the depot with the demand node i and $\theta_j^{\mathtt{r}}$ as the restoration time of the damaged node j.

Given two paths p, $p' \in \mathcal{P}_i^{\mathtt{d}}$ such that $p \neq p'$, we say that p dominates p' if $\mathcal{V}_p^{\mathtt{r}} \subseteq \mathcal{V}_{p'}^{\mathtt{r}}$ and $w_p \leq l_i^{\mathtt{d}}$. In this case, p' is a dominated path. For special cases where $w_p \leq l_i^{\mathtt{d}}$, $w_{p'} \leq l_i^{\mathtt{d}}$ and $\mathcal{V}_p^{\mathtt{r}} = \mathcal{V}_{p'}^{\mathtt{r}}$ for $p \neq p'$, we can eliminate one of the paths, either p or p', from set $\mathcal{P}_i^{\mathtt{d}}$. We define $\mathcal{S}_i^{\mathtt{d}} \subseteq \mathcal{P}_i^{\mathtt{d}}$ as the set of nondominated paths from the depot to demand node i. Given that $\mathcal{S}_i^{\mathtt{d}}$ considers only nondominated paths, there are not two different paths using the same damaged nodes, i.e., $\mathcal{V}_p^{\mathtt{r}} \neq \mathcal{V}_{p'}^{\mathtt{r}}\ \forall p, p' \in \mathcal{S}_i^{\mathtt{d}} : p \neq p'$. Finally, let $\mathcal{P}_i^{\mathtt{d}*} = \{p \in \mathcal{S}_i^{\mathtt{d}} \mid \mathcal{V}_p^{\mathtt{r}} = \emptyset\}$ be the set of nondominated paths that do not visit any damaged node. Using the notation above, we state Propositions 6 and 7 as follows.

Proposition 6. *There is at least one optimal solution for MCSRP1 in which the paths used to connect the depot with the demand nodes are nondominated paths. Such a solution satisfies the following inequalities.*

$$|\mathcal{E}_{p_i^*}| + |\mathcal{V}_{p_i^*}^{\mathtt{u}}| = \sum_{e\in\mathcal{E}_{p_i^*}} Y_{ei} + \sum_{j\in\mathcal{V}_{p_i^*}^{\mathtt{u}}} V_{ji}, \forall\ i \in \mathcal{V}^{\mathtt{d}},\ p_i^* \in \mathcal{P}_i^{\mathtt{d}*} : \mathcal{P}_i^{\mathtt{d}*} \neq \emptyset, \tag{6.68}$$

$$(|\mathcal{E}| + |\mathcal{V}|) \cdot (|\mathcal{V}_p^{\mathtt{r}}| - \sum_{j\in\mathcal{V}_p^{\mathtt{r}}} V_{ji}) \geq \sum_{e\in\mathcal{E}\setminus\mathcal{E}_p} Y_{ei} + \sum_{j\in\mathcal{V}\setminus\mathcal{V}_p} V_{ji}, \forall\ i \in \mathcal{V}^{\mathtt{d}}, p \in \mathcal{S}_i^{\mathtt{d}} : \mathcal{P}_i^{\mathtt{d}*} = \emptyset. \tag{6.69}$$

Proof. Let $p' \in \mathcal{P}_i^{\mathtt{d}}$ be a feasible dominated path and assume that p' is used to connect the depot to the demand node i. The accessibility time of node i depends on the restoration time of the damaged nodes in path p'. Then, $\theta_{p'i}^{\mathtt{d}} = \max_{j\in\mathcal{V}_{p'}^{\mathtt{r}}} \theta_j^{\mathtt{r}}$. Given that p' is a dominated path, there is one nondominated path p that dominates p', i.e., $\mathcal{V}_p^{\mathtt{r}} \subseteq \mathcal{V}_{p'}^{\mathtt{r}}$ and $w_p \leq l_i^{\mathtt{d}}$. Thus, p is also a feasible path. Furthermore, $\max_{j \in \mathcal{V}_p^{\mathrm{r}}} \theta_j^{\mathrm{r}} \leq \max_{j \in \mathcal{V}_{p'}^{\mathrm{r}}} \theta_j^{\mathrm{r}}$ since $\mathcal{V}_p^{\mathrm{r}} \subseteq \mathcal{V}_{p'}^{\mathrm{r}}$. Consequently, $\theta_{pi}^{\mathrm{d}} \leq \theta_{p'i}^{\mathrm{d}}$, and we can select p instead of p' without deteriorating the accessibility time of demand node i. The selection of the nondominated paths over dominated paths is imposed with inequalities (6.68) and (6.69). When there is a path p_i^*, this path is the only nondominated path to reach demand node i and can be fixed in the solution to MCSRP1 by using equations (6.68). If $\mathcal{P}_i^{\mathrm{d}*} \neq \emptyset$, the term $\sum_{j \in \mathcal{V}_p^{\mathrm{r}}} V_{ji} = |\mathcal{V}_p^{\mathrm{r}}|$ indicates that the damaged nodes of nondominated path p have been selected to reach demand node i, and inequalities (6.69) prohibit the selection of nodes and arcs that are not in the nondominated path p. □

Proposition 7. *If $\mathcal{P}_i^{\mathrm{d}*} = \emptyset$, the following inequalities can be added to MCSRP1 to set lower bounds for the accessibility time of the demand nodes:*

$$\sum_{j \in \mathcal{U}_i} V_{ji} \geq 1, \forall\ i \in \mathcal{V}^{\mathrm{d}} : \mathcal{P}_i^{\mathrm{d}*} = \emptyset, \tag{6.70}$$

$$Z_i^{\mathrm{d}} \geq \min_{k \in \mathcal{K}, j \in \mathcal{U}_i} (\rho_{k0j}^* + \delta_{kj}), \forall\ i \in \mathcal{V}^{\mathrm{d}} : \mathcal{P}_i^{\mathrm{d}*} = \emptyset, \tag{6.71}$$

$$\sum_{j \in n_i} V_{ji} = |n_i|, \forall\ i \in \mathcal{V}^{\mathrm{d}} : \mathcal{P}_i^{\mathrm{d}*} = \emptyset, \tag{6.72}$$

$$Z_i^{\mathrm{d}} \geq Z_j^{\mathrm{r}}, \forall\ j \in n_i, i \in \mathcal{V}^{\mathrm{d}} : \mathcal{P}_i^{\mathrm{d}*} = \emptyset, \tag{6.73}$$

$$Z_i^{\mathrm{d}} \geq \sum_{j \in \mathcal{V}^{\mathrm{r}}} \min_{\substack{k \in \mathcal{K}, l \in \mathcal{V}_0^{\mathrm{r}}: \\ l \neq j}} \left\{ \frac{\rho_{klj}^* + \delta_{kj}}{|\mathcal{K}|} \right\} \cdot V_{ji}, \forall\ i \in \mathcal{V}^{\mathrm{d}} : \mathcal{P}_i^{\mathrm{d}*} = \emptyset, \tag{6.74}$$

where ρ_{kij}^* is the shortest time for crew k to travel from node i to node j; $\mathcal{U}_i = \bigcup_{p \in \mathcal{S}_i^{\mathrm{d}}} \mathcal{V}_p^{\mathrm{r}}$ contains all the damaged nodes of the nondominated paths; and $n_i = \bigcap_{p \in \mathcal{S}_i^{\mathrm{d}}} \mathcal{V}_p^{\mathrm{r}}$ contains the damaged nodes that are used in all the nondominated paths.

Proof. If $\mathcal{P}_i^{\mathrm{d}*} = \emptyset$, it is clear that at least one damaged node j of the nondominated paths in $\mathcal{S}_i^{\mathrm{d}}$ must be used in the relief path $0 - i$ (inequalities (6.70)). In this case, some crew must arrive and repair such damaged node j. We know that to repair any damaged node j with a given crew k, the crew must arrive at node j consuming at least some travel time ρ_{k0j}^* and some repair time δ_{kj}. In this way, by selecting the minimum travel time plus the repair time to repair one of the damaged nodes of the nondominated paths in $\mathcal{S}_i^{\mathrm{d}}$, inequalities (6.71) establish a lower bound for the accessibility time of demand node i. If a node j exists in all the nondominated paths in set $\mathcal{S}_i^{\mathrm{d}}$, such node must be necessarily used in the relief paths $0 - i$ (inequalities (6.72)), and the demand node i does not become accessible before the restoration of such damaged node j (inequalities (6.73)). Finally, inequalities (6.74) state that all the damaged nodes that must be used in the paths from the depot to the demand node i should be repaired before node i becomes accessible. Given that we do not know the crew that will perform the restoration of each damaged node or the paths used by the crews in advance, we select the shortest repair time plus the travel time to arrive at the damaged nodes. Since any crew can be used to perform the restoration, the shortest time is divided by the number of crews. □

6.3.2 VIs related to the routing decisions

The VIs for the routing decisions are similar to those defined for the relief path decisions. We define $\mathcal{P}^{\mathtt{r}}_{ij}$ as the set of possible $i-j$ crew paths. Let $t^{\mathtt{v}}_{kjp}$ be the arrival time of crew k at damaged node $j \in \mathcal{V}^{\mathtt{r}}_p$ in path p; $t^{\mathtt{w}}_{kjp}$ be the waiting time of crew k at the damaged node $j \in \mathcal{V}^{\mathtt{r}}_p$ in path p; and t_{kp} be the total travel time of crew k in path p, i.e., $t_{kp} = \sum_{e\in\mathcal{E}_p} \tau_{ke}$. We define $\mathcal{P}^{\mathtt{r}*}_{kij}$ (resp. $\mathcal{F}^{\mathtt{r}*}_{kij}$) as the set of paths with the shortest travel time between nodes i and j with crew k using (resp. not using) damaged nodes, i.e.,

$$\mathcal{P}^{\mathtt{r}*}_{kij} = \{p \in \mathcal{P}^{\mathtt{r}}_{ij} \,|t_{kp} \le t_{kp'}, \forall p' \in \mathcal{P}^{\mathtt{r}}_{ij}\},\ \forall\ k \in \mathcal{K},\ i \in \mathcal{V}^{r}_0,\ j \in \mathcal{V}^{r},$$

$$\mathcal{F}^{\mathtt{r}*}_{kij} = \{p \in \mathcal{P}^{\mathtt{r}}_{ij} \,|\, \mathcal{V}^{\mathtt{r}}_p = \emptyset\ , t_{kp} \le t_{kp'}, \forall p' \in \mathcal{P}^{\mathtt{r}}_{ij} : \mathcal{V}^{\mathtt{r}}_{p'} = \emptyset\},\ \forall\ k \in \mathcal{K},\ i \in \mathcal{V}^{r}_0,\ j \in \mathcal{V}^{r}.$$

Given two paths $p', p \in \mathcal{P}^{\mathtt{r}}_{ij} : p \ne p'$, we say that p dominates p' for a crew k if $\mathcal{V}^{\mathtt{r}}_p \subseteq \mathcal{V}^{\mathtt{r}}_{p'}$ and $t_{kp} \le t_{kp'}$. For cases where $\mathcal{V}^{\mathtt{r}}_p = \mathcal{V}^{\mathtt{r}}_{p'}$ and $t_{kp} = t_{kp'}$ for $p \ne p'$, we can eliminate one of the paths, either p or p', from set $\mathcal{P}^{\mathtt{r}}_{ij}$. We define $\mathcal{S}^{\mathtt{r}}_{kij} \subseteq \mathcal{P}^{\mathtt{r}}_{ij}$ as the set of nondominated paths from node i to node j using crew k. We also define $\mathcal{D}_p$ as the set of paths using nodes of set $\mathcal{V}^{\mathtt{r}}_p$ that are not dominated by p. In this way,

$$\mathcal{D}_p = \{p' \in \mathcal{S}^{\mathtt{r}}_{kij} | \mathcal{V}^{\mathtt{r}}_p \subseteq \mathcal{V}^{\mathtt{r}}_{p'}, t_{kp} \ge t_{kp'}\}, \forall p \in \mathcal{S}^{\mathtt{r}}_{kij},\ k \in \mathcal{K},\ i \in \mathcal{V}^{\mathtt{r}}_0,\ j \in \mathcal{V}^{\mathtt{r}}.$$

Let p^*_{kij} and f^*_{kij} be the elements of sets $\mathcal{P}^{\mathtt{r}*}_{kij}$ and $\mathcal{F}^{\mathtt{r}*}_{kij}$, respectively. Additionally, we define h_p as the hth damaged node visited in path p, for $h = 1, ..., H$, where $H = |\mathcal{V}^{\mathtt{r}}_p|$. In this way, H_p denotes the last damaged node visited in path p. Let ρ_{kij} be the shortest travel time required for crew k to travel from node i to node j without using damaged nodes. Based on this notation, we state Propositions 8 and 9, which assume that crew k can repair both nodes i and j.

Proposition 8. *There is at least one optimal solution for MCSRP1 in which the paths used by crew k to travel from node i to node j are nondominated paths, where i and j are consecutive nodes in the schedule of crew k, i.e., $X_{kij} = 1$. Such a solution satisfies the following inequalities:*

$$\sum_{e\in\mathcal{E}_{f^*_{kij}}} P_{eij} + \sum_{l\in\mathcal{V}^{\mathtt{u}}_{f^*_{kij}}} N^{\mathtt{u}}_{lij} \ge (|\mathcal{E}_{f^*_{kij}}| + |\mathcal{V}^{\mathtt{u}}_{f^*_{kij}}|) \cdot X_{kij}, \forall k \in \mathcal{K}, i \in \mathcal{V}^{\mathtt{r}}_0, j \in \mathcal{V}^{\mathtt{r}} : \mathcal{F}^{\mathtt{r}*}_{kij} \ne \emptyset, t_{kf^*_{kij}} = t_{kp^*_{kij}}, \tag{6.75}$$

$$(|\mathcal{E}| + |\mathcal{V}|) \cdot (1 + |\mathcal{V}^{\mathtt{r}}_p| - X_{kij} - \sum_{h\in\mathcal{R}} \sum_{l\in\mathcal{V}^{\mathtt{r}}_p} N^{\mathtt{r}}_{lhij}) \ge \sum_{e\in\mathcal{E}\setminus \bigcup_{p'\in\mathcal{D}_p} \mathcal{E}_{p'}} P_{eij} + \sum_{l\in\mathcal{V}\setminus \bigcup_{p'\in\mathcal{D}_p} \mathcal{V}_{p'}} N^{\mathtt{u}}_{lij},$$

$$\forall\ k \in \mathcal{K},\ i \in \mathcal{V}^{\mathtt{r}}_0,\ j \in \mathcal{V}^{\mathtt{r}},\ p \in \mathcal{S}^{\mathtt{r}}_{kij} : (\mathcal{F}^{\mathtt{r}*}_{kij} = \emptyset) \vee (\mathcal{F}^{\mathtt{r}*}_{kij} \ne \emptyset, p \ne f^*_{kij}, t_{kf^*_{kij}} > t_{kp^*_{kij}}). \tag{6.76}$$

Proof. We need to prove that given a dominated path p', we can replace it by a nondominated path p without increasing the restoration time of damaged node j. According to constraints (6.3) and (6.4), the restoration time $\theta^{\mathtt{r}}_j$ of node j repaired by a given crew k that uses path p as crew path $i-j$ is calculated as $\theta^{\mathtt{r}}_j = \theta^{\mathtt{r}}_i + t_{kp} + \delta_{kj}$ if $\mathcal{V}^{\mathtt{r}}_p = \emptyset$, and $\theta^{\mathtt{r}}_j = \max\{\theta^{\mathtt{r}}_i + t_{kp} + \delta_{kj}, t^{\mathtt{w}}_{kH_pp} +$

$t^{\mathtt{v}}_{kH_pp} + \rho_{kH_pj} + \delta_{kj}\}$ if $\mathcal{V}^{\mathtt{r}}_p \neq \emptyset$. We focus on the case with $\mathcal{V}^{\mathtt{r}}_p \neq \emptyset$, and the development for the other case follows similarly. The arrival time $t^{\mathtt{v}}_{kh_pp}$ at the hth damaged node in path p can be computed as $t^{\mathtt{v}}_{kh_pp} = t^{\mathtt{v}}_{k(h-1)_pp} + t^{\mathtt{w}}_{k(h-1)_pp} + \rho_{k(h-1)_ph_p}$. Then, recursively, $t^{\mathtt{v}}_{kH_pp}$ can be evaluated as

$$t^{\mathtt{v}}_{kH_pp} = \sum_{h'=1}^{h'=H-1} t^{\mathtt{w}}_{k(h')_pp} + \sum_{h'=1}^{h'=H-1} \rho_{k(h')_p(h'+1)_p} + \rho_{ki1_p} + \theta^{\mathtt{r}}_i.$$

Then,

$$t^{\mathtt{w}}_{kH_pp} + t^{\mathtt{v}}_{kH_pp} + \rho_{kH_pj} = t^{\mathtt{w}}_{kH_pp} + \sum_{h'=1}^{h'=H-1} t^{\mathtt{w}}_{k(h')_pp} + \sum_{h'=1}^{h'=H-1} \rho_{k(h')_p(h'+1)_p} + \rho_{ki1_p} + \theta^{\mathtt{r}}_i + \rho_{kH_pj}.$$

Grouping similar terms, we obtain

$$t^{\mathtt{w}}_{kH_pp} + t^{\mathtt{v}}_{kH_pp} + \rho_{kH_pj} = \sum_{h'=1}^{h'=H} t^{\mathtt{w}}_{k(h')_pp} + t_{kp} + \theta^{\mathtt{r}}_i = \sum_{l\in\mathcal{V}^{\mathtt{r}}_p} t^{\mathtt{w}}_{klp} + t_{kp} + \theta^{\mathtt{r}}_i,$$

in which $t_{kp} = \sum\limits_{h'=1}^{h'=H-1} \rho_{k(h')_p(h'+1)_p} + \rho_{ki1_p} + \rho_{kH_pj}$. Therefore, the calculation of $\theta^{\mathtt{r}}_j$ can be expressed as

$$\theta^{\mathtt{r}}_j = \max\{\theta^{\mathtt{r}}_i + t_{kp} + \delta_{kj}, \sum_{l\in\mathcal{V}^{\mathtt{r}}_p} t^{\mathtt{w}}_{klp} + t_{kp} + \theta^{\mathtt{r}}_i + \delta_{kj}\}. \tag{6.77}$$

Now, let us consider a path p' dominated by a nondominated path p. We have $t_{kp} \leq t_{kp'}$ according to the definition. Then, $\theta^{\mathtt{r}}_i + t_{kp} + \delta_{kj} \leq \theta^{\mathtt{r}}_i + t_{kp'} + \delta_{kj}$. Additionally, since $\mathcal{V}^{\mathtt{r}}_p \subseteq \mathcal{V}^{\mathtt{r}}_{p'}$, the use of path p' implies waiting for the restoration of the nodes that belong to set $\mathcal{V}^{\mathtt{r}}_p$ and waiting for the restoration of additional damaged nodes that belong to $\mathcal{V}^{\mathtt{r}}_{p'} \setminus \mathcal{V}^{\mathtt{r}}_p$. Therefore, $\sum_{l\in\mathcal{V}^{\mathtt{r}}_p} t^{\mathtt{w}}_{klp} + t_{kp} + \theta^{\mathtt{r}}_i + \delta_{kj} \leq \sum_{l\in\mathcal{V}^{\mathtt{r}}_{p'}} t^{\mathtt{w}}_{klp'} + t_{kp'} + \theta^{\mathtt{r}}_i + \delta_{kj}$. Consequently, we can use p instead of p' without increasing the restoration time of node j. Inequalities (6.75)-(6.76) are analogous to inequalities (6.68)-(6.69) to force the selection of a nondominated path over dominated paths. If $\mathcal{F}^{\mathtt{r}*}_{kij} \neq \emptyset$ and $t_{kp^*_{kij}} = t_{kf^*_{kij}}$, f^*_{kij} is the only nondominated path in set $\mathcal{S}^{\mathtt{r}}_{kij}$ and can be fixed in the solution to MCSRP1 with equations (6.75) if $X_{kij} = 1$. Inequalities (6.76) prevent the selection of dominated paths over nondominated paths. If path p is a nondominated path and the damaged nodes of path p are used to travel from node i to node j ($\sum_{h\in\mathcal{R}}\sum_{l\in\mathcal{V}^{\mathtt{r}}_p} N^{\mathtt{r}}_{lhij} = |\mathcal{V}^{\mathtt{r}}_p|$), inequalities (6.76) prohibit the use of a path dominated by path p if $\mathcal{F}^{\mathtt{r}*}_{kij} = \emptyset$ or if $\mathcal{F}^{\mathtt{r}*}_{kij} \neq \emptyset, p \neq f^*_{kij}$ and $t_{kf^*_{kij}} > t_{kp^*_{kij}}$. □

Proposition 9. *The following inequalities can be added to MCSRP1 to set lower bounds for the restoration time of the damaged nodes.*

$$Z^{\mathtt{r}}_j \geq (t_{kp^*_{k0j}} + \delta_{kj}) \cdot X_{k0j} + \sum_{i\in\mathcal{V}^{\mathtt{r}}} ((t_{kp^*_{k0i}} + \delta_{ki} + t_{kp^*_{kij}} + \delta_{kj}) \cdot X_{kij}), \forall\, k \in \mathcal{K},\ j \in \mathcal{V}^{\mathtt{r}}, \tag{6.78}$$

$$Z^{\mathtt{r}}_j \geq Z^{\mathtt{r}}_i + t_{kp^*_{kij}} + \delta_{kj} - M \cdot (1 - X_{kij}), \forall\, k \in \mathcal{K},\ i \in \mathcal{V}^{\mathtt{r}}_0,\ j \in \mathcal{V}^{\mathtt{r}}. \tag{6.79}$$

Proof. Inequalities (6.78) are based on the fact that any damaged node j considered in the schedule of a given crew k must be reached by it using some path p. Although path p is unknown, if j is the first node in the schedule of crew k ($X_{k0j} = 1$), $t_{kp^*_{k0j}}$ can be used as a lower bound for the travel time in this path. Furthermore, if j is not the first node in the schedule of crew k, this crew must spend additional time to arrive and repair some node i ($t_{kp^*_{k0i}} + \delta_{ki}$) before traveling to node j. Inequalities (6.79) are similar to constraints (6.3) replacing the term $\sum_{e\in\mathcal{E}} \tau_{ke} \cdot P_{eij}$ with a lower bound for the travel time between nodes i and j ($t_{kp^*_{kij}}$). $\square$

The number of nodes in set $\mathcal{R}$ can be redefined for each pair of nodes i and j based on the number of damaged nodes of the nondominated paths. Thus, instead of $\mathcal{R}$, we can use a set $\mathcal{R}_{ij}$ defined as follows:

$$\mathcal{R}_{ij} = \{1, ..., \max_{p\in\mathcal{S}^{\mathtt{r}}_{kij}, k\in\mathcal{K}} |\mathcal{V}^{\mathtt{r}}_p|\}.$$

This redefinition of $\mathcal{R}$ can drastically reduce the number of variables depending on the position h because the use of fewer damaged nodes in the nondominated paths is expected.

6.3.3 Separation algorithms for the VIs

To generate the VIs related to relief path decisions, we need the estimation of sets $\mathcal{U}_i, \mathcal{P}^{\mathtt{d}*}_i, n_i$, and $\mathcal{S}^{\mathtt{d}}_i$. Analogously, to generate the VIs related to routing decisions, we need to estimate sets $\mathcal{P}^{\mathtt{r}*}_{kij}, \mathcal{F}^{\mathtt{r}*}_{kij}$, and $\mathcal{S}^{\mathtt{r}}_{kij}$. $\mathcal{P}^{\mathtt{d}*}_i, n_i, \mathcal{P}^{\mathtt{r}*}_{kij}$, and $\mathcal{F}^{\mathtt{r}*}_{kij}$ can be estimated by separation algorithms based on the shortest path problem (SPP). For instance, $\mathcal{P}^{\mathtt{r}*}_{kij}$ is found by solving the SPP between damaged nodes i and j over the original graph G. Additionally, $\mathcal{F}^{\mathtt{r}*}_{kij}$ is evaluated by solving the SPP between damaged nodes i and j over a graph G', in which the damaged nodes and the arcs incident to them are removed. Similarly, $\mathcal{P}^{\mathtt{d}*}_i$ is determined by solving the SPP between the depot and demand node i over the same graph G'.

Set n_i can be determined by solving one SPP for each damaged node $l \in \mathcal{V}^r$. Basically, to know if a damaged node l is an element of n_i, we remove node l and its incident arcs from graph G, and the SPP between the depot and demand node i is solved. If there is a path from the depot to demand node i with a cost less than or equal to $l^{\mathtt{d}}_i$, node l is not an element of n_i. Otherwise, we insert node l into n_i. Set n_i can also be found as $n_i = \bigcap_{p\in\mathcal{S}^{\mathtt{d}}_i} \mathcal{V}^{\mathtt{r}}_p$ if set $\mathcal{S}^{\mathtt{d}}_i$ is available.

The estimation of $\mathcal{U}_i, \mathcal{S}^{\mathtt{d}}_i$ and $\mathcal{S}^{\mathtt{r}}_{kij}$ is not trivial and may require the development of specialized algorithms that are not the focus here. Thus, we approximate these sets in such a way to maintain the inequalities valid, although they might be weaker. For instance, instead of considering all the nondominated paths in $\mathcal{S}^{\mathtt{r}}_{kij}$, we consider a subset $\widehat{\mathcal{S}}^{\mathtt{r}}_{kij} \subseteq \mathcal{S}^{\mathtt{r}}_{kij}$ with some nondominated paths. The algorithm used to find subset $\widehat{\mathcal{S}}^{\mathtt{r}}_{kij}$ is outlined in Algorithm 8. First, we find the shortest path p from the depot to demand node i considering the original graph G. Path p is a nondominated path since it is the path with the smallest t_{kp}. Then, we remove the damaged nodes $\mathcal{V}^r_p$ used in path p from graph G, and the SPP is solved again. The new path p' is a path that is nondominated by p because it uses different damaged nodes. Furthermore, p' is the shortest path using nodes $\mathcal{V}^r_{p'}$ and dominates other paths using such nodes but at a higher cost. The process is repeated iteratively until the shortest path algorithm cannot find more paths that are feasible. A similar idea can be used to find a subset $\widehat{\mathcal{S}}^{\mathtt{d}}_i \subseteq \mathcal{S}^{\mathtt{d}}_i$.

Algorithm 8 Algorithm to find the set $\widehat{\mathcal{S}}^{\mathtt{r}}_{kij}$.

```
Input:
Graph $G = (\mathcal{V}, \mathcal{E})$; Indices $i, j, k$; Parameter $\tau_{ke}$, $\forall e \in \mathcal{E}$;
Output:
Paths $p \in \widehat{\mathcal{S}}^{\mathtt{r}}_{kij}$;
 1: Initialization:
 2: $t_{kp} := 0$;
 3: while $t_{kp} < +\infty$ do
 4:     Find the shortest path $p$ from node $i$ to node $j$;
 5:     if path $p$ exists then
 6:         $t_{kp} := \sum_{e \in \mathcal{E}_p} \tau_{ke}$;
 7:         Save path $p$ in set $\widehat{\mathcal{S}}^{\mathtt{r}}_{kij}$;
 8:         Remove nodes in $\mathcal{V}^r_p$ and the arcs incident to them from graph $G$;
 9:     else
10:         $t_{kp} := +\infty$;
11:     end if
12: end while
13: return set $\widehat{\mathcal{S}}^{\mathtt{r}}_{kij}$;
```

Set $\mathcal{U}_i$ is simply considered as $\mathcal{U}_i = \mathcal{V}^r$, which maintains the validity of the inequalities involving $\mathcal{U}_i$. Finally, since we do not have the exact set $\mathcal{S}^{\mathtt{r}}_{kij}$, $\mathcal{R}_{ij}$ is approximated as follows:

$$\mathcal{R}_{ij} = \{1, ..., \max_{p \in \mathcal{P}^{\mathtt{r}}_{ij}} |\mathcal{V}^{\mathtt{r}}_p|\},$$

where $\max\limits_{p \in \mathcal{P}^{\mathtt{r}}_{ij}} |\mathcal{V}^{\mathtt{r}}_p|$ is equal to the number of nodes in the path with more damaged nodes in $\mathcal{P}^{\mathtt{r}}_{ij}$. Such a calculation can be performed with an algorithm to find the elementary longest path (the one with more damaged nodes) from node i to node j. We use an integer linear programming model from the literature (Bui et al., 2016) to find such a path.

6.4 Computational results

The goal of this section is twofold: first, to compare the performance of the proposed formulations and valid inequalities (Section 6.4.2); second, to analyze the solutions of the problem in a practical case based on a real-world natural disaster (Section 6.4.3). From this analysis, we illustrate the implication of the multiple crews in the problem and provide managerial insights that might be useful in practice. All the algorithms were coded in the C++ programming language and run on a PC with an AMD Opteron 6172 processor with 16.0 GB of RAM and a single thread. The MIP models were solved by the IBM CPLEX Optimization Solver 12.8. To avoid running out of memory, we allow CPLEX to store the branch-and-bound tree in a file. The stopping criterion was either the elapsed time exceeding the time limit of 3,600 seconds or the optimality gap being smaller than 10^{-4}. All the remaining parameters of CPLEX were kept at their default values.

6.4.1 Instance and experiment description

The models were tested using two different sets of instances. The first set (set L) is derived from the benchmark instances for the SCSRP. We selected the first 12 classes of instances presented in Section 4.3.1. For each class, we consider 12 instances. The second set of instances (set CS) is based on a real network affected by a disaster in the State of Rio de Janeiro in Brazil. This disaster has been studied before in the literature (Alem et al., 2016; Moreno et al., 2016, 2017, 2018) but with a different focus. The authors considered some damaged arcs along the network for different disaster scenarios, but they did not focus on the restoration of such arcs. Six main highways and 13 main cities were affected by this disaster. Although the highways may have been affected in more than one point, we consider some instances assuming one damaged node in each one of the affected highways. Additionally, we generated instances based on the original network of the disaster but randomly selected the location of the damaged nodes. In this respect, we generate instances with 6, 10 and 14 damaged nodes. Further details on the instance generation are provided in Appendix B.

Repair and travel times for the multiple crews were generated based on the literature (Taillard, 1999). They are stated as $\tau_{ke} = \alpha_k^1 \tau'_e$ and $\delta_{kj} = \alpha_k^2 \delta'_j$, where α_k^1 and α_k^2 are travel and repair factors randomly generated either in the interval $[0.4,\ 1.0]$ or in the interval $[1.0,\ 2.0]$, while δ'_j and τ'_e are the repair and travel times in the SCSRP, respectively. We generate the velocity factors in such a way that no single crew is much better or worse than the others. We also consider that the crews with heavier machinery may perform a faster restoration but may spend more time arriving at the damaged nodes. Thus, the crew with a travel factor generated in the interval $[0.4,\ 1.0]$ has a repair factor generated in the interval $[1.0,\ 2.0]$, and vice versa. Table 6.2 shows the main characteristics of the proposed instances. The first crew has factors $\alpha_1^1 = \alpha_1^2 = 1.0$. In this way, we keep the travel and repair times of this crew as those used in the SCSRP. Additionally, the crews have different factors over the classes of instances. There are 144 instances from the literature and 114 instances based on the real-world case. We run experiments with 1, 3 and 5 crews, totaling 774 instances.

Table 6.2: Set of instances.

Instance	Travel (repair) factors of the crews[1]					Demand	Damaged	Total	Total	Total
class	Crew 1	Crew 2	Crew 3	Crew 4	Crew 5	nodes	nodes	nodes	arcs	instances
L1	1.0 (1.0)	0.7 (1.2)	1.5 (0.7)	0.7 (1.3)	1.8 (0.9)	15	1 to 9	21 to 29	40 to 48	12
L2	1.0 (1.0)	0.7 (1.9)	1.7 (0.6)	0.9 (1.7)	1.3 (0.8)	15	1 to 9	21 to 29	38 to 46	12
L3	1.0 (1.0)	0.6 (2.0)	1.1 (0.4)	0.7 (1.6)	1.3 (0.9)	15	1 to 9	21 to 29	38 to 46	12
L4	1.0 (1.0)	0.7 (1.8)	1.6 (0.9)	0.9 (1.4)	1.6 (0.9)	19	2 to 10	27 to 35	42 to 50	12
L5	1.0 (1.0)	0.9 (1.8)	1.1 (0.6)	0.7 (1.8)	1.4 (0.6)	19	1 to 9	26 to 34	38 to 48	12
L6	1.0 (1.0)	0.8 (1.7)	1.8 (0.4)	0.5 (2.0)	1.8 (0.9)	19	1 to 9	26 to 34	40 to 48	12
L7	1.0 (1.0)	0.4 (1.9)	1.7 (0.6)	0.8 (1.6)	1.1 (0.8)	24	4 to 20	34 to 50	87 to 103	12
L8	1.0 (1.0)	0.8 (1.7)	1.4 (0.9)	0.7 (1.1)	1.6 (0.6)	24	4 to 22	34 to 52	93 to 111	12
L9	1.0 (1.0)	0.7 (1.5)	1.7 (0.9)	0.8 (1.9)	1.9 (0.5)	24	4 to 21	34 to 51	88 to 105	12
L10	1.0 (1.0)	0.7 (1.4)	1.4 (0.8)	0.6 (1.1)	1.4 (0.6)	28	5 to 29	40 to 64	123 to 147	12
L11	1.0 (1.0)	0.6 (1.3)	1.3 (0.5)	0.8 (1.1)	1.2 (0.9)	28	5 to 28	40 to 63	120 to 143	12
L12	1.0 (1.0)	0.5 (1.4)	1.6 (0.5)	0.9 (2.0)	1.8 (0.4)	28	5 to 28	40 to 63	118 to 141	12
CS0	1.0 (1.0)	0.7 (1.2)	1.5 (0.7)	0.7 (1.3)	1.8 (0.9)	13	6	66	95	6
CS1	1.0 (1.0)	0.7 (1.2)	1.5 (0.7)	0.7 (1.3)	1.8 (0.9)	13	6 to 14	66 to 74	95 to 103	18
CS2	1.0 (1.0)	0.7 (1.9)	1.7 (0.6)	0.9 (1.7)	1.3 (0.8)	13	6 to 14	66 to 74	95 to 103	18
CS3	1.0 (1.0)	0.6 (2.0)	1.1 (0.4)	0.7 (1.6)	1.3 (0.9)	13	6 to 14	66 to 74	95 to 103	18
CS4	1.0 (1.0)	0.7 (1.8)	1.6 (0.9)	0.9 (1.4)	1.6 (0.9)	20	6 to 14	66 to 74	95 to 103	18
CS5	1.0 (1.0)	0.9 (1.8)	1.1 (0.6)	0.7 (1.8)	1.4 (0.6)	20	6 to 14	66 to 74	95 to 103	18
CS6	1.0 (1.0)	0.8 (1.7)	1.8 (0.4)	0.5 (2.0)	1.8 (0.9)	20	6 to 14	66 to 74	95 to 103	18
Total										$258 \cdot 3 = 774$

[1] Values for $\alpha_k^1(\alpha_k^2)$. For instances with $|\mathcal{K}| < 5$ consider the first $|\mathcal{K}|$ crews.

The computational experiments considering the proposed models and valid inequalities were conducted in four phases, as presented in Table 6.3. First, we run the three formulations without including any valid inequality. Second, we run them all but include all the devised valid inequalities. Third, we run only formulation MCSRP3 with some of the VIs. The objective is to verify the impact of the different types of VIs on the performance of the formulations. In this respect, VIs were divided into four groups: (VIs1) VIs to set lower bounds for variables $Z_i^{\mathtt{d}}$ and V_{ji}; (VIs2) VIs to impose and select nondominated relief paths over dominated relief paths; (VIs3) VIs to set lower bounds for variables $Z_i^{\mathtt{r}}$; (VIs4) VIs to impose and select nondominated crew paths over dominated crew paths. Finally, we apply the graph reduction strategy proposed in Chapter 4 to improve the performance of the MRRP3+VIs approach. The graph reduction consists of solving the problem over a graph with a reduced number of nodes, thus deriving lower bounds for the variables of the original problem. It relies on the elimination of intersection nodes and arcs that are not directly connected to either damaged or demand nodes. The reduced graph is commonly built by splitting the set of damaged nodes into subsets according to an initial solution. Since we do not resort to trivial initial solutions, the damaged nodes are labeled from 1 to $|\mathcal{V}^r|$, and the subsets are built by selecting the nodes in increasing order of the label. We consider reduced graphs with 4 damaged nodes, which can be easily solved by the proposed formulations. A description of the graph reduction strategy is presented in 4.2.7.

6.4.2 Computational performance of the mathematical formulations

In this section, we analyze the computational performance of the proposed models and valid inequalities. First, we compare the three MCSRP formulations (MCSRP1, MCSRP2, MCSRP3) with and without the VIs. Figure 6.1 presents the performance profiles (Dolan and Moré, 2002) for the MCSRP models based on the optimality gap for the considered instances. The optimality gap is computed as $gap = \frac{Z^U - Z^L}{Z^U}$, in which Z^U is the upper bound or cost of the best integer

Table 6.3: Solution strategies.

Strategy	Description
MCSRP1	First MCSRP formulation.
MCSRP2	Second MCSRP formulation.
MCSRP3	Third MCSRP formulation.
MCSRP1+VIs	First MCSRP formulation + all VIs.
MCSRP2+VIs	Second MCSRP formulation + all VIs.
MCSRP3+VIs	Third MCSRP formulation + all VIs.
MCSRP3+VIs1	Third MCSRP formulation + VIs (6.70)-(6.74).
MCSRP3+VIs2	Third MCSRP formulation + VIs (6.68),(6.69).
MCSRP3+VIs3	Third MCSRP formulation + VIs (A.3),(A.4).
MCSRP3+VIs4	Third MCSRP formulation + VIs (A.5)-(A.8).
MCSRP3+VIs*	Third MCSRP formulation + all VIs + graph reduction.

solution and Z^L is the lower bound. The value $P(f,q)$ (y-axis) when $q > 0$ (x-axis) indicates the fraction of instances for which a strategy f provides solutions with a gap within a factor of 2^q of the best obtained gap. The value of $P(f,q)$ when $q = 0$ is the fraction of instances for which the strategy f reached the best gap. For a given instance, the best gap is the lowest gap found considering all the approaches. For example, the red asterisk (*) in Figure 6.1 indicates that for 92% of the instances, strategy MCSRP1+VIs provides solutions with gaps within a factor of $2^{0.58}$ (1.49) of the best gap.

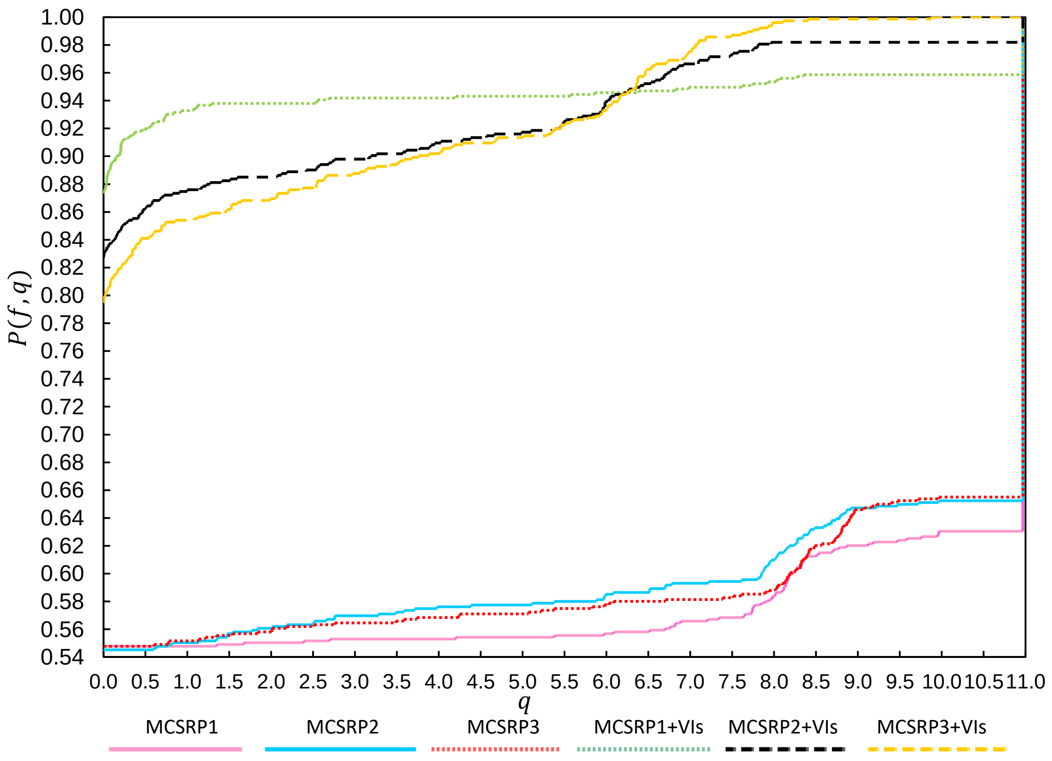


Figure 6.1: Performance profiles based on gap for the MCSRP formulations.

Note that the VIs significantly improve the computational performance of the three formulations. Without the VIs, models MCSRP1, MCSRP2, and MCSRP3 find the best gap for 54.65%, 54.39%, and 54.52% of the instances, respectively. When the inequalities are included, the percentage of instances with the best gap increases to 86.69%, 82.04%, and 79.32%, respectively. With the VIs, model MCSRP1 demonstrates good performance in approximately

92% of the considered cases, but it presents the worst convergence when all the instances are considered. MCSRP2+VIs and MCSRP3+VIs showed a more stable convergence, even though they achieved the best gap for a smaller number of instances when compared to MCSRP1+VIs. MCSRP3 outperformed MCSRP2, both with and without the VIs.

Table 6.4 shows the number and percentage of instances for which CPLEX found feasible solutions (#feas, %feas), the number and percentage of instances that CPLEX solved to optimality (#opt, %opt), the average elapsed time in seconds (Avg. time), and the average number of nodes processed in the branch-and-cut tree (nodes B&C). Tables D.1 and D.2 in D present additional results of the three models with and without the VIs for instances with different numbers of crews. For all models, the elapsed time is significantly reduced by the VIs. On average, for model MCSRP1 (MCSRP2, MCSRP3) the elapsed time is reduced by 51.82% (51.30%, 41.42%) in set L and 52.05% (29.52%, 23.80%) in set CS. The impact of the VIs is more pronounced in the CS instances. For example, the number of instances solved to optimality with model MCSRP3 in set CS increased 80.87% with the VIs, while in set L, it increased 18.36%.

Table 6.4: Average results of the MCSRP formulations.

	Set L (432 instances)					Set CS (342 instances)					
Solution method	#feas	%feas	#opt	%opt	Avg. time (seconds)	#feas	%feas	#opt	%opt	Avg. time (seconds)	Nodes B&C[1]
MCSRP1	365	84.49	316	73.15	1,065	121	35.38	113	33.04	2,427	24,078
MCSRP2	379	87.73	300	69.44	1,170	126	36.84	113	33.04	2,372	30,950
MCSRP3	381	88.19	305	70.60	1,151	126	36.84	115	33.63	2,362	17,566
MCSRP1+VIs	426	98.61	376	87.04	512	316	92.40	242	70.76	1,194	10,571
MCSRP2+VIs	432	100.00	373	86.34	570	328	95.91	209	61.11	1,590	26,020
MCSRP3+VIs	432	100.00	361	83.56	674	342	100.00	208	60.82	1,718	68,565

[1] Values based on the feasible solutions. Values for MCSRP1, MCSRP2, and MCSRP3 are not representative.

In both sets of instances, L and CS, MCSRP1 outperformed MCSRP2 and MCSRP3 regarding the number of optimal solutions, but it had difficulty in finding feasible solutions in more cases. Eliminating some symmetric solutions in the third formulation was effective in finding feasible solutions for all the considered instances within the time limit, but the number of solutions that proved optimal was smaller than in MCSRP1+VIs. The effectiveness of MCSRP3+VIs in finding feasible solutions appeared to be related to the number of nodes that can be processed in the B&C tree, which is significantly higher when solving MCSRP3+VIs compared to the other two formulations. For the instances based on the real-world disaster aftermath (set CS), model MCSRP1+VIs is 30.54% faster than MCSRP3+VIs and solves 14.05% more instances to optimality, although it fails to find feasible solutions in 7.6% of the cases. In contrast, MCSRP3+VIs finds feasible solutions for all CS instances. Evidently, there is a trade-off that can be explored in practical situations according to preferences or necessities of the decision-maker. On the one hand, MCSRP1+VIs is better at finding optimal solutions for some instances quickly, while it struggles to find feasible solutions in some other instances. On the other hand, MCSRP3+VIs always finds feasible solutions (many good-quality ones), but optimality certificate is slightly compromised. The second model presents a balance between the first and the third models. Regarding MCSRP1+VIs, MCSRP2+VIs returns more feasible solutions at the expenses of increased computational times, and fewer solutions proven optimal;

whereas regarding MCSRP3+VIs, MCSRP2+VIs is faster and provides more optimal solutions, even though it finds fewer feasible solutions.

Figure 6.2 presents the performance profiles based on the optimality gap to compare how the different valid inequalities affect the performance of MCSRP3. The impact of the different VIs in the performance of models MCSRP1 and MCSRP2 is similar. Notice that the performance profiles did not converge to $P(f,q) = 1$, indicating that none of the compared strategies could find feasible solutions for all the considered instances. The VIs with the highest impact on the gap of the solutions are those used to impose and select nondominated relief paths over dominated relief paths (VIs2), which found feasible solutions in 98.71% of the cases. The fraction of instances solved with the best gap increased from approximately 60% to 88% when the VIs2 were included. The VIs4, whose goal is to impose and select nondominated crew paths over dominated crew paths, also had a relevant impact on the gap. VIs2 and VIs4 helped to significantly reduce the number of solutions that needed to be explored by cutting off solutions with nondominated paths. VIs1 and VIs3 helped to improve the linear relaxation of the problem by setting lower bounds for the accessibility (Z_i^{d}) and restoration (Z_j^{r}) time. VIs1 had a more pronounced impact than VIs3 because the accessibility time is directly penalized in the objective function, while the restoration time of a given damaged node i does not directly affect the cost of the problem if node i is not considered in some relief path.

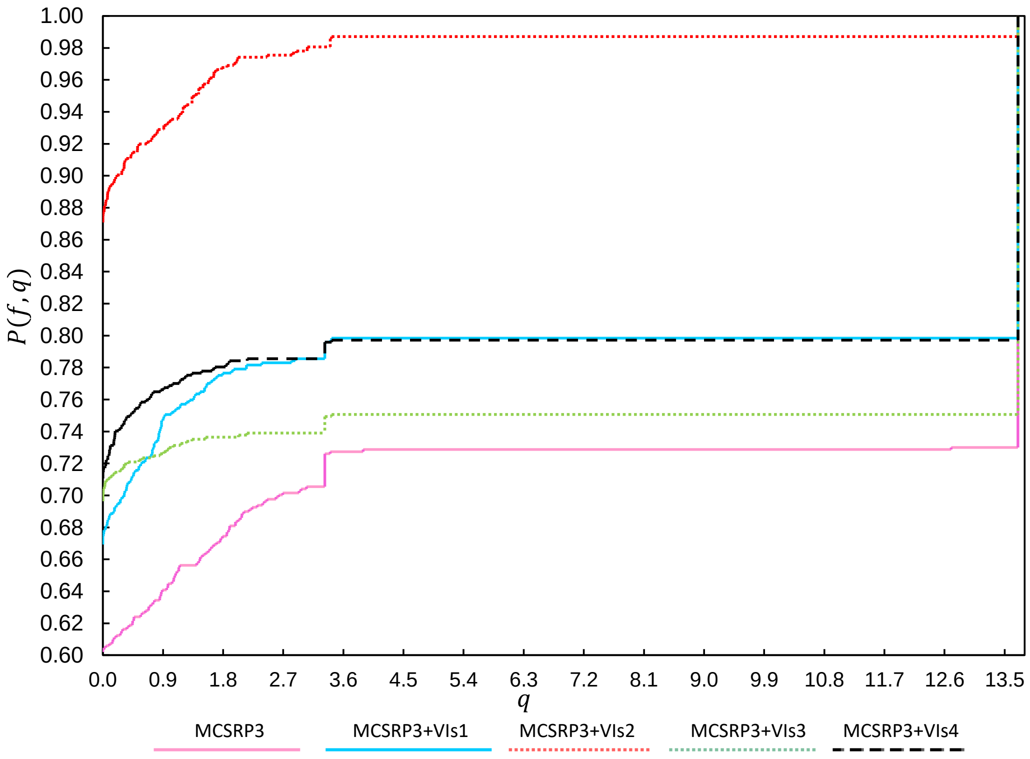


Figure 6.2: Performance profiles based on gap for the model MCSRP3 with different type of VIs.

Table 6.5 compares the average results of MCSRP3+VIs and the same strategy applying the graph reduction strategy (MCSRP3+VIs*). Additional results of the MCSRP3+VIs and MCSRP3+VIs* are presented in Tables D.3 and D.4 in D. The graph reduction strategy improved the average upper, lower bound and gap of the solutions. The average gap was reduced from 5.75% to 3.45% for L instances and from 9.70% to 4.71% for CS instances. The average elapsed time was slightly longer when the graph reduction strategy was applied because of the preposessing step required to reduce and solve the reduced graphs. On average, the instances based on the real case were harder to solve than the instances from the literature. The average time, for example, was 719 seconds with the MCSRP3+VIs* strategy for the L instances, while the average time of the strategy MSCRP3+VIs* for CS instances was 1,810 seconds. Furthermore, the average gap was smaller in the instances of set L. Moreover, on average, the instances at a higher number of crews were harder to solve.

Table 6.5: Average results of the MCSRP3+VIs and MCSRP3+VIs* strategies.

Instance	Solution method	#Crew	#Ins	#Opt	%Opt	Avg. upper bound	Avg. lower bound	Avg. gap (%)	Avg. time (seconds)
Set L	MCSRP3 + VIs	1	144	126	87.50	11,251	9,231	5.42	578.49
		3	144	117	81.25	5,562	4,420	5.47	734.53
		5	144	118	81.94	4,734	3,725	6.35	710.60
	MCSRP3 + VIs*	1	144	126	87.50	11,104	9,810	3.68	633.61
		3	144	129	89.58	5,321	4,756	3.11	761.51
		5	144	128	88.89	4,390	3,977	3.54	763.07
Set CS	MCSRP3 + VIs	1	114	85	74.56	127,720	115,219	5.26	1,247.62
		3	114	63	55.26	80,824	64,543	10.78	1,884.45
		5	114	60	52.63	72,640	59,176	13.05	2,024.23
	MCSRP3 + VIs*	1	114	86	75.44	126,348	116,578	4.16	1,396.17
		3	114	84	73.68	73,970	69,258	4.18	1,950.79
		5	114	71	62.28	69,881	63,952	5.79	2,083.51

Finally, Figure 6.3 presents the percentage of CS instances solved to optimality and the average elapsed times for different values of β. β is the factor by which the distance of a relief path $0-i$ can increase in relation to its shortest distance $dist_{0i}$. Then, $l_i^{\mathsf{d}} = (1+\beta)\cdot dist_{0i}$, and thus larger β values imply larger maximum distances l_i^{d} in the relief paths. As the results indicate, the larger the β's, the easier to solve the corresponding instance. In fact, when the constraints related to the maximum distances are relaxed ($\beta = \infty$), all instances are solved to optimality and the average elapsed time decreases more than 88% in relation to the case with $\beta = 5$. Basically, by allowing larger values for l_i^{d}, it is rather straightforward to find non-dominated relief paths, which increases the effectiveness of the inequalities proposed in Property 1. Also, with larger l_i^{d} values, the non-dominated relief paths tend to use fewer damaged nodes, thus reducing the number of repaired nodes and, consequently, simplifying the crew scheduling and routing decisions. In general, MCSRP3 + VIs* is able to return good-quality solutions within 1 hour of time limit for most of the practical instances, which is a reasonable time considering that the multicrew approach can significantly reduce the accessibility time of the demand nodes, as shown in the next section.

6.4.3 Practical Relevance: Road restoration in the Megadisaster of Rio de Janeiro in 2011

We now analyze our case study based on the so-called megadisaster of the Serrana Region in Rio de Janeiro, Brazil. This event that occurred in 2011 was characterized by heavy rain, floods, and landslides, compromising water, electricity, and transportation infrastructure systems. It

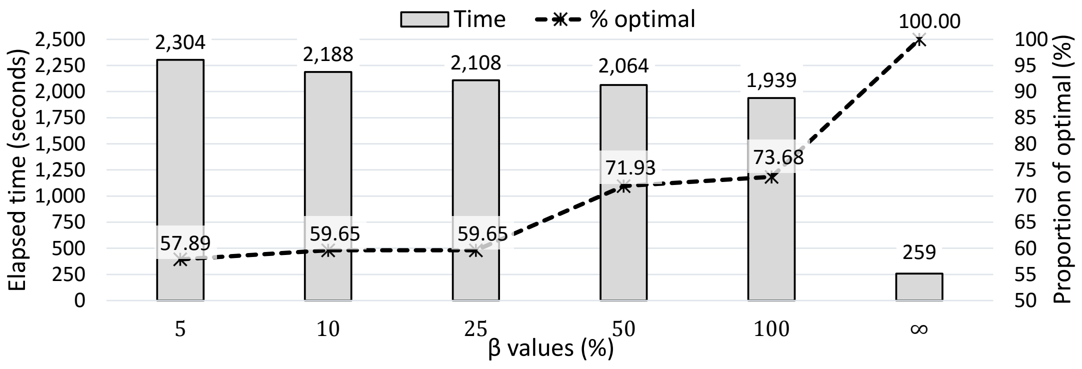


Figure 6.3: Proportion of optimal instances and average elapsed times for different values of β.

claimed hundreds of lives and affected thousands of people (Rio de Janeiro, 2011). Figure 6.4 shows the main cities and highways affected by the disaster, according to the Legislative Assembly of Rio de Janeiro (Rio de Janeiro, 2011). Figure 6.4 reveals that some cities (PE, TE, AR, TR, SA, SU, SJ) can be connected to the depot without using damaged nodes. For these cities, the proposed formulations found the optimal relief paths whose accessibility time was zero. For the other cities (NF, CO, BJ, MA, SS, SM), at least one damaged node have to be used to define the relief paths.

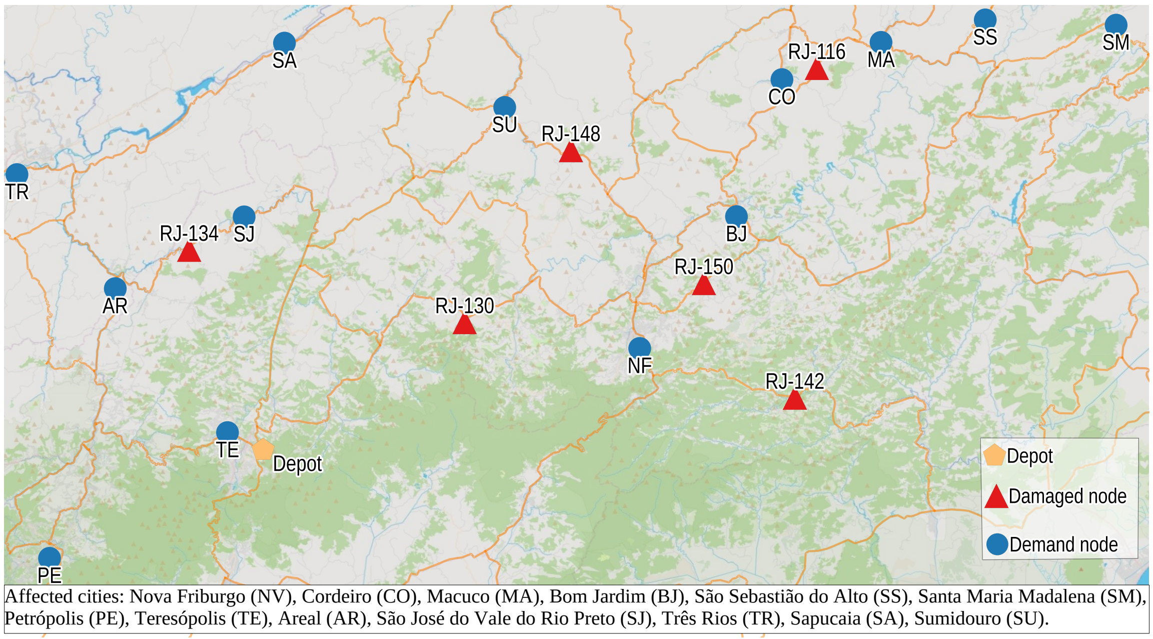

SA
SU
RJ-148
RJ-116
CO
MA
SS
SM
TR
RJ-134
SJ
BJ
RJ-150
AR
RJ-130
NF
RJ-142
TE
Depot
PE
Depot
Damaged node
Demand node
Affected cities: Nova Friburgo (NV), Cordeiro (CO), Macuco (MA), Bom Jardim (BJ), São Sebastião do Alto (SS), Santa Maria Madalena (SM), Petrópolis (PE), Teresópolis (TE), Areal (AR), São José do Vale do Rio Preto (SJ), Três Rios (TR), Sapucaia (SA), Sumidouro (SU).

Figure 6.4: Main cities and highways affected by the disaster.

A total of 342 CS instances were derived from the real case disaster by considering different number of damaged nodes, crews, and β values, as described in B. The average results of the CS instances are presented in Table 6.6. This table shows the total cost; the proportion of nodes repaired (% rep); the proportion of the repaired nodes that are used only in the relief paths (% rep relief paths), i.e., repaired nodes not used in the middle of crew paths; the proportion of required crews (% crew used); the proportion of demand nodes that need at least one damaged node to become accessible (% demand nodes); the best-case, the worst-case and the average accessibility time between demand nodes (best, worst, mean); the total accessibility time of the demand nodes (total); the difference between the worst-case and the best-case accessibility time (range = worst - best); and the average distance of the relief paths in kilometers.

The average proportion of nodes repaired was 42.85%. Thus, not all damaged nodes must be repaired to recover the accessibility of the network. Repaired damaged nodes are used in relief paths and/or crew paths. On average, 87.77% of the repaired damaged nodes were used in the relief paths only. The other repaired nodes were used in both the crew paths and the relief paths. As expected, the problem prioritizes the restoration of the damaged nodes in the relief paths. The proportion of repaired damaged nodes is not significantly affected by the number of crews. In fact, although the average proportion of repaired damaged nodes increases by 2.13% from 42.33 with 1 crew to 43.23 with 5 crews, for some instances the number of repaired damaged nodes decreases. This is the case of instance with $\beta = 5$ in class CS0, represented by the network given in Figure 6.4 and with optimal schedule shown in Figure 6.5.

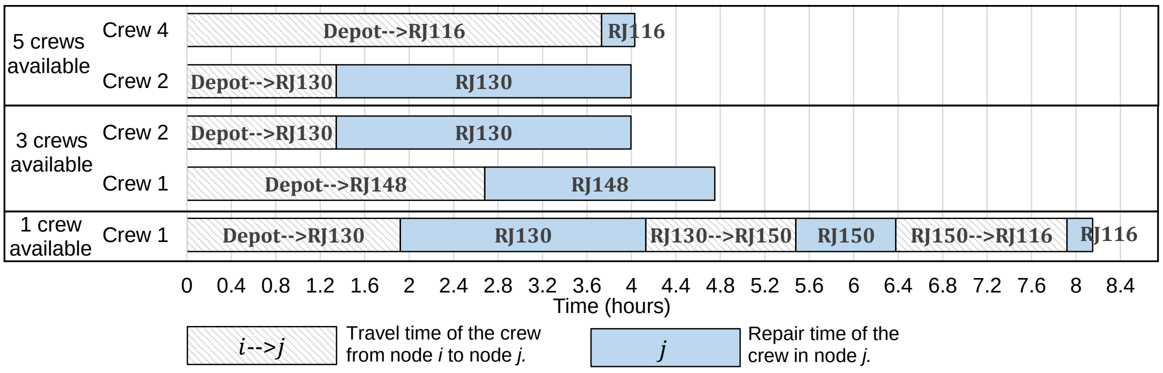


Figure 6.5: Scheduling of the crews.

Note in Figure 6.5 that with one crew, three damaged nodes (RJ-130, RJ-116, RJ-150) were repaired to restate the accessibility of the network after 8.15 hours. With three crews, the total time to restate the accessibility of the network decreased to 4.75 (41.71%) hours. Only two damaged nodes (RJ-130, RJ-148) were repaired in this case. With five crews, the solution indicates the restoration of two damaged nodes (RJ-130, RJ-116), and the time to restate the accessibility of the network was 4.03 hours, a reduction of 15.15% in relation to the case with 3 crews. A trade-off can be observed between increasing the number of crews, which may have a logistic cost in practice, and reducing the time to restore the accessibility of the demand nodes. However, at some point, increasing the number of crews may not significantly affect the accessibility time of the demand nodes. For the instance with $\beta = 5\%$ in class CS0, for example, no significant improvement was observed when five additional crews with the same characteristics than the first five crews were considered. Consequently, when the number of crews increases, more crews can become idle. Note in Table 6.6 that the utilization of the crews decrease from 100% with one crew to 64.20% with 5 crews.

Insight 1. *There is a remarkable trend in avoiding the damaged nodes not only in the relief*

Table 6.6: Average results for different number of crews, damaged nodes, and β values.

Damaged		β	Total		% rep re-	% crew	% demand	Accessibility time (hours)[2]					Dist. relief
nodes	# crew	(%)	cost	% rep	lief paths[1]	used[1]	nodes	Best	Worst	Mean	Total	Range	paths (km)
6	1	5	151,737	73.81	91.43	100.00	80.00	2.93	17.23	9.97	136.40	14.30	86.54
		10	150,029	69.05	91.43	100.00	76.70	2.93	16.53	9.86	134.39	13.60	86.67
		25	135,733	64.29	94.29	100.00	72.36	2.87	16.18	7.60	93.95	13.30	91.03
		50	49,487	40.48	88.89	100.00	52.80	2.23	7.20	3.65	37.43	4.97	100.92
		100	44,736	40.48	88.89	100.00	42.36	2.23	7.20	3.26	25.62	4.97	159.87
		∞	0.00	0.00	NA	NA	0.00	0.00	NA	NA	NA	NA	288.31
	3	5	88,574	71.43	90.71	90.48	80.00	1.99	7.40	4.66	61.50	5.41	86.07
		10	88,279	69.05	87.86	85.71	76.70	1.99	7.29	4.63	59.57	5.31	86.49
		25	80,576	66.67	90.00	85.71	72.36	1.92	7.51	4.17	50.24	5.59	90.71
		50	35,647	40.48	94.44	77.78	52.80	1.06	3.47	2.26	24.28	2.41	101.37
		100	33,146	40.48	94.44	77.78	42.36	1.71	3.52	2.25	17.75	1.81	164.84
		∞	0.00	0.00	NA	NA	0.00	0.00	NA	NA	NA	NA	288.31
	5	5	83,994	71.43	90.71	68.57	80.00	1.92	6.24	4.15	53.38	4.32	86.09
		10	83,715	69.05	90.71	62.86	76.70	1.92	6.24	4.15	51.66	4.32	86.47
		25	74,876	66.67	92.86	60.00	72.36	1.92	6.14	3.68	42.90	4.22	90.55
		50	35,413	42.86	83.33	53.33	52.80	1.06	3.47	2.21	23.87	2.41	101.24
		100	32,748	40.48	94.44	53.33	42.36	1.71	3.47	2.23	17.58	1.77	165.63
		∞	0.00	0.00	NA	NA	0.00	0.00	NA	NA	NA	NA	288.31
10	1	5	231,130	63.33	81.49	100.00	89.87	4.51	28.36	15.44	231.52	23.85	87.44
		10	226,349	61.67	81.19	100.00	89.87	4.51	28.40	14.53	217.88	23.89	88.66
		25	207,713	53.33	82.22	100.00	89.87	3.15	24.21	10.84	157.17	21.06	91.54
		50	92,286	38.33	91.67	100.00	67.50	2.77	12.36	6.53	70.65	9.59	104.66
		100	82,477	35.00	87.50	100.00	55.32	2.77	11.15	5.52	46.99	8.38	168.40
		∞	16,058	6.67	100.00	100.00	35.45	1.23	1.23	1.23	7.45	0.00	379.33
	3	5	125,001	63.33	81.09	100.00	89.87	2.06	11.02	6.34	93.67	8.96	87.58
		10	123,009	61.67	80.79	100.00	89.87	2.06	11.02	5.87	87.84	8.96	87.89
		25	113,911	55.00	76.11	100.00	89.87	2.06	9.22	4.99	74.59	7.15	91.50
		50	58,984	41.67	80.00	83.33	67.50	1.95	6.01	3.49	38.90	4.07	104.53
		100	52,608	35.00	84.72	77.78	55.32	1.95	5.70	3.29	27.57	3.75	169.99
		∞	7,152	6.67	100.00	33.33	35.45	0.92	0.92	0.92	3.64	0.00	381.81
	5	5	117,847	63.33	81.09	90.00	89.87	2.02	8.78	5.51	81.02	6.76	87.03
		10	115,892	61.67	86.35	86.67	89.87	2.01	8.80	5.09	75.45	6.79	87.95
		25	105,184	56.67	82.22	83.33	89.87	1.97	7.83	4.33	63.42	5.86	91.28
		50	58,061	41.67	75.83	63.33	67.50	1.92	5.07	3.32	37.16	3.15	104.44
		100	51,917	35.00	70.83	56.67	55.32	1.92	4.55	3.17	26.33	2.63	175.49
		∞	7,152	6.67	100.00	20.00	35.45	0.92	0.92	0.92	3.64	0.00	369.50
14	1	5	259,270	57.14	81.35	100.00	90.71	3.82	34.96	18.81	281.70	31.14	86.91
		10	255,115	55.95	87.37	100.00	90.71	3.82	35.36	17.69	265.60	31.54	88.73
		25	227,120	40.48	83.33	100.00	89.87	3.87	25.34	12.25	173.00	21.47	94.79
		50	87,220	29.76	86.11	100.00	74.68	2.02	12.49	6.12	76.06	10.46	104.62
		100	79,207	27.38	91.67	100.00	64.17	2.02	11.45	5.46	56.55	9.43	144.62
		∞	16,317	4.76	100.00	100.00	35.45	1.24	1.24	1.24	7.55	0.00	417.68
	3	5	152,538	59.52	78.41	100.00	90.71	2.39	13.82	8.12	119.36	11.43	87.09
		10	146,287	57.14	83.50	100.00	90.71	2.60	14.36	7.80	115.61	11.75	88.03
		25	129,439	42.86	86.35	100.00	89.87	1.69	10.92	6.00	87.77	9.23	93.30
		50	57,273	30.95	91.67	88.89	74.68	1.52	6.48	3.81	48.86	4.96	105.80
		100	51,413	27.38	87.50	83.33	64.17	1.52	6.60	3.49	36.96	5.08	149.84
		∞	7,255	4.76	100.00	33.33	35.45	0.92	0.92	0.92	3.68	0.00	393.31
	5	5	137,891	58.33	83.37	90.00	90.71	1.99	12.82	6.95	102.79	10.83	87.11
		10	136,825	55.95	82.90	86.67	90.71	1.99	11.69	6.69	99.29	9.71	87.85
		25	117,443	45.24	83.97	83.33	89.87	1.64	10.71	5.21	75.74	9.06	93.31
		50	56,273	30.95	85.56	63.33	74.68	1.52	5.41	3.59	46.30	3.89	104.35
		100	49,479	27.38	95.83	50.00	64.17	1.52	5.57	3.17	33.62	4.05	146.69
		∞	7,255	4.76	100.00	20.00	35.45	0.92	0.92	0.92	3.68	0.00	434.01
Average per number of crews		1	128,444	42.33	88.75	100.00	66.54	2.88	17.11	8.82	118.82	14.23	148.37
		3	75,061	43.00	87.51	83.38	66.54	1.78	7.42	4.30	55.99	5.64	147.69
		5	70,665	43.23	87.06	64.20	66.54	1.70	6.39	3.84	49.29	4.69	149.29
Average per β (%) values		5	149,776	64.63	84.41	93.23	86.86	2.62	15.63	8.89	129.04	13.00	86.87
		10	147,278	62.35	85.79	91.32	85.76	2.65	15.52	8.48	123.03	12.87	87.64
		25	132,444	54.58	85.71	90.26	84.04	2.34	13.12	6.56	90.98	10.77	92.00
		50	58,961	37.46	86.39	81.11	64.99	1.78	6.88	3.89	44.83	5.10	103.55
		100	53,081	34.29	88.43	77.65	53.95	1.93	6.58	3.54	32.11	4.65	160.60
		∞	6,799	3.81	100.00	51.11	23.63	1.03	1.03	1.03	4.94	0.00	360.06
Average all			91,390	42.85	87.77	82.53	66.54	2.12	10.31	5.65	74.70	8.19	148.45

[1] Values computed considering only the solutions with at least one repaired damaged node.
[2] Values computed considering only the demand nodes with accessibility time higher than 0.
NA: Not available.

paths but also in the route of the crews; thus, only a (usually) small number of damaged nodes end up being repaired to restore the accessibility of the network. Consequently, a further increase *in the number of crews is not necessarily followed by a relevant reduction in the accessibility time. In spite of it, if the decision-maker hires more crews, it is likely that some of them will become idle.*

More crews evidently cause a decrease in the accessibility time and, consequently, in the worst-case accessibility time. However, the impact concerning the average accessibility time was less pronounced when we have increasingly more crews, especially in networks with fewer damaged nodes. For example, the average accessibility time with 6 damaged nodes and $\beta = 5\%$ decreased from 9.97 with 1 crew to 4.66 (53.26%) with three crews, while the reduction in the average accessibility time from 3 to 5 crews was 10.94%. Additionally, our results reveal that the multiple crews have a more pronounced effect in reducing the worst-case accessibility time between the demand nodes. For the instances with 6 damaged nodes and $\beta = 5\%$, the reduction in the worst-case accessibility time was 15.68% from three to five crews, while the reduction in the best-case accessibility time was 3.52%. Figure 6.6 shows the accessibility times of the instance with $\beta = 5\%$ in class CS0 for different numbers of crews. For this particular case, the worst-case accessibility was reduced 50.55%, from 8.15 hours with 1 crew to 4.03 hours with 5 crews.

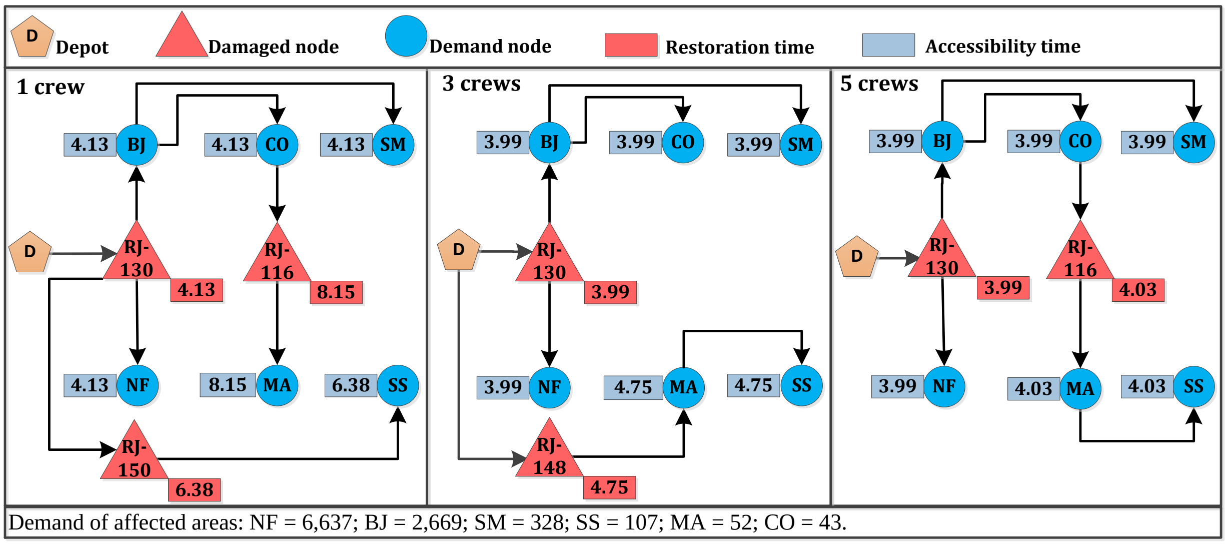


Figure 6.6: Nodes used in the relief paths to connect the depot with the demand nodes.

Insight 2. *Multiple crews help to decrease the accessibility time and to achieve more equitable accessibility times across the different demand nodes, which is a desirable feature in post-disaster settings.*

Figure 6.6 also illustrates the damaged nodes used in the relief paths to reach the demand nodes. For instance, the relief path to reach MA with one crew is defined by D→RJ-130→BJ→CO→RJ-116→MA. We did not consider intersection nodes in Figure 6.6. With one crew, MA was the last demand node to become connected, which occurred after 8.15 hours, when nodes RJ-116 and RJ-130 were repaired. SSA became connected after RJ-130 and RJ-150 were repaired while NF, CO, BJ, and SM became connected after RJ-130 was repaired. As expected, the cities with greater demand, NF and BJ, were some of the first cities to become accessible after 4.13 hours. However, some cities that have a smaller demand, such as CO and SM, also became accessible within 4.13 hours.

Insight 3. *Cities with greater demand are likely to be the first to become accessible from the depot. However, some cities with smaller demand can also became quickly accessible when their corresponding relief paths use the same damaged nodes as the relief paths associated with the cities with greater demand.*

With three crews, SS and MA became connected after RJ-148 was repaired, while NF, CO, BJ and SM became connected after RJ-130 was repaired. With five crews, SS and MA became connected after RJ-130 and RJ-116 were repaired, while NF, CO, BJ and SM became connected after RJ-130 was repaired. In the cases with 3 and 5 crews, only two crews performed the restoration of two damaged node. However, the repaired damaged nodes changed depending on the characteristics of the crews. For the considered instance, although two crews were enough to perform the restoration, the use of crew 4 instead of crew 1 reduced the total time to restate the accessibility of the network. The relief paths were also affected by the characteristics of the crews. For instance, relief path $0 - SM$ using RJ-116 is a good relief path only when crew 4 is available. Particularly, damaged node RJ-116 has a short repair time but it is far away from the depot. Crew 4, that has shorter travel times than crew 1, can perform a faster restoration of RJ-116. Consequently, RJ-116 became better than RJ-150 (used in the case with one crew) and RJ-148 (used in the case with three crews) in the relief path $0 - SM$. In general, when multiple crews are available, farther damaged nodes are usually allocated to the crews with shorter travel times to those nodes, while damaged nodes with longer repair times are allocated to crews that can perform a faster restoration, thus saving travel and restoration times, respectively. Moreover, the crews are usually allocated to repair groups of damaged nodes that are geographically close to each other, which also saves time.

Insight 4. *The heterogeneous characteristics of the crews can significantly affect the scheduling and relief paths decisions of the problem. The allocation of the crews to the damaged nodes depends on their characteristics, location of the damaged nodes, and repair times.*

Interestingly, in Figure 6.6 we can observe that SS and SM required repairing different damaged nodes to be accessible, even though such cities are geographically close to one another. The reason for this result is the maximum distance l_i^{d} allowed for the relief paths. For SS and SM, the maximum distances were $l_{SS}^{\mathrm{d}} = 156.27$ and $l_{SM}^{\mathrm{d}} = 165.41$, respectively. The feasible relief paths $0 - SS$ used one of the damaged nodes RJ-150, RJ-148 or RJ-116, while there was a relief path $0 - SM$ that did not require the use of those nodes. Such an alternative path was shorter than l_{SM}^{d} and then feasible to reach SM, but it was higher than l_{SS}^{d} and then infeasible to reach SS. If l_{SS}^{d} increases to 165, it would be possible to define a path to SS without using any of damaged nodes RJ-150, RJ-148 or RJ-116.

Insight 5. *The definition of the path to reach the demand nodes from the depot is not trivial since even nodes that can be geographically near each other could require the restoration of different nodes to become accessible. The reason for such behavior is mainly the maximum distance $l_i^{\mathbf{d}}$ imposed for the relief paths. The decision-maker should select $l_i^{\mathbf{d}}$ carefully since even small changes in this parameter for a given demand node i might lead to significantly different solutions.*

Table 6.6 reveals that the maximum distance $l_i^{\mathbf{d}}$, which is computed using different β values, can significantly affect the accessibility time and the number of repaired damaged nodes in the problem. Evidently, the accessibility time decreases for larger β values, mainly because more feasible relief paths are available. A straightforward consequence of having more feasible relief paths is the reduced number of demand nodes that need the restoration of at least one damaged node to be accessible since the additional relief paths might not require the use of such damaged nodes. For instance, the average number of demand nodes that require damaged nodes to become accessible decreases from 86.86% with $\beta = 5$ to 23.63% with $\beta = \infty$. In contrast, the average distance of the relief paths increases significantly for larger choices of β. Note, e.g., that the average distance increases more than four times, from 86.87 km when $\beta = 5\%$ to 360.06 km when $\beta = \infty$. Larger values of β imply in the restoration of fewer damaged nodes. In fact, when the maximum distance constraint is relaxed, no damaged node need to be repaired to recover the accessibility of the network in some cases. For example, damaged networks with 6 damaged nodes and $\beta = \infty$ did not require the restoration of these nodes. A clear example of this situation is the instances in class CS0 defined by the network in Figure 6.4, in which it is possible to find relief paths without requiring damaged nodes if $\beta = \infty$. However, when the number of damaged nodes increases, they become necessary to define the relief paths, even in the cases with $\beta = \infty$.

Insight 6. *In general, shorter maximum distances imply fewer feasible relief paths, which may increase the accessibility time of the demand nodes and increase the number of repaired damaged nodes. Longer maximum distances may reduce the accessibility time of the demand nodes and the number of repaired damaged nodes. However, they can lead to the selection of longer relief paths, which is undesirable in practical distribution or evacuation operations in post-disaster situations. Evidently, there is a trade-off between good accessibility times and the quality of the relief paths in terms of distance.*

Finally, Figure 6.7 illustrates the average accessibility time for different number of damaged nodes and instance classes. On average, the increase in the damaged nodes in a network of a given class increases the accessibility time of the solutions. However, such behavior can be different when we compare instances of different classes. For example, the average accessibility time for the instances of class CS4 with 10 damaged nodes was 6.76, while for the instances of class CS6 with 14 damaged nodes was 5.85 (13.46% smaller). Therefore, the increase in the accessibility time depends not only on the number of damaged nodes but also on the location of the damaged nodes. In Figure 6.4, for example, it is possible to observe some regions where fewer damaged nodes disrupting the accessibility could have a higher impact than a higher number of damaged nodes in regions where there is no demand nodes.

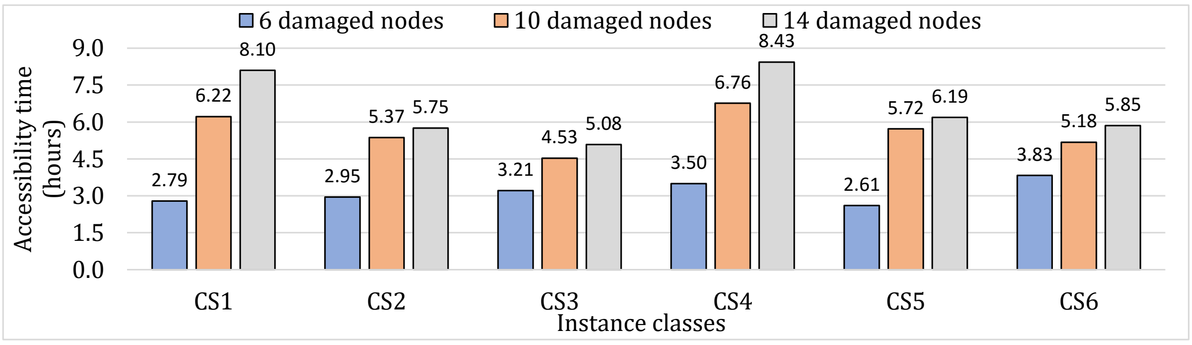


Figure 6.7: Average accessibility time for different damaged nodes and instance classes.

6.5 Final remarks of the chapter

In this chapter, we proposed three novel mathematical formulations for the multicrew scheduling and routing problem in road restoration. New valid inequalities were also developed. The first two formulations are based on the three-index and two-index formulation of the VRP. The third formulation eliminates a few variables and introduces new constraints to reduce the symmetry in the solutions of the problem. The valid inequalities are based on the dominance of the paths between nodes in the damaged network. We performed computational experiments using instances from the literature and based on a real disaster situation. The three mathematical formulations showed improvement with the addition of the VIs. The model based on the two-index VRP formulation showed the best performance for most of the instances. Furthermore, the elimination of symmetric solutions in the third formulation improved the performance of the model, especially in finding feasible solutions. The model based on the three-index VRP formulation provides optimality guarantees for a higher number of instances, but it has difficulty finding feasible solutions in some cases. The graph reduction strategy for deriving cuts from networks with fewer nodes also improves the results. Thus, the best approach was able to obtain good-quality solutions with less than 7% of the average optimality gap for the different instance classes.

The analysis of the practical case showed that, as expected, the use of more crews to solve the problem significantly reduces the time required to make the demand nodes accessible. However, the impact concerning the average accessibility time was less pronounced when we had increasingly more crews, especially in the networks with fewer damaged nodes. The multiple crews mainly affect the worst-case accessibility time between the demand nodes, thus providing more equitable accessibility times. Usually, the farthest damaged nodes were allocated to the crews that had the shortest travel time, while the damaged nodes with higher repair time were allocated to crews that can perform a faster restoration. The restoration of damaged nodes in the relief paths was prioritized over nodes in the crew paths. A few additional damaged nodes are repaired only if they are strictly necessary in the path of the crews to reach other damaged nodes. Furthermore, on average, fewer than 55% of the damaged nodes were required to restore the accessibility of the network. Such a proportion decreased when paths with higher distances were allowed to connect the depot with the demand nodes, but it is not significantly affected by the increase in the number of available crews.

Since the information about the actual situation of the damaged nodes after the extreme events is limited, and the consequences of the extreme events over the transportation networks cannot be accurately predicted, in the next chapter we consider the inherent uncertainties present in the input data.

Chapter 7

The Robust Crew Scheduling and Routing Problem

This chapter introduces the Robust Crew Scheduling and Routing Problem (RCSRP) in road restoration, which incorporates uncertain repair times via robust optimization. Also, the RC-SRP considers a new objective function based on latency. A mathematical formulation and a Benders decomposition based algorithm are developed for the problem. This chapter is organized as follows. Section 7.1 introduces the problem. Section 7.2 presents the novel RCSRP model. Section 7.3 develops the logic-based Benders decomposition. Section 7.4 discusses the computational results. Finally, Section 7.5 presents concluding remarks.

7.1 Introduction

In determining the repair operation decisions, it is important to consider the fact that the time required to repair the damaged nodes depends on the severity of the impact of the extreme events. For instance, the time to repair roads blocked by debris depends on the amount of debris, which may be hard to estimate. In the 2011 floods in the Serrana Region of Rio de Janeiro in Brazil, for example, different points of six main highways were affected (Rio de Janeiro, 2011), but no information about the time required to repair such points was available. Generally, repair times are unknown in those situations (Çelik et al., 2015) and are critical factors that need to be considered in the road restoration, mainly when the short-term response operations must be performed as soon as possible. However, since the introduction of the CSRP, authors have proposed several models and solution methods for the deterministic version of this problem (Maya-Duque et al., 2016; Kim et al., 2018; Shin et al., 2019), but no model or solution method has been proposed considering the inherent uncertainty in the repair times.

In this chapter, we introduce the Robust Crew Scheduling and Routing Problem (RCSRP) in road restoration, in which uncertain repair times belong to a convex set $\mathcal{U}$ known as the uncertainty set in Robust Optimization (RO) terminology. The goal in the RCSRP is to find the best solution that satisfies every realization of the uncertain parameters that belongs to $\mathcal{U}$. We consider the budgeted uncertainty set proposed by Bertsimas and Sim (2004), and develop a compact formulation and a logic-based Benders decomposition for the RCSRP. Logic–based Benders decomposition (Hooker and Ottosson, 2003) is an extension of the classical Benders decomposition method, where the generation of the Benders cuts are not necessarily based on solving the dual linear programs of the subproblems (Tran et al., 2016; Perez et al., 2019). The master problem obtained from the Benders decomposition is solved by a single search tree, exploring the generation of cuts inside the tree during the branch–and–bound process. This strategy has been recently referred to as Branch-and-Check algorithm (B&Ch) (Tran et al., 2016; Perez et al., 2019). It is important to highlight that the BBC approaches developed in Chapters 4 and 5 can be also considered as B&Ch algorithms. We also adapt the metaheuristic algorithms proposed for the SCSRP in Section 5.3 to find feasible solutions for the RCSRP and use the solutions from the metaheuristic to warm-start the B&Ch approach. Additionally, we introduce a new objective called "latency" for the problem, which is defined as the time at which a demand node is reached from the central depot. Therefore, the latency considers, for a given demand node, the time at which the demand node becomes accessible from the depot plus the travel time on the relief path connecting the depot with the demand node. Although the latency has been recently proposed for other variant of road restoration problems (Ajam et al., 2019), this objective has not yet been considered in the CSRP. We apply our proposed approaches to the case of floods and landslides in Brazil.

7.2 Mathematical formulation

We assume that the uncertain repair times for a given crew k belong to a convex set $\mathcal{U}_k$ known as the uncertainty set. The goal in the RCSRP is to find the best solution that satisfies every realization of the uncertain parameters that belongs to $\mathcal{U}_k$. We consider the budgeted uncertainty set proposed by Bertsimas and Sim (2004), as this set provide robust counterparts as tractable as their original deterministic formulations (Bertsimas and Sim, 2003; Alem et al., 2018; Munari et al., 2019).

Let $\widetilde{\delta}_{ki} \in [\delta_{ki} - \hat{\delta}_{ki}, \delta_{ki} + \hat{\delta}_{ki}]$ be rewritten as $\widetilde{\delta}_{ki} = \delta_{ki} + \hat{\delta}_{ki}\xi_{ik}$, where ξ_{ik} is a random variable that assumes values in the interval $[-1, 1]$. Let $\mathcal{B}_k \subseteq \mathcal{V}^{\mathfrak{r}}$ be the nodes considered in the schedule of crew k. For a given crew k, we represent the uncertainty set $\mathcal{U}_k$ as follows:

$$\mathcal{U}_k(\Gamma) = \left\{ \boldsymbol{\delta}_k \in \mathbb{R}_+^{|\mathcal{V}^{\mathfrak{r}}|} | \widetilde{\delta}_{ki} = \delta_{ki} + \hat{\delta}_{ki}\xi_{ik}, \sum_{i \in \mathcal{B}_k} \xi_{ik} \leq \Gamma, 0 \leq \xi_{ik} \leq 1, i \in \mathcal{B}_k \right\}, \tag{7.1}$$

where the cumulative uncertainty of the random variable is bounded by its budget of uncertainty Γ. If $\Gamma = 0$, then the uncertainties are not taken into account, i.e., no repair time assume its worst case deviation. On the other hand, larger budgets of uncertainty indicate more repair times assuming its worst-case deviation and express more conservative/robust solutions.

The structure of the uncertainty set (7.1) allows us to calculate the restoration time of damaged nodes in a route $\mathcal{B}_k$ using recursive equations, similar to the one proposed for the robust vehicle routing problem (Munari et al., 2019). Let $\mathcal{B}_k = (v_0, v_1, \cdots, v_j, \cdots, v_n)$ be the schedule of crew k, where v_j is the jth damaged node repaired by crew k and $v_0 = 0$. We call crew path $v_{(j-1)} - v_j$ the sequence of nodes and arcs used by the crew to travel from $v_{(j-1)}$ to v_j. Let $\phi_j = 1$, if some damaged node is used in the crew path $v_{(j-1)} - v_j$ and $\phi_j = 0$, otherwise. Let $Z^{\mathfrak{r}}_{v_j\gamma}$ be the restoration time of damaged node $v_j \in \mathcal{B}_k$ when up to $\gamma \leq \Gamma$ repair times reach their worst case. If $\phi_j = 0$, $Z^{\mathfrak{r}}_{v_j\gamma}$ can be computed by the recursion:

$$Z^{\mathfrak{r}}_{v_j\gamma} = \begin{cases} 0, & \text{if } j = 0; \\ Z^{\mathfrak{r}}_{v_{(j-1)}\gamma} + \delta_{kv_j} + \rho_{kv_{(j-1)}v_j}, & \text{if } j \geq 1, \gamma = 0; \\ \max\{Z^{\mathfrak{r}}_{v_{(j-1)}\gamma} + \delta_{kv_j} + \rho_{kv_{(l-1)}v_l}, & \\ \qquad Z^{\mathfrak{r}}_{v_{(l-1)}(\gamma-1)} + \delta_{kv_j} + \hat{\delta}_{kv_j} + \rho_{kv_{(j-1)}v_j}\}, & \text{if } j \geq 1, \gamma \geq 1; \end{cases} \tag{7.2}$$

where $\rho_{kv_{(j-1)}v_j}$ is the shortest travel time of crew k between nodes $v_{(j-1)}$ and v_j without using damaged nodes. Equations (7.2) define the restoration time when no damaged nodes are visited in crew path $v_{(j-1)} - v_j$. In this case, for a given node v_j repaired by crew k, the restoration time $Z^{\mathfrak{r}}_{v_j\gamma}$ is the sum of three components: the restoration time of the predecessor node $v_{(j-1)}$, the travel time in the crew path $v_{(j-1)} - v_j$, and the repair time of node j. Regarding the repair time, the basic idea in recursive equation (7.2) is to check for each damaged node v_j if the repair time $\widetilde{\delta}_{kv_j}$ must be one of the γ values assuming the worst case or the γ worst cases must be considered in the repair times of the damaged nodes repaired before v_j.

If $\phi_j = 1$, at least one damaged node is used in the crew path $v_{(j-1)} - v_j$. Let $\mathcal{L}_j = (w_1, \cdots, w_h, \cdots, w_H)$ be the set of damaged nodes used by crew k to travel from $v_{(j-1)}$ to v_j, where $h \leq H$ is the position at which node w_h is visited in the path to node v_j, and H is the number of damaged nodes visited in the path to node v_j. Hence, crew k must travel between consecutive nodes $v_{(j-1)}, v_j$ in its schedule using undamaged nodes, nodes repaired by other crew $k' \neq k$ or nodes repaired by crew k before $v_{(j-1)}$. Crew k may have to wait for the restoration of some damaged nodes in set $\mathcal{L}_j$. Therefore, let $T^{\mathrm{s}}_{hv_j\gamma}$ be the exact time at which the crew k arrives at damaged node w_h in the path to node v_j when up to γ repair times reach their worst-case; and $T^{\mathrm{w}}_{hv_j\gamma}$ be the waiting time at the damaged node w_h in the path to node v_j when up to γ repair times reach their worst-case. If $\phi_j = 1$, $Z^{\mathrm{r}}_{v_j\gamma}$ can be computed by the recursion:

$$Z^{\mathrm{r}}_{v_j\gamma} = \begin{cases} Z^{\mathrm{r}}_{v_{(j-1)}\gamma} + \delta_{kv_j} + \rho_{kw_Hv_j} + T^{\mathrm{w}}_{Hv_j\gamma} + T^{\mathrm{s}}_{Hv_j\gamma}, & \text{if } j \geq 1, \gamma = 0; \\ \max\{Z^{\mathrm{r}}_{v_{(j-1)}\gamma} + \delta_{kv_j} + \rho_{kw_Hv_j} + T^{\mathrm{w}}_{Hv_j\gamma} + T^{\mathrm{s}}_{Hv_j\gamma}, & \\ \quad Z^{\mathrm{r}}_{v_{(j-1)}(\gamma-1)} + \delta_{kv_j} + \hat{\delta}_{kv_j} + \rho_{kw_Hv_j} + T^{\mathrm{w}}_{Hv_j(\gamma-1)} + T^{\mathrm{s}}_{Hv_j(\gamma-1)}\}, & \text{if } j \geq 1, \gamma \geq 1; \end{cases} \tag{7.3}$$

where

$$T^{\mathrm{s}}_{hv_j\gamma} = \begin{cases} T^{\mathrm{w}}_{(h-1)v_j\gamma} + T^{\mathrm{s}}_{(h-1)v_j\gamma} + \rho_{kw_{(h-1)}w_h}, & \text{if } h \geq 2, \gamma \geq 0; \\ Z^{\mathrm{r}}_{v_{(j-1)}\gamma} + \rho_{kv_{(j-1)}w_h}, & \text{if } h = 1, \gamma \geq 0; \end{cases} \tag{7.4}$$

$$T^{\mathrm{w}}_{hv_j\gamma} = \max\{0, Z^{\mathrm{r}}_{w_h\Gamma} - T^{\mathrm{s}}_{hv_j\gamma}\}, \forall\, h \geq 1, \gamma \geq 0. \tag{7.5}$$

Equations (7.3) define the restoration time when there is some damaged nodes visited in the crew path $v_{(j-1)} - v_j$. In this case, for a given node v_j repaired by crew k, the restoration time is the sum of the following components: the time when the crew departs from the last damaged node w_H visited in the path ($T^{\mathrm{w}}_{Hv_j\gamma} + T^{\mathrm{s}}_{Hj\gamma}$), the shortest travel time from node w_H to node v_j without using damaged nodes, and the repair time of node v_j. The arrival time of the crew at nodes visited in the crew path $v_{(j-1)} - v_j$ is computed with equations (7.4) while the waiting time of the crew in a given damaged node w_h visited in the crew path $v_{(j-1)} - v_j$ is calculated with equation (7.5). In this case, the waiting time in a damaged node w_h is calculated as the difference between the restoration time of the damaged node w_h and the arrival time of crew k at damaged node w_h.

Based on model MCSRP3 previously proposed for the MCSRP in Chapter 6 and the recursive equations (7.2), (7.3), we propose a mathematical formulation for the RCSRP. The sets, parameters and variables used in the model are formally defined as follows.

Sets

$\mathcal{V}$	All nodes.
$\mathcal{V}^{\mathrm{r}} \subset \mathcal{V}$	Damaged nodes.
$\mathcal{V}^{\mathrm{r}}_0 = \mathcal{V}^{\mathrm{r}} \cup \{0\}$	Damaged nodes including the source node 0 (depot).
$\mathcal{V}^{\mathrm{u}} \subset \mathcal{V}$	Undamaged nodes ($\mathcal{V}^{\mathrm{u}} = \mathcal{V}/\mathcal{V}^{\mathrm{r}}$).
$\mathcal{V}^{\mathrm{d}} \subset \mathcal{V}^{\mathrm{u}}$	Demand nodes.
$\mathcal{E}$	Arcs.

$\mathcal{E}_i \subseteq \mathcal{E}$	Arcs incident to node $i \in \mathcal{V}$.
$\mathcal{R} = \{1, \cdots, \lvert\mathcal{V}^{\mathrm{r}}\rvert\}$	Positions at which an already repaired damaged node can be visited in a path between two damaged nodes.
$\mathcal{K}$	Available crews.
$\mathcal{K}_i \subseteq \mathcal{K}$	Crews able to repair the damaged node $i \in \mathcal{V}^{\mathrm{r}}$.

Parameters

d_i	Demand of node $i \in \mathcal{V}^{\mathrm{d}}$.
t_e	Travel time associated to the arcs used in the relief paths.
δ_{ki}	Nominal repair time of crew $k \in \mathcal{K}_i$ at node $i \in \mathcal{V}^{\mathrm{r}}$.
$\hat{\delta}_{ki}$	Deviation of the repair time of crew $k \in \mathcal{K}_i$ at node $i \in \mathcal{V}^{\mathrm{r}}$ from its nominal value.
τ_{ke}	Travel time of crew $k \in \mathcal{K}$ on arc $e \in \mathcal{E}$.
ρ_{kij}	Shortest travel time of crew k between nodes $i \in \mathcal{V}$ and $j \in \mathcal{V}$ without using damaged nodes.
ℓ_e	Length of arc $e \in \mathcal{E}$.
l_i^{d}	Maximum distance allowed between node 0 and demand node $i \in \mathcal{V}^{\mathrm{d}}$.
M	Sufficiently large number.

Decision variables

Q_i	Binary variable equal to 1 if node $i \in \mathcal{V}^{\mathrm{r}}$ is repaired.
W_{ik}	Binary variable equal to 1 if node $i \in \mathcal{V}^{\mathrm{r}}$ is repaired by crew k.
X_{ij}	Binary variable equal to 1 if node $j \in \mathcal{V}_0^{\mathrm{r}}$ is repaired immediately after node $i \in \mathcal{V}_0^{\mathrm{r}}$.
P_{eij}	Binary variable equal to 1 if arc $e \in \mathcal{E}$ is used either in the path from node $i \in \mathcal{V}_0^{\mathrm{r}}$ to node $j \in \mathcal{V}^{\mathrm{r}}$ or in the path from node $j \in \mathcal{V}^{\mathrm{r}}$ to node $i \in \mathcal{V}^{\mathrm{r}}$ with $i < j$.
N_{lij}	Binary variable equal to 1 if node $l \in \mathcal{V}^{\mathrm{r}}$ is used either in the path from node $i \in \mathcal{V}_0^{\mathrm{r}}$ to node $j \in \mathcal{V}^{\mathrm{r}}$ or in the path from node $j \in \mathcal{V}^{\mathrm{r}}$ to node $i \in \mathcal{V}^{\mathrm{r}}$ with $i < j$.
R_{lhj}	Binary variable equal to 1 if node $l \in \mathcal{V}^{\mathrm{r}}$ is the hth damaged node visited in the path to node $j \in \mathcal{V}^{\mathrm{r}}$.
Y_{ej}	Binary variable equal to 1 if arc $e \in \mathcal{E}$ is used in the path from node 0 to node $j \in \mathcal{V}^{\mathrm{d}}$.
V_{lj}	Binary variable equal to 1 if node $l \in \mathcal{V}$ is used in the path from node 0 to node $j \in \mathcal{V}^{\mathrm{d}}$.
$T_{lhj\gamma}^{\mathrm{s}}$	Time at which the damaged node $l \in \mathcal{V}^{\mathrm{r}}$ in the position $h \in \mathcal{R}$ is visited in the path to node $j \in \mathcal{V}^{\mathrm{r}}$ (arrival time) when up to γ repair times reach their worst case.
$T_{lhj\gamma}^{\mathrm{w}}$	Waiting time at the damaged node $l \in \mathcal{V}^{\mathrm{r}}$ visited in the position $h \in \mathcal{R}$ in the path to node $j \in \mathcal{V}^{\mathrm{r}}$ when up to γ repair times reach their worst case.
$Z_{i\gamma}^{\mathrm{r}}$	Restoration time of damaged node $i \in \mathcal{V}_0^{\mathrm{r}}$ when up to γ repair times reach their worst case.
Z_i^{d}	Accessibility time of demand node $i \in \mathcal{V}^{\mathrm{d}}$.

The model for the RCSRP is formulated as follows:

$$\min \sum_{i\in\mathcal{V}^{\text{d}}} d_i(Z_i^{\text{d}} + \sum_{e\in\mathcal{E}} t_e Y_{ei}), \tag{7.6}$$

s.t.

`Relief paths constraints` (7.7) − (7.11),

`Linearization of the recursive equations` (7.12) − (7.20),

`Assignment and scheduling constraints` (7.21) − (7.28),

`Routing constraints` (7.29) − (7.33),

`Synchronization constraints` (7.34) − (7.39),

`Domain of the decision variables` (7.40) − (7.49).

The objective function (7.6) minimizes the sum of the latency of the demand nodes weighted by its demand. The latency for a demand node i is defined as the accessibility time Z_i^{d} plus the travel time $\sum_{e\in\mathcal{E}} t_e Y_{ei}$ in the relief path $0-i$. Note that the accessibility time objective, more common in the CSRP literature, can be obtained by removing the expression $\sum_{e\in\mathcal{E}} t_e Y_{ei}$ from the objective function (7.6).

Relief path constraints: Constraints (7.7) define the accessibility time for the demand nodes. Constraints (7.8)-(7.10) define the relief paths between the depot and the demand nodes. For a given relief path $0-i$, there is one arc incident to node 0 (constraints (7.8)), one arc incident to node i (constraint (7.9)) and two arcs incident for each node l in the middle of this path (constraints (7.10)). Constraints (7.11) prohibit the use of relief paths whose total distance is greater than the maximum distance allowed.

$$Z_i^{\text{d}} \geq Z_{j\Gamma}^{\text{r}} - M(1 - V_{ji}),\ \forall\ i \in \mathcal{V}^{\text{d}},\ j \in \mathcal{V}^{\text{r}}, \tag{7.7}$$

$$\sum_{e\in\mathcal{E}_0} Y_{ei} = 1,\ \forall\ i \in \mathcal{V}^{\text{d}}, \tag{7.8}$$

$$\sum_{e\in\mathcal{E}_i} Y_{ei} = 1,\ \forall\ i \in \mathcal{V}^{\text{d}}, \tag{7.9}$$

$$\sum_{e\in\mathcal{E}_l} Y_{ei} = 2V_{li},\ \forall\ j \in \mathcal{V}^{\text{d}},\ l \in \mathcal{V} \setminus \{0,\ i\}, \tag{7.10}$$

$$\sum_{e\in\mathcal{E}} \ell_e Y_{ei} \leq l_i^{\text{d}},\ \forall\ i \in \mathcal{V}^{\text{d}}. \tag{7.11}$$

Linearization of the recursive equations: Constraints (7.12)-(7.15) define the restoration time for different γ values when no damaged nodes are visited in crew path $i-j$, or when there is no waiting time associated with the visited damaged nodes. These constraints also prevent subtours. Constraints (7.12), (7.13) consider the case when the repair time of node j is one of the γ repair times assuming the worst case, while constraints (7.14), (7.15) consider that the γ repair times assuming the worst case are considered before node j. Constraints (7.12), (7.14) are defined for $i<j$ while constraints (7.13), (7.15) are defined for $i>j$. Similarly, constraints (7.16), (7.17) define the restoration time for different γ values when there is waiting time associated with the damaged nodes visited in a given crew path $i-j$. Constraints (7.16),(7.17) are activated only for the last node visited in the crew path $i-j$, i.e., when there is no node visited in the position $h+1$ ($\sum_{l\in\mathcal{V}^r} R_{l(h+1)j} = 0$). Constraints (7.18) evaluate the arrival time of crew

k at the first damaged node l visited in crew path $i-j$. Similarly, constraints (7.19) define the arrival time of crew k at the damaged node l visited in position $h>1$ based on the departure time of node v visited in position $h-1$. Finally, constraints (7.20) compute the waiting time of the crew in damaged node l visited in position h.

$$Z^{\mathrm{r}}_{j\gamma} \geq Z^{\mathrm{r}}_{i\gamma} + \sum_{e\in\mathcal{E}} \tau_{ke} P_{eij} + \delta_{kj} - M(2 - X_{ij} - W_{kj}),\ \forall\ k\in\mathcal{K},\ i\in\mathcal{V}^{\mathrm{r}}_0,\ j\in\mathcal{V}^{\mathrm{r}} : i<j,\ \gamma = 0,\dots,\Gamma, \tag{7.12}$$

$$Z^{\mathrm{r}}_{j\gamma} \geq Z^{\mathrm{r}}_{i\gamma} + \sum_{e\in\mathcal{E}} \tau_{ke} P_{eji} + \delta_{kj} - M(2 - X_{ij} - W_{kj}),\ \forall\ k\in\mathcal{K},\ i\in\mathcal{V}^{\mathrm{r}},\ j\in\mathcal{V}^{\mathrm{r}} : i>j,\ \gamma = 0,\dots,\Gamma, \tag{7.13}$$

$$Z^{\mathrm{r}}_{j\gamma} \geq Z^{\mathrm{r}}_{i(\gamma-1)} + \sum_{e\in\mathcal{E}} \tau_{ke} P_{eij} + \delta_{kj} + \hat{\delta}_{kj} - M(2 - X_{ij} - W_{kj}),\ \forall\ k\in\mathcal{K},\ i\in\mathcal{V}^{\mathrm{r}}_0,\ j\in\mathcal{V}^{\mathrm{r}} : i<j,\ \gamma = 1,\dots,\Gamma, \tag{7.14}$$

$$Z^{\mathrm{r}}_{j\gamma} \geq Z^{\mathrm{r}}_{i(\gamma-1)} + \sum_{e\in\mathcal{E}} \tau_{ke} P_{eji} + \delta_{kj} + \hat{\delta}_{kj} - M(2 - X_{ij} - W_{kj}),\ \forall\ k\in\mathcal{K},\ i\in\mathcal{V}^{\mathrm{r}},\ j\in\mathcal{V}^{\mathrm{r}} : i>j,\ \gamma = 1,\dots,\Gamma, \tag{7.15}$$

$$Z^{\mathrm{r}}_{j\gamma} \geq \sum_{l\in\mathcal{V}^{\mathrm{r}}} (T^{\mathrm{w}}_{lhj\gamma} + T^{\mathrm{s}}_{lhj\gamma} + \rho_{klj} R_{lhj}) + \delta_{kj} - M(1 - W_{kj} + \sum_{l\in\mathcal{V}^{\mathrm{r}}} R_{(h+1)lj}),$$
$$\forall\ j\in\mathcal{V}^{\mathrm{r}},\ h\in\mathcal{R}\setminus\{|\mathcal{R}|\},\ k\in\mathcal{K},\ \gamma = 0,\dots,\Gamma, \tag{7.16}$$

$$Z^{\mathrm{r}}_{j\gamma} \geq \sum_{l\in\mathcal{V}^{\mathrm{r}}} (T^{\mathrm{w}}_{lhj(\gamma-1)} + T^{\mathrm{s}}_{lhj(\gamma-1)} + \rho_{klj} R_{lhj}) + \delta_{kj} + \hat{\delta}_{kj} - M(1 - W_{kj} + \sum_{l\in\mathcal{V}^{\mathrm{r}}} R_{(h+1)lj}),$$
$$\forall\ j\in\mathcal{V}^{\mathrm{r}},\ h\in\mathcal{R}\setminus\{|\mathcal{R}|\},\ k\in\mathcal{K},\ \gamma = 1,\dots,\Gamma, \tag{7.17}$$

$$T^{\mathrm{s}}_{1lj\gamma} \geq Z^{\mathrm{r}}_{i\gamma} + \sum_{k\in\mathcal{K}} \rho_{kil} W_{kj} - M(2 - X_{ij} - R_{1lj}),\ \forall\ i\in\mathcal{V}^{\mathrm{r}}_0,\ j\in\mathcal{V}^{\mathrm{r}},\ l\in\mathcal{V}^{\mathrm{r}},\ \gamma = 0,\dots,\Gamma, \tag{7.18}$$

$$T^{\mathrm{s}}_{lhj\gamma} \geq \sum_{p\in\mathcal{V}^{\mathrm{r}}} (T^{\mathrm{w}}_{(h-1)pj\gamma} + T^{\mathrm{s}}_{(h-1)pj\gamma} + R_{(h-1)pj}\rho_{kpl}) - M(2 - R_{lhj} - W_{kj}),$$
$$\forall\ k\in\mathcal{K}, l\in\mathcal{V}^{\mathrm{r}}, j\in\mathcal{V}^{\mathrm{r}},\ h\in\mathcal{R}\setminus\{1\},\ \gamma = 0,\dots,\Gamma, \tag{7.19}$$

$$T^{\mathrm{w}}_{lhj\gamma} \geq Z^{\mathrm{r}}_{l\Gamma} - T^{\mathrm{s}}_{lhj\gamma} - M(1 - R_{lhj}),\ \forall\ l\in\mathcal{V}^{\mathrm{r}},\ j\in\mathcal{V}^{\mathrm{r}},\ h\in\mathcal{R},\ \gamma = 0,\dots,\Gamma. \tag{7.20}$$

Assignment and scheduling constraints: Constraints (7.21) force the allocation of crews to damaged nodes that are repaired. Constraints (7.22),(7.23) state that damaged nodes used either in relief paths or in crew paths must be repaired. Constraints (7.24)-(7.26) define the schedule for the damaged nodes that must be repaired. Constraints (7.27) guarantee that if nodes i and j are considered in the same schedule ($X_{ij} = 1$), both are repaired by the same crew. Constraints (7.28) force the consideration of different crews for the different schedules.

$$Q_l = \sum_{k\in\mathcal{K}_l} W_{kl},\ \forall\ l\in\mathcal{V}^{\mathrm{r}}, \tag{7.21}$$

$$|\mathcal{V}^{\mathrm{d}}| Q_l \geq \sum_{i\in\mathcal{V}^{\mathrm{d}}} V_{li},\ \forall\ l\in\mathcal{V}^{\mathrm{r}}, \tag{7.22}$$

$$|\mathcal{V}^{\mathrm{r}}| Q_l \geq \sum_{i\in\mathcal{V}^{\mathrm{r}}_0} \sum_{\substack{j\in\mathcal{V}^{\mathrm{r}}:\\ j>i}} N_{lij},\ \forall\ l\in\mathcal{V}^{\mathrm{r}}, \tag{7.23}$$

$$\sum_{\substack{i\in\mathcal{V}^{\mathrm{r}}_0:\\ i\neq j}} X_{ij} = Q_j,\ \forall\, j\in\mathcal{V}^{\mathrm{r}}, \tag{7.24}$$

$$\sum_{\substack{i \in \mathcal{V}_0^{\mathrm{r}}:\\ i \neq l}} X_{il} - \sum_{\substack{j \in \mathcal{V}_0^{\mathrm{r}}:\\ j \neq l}} X_{lj} = 0, \forall\, l \in \mathcal{V}_0^{\mathrm{r}}, \tag{7.25}$$

$$\sum_{j \in \mathcal{V}^{\mathrm{r}}} X_{0j} \leq |\mathcal{K}|, \tag{7.26}$$

$$W_{kj} \geq W_{ki} + X_{ij} - 1, \forall\, i \in \mathcal{V}^{\mathrm{r}},\ j \in \mathcal{V}^{\mathrm{r}},\ k \in \mathcal{K}, \tag{7.27}$$

$$\sum_{\substack{k' \in \mathcal{K}:\\ k' \neq k}} W_{k'j} \geq W_{ki} + X_{0i} + X_{0j} - 2, \forall\, i \in \mathcal{V}^{\mathrm{r}},\ j \in \mathcal{V}^{\mathrm{r}} : i \neq j,\ k \in \mathcal{K}. \tag{7.28}$$

Routing constraints: Constraints (7.29)-(7.33) define the paths of the crews. For a crew traveling from $i \in \mathcal{V}_0^{\mathrm{r}}$ to $j \in \mathcal{V}^{\mathrm{r}}$, constraints (7.29),(7.31) force the use of an arc incident to node i, constraints (7.30),(7.32) force the use of an arc incident to node j, and constraints (7.33) force the use of two arcs incident to nodes l in the middle of the crew path $i - j$.

$$\sum_{e \in \mathcal{E}_0} P_{e0j} = X_{0j}, \forall\, j \in \mathcal{V}^{\mathrm{r}}, \tag{7.29}$$

$$\sum_{e \in \mathcal{E}_j} P_{e0j} = X_{0j}, \forall\, j \in \mathcal{V}^{\mathrm{r}}, \tag{7.30}$$

$$\sum_{e \in \mathcal{E}_i} P_{eij} = X_{ij} + X_{ji}, \forall\, i \in \mathcal{V}^{\mathrm{r}},\ j \in \mathcal{V}^{\mathrm{r}} : i < j, \tag{7.31}$$

$$\sum_{e \in \mathcal{E}_j} P_{eij} = X_{ij} + X_{ji}, \forall\, i \in \mathcal{V}^{\mathrm{r}},\ j \in \mathcal{V}^{\mathrm{r}} : i < j, \tag{7.32}$$

$$\sum_{e \in \mathcal{E}_l} P_{eij} = 2N_{lij}, \forall\, i \in \mathcal{V}_0^{\mathrm{r}},\ j \in \mathcal{V}^{\mathrm{r}},\ l \in \mathcal{V} \setminus \{i,\ j\} : i < j. \tag{7.33}$$

Synchronization constraints: Constraints (7.34)-(7.39) synchronize the arrival of crew k in damaged node l visited in crew path $i-j$. Here, i and j are damaged nodes repaired by crew k, while l is a damaged node used in path $i - j$. Thus, when crew k arrives at this node l, it either waits for node l to be repaired by another crew if this node is still damaged, or it can cross without waiting if l has been already repaired. A damaged node l cannot be visited more than once in the path to node j (constraints (7.34)). Also, a given crew cannot visit different damaged nodes simultaneously (constraints (7.35)), i.e., in the same position h. Constraints (7.36) ensure that damaged nodes in path $i - j$ must be visited in consecutive positions. Constraints (7.37)-(7.39) link the allocation of damaged nodes to positions with the definition of the crew paths. Thus, only damaged nodes considered in the crew path $i-j$ ($N_{lij} = 1$) can be allocated to some position defined by the variable R_{lhj}.

$$\sum_{h \in \mathcal{R}} R_{lhj} \leq 1, \forall\, j \in \mathcal{V}^{\mathrm{r}},\ l \in \mathcal{V}^{\mathrm{r}}, \tag{7.34}$$

$$\sum_{l \in \mathcal{V}^{\mathrm{r}}} R_{lhj} \leq 1, \forall\, j \in \mathcal{V}^{\mathrm{r}},\ h \in \mathcal{R}, \tag{7.35}$$

$$\sum_{l \in \mathcal{V}^{\mathrm{r}}} R_{lhj} \leq \sum_{l \in \mathcal{V}^{\mathrm{r}}} R_{(h-1)lj}, \forall\, j \in \mathcal{V}^{\mathrm{r}},\ h \in \mathcal{R} \setminus \{1\}, \tag{7.36}$$

$$\sum_{h \in \mathcal{R}} R_{lhj} \geq N_{lij} + X_{ij} - 1, \forall\, i \in \mathcal{V}^{\mathrm{r}},\ l \in \mathcal{V}^{\mathrm{r}},\ j \in \mathcal{V}^{\mathrm{r}} : i < j, \tag{7.37}$$

$$\sum_{h \in \mathcal{R}} R_{lhi} \geq N_{lij} + X_{ji} - 1, \forall\, i \in \mathcal{V}^{\mathrm{r}},\ l \in \mathcal{V}^{\mathrm{r}},\ j \in \mathcal{V}^{\mathrm{r}} : i < j, \tag{7.38}$$

$$\sum_{h\in\mathcal{R}}\sum_{j\in\mathcal{V}^{\mathrm{r}}} R_{lhj} = \sum_{i\in\mathcal{V}_0^{\mathrm{r}}}\sum_{j\in\mathcal{V}^{\mathrm{r}}:j>i} N_{lij},\ \forall\ l\in\mathcal{V}^{\mathrm{r}}. \tag{7.39}$$

Domain of the decision variables: Constraints (7.40)-(7.49) impose the domain of the decision variables.

$$Q_j, W_{kj} \in \{0,\ 1\},\ \forall\ j\in\mathcal{V}^{\mathrm{r}},\ \ k\in\mathcal{K}, \tag{7.40}$$
$$X_{ij} \in \{0,\ 1\},\ \forall\ i\in\mathcal{V}_0^{\mathrm{r}},\ \ j\in\mathcal{V}_0^{\mathrm{r}}, \tag{7.41}$$
$$R_{hij} \in \{0,\ 1\},\ \forall\ i\in\mathcal{V}^{\mathrm{r}},\ \ j\in\mathcal{V}^{\mathrm{r}},\ \ h\in\mathcal{R}, \tag{7.42}$$
$$N_{lij} \in \{0,\ 1\},\ \forall\ i\in\mathcal{V}_0^{\mathrm{r}},\ \ j\in\mathcal{V}^{\mathrm{r}},\ \ l\in\mathcal{V}, \tag{7.43}$$
$$P_{eij} \geq 0,\ \forall\ i\in\mathcal{V}_0^{\mathrm{r}},\ \ j\in\mathcal{V}^{\mathrm{r}},\ \ e\in\mathcal{E}, \tag{7.44}$$
$$V_{li} \in \{0,\ 1\},\ \forall\ i\in\mathcal{V}^{\mathrm{d}},\ \ l\in\mathcal{V}, \tag{7.45}$$
$$Y_{el} \geq 0,\ \forall\ l\in\mathcal{V}^{\mathrm{d}},\ \ e\in\mathcal{E}, \tag{7.46}$$
$$T^{\mathrm{s}}_{lhj\gamma}, T^{\mathrm{w}}_{lhj\gamma}\ \geq 0,\ \forall\ l\in\mathcal{V}^{\mathrm{r}}, j\in\mathcal{V}^{\mathrm{r}},\ \ h\in\mathcal{R}, \gamma=0,\cdots,\Gamma, \tag{7.47}$$
$$Z^{\mathrm{r}}_{i\gamma}\ \geq 0,\ \forall\ i\in\mathcal{V}_0^{\mathrm{r}},\ \ \gamma=0,\cdots,\Gamma, \tag{7.48}$$
$$Z^{\mathrm{d}}_{j}\ \geq 0,\ \forall\ j\in\mathcal{V}^{\mathrm{d}}. \tag{7.49}$$

7.3 Solution approach

In this section, we propose a logic–based Benders decomposition (Hooker and Ottosson, 2003) approach to solve the RCSRP, in which the master problem (MP) involves assignment, scheduling and relief path definition decisions and the subproblem (SP) considers relief path definition decisions and crew routing decisions. Both, MP and SP, include integer variables and therefore standard duality theory cannot be applied to derive cuts.

The relief path definition decisions are considered in both, MP and SP subproblems. In the MP, these decisions help to increase the lower bound, and thus improving the performance of the method. On the other hand, we notice that adding the relief paths variables in the subproblem does not significantly increase the difficulty of the subproblem while reducing the number of solutions that need to be "updated" with feasibility or optimility cuts in the MP and consequently fewer cuts that are needed to guarantee convergence.

The MP obtained from the Benders decomposition is solved by a single search tree, exploring the generation of cuts inside the tree during the branch–and–bound process. This strategy has been recently referred to as Branch-and-Benders-Cut (BBC) (Gendron et al., 2016; Errico et al., 2017) or Branch-and-Check algorithm (B&Ch) (Tran et al., 2016; Perez et al., 2019). Usually, B&Ch is more common in the context of logic–based Benders decomposition approaches.

7.3.1 Logic-based Benders reformulation (LBBR)

Let $\theta^{\mathrm{r}}_{j\gamma}$ be an auxiliary variable that represents the restoration time of damaged node j when γ repair times assume their worst case value. Let $\beta_j(\bar{\boldsymbol{q}}, \Gamma)$ be a function that returns the optimal restoration time of damaged node $j \in \mathcal{V}^{\mathrm{r}}$ given an assignment-scheduling solution $\bar{\boldsymbol{q}} = (\bar{\boldsymbol{X}}, \bar{\boldsymbol{W}}, \bar{\boldsymbol{Q}})$ and a budget of uncertainty Γ. Also, let $\mathcal{SP}$ be the set of assignment-scheduling solutions for which a feasible crew routing decision can be defined. The logic-based Benders reformulation (LBBR) for the RCSRP is as follows:

$$(LBBR) \quad \min \sum_{i\in\mathcal{V}^{\mathrm{d}}} d_i(Z_i^{\mathrm{d}} + \sum_{e\in\mathcal{E}} t_e Y_{ei}), \tag{7.50}$$

s.t.

$$\texttt{Relief paths constraints } (7.8)-(7.11), \tag{7.51}$$

$$\texttt{Assignment and scheduling constraints } (7.21),(7.22),(7.24)-(7.28), \tag{7.52}$$

$$\texttt{Domain of the decision variables } (7.40),(7.41),(7.45),(7.46),(7.49), \tag{7.53}$$

$$Z_i^{\mathrm{d}} \geq \theta_{j\Gamma}^{\mathrm{r}} - M\cdot(1-V_{ji}),\ \forall\ i\in\mathcal{V}^{\mathrm{d}},\ j\in\mathcal{V}^{\mathrm{r}}; \tag{7.54}$$

$$\theta_{j\Gamma}^{\mathrm{r}} \geq \beta_j(\boldsymbol{q},\Gamma),\ \forall\ j\in\mathcal{V}^{\mathrm{r}}; \tag{7.55}$$

$$\boldsymbol{q}\in\mathcal{SP}; \tag{7.56}$$

$$\theta_{j\gamma}^{\mathrm{r}} \geq 0,\ \forall\ j\in\mathcal{V}^{\mathrm{r}}, \gamma=0,\cdots,\Gamma. \tag{7.57}$$

Constraints (7.54) define the accessibility time for the demand nodes. Constraints (7.55) updated variable $\theta_{j\Gamma}^{\mathrm{r}}$ according to the optimal restoration time of damaged node j. Constraint (7.56) prohibits infeasible assignment-scheduling solutions for which it is not possible to find synchronized routes of the crews. Finally, constraints (7.57) define the domain of $\theta_{j\gamma}^{\mathrm{r}}$ variables.

Given a solution $\bar{\boldsymbol{q}}$ and a budget of uncertainty Γ, $\beta_j(\bar{\boldsymbol{q}},\Gamma)=\bar{Z}_{j\Gamma}^{\mathrm{r}*}$, where $\bar{Z}_{j\Gamma}^{\mathrm{r}*}$ is equal to the value of variable $Z_{i\Gamma}^{\mathrm{r}}$ in the optimal solution of the following subproblem (SP).

$$(SP)\ \min \sum_{i\in\mathcal{V}^{\mathrm{d}}} d_i(Z_i^{\mathrm{d}} + \sum_{e\in\mathcal{E}} t_e Y_{ei}), \tag{7.58}$$

s.t.

$$\texttt{Relief paths constraints } (7.7)-(7.11), \tag{7.59}$$

$$\texttt{Routing and synchronization constraints } (7.33)-(7.36), \tag{7.60}$$

$$\texttt{Domain of the decision variables } (7.42)-(7.49), \tag{7.61}$$

$$Z_{j\gamma}^{\mathrm{r}} \geq Z_{i\gamma}^{\mathrm{r}} + \sum_{e\in\mathcal{E}} \tau_{ke}P_{eij} + \delta_{kj},\ \forall\ k\in\mathcal{K},\ i\in\mathcal{V}_0^{\mathrm{r}},\ j\in\mathcal{V}^{\mathrm{r}}: i<j, \bar{X}_{ij}=1, \bar{W}_{kj}=1,\ \gamma=0,\dots,\Gamma, \tag{7.62}$$

$$Z_{j\gamma}^{\mathrm{r}} \geq Z_{i\gamma}^{\mathrm{r}} + \sum_{e\in\mathcal{E}} \tau_{ke}P_{eji} + \delta_{kj},\ \forall\ k\in\mathcal{K},\ i\in\mathcal{V}^{\mathrm{r}},\ j\in\mathcal{V}^{\mathrm{r}}: i>j, \bar{X}_{ij}=1, \bar{W}_{kj}=1,\ \gamma=0,\dots,\Gamma, \tag{7.63}$$

$$Z_{j\gamma}^{\mathrm{r}} \geq Z_{i(\gamma-1)}^{\mathrm{r}} + \sum_{e\in\mathcal{E}} \tau_{ke}P_{eij} + \delta_{kj} + \hat{\delta}_{kj},\ \forall\ k\in\mathcal{K},\ i\in\mathcal{V}_0^{\mathrm{r}},\ j\in\mathcal{V}^{\mathrm{r}}: i<j, \bar{X}_{ij}=1, \bar{W}_{kj}=1,\ \gamma=1,\dots,\Gamma, \tag{7.64}$$

$$Z_{j\gamma}^{\mathrm{r}} \geq Z_{i(\gamma-1)}^{\mathrm{r}} + \sum_{e\in\mathcal{E}} \tau_{ke}P_{eji} + \delta_{kj} + \hat{\delta}_{kj},\ \forall\ k\in\mathcal{K},\ i\in\mathcal{V}^{\mathrm{r}},\ j\in\mathcal{V}^{\mathrm{r}}: i>j, \bar{X}_{ij}=1, \bar{W}_{kj}=1,\ \gamma=1,\dots,\Gamma, \tag{7.65}$$

$$Z_{j\gamma}^{\mathrm{r}} \geq \sum_{l\in\mathcal{V}^{\mathrm{r}}} (T_{lhj\gamma}^{\mathrm{w}} + T_{lhj\gamma}^{\mathrm{s}} + \rho_{klj}R_{lhj}) + \delta_{kj} - M(\sum_{l\in\mathcal{V}^{\mathrm{r}}} R_{(h+1)lj}),$$
$$\forall\ j\in\mathcal{V}^{\mathrm{r}},\ k\in\mathcal{K}: \bar{W}_{kj}=1,\ h\in\mathcal{R}\setminus\{|\mathcal{R}|\},\ \gamma=0,\dots,\Gamma, \tag{7.66}$$

$$Z_{j\gamma}^{\mathrm{r}} \geq \sum_{l\in\mathcal{V}^{\mathrm{r}}} (T_{lhj(\gamma-1)}^{\mathrm{w}} + T_{lhj(\gamma-1)}^{\mathrm{s}} + \rho_{klj}R_{lhj}) + \delta_{kj} + \hat{\delta}_{kj} - M(\sum_{l\in\mathcal{V}^{\mathrm{r}}} R_{(h+1)lj}),$$

$$\forall\, j \in \mathcal{V}^{\mathrm{r}},\ k \in \mathcal{K} : \bar{W}_{kj} = 1,\ h \in \mathcal{R} \setminus \{|\mathcal{R}|\},\ \gamma = 1, \dots, \Gamma, \tag{7.67}$$

$$T^{\mathrm{s}}_{1lj\gamma} \geq Z^{\mathrm{r}}_{i\gamma} + \sum_{k \in \mathcal{K}} \rho_{kil} \bar{W}_{kj} - M(1 - R_{1lj}),\ \forall\, i \in \mathcal{V}^{\mathrm{r}}_0,\ j \in \mathcal{V}^{\mathrm{r}} : \bar{X}_{ij} = 1,\ l \in \mathcal{V}^{\mathrm{r}},\ \gamma = 0, \dots, \Gamma, \tag{7.68}$$

$$T^{\mathrm{s}}_{lhj\gamma} \geq \sum_{p \in \mathcal{V}^{\mathrm{r}}} (T^{\mathrm{w}}_{(h-1)pj\gamma} + T^{\mathrm{s}}_{(h-1)pj\gamma} + R_{(h-1)pj} \rho_{kpl}) - M(1 - R_{lhj}),$$

$$\forall\, k \in \mathcal{K}, j \in \mathcal{V}^{\mathrm{r}} : \bar{W}_{kj} = 1, l \in \mathcal{V}^{\mathrm{r}},\ h \in \mathcal{R} \setminus \{1\},\ \gamma = 0, \dots, \Gamma, \tag{7.69}$$

$$T^{\mathrm{w}}_{lhj\gamma} \geq Z^{\mathrm{r}}_{l\Gamma} - T^{\mathrm{s}}_{lhj\gamma} - M(1 - R_{lhj}),\ \forall\, l \in \mathcal{V}^{\mathrm{r}} : \bar{Q}_l = 1,\ j \in \mathcal{V}^{\mathrm{r}} : \bar{Q}_j = 1,\ h \in \mathcal{R},\ \gamma = 0, \dots, \Gamma, \tag{7.70}$$

$$\sum_{e \in \mathcal{E}_0} P_{e0j} = 1,\ \forall\, j \in \mathcal{V}^{\mathrm{r}} : \bar{X}_{0j} = 1, \tag{7.71}$$

$$\sum_{e \in \mathcal{E}_j} P_{e0j} = 1,\ \forall\, j \in \mathcal{V}^{\mathrm{r}} : \bar{X}_{0j} = 1, \tag{7.72}$$

$$\sum_{e \in \mathcal{E}_i} P_{eij} = 1,\ \forall\, i \in \mathcal{V}^{\mathrm{r}},\ j \in \mathcal{V}^{\mathrm{r}} : i < j, \bar{X}_{ij} + \bar{X}_{ji} = 1, \tag{7.73}$$

$$\sum_{e \in \mathcal{E}_j} P_{eij} = 1,\ \forall\, i \in \mathcal{V}^{\mathrm{r}},\ j \in \mathcal{V}^{\mathrm{r}} : i < j, \bar{X}_{ij} + \bar{X}_{ji} = 1, \tag{7.74}$$

$$\sum_{h \in \mathcal{R}} R_{lhj} \geq N_{lij},\ \forall\, i \in \mathcal{V}^{\mathrm{r}},\ l \in \mathcal{V}^{\mathrm{r}},\ j \in \mathcal{V}^{\mathrm{r}} : i < j, \bar{X}_{ij} = 1, \tag{7.75}$$

$$\sum_{h \in \mathcal{R}} R_{lhi} \geq N_{lij},\ \forall\, i \in \mathcal{V}^{\mathrm{r}},\ l \in \mathcal{V}^{\mathrm{r}},\ j \in \mathcal{V}^{\mathrm{r}} : i < j, \bar{X}_{ji} = 1, \tag{7.76}$$

$$\sum_{h \in \mathcal{R}} \sum_{j \in \mathcal{V}^{\mathrm{r}}} R_{lhj} = \sum_{i \in \mathcal{V}^{\mathrm{r}}_0} \sum_{j \in \mathcal{V}^{\mathrm{r}} : j > i} N_{lij},\ \forall\, l \in \mathcal{V}^{\mathrm{r}} : \bar{Q}_l = 1, \tag{7.77}$$

$$\sum_{j \in \mathcal{V}^{\mathrm{d}}} V_{lj} + \sum_{i \in \mathcal{V}^{\mathrm{r}}_0} \sum_{j \in \mathcal{V}^{\mathrm{r}}} (N_{lij} + N_{jil}) + \sum_{h \in \mathcal{R}} \sum_{j \in \mathcal{V}^{\mathrm{r}}} (R_{lhj} + R_{jhl}) + \sum_{i \in \mathcal{V}^{\mathrm{r}}_0} \sum_{e \in \mathcal{E}} P_{eil} \leq 0,\ \forall\, l \in \mathcal{V}^{\mathrm{r}} : \bar{Q}_l = 0, \tag{7.78}$$

$$\sum_{l \in \mathcal{V}} N_{lij} + \sum_{e \in \mathcal{E}} P_{eij} \leq 0,\ \forall\, i \in \mathcal{V}^{\mathrm{r}}_0,\ j \in \mathcal{V}^{\mathrm{r}} : \bar{X}_{ij} = 0. \tag{7.79}$$

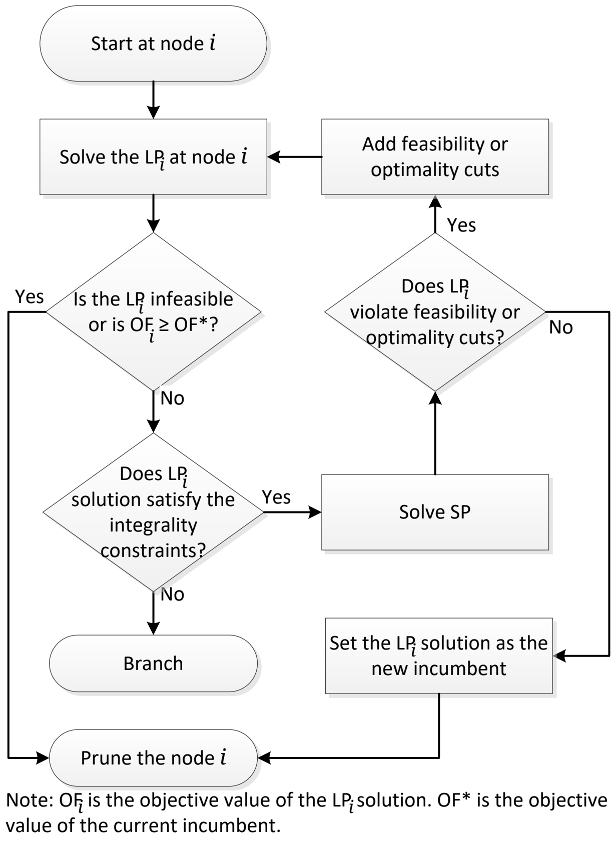
Constraints (7.62)-(7.77) are constraints of the original RCSRP with the fixed values $(\bar{\boldsymbol{X}}, \bar{\boldsymbol{W}}, \bar{\boldsymbol{Q}})$ of the assignment-scheduling variables. These constraints apply for the damaged nodes allocated to some crew, i.e., $\forall i, j : \bar{X}_{ij} = 1, \forall k, j : \bar{W}_{kj} = 1, \forall j : \bar{Q}_j = 1$. We add constraints (7.78), (7.79) to prohibit routing decisions between damaged nodes that are not allocated to any crew, i.e., $\forall i, j : \bar{X}_{ij} = 0, \forall k, j : \bar{W}_{kj} = 0, \forall j : \bar{Q}_j = 0$. This way we avoid the generation of unnecessary routing and synchronization constraints (7.62)-(7.77) for damaged nodes that are not repaired by the crews.

7.3.2 Branch-and-check algorithm (B&Ch)

We develop a B&Ch to solve the LBBR model (7.50)-(7.57). Basically, we define a master problem (MP) by removing constraints (7.55), (7.56) from the original LBBR model and solve the MP by using a branch-and-bound method. The removed constraints are dynamically checked in the nodes of the branch-and-bound tree and optimality/feasibility cuts are added to the MP when these constraints are violated. The proposed Benders optimality/feasibility cuts are presented in Section 7.3.3.

Figure 7.1 shows a flowchart of the main steps carried out in B&Ch algorithm at each node of the branch-and-bound tree. At each node i, we solve the linear relaxation of the current MP, denoted by LP$_i$. If the LP$_i$ is infeasible or the objective value of the LP$_i$ solution (OF$_i$) is higher than or equal to the objective value of the current incumbent solution, then node i is pruned. Otherwise, integrality constraints are checked, and if the LP$_i$ solution is not integer feasible, then branching is performed. If the LP$_i$ solution is integer feasible, we solve the subproblem SP (7.58)-(7.79) to verify violation of constraints (7.55), (7.56). A constraint of the set of constraints (7.55) is violated for a given node j if $\bar{\theta}^{\mathfrak{r}*}_{j\Gamma} < \bar{Z}^{\mathfrak{r}*}_{i\Gamma}$, where $\bar{\theta}^{\mathfrak{r}*}_{j\Gamma}$ is the value of variable $\theta^{\mathfrak{r}}_{j\Gamma}$ in the LP$_i$ solution. Constraint (7.56) is violated if the subproblem SP is infeasible. If no constraint is violated, then the LP$_i$ solution is feasible for the original LBBR and is set as the new incumbent solution. Otherwise, the MP is modified by the addition of Benders cuts, LP$_i$ is resolved, and the described steps are applied again. General-purpose optimization software may additionally rely on automated cuts and/or heuristics that are not included in Figure 7.1.

Figure 7.1: Flowchart illustrating the main steps of the B&Ch algorithm in a given node i of the branch-and-bound.


Start at node i
Solve the LP$_i$ at node i
Add feasibility or optimality cuts
Yes
Is the LP$_i$ infeasible or is OF$_i$ ≥ OF*?
Yes
No
Does LP$_i$ violate feasibility or optimality cuts?
No
Does LP$_i$ solution satisfy the integrality constraints?
Yes
No
Solve SP
Branch
Set the LP$_i$ solution as the new incumbent
Prune the node i
Note: OF$_i$ is the objective value of the LP$_i$ solution. OF* is the objective value of the current incumbent.

7.3.3 Logic-based Benders Cuts

Every time an integer solution is found at the nodes of the branch-and-bound tree, the subproblem SP is solved and the corresponding feasibility/optimality Benders cuts are added to the MP. We rely on logic-based Benders cuts based on particular characteristics of the problem. Proposition 10 state feasibility and optimality cuts for the MP.

Proposition 10. *Let $\bar{\boldsymbol{q}} = (\bar{\boldsymbol{X}}, \bar{\boldsymbol{W}}, \bar{\boldsymbol{Q}})$ be an assignment-scheduling solution for the MP. Let*

$$\Theta_{\bar{\boldsymbol{q}}}(\boldsymbol{q}) = \sum_{i\in\mathcal{V}_0^{\mathrm{r}}} \sum_{\substack{j\in\mathcal{V}_0^{\mathrm{r}}:\\ \bar{X}_{ij}=1}} (X_{ij}-1) + \sum_{k\in\mathcal{K}} \sum_{\substack{j\in\mathcal{V}^{\mathrm{r}}:\\ \bar{W}_{kj}=1}} (W_{kj}-1) + \sum_{\substack{j\in\mathcal{V}^{\mathrm{r}}:\\ \bar{Q}_{j}=1}} (Q_j-1) - \sum_{i\in\mathcal{V}_0^{\mathrm{r}}} \sum_{\substack{j\in\mathcal{V}_0^{\mathrm{r}}:\\ \bar{X}_{ij}=0}} X_{ij} - \sum_{k\in\mathcal{K}} \sum_{\substack{j\in\mathcal{V}^{\mathrm{r}}:\\ \bar{W}_{kj}=0}} W_{kj} - \sum_{\substack{j\in\mathcal{V}^{\mathrm{r}}:\\ \bar{Q}_{j}=0}} Q_j.$$

If the assignment-scheduling solution $\bar{\boldsymbol{q}}$ violates constraints (7.56), a valid feasibility cut for the problem is

$$\Theta_{\bar{\boldsymbol{q}}}(\boldsymbol{q}) \leq -1. \tag{7.80}$$

If the assignment-scheduling solution $\bar{\boldsymbol{q}}$ violates any constraint of the set of constraints (7.55) for a given node j, a valid optimality cut for the problem is

$$\theta_j^{\mathrm{r}} \geq \beta_j(\bar{\boldsymbol{q}}, \Gamma) + \beta_j(\bar{\boldsymbol{q}}, \Gamma) \cdot \Theta_{\bar{\boldsymbol{q}}}(\boldsymbol{q}). \tag{7.81}$$

Proof. *If we replace the solution $\bar{\boldsymbol{q}}$ in the expression $\Theta_{\bar{\boldsymbol{q}}}(\boldsymbol{q})$, then $\Theta_{\bar{\boldsymbol{q}}}(\boldsymbol{q}) = 0$. For any other solution $\bar{\boldsymbol{q}}' \neq \bar{\boldsymbol{q}}$, $\Theta_{\bar{\boldsymbol{q}}}(\boldsymbol{q}) < 0$. Therefore, to prohibit the selection of solution $\bar{\boldsymbol{q}}$ we can add equation (7.80). Similarly, if $\bar{\boldsymbol{q}}$ is selected as a solution of the MP, $\beta_j(\bar{\boldsymbol{q}}, \Gamma) + \beta_j(\bar{\boldsymbol{q}}, \Gamma) \cdot \Theta_{\bar{\boldsymbol{q}}}(\boldsymbol{q}) = \beta_j(\bar{\boldsymbol{q}}, \Gamma) = \bar{Z}_{j\Gamma}^{\mathrm{r}*}$, and thus the cost $\beta_j(\bar{\boldsymbol{q}}, \Gamma)$ is activated as a lower bound for variable $\theta_{j\Gamma}^{\mathrm{r}}$ in the MP. Otherwise, $\beta_j(\bar{\boldsymbol{q}}, \Gamma) + \beta_j(\bar{\boldsymbol{q}}, \Gamma) \cdot \Theta_{\bar{\boldsymbol{q}}}(\boldsymbol{q}) \leq 0$, and the lower bound $\beta_j(\bar{\boldsymbol{q}}, \Gamma)$ is not activated in the MP.*

7.3.4 Valid inequalities

We adapt the valid inequalities proposed for the MCSRP (Chapter 6) to the RCSRP model and the LBBR. For the inequalities related to relief path proposed in Propositions 6 and 7 we redefine dominated path as follows: Given two paths p, $p' \in \mathcal{P}_i^{\mathrm{d}}$ such that $p \neq p'$, we say that p dominates p' if $\mathcal{V}_p^{\mathrm{r}} \subseteq \mathcal{V}_{p'}^{\mathrm{r}}$ and $w_p \leq w_{p'}$. The equations are the same as presented in Proposition 6 and 7. The inequalities related to routing decisions proposed in Proposition 8 are added to the RCSRP model, but they are not used in the LBBR since the LBBR does not consider routing decisions explicitly. On the other hand, inequalities proposed in Proposition 9 are modified and added to the RCSRP model and to the LBBR as follows:

$$Z_{j\gamma}^{\mathrm{r}} \geq Z_{i\gamma}^{\mathrm{r}} + \rho_{kij}^{*} + \delta_{kj} - M(2 - X_{ij} - W_{kj}), \forall\, k \in \mathcal{K},\, i \in \mathcal{V}_0^{\mathrm{r}},\; j \in \mathcal{V}^{\mathrm{r}},\; \gamma = 0, \ldots, \Gamma, \tag{7.82}$$

$$Z_{j\gamma}^{\mathrm{r}} \geq Z_{i(\gamma-1)}^{\mathrm{r}} + \rho_{kij}^{*} + \delta_{kj} + \hat{\delta}_{kj} - M(2 - X_{ij} - W_{kj}), \forall\, k \in \mathcal{K},\, i \in \mathcal{V}_0^{\mathrm{r}},\; j \in \mathcal{V}^{\mathrm{r}},\; \gamma = 1, \ldots, \Gamma, \tag{7.83}$$

$$Z_{j\gamma}^{\mathrm{r}} \geq (\rho_{k0j}^{*} + \delta_{kj}) \cdot X_{k0j} + \sum_{i\in\mathcal{V}^{\mathrm{r}}} ((\rho_{k0i}^{*} + \delta_{ki} + \rho_{kij}^{*} + \delta_{kj}) \cdot X_{kij}), \forall k \in \mathcal{K},\; j \in \mathcal{V}^{\mathrm{r}}. \tag{7.84}$$

Inequalities (7.82)-(7.84) are for the RCSRP model. For the LBBR model, we replace $Z_{i\gamma}^{\mathrm{r}}$ with $\theta_{i\gamma}^{\mathrm{r}}$.

7.3.5 Metaheuristic algorithms

The metaheuristic algorithms GA and SA proposed for the SCSRP in Section 5.3 are adapted to the RCSRP. The main changes consist on the consideration of two new local search operators. An exchange operator, in which two damaged nodes allocated to different crews are randomly selected and exchanged, and an insertion operator that randomly takes a damaged node and insert it into the schedule of one crew. Another relevant change is that instead of using the feasibility and optimality check algorithm to verify the feasibility and cost of the schedules, we resort to the subproblem SP (7.58)-(7.79). In the computational experiment presented in Section 7.4, we will consider only the SA metaheuristic. The GA metaheuristic was tested to solve the RCSRP in some preliminary experiments, but its performance was significantly worse than SA. Remember that GA must check feasibility and optimality twice at each iteration (see Algorithm 5), while SA need to check feasibility and optimality only once per iteration (see Algorithm 7). In the RCSRP, the evaluation of the cost and feasibility of the solutions in the metaheuristics requires to solve the subproblem SP, which is an MIP model and therefore significantly deteriorates the performance of GA.

7.4 Computational results

In this section, we report the computational performance of the proposed robust approach. Our specific goals are threefold. The first goal is studying the behavior of the RCSRP via a general-purpose optimization software. The second goal relies on evaluating the efficiency of the tailored solution methods in providing good-quality solutions within a plausible running time. The last goal is investigating the quality of the robust solutions.

All the algorithms were coded in C++ programming language and run on a Linux PC with an AMD Opteron 6172 processor with 16.0 GB of RAM and a single thread. Benders cuts are added using the Callback classes available in the Concert Technology Library. The RCSRP model and the MP and SP subproblems were solved by CPLEX Optimization Solver 12.8. To avoid running out of memory, we allow CPLEX to store the branch-and-bound tree in a file. As we use lazy constraints Callback, CPLEX automatically turns off nonlinear reductions and dual reductions. The stopping criterion on CPLEX was either the elapsed time exceeding the time limit of 3,600 seconds or the optimality gap being smaller than 10^{-4}. All the remaining parameters of CPLEX were kept at their default values. For the metaheuristic, the stopping criterion was either the elapsed time exceeding the time limit of 3,600 seconds or performing 500 iterations without improving the best solution. The algorithms were tested using the CS instance classes for the MCSRP presented in Section 6.4.1. CS instances are based on the 2011 megadisaster of the Serrana Region in Rio de Janeiro, Brazil. The deviation of the repair times was assumed as $\hat{\delta}_{ki} = \alpha\delta_{ki}$ with $\alpha = 0.1, 0.25, 0.5$. To test the performance of the model and solution methods, we consider $\Gamma = 0, 1, 3$ for the instances with 6 damaged nodes, $\Gamma = 0, 3, 5$ for the instances with 10 damaged nodes and $\Gamma = 0, 4, 8$ for the instances with 14 damaged nodes. By combining the Γ and α values for the different instances classes, we tested 1,596 instances in total. Table 7.1 summarizes the solution methods used in the experiments. All strategies use the valid inequalities proposed for the MCSRP, as described in Section 7.3.4.

Table 7.1: Solution strategies.

Strategy	Description
M-RCSRP	MIP model (7.6)-(7.49).
B&Ch	LBBR model (7.50)-(7.57) solved by the branch-and-check algorithm.
SA	Simulated annealing metaheuristic.
SA-B&Ch	Warm-start the B&Ch using a solution from SA.

7.4.1 Computational performance of the solution approaches

In this section, we analyze the performance of the solution approaches for the RCSRP. Table 7.2 shows for each approach the total number of instances (#ins), the number of instances for which the approach finished with feasible and optimal solutions (#feas, #opt), each one followed by the corresponding percentage with respect to the total number of instances (%feas, %opt), the average upper and lower bounds, the average relative gap in percentage, and the average elapsed time.

Table 7.2: Comparison of the solution approaches.

Solution method	#ins	#feas	%feas	#opt	%opt	Avg. upper bound	Avg. lower bound	Avg. gap (%)	Avg. time (seconds.)
M-RCSRP	1,596	834	52.26	248	15.54	311,629	197,946	41.08	2,877.16
B&Ch	1,596	1,151	72.12	347	21.75	302,344	200,521	38.21	2,408.73
SA	1,596	1,596	100.00	NA	NA	258,562	NA	NA	1,277.16
SA-B&Ch	1,596	1,596	100.00	534	33.46	260,429	211,321	20.83	2,160.73

*NA: Not available.

The strategy M-RCSRP found optimal solutions for only 15.54% of the cases (834 instances) and failed at finding feasible solutions in 47.74% of the cases. The B&Ch algorithm increased the number of optimal solutions to 1,151 (21.75%) and was not able to provide a feasible solution in only 27.88% of the cases. The SA was able to find feasible solutions for all tested instances. Also, on average, the upper bound obtained by SA was 14.48% smaller than in the B&Ch. Additionally, the SA metaheuristic improves the elapsed time by 46.98%, on average, in relation to B&Ch. Finally, providing an initial solution with SA to warm-start the B&Ch algorithm seems to be a good strategy, as SA-B&Ch reduced the upper bound by 13.86%, in relation to the B&Ch algorithm. Regarding the average elapsed time, SA-B&Ch is faster than B&Ch, but it has a higher average elapsed time than SA. Finally, regarding the gap and lower bound, the improvements were 5.38% and 45.48%, respectively, with respect to B&Ch.

Table 7.3 shows the average results of the SA-B&Ch strategy for instances with different numbers of damaged nodes and different values of Γ and α. Unsurprisingly, the cost of the solutions increases for higher values of Γ and α. Also, instances with higher Γ and α values are harder to solve. For example, for instances with $\alpha = 0$ (deterministic case), the average gap and elapsed time were 17.74% and 2,077 seconds, respectively. Such values increased to 24.29% and 2,542 seconds in the instances with $\alpha = 0.5$. Particularly, the performance of the method in the deterministic case is significantly better than in the cases when considered $\Gamma > 0$ and $\alpha > 0$.

Table 7.3: Average results of the SA-B&Ch method.

# damaged						Avg. upper	Avg. lower	Avg.	Avg. Time
nodes	Γ	α	#ins	#opt	%opt	bound	bound	gap (%)	(seconds)
	0	0.00	84	73	86.90	177,293	174,549	2.55	760.38
		0.10	84	67	79.76	182,404	178,005	3.41	892.49
	1	0.25	84	58	69.05	192,188	183,525	5.51	1,114.67
6		0.50	84	57	67.86	201,587	191,407	6.05	1,004.27
		0.10	84	59	70.24	182,952	177,365	4.05	1,008.87
	3	0.25	84	51	60.71	194,539	184,209	6.31	1,191.04
		0.50	84	48	57.14	206,148	192,624	7.56	1,213.74
	0	0.00	72	26	36.11	241,687	206,807	20.43	2,173.08
		0.10	72	18	25.00	249,614	203,136	24.62	2,624.73
	3	0.25	72	12	16.67	270,872	220,179	24.71	2,448.89
10		0.50	72	12	16.67	292,145	222,568	29.82	2,898.92
		0.10	72	12	16.67	254,934	204,514	25.78	2,900.83
	5	0.25	72	12	16.67	273,261	220,048	25.47	2,843.95
		0.50	72	11	15.28	292,884	227,089	28.46	2,970.60
	0	0.00	72	5	6.94	298,418	226,086	30.24	3,299.60
		0.10	72	3	4.17	314,151	222,169	35.28	3,552.85
	4	0.25	72	2	2.78	332,746	252,848	30.01	3,576.68
14		0.50	72	2	2.78	370,059	261,091	35.45	3,574.44
		0.10	72	2	2.78	318,012	226,490	34.78	3,548.65
	7	0.25	72	2	2.78	334,865	231,713	36.80	3,509.43
		0.50	72	2	2.78	381,221	257,597	38.43	3,590.70

7.4.2 Robustness analysis

We designed a robustness analysis based on a Monte Carlo simulation to evaluate the quality of solutions resulting from the robust optimization approach. The simulation was performed by generating $N = 10{,}000$ random uniform realizations for repair times in the half-interval $[\delta_{ki}, \delta_{ki} + \hat{\delta}_{ki}]$ for all $i \in \mathcal{V}^{\mathrm{r}}$. Let $\chi^{\alpha\Gamma}$ be the solution of the RCSRP for a given budget of uncertainty Γ and deviation α. For each run (realization) r of the simulation, we fixed the assignment and scheduling decisions (Q_i, W_{ik}, X_{ij}), the routing decisions $(P_{eij}, N_{lij}, R_{lhj})$ and the relief path decisions (Y_{ej}, V_{lj}) from the solution $\chi^{\alpha\Gamma}$ and calculate the latency $LT_i = Z_i^{\mathrm{d}} + \sum_{e\in\mathcal{E}} t_e Y_{ei}$ for each demand node $i \in \mathcal{V}^{\mathrm{d}}$.

Let $FO_0(\chi^{\alpha\Gamma})$ be the total cost of the solution $\chi^{\alpha\Gamma}$ and $LT_{i0}(\chi^{\alpha\Gamma})$ be the latency of demand node i in the solution $\chi^{\alpha\Gamma}$. Similarly, let $FO_r(\chi^{\alpha\Gamma})$ be the total cost of the run r when fixed the solution $\chi^{\alpha\Gamma}$ and $LT_{ir}(\chi^{\alpha\Gamma})$ be the latency of demand node i when fixed the solution $\chi^{\alpha\Gamma}$ in the run r. We call FO_0 and LT_{i0} as the "*promise*" total cost and latency, while FO_r and LT_{ir} are the "*actual*" total cost and latency. Then, we say that there is a cost violation in a run r of the simulation if the promise cost is smaller than the actual cost, i.e., $FO_0 < FO_r$. Similarly, we say that there is a latency violation in a run r of the simulation if the promise latency is smaller than the actual latency, i.e., $LT_{i0} < LT_{ir}$, for some demand node $i \in \mathcal{V}^{\mathrm{d}}$. Based on these definitions, we propose three performance measures for the quality of the solution $\chi^{\alpha\Gamma}$ as follows:

- Price of robustness (PoR). PoR is defined as $\frac{FO_0(\chi^{\alpha\Gamma}) - FO_0(\chi^{00})}{FO_0(\chi^{00})} \cdot 100\%$ for a given solution $\chi^{\alpha\Gamma}$, in which $FO_0(\chi^{00})$ represents the cost of the deterministic problem.

- Probability of cost violation (PCV). Given a solution $\chi^{\alpha\Gamma}$, PCV is defined as the fraction of runs (out of N) for which $FO_0(\chi^{\alpha\Gamma}) < FO_r(\chi^{\alpha\Gamma})$.

- Probability of latency violation (PLV). Given a solution $\chi^{\alpha\Gamma}$, PLV is defined as the fraction of runs (out of N) for which $LT_{i0}(\chi^{\alpha\Gamma}) < LT_{ir}(\chi^{\alpha\Gamma})$, for some demand node $i \in \mathcal{V}^{\mathsf{d}}$. We say that a solution $\chi^{\alpha\Gamma}$ is immunized against uncertainty if $LT_{i0} < LT_{ir} \forall r = 1, ..., N$.

We analyze the robustness of the solutions of the instances with 10 damaged nodes, for which we run additional experiments with $\alpha = 0.25$ and $\Gamma = 0, 1, 2, 3, 4, 5$. The average performance measures for the considered instances with different values of β and number of crews (k) are presented in Table 7.4 and Figure 7.2. Recall that β indicates the factor by which the distance between the depot and the demand nodes can increase with respect to the shortest distance. Since the instances were not solved to optimality, the price of robustness for some few cases can decrease for increasing values of Γ.

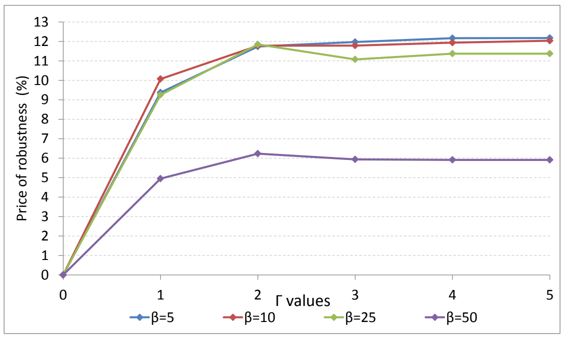
Table 7.4: Average results of the performance measures.

		β values					# vehicles			
	Γ	β =5	β =10	β =25	β =50	Avg.	$k = 1$	$k = 3$	$k = 5$	Avg.
Price of robustness (%)	0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	1	9.37	10.07	9.25	4.95	8.41	7.68	9.44	8.95	8.69
	2	11.74	11.78	11.84	6.23	10.40	10.81	11.11	9.74	10.55
	3	11.97	11.78	11.08	5.93	10.19	11.30	10.21	9.40	10.30
	4	12.17	11.93	11.37	5.91	10.34	11.59	10.33	9.41	10.44
	5	12.18	12.03	11.37	5.91	10.37	11.60	10.43	9.41	10.48
Runs with cost violation (%)	0	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	1	24.90	24.79	21.75	10.54	20.50	20.50	20.40	19.78	20.23
	2	6.31	3.21	3.19	2.73	3.86	5.29	4.40	0.04	3.24
	3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Runs with latency violation (%)	0	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	1	87.46	81.24	82.71	58.93	77.59	88.91	71.61	59.01	73.18
	2	8.58	3.36	3.32	2.75	4.50	5.96	4.42	0.08	1.78
	3	0.32	0.49	0.17	0.00	0.25	0.60	0.00	0.00	0.20
	4	0.01	0.05	0.00	0.00	0.02	0.04	0.00	0.00	0.01
	5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

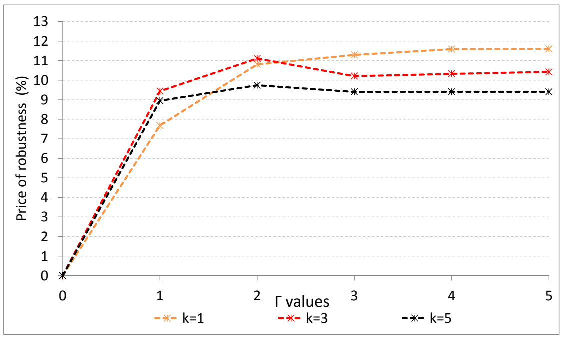
Note in Table 7.4 and Figure 7.2 that the deterministic approach ($\Gamma = 0$) fails in protecting against uncertainty for the considered instances. Indeed, solutions with $\Gamma = 0$ have cost and latency violation in 100% of the runs of the simulation. Additionally, the promise latency is significantly subestimated in relation to the actual latency value of the demand nodes. When higher values of Γ are considered, the probabilities of cost and latency violation are considerably reduced.

It is interesting to observe that we may find latency violation in a solution with no cost violation. For example, for $\Gamma = 3, \beta = 5$ the probability of cost violation is null while the probability of latency violation is 0.32% (3,200 out of 10,000 runs). In fact, the probability of latency violation is always higher than the probability of cost violation. The reason of this behavior is that, in the total cost OF_r, the high values of latency for some demand nodes are covered by some other demand nodes with small latency values.

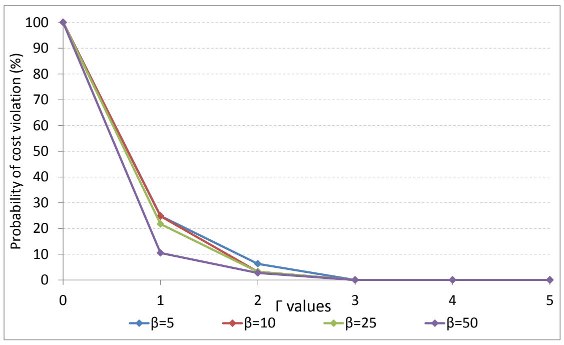
It seems that the robust solutions strike a good trade-off between decreasing the probability of cost and latency violation and not being too conservative. Note that the consideration of one repair time assuming the worst-case value ($\Gamma = 1$) reduces significantly (more than 70%) the probability of cost violation in relation to the deterministic problem ($\Gamma = 0$) while that the price



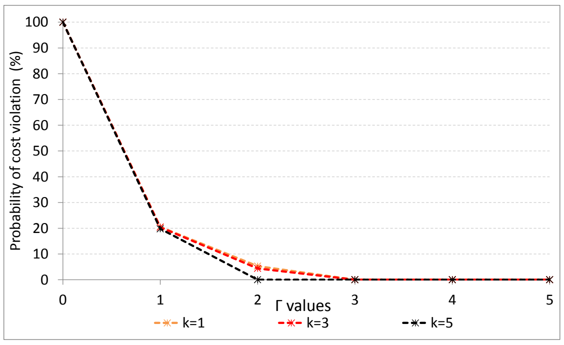
(a) PoR for diferent β values.



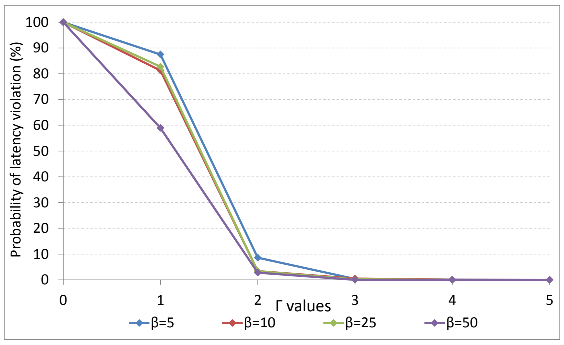
(b) PoR for diferent number of crews.



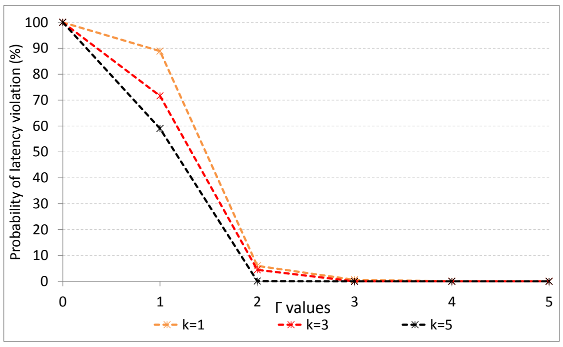
(c) RCV for diferent β values.



(d) RCV for diferent number of crews.



(e) RLV for diferent β values.



(f) RLV for diferent number of crews.

Figure 7.2: Average results of the performance measures.

of the robustness increases less than 11%. However, the impact of increasing the Γ values is less significant in the latency violation. In some cases, the probability of latency violation decreases less than 15% from $\Gamma = 0$ to $\Gamma = 1$. With $\Gamma = 5$, the robust solutions seem to be protected against cost and latency violation and the price of robustness is smaller than 13%.

The probability of cost and latency violation is smaller for higher values of β and when more vehicles are available to perform the repair operations. For $\Gamma = 1$, for example, the probability of latency violation is 87.47% when $\beta = 5$, whereas it decreases to 58.93% when $\beta = 50$. Also for $\Gamma = 1$, the probability of latency violation decreases from 88.91% when $k = 1$ to 59.01% when $k = 3$. In the RCSRP, higher values of β and crews imply, on average, in fewer damaged nodes considered in the schedules of the crews, as discussed in Section 6.4.3, and thus in fewer uncertain parameters considered in each schedule.

7.5 Final remarks of the chapter

In this chapter, we introduced the robust crew scheduling and routing problem (RCSRP) in road restoration. We extended the best model developed in Chapter 6 to consider uncertain repair times via a robust optimization. The resulting robust optimization model is based on recursive equations that verify for each damaged node in the schedule of the crews, if its repair time must be considered as one attaining its worst-case value. We also proposed a tailored B&Ch approach to solve the problem, based on the decomposition of the robust optimization model into a master problem with assignment, scheduling and relief paths decisions and a subproblem with relief path decisions. Additionally, we adapted the simulated annealing (SA) metaheuristic proposed for the single crew version of the problem (Section 5.3) to the RCSRP.

As expected, the difficulty of the problem increases after we incorporate uncertainty into repair times. The robust optimization model failed at finding even feasible solutions for almost half of the considered instances. The B&Ch algorithm found feasible solutions for most of the instances, but the average gap was higher than 35%. On the other hand, the SA was able to obtain feasible solutions for all the tested instances. Finally, the use of an initial solution from the SA to warm-start the B&Ch was the strategy that presented the best results, finding feasible solutions for all the instances and with an average gap of 20%.

The robustness analysis evidenced that the deterministic approach fails in protecting against uncertainty for the considered instances. In the robust approach, cost and latency violation measures were defined. The results indicate that the probability of cost and latency violation is significantly reduced when increasing the budget of uncertainty. The robust solutions strike a good trade-off between decreasing the probability of cost and latency violation and not increasing significantly the cost of the solutions. The results also indicate that the probability of cost and latency violation is smaller for higher values of β and when more vehicles are available to perform the repair operations.

Chapter 8

Conclusions

In this thesis we addressed the crew scheduling and routing problem (CSRP) in road restoration. The problem is typically found in post-disaster situations where the damaged network must be repaired as quickly as possible to promote an effective short-term response. The joint presence of scheduling and routing decisions explains the complexity of solving such problem, for which commercial solvers cannot be efficiently used to straightforwardly solve the available formulations, even for small-scale instances. In Chapters 4 and 5, we proposed solution approaches based on Benders decomposition for the standard deterministic variant of the problem, which considers a single crew available to perform the restoration activities. Chapter 4 developed the first exact BBC solution approach that is able to obtain feasible solutions and lower bounds for all instances from the literature, including very large-scale instances. We employed feasibility cuts, multiple optimality cuts, and specialized valid inequalities, which have enhanced the performance of the BBC approach. The graph reduction strategy for deriving cuts from networks with fewer nodes also improves the performance of the BBC.

In Chapter 5 we proposed two metaheuristics, an enhanced BBC algorithm and a hybrid approach (HBBC) to solve the CSRP. The metaheuristics are the first genetic algorithm and simulated annealing proposed for the CSRP. They are based on the decomposition of the problem into smaller subproblems and the use of specialized algorithms to evaluate the candidate solutions. The BBC is based on an improved Benders reformulation of the problem and enhances the decomposition proposed in Chapter 4 by using a different variable partitioning scheme. The HBBC is an exact hybrid method that uses a metaheuristic to obtain good-quality solutions at early stages of the search tree as well as to improve the performance of solving the master problem by exploring the neighborhood of the incumbent solutions to generate more effective Benders cuts. The results of extensive computational experiments with benchmark instances provide evidence that the combination of the metaheuristic with the BBC, resulting in the hybrid algorithm HBBC, significantly reduces the objective value of the solutions and the time spent to find good-quality solutions. The improvements obtained with the proposed approaches may have great value to aid decision making in practice. The reduction in the value of the solutions directly impacts the time at which the demand nodes are accessible from the supply node, thus reducing the time that victims in the affected areas wait for supplies, evacuation, rescue and medical assistance.

In Chapter 6 we introduced a new deterministic variant of the problem that addresses multiple heterogeneous crews available to perform the restoration activities. Three novel mathematical formulations and new valid inequalities were devised for this new variant of the problem. The first two formulations are based on the three-index and two-index formulation of the VRP. The third formulation eliminates a few variables and introduces new constraints to reduce the symmetry in the solutions of the problem. The valid inequalities are based on the dominance of the paths between nodes in the damaged network. The three mathematical formulations showed improvement with the addition of these valid inequalities. From the analysis of a practical case, we explicitly provide insights for the decision-makers in practice. The results indicate that, as expected, the use of more crews to solve the problem significantly reduces the time required to restore the accessible of the network. The multiple crews mainly affect the worst-case accessibility time between the demand nodes, thus providing more equitable accessibility times. Usually, the farthest damaged nodes were allocated to the crews that had the shortest travel time, while the damaged nodes with higher repair time were allocated to crews that can perform a faster restoration.

Finally, in Chapter 7 we extended the best model developed in Chapter 6 to incorporate uncertain repair times via robust optimization, based on recursive equations recently proposed to consider uncertainty in the vehicle routing problem. The difficulty of the problem increases when we incorporate uncertainty into the repair times. A Benders decomposition-based method was developed to solve the robust CSRP (RCSRP). The metaheuristic approaches developed in Chapter 5 for the single crew version of the problem were adapted to the RCSRP and the SA algorithm performed significantly better than GA in this variant. The proposed approaches were able to obtain robust feasible solutions for all the considered instances. The robustness analysis evidenced that the deterministic approach fails in protecting against uncertainty for the considered instances. On the other hand, the robust solutions strike a good trade-off between decreasing the probability of cost and latency violation and not increasing significantly the cost of the solutions.

8.1 Future research

There are several possible future research directions for the continuity of this study, some of them are described as follows:

- **Integrated relief distribution decisions.** In the literature, a few authors have considered the single crew scheduling and routing problem with integrated relief distribution decisions (Shin et al., 2019). It would be interesting to integrate the relief distribution decisions in the robust heterogeneous multicrew scheduling and routing problem.

- **Multiple depots.** In future researches, we can also consider the integration of decisions related to the location of multiple depots as well as the allocation of crews to depots.

- **Uncertainty in other parameters.** Parameters as the travel time and demand could be particularly difficult to estimate in post-disaster situation, and they can have a significant impact in the solutions of the problem, particularly if relief distribution decisions are considered together with road restoration decisions. Hence, an interesting topic of research is to consider the uncertainty in these parameters of the problem.

- **Dynamic approaches**. An alternative way to reduce the impact of ignoring the uncertainties in the problem is the development of dynamic approaches that can incorporate information about the disaster in real time, thus providing solutions to the problem according to the most updated information.

- **Solution methods.** Alternative decomposition and formulations of the problem could be explored. Particularly, solution methods based on column generation could have a good performance at solving the addressed variants of the CSRP. The similarity of the CSRP with the vehicle routing problem could be a starting point to develop this kind of methods. Also, the properties of path dominance defined for the problem could help in the development of labeling algorithms.

- **Alternative objective functions.** Alternative objective functions are a recent trend of research in the humanitarian logistics field. Social concern objective functions are important in the post-disaster operation to perform fair, equitable and effective response operations. These functions could be explored in the CSRP, specially in variants with integrated relief distribution decisions.

- **Practical applications of the proposed models and methods.** The application of the proposed models and solution methods in real disaster situations could definitively be a good direction of research. The proposed models and methods are a first step to further develop faster solution approaches and user-friendly decisions-support tools that can help decision-makers in the aftermath of disasters.

Appendix A

VIs related to routing decisions for the second and third MCSRP formulations

Inequalities (A.1)-(A.4) are the valid inequalities related to routing decisions for MCSRP2. Inequalities (A.1),(A.2) are equivalent to inequalities (6.75),(6.76) to select non-dominated paths over dominated paths. Inequalities (A.3), (A.4) are equivalent to inequalities (6.78), (6.79) to set lower bounds for the variables $Z^{\mathtt{r}}_{j}$.

$$\sum_{e\in\mathcal{E}_{f^*_{kij}}} P_{eij} + \sum_{l\in\mathcal{V}^{\mathtt{u}}_{f^*_{kij}}} N^{\mathtt{u}}_{lij} \geq (|\mathcal{E}_{f^*_{kij}}| + |\mathcal{V}^{\mathtt{u}}_{f^*_{kij}}|)\cdot(X'_{ij} + W'_{kj} - 1),$$

$$\forall k\in\mathcal{K}, i\in\mathcal{V}^{\mathtt{r}}_0, j\in\mathcal{V}^{\mathtt{r}} : \mathcal{F}^{\mathtt{r}*}_{kij}\neq\emptyset, t_{kf^*_{kij}} = t_{kp^*_{kij}}, \tag{A.1}$$

$$(|\mathcal{E}|+|\mathcal{V}|)\cdot(2+|\mathcal{V}^{\mathtt{r}}_p| - X'_{ij} - W'_{kj} - \sum_{h\in\mathcal{R}}\sum_{l\in\mathcal{V}^{\mathtt{r}}_p} N^{\mathtt{r}}_{lhij}) \geq \sum_{e\in\mathcal{E}\setminus\bigcup\limits_{p'\in\mathcal{D}_p}\mathcal{E}_{p'}} P_{eij} + \sum_{l\in\mathcal{V}\setminus\bigcup\limits_{p'\in\mathcal{D}_p}\mathcal{V}_{p'}} N^{\mathtt{u}}_{lij},$$

$$\forall\ k\in\mathcal{K},\ i\in\mathcal{V}^{\mathtt{r}}_0,\ j\in\mathcal{V}^{\mathtt{r}}, p\in\mathcal{S}^{\mathtt{r}}_{kij} : (\mathcal{F}^{\mathtt{r}*}_{kij}=\emptyset)\vee(\mathcal{F}^{\mathtt{r}*}_{k0j}\neq\emptyset, p\neq f^*_{kij}, t_{kf^*_{kij}} > t_{kp^*_{kij}}), \tag{A.2}$$

$$Z^{\mathtt{r}}_j \geq (t_{kp^*_{k0j}} + \delta_{kj})\cdot X'_{0j} + \sum_{i\in\mathcal{V}^{\mathtt{r}}}((t_{kp^*_{k0i}} + \delta_{ki} + t_{kp^*_{kij}} + \delta_{kj})\cdot X'_{ij}) - M\cdot(1-W'_{kj}),$$

$$\forall\, k\in\mathcal{K},\ j\in\mathcal{V}^{\mathtt{r}}, \tag{A.3}$$

$$Z^{\mathtt{r}}_j \geq Z^{\mathtt{r}}_i + t_{kp^*_{kij}} + \delta_{kj} - M\cdot(2 - W'_{kj} - X'_{ij}), \forall\, k\in\mathcal{K},\ i\in\mathcal{V}^{\mathtt{r}}_0,\ j\in\mathcal{V}^{\mathtt{r}}. \tag{A.4}$$

Inequalities (A.3),(A.4) are the same for MCSRP3. Additionally, we can state the following valid inequalities for this formulation.

$$\sum_{e\in\mathcal{E}_{f^*_{k0j}}} P_{e0j} + \sum_{l\in\mathcal{V}^{\mathtt{u}}_{f^*_{k0j}}} N^{\mathtt{u}}_{l0j} \geq (|\mathcal{E}_{f^*_{k0j}}| + |\mathcal{V}^{\mathtt{u}}_{f^*_{k0j}}|)\cdot(X'_{0j} + W'_{kj} - 1),$$

$$\forall k\in\mathcal{K}, j\in\mathcal{V}^{\mathtt{r}} : (\mathcal{F}^{\mathtt{r}*}_{k0j}\neq\emptyset, t_{kf^*_{k0j}} = t_{kp^*_{k0j}}), \tag{A.5}$$

$$(|\mathcal{E}|+|\mathcal{V}|)\cdot(2+|\mathcal{V}^{\mathtt{r}}_p| - X'_{0j} - W'_{kj} - \sum_{h\in\mathcal{R}}\sum_{l\in\mathcal{V}^{\mathtt{r}}_p} N^{\mathtt{r}}_{lh0j}) \geq \sum_{e\in\mathcal{E}\setminus\bigcup\limits_{p'\in\mathcal{D}_p}\mathcal{E}_{p'}} P_{e0j} + \sum_{l\in\mathcal{V}\setminus\bigcup\limits_{p'\in\mathcal{D}_p}\mathcal{V}_{p'}} N^{\mathtt{u}}_{l0j},$$

$$\forall\ k\in\mathcal{K},\ j\in\mathcal{V}^{\mathtt{r}}, p\in\mathcal{S}^{\mathtt{r}}_{k0j} : (\mathcal{F}^{\mathtt{r}*}_{k0j}=\emptyset)\vee(\mathcal{F}^{\mathtt{r}*}_{k0j}\neq\emptyset, p\neq f^*_{k0j}, t_{kf^*_{k0j}} > t_{kp^*_{k0j}}) \tag{A.6}$$

$$\sum_{e \in \mathcal{E}_{f^*_{kij}}} P_{eij} + \sum_{l \in \mathcal{V}^{\mathrm{u}}_{f^*_{kij}}} N^{\mathrm{u}}_{lij} \geq (|\mathcal{E}_{f^*_{kij}}| + |\mathcal{V}^{\mathrm{u}}_{f^*_{kij}}|) \cdot (X'_{ij} + X'_{ji} + W'_{kj} - 1),$$

$$\forall k \in \mathcal{K}, i \in \mathcal{V}^{\mathrm{r}}, j \in \mathcal{V}^{\mathrm{r}} : (\mathcal{F}^{\mathrm{r}*}_{kij} \neq \emptyset, t_{kf^*_{kij}} = t_{kp^*_{kij}}, i < j), \tag{A.7}$$

$$(|\mathcal{E}| + |\mathcal{V}|) \cdot (2 + |\mathcal{V}^{\mathrm{r}}_p| - X'_{ij} - X'_{ji} - W'_{kj} - \sum_{h \in \mathcal{R}} \sum_{l \in \mathcal{V}^{\mathrm{r}}_p} N^{\mathrm{r}}_{lhij}) \geq \sum_{e \in \mathcal{E} \setminus \bigcup\limits_{p' \in \mathcal{D}_p} \mathcal{E}_{p'}} P_{eij} + \sum_{l \in \mathcal{V} \setminus \bigcup\limits_{p' \in \mathcal{D}_p} \mathcal{V}_{p'}} N^{\mathrm{u}}_{lij},$$

$$\forall\ k \in \mathcal{K},\ i \in \mathcal{V}^{\mathrm{r}},\ j \in \mathcal{V}^{\mathrm{r}}, p \in \mathcal{S}^{\mathrm{r}}_{kij} : (\mathcal{F}^{\mathrm{r}*}_{kij} = \emptyset, i < j) \vee (\mathcal{F}^{\mathrm{r}*}_{kij} \neq \emptyset, p \neq f^*_{kij}, t_{kf^*_{kij}} > t_{kp^*_{kij}}, i < j). \tag{A.8}$$

Inequalities (A.5)-(A.6) are defined for $i = 0$ and are equivalent to inequalities (A.1)-(A.2). Inequalities (A.7)-(A.8) are defined for $i \neq 0$ and are equivalent to inequalities (A.1)-(A.2).

Appendix B

Instance generation from the real case disaster

The Megadisaster of the Serrana region of Rio de Janeiro affected different cities and caused traffic blockages due to landslides and flooding in different points of six of the main highways (Rio de Janeiro, 2011). Initially, we have assumed one damaged node in each of the affected highways. Since some of the highways were affected in more than one location, we also generate instances considering a higher number of damaged nodes. The demand in the different cities is shown in Table B.1. For cities 1-13 the demand is equal to the number of affected people in the disaster in 2011. There was no reported demand for cities 14-20. Thus, we generate the demand of the cities 14-20 as a proportion of their total population.

Table B.1: Demand of the affected cities (Rio de Janeiro, 2011).

	City	Demand
1	Nova Friburgo	6,637
2	Cordeiro	43
3	Macuco	52
4	Bom Jardim	2,669
5	São Sebastião do alto	107
6	Santa Maria Madalena	328
7	Petrópolis	7,214
8	São José do Vale do Rio Preto	395
9	Três Rios	9
10	Areal	737
11	Sapucaia	40
12	Teresópolis	17,029
13	Sumidouro	801
14	Conceição de Macabu	782
15	Casimiro de Abreu	1305
16	Trajano de Moraes	454
17	Cachoeiras de Macacu	2005
18	Duas Barras	406
19	Cantagalo	731
20	Carmo	643
Total		42,387

The damaged network based on the real disaster is shown in Figure B.1. The real distance of the arcs was calculated via Google Maps®. The travel time for a single crew was computed based on the distance and assuming a speed of 25 kilometers per hour for the crew. We have not found information about the repair time of the damaged nodes. This way, for a single crew, we have generated the repair time based on the travel time (which is proportional to the distance) on the highway where the damaged node is located. The idea was to generate higher repair times for longer highways, in general higher than the travel times. Thus, for a single crew, the repair time of a damaged node i located in a given arc (highway) e was randomly generated from the interval $[2 \cdot \text{time}_e\,, 5 \cdot \text{time}_e]$, in which time_e is the travel time on arc e. Travel and repair times for the multiple crews were generated from the values of a single crew, as described in Section 6.4.1. The maximum distance from the depot to the demand nodes (l_i^{d}) was calculated as in the literature (Maya-Duque et al., 2016; Moreno et al., 2019), using a parameter β that indicates the factor by which the distance between the depot and the demand nodes can increase with respect to the shortest distance. Thus, $l_i^{\mathsf{d}} = (1+\beta) \cdot dist_{0i}$, in which $dist_{0i}$ is the shortest distance between the depot and the demand node i. We consider six values for β (0.05, 0.1, 0.25, 0.5, 1, ∞), where ∞ represents a sufficiently large number indicating that the constraint imposing the maximum distance l_i^{d} is relaxed.

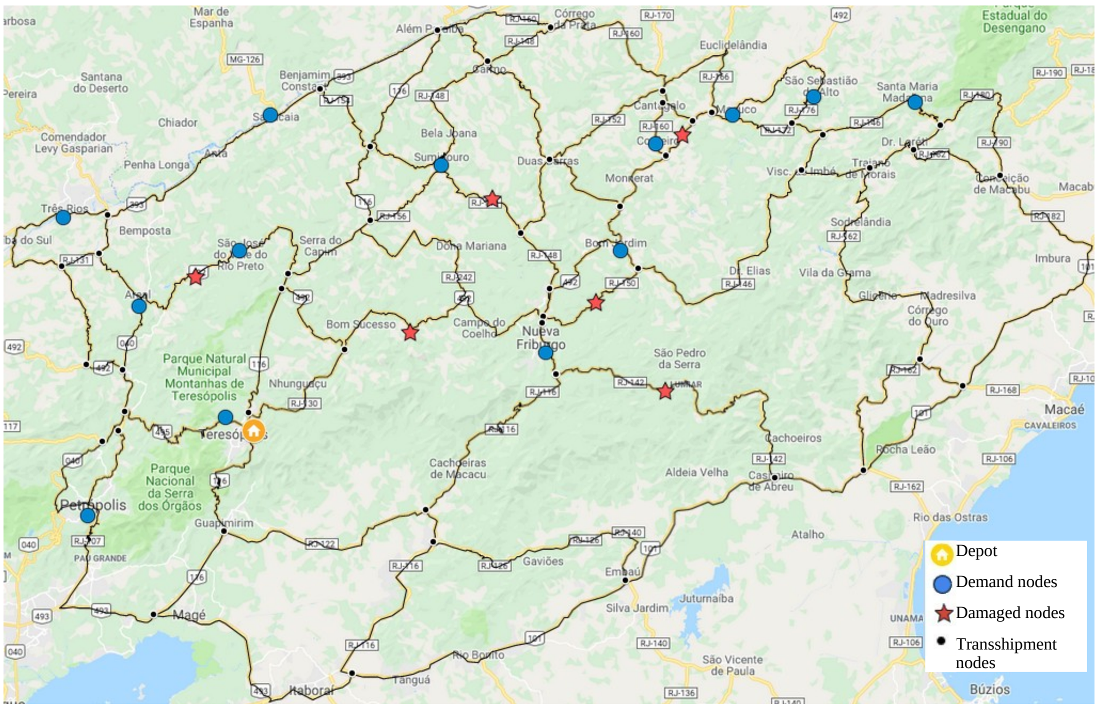


Figure B.1: Damaged network based on the real disaster.

From the damaged network of Figure B.1, we generate seven classes of instances. The class CS0 has 6 damaged nodes located in the highways originally affected by the disaster in 2011. Furthermore, class CS0 considers the first 13 cities (cities 1-13) as the demand nodes. Considering the six values for β, a total of 6 instances were generated in class CS0. In class CS1, we have generated 3 damaged networks considering 6, 10 and 14 damaged nodes. Damaged networks in a same class share some damaged nodes. Let CS1-$|\mathcal{V}^r|$ be the damaged network of class CS1 with $|\mathcal{V}^r|$ damaged nodes. In the damaged network CS1-6, the 6 damaged nodes were located in 6 randomly selected arcs. Similar to the categorization used in Akbari and Salman (2017b), we divide the arcs into three groups according to their proximity to the affected areas, as high, medium and low-risk arcs. Then, for the location of a given damaged node, the probability of selecting a high, medium and low-risk arc was set to 0.15, 0.35, and 0.5, respectively. The damaged network CS1-10 considers the 6 damaged nodes in CS1-6 and 4 additional randomly located damaged nodes. Similarly, the damaged network CS1-14 considers the 10 damaged nodes in CS1-10 and 4 additional randomly located damaged nodes. For each one of the three damaged networks, the six values of β were considered, totaling 18 instances in class CS1. The same procedure was used to generate classes CS2 and CS3. Classes CS1-CS3 consider the first 13 cities in Table B.1 as the demand nodes. Finally, classes CS4-CS6 are based on the the same damaged networks of classes CS1-CS3, but considering all the cities presented in Table B.1 as the demand nodes. Thus, there are 114 instances based on the real-world case. We run experiments with 1, 3 and 5 crews, totaling 342 CS instances.

Appendix C

Additional computational results for the SCSRP

Tables C.1 and C.2 present the average results of the BBC and HBBC solution methods for different classes of instances grouped according to the size of the network.

Table C.1: Comparison of the exact BBC solution approaches.

Solution method	Instance classes	#ins	#opt	%opt	Avg. upper bound	Avg. lower bound	Avg. gap (%)	Avg. time (sec.)	Avg. best time[1](sec.)
BBC1	1, 2, 3	60	43	71.67	30,502	24,411	8.14	1,139.93	1,044.04
	4, 5, 6	60	30	50.00	6,336	10,325	9.40	1,839.95	1,680.40
	7, 8, 9	60	24	40.00	19,330	15,950	16.98	2,193.77	2,023.08
	10, 11, 12	60	41	68.33	38,539	31,004	7.31	1,236.69	1,174.83
	13, 14, 15	60	20	33.33	8,196	19,246	3.06	2,526.45	2,328.36
	16, 17, 18	30	8	26.67	297,915	1,768	73.33	2,782.49	2,503.31
	19, 20, 21	30	2	6.67	171,547	12,581	63.13	2,959.21	2,765.48
	22, 23, 24	30	1	3.33	538,291	26,534	72.45	3,407.97	3,137.27
	All	390	169	43.33	113,048	17,427	28.73	2,238.42	2,058.13
BBC2	1, 2, 3	60	43	71.67	32,151	25,295	6.72	1,073.87	1,056.54
	4, 5, 6	60	30	50.00	8,627	6,520	7.74	1,813.94	1,573.02
	7, 8, 9	60	24	40.00	15,232	7,226	13.91	2,167.96	1,895.85
	10, 11, 12	60	43	71.67	38,694	31,977	6.29	1,057.97	1,007.75
	13, 14, 15	60	20	33.33	8,196	7,881	2.29	2,472.36	2,194.44
	16, 17, 18	30	8	26.67	297,915	32,875	51.86	2,754.94	2,258.25
	19, 20, 21	30	2	6.67	171,547	15,830	55.38	2,929.91	2,615.22
	22, 23, 24	30	1	3.33	538,291	36,207	66.48	3,374.23	2,987.47
	All	390	171	43.85	110,763	20,494	25.03	2201.55	1,889.79
BBC3	1, 2, 3	60	43	71.67	32,151	25,328	6.71	1,080.51	1,003.94
	4, 5, 6	60	30	50.00	8,704	6,528	7.76	1,815.49	1,595.92
	7, 8, 9	60	24	40.00	15,306	7,222	13.93	2,224.59	2,127.39
	10, 11, 12	60	43	71.67	38,694	31,978	6.29	1,059.75	1,020.17
	13, 14, 15	60	20	33.33	8,205	7,881	2.29	2,474.54	2,030.34
	16, 17, 18	30	8	26.67	295,963	32,881	51.78	2,758.30	2,250.02
	19, 20, 21	30	2	6.67	166,133	15,834	55.30	2,926.88	2,606.95
	22, 23, 24	30	1	3.33	526,655	36,212	65.85	3,324.88	3,051.20
	All	390	171	43.85	110,291	20,499	24.99	2,203.59	1,891.82
BBC3*	1, 2, 3	60	43	71.67	32,151	26,098	5.51	1,153.71	1,135.08
	4, 5, 6	60	30	50.00	8,897	7,044	6.50	1,814.71	1,598.69
	7, 8, 9	60	24	40.00	15,159	7,646	12.68	2,098.03	1,734.70
	10, 11, 12	60	43	71.67	38,694	32,564	5.56	1,101.18	1,052.99
	13, 14, 15	60	20	33.33	8,541	8,131	2.20	2,470.04	2,192.38
	16, 17, 18	30	8	26.67	357,327	33,251	53.27	2,751.29	2,205.26
	19, 20, 21	30	2	6.67	198,129	16,378	54.31	2,928.72	2,624.16
	22, 23, 24	30	1	3.33	580,125	36,540	66.57	3,424.34	3,033.84
	All	390	171	43.85	117,355	21,157	24.67	2,215.35	1,910.44

[1] Time spent to find the best upper bound.

Table C.2: Comparison of the exact HBBC solution approaches.

Solution method	Instance classes	#ins	#opt	%opt	Avg. upper bound	Avg. lower bound	Avg. gap (%)	Avg. time (sec.)	Avg. best time[1](sec.)
HBBC1	1, 2, 3	60	48	80.00	31,232	25,355	6.06	753.86	0.28
	4, 5, 6	60	34	56.67	19,267	11,038	17.12	1,562.06	1.62
	7, 8, 9	60	29	48.33	32,326	16,930	23.99	1,872.77	9.09
	10, 11, 12	60	47	78.33	38,345	31,782	6.33	815.79	0.63
	13, 14, 15	60	21	35.00	52,084	26,009	27.05	2,343.41	24.79
	16, 17, 18	30	10	33.33	55,102	32,789	23.20	2,402.79	88.30
	19, 20, 21	30	8	26.67	49,574	15,830	43.12	2,644.80	268.34
	22, 23, 24	30	4	13.33	99,715	36,271	45.67	3,172.22	340.58
	All	390	201	51.54	42,377	23,625	21.01	1,762.74	59.23
HBBC2	1, 2, 3	60	51	85.00	31,232	26,645	4.48	634.24	0.19
	4, 5, 6	60	38	63.33	19,038	11,527	15.04	1,331.01	2.42
	7, 8, 9	60	30	50.00	32,154	17,132	23.06	1,824.65	7.15
	10, 11, 12	60	47	78.33	38,345	32,075	5.96	817.21	0.88
	13, 14, 15	60	26	43.33	51,997	26,584	24.81	2,063.03	30.03
	16, 17, 18	30	12	40.00	54,637	32,985	22.25	2,174.72	103.51
	19, 20, 21	30	8	26.67	49,078	15,942	42.17	2,648.38	356.15
	22, 23, 24	30	3	10.00	99,181	36,309	45.60	3,241.94	661.24
	All	390	215	55.13	42,187	24,089	19.75	1,646.56	92.48
HBBC2*	1, 2, 3	60	51	85.00	31,232	27,396	3.54	630.57	0.32
	4, 5, 6	60	38	63.33	19,006	12,340	12.54	1,333.82	2.57
	7, 8, 9	60	30	50.00	32,145	18,209	20.86	1,864.03	13.77
	10, 11, 12	60	47	78.33	38,345	32,755	5.23	806.55	1.48
	13, 14, 15	60	26	43.33	51,963	27,505	23.57	2,068.16	32.22
	16, 17, 18	30	12	40.00	54,637	33,369	20.86	2,172.90	155.64
	19, 20, 21	30	8	26.67	48,934	16,482	40.07	2,648.29	327.90
	22, 23, 24	30	3	10.00	99,096	36,854	43.56	3,240.62	499.23
	All	390	215	55.13	42,158	24,855	18.15	1,651.39	83.35
GR-HBBC2*	1, 2, 3	60	54	90.00	31,232	30,076	1.02	713.16	0.14
	4, 5, 6	60	42	70.00	19,006	15,667	5.27	1,086.71	2.20
	7, 8, 9	60	32	53.33	32,145	25,431	9.49	1,744.60	5.51
	10, 11, 12	60	49	81.67	38,345	36,122	2.10	737.94	2.88
	13, 14, 15	60	31	51.67	51,963	36,521	12.72	1,746.30	74.48
	16, 17, 18	30	13	43.33	54,588	36,713	14.88	2,047.63	301.98
	19, 20, 21	30	12	40.00	48,757	19,810	31.94	2,173.21	290.58
	22, 23, 24	30	6	20.00	98,812	43,716	33.71	2,881.99	550.99
	All	390	239	61.28	42,118	29,836	10.90	1,473.87	101.07

[1] Time spent to find the best upper bound.

Appendix D

Additional computational results for the MCSRP

Tables D.1 and D.2 show the average results of the three proposed formulations with and without the valid inequalities for different numbers of crews. The average upper bound, lower bound, and gap presented in columns 9 to 11 are computed using all the instances with feasible solutions and hence they cannot be compared directly for different approaches since some of them do not return feasible solutions for some instances. On the other hand, the average upper bound, lower bound and gap presented in columns 6 to 8 are computed using only the results of instances for which all solution approaches found feasible solutions and hence they can be directly compared. These values confirm the discussion presented in Section 6.4.2, showing that the VIs help to improve the average gap, upper bound and lower bound of the solutions.

Tables D.3 and D.4 show the average results of the MCSRP3+VIs and MCSRP3+VIs* strategies, respectively, for the different classes of instances.

Table D.1: Average results of the three MCSRP models with and without the VIs for the instances from the literature.

					Common feasible instances[1]			All feasible instances[2]		
Solution method	# Crew	%Feas	%Opt	Avg. time (seconds)	Avg. upper bound	Avg. lower bound	Avg. gap (%)	Avg. upper bound	Avg. lower bound	Avg. gap (%)
MCSRP1	1	84.72	75.00	994.15	8,510	7,103	3.21	9,091	7,119	4.31
	3	84.03	72.22	1,088.66	3,976	3,244	4.07	3,976	3,244	4.07
	5	84.72	72.22	1,113.14	4,000	2,903	4.86	3,950	2,872	4.78
MCSRP2	1	88.89	76.39	926.83	8,456	7,905	1.16	10,098	8,026	3.82
	3	87.50	65.28	1,275.78	3,953	3,194	3.83	4,387	3,304	5.48
	5	86.81	66.67	1,310.07	3,702	2,896	4.76	3,935	2,884	6.49
MCSRP3	1	88.89	77.08	952.37	8,609	7,263	2.45	10,447	7,291	5.33
	3	87.50	68.06	1,208.17	4,055	2,904	6.12	4,352	3,015	7.04
	5	88.19	66.67	1,293.73	3,920	2,796	5.88	4,129	2,752	8.28
MCSRP1 + VIs	1	100.00	86.11	512.33	8,456	8,455	0.00	11,168	9,164	5.61
	3	97.92	86.11	552.49	3,924	3,750	0.75	5,039	4,296	3.76
	5	97.92	88.89	473.48	3,597	3,596	0.00	4,187	3,754	3.45
MCSRP2 + VIs	1	100.00	87.50	466.59	8,456	8,456	0.00	11,675	9,209	5.64
	3	100.00	86.81	574.45	3,923	3,913	0.05	6,615	4,489	5.87
	5	100.00	84.72	669.77	3,598	3,552	0.34	4,616	3,891	5.30
MCSRP3 + VIs	1	100.00	87.50	578.49	8,456	8,453	0.01	11,251	9,231	5.42
	3	100.00	81.25	734.53	3,930	3,811	0.50	5,562	4,420	5.47
	5	100.00	81.94	710.60	3,626	3,464	0.80	4,734	3,725	6.35

[1] Values based on solutions of instances that are feasible in all solution approaches.

[2] Values based on all the instances with feasible solutions for a given approach.

Table D.2: Average results of the three MCSRP models with and without the VIs for the instances based on the real case.

					Common feasible instances[1]			All feasible instances[2]		
Solution method	# Crew	%Feas	%Opt	Avg. time (seconds)	Avg. upper bound	Avg. lower bound	Avg. gap (%)	Avg. upper bound	Avg. lower bound	Avg. gap (%)
MCSRP1	1	42.11	39.47	2,223.74	78,474	75,549	3.14	78,474	75,549	3.14
	3	31.58	28.07	2,559.94	35,929	33,902	5.02	35,454	33,482	4.88
	5	32.46	31.58	2,496.00	35,560	34,669	1.58	35,560	34,669	1.58
MCSRP2	1	42.11	38.60	2,156.08	78,467	75,944	2.10	78,467	75,944	2.10
	3	34.21	28.95	2,534.52	35,929	33,205	6.26	37,113	34,599	5.78
	5	34.21	31.58	2,426.17	35,560	34,669	1.58	38,413	37,568	1.50
MCSRP3	1	42.11	38.60	2,156.10	78,467	75,944	2.10	78,467	75,944	2.10
	3	33.33	29.82	2,532.74	35,929	34,483	3.24	38,069	36,699	3.07
	5	35.09	32.46	2,395.94	35,560	34,669	1.58	37,912	37,088	1.46
MCSRP1 + VIs	1	93.86	68.42	1,264.27	78,467	78,467	0.00	125,474	111,894	5.55
	3	93.86	76.32	1,015.12	35,929	35,929	0.00	69,168	65,841	3.87
	5	89.47	67.54	1,302.04	35,560	35,560	0.00	65,888	59,970	7.60
MCSRP2 + VIs	1	97.37	71.93	1,225.11	78,467	78,467	0.00	125,612	113,517	5.13
	3	94.74	58.77	1,670.53	35,929	35,929	0.00	71,501	63,627	7.36
	5	95.61	52.63	1,873.20	35,560	35,560	0.00	68,479	58,362	10.78
MCSRP3 + VIs	1	100.00	74.56	1,247.62	78,467	78,467	0.00	127,720	115,219	5.26
	3	100.00	55.26	1,884.45	35,929	35,929	0.00	80,824	64,543	10.78
	5	100.00	52.63	2,024.23	35,560	35,560	0.00	72,640	59,176	13.05

[1] Values based on solutions of instances that are feasible in all solution approaches.

[2] Values based on all the instances with feasible solutions for a given approach.

Table D.3: Average results of the MCSRP3+VIs strategy for the different instance classes.

Instances	#Crew	#Ins	#Opt	%Opt	Avg. upper bound	Avg. lower bound	Avg. gap (%)	Avg. time (seconds)
L1, L2, L3	1	36	36	100.00	14,121	14,118	0.00	180.65
	3	36	33	91.67	6,708	6,453	0.69	407.86
	5	36	32	88.89	6,114	5,768	1.09	406.77
L4, L5, L6	1	36	36	100.00	11,101	11,095	0.00	298.64
	3	36	31	86.11	4,811	4,669	0.99	533.29
	5	36	31	86.11	4,371	4,178	1.56	626.29
L7, L8, L9	1	36	27	75.00	6,395	4,789	7.80	902.51
	3	36	27	75.00	3,615	2,956	5.62	902.10
	5	36	29	80.56	2,751	2,206	6.44	804.15
L10, L11, L12	1	36	27	75.00	13,389	6,920	13.87	932.17
	3	36	26	72.22	7,114	3,603	14.58	1,094.88
	5	36	26	72.22	5,699	2,747	16.30	1,005.20
CS0	1	6	6	100.00	19,686	19,686	0.00	0.12
	2	6	6	100.00	18,995	18,995	0.00	1.83
	3	6	6	100.00	18,976	18,975	0.00	1.96
CS1, CS2, CS3	1	54	43	79.63	106,250	101,232	2.94	1,125.37
	2	54	29	53.70	69,936	57,229	9.92	2,060.88
	3	54	29	53.70	65,193	53,880	11.83	2,051.85
CS4, CS5, CS6	1	54	36	66.67	161,192	139,821	8.16	1,508.48
	2	54	28	51.85	98,581	76,918	12.84	1,917.19
	3	54	25	46.30	86,049	68,939	15.71	2,221.30

Table D.4: Average results of the MCSRP3+VIs* strategy for the different instance classes.

Instances	#Crew	#Ins	#Opt	%Opt	Avg. upper bound	Avg. lower bound	Avg. gap (%)	Avg. time (seconds)
L1, L2, L3	1	36	36	100.00	14,121	14,120	0.00	191.86
	3	36	36	100.00	6,684	6,684	0.00	413.08
	5	36	36	100.00	6,026	6,025	0.00	406.98
L4, L5, L6	1	36	36	100.00	11,101	11,101	0.00	326.91
	3	36	36	100.00	4,811	4,811	0.00	622.01
	5	36	36	100.00	4,361	4,361	0.00	639.57
L7, L8, L9	1	36	27	75.00	6,242	5,103	5.50	903.19
	3	36	31	86.11	3,464	3,197	2.32	904.10
	5	36	30	83.33	2,644	2,315	3.88	888.68
L10, L11, L12	1	36	27	75.00	12,951	8,916	9.24	1,112.49
	3	36	26	72.22	6,325	4,334	10.13	1,106.88
	5	36	26	72.22	4,530	3,207	10.26	1,117.06
CS0	1	6	6	100.00	19,686	19,686	0.00	0.22
	2	6	6	100.00	18,995	18,995	0.00	1.84
	3	6	6	100.00	18,976	18,976	0.00	2.41
CS1, CS2, CS3	1	54	43	79.63	106,142	101,807	2.50	1,292.92
	2	54	40	74.07	63,816	60,840	3.20	2,078.90
	3	54	34	62.96	61,371	57,419	4.69	2,139.12
CS4, CS5, CS6	1	54	37	68.52	158,404	142,114	6.27	1,654.53
	2	54	38	70.37	90,233	83,260	5.62	2,039.23
	3	54	31	57.41	84,047	75,483	7.53	2,259.14

Bibliography

Adulyasak, Y., Cordeau, J.-F., and Jans, R. (2015). Benders Decomposition for Production Routing Under Demand Uncertainty. *Operations Research*, 63(4):851–867.

Ahmadi, M., Seifi, A., and Tootooni, B. (2015). A humanitarian logistics model for disaster relief operation considering network failure and standard relief time: A case study on San Francisco district. *Transportation Research Part E: Logistics and Transportation Review*, 75:145–163.

Ajam, M., Akbari, V., and Salman, F. S. (2019). Minimizing latency in post-disaster road clearance operations. *European Journal of Operational Research*, 277(3):1098–1112.

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

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

Alem, D., Clark, A., and Moreno, A. (2016). Stochastic network models for logistics planning in disaster relief. *European Journal of Operational Research*, 255(1):187–206.

Alem, D., Curcio, E., Amorim, P., and Almada-Lobo, B. (2018). A computational study of the general lot-sizing and scheduling model under demand uncertainty via robust and stochastic approaches. *Computers & Operations Research*, 90:125–141.

Altay, N. and Green, W. G. (2006). OR/MS research in disaster operations management. *European Journal of Operational Research*, 175(1):475–493.

Alvarez, A., Munari, P., and Morabito, R. (2018). Iterated local search and simulated annealing algorithms for the inventory routing problem. *International Transactions in Operational Research*, 25(6):1785–1809.

Applegate, D., Bixby, R., Chvátal, V., and Cook, W. (2018). Concorde TSP solver. www.tsp.gatech.edu/concorde.html. Accessed on 01/02/2018.

Arslan, O. and Karaşan, O. E. (2016). A Benders decomposition approach for the charging station location problem with plug-in hybrid electric vehicles. *Transportation Research Part B: Methodological*, 93:1339–1351.

Bai, R., Blazewicz, J., Burke, E. K., Kendall, G., and McCollum, B. (2012). A simulated annealing hyper-heuristic methodology for flexible decision support. *4OR*, 10(1):43–66.

Balin, S. (2011). Parallel machine scheduling with fuzzy processing times using a robust genetic algorithm and simulation. *Information Sciences*, 181(17):3551–3569.

Baz, M., Hunsaker, B., and Prokopyev, O. (2009). How much do we "pay" for using default parameters? *Computational Optimization and Applications*, 48(1):91–108.

Benders, J. (1962). Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4(1):238–252.

Berktaş, N., Kara, B. Y. Y., Karaşan, O. E., Berktas, N., Kara, B. Y. Y., and Karasan, O. E. (2016). Solution methodologies for debris removal in disaster response. *EURO Journal on Computational Optimization*, 4(3-4):403–445.

Bertrand, J. W. M. and Fransoo, J. C. (2002). Operations management research methodologies using quantitative modeling. *International Journal of Operations & Production Management*, 22(2):241–264.

Bertsimas, D. and Sim, M. (2003). Robust discrete optimization and network flows. *Mathematical Programming*, 98(1-3):49–71.

Bertsimas, D. and Sim, M. (2004). The Price of Robustness. *Operations Research*, 52(1):35–53.

Booth, W. (2010). Washington post. http://www.washingtonpost.com/wp-dyn/content/article/2010/03/06/AR2010030602544.html. Accessed on 20/08/2019.

Bui, Q. T., Deville, Y., and Pham, Q. D. (2016). Exact methods for solving the elementary shortest and longest path problems. *Annals of Operations Research*, 244(2):313–348.

Burke, E., Kendall, G., Newall, J., Hart, E., Ross, P., and Schulenburg, S. (2003). *Hyper-Heuristics: An Emerging Direction in Modern Search Technology*, pages 457–474. Springer US, Boston, MA.

Çelik, M. (2016). Network restoration and recovery in humanitarian operations: Framework, literature review, and research directions. *Surveys in Operations Research and Management Science*, 21(2):47–61.

Çelik, M., Ergun, Ö., and Keskinocak, P. (2015). The Post-Disaster Debris Clearance Problem Under Incomplete Information. *Operations Research*, 63(1):65–85.

Chang, M. S. and Li, D. C. (2010). A sampling-Based approximation method applied to stochastic real-time emergency rehabilitation scheduling problem. *International Journal of Intelligent Transportation Systems Research*, 8(1):42–55.

Conover, W. (1999). Practical nonparametric statistics, john wiley & sons. *INC, New York*.

Costa, A. M. (2005). A survey on benders decomposition applied to fixed-charge network design problems. *Computers & Operations Research*, 32(6):1429–1450.

Dantzig, G., Fulkerson, R., and Johnson, S. (1954). Solution of a Large-Scale Traveling-Salesman Problem. *Journal of the Operational Research Society of America*, 2(4):393–410.

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271.

Dolan, E. D. and Moré, J. J. (2002). Benchmarking optimization software with performance profiles. *Mathematical Programming*, 91(2):201–213.

Dowsland, K. A., Soubeiga, E., and Burke, E. (2007). A simulated annealing based hyper-heuristic for determining shipper sizes for storage and transportation. *European Journal of Operational Research*, 179(3):759–774.

Drake, J. H., Kheiri, A., Özcan, E., and Burke, E. K. (2019). Recent Advances in Selection Hyper-heuristics. *European Journal of Operational Research.*

EM-DAT (2019). The international disaster database. http://www.emdat.be/advanced_search/index.html. Accessed on 01/06/2019.

Errico, F., Crainic, T. G., Malucelli, F., and Nonato, M. (2017). A Benders Decomposition Approach for the Symmetric TSP with Generalized Latency Arising in the Design of Semiflexible Transit Systems. *Transportation Science*, 51(2):706–722.

Feng, C.-m. and Wang, T.-c. (2003). Highway Emergency Rehabilitation Scheduling in Post-Earthquake 72 Hours. *Journal of the Eastern Asia Society for Transportation Studies*, 5:3276–3285.

Fischetti, M., Ljubic, I., and Sinnl, M. (2017). Redesigning Benders Decomposition for Large-Scale Facility Location. *Management Science*, 63(7):2146–2162.

Galindo, G. and Batta, R. (2013). Review of recent developments in OR/MS research in disaster operations management. *European Journal of Operational Research*, 230(2):201–211.

Galvão, R. D., Chiyoshi, F. Y., and Morabito, R. (2005). Towards unified formulations and extensions of two classical probabilistic location models. *Computers & Operations Research*, 32(1):15–33.

Gendron, B., Lucena, A., da Cunha, A. S., and Simonetti, L. (2014). Benders Decomposition, Branch-and-Cut, and Hybrid Algorithms for the Minimum Connected Dominating Set Problem. *INFORMS Journal on Computing*, 26(4):645–657.

Gendron, B., Scutellà, M. G., Garroppo, R. G., Nencioni, G., and Tavanti, L. (2016). A branch-and-Benders-cut method for nonlinear power design in green wireless local area networks. *European Journal of Operational Research*, 255(1):151–162.

Gogna, A. and Tayal, A. (2013). Metaheuristics: review and application. *Journal of Experimental & Theoretical Artificial Intelligence*, 25(4):503–526.

Goldschmidt, K. H. and Kumar, S. (2016). Humanitarian operations and crisis/disaster management: A retrospective review of the literature and framework for development. *International Journal of Disaster Risk Reduction*, 20(March):1–13.

Han, L. and Kendall, G. (2003). Guided Operators for a Hyper-Heuristic Genetic Algorithm. In *Lecture Notes in Computer Science*, volume 2903, pages 807–820.

Hjorring, C. and Holt, J. (1999). New optimality cuts for a single-vehicle stochastic routing problem. *Annals of Operations Research*, 86:569–584.

Hooker, J. and Ottosson, G. (2003). Logic-based Benders decomposition. *Mathematical Programming*, 96(1):33–60.

Hu, S., Han, C., Dong, Z. S., and Meng, L. (2019). A multi-stage stochastic programming model for relief distribution considering the state of road network. *Transportation Research Part B: Methodological*, 123:64–87.

Hutter, F., Hoos, H. H., Leyton-Brown, K., and Stützle, T. (2009). Paramils: An automatic algorithm configuration framework. *Journal of Artificial Intelligence Research*, 36(1):267–306.

Irnich, S., Toth, P., and Vigo, D. (2014). Chapter 1: The Family of Vehicle Routing Problems. In Toth, P. and Vigo, D., editors, *Vehicle Routing: Problems, Methods, and Applications*, pages 1–33. SIAM.

Karakatič, S. and Podgorelec, V. (2015). A survey of genetic algorithms for solving multi depot vehicle routing problem. *Applied Soft Computing Journal*, 27:519–532.

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

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

Klingman, D., Napier, A., and Stutz, J. (1974). Netgen: A program for generating large scale capacitated assignment, transportation, and minimum cost flow network problems. *Management Science*, 20(5):814–821.

Kumar, R., Kumar, N., and Karambir (2012). A Comparative Analysis of PMX , CX and OX Crossover operators for solving Travelling Salesman Problem. *International Journal of Latest Research in Science and Technology*, 1(2):98–101.

Laporte, G. and Louveaux, F. V. (1993). The integer L-shaped method for stochastic integer programs with complete recourse. *Operations Research Letters*, 13(3):133–142.

Laporte, G., Louveaux, F. V. F., Hamme, L. V., and van Hamme, L. (2014). An Integer L-Shaped Algorithm for the Capacitated Vehicle Routing Problem with Stochastic Demands. *Operations Research*, 50(3):415–423.

Larranaga, P., Kuijpers, C., Murga, R., Inza, I., and Dizdarevic, S. (1999). Genetic Algorithms for the Travelling Salesman Problem: A Review of Representations and Operators. *Artificial Intelligence Review*, 13(Holland 1975):129–170.

Li, H., Jian, X., Chang, X., and Lu, Y. (2018). The generalized rollon-rolloff vehicle routing problem and savings-based algorithm. *Transportation Research Part B: Methodological*, 113:1–23.

Liberatore, F., Ortuño, M. T., Tirado, G., Vitoriano, B., and Scaparra, M. P. (2014). A hierarchical compromise model for the joint optimization of recovery operations and distribution of emergency goods in Humanitarian Logistics. *Computers & Operations Research*, 42:3–13.

Lin, Z.-Z., Bean, J. C., and White, C. C. (2004). A Hybrid Genetic/Optimization Algorithm for Finite-Horizon, Partially Observed Markov Decision Processes. *INFORMS Journal on Computing*, 16(1):27–38.

Martins de Sá, E. M., De Camargo, R. S., and De Miranda, G. (2013). An improved Benders decomposition algorithm for the tree of hubs location problem. *European Journal of Operational Research*, 226(2):185–202.

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

Miller, C. E., Tucker, a. W., and Zemlin, R. a. (1960). Integer Programming Formulation of Traveling Salesman Problems. *Journal of the ACM*, 7(4):326–329.

Montero, E., Riff, M.-c., and Neveu, B. (2014). A beginner's guide to tuning methods. *Applied Soft Computing*, 17:39–51.

Morabito, R. and Pureza, V. (2010). Modelagem e simulação. *Metodologia de pesquisa em engenharia de produção e gestção de operações*, 3:165–194.

Moreno, A., Alem, D., and Ferreira, D. (2016). Heuristic approaches for the multiperiod location-transportation problem with reuse of vehicles in emergency logistics. *Computers & Operations Research*, 69:79–96.

Moreno, A., Alem, D., Ferreira, D., and Clark, A. (2018). An effective two-stage stochastic multi-trip location-transportation model with social concerns in relief supply chains. *European Journal of Operational Research*, 269(3):1050–1071.

Moreno, A., Ferreira, D., and Alem, D. (2017). A bi-objective model for the location of relief centers and distribution of commodities in disaster response operations. *DYNA*, 84:356 – 366.

Moreno, A., Munari, P., and Alem, D. (2019). A branch-and-Benders-cut algorithm for the Crew Scheduling and Routing Problem in road restoration. *European Journal of Operational Research*, 275(1):16–34.

Moreno, A., Munari, P., and Alem, D. (2020). Decomposition-based algorithms for the crew scheduling and routing problem in road restoration. *Computers & Operations Research*, 119(1):104935.

Munari, P., Moreno, A., De La Vega, J., Alem, D., Gondzio, J., and Morabito, R. (2019). The Robust Vehicle Routing Problem with Time Windows: Compact Formulation and Branch-Price-and-Cut Method. *Transportation Science*, 53(4):1043–1066.

Noyan, N. (2012). Risk-averse two-stage stochastic programming with an application to disaster management. *Computers & Operations Research*, 39(3):541–559.

Ohlmann, J. W. and Thomas, B. W. (2007). A Compressed-Annealing Heuristic for the Traveling Salesman Problem with Time Windows. *INFORMS Journal on Computing*, 19(1):80–90.

Öncan, T., Altinel, I. K., and Laporte, G. (2009). A comparative analysis of several asymmetric traveling salesman problem formulations. *Computers & Operations Research*, 36(3):637–654.

OSHA (2019). Flood preparedness and response. *Occupational Safety and Health Administration*. https://www.osha.gov/dts/weather/flood/response.html. Accessed on 01/08/2019.

Özdamar, L. and Ertem, M. A. (2015). Models, solutions and enabling technologies in humanitarian logistics. *European Journal of Operational Research*, 244(1):55–65.

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

Perez, K., Adulyasak, Y., and Jans, R. (2019). Logic-based Benders reformulations for integrated process configuration and production planning problems. Technical report, Les Cahiers du GERAD G–2019–73, GERAD, HEC Montréal, Canada.

Poojari, C. A. and Beasley, J. E. (2009). Improving benders decomposition using a genetic algorithm. *European Journal of Operational Research*, 199(1):89–97.

Pramudita, A. and Taniguchi, E. (2014). Model of debris collection operation after disasters and its application in urban area. *International Journal of Urban Sciences*, 18(2):218–243.

Pramudita, A., Taniguchi, E., and Qureshi, A. G. (2012). Undirected Capacitated Arc Routing Problems in Debris Collection Operation After Disaster. *Infrastructure Planning and Management*, 68(5):805–813.

Pramudita, A., Taniguchi, E., and Qureshi, A. G. (2014). Location and Routing Problems of Debris Collection Operation after Disasters with Realistic Case Study. *Procedia - Social and Behavioral Sciences*, 125:445–458.

Rahmaniani, R., Crainic, T. G., Gendreau, M., and Rei, W. (2017). The Benders decomposition algorithm: A literature review. *European Journal of Operational Research*, 259(3):801–817.

Raidl, G. R. (2015). Decomposition based hybrid metaheuristics. *European Journal of Operational Research*, 244(1):66–76.

Rawls, C. G. and Turnquist, M. a. (2010). Pre-positioning of emergency supplies for disaster response. *Transportation Research Part B: Methodological*, 44(4):521–534.

Rei, W., Cordeau, J.-F., Gendreau, M., and Soriano, P. (2009). Accelerating Benders Decomposition by Local Branching. *INFORMS Journal on Computing*, 21(2):333–345.

Rio de Janeiro (2011). Resolução Nº 09/2011 da Assembléia Legislativa. in Portuguese.

Ross, S., Pineau, J., Paquet, S., and Chaib-draa, B. (2008). Online planning algorithms for pomdps. *J. Artif. Int. Res.*, 32(1):663–704.

Sahin, H., Kara, B. Y., and Karasan, O. E. (2016). Debris removal during disaster response: A case for Turkey. *Socio-Economic Planning Sciences*, 53:49–59.

Salazar-González, J.-J. and Santos-Hernández, B. (2015). The split-demand one-commodity pickup-and-delivery travelling salesman problem. *Transportation Research Part B: Methodological*, 75:58–73.

Shao, S., Sherali, H. D., and Haouari, M. (2017). A Novel Model and Decomposition Approach for the Integrated Airline Fleet Assignment, Aircraft Routing, and Crew Pairing Problem. *Transportation Science*, 51(1):233–249.

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

Taşkin, Z. C. and Cevik, M. (2013). Combinatorial Benders cuts for decomposing IMRT fluence maps using rectangular apertures. *Computers & Operations Research*, 40(9):2178–2186.

Taillard, E. D. (1999). A heuristic column generation method for the heterogeneous fleet VRP. *RAIRO - Operations Research*, 33(1):1–14.

Tang, C.-H., Yan, S., and Chang, C.-W. (2009). Short-term work team scheduling models for effective road repair and management. *Transportation Planning and Technology*, 32(3):289–311.

Tran, T. T., Araujo, A., and Beck, J. C. (2016). Decomposition methods for the parallel machine scheduling problem with setups. *INFORMS Journal on Computing*, 28(1):83–95.

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

Tzeng, G.-H., Chen, Y.-W., and Lin, C.-Y. (2000). Fuzzy Multi-objective Reconstruction Plan for Post-earthquake Road-network by Genetic Algorithm. *Research and Practice in Multiple Criteria Decision Making*, 487:510–528.

Van Wassenhove, L., Martinez, A., and Stapleton, O. (2010). Insead humanitarian research group. https://studylib.net/doc/8398552/analysis-of-the-relief-supply-chain-in-the-first-week-after. Accessed on 20/08/2019.

Xu, B. and Song, Y. (2015). An Ant Colony-based Heuristic Algorithm for Joint Scheduling of Post-earthquake Road Repair and Relief Distribution. *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, 13(2):632.

Yan, S., Chu, J. C., and Shih, Y.-L. (2014). Optimal scheduling for highway emergency repairs under large-scale supply-demand perturbations. *IEEE Transactions on Intelligent Transportation Systems*, 15(6):2378–2393.

Yan, S. and Shih, Y.-L. (2007). A time-space network model for work team scheduling after a major disaster. *Journal of the Chinese Institute of Engineers*, 30(1):63–75.

Yan, S. and Shih, Y.-L. (2009). Optimal scheduling of emergency roadway repair and subsequent relief distribution. *Computers and Operations Research*, 36(6):2049–2065.

Yan, S. and Shih, Y.-L. (2012). An ant colony system-based hybrid algorithm for an emergency roadway repair time-space network flow problem. *Transportmetrica*, 8(5):361–386.