



HAL
open science

Two variants of the Ring Star Problem

Julien Khamphousone

► **To cite this version:**

Julien Khamphousone. Two variants of the Ring Star Problem. Other [cs.OH]. Université Paris sciences et lettres, 2023. English. NNT : 2023UPSLD021 . tel-04319443

HAL Id: tel-04319443

<https://theses.hal.science/tel-04319443v1>

Submitted on 2 Dec 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



THÈSE DE DOCTORAT
DE L'UNIVERSITÉ PSL

Préparée à l'Université Dauphine PSL

Two variants of the Ring Star Problem

Soutenue par

Julien KHAMPHOUSONE

Le 10 Novembre 2023

École doctorale n°543

Spécialité

Informatique

Composition du jury :

Safia KEDAD-SIDHOUM

Professeure des Universités, CNAM Paris

Rapportrice

Laurent HOUSSIN

Professeur Associé, ISAE-SUPAERO

Rapporteur

Dominique QUADRI

Professeure des Universités
Université Paris-Saclay

Présidente du Jury

André ROSSI

Professeur des Universités
Université Paris Dauphine - PSL

Directeur de thèse

Fabián CASTAÑO

Chercheur R&D, Frubana

Encadrant de thèse

Sonia TOUBALINE

Maîtresse de Conférences
Université Paris Dauphine - PSL

Encadrante de thèse

“Cette thèse en informatique qui a pour sujet, en anglais, “Branch-and-Benders-cut exact methods for two fault-tolerant Ring Star problem variants”, prolonge le chemin entrepris depuis ce premier ordinateur à écran cathodique que mon père m’a fait découvrir à quatre ans.”

Julien Khamphousone

Dedicated to Shagana Mary Progam

Acknowledgements

I acknowledge...

Scholarship

Professor **André Rossi**, Doctor **Sonia Toubaline** and Doctor **Fabián Castaño**, PhD supervisors. **Safia Kedad-Sidhoum**, **Laurent Houssin** and **Dominique Quadri** that accepted to be my PhD jury members. Every student I had the chance to be alongside in my school classes and in particular the The LAMSADE's PhD students. Every teacher I had the chance to learn from, in particular:

(Français) École du Parc, Choisy-Le-Roi

- Maîtresse Autin [CPA] (2002-2003)
- Maîtresse Belsœur [CE1]
- Maîtresses Dayraud et Ruffieux [CE2A]
- Maîtresse Belsœur [CM1]
- Maîtresse Prévost [CM2B] (2006-2007)

Collège Saint-André, Choisy-Le-Roi

- Mme Helluy [6e B, Mathématiques] (2007-2008)
- M. Chedid [5e D, Mathématiques]
- Mme Foucriat [4e B, Mathématiques]
- Mme Chemin [4e B, Sciences Physiques]
- Mme Nollent [3e A, Éducation Musicale]
- Mme Chemin [3e A, Sciences Physiques]

- M. Azeba [3e A, Mathematics] (2010-2011)

(French and English) Lycée Guillaume Apollinaire, Thiais (94)

- Mme Castano [2de générale, Physics and Chemistry] (2011-2012)
- Mme Grosol [2de générale, English]
- M. Filey [2de générale, Mathematics]
- Mme Wong [1re S1-SI, Mathematics]
- Mme Chauvin [1re S1-SI, Physics and Chemistry]
- Mme Bonneron [1re S1-SI, English] (2012-2013)

Lycée Lakanal, Sceaux (92)

- Mme Proix, [Tle S, Physics and Chemistry] (2013-2014)
- M. Leterrier, *Agrégation jury member* [Tle S, Mathematics]

Lycée Paul Valéry, Paris 12e

- Dr. Chloé Mullaert [Mathematics] (2014-2015, MPSI)
- Dr. Gael Benabou [Mathematics] (2015-2016, MP)

Top French Engineering School ENSIIE, Évry

- Dr. Dimitri Watel [Computer Science] (2016-2019)
- Dr. Alain Faye [Computer Science] (2016-2019)

Conservatoire National des Arts et Métiers, Paris 3e

- The *Master Parisien de Recherche Opérationnelle (MPRO)* Professorship [Operations Research] (2018 - 2019)

Université Paris Dauphine, Paris 16e

- Pr. Jérôme Lang, [Operations Research and Artificial Intelligence] (2023)
- The (French) Laboratoire d'Analyse et de Modélisation de Systèmes pour l'Aide à la Décision (LAMSADE) researchers

Family

Jeannie Lao, grandmother, Martin Lao, grandfather, Virginie Lao, mother, Pang Khamphousone, father, Manivanh Khamphousone, grandmother, Oudone Khamphousone, grandfather, Alicia Khamphousone, younger sister. My five uncles and two aunts: Éric Lao, Saly Ly, adopted, H el ene Panyanouvath, Jacques Khamphousone, Somchit Khamphousone, Padit Khamphousone, Mouang Khamphousone. My cousins and in particular Andy Panyanouvath and Pranavong Khamphousone.

Kiwi Khamphousone, a little like my young brother, my cat and my love for biology.

Healthcare

- Dr. Pascale Romain, general practitioner
- Dr. Guillaume Laperrousaz, Confidential
- Dr. Isaac Salem, Confidential
- Dr. Georges Kill, Confidential
- Dr. Olivier Confidential, virologist and CNRS researcher

Future

Shagana Progam and her family until the end of my life.

Friends

Pierre-Fran ois Villard (ENPC), qui a relev e un homme   terre pendant Dassault Syst emes. Dr. Ali Tlili (Dassault Syst emes), pour son humour si Ali !  glantine Boucher (MPSI-MP comrade), pour m'avoir transmis l'importance de l'amiti e. S everine Bonnech ere (MPRO comrade), Bastien Rolland (MPRO comrade), Victor Kani (MPRO comrade), Guillaume Crognier (MPRO comrade), impressionnants en recherche op eracionnelle, impressionnants humainement. Toby Ratantely (NA-Motorsports and Orange), pour son aide, sa gentillesse et sa bienveillance.

The NA-Motorsports community and the three FMWC titles, FMWC Individual (2021 and 2022) and FMWC Duo (2022).

Technology

Alan Mathison Turing, The Massachusetts Institute of Technology (MIT) for the Julia Mathematical Programming Language and the European Organization for Nuclear Research (CERN) for the World Wide Web. Maurice Diamantini (Chercheur R&D, ENSTA Paris).

Spirituality and open-mindedness

Julie Sergeant, former Haute École de Commerce de Paris student (HEC Paris).
Daniel HUYNH (français), HUYNH-Buntha (Cambodgien avec lettres françaises),
Christelle Ngor's adoptive father that open myself into the path of spirituality.

Contents

Introduction	9
1 State of the Art	13
1.1 Networks design	13
1.2 Ring Star Problem	14
1.3 Robustness	16
1.4 Survivability	17
1.5 Resilience	18
1.6 Benders decomposition method	19
1.6.1 Litterature review on Benders decomposition method	19
1.6.2 Illustration of the classical Benders decomposition	20
2 A Survivable variant of the Ring Star Problem	25
2.1 Chapter's abstract	25
2.2 Chapter's structure	26
2.3 Survivable Ring Star Problem definition	26
2.4 An ILP formulation of 1-S-RSP	27
2.4.1 Brute-force algorithms of polynomial time complexity for the cases of three and four hubs	28
2.4.2 An ILP formulation for at least five hubs	28
2.5 A Branch-and-Benders-cut decomposition of 1-S-RSP	32
2.5.1 Description of the master problem and the subproblem	32
2.5.2 Linear programming formulation of the subproblem	35
2.5.3 Computing the dual of LP_0 and LP_i^+ for all the nodes i in V	40
2.5.4 Computing optimal primal and dual solutions to the sub- problems	41
2.5.5 Polynomial-time algorithm to the subproblem's dual	43
2.5.6 Optimality cuts of the Benders decomposition	44
2.5.7 Polyhedral analysis of constraints (2.14)	44
2.5.8 An instance transformation for 1-S-RSP	48

2.6	Enhancing and solving the ILP and the Branch-and-Benders-cut of 1-S-RSP	50
2.6.1	Introduction of 2-opt backup	50
2.6.2	Description of the ILP and Branch-and-Benders-cut algorithms	51
2.7	Numerical Experiments	51
3	A Resilient variant of the Ring Star Problem	61
3.1	Chapter's abstract	61
3.2	Chapter's structure	61
3.3	Resilient Ring Star Problem definition	61
3.4	A MILP formulation for Resilient RSP	62
3.4.1	Post-optimization procedure	66
3.5	A Branch-and-Benders-cut decomposition of 1-R-RSP	68
3.5.1	Master problem	68
3.5.2	Benders subproblem	69
3.5.3	Benders optimality cut	75
3.6	Numerical Experiments	76
3.7	Resilient or Survivable Ring Star Problem?	78
3.7.1	Properties	78
3.7.2	Computational method to solve 1-R-RSP(F) on an interval	80
3.7.3	Solving 1-R-RSP(F) for all $F \geq F_\ell$	83
3.7.4	Comparing the cost of 1-S-RSP and 1-R-RSP(F)	84
3.7.5	Numerical experiments to choose between Survivable or Resilient RSP	84
4	Conclusions	91
4.1	Overview	91
4.2	Limitations	91
4.3	Future works	92
	List of Figures	95
	List of Tables	97
	Bibliography	99

Introduction

In the context of telecommunications, transportation, and logistics, among others, designing networks based on specific architectures is an important task as it describes how the components and entities should be linked together in order to construct a desired network that performs efficiently and achieves predefined goals. A typical application of such a network design arises in the field of telecommunication networks, in which user nodes lie on an access network, and are connected to concentrators. These concentrators are linked together to form a backbone network, which is connected to a central unit called the root [26]. More applications and references can be found in the survey [37]. There are many ways to design networks and several architectural models can be proposed. In this thesis, we consider a tributary and backbone architecture where we seek to select a set of nodes, called hubs, and interconnect them to define the backbone network. The remaining nodes, called terminals are connected to hubs and form the tributary network. The objective is to determine an efficient and cost-effective tributary and backbone network.

We consider in particular the Ring Star problem where the ring is the backbone network and the star the tributary one. This problem has been studied many times in the literature, and has applications in telecommunications, logistics, urbanism, and more. The famous Traveling Salesman Problem is a particular case of the Ring Star Problem where every node belongs to the ring. Hence, the Ring Star Problem is also NP-Hard. Resilience and survivability are two concepts related to robustness and have been receiving a lot of attention in recent years.

Resilience and survivability are responses to uncertainty that occur when unexpected events disturb the system. There have been applications for resilience and survivability not only in combinatorial optimization but in many fields in sciences, from biology to economy. Thus, we are motivated to introduce during my thesis two variants of the Ring Star Problem, a Survivable one and a Resilient one. Both variants consider the case where at most one hub fails. In the Survivable

variant, the ring star network will be preserved whatever hub fails while in the Resilient variant, two corrective operations occurs when a hub fails to preserve the ring star network.

Chapter 1 contains the state of the art in fields of Networks design, the Ring Star Problem, Robustness, Survivability, Resilience, and presents the Benders Decomposition method used to solve our two variants problems. Chapter 2 introduces the variant "a Survivable variant of the Ring Star Problem named *1-S-RSP*". It studies the case where at most one node in a given subset of so-called uncertain nodes can fail if selected as a hub. The variant's objective is to minimize the ring star network design plus the costs associated with coping with the failure of any uncertain hub. This survivable variant of RSP is modeled as an Integer Linear Program (ILP) and also a Branch-and-Benders-cut decomposition, the sub-problem of this decomposition being solved by a polynomial algorithm. Several other improvements are made to both solving approaches, in particular taking advantage of an advanced two-opt heuristic adapted for the survivable case and a transformation of the input instance. Mathematical proofs and extended numerical experiments are carried out.

Chapter 3 is about a second variant of the Ring Star Problem named "Resilient RSP" that is solved using both an ILP and a Branch-and-Benders-cut algorithm. The Resilient RSP is a problem where when a node fails in a given subset of so-called uncertain nodes two corrective operations occur to repair the Ring Star structure, first, a backup edge is paid to reconnect the ring and all non-hubs that were connected to the failing hub must be reallocated to the nearest remaining hub. Extensive numerical experiments are also carried out. At the end of this chapter, a tradeoff is studied, we address the question of deciding whether the survivable or the resilient approach is the most appropriate one, as a function of the failure duration.

The conclusion in Chapter 4 discusses some limitations of our work and some interesting perspectives. This thesis manuscript ends with a Bibliography section.

**Branch-and-Benders-cut exact
methods for two fault-tolerant
Ring Star problem variants**

Chapter 1

State of the Art

This State of the Art is organized in topics, first we review networks design, afterwards the Ring Star Problem which is a network design problem, then we focus on three concepts of uncertainty, robustness, survivability and resilience and finally we cover the Benders decomposition.

1.1 Networks design

Networks design is a wide field with many problems that have been studied in the literature. Networks design have applications in telecommunications as it is critical to decide where to locate for instance the antennas or other key structures. In (1993), Kershenbaum [34] wrote a book on telecommunications networks design algorithms. Network design also have applications in transportation networks where one should decide where to build stations, roads, bridges, etc. for buses, trucks, airports for planes, ports for boats and so on. Farahani [21] *et al.* (2013) published a review of urban transportation network design problems. It also has applications in supply chain. For instance, in a warehouse implementing a pallet rack storage system, network design may be of great help.

A well known and interesting network design problem is the facility location problem that has been extensively studied in operations research and network design. Shmoys *et al.* [52] (1997) present new approximation algorithms in facility locations problems. In every facility location problem that they study, there is a set of locations where facilities may be built with a fixed cost and a set of clients that are assigned to a facility location with another allocated cost. The objective is to determine a set of locations where facilities are opened and minimize the total facility and assignment costs. In the uncapacitated case, the facility can serve as many clients as necessary and in the capacitated case, each facility can serve at

most Q clients where Q is an input parameter. Farahani *et al.* [22] (2010) present a survey on multiple criteria facility location problems while Ortiz-Astorquiza *et al.* [46] (2018) conduct a review on multi-level facility location problems.

Other famous and attracting problems in networks design are Hub location problems (HLP). HLP are usually harder to solve than facility location problems. These problems consist of locating hub facilities, designing the network, often deciding the origins and destinations, and choosing the routing of flows through the network, while optimizing an objective function. After a state of the art of hub location problem in Alumur *et al.* [1] (2008), Alumur *et al.* [2] (2021) provide insights for better modeling HLPs to help create a roadmap for future HLP research. Several variants of HLP exist, Martins *et al.* [51] (2015) introduce the Hub Line Location Problem (HLLP) where the hubs are connected by a line and non-hubs are connected to hubs. They present an exact Benders decomposition algorithm for this problem. A topology that also has a backbone and tributary architecture is the hub-and-spoke system, introduced by O’Kelly [45] (1987). Many works on hub-and-spoke problems were published in operations research and location or transportation journals. In a hub-and-spoke structure, every pair of hubs is connected, while non-hubs are connected to the closest hub. Bryan *et al.* [12] (1999) survey advances in analysis of the hub location problem and its variants including the hub-and-spoke problem.

Another problem that is in the scope of networks design is the Median Cycle Problem (MCP) introduced by Labbé *et al.* [39] (2005). In this problem, the aim is to determine a single cycle through a subset of nodes involving the sum of the two next costs: a routing cost associated with the aforementioned cycle and the cost of assigning nodes not on the cycle to visited nodes. The objective is to minimize the cycle’s cost and the assignment cost subject to an upper bound on the total assignment cost. They present a MILP and enhance it with a class of valid inequalities. A branch-and-cut algorithm is also described alongside separation procedures.

In the next section, we describe and give a literature review of the Ring Star Problem which is the primary topology of interest in this thesis and is a backbone and tributary structure.

1.2 Ring Star Problem

We consider a specific tributary and backbone architecture, a subfield of networks design, called the ring-star network. The problem input is a complete mixed

weighted graph, with arcs from and to every node as well as edges between any two different nodes and having a specific node called the depot. The RING STAR PROBLEM (RSP) consists in selecting a subset of *hubs* including the depot and linking them with edges to form a cycle called the *ring*. Each remaining node is called a terminal and must be connected to exactly one hub using an arc, which is the star topology part. The objective of RSP is to minimize the sum of three costs corresponding to (i) selecting the hubs, (ii) selecting the edges to form the ring, and (iii) selecting the arcs to connect each terminal to a hub. RSP is known to be NP-hard since the Traveling Salesman Problem is a special case when the assignment costs are large compared to the ring costs. Figure 1.1 shows an example of such a network.

RSP has been widely studied in the literature. Xu *et al.* [56] (1999) considered a digital data service problem in the telecommunication industry where hubs have to be selected among a given subset and link end offices to hubs at minimum cost. They model this problem as RSP, propose an Integer Linear Problem (ILP) formulation, and address it with a tabu search method. Labbé *et al.* [40] (2004) proposed the first Mixed Integer Programming model, strengthened with valid inequalities resulting of a polyhedral analysis and solved it with a branch-and-cut algorithm. Kedad-Sidhoum and Nguyen [32] (2010) later proposed a strengthened approach where the depot is duplicated and that changes the ring to an st-chain that starts from the depot and ends at a special node that is a duplicate of the depot. Their formulation allows to derive facet-defining inequalities and their branch-and-cut algorithm is shown to be more efficient than the cycle based branch-and-cut from Labbé. Simonetti *et al.* [53] (2011) also proposed a new integer programming formulation. They implemented a branch-and-cut algorithm for RSP based on a reduction of the latter to a minimum Steiner tree problem defined over a layered graph, with the introduction of side constraints.

Several heuristics and approximation algorithms have been proposed to address RSP. Dias *et al.* [17] (2006) use a hybrid method composed of a General Variable Neighborhood Search (GVNS) alongside a Greedy Randomized Adaptive Search Procedure (GRASP) to approach heuristically the Ring Star Problem while Calvete *et al.* [13] (2013) model the problem using a bi-level optimization formulation and propose an evolutionary algorithm to solve it. Recently, Zang *et al.* [57] (2020) proposed an ant colony system heuristic for RSP. Chen *et al.* [14] (2021) gave three approximation algorithms for the RSP with fixed edge-cost ratio. In the variant they studied, the edge and arcs costs are proportional and their ratio is an additional problem input. Baldacci *et al.* [4] (2007) studied a variant of

the Ring Star Problem, the Capacitated m -Ring-Star Problem which consists of selecting a set of m rings that pass through the depot and non-hubs (customers) and a potential subset of hubs nodes (Steiner nodes), while customers not belonging to any ring must be assigned to a visited (customer or Steiner) node. Moreover, the rings must be node-disjoint and the number of customers assigned to the rings is bounded by an input capacity Q . This problem is also a generalization of the Traveling Salesman problem, hence is also NP-Hard. Najiami *et al.* [44] (2010) gave a heuristic procedure variant of RSP. They propose two Integer Programming formulations for this problem and strengthen it with valid inequalities.

Some hubs may fail, hence building a robust Ring Star network that keeps the structure of the architecture even if one hub fails is an interesting problem to study. To the best of our knowledge, survivable and resilient variants of RSP have never been studied in the literature. We will consider these two variants in this thesis. Therefore we will review the concepts of robustness, survivability, and resilience in a network design context.

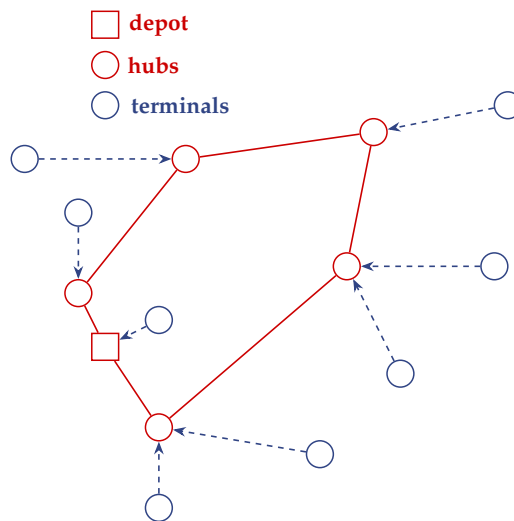


Figure 1.1: Ring Star network

1.3 Robustness

Robustness in optimization aims to deal with data uncertainty. When data is uncertain, the question of designing a solution that is relevant towards such uncertainty is natural. Such design is called "Robust". The case study by Ben-Tal and Nemirovski [7] (2000) on 90 linear programs from the NETLIB collection shows that a solution to a linear program can be completely meaningless with data contaminated with small uncertainty. Two groups of authors independently stud-

ied robust optimization, El Ghaoui *et al.* [18, 19] (1997, 1998) and Ben-Tal and Nemirovski [6, 8, 7] (1998, 1999, 2000). Bertsimas and Sim [10] (2004) introduce a smart way to control what they call "the price of robustness": traditional approaches of robustness are focused on the worst-case scenario, which tends to make robust solutions too conservative (or overly pessimistic). Hence, they propose to use a new input parameter, often denoted by Γ which is an upper bound on the number of parameters that can be simultaneously affected by uncertainty. This gives the decision maker more control over how conservative the robust solution should be. The notion of robustness has many definitions and is frequently revisited and extended, see for instance Kouvelis *et al.* [38] (1996) and Ben-Tal *et al.* [5] (2009) which are two books with different visions on robust optimization. Kouvelis *et al.* [38] (1996) deal with decision-making with data uncertainty and a strong importance on operations and production management applications. Roy [49] (2010) studied robustness in operational research and decision aiding and in particular describes "robustness concerns" as an *a priori* concerns that must be taken into account at the moment the problem is formulated, as opposed to "robustness analysis" or "sensitivity analysis" which are conducted after the problem is solved. More recently, in the context of attack scenarios, robust optimization has also been studied considering multiobjective decision-making. Wang *et al.* [54] (2021) consider a multiobjective network robustness optimization problem which they solve with a computationally efficient evolutionary algorithm. In all these works described above uncertainty is related to data input. Some other types of uncertainties exist such as those concerned with maintaining network structures after some disturbances when designing it. The conservation of properties when unexpected failures affect the network can be achieved in two different ways. If the solution can stand the considered class of failures without any corrective action, we talk about survivability, whereas resilience is concerned with the pre-planning at the design time of corrective actions, which may incur additional costs when they are applied.

1.4 Survivability

In the presence of hazardous events (attacks, failures, and so on), designing a reliable topological network is essential. Survivable network design problems have widely been studied in the literature [27, 33] (1995, 2005). These problems aim at building *survivable* networks, *i.e.*, networks that remain operational despite the occurrence of failures or attacks. In the field of survivable network design problems, Labbé *et al.* [41] (2005) consider a fully interconnected star problem where the hubs are linked by a clique. The authors perform a polyhedral analysis of

the proposed model and solve it with a branch-and-cut algorithm. Fouilhoux *et al.* [24] (2012) study a 2-edge connected star problem where the backbone is a 2-edge connected subgraph. For all $k \in \mathbb{N}$, a graph is k -edge connected, if there are at least k edge-disjoint paths between any two different nodes. In [31] (2014), Karaşan *et al.* consider 2-edge connected star problems where each terminal is connected to two hubs. Both papers provide integer programming formulations and valid inequalities for those problems, analyze facet-defining inequalities and present both heuristic and exact separation algorithms. In such works, the survivability is considered either for both tributary and backbone networks Karsan *et al.* [31] (2014) or only for the backbone network [41, 24] (2005, 2012). In both cases, the topological structure may not be preserved if a hub fails. Indeed, as shown in Figure 1.2, the backbone structure is not connected anymore when the shaded hub fails.

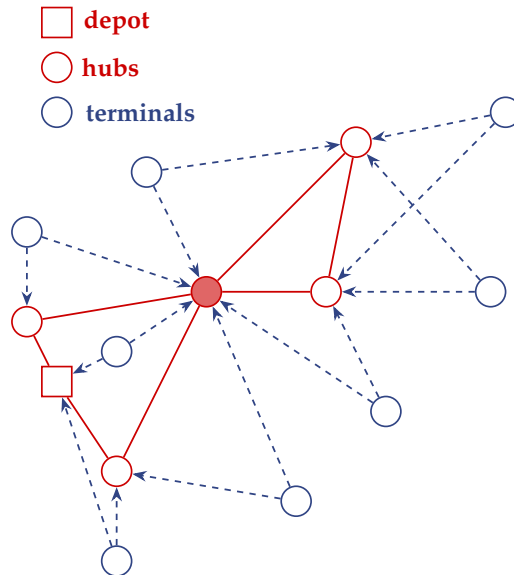


Figure 1.2: 2-edge connected star, dual homed network [31]

1.5 Resilience

Resilience is often described in the literature as when a system or a network gets damaged, how long or how much resource or energy will it need to recover its full functionality. Holling [29] (1973) first introduced the term of resilience in the context of ecology. Since then, in the field of computer science, Liao *et al.* [43] (2018) have proposed a resilience optimization model for transportation networks under natural or man-made disasters. They here describe "resilience" as the faculty to recover from damage under unexpected conditions and seek to improve

the resilience of transportation infrastructures.. Zang *et al.* [58] (2017) study a stochastic programming approach for resilient hub location in a power projection network considering random hub failures. The failures of the hubs, which are critical facilities, due to uncertainty threats like natural disasters or man-made attacks could lead to excessive loss or cost. They have the same definition of resilience as in Liao *et al.* [43] (2018), *i.e.*, resilience is the ability of a network to withstand and reduce the impact of damages. In Gu *et al.* [28] (2020), the performance of a transportation network is studied under three types of perturbations: reliability, vulnerability and resilience. In An *et al.* [3] (2015), they study the failures of hubs in a hub-and-spoke network by addressing resilient hub-and-spoke networks where backup hubs and alternative routes are taken into consideration to handle hub failures. They model nonlinear mixed integer formulations for such networks problems.

1.6 Benders decomposition method

1.6.1 Literature review on Benders decomposition method

In this thesis, we resort to the Benders decomposition approach which is implemented in a more efficient way than in the original approach. The combination of Benders decomposition and a cutting plane algorithm is referred to as the Branch-and-Benders-cut method Cotado [15] (2006). The Benders decomposition method has been introduced by Jacques Benders [9] (1962), it is an exact method and many works have been carried out using this approach since then. It consists of decomposing the studied problem into two problems called master problem and subproblem. The master problem includes the complicating variables and the subproblem is easy to solve once the complicating variables are fixed. At each iteration, the master problem, written as a mixed integer linear program, is solved to optimality, then the subproblem is also solved, using the decision variables of the master problem as input parameters. After solving the subproblem, a feasibility cut or an optimality cut is added to the master problem or the current solution of the master problem is proven optimal. Many papers are related to this decomposition. Geoffrion *et al.* [25] (1972) generalized Benders' approach to a broader class of programs in which the parametrized subproblem can be non linear and convex instead of being linear. Bloom [11] (1983) used the generalized Benders decomposition from Geoffrion to solve an electricity generating capacity expansion planning problem. In Rahmaniani *et al.* [47] (2017) a comprehensive survey of the Benders decomposition algorithm is provided. Dieter *et al.* [55] (2019) present two algorithms to clarify some of the Benders decomposition algorithms presented be-

tween 2000 and 2010. In general Benders decompositions, we consider problems of the form $\min\{cx + dy : Gx + Hy \geq b, x \in \mathbb{N}^{m_1}, y \in \mathbb{R}_+^{m_2}\}$. The problem is split into a master problem on the x variables and a subproblem on the y variables which are generally of linear form. Fakhri *et al.* [20] (2017) consider the case where the subproblem variables are integer and present an exact method for this particular case. Keyvanshokoo *et al.* [35] (2016) have used an accelerated Benders decomposition to solve a hybrid robust and stochastic optimization for closed-loop supply chain network design. Recently, Juvin *et al.* [30] (2023) studied the flexible job-shop scheduling problem, a generalization of the job-shop scheduling problem, and considered preemptible tasks for this generalization. They use a logic-based Benders decomposition to solve this problem.

1.6.2 Illustration of the classical Benders decomposition

We consider the mixed integer linear program below, referred to as the *original problem*, that will be addressed using the Benders decomposition.

$$\begin{aligned} \text{Minimize} \quad & cx + dy \\ & Gx + Hy \geq b \\ & x \in \mathbb{N}^{m_1} \\ & y \in \mathbb{R}_+^{m_2} \end{aligned}$$

The input data are as follows: c is a vector of \mathbb{R}^{m_1} , d is a vector of \mathbb{R}^{m_2} , b is a vector of \mathbb{R}^n , G is an n -by- m_1 real matrix and H is an n -by- m_2 real matrix.

We also make the following two assumptions: first, we suppose there exist n_1 and n_2 such that $n_1 \geq 1$ and $n = n_1 + n_2$, and $H(i, j) = 0$ for all $(i, j) \in \{1, \dots, n_1\} \times \{1, \dots, m_2\}$. Hence G, H and the vector b can be partitioned into two blocks having n_1 and n_2 rows:

$$G = \begin{array}{c} \xrightarrow{m_1} \\ \boxed{\begin{array}{c} G_1 \\ G_2 \end{array}} \begin{array}{l} \updownarrow n_1 \\ \updownarrow n_2 \end{array} \end{array} \quad H = \begin{array}{c} \xrightarrow{m_2} \\ \boxed{\begin{array}{c} H_1 \\ H_2 \end{array}} \begin{array}{l} \updownarrow n_1 \\ \updownarrow n_2 \end{array} \end{array} \quad b = \begin{array}{c} \xrightarrow{1} \\ \boxed{\begin{array}{c} b_1 \\ b_2 \end{array}} \begin{array}{l} \updownarrow n_1 \\ \updownarrow n_2 \end{array} \end{array}$$

Where H_1 is the zero n_1 -by- m_2 matrix.

Now, we apply the Benders decomposition of the original problem by creating a *master problem* and a *subproblem*, based on a partition of the variables: x are the *complicating* variables, because once they are fixed, solving for the remaining y variables is an easy problem. In this section, the partition is given, but in general, the user has to determine this partition, that may not be unique.

The master problem is denoted by MP, and is built by replacing the contribution of the y variables in the objective function of the original problem, *i.e.*, dy , by a new nonnegative variable denoted by η . The constraints of the master problem are the first m_1 ones, *i.e.*, those of the original problem for which the y -variables are absent:

$$\text{MP} : \begin{cases} \text{Minimize} & cx + \eta \\ & G_1x \geq b_1 \\ & x \in \mathbb{N}^{m_1} \\ & \eta \geq 0 \end{cases}$$

The subproblem is denoted by $\text{SP}(\hat{x})$ because it is solved only once MP has been solved to optimality. The optimal solution of MP is denoted by the pair $(\hat{x}, \hat{\eta})$, where \hat{x} is a constant vector of \mathbb{N}^{m_1} , and $\hat{\eta}$ is a constant nonnegative real number. The subproblem is to minimize the contribution of the y -variables in the objective function of the original problem, while satisfying the constraints of the original problem for the (fixed) solution vector \hat{x} returned by the master problem:

$$\text{SP}(\hat{x}) : \begin{cases} \text{Minimize} & dy \\ & H_2y \geq b_2 - G_2\hat{x} \\ & y \in \mathbb{R}_+^{m_2} \end{cases}$$

It can be observed that $\text{SP}(\hat{x})$ is a linear program, so it can be solved in polynomial time. We then solve $\text{SP}(\hat{x})$ to optimality, one of the following four situations occurs.

1. $\text{SP}(\hat{x})$ turns out to be unbounded. This means that the original problem is also unbounded, and the solution process stops.
2. $\text{SP}(\hat{x})$ turns out to be infeasible. This means that \hat{x} , the optimal solution found for the subproblem does not allow to set the y -variables, *i.e.*, (\hat{x}, \hat{y}) is not a feasible solution to the original problem. As a consequence, a Benders *feasibility cut* must be added to the master problem before iterating the process.
3. $\text{SP}(\hat{x})$ has an optimal solution \hat{y} , and $\hat{\eta} < d\hat{y}$. Then, (\hat{x}, \hat{y}) is a feasible solution to the original problem, but is not optimal, as the contribution of the subproblem in MP (namely η) is not set appropriately. This indicates that a Benders *optimality cut* must be added to the master problem before iterating the process.
4. $\text{SP}(\hat{x})$ has an optimal solution \hat{y} , and $\hat{\eta} = d\hat{y}$. Then, (\hat{x}, \hat{y}) is an optimal solution to the original problem, and the solution process stops.

When situations 2 or 3 occur, we need to separate a feasibility cut, or an optimality cut. To do this, we use the fact that $SP(\hat{x})$ is a linear program, and consider its dual denoted by $DSP(\hat{x})$, whose objective function is to maximize $\alpha(b - G_2\hat{x})$, where α is an n_2 -element vector of decision variables. We now reexamine situations 2 and 3 in more details:

2. $SP(\hat{x})$ is infeasible implies that $DSP(\hat{x})$ is unbounded. By the Minkowski-Weyl theorem, the feasible set of $DSP(\hat{x})$ can be expressed as a convex combination of extreme points, plus a nonnegative linear combination of extreme rays. Since $DSP(\hat{x})$ is a maximization problem, there exists an extreme ray in the feasible set representation of $DSP(\hat{x})$ that is responsible for its unboundedness. This extreme ray, denoted by α^r , should be banned, which leads to an inequality of the form $\alpha^r(b - G_2\hat{x}) \leq 0$. This feasibility cut actually cuts off the current solution \hat{x} of MP, so it should be added to MP.
3. $SP(\hat{x})$ has an optimal solution \hat{y} that satisfies $\hat{\eta} < d\hat{y}$. Again, by the Minkowski-Weyl theorem, the optimal solution to $SP(\hat{x})$ corresponds to an extreme point α^p of the feasible set of $DSP(\hat{x})$. Because of this solution, $\hat{\eta}$ cannot be less than the optimal objective value of $DSP(\hat{x})$, hence $\alpha^p(b - G_2\hat{x}) \leq \hat{\eta}$ is an optimality cut to be added to MP.

Figure 1.3 shows the framework for the classical Benders decomposition method with a flowchart. In this figure, $SP(\hat{x})$ is the optimal objective value of the subproblem. In the classical Benders decomposition approach, one Master Problem (a Mixed Integer Linear Program) is solved to optimality at each iteration. At each iteration, a new branch-and-bound tree is built from scratch and a lot of time is likely spent revisiting candidate solutions that have been eliminated earlier in the search. We can instead build a single search tree and generate valid cuts for the integer (and fractional) solutions encountered inside the tree, attaining the same optimal solution, this is what we call the Branch-and-Benders-cut scheme that will be used during this thesis.

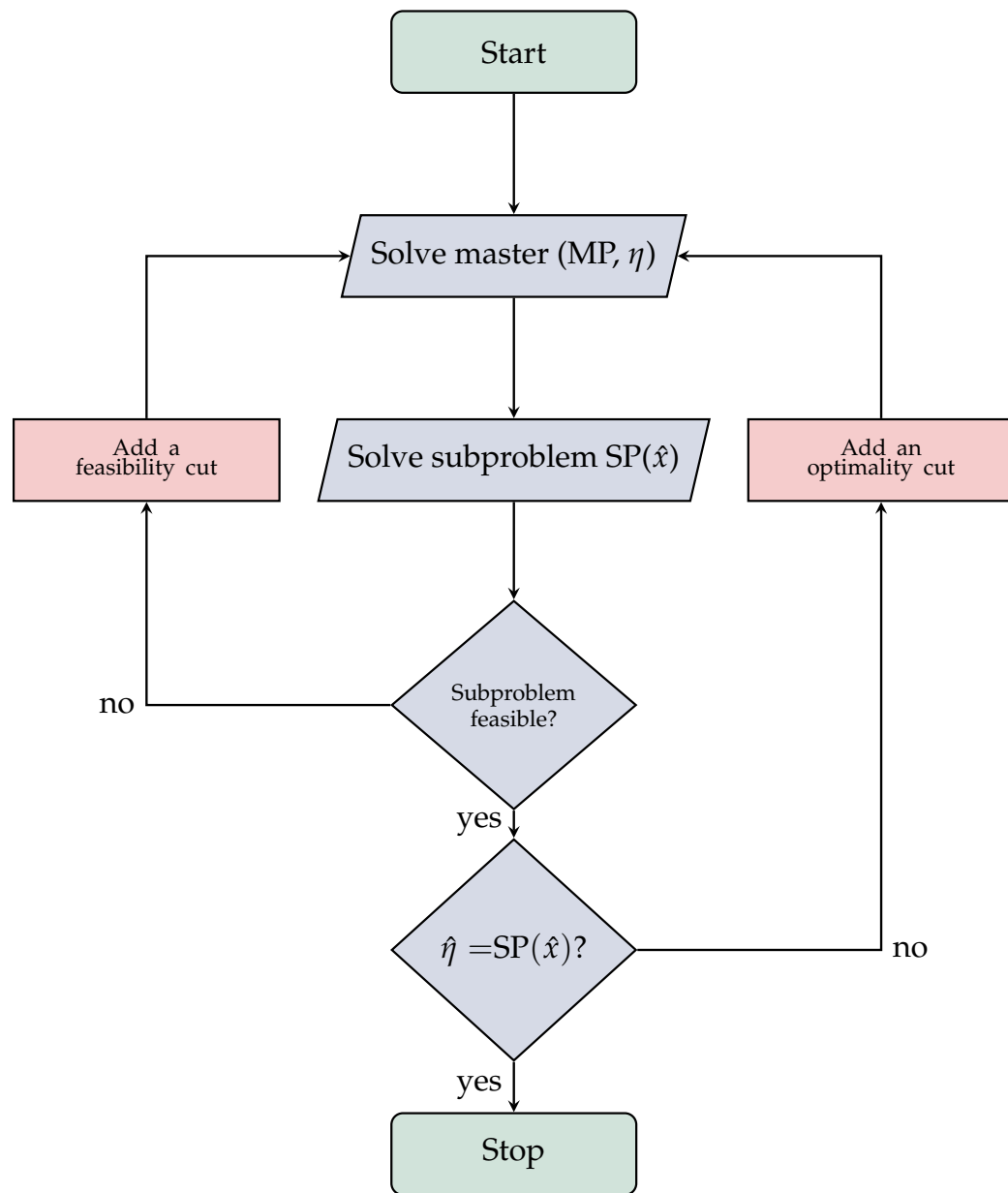


Figure 1.3: Classical Benders decomposition scheme

Chapter 2

A Survivable variant of the Ring Star Problem

2.1 Chapter's abstract

The Ring Star Problem consists in selecting a subset of nodes called *hubs* including the depot and linking them with a cycle, the remaining nodes being connected to exactly one hub, at minimum cost. We study a survivable variant of the Ring Star Problem where at most one node in a given subset of so-called uncertain nodes can fail if selected as a hub. We model this problem as an Integer Linear Program (ILP), that is also addressed with a Branch-and-Benders-cut decomposition. The Benders subproblem is turned into a linear program with the addition of new inequalities that are shown to be facet-defining, and several enhancements to both the ILP and Branch-and-Benders-cut algorithm are also presented. Both approaches are compared on the basis of extensive numerical experiments that bring the following conclusions. First, the survivable variant is shown to be much harder than the original Ring Star Problem, and the extra cost induced by survivability is significant. Second, the ILP formulation tends to produce tighter lower bounds but memory issues are frequent for large instances. Finally, the Branch-and-Benders-cut algorithm returns feasible solutions that are often of better quality than those produced by ILP, and is less frequently subjected to memory issues on the considered set of instances. This chapter's content has been submitted to Networks journal and part of it has been published in INOC [36] (2022). It has also been presented in ROADEF 2022.

2.2 Chapter's structure

This chapter is structured as follows: in Section 2.3 we introduce the SURVIVABLE RING STAR PROBLEM and present an ILP formulation in Section 2.4. We describe a Branch-and-Benders-cut decomposition in Section 2.5, and propose enhancements in Section 2.6. Finally, numerical experiments are reported in Section 2.7 where both approaches to address the SURVIVABLE RING STAR PROBLEM are compared, and the results are discussed.

2.3 Survivable Ring Star Problem definition

We define in Table 2.1 some of the notations used in this work. In this section, we first recall the Ring Star problem and then introduce the Survivable Ring Star Problem.

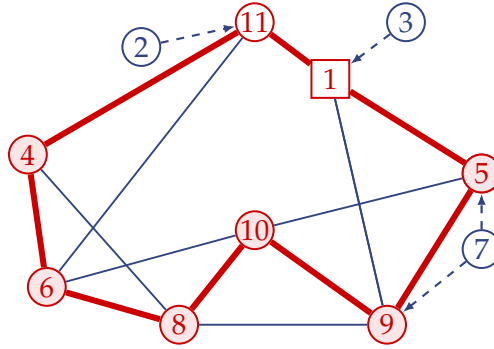
The input of RSP is the complete mixed graph $G = (V, E \cup A)$ with $|V| = n \in \mathbb{N}$ nodes, $n \geq 3$ where node 1 is a specific node called the depot, $E = \{ij \mid (i, j) \in V^2, i < j\}$ is the edge set and A is the arc set. Building the ring consists of selecting a subset $H \subseteq V$, which cardinality is not known in advance and $1 \leq |H| \leq n$, then linking the selected hubs of H by a cycle using $|H|$ edges of E . The depot node has to be in H . The cost of opening a hub $i \in V$ is $o_i \in \mathbb{R}_+$ and the cost of selecting edge $ij \in E$ between hubs i and j is $r_{ij} \in \mathbb{R}_+$. The star requirement of the problem asks that each terminal in $T = V \setminus H$ be connected to exactly one hub in H . The cost of selecting arc $(i, j) \in A$ to connect terminal $i \in T$ to hub $j \in H$ is $s_{ij} \in \mathbb{R}_+$. Finally, the RSP is to design a minimum-cost Ring Star network, where the cost is composed of the sum of selecting the hubs, linking them to form the ring and connecting the terminals.

The SURVIVABLE RING STAR PROBLEM, referred to as 1-S-RSP where 1 is the maximum number of simultaneous hub failures, has one more input than RSP: $\tilde{V} \subseteq V$ is a subset of nodes that can fail if they are selected as hubs. Following the terminology of Rossi *et al.* [48], the hubs in \tilde{V} are called *uncertain*, whereas the hubs of $V \setminus \tilde{V}$ are called *certain* as they are not supposed to fail. The aim of problem 1-S-RSP is to build a minimum-cost subgraph of G such that if at most one uncertain hub fails, the ring can be restored using a backup edge, and each terminal is still connected to at least one surviving hub. To achieve this, for any uncertain hub, an additional edge has to join its two neighbors in the ring. This edge is called a *backup edge*, and is used whenever the hub in question fails. Each terminal is connected either to a single certain hub or to two uncertain hubs. Thus, when an uncertain hub fails, the terminals connected to it are still connected to the ring. Note that when \tilde{V} is empty, 1-S-RSP reduces to RSP. Figure 2.1 shows an

illustration of a solution to the 1-S-RSP with $\tilde{V} = V \setminus \{1, 11\}$. In this solution, the terminals are $T = \{2, 3, 7\}$ and the hubs are $H = V \setminus T$. Backup edges are shown in blue, and it can be observed that the number of backup edges is equal to the number of uncertain hubs. This property can be shown to hold when the ring is made of 5 or more hubs.

Table 2.1: Table of Notations

V	Set of nodes, $V = \{1, 2, \dots, n\}$
\tilde{V}	Set of uncertain nodes, $\tilde{V} \subseteq V$
V^\neq	Set of couples of different nodes $V^\neq = \{(i, j) \in V^2 : i \neq j\}$
E	Set of edges, $E = \{ij \mid (i, j) \in V^2, i < j\}$
A	Set of arcs, $A = \{(i, j) \mid (i, j) \in V^\neq\}$
\tilde{J}	Set of triplets where the second element is an uncertain node $\tilde{J} = \{(i, j, k) \in V^3 : j \in \tilde{V}, i \neq j, j \neq k, i < k\}$
s	n -by- n nonnegative matrix of star costs
r	n -by- n nonnegative symmetric matrix of ring costs
o	n -element nonnegative array of opening costs

Figure 2.1: An 11-node solution of the 1-S-RSP with $\tilde{V} = V \setminus \{1, 11\}$

2.4 An ILP formulation of 1-S-RSP

In this section, we present an ILP formulation for 1-S-RSP. This ILP exploits the fact that the number of backup edges is equal to the number of uncertain hubs. However, this property does not hold when the ring contains 3 or 4 hubs, so these two particular cases are dealt with by two brute-force algorithms. Problem 1-S-RSP is then addressed as follows: first, a brute-force algorithm is called for rings containing 3 hubs; second, another brute-force algorithm is used for 4-hub rings;

and third, the ILP formulation where the ring has at least 5 hubs is solved. The best solution found is then returned.

2.4.1 Brute-force algorithms of polynomial time complexity for the cases of three and four hubs

A feasible solution to 1-S-RSP has a three-hub ring only if all these hubs are certain. Indeed, if one of these three hubs is uncertain, then its failure leaves only two hubs which are not enough to form a ring. The case of a four-hub ring is also special in the sense that a single backup edge can be used to cope with the failure of two different hubs. As shown in Figure 2.2, the backup edge that joins nodes 1 and 3 is used when either hub 2 or 4 fails. It can also be seen that the number of backup edges (two) is not equal to the number of uncertain hubs (three). We introduce two brute-force algorithms to deal with the particular cases where the ring has three or four hubs. Then, the proposed ILP formulation assumes that the ring has at least 5 hubs and the cutoff parameter of the solver is set to the best solution value found by the brute-force algorithms. Hence, the solver stops as soon as it finds out that there is no optimal solution with at least 5 hubs. Algorithms 1 and 2 are based on an exhaustive enumeration to determine an optimal solution with 3 hubs in $\mathcal{O}(n^3)$ time and 4 hubs in $\mathcal{O}(n^4)$ time.

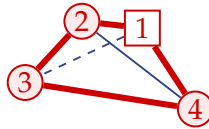


Figure 2.2: 1-S-RSP instance with $\tilde{V} = \{2, 3, 4\}$ where the dashed backup edge is used when hub 2 or 4 fails.

2.4.2 An ILP formulation for at least five hubs

The core of the ILP formulation of 1-S-RSP is inspired from [41], and uses mostly the same notations. All the decision variables of 1-S-RSP are Boolean, and in the sequel, \mathbb{B} denotes the discrete set $\{0, 1\}$. The x variables are used to model the selection of the edges of the ring and the y variables model the status of the nodes (to distinguish hubs and terminals), as well as the connection of terminals to hubs. The additional new x' variables are introduced to model the selection of the backup edges. More precisely, for all ij in E , x_{ij} is set to one if and only if edge ij belongs to the ring. For the sake of readability, we may sometimes refer to x_{ij} for $i > j$, in which case x_{ij} is replaced with x_{ji} in the computer implementation. For all $(i, j) \in V^2$, variable y_{ij} is set to one if and only if terminal i is assigned to hub $j \neq i$. The variable y_{jj} is set to one if j is selected as a hub, and it is set to 0

Algorithm 1: A brute-force algorithm to find a three-hub optimal solution to 1-S-RSP

```

1  $bestobjval \leftarrow +\infty$ 
2  $bestsol \leftarrow \emptyset$ 
3 foreach  $v_1 \in V \setminus (\tilde{V} \cup \{1\})$  do
4   foreach  $v_2 \in V \setminus (\tilde{V} \cup \{1\}) : v_2 > v_1$  do
5      $currentobjval \leftarrow o_1 + o_{v_1} + o_{v_2} + r_{1v_1} + r_{1v_2} + r_{v_1v_2}$ 
6     foreach  $v \in V \setminus \{1, v_1, v_2\}$  do
7        $starcost \leftarrow s_{v1}$ 
8       if  $s_{vv_1} < starcost$  then
9          $starcost \leftarrow s_{vv_1}$ 
10      if  $s_{vv_2} < starcost$  then
11         $starcost \leftarrow s_{vv_2}$ 
12       $currentobjval \leftarrow currentobjval + starcost$ 
13    if  $currentobjval < bestobjval$  then
14       $bestobjval \leftarrow currentobjval$ 
15       $bestsol \leftarrow \{1, v_1, v_2\}$ 
16 return  $bestsol, bestobjval$ 

```

if it is a terminal. Finally, for all $(i, k) \in V^2$ such that $i < k$, variable x'_{ik} indicate whether hubs i and k are the common neighbor of an uncertain hub and thus a backup connection is established between them. Note that the backup edges do not need to form a cycle, as can be seen in the example of Figure 2.1. For the sake of illustration, Figure 2.3 shows all nonzero x , x' , and y variables of an 11-node instance of 1-S-RSP, except nonzero y_{ii} variables. The number of edges in the ring is equal to the number of hubs, and there are six backup edges (i.e., nonzero x'_{ij} variables) because there are six uncertain hubs.

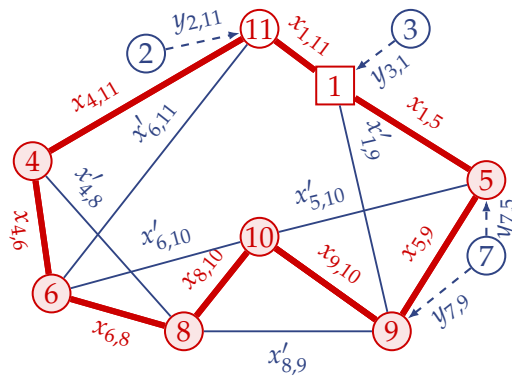


Figure 2.3: Nonzero variables for a solution to an 11-node instance of 1-S-RSP

Algorithm 2: A brute-force algorithm to find a four-hub optimal solution to 1-S-RSP

```

1 bestobjval  $\leftarrow +\infty$ 
2 bestsol  $\leftarrow \emptyset$ 
3 for  $v_1 = 2$  to  $n - 2$  do
4   for  $v_2 = v_1 + 1$  to  $n - 1$  do
5     for  $v_3 = v_2 + 1$  to  $n$  do
6       currentobjval  $\leftarrow o_1 + o_{v_1} + o_{v_2} + o_{v_3}$ 
7        $H \leftarrow \{1, v_1, v_2, v_3\}$ 
8       foreach  $v \in V \setminus \{1, v_1, v_2, v_3\}$  do
9         costcertain  $\leftarrow +\infty$ 
10        costuncertain  $\leftarrow [+ \infty, + \infty]$ 
11        foreach  $h \in H$  do
12          if  $h \in V \setminus \tilde{V}$  and  $s_{vh} < \textit{costcertain}$  then
13             $\textit{costcertain} \leftarrow s_{vh}$ 
14          else if  $h \in \tilde{V}$  then
15            if  $s_{vh} < \textit{costuncertain}[1]$  then
16               $\textit{costuncertain}[2] \leftarrow \textit{costcertain}[1]$ 
17               $\textit{costuncertain}[1] \leftarrow s_{vh}$ 
18            else if  $s_{vh} < \textit{costuncertain}[2]$  then
19               $\textit{costuncertain}[2] \leftarrow s_{vh}$ 
20          currentobjval  $\leftarrow \textit{currentobjval} +$ 
            $\min(\textit{costcertain}, \textit{costuncertain}[1] + \textit{costuncertain}[2])$ 
           /* Testing three solutions based on 1,  $v_1$ ,  $v_2$  and  $v_3$  */
           */
21           $\textit{sol}_1 \leftarrow r_{1v_1} + r_{v_1v_2} + r_{v_2v_3} + r_{1v_3}$ 
22          if  $v_1 \in \tilde{V}$  or  $v_3 \in \tilde{V}$  then
23             $\textit{sol}_1 \leftarrow \textit{sol}_1 + r_{1v_2}$ 
24          if  $1 \in \tilde{V}$  or  $v_2 \in \tilde{V}$  then
25             $\textit{sol}_1 \leftarrow \textit{sol}_1 + r_{v_1v_3}$ 
26           $\textit{sol}_2 \leftarrow r_{1v_1} + r_{v_1v_3} + r_{v_2v_3} + r_{1v_2}$ 
27          if  $v_1 \in \tilde{V}$  or  $v_2 \in \tilde{V}$  then
28             $\textit{sol}_2 \leftarrow \textit{sol}_2 + r_{1v_3}$ 
29          if  $1 \in \tilde{V}$  or  $v_3 \in \tilde{V}$  then
30             $\textit{sol}_2 \leftarrow \textit{sol}_2 + r_{v_1v_2}$ 
31           $\textit{sol}_3 \leftarrow r_{1v_2} + r_{v_1v_2} + r_{v_1v_3} + r_{1v_3}$ 
32          if  $v_2 \in \tilde{V}$  or  $v_3 \in \tilde{V}$  then
33             $\textit{sol}_3 \leftarrow \textit{sol}_3 + r_{1v_1}$ 
34          if  $1 \in \tilde{V}$  or  $v_1 \in \tilde{V}$  then
35             $\textit{sol}_3 \leftarrow \textit{sol}_3 + r_{v_2v_3}$ 
36          if  $\textit{currentobjval} + \min(\textit{sol}_1, \textit{sol}_2, \textit{sol}_3) < \textit{bestobjval}$  then
37             $\textit{bestobjval} \leftarrow \textit{currentobjval} + \min(\textit{sol}_1, \textit{sol}_2, \textit{sol}_3)$ 
38             $\textit{bestsol} \leftarrow \{1, v_1, v_2, v_3\}$ 
39 return bestsol, bestobjval

```

The ILP formulation for 1-S-RSP is as follows:

$$\text{Min } z = \sum_{ij \in E} r_{ij}(x_{ij} + x'_{ij}) + \sum_{i \in V} o_i y_{ii} + \sum_{(i,j) \in A} s_{ij} y_{ij}$$

$$\sum_{\substack{j \in V \\ i < j}} x_{ij} + \sum_{\substack{j \in V \\ i > j}} x_{ji} = 2y_{ii} \quad \forall i \in V \quad (2.1)$$

$$x(\delta(S)) \geq 2 \sum_{j \in S} y_{ij} \quad \forall S \subset V, 1 \notin S, i \in S \quad (2.2)$$

$$\sum_{\substack{j \in V \setminus \tilde{V} \\ i \neq j}} 2y_{ij} + \sum_{\substack{j \in \tilde{V} \\ i \neq j}} y_{ij} = 2(1 - y_{ii}) \quad \forall i \in V \quad (2.3)$$

$$\sum_{i \in V} \sum_{j \in V, i < j} x_{ij} \geq 5 \quad (2.4)$$

$$x_{ij} + x_{jk} \leq 1 + x'_{ik} \quad \forall (i, j, k) \in \tilde{J} \quad (2.5)$$

$$y_{ij} \leq y_{jj} \quad \forall (i, j) \in A \quad (2.6)$$

$$y_{11} = 1 \quad (2.7)$$

$$y_{ij} \in \mathbb{B} \quad \forall (i, j) \in V^2$$

$$x_{ij} \in \mathbb{B} \quad \forall ij \in E$$

$$x'_{ij} \in \mathbb{B} \quad \forall ij \in E$$

In the sequel, for any subset S of V , we define $\delta(S)$ as the set of all the edges of E that have one endpoint in S and another endpoint in $V \setminus S$. We also define $x(F) = \sum_{i \in F, i < j} x_{ij}$ for all $F \subseteq E$. Constraints (2.1) enforce that each hub is incident to two edges in the ring, and each terminal is incident to zero edges. Constraints (2.2) are connectivity constraints, they are identical to the ones in [40]. They state that any set S that does not contain the depot must be connected to $V \setminus S$ by at least two edges, if S contains a hub i . In that case, y_{ii} is equal to one, and so is $\sum_{j \in S} y_{ij}$. The latter expression appears in the right-and side of (2.2) as the resulting inequalities are stronger when $\sum_{j \in S} y_{ij}$ is used instead of just y_{ii} . Constraints (2.3) enforce that each terminal is connected to either one certain hub or two distinct uncertain hubs. Constraint (2.4) states that the ring contains at least five hubs. Constraints (2.5) enforce that edge ik has to be selected as a backup edge if node j is an uncertain hub adjacent to hubs i and k in the ring; there are $\frac{1}{2}|\tilde{V}|(n-1)(n-2) = \mathcal{O}(|\tilde{V}|n^2)$ such inequalities. Constraints (2.6) enforce that terminals can be connected to hubs only. Constraint (2.7) sets the depot as a hub, and the remaining constraints define the domain of the variables. This model can be solved using a branch-and-cut or a Branch-and-Benders-cut approach where subtour elimination constraints

are relaxed and separated on-the-fly when they are found to be violated.

2.5 A Branch-and-Benders-cut decomposition of 1-S-RSP

In the framework of Benders decomposition described in Section 1.6, a new branch-and-bound search tree is built from scratch for the master problem, which may not be very efficient. Consequently, we devise a Branch-and-Benders-cut decomposition for 1-S-RSP that differs from the classical Benders decomposition by generating a unique branch-and-bound search tree for the master problem where the subproblem is solved whenever an integer solution is found. The corresponding cuts are added to the master problem following a cutting-plane procedure. This method has been used by Fortz *et al.* [23] to address a multi-layer network design problem. Figure 2.4 illustrates with a flowchart the Branch-and-Benders-cut decomposition scheme. In the context of this chapter, the feasibility cuts are subtour elimination constraints and the optimality cuts enforce the contribution of the star cost in the Master problem. Finally, connectivity cuts can also be added to the Master problem at fractional nodes, using the Edmonds-Karp Min cut algorithm [42]. In this section, we propose new subtour elimination constraints for the master problem, and then state the master problem and subproblem of the decomposition. Subsection 2.5.2 shows how to rewrite the latter one as a linear program by adding valid inequalities. In Subsections 2.5.3 and 2.5.4, after decomposing the subproblem into $n + 1$ smaller linear problems, we give the solution of the dual of each part. Finally, Subsection 2.5.5 presents a polynomial algorithm to solve the subproblem, Subsection 2.5.6 shows the corresponding optimality cuts, and a polyhedral analysis of the valid inequalities added to the subproblem is given in Subsection 2.5.7.

2.5.1 Description of the master problem and the subproblem

If the ring is known, *i.e.*, the y_{ii} variables and the x_{ij} variables are fixed, determining all the remaining variables is easy. Indeed, when the ring has 5 or more hubs, x'_{ik} can be set to one if and only if $j \in \tilde{V}$ and $y_{jj} = x_{ij} = x_{jk} = 1$. Otherwise, x'_{ik} is set to zero. Furthermore, for each terminal i , the optimal decision is to select the option that incurs the minimum cost: the first option consists of connecting terminal i to the closest certain hub, and the second one is to connect it to its two nearest uncertain hubs. As a consequence, we devise a Benders decomposition of the ILP formulation of 1-S-RSP, in which the master problem is to determine the ring, *i.e.* the y_{ii} and x_{ij} variables, and the subproblem is to complement the ring with backup edges and to connect the terminals to the ring at minimum cost.

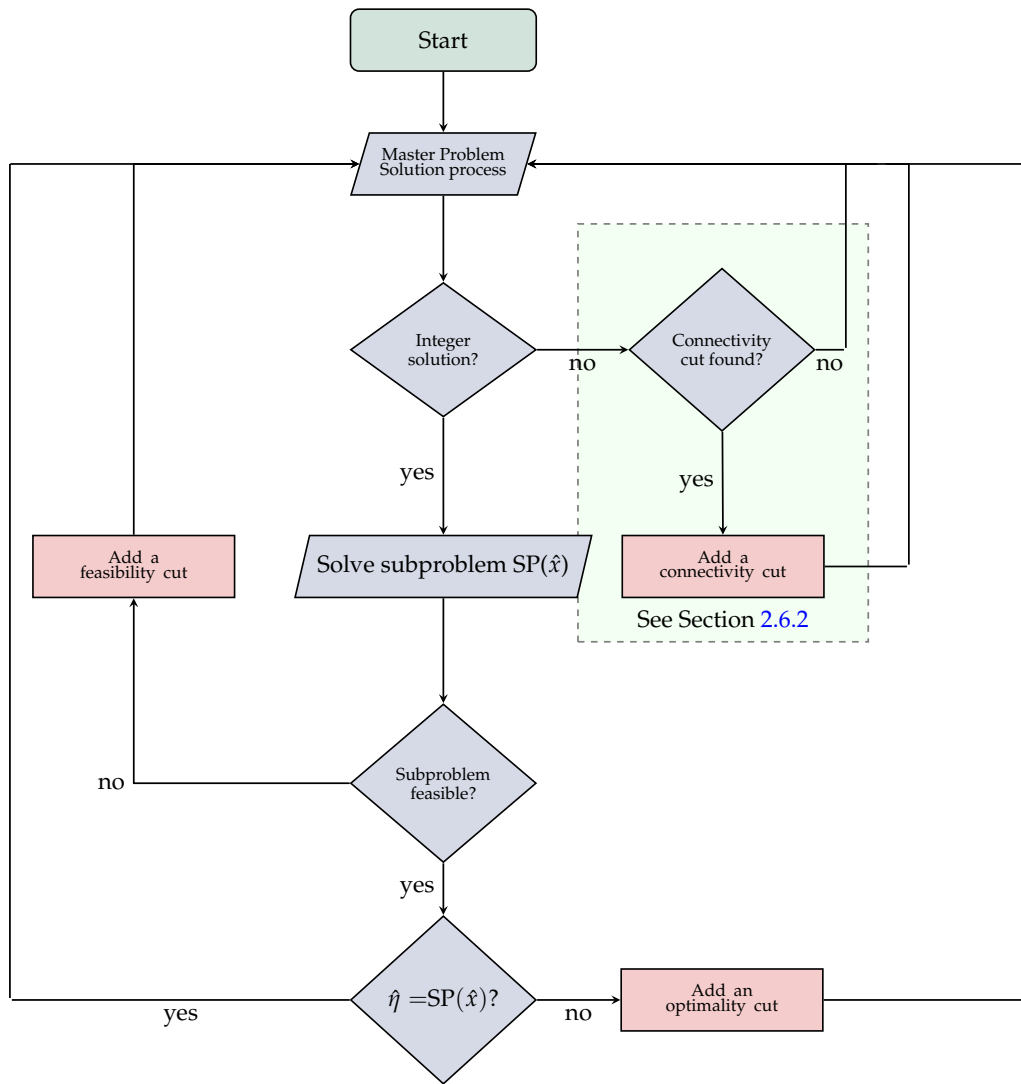


Figure 2.4: Branch-and-Benders-cut decomposition scheme

A new variable $\lambda \in \mathbb{R}_+$ is added to the master problem to account for the contribution of the subproblem to the objective function as in the classical Benders decomposition method.

Before we can apply the Benders decomposition, the subtour elimination constraints (2.2) must be replaced because they mix variables of the master problem (x variables) and variables of the subproblems y_{ij} (when $i \neq j$). We then propose the following constraints to enforce subtour elimination in the master problem:

$$x(S) \leq |S| - 1 \quad \forall S \subseteq V, 1 \notin S \quad (2.8)$$

$$x(S) \leq |S| - y_{ii} \quad \forall S \subseteq V, 1 \in S, i \in V \setminus S \quad (2.9)$$

In order to show that these inequalities are valid subtour elimination constraints, let RP be the ILP formulation of 1-S-RSP where constraints (2.2) are relaxed. For each feasible solution to RP , let $G_x = (V, E_x)$ be the non-directed graph on the vertex set V , where $E_x = \{ij \in E : x_{ij} = 1\}$. This graph represents the edges of the ring and may contain one or more cycles. A cycle in G_x that does not contain the depot is called a *subtour*. A solution to 1-S-RSP is a solution to RP in which the associated graph G_x is free of subtours. By constraints (2.1), the vertices in G_x have degree 0 or 2, so any feasible solution to RP satisfies the following properties:

Property 1: $|S| - x(S) \geq 0 \quad \forall S \subseteq V$.

Proof: We prove this property by contradiction. Assume that there exists $S \subseteq V$ such that $|S| - x(S) < 0$. The quantity $x(S)$ is the number of edges of G_x with both endpoints in S , which is also the number of edges of $G_x[S]$, the subgraph of G_x induced by S . Since each of these edges contributes 2 to the degree sum, we have

$$|S| - \frac{1}{2} \sum_{i \in S} d_{G_x}(i) < 0 \text{ which is equivalent to: } 2|S| < \sum_{i \in S} d_{G_x}(i)$$

Where $d_{G_x}(i)$ is the degree of vertex $i \in V$ in the graph G_x . After dividing the last inequality by $|S|$, we obtain that the average degree of the vertices in S is strictly larger than 2 in G_x , which is a contradiction as constraints (2.1) enforce that the maximum degree in the ring is 2.

■

Property 2: $|S| - x(S) = 0$ if and only if $G_x[S]$ is a collection of cycles.

Proof: We first assume that $G_x[S]$ is a collection of cycles. This implies that $G_x[S]$ has $|S|$ edges, so $|S| = x(S)$. Conversely, if $2|S| = \sum_{i \in S} d_{G_x}(i)$, then the average degree is 2 in $G_x[S]$, so each node of S must have degree 2, hence $G_x[S]$ is a collection of cycles.

■

If $G_x[S]$ is not a collection of cycles, then $|S| - x(S) \geq 1$ because the left-hand side is an integer quantity. It can be deduced that if $G_x[S]$ is a collection of cycles and there exists at least one hub in $V \setminus S$, then the current solution to RP has a subtour. So to prevent subtours, one could enforce the following constraints:

$$|S| - x(S) \geq \frac{1}{|V \setminus S|} \sum_{i \in V \setminus S} y_{ii} \quad \forall S \subseteq V$$

Where $\frac{1}{|V \setminus S|} \sum_{i \in V \setminus S} y_{ii}$ is in $(0, 1]$ if there exists a hub in $V \setminus S$, and zero otherwise. If S is a subset of V that does not contain the depot, then in that case, $|S| - x(S)$ is strictly positive, because $1 \in V \setminus S$ and $y_{11} = 1$. So for all the sets S that do not contain 1, we can strengthen the above inequalities to (2.8). These constraints dominate the above inequalities, since $1 \geq \frac{1}{|V \setminus S|} \sum_{i \in V \setminus S} y_{ii}$. Now, if S contains 1,

we can also strengthen the inequalities to (2.9). Again, these constraints dominate the above inequalities if we choose $i \in V \setminus S$ such that $y_{ii} = 1$, as we have $y_{ii} \geq \frac{1}{|V \setminus S|} \sum_{j \in V \setminus S} y_{jj}$.

The master problem can then be stated as:

$$\text{Min } \sum_{ij \in E} r_{ij} x_{ij} + \sum_{i \in V} o_i y_{ii} + \lambda$$

Subject to (2.1), (2.4), (2.7), (2.8), (2.9) and

$$\begin{aligned} y_{ii} &\in \mathbb{B} & \forall i \in V \\ x_{ij} &\in \mathbb{B} & \forall ij \in E \\ \lambda &\in \mathbb{R}_+ \end{aligned}$$

After solving the master problem, the numerical value of the x_{ij} and y_{jj} variables are stored as \hat{x}_{ij} and \hat{y}_{jj} , then passed to the subproblem, whose primal can be stated as:

$$\begin{aligned} \text{Min } \lambda &= \sum_{ij \in E} r_{ij} x'_{ij} + \sum_{(i,j) \in A} s_{ij} y_{ij} \\ \sum_{\substack{j \in V \setminus \tilde{V} \\ i \neq j}} 2y_{ij} + \sum_{\substack{j \in \tilde{V} \\ i \neq j}} y_{ij} &= 2(1 - \hat{y}_{ii}) & \forall i \in V \end{aligned} \quad (2.10)$$

$$x'_{ik} \geq \hat{x}_{ij} + \hat{x}_{jk} - 1 \quad \forall (i, j, k) \in \tilde{J} \quad (2.11)$$

$$y_{ij} \leq \hat{y}_{jj} \quad \forall (i, j) \in A, \quad (2.12)$$

$$y_{ij} \in \mathbb{B} \quad \forall (i, j) \in A \quad (2.13)$$

$$x'_{ij} \in \mathbb{B} \quad \forall ij \in E$$

Constraints (2.10), (2.11) and (2.12) are derived from (2.3), (2.5) and (2.6) respectively. This subproblem is easy to solve. Indeed, for each uncertain hub j having hubs i and k as neighbors in the ring, we should set $x'_{ik} = 1$.

2.5.2 Linear programming formulation of the subproblem

If node i is a terminal (*i.e.*, $\hat{y}_{ii} = 0$), we define m_i and m'_i as the two distinct closest hubs in \tilde{V} , and m_i^* as the closest hub in $V \setminus \tilde{V}$, if they exist. If not, the corresponding values are set to zero.

$$\bullet \quad m_i = \begin{cases} 0 & \text{if } |\{j \in \tilde{V} : \hat{y}_{jj} = 1\}| = 0 \\ \arg \min_{j \in \tilde{V} : \hat{y}_{jj} = 1} s_{ij} & \text{otherwise} \end{cases}$$

$$\bullet m'_i = \begin{cases} 0 & \text{if } |\{j \in \tilde{V} \setminus \{m_i\} : \hat{y}_{jj} = 1\}| = 0 \\ \arg \min_{j \in \tilde{V} \setminus \{m_i\} : \hat{y}_{jj} = 1} s_{ij} & \text{otherwise} \end{cases}$$

$$\bullet m_i^* = \begin{cases} 0 & \text{if } |\{j \in V \setminus \tilde{V} : \hat{y}_{jj} = 1\}| = 0 \\ \arg \min_{j \in V \setminus \tilde{V} : \hat{y}_{jj} = 1} s_{ij} & \text{otherwise} \end{cases}$$

Because m_i , m'_i , and m_i^* are indices in $V \cup \{0\}$, we let 0 be a dummy vertex for which $s_{i,0}$ is infinite for all $i \in V$. Node 0 cannot be a hub or a terminal and it does not appear in any solution to 1-S-RSP, as its only purpose is to let s_{i,m_i} be infinite when there is no uncertain hub, s_{i,m'_i} be infinite when there is at most one uncertain hub, and s_{i,m_i^*} be infinite when there is no certain hub in the ring. If $s_{i,m_i} + s_{i,m'_i} < s_{i,m_i^*}$, then terminal i is connected to its two closest uncertain neighbors in the ring. Otherwise, terminal i is connected to the closest certain hub in the ring. Since the ring has at most three hubs, $s_{i,m_i} + s_{i,m'_i}$ and s_{i,m_i^*} cannot be both set to infinity.

If $\{j \in \tilde{V} : \hat{y}_{jj} = 1\}$ is empty, then the linear programming relaxation of the subproblem is integral. Otherwise, the subproblem, originally an integer linear program, can be stated as a linear program by adding the following constraint: $y_{ik} \leq \sum_{j \in \tilde{V} \setminus \{k\} : \hat{y}_{jj} = 1} y_{ij}$ for all $i \in V$ such that $\hat{y}_{ii} = 0$, and for all $k \in \tilde{V}$ such that $s_{ik} = \min_{j \in \tilde{V} : \hat{y}_{jj} = 1} s_{ij}$. This constraint states that if terminal i is connected to an uncertain hub, then it should be connected to at least another one. If $\{j \in \tilde{V} \setminus \{k\} : \hat{y}_{jj} = 1\}$ is empty, then this inequality reads $y_{ik} \leq 0$ which forbids terminal i to be connected to the unique uncertain hub k .

Since (2.12) dominates (2.13), and because the objective function “pushes” the x' variables downward, integrality constraints can be dropped, and it is shown that the linear relaxation of the subproblem (with the new constraints above) has an integral optimal solution.

We can take advantage of the subproblem’s structure and decompose it as $n + 1$ independent subsubproblems: IP_0 sets the backup edges, and for all $i \in V$, IP_i sets the y_{ij} variables for all $j \in V \setminus \{i\}$. The objective function of the subproblem is then the sum of the $n + 1$ objective values. More precisely, we have:

$$(\text{IP}_0) : \begin{cases} \text{Min } \sum_{ij \in E} r_{ij} x'_{ij} \\ x'_{ik} \geq \hat{x}_{ij} + \hat{x}_{jk} - 1 \\ x'_{ij} \in \mathbb{B} \end{cases} \quad \begin{matrix} \forall (i, j, k) \in \tilde{\mathcal{J}} \\ \forall ij \in E \end{matrix}$$

$$(\text{IP}_i) : \begin{cases} \text{Min } \sum_{j \in V \setminus \{i\}} s_{ij} y_{ij} \\ \sum_{\substack{j \in V \setminus \tilde{V} \\ i \neq j}} 2y_{ij} + \sum_{\substack{j \in \tilde{V} \\ i \neq j}} y_{ij} = 2(1 - \hat{y}_{ii}) \\ -y_{ij} \geq -\hat{y}_{jj} & \forall j \in V \setminus \{i\} \\ y_{ij} \in \mathbb{B} & \forall j \in V \setminus \{i\} \end{cases} \quad \begin{matrix} (2.10) \\ (2.12) \end{matrix}$$

First, we show that the LP-relaxation of IP_0 has an integer solution. Indeed, IP_0 is a minimization problem on x' variables, they appear in the objective function with a nonnegative coefficient and the constraints allow to set them to their lower bound, which are all integer. Hence, setting x'_{ik} to $\max(0, \hat{x}_{ij} + \hat{x}_{jk} - 1)$ yields an optimal integer solution. Note that in the LP-relaxation of IP_0 , we can also ignore $x'_{ik} \leq 1$, as there is no optimal solution with $x'_{ik} > 1$.

Second, for any given $i \in V$, we search for an optimal solution to IP_i . If $\hat{y}_{ii} = 1$, then the unique solution is to set y_{ij} to zero for all $j \in V \setminus \{i\}$ and it incurs a zero optimal objective value. This solution is also the only one for the LP-relaxation of IP_i .

Now, let i be a terminal, *i.e.*, $\hat{y}_{ii} = 0$. We distinguish two sub-cases:

- if \tilde{V} is empty, or y_{ij} is set to zero for all $j \in \tilde{V} \setminus \{i\}$ by constraint (2.12), then constraint (2.10) becomes $\sum_{\substack{j \in V \setminus \tilde{V} \\ i \neq j}} y_{ij} = 1$. Then IP_i and its LP-relaxation have the same optimal solution, which consists in setting $y_{im_i^*}$ to one and all other decision variables to zero, the optimal objective value being $s_{im_i^*}$.
- if $V \setminus (\tilde{V} \cup \{i\})$ is empty, or y_{ij} is set to zero for all $j \in V \setminus (\tilde{V} \cup \{i\})$ by constraint (2.12), then constraint (2.10) becomes $\sum_{j \in \tilde{V} \setminus \{i\}} y_{ij} = 2$. Then IP_i and its LP-relaxation have an optimal solution in common, that consists in setting y_{im_i} and $y_{im_i'}$ to one and all other decision variables to zero, the optimal objective value being $s_{im_i} + s_{im_i'}$.

The remaining cases are less straightforward: node i is a terminal and both the set of uncertain hubs, *i.e.*, $(\tilde{V} \setminus \{i\}) \cap \{j \in V : \hat{y}_{jj} = 1\}$ and the set of certain hubs, *i.e.*, $(V \setminus (\tilde{V} \cup \{i\})) \cap \{j \in V : \hat{y}_{jj} = 1\}$ are nonempty. In that case, the LP-relaxation of IP_i may have a non-integer optimal solution. Even if solving IP_i remains straightforward, we need to formulate it as a linear program, so that its dual can be solved, as separating Benders optimality cuts requires an optimal solution to the subproblem's dual. To this end, we propose a pure linear

programming formulation for IP_i , by adding valid inequalities and then showing that the LP-relaxation of this improved formulation is integral. We define $\Omega_i = \{j \in \tilde{V} \setminus \{i\} : \hat{y}_{jj} = 1, s_{ij} = s_{im_i}\}$ as the set containing all the uncertain hubs that are at minimum distance from terminal i , including hub m_i if it exists. We now introduce valid inequalities for IP_i to enforce that if a minimum-cost uncertain hub has been selected to connect terminal i , then at least another uncertain hub should be selected as well. This is due to the requirement that any terminal should either be connected to a certain hub, or to two uncertain hubs.

$$y_{ik} \leq \sum_{\substack{j \in \tilde{V} \setminus \{k\} \\ \hat{y}_{jj}=1}} y_{ij} \quad \forall k \in \Omega_i \quad (2.14)$$

Let LP_i^+ be the LP-relaxation of IP_i with the addition of (2.14). Relaxing the integrality constraints yields to $y_{ij} \leq 1$ for all $j \in V \setminus \{i\}$, but it can be observed that (2.12) dominate these constraints, so they can be dropped and all decision variables in LP_i^+ are simply subject to non-negativity constraints. We now show that LP_i^+ has an optimal solution that is integral. To this end, we use the following lemma:

Lemma 1. *In any optimal solution to LP_i^+ , $y_{ij} = 0$ for all j in $V \setminus (\tilde{V} \cup \{i\})$ such that $s_{ij} > s_{im_i^*}$ and for all j in $\tilde{V} \setminus \{i\}$ such that $s_{ij} > s_{im_i'}$.*

Proof Lemma 1 is proven by contradiction. Let $(y_{ij})_{j \in V \setminus \{i\}}$ be an optimal solution to LP_i^+ . We first assume that there exists q in $V \setminus (\tilde{V} \cup \{i\})$ with $s_{iq} > s_{im_i^*}$ and $y_{iq} > 0$. Then we can transform this optimal solution by increasing $y_{im_i^*}$ by y_{iq} and by setting y_{iq} to zero. The resulting solution is still feasible, but since its objective value has decreased by $(s_{iq} - s_{im_i^*})y_{iq} > 0$, the original solution cannot be optimal which is a contradiction.

We now assume that $(y_{ij})_{j \in V \setminus \{i\}}$ is an optimal solution to LP_i^+ such that there exists q in $\tilde{V} \setminus \{i\}$ such that $s_{iq} > s_{im_i'} \geq s_{m_i}$ and $y_{iq} > 0$. We consider the following two subcases. If $s_{im_i} = s_{im_i'}$, then both m_i and m_i' are subjected to constraint (2.14). If one of them is not tight, say m_i' , then we can transform the current optimal solution to LP_i^+ into a feasible solution by increasing $y_{im_i'}$ by $\varepsilon = \min \left(y_{iq}, \frac{1}{2} \sum_{j \in \Omega_i \setminus \{m_i'\}} y_{ij} - \frac{1}{2} y_{im_i'} \right) > 0$ and by decreasing y_{iq} by ε . In the process, the objective value of the new feasible solution has been decreased by $\varepsilon(s_{iq} - s_{im_i'}) > 0$, which is a contradiction. The argument and the conclusion are the same if constraint (2.14) is not tight for m_i . This means that constraint (2.14) has

to be tight for both m_i and m'_i . The corresponding two equalities yield $y_{ij} = 0$ for all $j \in \tilde{V} \setminus \{i, m_i, m'_i\}$, which again contradicts $y_{iq} > 0$. The second subcase occurs when $s_{im_i} < s_{im'_i}$. Now, constraint (2.14) does not apply to m'_i as $m'_i \notin \Omega_i$, so we can transform the optimal solution to LP_i^+ by increasing $y_{im'_i}$ by y_{iq} and by setting y_{iq} to zero. The resulting solution is feasible and its objective value has decreased by $(s_{iq} - s_{im'_i})y_{iq} > 0$, which again contradicts the fact that the initially considered solution was optimal. ■

Corollary 1.1. *Any optimal solution to LP_i can be transformed into another optimal solution such that $y_{im'_i}$ is the only nonzero y_{ij} variable for all j in $V \setminus (\tilde{V} \cup \{i\})$ and $y_{im_i} = y_{im'_i}$ are the only nonzero variables for all j in $\tilde{V} \setminus \{i\}$.*

Proof By Lemma 1, the only values of $j \in V \setminus (\tilde{V} \cup \{i\})$ such that y_{ij} can be nonzero in an optimal solution to LP_i^+ are those for which $s_{ij} = s_{im'_i}$. If there are more than one such y_{ij} variables, we can transform any optimal solution by updating $y_{im'_i}$ to $\sum_{j \in V \setminus (\tilde{V} \cup \{i\})} y_{ij}$ and by setting y_{ij} to zero for all $j \in V \setminus (\tilde{V} \cup \{i, m'_i\})$.

This transformation yields a feasible solution and does not change its optimal objective value, since $s_{ij} = s_{im'_i}$ for all j in $V \setminus (\tilde{V} \cup \{i\})$. Still by Lemma 1, the only values of $j \in \tilde{V} \setminus \{i\}$ such that y_{ij} can be nonzero in an optimal solution to LP_i^+ are those for which $s_{ij} \leq s_{im'_i}$.

- We first consider the subcase in which $s_{im_i} = s_{im'_i}$. This implies that constraint (2.14) applies for all $j \in \tilde{V} \setminus \{i\}$ such that y_{ij} can be nonzero. If constraint (2.14) is tight for both m_i and m'_i , then the corresponding two equalities imply that $y_{im_i} = y_{im'_i}$ are the only nonzero variables for all j in $\tilde{V} \setminus \{i\}$. If constraint (2.14) is not tight for m'_i , or similarly, for m_i , then there exists $q \in \tilde{V} \setminus \{i, m_i, m'_i\}$ such that $y_{iq} > 0$ and the current optimal solution to LP_i^+ can be transformed into a feasible solution by increasing $y_{im'_i}$ by $\varepsilon = \min \left(y_{iq}, \frac{1}{2} \sum_{j \in \tilde{V} \setminus \{i, m'_i\}} y_{ij} - \frac{1}{2} y_{im'_i} \right) > 0$ and by decrementing y_{iq} by ε . This transformation yields a feasible solution to LP_i^+ , which is still optimal as the objective value of the initial optimal solution has been added $(s_{im'_i} - s_{iq})\varepsilon = 0$, but the new optimal solution has one nonzero variable less than the original one. This process can be repeated at most $|\tilde{V} \setminus \{i, m_i, m'_i\}|$ times to produce a solution in which $y_{im_i} = y_{im'_i}$ are the only nonzero variables for all j in $\tilde{V} \setminus \{i\}$.

- We now consider the subcase in which $s_{im_i} < s_{im'_i}$. This implies that constraint (2.14) applies to m_i only, so any optimal solution to LP_i^+ can be transformed by updating $y_{im'_i}$ to $\sum_{j \in \tilde{V} \setminus \{i, m_i\}} y_{ij}$ and by setting y_{ij} to zero for all $j \in \tilde{V} \setminus \{i, m_i, m'_i\}$. This yields a feasible solution whose objective value is the same as the one of the original optimal solution. As a result, constraint (2.14), which only applies to m_i , is tight. It implies that $y_{im_i} = y_{im'_i}$ and these two are the only nonzero variables for all j in $\tilde{V} \setminus \{i\}$. ■

Corollary 1.2. *Any optimal solution to LP_i^+ can be transformed into an integral optimal solution.*

Proof By corollary 1.1, any optimal solution to LP_i^+ can be transformed into an optimal solution having at most three nonzero decision variables, namely $y_{im'_i}$, y_{im_i} and $y_{im_i^*}$. It has also been shown that $y_{im_i} = y_{im'_i}$. Hence, by dropping all the zero decision variables, constraint (2.10) reduces to $y_{im_i^*} + y_{im_i} = 1$, which means that variable y_{im_i} can be eliminated. By definition of m_i^* , constraint (2.12) is $y_{im_i^*} \leq 1$, so LP_i^+ is a single continuous variable linear program which is to minimize $s_{im_i^*} y_{im_i^*} + (s_{im_i} + s_{im'_i})(1 - y_{im_i^*}) = (s_{im_i} + s_{im'_i}) + (s_{im_i^*} - s_{im_i} - s_{im'_i})y_{im_i^*}$ subject to $0 \leq y_{im_i^*} \leq 1$. So when $s_{im_i^*} \leq s_{im_i} + s_{im'_i}$, the optimal solution is to set $y_{im_i^*} = 1$, which yields an optimal objective value of $s_{im_i^*}$ and implies that $y_{im_i} = y_{im'_i} = 0$. When $s_{im_i^*} > s_{im_i} + s_{im'_i}$, the optimal solution is to set $y_{im_i^*} = 0$, which yields an optimal objective value of $s_{im_i} + s_{im'_i}$ and implies that $y_{im_i} = y_{im'_i} = 1$. This shows that in all cases, an integral optimal solution to LP_i^+ can be found. ■

2.5.3 Computing the dual of LP_0 and LP_i^+ for all the nodes i in V

The dual DLP_0 of LP_0 is

$$(DLP_0) : \begin{cases} \text{Maximize} & \sum_{(i,j,k) \in \tilde{J}} (\hat{x}_{ij} + \hat{x}_{jk} - 1) \beta_{ijk} \\ \sum_{j \in \tilde{V}: j \neq i, j \neq k} \beta_{ijk} & \leq r_{ik}, & \forall ik \in E \\ \beta_{ijk} & \geq 0, & \forall (i, j, k) \in \tilde{J} \end{cases} \quad (2.15)$$

For all $i \in V$, the dual of LP_i^+ is denoted by DLP_i^+ and is defined as follows:

If $\hat{y}_{ii} = 1$,

$$(\text{DLP}_i^+) : \begin{cases} \text{Max } \lambda_i = 2(1 - \hat{y}_{ii})\alpha_i - \sum_{j \in V \setminus \{i\}} \hat{y}_{jj}\gamma_{ij} \\ 2\alpha_i - \gamma_{ij} \leq s_{ij} & \forall j \in V \setminus (\tilde{V} \cup \{i\}) \\ \alpha_i - \gamma_{ij} \leq s_{ij} & \forall j \in V \setminus \{i\} \\ \alpha_i \in \mathbb{R} \\ \gamma_{ij} \geq 0 & \forall j \in V \setminus \{i\} \end{cases}$$

If $\hat{y}_{ii} = 0$,

$$(\text{DLP}_i^+) : \begin{cases} \text{Maximize } \lambda_i = 2(1 - \hat{y}_{ii})\alpha_i - \sum_{j \in V \setminus \{i\}} \hat{y}_{jj}\gamma_{ij} \\ 2\alpha_i - \gamma_{ij} \leq s_{ij} & \forall j \in V \setminus (\tilde{V} \cup \{i\}) \\ \alpha_i - \gamma_{ij} - \delta_{ij} + \sum_{k \in \Omega_i \setminus \{j\}} \delta_{ik} \leq s_{ij} & \forall j \in \Omega_i \\ \alpha_i - \gamma_{ij} + \sum_{k \in \Omega_i} \delta_{ik} \leq s_{ij} & \forall j \in \tilde{V} \setminus (\Omega_i \cup \{i\}) \\ \alpha_i \in \mathbb{R} \\ \gamma_{ij} \geq 0 & \forall j \in V \setminus \{i\} \\ \delta_{ij} \geq 0 & \forall j \in \Omega_i \end{cases}$$

2.5.4 Computing optimal primal and dual solutions to the subproblems

For all $i \in V$, we propose a feasible solution to LP_i^+ and a feasible solution to DLP_i^+ and show that they have the same objective value in the primal and in the dual respectively, for providing the proof of their optimality. This common numerical value is the cost incurred by the connection of node i to the ring. The study distinguishes 5 cases.

Case 1: Node i is a hub

For all node $i \in V$ such that $\hat{y}_{ii} = 1$, the unique feasible solution to LP_i^+ is to set y_{ij} to zero for all $j \in V \setminus \{i\}$, leading to a zero objective value for the primal. Similarly, a feasible solution to DLP_i^+ is to set α_i and γ_{ij} to zero for all $j \in V \setminus \{i\}$, leading to a zero objective value for the dual. But we can propose an alternative solution, that can be shown to yield stronger optimality cuts. This solution consists in setting $\alpha_i = \min \left(\min_{j \in V \setminus (\tilde{V} \cup \{i\})} \frac{1}{2}s_{ij}, \min_{j \in \tilde{V} \setminus \{i\}} s_{ij} \right)$, with $\gamma_{ij} = 0$ for all $j \in V \setminus \{i\}$. This solution is also feasible for DLP_i^+ and leads to a zero objective value as well. Now we show

that it produces a stronger optimality cut than the one we get by setting all the dual variables to zero. The optimality cut that we have when dual variables are set to zero for all hubs is

$$\lambda \geq \sum_{i \in V, \hat{y}_{ii}=0} 2(1 - y_{ii})\alpha_i + \sum_{(i,j,k) \in \tilde{J}} (x_{ij} + x_{jk} - 1)\beta_{ijk} - \sum_{(i,j) \in V^\neq} y_{jj}\gamma_{ij}$$

Now, if we set $\alpha_i = \min \left(\min_{j \in V \setminus (\tilde{V} \cup \{i\})} \frac{1}{2}s_{ij}, \min_{j \in \tilde{V} \setminus \{i\}} s_{ij} \right)$ for all $i \in V$ such that $\hat{y}_{ii} = 1$, we have the following optimality cut:

$$\lambda \geq \sum_{i \in V, \hat{y}_{ii}=1} (1 - y_{ii})\alpha_i + \sum_{i \in V, \hat{y}_{ii}=0} 2(1 - y_{ii})\alpha_i + \sum_{(i,j,k) \in \tilde{J}} (x_{ij} + x_{jk} - 1)\beta_{ijk} - \sum_{(i,j) \in V^\neq} y_{jj}\gamma_{ij}$$

Since $\sum_{i \in V, \hat{y}_{ii}=1} (1 - y_{ii})\alpha_i \geq 0$, the new optimality cut dominates the original one.

Case 2: Node i is a terminal and $s_{im_i^*} \leq 2s_{im_i}$

For all node $i \in V$ such that $\hat{y}_{ii} = 0$ and $s_{im_i^*} \leq 2s_{im_i}$, the primal optimal solution consists in connecting i to the certain hub m_i^* at a cost of $s_{im_i^*}$, so $y_{im_i^*} = 1$ and $y_{ij} = 0$ for all $j \in V \setminus \{i, m_i^*\}$. A feasible solution to DLP_i^+ can be to set $\alpha_i = \frac{1}{2}s_{im_i^*}$, $\gamma_{ij} = \max(0, s_{im_i^*} - s_{ij})$ for all $j \in V \setminus (\tilde{V} \cup \{i\})$, $\gamma_{ij} = \max\left(0, \frac{1}{2}s_{im_i^*} - s_{ij}\right)$ for all $j \in \tilde{V} \setminus \{i\}$ and $\delta_{ij} = 0$ for all $j \in \Omega_i$. Since $s_{im_i^*} \leq 2s_{im_i}$, $\gamma_{ij} = 0$ for all $j \in V$ such that $\hat{y}_{jj} = 1$, so the corresponding objective value of the dual is $s_{im_i^*}$.

Case 3: Node i is a terminal and $2s_{im_i} < s_{im_i^*} \leq s_{im_i} + s_{im'_i}$

First, it should be observed that if $|\Omega_i| \geq 2$, then $s_{im_i} = s_{im'_i}$, which contradicts the hypothesis of the present case. Hence we now assume that $|\Omega_i| = 1$.

For all node $i \in V$ such that $\hat{y}_{ii} = 0$ and $2s_{im_i} < s_{im_i^*} \leq s_{im_i} + s_{im'_i}$, the primal optimal solution consists in connecting i to the certain hub m_i^* , at a cost of $s_{im_i^*}$, so $y_{im_i^*} = 1$ and $y_{ij} = 0$ for all $j \in V \setminus \{i, m_i^*\}$. A feasible solution to DLP_i^+ can be to set $\alpha_i = \frac{1}{2}s_{im_i^*}$, $\gamma_{ij} = \max(0, s_{im_i^*} - s_{ij})$ for all $j \in V \setminus (\tilde{V} \cup \{i\})$.

Now, we consider $j \in \tilde{V} \setminus \{i\}$. Since $|\Omega_i| = 1$, we have $s_{im_i} < s_{im'_i}$ and the second constraint of DLP_i^+ reads $\alpha_i - \gamma_{im_i} - \delta_{im_i} \leq s_{im_i}$. Since $\hat{y}_{m_i m_i} = 1$ by definition, we shall have $\gamma_{im_i} = 0$ to let the objective function be $s_{im_i^*}$. Hence, the only option left to satisfy the constraint is to set $\delta_{im_i} = \frac{1}{2}s_{im_i^*} - s_{im_i}$, which is strictly positive. Now, the third constraint of DLP_i^+ reads: $\alpha_i - \gamma_{ij} + \delta_{im_i} \leq s_{ij}$ for all $j \in \tilde{V} \setminus \{i, m_i\}$. Replacing α_i and δ_{im_i} by their values yields: $s_{im_i^*} - s_{im_i} - \gamma_{ij} \leq s_{ij}$ for all $j \in \tilde{V} \setminus \{i, m_i\}$. We know that $s_{ij} \geq s_{im'_i}$ for all $j \in \tilde{V} \setminus \{i, m_i\}$ such that

$\hat{y}_{jj} = 1$, so setting $\gamma_{ij} = \max(0, s_{im_i^*} - s_{im_i} - s_{ij})$ for all $j \in \tilde{V} \setminus \{i, m_i\}$ ensures that the dual solution is feasible since we can show that $\gamma_{ij} = 0$ for all $j \in \tilde{V} \setminus \{i, m_i\}$ such that $\hat{y}_{jj} = 1$.

Case 4: Node i is a terminal and $2s_{im_i'} \leq s_{im_i^*}$

For all node $i \in V$ such that $\hat{y}_{ii} = 0$ and $2s_{im_i'} \leq s_{im_i^*}$, an optimal solution to LP_i^+ consists in connecting i to the uncertain hubs m_i and m_i' , at a cost of $s_{im_i} + s_{im_i'}$, so $y_{im_i} = y_{im_i'} = 1$ and $y_{ij} = 0$ for all $j \in V \setminus \{i, m_i, m_i'\}$. A feasible solution to DLP_i^+ can be to set $\alpha_i = \frac{1}{2}(s_{im_i} + s_{im_i'})$, $\gamma_{ij} = \max(0, s_{im_i} + s_{im_i'} - s_{ij})$ for all $j \in V \setminus (\tilde{V} \cup \{i\})$, $\gamma_{ij} = 0$ and $\delta_{ij} = \max\left(0, \frac{1}{2}(s_{im_i} + s_{im_i'}) - s_{ij}\right)$ for all $j \in \Omega_i$ and $\gamma_{ij} = \max\left(0, \frac{1}{2}(s_{im_i} + s_{im_i'}) - s_{ij}\right)$ for all $j \in \tilde{V} \setminus (\Omega_i \cup \{i\})$. Since $s_{im_i^*} \geq 2s_{im_i'} \geq s_{im_i} + s_{im_i'}$, $\gamma_{ij} = 0$ for all $j \in V$ such that $\hat{y}_{jj} = 1$, so the corresponding objective value of the dual is $s_{im_i} + s_{im_i'}$.

Case 5: Node i is a terminal and $s_{im_i} + s_{im_i'} < s_{im_i^*} < 2s_{im_i'}$

First, it should be observed that if $|\Omega_i| \geq 2$, then $s_{im_i} = s_{im_i'}$, which contradicts the hypothesis of the present case. Hence we now assume that $|\Omega_i| = 1$.

For all node $i \in V$ such that $\hat{y}_{ii} = 0$ and $s_{im_i} + s_{im_i'} < s_{im_i^*} < 2s_{im_i'}$, an optimal solution to LP_i^+ consists in connecting i to the uncertain hubs m_i and m_i' , at a cost of $s_{im_i} + s_{im_i'}$, so $y_{im_i} = y_{im_i'} = 1$ and $y_{ij} = 0$ for all $j \in V \setminus \{i, m_i, m_i'\}$. A feasible solution to DLP_i^+ can be to set $\alpha_i = \frac{1}{2}(s_{im_i} + s_{im_i'})$ and $\gamma_{ij} = \max(0, s_{im_i} + s_{im_i'} - s_{ij})$ for all $j \in V \setminus (\tilde{V} \cup \{i\})$. We can easily check that for all $j \in V \setminus (\tilde{V} \cup \{i\})$ such that $\hat{y}_{jj} = 1$, γ_{ij} is zero.

Now, we consider $j \in \tilde{V} \setminus \{i\}$. Since $|\Omega_i| = 1$, we have $s_{im_i} < s_{im_i'}$ and the second constraint of DLP_i^+ reads $\alpha_i - \gamma_{im_i} - \delta_{im_i} \leq s_{im_i}$. Since $\hat{y}_{m_i m_i} = 1$ by definition, we shall have $\gamma_{im_i} = 0$ to let the objective function be $s_{im_i} + s_{im_i'}$. Hence, the only option left to satisfy the constraint is to set $\delta_{im_i} = \frac{1}{2}(s_{im_i'} - s_{im_i})$, which is strictly positive. Now, the third constraint of DLP_i^+ reads: $\alpha_i - \gamma_{ij} + \delta_{im_i} \leq s_{ij}$ for all $j \in \tilde{V} \setminus \{i, m_i\}$. Replacing α_i and δ_{im_i} by their values yields: $s_{im_i'} - \gamma_{ij} \leq s_{ij}$ for all $j \in \tilde{V} \setminus \{i, m_i\}$. We know that $s_{ij} \geq s_{im_i'}$ for all $j \in \tilde{V} \setminus \{i, m_i\}$ such that $\hat{y}_{jj} = 1$, so setting $\gamma_{ij} = \max(0, s_{im_i'} - s_{ij})$ for all $j \in \tilde{V} \setminus \{i, m_i\}$ ensures that the dual solution is has an objective of $s_{im_i} + s_{im_i'}$, since we can show that $\gamma_{ij} = 0$ for all $j \in \tilde{V} \setminus \{i, m_i\}$ such that $\hat{y}_{jj} = 1$.

2.5.5 Polynomial-time algorithm to the subproblem's dual

Even if the subproblem can be stated as a linear program, it has a cubic number of β_{ijk} variables, which makes it long to solve. In order to solve it faster, we take

advantage of the fact that these variables are independent of the other variables. Algorithm 3 solves the dual of Benders subproblem's relaxation. It takes advantage of the fact that at most $|\tilde{V}|$ of the β_{ijk} are non-zeros and compute them in a quadratic running time. Knowing \hat{x}_{ij} and \hat{y}_{jj} , we initially let β'_j be the adjacency list of node $j \in \tilde{V}$ such that $\hat{y}_{jj} = 1$: $\beta'_j[1]$ and $\beta'_j[2]$ are its two neighbors in the ring. Note that for all $(i, j, k) \in \tilde{J}$, $\beta_{ijk} \neq 0$ implies that β'_j is nonempty (the converse does not hold when some ring costs are zero). Algorithm 3 returns an optimal solution to the subproblem's dual in $\mathcal{O}(|V|^2)$ time.

2.5.6 Optimality cuts of the Benders decomposition

The optimality cut that is added to the master problem of the Benders decomposition is:

$$\lambda \geq \sum_{i \in V} 2(1 - y_{ii})\alpha_i + \sum_{(i,j,k) \in \tilde{J}} (x_{ij} + x_{jk} - 1)\beta_{ijk} - \sum_{(i,j) \in A} y_{jj}\gamma_{ij}$$

When Algorithm 3 terminates, β_{ijk} is set to r_{ik} for all $(i, j, k) \in \tilde{J}$ and the optimality cut can be written as:

$$\lambda \geq \sum_{i \in V: \hat{y}_{ii}=0} 2(1 - y_{ii})\alpha_i - \sum_{(i,j) \in A: \hat{y}_{ii}=0} \gamma_{ij}y_{jj} + \sum_{j \in \tilde{V}: \beta'_j \neq \emptyset} (x_{\beta'_j[1]j} + x_{j\beta'_j[2]} - 1)r_{\beta'_j[1]\beta'_j[2]}$$

2.5.7 Polyhedral analysis of constraints (2.14)

In this section, we consider a terminal $i \in V$ and show that constraint (2.14) is facet-defining for IP_i when $H \cap (V \setminus \tilde{V})$ is nonempty and $H \cap \tilde{V}$ has cardinality at least three where $H = \{j \in V : \hat{y}_{jj} = 1\}$ is the set of hubs as per the solution of the master problem.

Constraint (2.10) is an equality, so it can be removed by substituting one variable of IP_i , say $y_{im_i^*}$. Hence

$$y_{im_i^*} = 1 - \sum_{\substack{j \in H \cap (V \setminus \tilde{V}) \\ j \neq m_i^*}} y_{ij} - \frac{1}{2} \sum_{j \in H \cap \tilde{V}} y_{ij}$$

As a result, the original nonnegativity constraint $y_{im_i^*} \geq 0$ becomes:

$$\sum_{\substack{j \in H \cap (V \setminus \tilde{V}) \\ j \neq m_i^*}} y_{ij} + \frac{1}{2} \sum_{j \in H \cap \tilde{V}} y_{ij} \leq 1$$

We can also remove all the decision variables $y_{ij} \leq \hat{y}_{jj}$ for which $\hat{y}_{jj} = 0$. This leads to a $|H| - 1$ decision variable formulation of IP_i :

Algorithm 3: Solving the subproblem's dual of 1-S-RSP

```

1 Input:  $(\hat{y}_{ii})_{i \in V}, (\hat{x}_{ij})_{ij \in E}$  booleans
2 Output:  $(\alpha_i)_{i \in V}, (\beta'_j)_{j \in \tilde{V}}, (\gamma_{ij})_{(i,j) \in A}, (\delta_{ij})_{i \in V: \hat{y}_{ii}=0, j \in \Omega_i}$ 
3 foreach  $j \in \tilde{V}$  do
4    $\beta'_j \leftarrow []$ 
5 foreach  $i \in V$  do
6    $\alpha_i \leftarrow 0$ 
7   foreach  $j \in V : j \neq i$  do
8     if  $j > i$  and  $\hat{x}_{ij} = 1$  and  $i \in \tilde{V}$  then
9        $\beta'_i \leftarrow \beta'_i \cup \{j\}$ 
10    if  $j > i$  and  $\hat{x}_{ij} = 1$  and  $j \in \tilde{V}$  then
11       $\beta'_j \leftarrow \beta'_j \cup \{i\}$ 
12     $\gamma_{ij} \leftarrow 0$ 
13 foreach  $i \in V : \hat{y}_{ii} = 0$  do
14    $\Omega_i \leftarrow \{j \in \tilde{V} \setminus \{i\} : \hat{y}_{jj} = 1, s_{ij} = s_{im_i}\}$ 
15   foreach  $j \in \Omega_i$  do
16      $\delta_{ij} \leftarrow 0$ 
17   if  $s_{im_i^*} \leq 2s_{im_i}$  then
18      $\alpha_i \leftarrow \frac{1}{2}s_{im_i^*}$ 
19     foreach  $j \in V \setminus (\tilde{V} \cup \{i\})$  do  $\gamma_{ij} \leftarrow \max(0, s_{im_i^*} - s_{ij})$ 
20     foreach  $j \in \tilde{V} \setminus \{i\}$  do  $\gamma_{ij} \leftarrow \max(0, \frac{1}{2}s_{im_i^*} - s_{ij})$ 
21   else if  $2s_{im_i} < s_{im_i^*}$  and  $s_{im_i^*} \leq s_{im_i} + s_{im_i'}$  then
22      $\alpha_i \leftarrow \frac{1}{2}s_{im_i^*}, \delta_{im_i} \leftarrow \frac{1}{2}s_{im_i^*} - s_{im_i}$ 
23     foreach  $j \in V \setminus (\tilde{V} \cup \{i\})$  do  $\gamma_{ij} \leftarrow \max(0, s_{im_i^*} - s_{ij})$ 
24     foreach  $j \in \tilde{V} \setminus (\Omega_i \cup \{i\})$  do  $\gamma_{ij} \leftarrow \max(0, s_{im_i^*} - s_{im_i} - s_{ij})$ 
25   else if  $2s_{im_i'} \leq s_{im_i^*}$  then
26      $\alpha_i \leftarrow \frac{1}{2}(s_{im_i} + s_{im_i'})$ 
27     foreach  $j \in V \setminus (\tilde{V} \cup \{i\})$  do  $\gamma_{ij} \leftarrow \max(0, s_{im_i} + s_{im_i'} - s_{ij})$ 
28     foreach  $j \in \Omega_i$  do  $\gamma_{ij} \leftarrow 0, \delta_{ij} \leftarrow \max(0, \frac{1}{2}(s_{im_i} + s_{im_i'}) - s_{ij})$ 
29     foreach  $j \in \tilde{V} \setminus (\Omega_i \cup \{i\})$  do  $\gamma_{ij} \leftarrow \max(0, \frac{1}{2}(s_{im_i} + s_{im_i'}) - s_{ij})$ 
30   else if  $s_{im_i} + s_{im_i'} < s_{im_i^*}$  and  $s_{im_i^*} < 2s_{im_i'}$  then
31      $\alpha_i \leftarrow \frac{1}{2}(s_{im_i} + s_{im_i'}), \delta_{im_i} \leftarrow \frac{1}{2}(s_{im_i'} - s_{im_i})$ 
32     foreach  $j \in V \setminus (\tilde{V} \cup \{i\})$  do  $\gamma_{ij} \leftarrow \max(0, s_{im_i} + s_{im_i'} - s_{ij})$ 
33     foreach  $j \in \tilde{V} \setminus (\Omega_i \cup \{i\})$  do  $\gamma_{ij} \leftarrow \max(0, s_{im_i'} - s_{ij})$ 
34 return  $(\alpha_i)_{i \in V}, (\beta'_j)_{j \in \tilde{V}}, (\gamma_{ij})_{(i,j) \in A}, (\delta_{ij})_{i \in V: \hat{y}_{ii}=0, j \in \Omega_i}$ 

```

$$(\text{IP}_i) : \left\{ \begin{array}{ll}
\text{Min } s_{im_i^*} + \sum_{j \in (H \setminus \{m_i^*\}) \cap (V \setminus \tilde{V})} (s_{ij} - s_{im_i^*}) y_{ij} + \sum_{j \in H \cap \tilde{V}} (s_{ij} - \frac{1}{2} s_{im_i^*}) y_{ij} & \\
y_{ij} \leq 1 & \forall j \in H \setminus \{m_i^*\} \\
\sum_{\substack{j \in H \cap (V \setminus \tilde{V}) \\ j \neq m_i^*}} y_{ij} + \frac{1}{2} \sum_{j \in H \cap \tilde{V}} y_{ij} \leq 1 & \\
y_{ik} \leq \sum_{j \in (H \cap \tilde{V}) \setminus \{k\}} y_{ij} & \forall k \in \Omega_i \\
y_{ij} \in \mathbb{B} & \forall j \in H \setminus \{m_i^*\}
\end{array} \right.$$

Determining the dimension of $\text{conv}(\text{IP}_i)$

We show that $\text{conv}(\text{IP}_i)$ has dimension $|H| - 1$. To do so, we build $|H|$ affinely independent points of $\mathbb{B}^{|H|-1}$ in $\text{conv}(\text{IP}_i)$ as follows:

- Point $P^{m_i^*}$ has all its entries equal to zero.
- For all $j \in (H \setminus \{m_i^*\}) \cap (V \setminus \tilde{V})$, point P^j is defined by $P_j^j = 1$, other entries are zero.
- For all $j \in H \cap (\tilde{V} \setminus \{m_i\})$, point P^j is defined by $P_j^j = 1$ and $P_{m_i}^j = 1$. Other entries are zero, so P^j has two nonzero entries.
- Point P^{m_i} is defined by $P_{j_1}^{m_i} = P_{j_2}^{m_i} = 1$ for some $j_1 \neq j_2$ in $H \cap (\tilde{V} \setminus \{m_i\})$, other entries are zero. Note that j_1 and j_2 can always be found since $H \cap \tilde{V}$ has cardinality at least three.

We form M , the $(|H| - 1)$ -by- $|H|$ matrix using P^j as its columns for all $j \in H$. Assuming that the nodes in the set $H \cap (V \setminus \tilde{V})$ are reindexed in $\{1, \dots, h_1\}$ (where m_i^* has index 1) and the nodes in $H \cap \tilde{V}$ are reindexed in $\{h_1 + 1, \dots, |H|\}$, (where m_i has index $|H|$), we can represent M as:

$$M = \begin{pmatrix} 0 & 1 & \cdots & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & & \vdots & \vdots \\ \vdots & 0 & \cdots & 1 & 0 & \cdots & 0 & 0 \\ \vdots & 0 & \cdots & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & 0 & \ddots & 1 & 1 \\ 0 & 0 & \cdots & 0 & 1 & \cdots & 1 & 0 \end{pmatrix} \quad M' = \begin{pmatrix} 1 & \cdots & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots & & \vdots & \vdots \\ 0 & \cdots & 1 & 0 & \cdots & 0 & 0 \\ 0 & \cdots & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & 0 & \ddots & 1 & 1 \\ 0 & \cdots & 0 & 1 & \cdots & 1 & 0 \end{pmatrix}$$

To show that the $|H|$ points that constitute the columns of M are affinely independent, we subtract the first column of M to all the other ones in order to create

$|H| - 1$ directions, then the first column is removed. This yields M' , a block diagonal $(|H| - 1)$ -by- $(|H| - 1)$ matrix that can be written as $\text{diag}(A, B)$. The first diagonal block, denoted by A , is the identity matrix of order $h_1 = |(H \setminus \{m_i^*\}) \cap (V \setminus \tilde{V})|$ and B , which is a square matrix of order $|H \cap \tilde{V}|$, can itself be split into diagonal blocks. Block $B_{1,1}$ is the identity matrix of order $|(H \setminus \{m_i\}) \cap \tilde{V}|$, block $B_{1,2}$ is an $|(H \setminus \{m_i\}) \cap \tilde{V}|$ -element column vector full of zeros, except for rows j_1 and j_2 where it has one entries, j_1 and j_2 are two distinct elements of $(H \setminus \{m_i\}) \cap \tilde{V}$. Block $B_{2,1}$ is an $|(H \setminus \{m_i\}) \cap \tilde{V}|$ -element row vector full of ones and block $B_{2,2}$ is a one-by-one matrix that contains zero. It will be shown that M' is non-singular by expressing its determinant, as $\det(A) \times \det(B) = \det(B)$. It is first shown that B is non-singular: Since $B_{1,1}$ is the identity matrix of order $|(H \setminus \{m_i\}) \cap \tilde{V}|$, the rank of B is at least $|(H \setminus \{m_i\}) \cap \tilde{V}|$. Now, if the last column of B could be written as a linear combination of the first $|(H \setminus \{m_i\}) \cap \tilde{V}|$ columns of B , we could express this column as $\sum_{j \in (H \setminus \{m_i\}) \cap \tilde{V}} \mu_j B_j = B_{m_i}$ where B_j denotes the j -th column of B and μ_j is a real number. This would imply $\mu_j = 0$ for all $j \in (H \setminus \{m_i, j_1, j_2\}) \cap \tilde{V}$, $\mu_{j_1} = \mu_{j_2} = 1$ and $\mu_{j_1} + \mu_{j_2} = 0$ because of the last row of B . This system has no solution, which implies that all the columns of B are linearly independent. As a result, $\det(B)$ is nonzero and M' is non-singular. Consequently, $\text{conv}(\text{IP}_i)$ has dimension $|H| - 1$.

Showing that constraint (2.14) is facet-defining

We show that for all $k \in \Omega_i$, the inequality induced by constraint (2.14) is facet-defining:

$$y_{ik} \leq \sum_{j \in (H \cap \tilde{V}) \setminus \{k\}} y_{ij}$$

To this end, we find $|H| - 1$ points in $\text{conv}(\text{IP}_i)$ as we did for showing that $\text{conv}(\text{IP}_i)$ has dimension $|H| - 1$, but we use k instead of m_i and we do not take point P^k as it does not satisfy constraint (2.14) to equality. It can be checked that constraint (2.14) is satisfied to equality for all the points P^j with $j \in H \cap (V \setminus \tilde{V})$ as $0 = 0$ and for all P^j with $j \in H \cap (\tilde{V} \setminus \{m_i\})$ we have $1 = 1$.

We form M_k , the $(|H| - 1)$ -by- $(|H| - 1)$ matrix using P^j as its columns for all $j \in H \setminus \{k\}$. Assuming that the nodes in the set $H \cap (V \setminus \tilde{V})$ are reindexed in $\{1, \dots, h_1\}$ (where m_i^* has index 1) and the nodes in $H \cap \tilde{V}$ are reindexed in $\{h_1 + 1, \dots, |H|\}$, (where k has index $|H|$). Then, we form M'_k by subtracting the first column to all the other columns of M_k and we remove the first column and the last row. This yields the identity matrix of order $|H| - 2$, hence we have found $|H| - 1$ affinely independent points of $\text{conv}(\text{IP}_i)$ that satisfy (2.14) to equality. This

proves that (2.14) is facet-defining.

$$M_k = \begin{pmatrix} 0 & 1 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & & \vdots \\ \vdots & 0 & \cdots & 1 & 0 & \cdots & 0 \\ \vdots & 0 & \cdots & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & 0 & \ddots & 1 \\ 0 & 0 & \cdots & 0 & 1 & \cdots & 1 \end{pmatrix} \quad M'_k = \begin{pmatrix} 1 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & & \vdots \\ 0 & \cdots & 1 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 1 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 1 \end{pmatrix}$$

2.5.8 An instance transformation for 1-S-RSP

In this section, we propose an instance transformation to modify the expression of the objective function of 1-S-RSP so that the Benders decomposition algorithm performs better. Indeed, when star costs are high, they are “invisible” to the master problem and many optimality cuts are required for improving the lower bound of the master problem which discovers these costs gradually through the introduction of optimality cuts. The proposed transformation aims at transferring as much cost as possible from the subproblem to the master problem in order to improve the lower bound of the latter. The idea is then to minimize the contribution of the subproblem to the original objective function value.

Transformation for the star costs

For all $i \in V$, let c_i be the contribution of this node to the solution cost. It is defined as:

$$c_i = o_i y_{ii} + \sum_{j \in V \setminus (\bar{V} \cup \{i\})} s_{ij} y_{ij} + \sum_{j \in \bar{V} \setminus \{i\}} s_{ij} y_{ij}$$

We can observe that exactly one of the following three situations happens in any optimal solution to 1-S-RSP:

1. Node i is selected as a hub, so $y_{ii} = 1$ and $y_{ij} = 0$ for all $j \in V \setminus \{i\}$, which yields $c_i = o_i$
2. Node i is a terminal connected to a certain hub j , so $y_{ij} = 1$ and $y_{ik} = 0$ for all $k \in V \setminus \{j\}$, which yields $c_i = s_{ij}$
3. Node i is a terminal connected to two distinct uncertain hubs j and j' , so $y_{ij} = y_{ij'} = 1$ and $y_{ik} = 0$ for all $k \in V \setminus \{j, j'\}$, which yields $c_i = s_{ij} + s_{ij'}$

For all $i \in V$, let \overline{offset}_i be a lower bound on the cost incurred by node i if it is a terminal:

$$\overline{offset}_i = \min \left(\min_{k \in V \setminus (\tilde{V} \cup \{i\})} s_{ik}, 2 \min_{k \in \tilde{V} \setminus \{i\}} s_{ik} \right)$$

This lower bound returns the minimum over two quantities. The first one is the minimum cost for connecting i to a certain hub, and the second one is the minimum cost for connecting i to two uncertain hubs. We can now express c_i as follows:

$$c_i = \overline{offset}_i + (o_i - \overline{offset}_i) y_{ii} + \sum_{j \in V \setminus (\tilde{V} \cup \{i\})} (s_{ij} - \overline{offset}_i) y_{ij} + \sum_{j \in \tilde{V} \setminus \{i\}} \left(s_{ij} - \frac{1}{2} \overline{offset}_i \right) y_{ij}$$

The numerical value of c_i can be checked to be correct in the above-mentioned three cases, and it can be observed that the y_{ij} variables are all multiplied by a nonnegative constant. The objective function of the ILP formulation of 1-S-RSP can be updated accordingly.

Transformation of the contribution of backup edges

The presence of backup edges in any optimal solution to 1-S-RSP is the consequence of the selection of hubs in \tilde{V} . So if edge ij is selected as a ring edge (*i.e.*, $x_{ij} = 1$), with $i \in \tilde{V}$, then the selection of this edge implies that a backup edge, incident to j should be present to cope the failure of node i . The cost of this edge is at least $\min_{k \in V \setminus \{j\}} r'_{kj}$ and since this backup edge is also connected to another hub, we can raise the cost of edge (i, j) to $r_{ij} + \frac{1}{2} \epsilon_j$ where we define $\epsilon_j = \min_{k \in V \setminus \{j\}} r'_{kj}$ for all $j \in V$, as the minimum cost of a backup edge incident to j . Consequently, the cost of any backup edge (i, j) must be decreased by $\frac{1}{2}(\epsilon_i + \epsilon_j)$, which maintains nonnegativity.

$$\left\{ \begin{array}{ll} \overline{offset} \leftarrow \sum_{i \in V} \overline{offset}_i & \\ \overline{o}_i \leftarrow o_i - \overline{offset}_i & \forall i \in V \\ \overline{r}_{ij} \leftarrow r_{ij} & \forall i \in V \setminus \tilde{V}, \forall j \in V \setminus \tilde{V} : i < j \\ \overline{r}_{ij} \leftarrow r_{ij} + \frac{1}{2} \epsilon_j & \forall i \in \tilde{V}, \forall j \in V \setminus \tilde{V} : i < j \\ \overline{r}_{ij} \leftarrow r_{ij} + \frac{1}{2} \epsilon_i & \forall i \in V \setminus \tilde{V}, \forall j \in \tilde{V} : i < j \\ \overline{r}_{ij} \leftarrow r_{ij} + \frac{1}{2} \epsilon_i + \frac{1}{2} \epsilon_j & \forall i \in \tilde{V}, \forall j \in \tilde{V} : i < j \\ \overline{r}'_{ij} \leftarrow r'_{ij} - \frac{1}{2} \epsilon_i - \frac{1}{2} \epsilon_j & \forall i \in V, \forall j \in V : i < j \\ \overline{s}_{ij} \leftarrow s_{ij} - \overline{offset}_i & \forall i \in V, \forall j \in V \setminus (\tilde{V} \cup \{i\}) \\ \overline{s}_{ij} \leftarrow s_{ij} - \frac{1}{2} \overline{offset}_i & \forall i \in V, \forall j \in \tilde{V} \setminus \{i\} \end{array} \right.$$

Consequently the objective function of 1-S-RSP is now:

$$\text{Minimize } \overline{offset} + \sum_{i \in V \setminus \tilde{V}} \overline{o}_i y_{ii} + \sum_{i \in V} \sum_{\substack{j \in V \\ i < j}} (\overline{r}_{ij} x_{ij} + \overline{r}'_{ij} x'_{ij}) + \sum_{i \in V} \sum_{j \in V \setminus \{i\}} \overline{s}_{ij} y_{ij}$$

The advantage of this instance transformation is that the objective value of the master problem gets larger, while the objective value of the subproblem gets smaller (the sum remains the same). The transformation also guarantees that ring costs and star costs remain nonnegative, which is necessary for solving the subproblem, as stated in Subsection 2.5.2. By contrast, transformed hub selection costs \bar{d}_i are unrestricted in sign.

2.6 Enhancing and solving the ILP and the Branch-and-Benders-cut of 1-S-RSP

2.6.1 Introduction of 2-opt backup

In order to improve a 1-S-RSP feasible solution, we introduce a strategy, called 2-opt backup, an extension of the classical 2-opt algorithm proposed by G.A. Croes [16]. 2-opt backup is applied to both ILP and Branch-and-Benders-cut decomposition. It consists of improving the ring of the current solution by the mean of 2-opt backup moves. These moves differ from classical 2-opt moves as they also consider backup edges. Algorithm 4 describes the 2-opt backup procedure which takes as input argument the ring and backup edges x and x' as well as their associated costs r and r' , the array of hubs H that appear in the same order as in the ring, the cost of the input solution stored in *previous_cost* and \tilde{V} . We use H_i for $i \in \{1, \dots, |H|\}$ to denote the i -th element of H , that is $H[i]$. Figure 2.5 illustrate a 2-opt backup move where the old backup edge $x'_{i-1, i+1}$ is shown and replaced by $x'_{i-1, j}$, and a part of the ring is replaced from $[i-1, i, i+1, i+2, \dots, j-1, j, j+1, j+2]$ to $[i-1, i, j, j-1, \dots, i+2, i+1, j+1, j+2]$ exactly as in Algorithm 4.

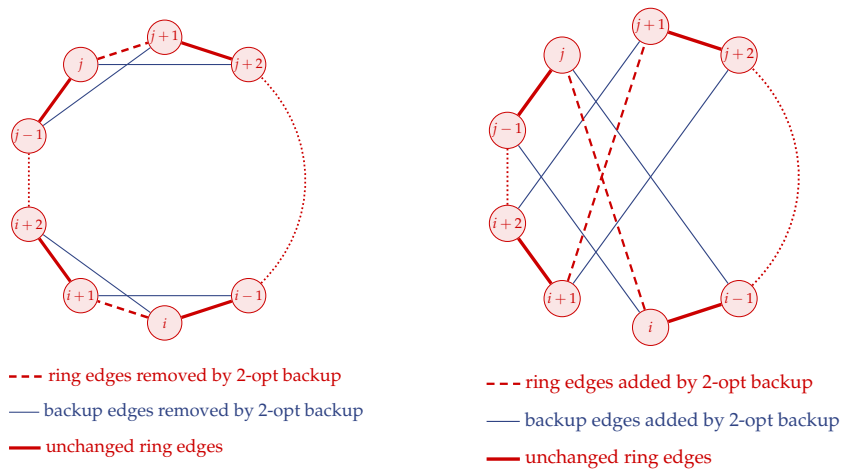


Figure 2.5: Illustration of edges before (left) and after (right) a 2-opt backup move

2.6.2 Description of the ILP and Branch-and-Benders-cut algorithms

The ILP model is solved without constraints (2.2). Subtour elimination constraints are separated at fractional nodes using the Edmonds-Karp Min cut algorithm [42] and are added to the model thanks to user cuts callbacks. We stop separating such constraints as soon as 2000 inequalities have been generated. This limitation does not apply to the inequalities that are generated at integer nodes of the solver's branch-and-bound tree under the form of lazy constraints, like the subtour elimination constraints found at integer nodes, and the Benders optimality cuts. Every time an integer solution is found, we check whether it contains a subtour or not. If it contains a subtour, we add the corresponding violated constraint (2.2), otherwise we try to improve the ring with 2-opt backup from Section 2.6.1. If the improved solution is better than the incumbent, then the incumbent is updated accordingly.

The Branch-and-Benders-cut algorithm is implemented as follows. The instance transformation introduced in Subsection 2.5.8 is first applied. Similarly to ILP, constraints (2.8) and (2.9) are not initially present in the model, and we separate them at fractional nodes of the search tree as in the ILP, using the same limit of 2000 user cuts callbacks. Every time the Branch-and-Benders-cut algorithm finds an integer solution, we check whether it contains a subtour or not. We add a subtour elimination constraint implemented as a lazy constraint if it does have a subtour, otherwise, we call 2-opt backup. If the improved solution outperforms the integer solution of the current node in the search tree, then it replaces it. We then solve the subproblem and submit an optimality cut as in Subsection 2.5.6.

2.7 Numerical Experiments

The ILP introduced in Section 2.4 and the Branch-and-Benders-cut algorithm presented in Section 2.5, are implemented with the improvements described in Section 2.6.2, and tested on two classes of instances. Class I is a collection of $32 \times 3 = 96$ TSP instances taken from TSPLIB [50] involving between 51 and 200 nodes. In these instances, the nodes are described by their coordinates. Let l_{ij} be the Euclidean distance between nodes i and j of V . Following [41], we define the ring cost by $r_{ij} = \lceil \alpha l_{ij} \rceil$ and the star cost by $s_{ij} = \lceil (10 - \alpha) l_{ij} \rceil$ where α is in the set $\{3, 5, 7\}$, for all (i, j) in V^2 . The influence of parameter α on solutions structure is shown on Figure 2.6 which originates from Calvete [13] *et al.* (2013). Opening costs are zero for these instances, that have been used in [32] and in [41] to compare solution approaches for RSP.

Algorithm 4: A 2-opt backup algorithm to improve a feasible solution of 1-S-RSP

```

1 Input:  $(x_{ij})_{ij \in E}, (x'_{ij})_{ij \in E}, H, \text{previous\_cost}, \tilde{V}, (r_{ij})_{ij \in E}, (r'_{ij})_{ij \in E}$ 
2 Output:  $(x_{ij})_{ij \in E}$ : Improved ring,  $(x'_{ij})_{ij \in E}$ : Corresponding improved backup edges,
    $\text{new\_cost}$ 
3  $\text{improving} \leftarrow \text{true}$ 
4  $\text{new\_cost} \leftarrow \text{previous\_cost}$ 
5 while  $\text{improving}$  do
6    $\text{improving} \leftarrow \text{false}$ 
7   foreach  $h_i \in H$  do
8     foreach  $h_j \in H$  and  $j \notin \{i-1, i, i+1, i+2\}$  do
9        $\text{saving} \leftarrow r_{H_i, H_{i+1}} + r_{H_j, H_{j+1}} - r_{H_i, H_j} - r_{H_{i+1}, H_{j+1}}$ 
10      if  $H_i \in \tilde{V}$  then
11         $\text{saving} \leftarrow \text{saving} + r'_{H_{i-1}, H_{i+1}} - r'_{H_{i-1}, H_j}$ 
12      if  $H_j \in \tilde{V}$  then
13         $\text{saving} \leftarrow \text{saving} + r'_{H_{j-1}, H_{j+1}} - r'_{H_i, H_{j-1}}$ 
14      if  $H_{i+1} \in \tilde{V}$  then
15         $\text{saving} \leftarrow \text{saving} + r'_{H_i, H_{i+2}} - r'_{H_{i+2}, H_{j+1}}$ 
16      if  $H_{j+1} \in \tilde{V}$  then
17         $\text{saving} \leftarrow \text{saving} + r'_{H_j, H_{j+2}} - r'_{H_{i+1}, H_{j+2}}$ 
18      if  $\text{saving} > 0$  then
19         $\text{new\_cost} \leftarrow \text{new\_cost} - \text{saving}$ 
20        Replace edges  $x_{i,i+1}$  and  $x_{j,j+1}$  by  $x_{i,j}$  and  $x_{i+1,j+1}$  in  $H$ 
21        Set the following backup edges variables to zero:  $x'_{i-1,i+1}, x'_{i,i+2},$ 
            $x'_{j-1,j+1}$  and  $x'_{j,j+2}$ 
22        if  $H_i \in \tilde{V}$  then
23          Set the backup edge variable  $x'_{i-1,j}$  to one
24        if  $H_j \in \tilde{V}$  then
25          Set the backup edge variable  $x'_{i,j-1}$  to one
26        if  $H_{i+1} \in \tilde{V}$  then
27          Set the backup edge variable  $x'_{i+2,j+1}$  to one
28        if  $H_{j+1} \in \tilde{V}$  then
29          Set the backup edge variable  $x'_{i+1,j+2}$  to one
30        In the ring  $H$ , replace the hub sequence
            $[i, i+1, i+2, \dots, j-1, j, j+1]$  by the hub sequence
            $[i, j, j-1, \dots, i+2, i+1, j+1]$ 
31         $\text{improving} \leftarrow \text{true}$ 
32 return  $(x_{ij})_{ij \in E}, (x'_{ij})_{ij \in E}, \text{new\_cost}$ 

```

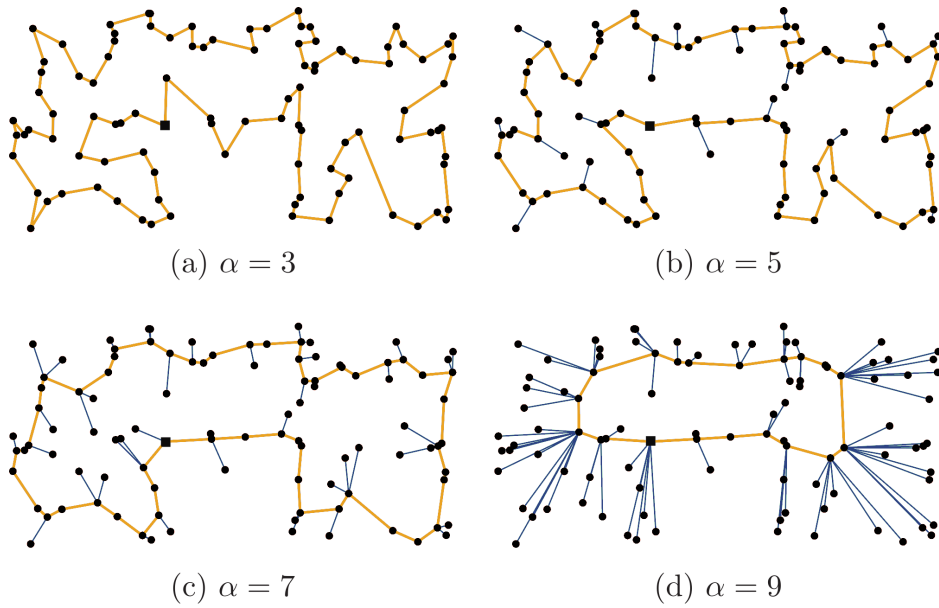


Figure 2.6: 4 optimal RSP solutions with the same instance except the ring and star costs for 4 values of input parameter α

Class II is a collection of $8 \times 5 = 40$ new, artificially generated instances where nodes have coordinates uniformly drawn from $\{0, \dots, 1000\}^2$. Ring and star costs are both computed as the Euclidean distance, $r_{ij} = s_{ij} = \lceil l_{ij} \rceil$ for all (i, j) in V^2 . For all i in V , o_i is uniformly drawn from $\{0, \dots, 1000\}$. The number of nodes n varies in $\{20, 30, 40, 60, 100, 125, 150, 200\}$. The depot is always chosen as the first node. For each n , we randomly generate 5 of those instances.

We use Julia v1.8.5, JuMP v1.10.0, and Gurobi v10.0.1 to solve all linear programs and integer linear programs. Gurobi is used with default settings, except for the number of threads which is set to four, and PreCrush which is set to one, to enable user cuts in the B&BC algorithm. We enforce a time limit of one hour per instance. The computer used to carry out the computational experiments is equipped with an Intel(R) Core i7-7700 processor running at 3.60 GHz and 16 GBytes of RAM, under GNU/Linux.

Table 2.2 presents an analysis of the impact of \tilde{V} based on Class II instances 30.1 and 60.1 where \tilde{V} takes 5 different sizes. All instances are solved with ILP. In particular when $\tilde{V} = \emptyset$, the problem reduces to RSP. The columns of this table are as follows: **CPU** is the CPU time in seconds to solve the ILP model. “TL” is indicated when the time limit of one hour is reached; **Gap** represents the relative optimality gap of ILP. Its formula is $\frac{UB - LB}{UB}$ where LB and UB are the ILP objective bound and incumbent objective value, both **LB** and **UB** are also displayed in Table 2.2. Column **n_constraints** shows the number of rows resulting from con-

straints (2.1) and (2.3)–(2.7) in the ILP model. It can be seen in this table that the problem difficulty increases drastically with the size of \tilde{V} , so it is clear that 1-S-RSP is much harder than the Ring Star Problem. Moreover, the number of uncertain nodes has a significant impact on the total number of constraints, since (2.5) has a cubic number of inequalities. We consider the most challenging setting, as a consequence, \tilde{V} is set to $V \setminus \{1\}$ for both Class I and Class II instances, meaning the depot is certain and all other nodes are uncertain.

Tables 2.3 and 2.4 present the results for Class I and Class II instances, respectively. The columns of these tables are as follows: **CPU BF is the CPU time spent by brute-force Algorithms 1 and 2, it is the same for the ILP and the Branch-and-Benders-cut models.** The brute-force algorithms are launched prior to the ILP and the Branch-and-Benders-cut models. **CPU** is the total CPU time in seconds, it includes the CPU time of column CPU BF. “TL” is indicated when the time limit of one hour is reached; **Gap** represents the relative optimality gap of both methods. Its formula is the same as in Table 2.2. Both **LB** and **UB** are also displayed in Tables 2.3 and 2.4; $|H|$ is the number of hubs in the best solution found. When $|H| = 4$, the best solution is the one returned by Algorithm 2. Note that since there is a single certain node in the considered instances, Algorithm 1 cannot find any ring with three hubs. **n_subtour** is the number of subtour elimination constraints, i.e., feasibility cuts for Benders, and lazy constraints for ILP. Columns **Nodes** show the total number of explored nodes in the solver’s branch-and-bound tree during the search. **CPU SP is the CPU time in seconds for solving the Benders subproblems. To obtain the running time of the master problem, we should subtract CPU BF and CPU SP to CPU.** **n_cut gives the number of optimality cuts in the Benders decomposition, which is also the number of times the subproblem is solved.**

Each line in Table 2.3 presents an instance with the format “Instance name $n - \alpha$ ” where n is the total number of nodes in the instance and α is the aforementioned parameter. For Table 2.4, the format is “Class II $n.ID$ ” where ID varies from 1 to 5 and corresponds to the 5 randomly generated instances.

It can be seen in Table 2.3 that for Class I instances, the ILP model runs out of memory for 17 instances when the number of nodes exceeds 150, whereas this happens only once for the instance d 198-7 with the Branch-and-Benders-cut algorithm. The upper bound returned by the Branch-and-Benders-cut algorithm is always less than or equal to the one returned by ILP. The gap is better for the Branch-and-Benders-cut algorithm on 63 instances out of 95, and the lower bound is better for the Branch-and-Benders-cut algorithm on 30 instances out of 95. When an algorithm runs out of memory, we consider the other one to be better for LB, UB, and the gap. No comparison can be made for the instance d 198-7 as both algorithms run out of memory. From these observations, we conclude

that Branch-and-Benders-cut tends to produce better quality solutions, while ILP tends to return better lower bounds (except for $\alpha = 3$). The Branch-and-Benders-cut algorithm has fewer memory issues than the ILP model on larger instances. As expected, the number of explored nodes in the solver's branch-and-bound tree is significantly less with the ILP due to its size. The Branch-and-Benders-cut algorithm has always a better gap than the ILP model when $\alpha = 3$, which suggests that it is more efficient when edge costs are relatively low. On the opposite, when $\alpha = 7$, ILP returns a better gap except when memory issues occur, and for instances pr 136-7 and pr 152-7. It can also be observed that the CPU time spent solving Benders subproblems increases with α at a much faster pace than the number of optimality cuts. Indeed, the computational effort required to solve the Benders subproblem is larger when the ring has a low number of hubs. This is because there are a lot of terminals to assign to the (small) ring and the running time of the algorithm that solves the subproblem increases quadratically with the number of terminals. Not surprisingly, we also observe that for $\alpha = 3$, the best available solutions have very few terminals, because of relatively large star costs.

Table 2.4 reports results on Class II instances, for which ILP finds an optimal solution to seven small instances. The average value of the integrality gap of ILP in these instances is 1.29 with a standard deviation of 0.05. This relatively large integrality gap suggests that the problem formulation may be improved, independently of the subtour elimination constraints, that do not need to be added in large number for these instances. On Class II instances, the Branch-and-Benders-cut model is much less efficient than ILP for small values of n , however, ILP shows memory difficulties for all instances with 150 nodes and more while the Branch-and-Benders-cut model succeeds in obtaining solutions. We observe that for small instances, the number of optimality cuts varies a lot from one instance to another even if they have the same characteristics, see for example Class II - 20. The same happens with the CPU time and the number of nodes in the branch-and-bound tree. For the remaining instances of Table 2.4, the number of optimality cuts seems homogeneous but this is just because the time limit is the same for all instances and they are not solved to optimality. So the number of optimality cuts that are reported corresponds to the number of cuts that have been generated in one hour, but the total number of cuts needed to obtain an optimal solution may vary significantly from one instance to the next for the same value of n .

In Table 2.5, six instances of Class I and six instances of Class II are solved without 2-opt backup, and the variation of the results with respect to those obtained with 2-opt backup in Tables 3.1 and 2.4 are displayed. When no solution improving over the one returned by the brute-force algorithms is found, 2-opt backup is not called and the results remain the same as in Tables 3.1 and 2.4. This happens

for eil 51-7, pr 124-3, pr 124-5, and pr 124-7. It can be seen that for ClassII - 20.1, the only instance solved to optimality, 2-opt backup is detrimental to the CPU time of ILP, but beneficial to the running time of the Branch-and-Benders-cut algorithm. The optimality gap of the ILP is generally increased when 2-opt backup is not used, and the effect is even more significant for Benders. When no time is consumed by 2-opt backup, the solver runs out of memory in less than one hour with instance ClassII - 125.1 while it does not have enough time to saturate all the memory when 2-opt backup is used. The effect of 2-opt backup on the lower bound seems to be slightly detrimental for large instances, but improves the upper bound very significantly, especially in the case of the Branch-and-Benders-cut algorithm.

Table 2.2: Impact of \tilde{V} 's cardinality

Instance n .ID	$n = V $	\tilde{V}	ILP				
			CPU	Gap	LB	UB	n_constraints
ClassII - 30.1		\emptyset (Ring Star Problem)	5.59	0%	6,912	6,912	871
		$\{2, \dots, 8\}$	5.44	0%	7,545	7,545	3,713
		$\{2, \dots, 15\}$	5.86	0%	8,335	8,335	6,555
		$\{2, \dots, 23\}$	5.10	0%	8,800	8,800	9,803
		$\{2, \dots, 30\} = V \setminus \{1\}$	TL	5.0%	13,535	14,247	12,645
ClassII - 60.1		\emptyset (Ring Star Problem)	5.92	0%	9,906	9,906	3,541
		$\{2, \dots, 15\}$	6.24	0%	9,920	9,920	27,495
		$\{2, \dots, 30\}$	7.63	0%	10,862	10,862	53,160
		$\{2, \dots, 45\}$	22.76	0%	13,980	13,980	78,825
		$\{2, \dots, 60\} = V \setminus \{1\}$	TL	19.8%	18,626	23,229	104,490

Concluding remarks on the present chapter are given in Chapter 4.

Table 2.3: TSPLIB instances Class I

Instance $n-\alpha$	ILP								Branch-and-Benders-cut								
	$n = V $	CPU BF	CPU	Gap	LB	UB	$ H $	n_{subtour}	Nodes	CPU	CPU SP	Gap	LB	UB	$ H $	n_{subtour}	n_{cut}
eil 51-3	0.31	TL	37.0%	2,257	3,588	50	121	78,603	TL	0.56	24.0%	2,457	3,236	50	5,176	2,423	468,554
eil 51-5	0.32	TL	32.4%	3,656	5,415	42	107	54,287	TL	16.42	28.9%	3,607	5,078	27	13,233	5,278	708,228
eil 51-7	0.37	TL	15.3%	3,532	4,171	4	16	32,403	TL	22.44	27.7%	3,013	4,171	4	10,705	5,552	945,219
berlin 52-3	0.34	TL	36.8%	39,849	63,065	52	167	63,312	TL	0.78	28.5%	41,225	57,728	52	6,364	2,709	423,356
berlin 52-5	0.41	TL	37.0%	62,860	99,903	4	287	67,975	TL	11.32	33.7%	59,510	89,875	25	11,946	3,989	618,813
berlin 52-7	0.32	TL	2.4%	63,689	65,307	4	20	41,342	TL	18.19	19.8%	52,348	65,307	4	11,064	4,686	853,647
brazil 58-3	0.50	TL	41.5%	133,661	228,852	57	68	37,606	TL	0.62	38.3%	125,103	202,818	58	3,969	2,510	431,328
brazil 58-5	0.53	TL	45.0%	204,560	372,410	52	102	27,887	TL	9.59	46.7%	171,404	321,835	36	14,922	2,547	606,453
brazil 58-7	0.56	TL	31.8%	199,565	292,931	4	29	36,218	TL	14.39	57.4%	124,501	292,931	4	18,845	2,951	951,187
st 70-3	1.30	TL	42.5%	3,462	6,030	67	59	15,626	TL	1.21	33.8%	3,475	5,252	70	4,138	1,833	413,721
st 70-5	1.73	TL	41.0%	5,573	9,453	55	51	8,315	TL	14.21	43.0%	4,903	8,616	47	12,134	2,450	620,976
st 70-7	1.21	TL	47.6%	5,475	10,455	4	25	21,826	TL	36.72	57.1%	3,909	9,126	25	15,980	4,069	650,581
eil 76-3	1.78	TL	42.4%	2,817	4,895	75	74	2,640	TL	0.33	24.1%	3,151	4,155	76	5,138	1,019	385,440
eil 76-5	1.81	TL	55.7%	4,522	10,215	4	39	3,603	TL	22.68	27.3%	4,542	6,253	43	12,202	2,387	628,758
eil 76-7	1.78	TL	37.9%	4,204	6,777	4	20	13,126	TL	50.41	45.0%	3,455	6,283	24	17,298	4,341	668,284
pr 76-3	1.74	TL	67.4%	570,777	1,754,414	76	127	4,449	TL	1.00	38.1%	533,226	862,198	74	5,225	1,180	315,232
pr 76-5	1.68	TL	39.0%	874,185	1,434,512	62	72	13,798	TL	14.18	46.4%	718,698	1,341,315	62	10,506	1,893	531,147
pr 76-7	1.75	TL	50.0%	876,968	1,755,899	4	35	26,342	TL	41.84	58.1%	598,681	1,429,548	29	15,609	3,398	621,739
gr 96-3	5.71	TL	67.3%	2,670	8,188	63	81	1,868	TL	0.81	31.6%	2,775	4,058	96	4,346	750	365,752
gr 96-5	5.56	TL	71.3%	4,235	14,773	4	34	2,336	TL	14.47	41.5%	3,828	6,547	70	11,525	1,086	523,031
gr 96-7	5.55	TL	57.1%	4,184	9,767	4	6	2,254	TL	40.12	59.0%	2,838	6,928	40	17,009	1,997	576,887
rat 99-3	6.44	TL	80.4%	6,525	33,320	98	77	1,614	TL	0.36	27.5%	7,033	9,707	99	3,495	737	305,698
rat 99-5	6.47	TL	70.9%	10,669	36,675	4	37	1,741	TL	15.50	36.9%	10,258	16,272	78	12,847	1,091	580,581
rat 99-7	6.45	TL	56.5%	10,568	24,319	4	8	4,407	TL	45.44	57.7%	7,746	18,322	32	17,895	1,942	561,885
kroA 100-3	7.03	TL	47.2%	106,098	201,237	94	56	2,070	TL	0.82	34.3%	109,571	166,981	100	4,021	577	369,882
kroA 100-5	6.96	TL	44.0%	172,987	308,973	85	64	1,672	TL	17.73	45.3%	149,994	274,401	73	11,337	1,152	586,367
kroA 100-7	6.84	TL	50.0%	181,029	362,424	4	10	1,852	TL	37.10	63.7%	114,866	316,689	37	17,722	1,544	533,958
kroB 100-3	6.87	TL	52.7%	113,430	239,810	95	59	1,750	TL	1.16	36.2%	111,470	174,824	99	4,753	673	322,061
kroB 100-5	6.70	TL	55.6%	181,735	409,546	90	46	1,877	TL	19.43	50.0%	150,203	300,691	62	13,596	1,145	617,033
kroB 100-7	6.76	TL	47.4%	185,176	352,463	4	19	1,847	TL	41.84	64.3%	112,557	315,676	25	18,412	1,884	643,089
kroC 100-3	6.81	TL	87.1%	108,214	840,469	4	42	1,835	TL	1.24	33.7%	107,916	162,886	99	4,294	819	294,617
kroC 100-5	7.00	TL	71.9%	175,217	625,252	4	33	1,914	TL	17.86	47.1%	148,107	280,119	71	11,595	1,153	578,464
kroC 100-7	6.74	TL	56.3%	177,666	407,063	4	10	1,948	TL	51.97	64.8%	112,115	319,311	36	16,428	2,159	661,483
kroD 100-3	6.79	TL	86.1%	110,184	797,532	4	32	2,057	TL	1.05	36.0%	108,531	169,665	99	4,215	667	375,163
kroD 100-5	6.74	TL	70.1%	178,337	596,592	4	39	2,115	TL	20.87	49.5%	148,546	294,483	76	13,683	1,204	562,854
kroD 100-7	6.78	TL	54.2%	179,828	393,163	4	6	1,903	TL	36.23	62.5%	118,856	317,381	33	14,114	1,573	486,118
kroE 100-3	6.81	TL	84.6%	115,327	752,307	4	39	1,692	TL	1.02	36.1%	112,565	176,378	98	4,391	631	368,760
kroE 100-5	7.06	TL	50.9%	183,571	374,493	88	54	1,725	TL	16.02	47.3%	152,610	289,726	73	12,598	1,017	574,439
kroE 100-7	6.89	TL	51.6%	185,232	383,418	4	11	1,914	TL	40.56	66.4%	112,096	334,026	29	18,039	1,664	562,033
rd 100-3	8.59	TL	86.5%	40,868	303,678	4	43	1,648	TL	0.82	32.8%	41,839	62,299	100	41,11	655	364,425
rd 100-5	7.87	TL	71.1%	65,339	226,679	4	47	1,762	TL	19.19	44.2%	57,356	102,863	71	10,253	1,111	518,493
rd 100-7	7.24	TL	55.9%	65,467	148,579	4	8	1,935	TL	43.19	60.5%	43,571	110,381	37	12,873	1,746	511,121
eil 101-3	7.14	TL	79.5%	3,287	16,047	4	45	1,453	TL	0.37	24.1%	3,674	4,841	101	5,229	620	333,443
eil 101-5	7.21	TL	54.8%	5,357	11,863	4	27	1,634	TL	17.66	30.7%	5,461	7,882	65	10,663	1,122	506,929
eil 101-7	7.05	TL	33.4%	5,062	7,611	4	2	1,748	TL	62.91	40.9%	4,168	7,062	25	15,147	2,545	674,635
lin 105-3	7.80	TL	76.4%	75,081	318,579	103	71	1,632	TL	1.60	41.3%	67,715	115,490	105	5,223	516	315,384
lin 105-5	7.83	TL	76.0%	120,579	504,308	4	42	1,533	TL	13.87	57.8%	86,959	206,402	72	12,720	778	516,862
lin 105-7	7.79	TL	61.5%	128,352	333,495	4	12	1,771	TL	28.14	72.2%	69,366	250,115	44	15,067	1,128	461,921

Table 2.3: TSPLIB instances Class I (Continued)

Instance n - α	ILP							Branch-and-Benders-cut										
	$n = V $	CPU	BF	CPU	Gap	LB	UB	$ H $	n_{subtour}	Nodes	CPU	CPU SP	Gap	LB	UB	$ H $	n_{subtour}	n_{cut}
pr 107-3	9.32	TL	91.9%	190,911	2,385,632	4	30	2,691	TL	1.11	41.1%	199,286	338,561	107	3,345	1,083	349,290	
pr 107-5	8.59	TL	59.4%	295,875	728,943	88	52	2,196	TL	11.48	62.1%	220,363	582,574	63	26,069	500	636,886	
pr 107-7	8.51	TL	60.3%	305,257	770,197	56	36	2,422	TL	29.31	77.7%	157,259	707,651	50	30,922	996	758,536	
gr 120-3	15.63	TL	86.4%	8,187	60,433	4	56	1,260	TL	0.36	27.1%	8,871	12,175	120	4,196	377	316,598	
gr 120-5	15.62	TL	69.9%	13,348	44,448	4	41	1,771	TL	19.41	38.1%	12,633	20,417	70	12,538	733	565,381	
gr 120-7	16.05	TL	54.6%	12,898	28,444	4	15	1,653	TL	53.43	56.9%	9,248	21,485	35	19,144	1,426	641,049	
pr 124-3	17.78	TL	91.9%	303,039	3,761,838	4	40	1,472	TL	0.73	43.1%	286,074	503,516	123	4,172	295	353,965	
pr 124-5	18.03	TL	82.7%	480,318	2,789,242	4	33	1,518	TL	12.37	57.3%	379,314	889,813	100	11,396	459	416,275	
pr 124-7	18.35	TL	71.3%	510,575	1,780,801	4	21	1,482	TL	33.73	71.4%	295,194	1,034,222	48	13,645	833	473,415	
bier 127-3	20.41	TL	70.8%	578,263	1,986,425	45	28	1,769	TL	0.88	32.7%	611,707	909,386	127	4,187	338	222,848	
bier 127-5	20.16	TL	53.1%	953,785	2,034,727	4	31	1,229	TL	13.70	43.1%	809,667	1,424,299	93	9,449	526	396,276	
bier 127-7	20.13	TL	30.5%	893,477	1,286,204	4	20	1,273	TL	21.62	53.9%	592,431	1,286,204	4	11,146	520	609,578	
ch 130-3	23.11	TL	86.4%	31,120	229,913	4	64	1,479	TL	1.18	39.2%	30,131	49,579	130	4,879	388	420,793	
ch 130-5	23.17	TL	70.2%	50,168	168,914	4	38	2,047	TL	20.09	51.6%	40,161	83,019	88	9,700	658	480,737	
ch 130-7	23.31	TL	51.8%	51,812	107,549	4	9	1,167	TL	53.26	65.3%	30,237	87,323	37	15,141	1,181	580,641	
pr 136-3	30.15	TL	73.7%	451,092	1,716,241	135	52	1,842	TL	0.54	30.1%	528,825	756,690	136	3,600	346	284,840	
pr 136-5	30.45	TL	76.5%	786,663	3,357,899	4	63	1,441	TL	21.58	39.6%	762,796	1,264,352	84	13,435	552	542,474	
pr 136-7	30.26	TL	64.0%	773,278	2,151,626	4	30	1,227	TL	53.78	62.1%	537,144	1,418,319	35	16,061	858	523,914	
gr 137-3	30.00	TL	92.5%	2,087	28,100	4	34	1,291	TL	11.34	85.3%	784	5,338	101	7,823	349	399,114	
gr 137-5	29.99	TL	84.4%	3,239	20,888	4	31	1,386	TL	24.62	88.5%	820	7,181	80	12,748	578	503,091	
gr 137-7	31.40	TL	73.1%	3,665	13,645	4	32	1,224	TL	43.82	90.1%	730	7,419	51	17,289	848	606,392	
pr 144-3	40.51	TL	94.0%	262,330	4,440,560	4	38	1,163	TL	3.89	56.3%	227,456	520,611	141	3,826	406	213,421	
pr 144-5	45.04	TL	88.2%	384,567	3,270,326	4	36	1,291	TL	17.16	78.8%	200,821	950,844	109	12,864	435	541,317	
pr 144-7	38.00	TL	81.4%	389,988	2,097,469	4	14	1,352	TL	44.06	84.7%	180,516	1,181,508	52	12,236	656	435,516	
ch 150-3	50.59	OUT OF MEMORY							TL	0.64	31.5%	35,479	51,845	148	3,861	215	419,994	
ch 150-5	53.92	OUT OF MEMORY							TL	32.86	45.6%	49,582	91,204	96	11,235	564	503,043	
ch 150-7	51.09	TL	61.6%	57,221	149,030	4	17	1,135	TL	53.82	64.2%	36,116	100,907	40	12,819	738	485,936	
kroA 150-3	53.21	TL	75.6%	128,106	526,826	81	44	1,566	TL	0.83	37.0%	133,195	211,459	149	3,962	219	429,156	
kroA 150-5	51.87	OUT OF MEMORY							TL	22.44	47.8%	181,269	347,620	109	9,818	463	432,975	
kroA 150-7	48.99	TL	56.3%	225,404	515,900	4	12	1,171	TL	53.73	65.5%	135,500	393,675	45	15,027	717	523,385	
kroB 150-3	54.15	OUT OF MEMORY							TL	0.99	35.7%	131,973	205,416	148	3,859	212	417,708	
kroB 150-5	53.48	TL	76.4%	213,880	909,950	4	29	1,500	TL	30.61	51.6%	178,051	368,236	124	11,666	564	574,035	
kroB 150-7	54.37	TL	62.1%	219,400	580,224	4	14	1,808	TL	72.43	66.5%	129,009	385,297	57	14,616	933	564,083	
pr 152-3	53.82	TL	95.2%	304,290	634,0458	4	14	1,454	TL	3.26	50.7%	308,839	626,591	146	4,842	260	202,203	
pr 152-5	54.63	OUT OF MEMORY							TL	25.62	67.7%	368,917	1,144,246	85	11,262	408	420,985	
pr 152-7	53.95	TL	83.8%	481,421	2,976,190	4	30	2,183	TL	33.39	80.2%	279,497	1,417,690	54	13,544	456	413,989	
u 159-3	67.84	OUT OF MEMORY							TL	0.44	33.9%	227,697	344,631	158	3,607	171	379,612	
u 159-5	68.67	OUT OF MEMORY							TL	12.94	46.0%	320,234	593,091	124	10,119	268	385,372	
u 159-7	74.41	OUT OF MEMORY							TL	30.89	66.8%	232,807	701,283	53	11,619	369	395,303	
d 198-3	217.97	OUT OF MEMORY							TL	0.80	40.0%	74,440	124,172	198	4,355	139	205,655	
d 198-5	183.70	OUT OF MEMORY							TL	13.26	66.0%	94,576	278,430	90	15,103	92	506,090	
d 198-7	206.90	OUT OF MEMORY							OUT OF MEMORY									
kroA 200-3	199.44	OUT OF MEMORY							TL	0.88	38.6%	147,393	240,394	197	3,116	107	342,036	
kroA 200-5	215.01	OUT OF MEMORY							TL	28.83	53.4%	200,507	430,430	127	9,422	258	381,768	
kroA 200-7	224.94	OUT OF MEMORY							TL	74.11	69.2%	140,250	456,037	64	14,084	468	539,046	
kroB 200-3	225.57	OUT OF MEMORY							TL	0.72	35.8%	149,463	232,886	200	3,378	100	282,332	
kroB 200-5	227.45	OUT OF MEMORY							TL	36.56	57.0%	203,088	472,837	108	10,584	274	406,513	
kroB 200-7	248.91	OUT OF MEMORY							TL	28.74	69.4%	143,055	468,114	60	9,713	186	333,348	

Table 2.4: Randomly generated instances Class II

Instance n .ID	ILP						Branch-and-Benders-cut										
	$n = V $	CPU BF	CPU	Gap	LB	UB	$ H $	n_{subtour}	Nodes	CPU	CPU SP	Gap	LB	UB	$ H $	n_{subtour}	n_{cut}
ClassII - 20.1	0.00	9.11	0.0%	10,200	10,200	4	3	3,578	156.11	0.77	0.0%	10,200	10,200	4	367	2028	318,762
ClassII - 20.2	0.00	6.57	0.0%	9,748	9,748	4	4	2,456	19.17	0.27	0.0%	9,748	9,748	4	51	613	39,770
ClassII - 20.3	0.00	5.53	0.0%	10,626	10,626	4	0	362	12.05	0.16	0.0%	10,626	10,626	4	32	303	17,521
ClassII - 20.4	0.00	5.49	0.0%	8,973	8,973	4	0	59	8.60	0.06	0.0%	8,973	8,973	4	14	201	6,538
ClassII - 20.5	0.00	6.70	0.0%	10,636	10,636	4	1	2,913	26.49	0.34	0.0%	10,636	10,636	4	45	801	52,752
ClassII - 30.1	0.03	TL	5.0%	13,535	14,247	12	122	336,852	TL	11.31	23.1%	10,818	14,071	9	6,644	11,149	690,368
ClassII - 30.2	0.03	TL	7.3%	15,216	16,422	11	211	432,828	TL	9.85	20.1%	12,579	15,760	8	7,268	11,419	725,196
ClassII - 30.3	0.03	14.09	0.0%	13,562	13,562	4	0	4,318	59.39	1.21	0.0%	13,562	13,562	4	71	1,065	92,113
ClassII - 30.4	0.01	707.83	0.0%	14,226	14,226	4	21	59,512	TL	13.47	14.3%	12,178	14,226	4	3,841	14,313	723,707
ClassII - 30.5	0.03	TL	1.1%	16,665	16,866	4	29	348,546	TL	14.37	19.8%	13,520	16,866	4	6,355	13,213	834,502
ClassII - 40.1	0.09	TL	12.8%	16,387	18,807	14	182	87,176	TL	18.64	25.0%	13,519	18,044	14	9,694	9,055	720,714
ClassII - 40.2	0.09	TL	12.4%	16,993	19,418	4	147	71,162	TL	19.15	27.0%	13,773	18,868	10	9,729	9,800	702,605
ClassII - 40.3	0.09	TL	15.0%	17,562	20,663	18	126	82,723	TL	17.33	31.2%	14,198	20,663	13	10,936	8,364	696,748
ClassII - 40.4	0.09	TL	7.7%	16,457	17,848	4	60	61,405	TL	24.08	18.5%	14,542	17,848	4	6,583	10,719	819,114
ClassII - 40.5	0.08	TL	10.4%	16,431	18,349	15	95	74,015	TL	21.99	28.0%	13,091	18,205	13	9,276	9,860	690,326
ClassII - 60.1	0.57	TL	19.8%	18,626	23,229	18	124	21,647	TL	38.19	32.1%	15,124	22,298	16	11,797	6,210	776,880
ClassII - 60.2	0.53	TL	13.0%	18,743	21,553	17	103	18,928	TL	48.81	24.0%	15,982	21,040	14	10,695	7,092	781,985
ClassII - 60.3	0.53	TL	14.7%	20,732	24,328	4	120	28,213	TL	45.62	23.5%	17,666	23,108	15	10,362	6,562	686,436
ClassII - 60.4	0.53	TL	21.3%	22,348	28,409	20	142	44,166	TL	44.80	29.6%	17,850	25,359	13	12,673	6,542	772,988
ClassII - 60.5	0.54	TL	20.1%	20,912	26,191	19	71	32,436	TL	31.23	29.6%	17,026	24,199	15	11,048	5,451	658,263
ClassII - 100.1	5.98	TL	16.7%	26,563	31,902	32	79	2,088	TL	73.76	31.3%	21,356	31,102	21	10,526	2,888	557,919
ClassII - 100.2	6.01	TL	37.1%	23,846	37,918	4	41	1,753	TL	61.85	40.7%	18,062	30,485	23	11,056	2,605	583,691
ClassII - 100.3	5.96	TL	34.9%	25,881	39,776	4	71	1,958	TL	87.13	28.7%	21,331	29,952	18	10,788	3,572	700,113
ClassII - 100.4	5.96	TL	28.4%	25,253	35,284	38	70	1,875	TL	56.41	34.6%	19,660	30,091	26	10,758	2,414	560,264
ClassII - 100.5	5.89	TL	15.1%	26,871	31,685	29	82	1,768	TL	52.31	36.7%	20,505	32,427	24	9,887	2,205	478,566
ClassII - 125.1	16.40	TL	37.2%	30,752	49,028	33	99	1,363	TL	86.19	33.8%	24,318	36,737	28	11,462	1,919	584,555
ClassII - 125.2	16.28	TL	19.1%	30,306	37,483	36	51	1,502	TL	78.09	31.7%	24,575	36,014	24	10,112	1,825	521,375
ClassII - 125.3	16.08	TL	35.1%	29,856	46,008	4	50	1,491	TL	63.30	35.1%	23,735	36,599	22	9,573	1,454	468,394
ClassII - 125.4	15.84	TL	36.3%	28,817	45,257	4	64	1,727	TL	76.93	39.8%	21,122	35,128	33	11,189	1,673	554,088
ClassII - 125.5	16.30	TL	47.1%	29,860	56,531	4	70	1,553	TL	79.23	34.0%	23,485	35,594	29	11,085	1,783	571,615
ClassII - 150.1	39.01	OUT OF MEMORY							TL	79.30	34.3%	26,027	39,664	35	8,443	1,036	441,501
ClassII - 150.2	39.19	OUT OF MEMORY							TL	95.37	34.4%	26,594	40,563	27	9,625	1,240	472,963
ClassII - 150.3	39.67	OUT OF MEMORY							TL	94.80	35.7%	25,283	39,330	33	9,983	1,246	514,406
ClassII - 150.4	40.03	OUT OF MEMORY							TL	98.45	38.3%	25,687	41,682	32	10,111	1,325	550,316
ClassII - 150.5	40.03	OUT OF MEMORY							TL	95.33	35.6%	27,252	42,339	32	9,563	1,315	485,729
ClassII - 200.1	173.61	OUT OF MEMORY							TL	110.18	38.7%	30,212	49,353	43	8,407	714	428,736
ClassII - 200.2	174.60	OUT OF MEMORY							TL	108.87	37.1%	29,995	47,761	43	8,276	642	395,399
ClassII - 200.3	173.61	OUT OF MEMORY							TL	113.77	40.4%	29,816	50,077	36	8,263	729	444,891
ClassII - 200.4	180.11	OUT OF MEMORY							TL	107.59	40.0%	29,572	49,344	44	8,822	696	446,267
ClassII - 200.5	173.63	OUT OF MEMORY							TL	111.37	40.2%	29,482	49,379	43	7,345	668	389,473

Table 2.5: Variation of the results when 2-opt backup is not used

Instance n - α	ILP				Branch-and-Benders-cut			
	$n = V $	CPU	Gap	LB	UB	CPU	Gap	LB
eil 51-3	TL	+14.05%	-1.02%	+7.80%	TL	+187.92%	-0.60%	+144.78%
eil 51-5	TL	+15.43%	-2.43%	+5.36%	TL	+42.91%	-0.36%	+20.76%
eil 51-7	TL	0.00%	0.00%	0.00%	TL	0.00%	0.00%	0.00%
pr 124-3	TL	0.00%	0.00%	0.00%	TL	+113.92%	+2.41%	+647.11%
pr 124-5	TL	0.00%	0.00%	0.00%	TL	+51.13%	-1.59%	+213.46%
pr 124-7	TL	0.00%	0.00%	0.00%	TL	+17.09%	-1.53%	+72.19%
ClassII - 20.1	-1.54%	0.00%	0.00%	0.00%	+20.86%	0.00%	0.00%	0.00%
ClassII - 30.1	TL	+38.00%	+0.97%	+3.05%	TL	+17.75%	+1.46%	+7.16%
ClassII - 40.1	TL	+35.16%	-1.32%	+4.07%	TL	+42.00%	+0.45%	+16.81%
ClassII - 60.1	TL	-1.52%	+0.27%	-0.09%	TL	+41.43%	-1.37%	+22.54%
ClassII - 100.1	TL	+22.16%	+0.26%	+5.00%	TL	+40.89%	+1.09%	+24.28%
ClassII - 125.1	OUT OF MEMORY				TL	+22.49%	+0.16%	+13.33%

Chapter 3

A Resilient variant of the Ring Star Problem

3.1 Chapter's abstract

We study a Resilient variant of the Ring Star Problem named 1-R-RSP in which the failure of a node in a given subset of so-called uncertain nodes triggers two corrective operations to repair the Ring Star structure. We model this problem as an Integer Linear Program (ILP), that is also addressed with a Branch-and-Benders-cut decomposition. Several enhancements to both the ILP and Branch-and-Benders-cut algorithm are also presented. We then compare the solutions of the resilient and survivable variants, in order to determine which variant is the most appropriate, as a function of failure frequency. This chapter's work has been presented in ROADEF 2021 and 2023.

3.2 Chapter's structure

This chapter is structured as follows: in Section 3.3 we introduce the RESILIENT RING STAR PROBLEM and present an ILP formulation in Section 3.4. We describe a Branch-and-Benders-cut approach in Section 3.5. Numerical experiments are carried out for the two aforementioned approaches in Section 3.6. Finally, in Section 3.7, we address the question of determining which of the two strategies leads to the smallest cost, as a function of the failures duration.

3.3 Resilient Ring Star Problem definition

The **Ring** part aims to select a subset $H \subseteq V$ and link up all hubs of H with a cycle using edges of E . The cost of opening a hub $i \in V$ is $d_i \in \mathbb{R}_+$ and the cost of

selecting an edge $ij \in E$ between two hubs i and j is $c_{ij} \in \mathbb{R}_+$.

The **Star** requires that each terminal in $T = V \setminus H$ must be connected to exactly one hub in H . The cost of selecting an arc $(i, j) \in A$ to connect a terminal $i \in T$ to a hub $j \in H$ is $d_{ij} \in \mathbb{R}_+$.

The resilient ring-star network is designed so that when a hub $h \in \tilde{V}$ fails, two corrective operations occur:

- The **Ring** correction consists in restoring the ring by adding a backup edge that joins the two neighbors of the failed hub h . The cost of selecting a backup edge $ij \in E$ is $c'_{ij} \in \mathbb{R}_+$.
- The **Star** repair operation is to connect the terminals that were originally connected to the failing hub to another hub with backup arcs. The cost of selecting a backup arc $(i, j) \in A^\neq$ is $d'_{ij} \in \mathbb{R}_+$ where $A^\neq = \{(i, j) \in A \mid i \neq j\}$.

For the rest of this document we will use the following notations: For a set of nodes $S \subseteq H$, $E(S) = \{ij \in E \mid (i, j) \in S^2\}$, $E(S)$ is the set of all edges of E having both their endpoints in S . For a node $i \in V$, $\delta(i) = \{ij \mid ij \in E\}$ is the set of all edges incident to i .

3.4 A MILP formulation for Resilient RSP

We consider the case where $k = 1$ and let $F \in \mathbb{R}_+$ be the total duration of failures where a failure is a state when an uncertain hub is down. The 1-R-RSP can be formulated as a MILP described below. We will adapt the st-chains formulation for RSP given by Kedad-Sidhoum and Nguyen to 1-R-RSP. In particular, we set $s = v_1$ because 1 is the depot. We also add a dummy node t , a clone of the depot s in G to obtain a new graph $G' = (V', E' \cup A^\neq)$ where $V' = V \cup \{t\}$, $E' = E \cup \{st\} \cup \{tu \mid su \in E\}$. We note $\tilde{J} = \{(i, j, k) \in V \times \tilde{V} \times V' : i \neq j, j \neq k, i < k\}$. The following decision variables are used.

The binary variable x_{ij} , $\forall ij \in E'$ is such that $x_{ij} = 1$ if edge ij is selected as a regular edge for connecting hubs i and j . The binary variable y_{ij} , $\forall (i, j) \in A^\neq$ is such that $y_{ij} = 1$ if arc (i, j) is selected as a regular arc for connecting terminal i to hub j . The binary variable y_{ii} , $\forall i \in V'$ is such that $y_{ii} = 1$ if node i is selected as a hub. The binary variable x'_{ij} , $\forall ij \in E'$ is such that $x'_{ij} = 1$ if edge ij is selected as a backup edge between hubs i and j . The binary variable $y'_{ij} \in \mathbb{B}$, $\forall (i, j) \in A^\neq$ is such that $y'_{ij} = 1$ if arc (i, j) is selected as a backup arc between terminal i to hub j .

There are also two other decision variables, $\theta_{ij} \in \mathbb{R}_+$, $\forall (i, j) \in V \times \tilde{V}, i \neq j$ which is an upper bound on the cost of the re-connection of terminal $i \in T$ to the ring when hub $j \in \tilde{V}$ fails and $B \in \mathbb{R}_+$ which is the maximum total cost per unit of time incurred by re-connecting the terminals to the ring when the hub they are regularly connected to fails and fixing the ring by adding a backup edge in between two neighbors of the failing hub.

Constraints

Ring constraints

- (3.1) The ring must be made of at least four hubs. Indeed, if the ring has 3 or less hubs, the number of surviving hubs upon failure of any of them does not leave enough hubs to reform a ring by adding a backup edge. This case is sorted out by a brute force algorithm.
- (3.2) The selected edges in E must form a cycle while (3.3) prevent two or more distinct cycles. For all strict subset S of V having cardinality at most $\frac{1}{2}|V|$, the number of edges joining pairs of hubs in S must be strictly less than $|S|$ (to avoid having a cycle) provided that at least one hub is selected in $V \setminus S$.
- (3.4), (3.5) The depot and its clone must be connected.
- (3.6) The depot is a hub and cannot be assigned to another node.

$$\sum_{i \in V} y_{ii} \geq 4 \quad (3.1)$$

$$\sum_{ij \in \delta(i)} x_{ij} = 2y_{ii}, \quad \forall i \in V' \setminus \{s, t\} \quad (3.2)$$

$$x(\delta(S)) \geq 2 \sum_{j \in S} y_{ij}, \quad \forall S \subset V' \setminus \{s, t\}, \forall i \in S \quad (3.3)$$

$$\sum_{i \in V \setminus \{s\}} x_{si} = 1 \quad (3.4)$$

$$\sum_{i \in V \setminus \{s\}} x_{it} = 1 \quad (3.5)$$

$$y_{ss} = y_{tt} = 1, y_{si} = 0, \quad \forall i \in V' \setminus \{s, t\} \quad (3.6)$$

Star constraints

- (3.7) A node is either selected as a hub or is a terminal and is connected to exactly one hub.
- (3.8) If a terminal is connected to a hub, then this hub must be opened and the edge between the terminal and the hub must not be in the cycle.

$$\sum_{j \in V} y_{ij} = 1, \quad \forall i \in V \quad (3.7)$$

$$y_{ij} \leq y_{jj} - x_{ij}, \quad \forall (i, j) \in A^\neq \quad (3.8)$$

Backup constraints

- (3.9) For a selected uncertain hub and its neighbors in the ring, the edge between those two last hubs is a backup edge.
- (3.9') If we consider that the depot can fail, we must select a backup edge between its two neighbors. These constraints can be removed if the depot is certain.
- (3.10) Every terminal has a backup arc to recover from a failure only if it is connected to an uncertain hub.
- (3.11) Whenever a backup arc joins a terminal to a hub, this hub should be opened and this arc can not be a regular arc
- (3.11') There is at most one arc or edge between any two nodes i and j . In addition, j must be a hub if there is an edge or an arc from i to j . If we have (3.11') then we can remove (3.11) and (3.8) in the ILP. This is not the case when the Benders decomposition is applied because we want the Master problem of the Benders decomposition to have classical RSP solutions.

$$x_{ij} + x_{jk} - 1 \leq x'_{ik}, \quad \forall (i, j, k) \in \tilde{J} \quad (3.9)$$

$$x_{si} + x_{kt} - 1 \leq x'_{ik}, \quad \forall (i, k) \in (V \setminus \{s\})^2, i < k \quad (3.9')$$

$$\sum_{(i,j) \in A^\neq} y'_{ij} = 1 - y_{ii} - \sum_{j \in V \setminus \{\tilde{V} \cup \{i\}\}} y_{ij}, \quad \forall i \in V \quad (3.10)$$

$$y'_{ij} \leq y_{jj} - y_{ij}, \quad \forall (i, j) \in A^\neq \quad (3.11)$$

$$x_{ij} + y_{ij} + x'_{ij} + y'_{ij} \leq y_{jj}, \quad \forall (i, j) \in A^\neq \quad (3.11')$$

Backup cost constraints

- (3.12) Let $i \in V$ be a terminal regularly connected to a hub $j \in \tilde{V}$. The existence of this regular connection in the solution leads to $y_{ij} = 1$. The cost for reconnecting terminal i to another hub whenever hub j fails can be computed as:

$$\max_{k \in V \setminus \{i, j\}} d'_{ik} y'_{ik}$$

Let θ_{ij} be an upper bound on the cost of reconnecting terminal i to the ring when hub $j \in \tilde{V}$ fails.

(3.13) Finally, the maximum cost incurred by the failure of any hub is B . The first term of the left hand side is the cost of fixing the ring by using the backup edge ik when hub $j \in \tilde{V}$ fails. The second one is the cost of using backup arcs to reconnect the terminals to surviving hubs when h , the hub to which they were regularly connected to, fails

$$\sum_{k \in V \setminus \{i,j\}} d'_{ik}(y'_{ik} + y_{ij} - 1) \leq \theta_{ij}, \quad \forall (i,j) \in V \times \tilde{V} \setminus \{i\} \quad (3.12)$$

$$c'_{ik}(x'_{ik} + x_{ij} + x_{jk} - 2) + \sum_{t \in V \setminus \{j\}} \theta_{tj} \leq B, \quad \forall (i,j,k) \in \tilde{J} \quad (3.13)$$

Objective function

An optimal solution to 1-R-RSP should minimize the total cost which is the deployment cost: (selecting hubs, connecting them with the ring, and connecting terminals to the ring), plus an upper bound of the fixing cost, which is computed by assuming that the hub incurring the highest fixing cost fails for F units of time during the time horizon, i.e., minimize:

1. The cost of selecting hubs that will be part of the ring;
2. The cost of connecting all terminals to the ring with regular arcs;
3. The cost of regular edges to form the ring;
4. An upper bound of the fixing cost, which is computed by assuming that the hub incurring the highest fixing cost fails for F units of time during the time horizon

$$\text{Minimize } \sum_{i \in V} d_i y_{ii} + \sum_{(i,j) \in A^\#} d_{ij} y_{ij} + \sum_{ij \in E'} c_{ij} x_{ij} + FB$$

The mathematical model of 1-R-RSP is then given by the following MILP:

$$\begin{array}{l}
\text{Minimize} \\
\sum_{i \in V} d_i y_{ii} + \sum_{(i,j) \in A^\neq} d_{ij} y_{ij} + \sum_{ij \in E'} c_{ij} x_{ij} + FB \\
\sum_{i \in V} y_{ii} \geq 4 \\
\sum_{ij \in \delta(i)} x_{ij} = 2y_{ii} \quad \forall i \in V' \setminus \{s, t\} \\
x(\delta(S)) \geq 2 \sum_{j \in S} y_{ij} \quad \forall S \subset V' \setminus \{s, t\}, \forall i \in S \\
\sum_{i \in V \setminus \{s\}} x_{si} = 1 \\
\sum_{i \in V \setminus \{s\}} x_{it} = 1 \\
y_{si} = 0 \quad \forall i \in V' \setminus \{s, t\} \\
y_{tt} = 1 \\
y_{ss} = 1 \\
\sum_{j \in V} y_{ij} = 1 \quad \forall i \in V \\
x'_{ik} \geq x_{ij} + x_{jk} - 1 \quad \forall (i, j, k) \in \tilde{J} \\
x'_{ik} \geq x_{si} + x_{kt} - 1 \quad \forall (i, k) \in (V \setminus \{s\})^2, i < k \\
y_{jj} \geq x_{ij} + y_{ij} + y'_{ij} + x'_{ij} \quad \forall (i, j) \in A^\neq \\
1 - y_{ii} - \sum_{j \in V \setminus \{\tilde{V} \cup \{i\}\}} y_{ij} = \sum_{(i,j) \in A^\neq} y'_{ij} \quad \forall i \in V \\
\sum_{k \in V \setminus \{i, j\}} d'_{ik} (y'_{ik} + y_{ij} - 1) \leq \theta_{ij} \quad \forall (i, j) \in V \times \tilde{V} \setminus \{i\} \\
c'_{ik} (x'_{ik} + x_{ij} + x_{jk} - 2) + \sum_{t \in V \setminus \{j\}} \theta_{tj} \leq B \quad \forall (i, j, k) \in \tilde{J} \\
x_{ij} \in \mathbb{B} \quad \forall ij \in E' \\
x'_{ij} \in \mathbb{B} \quad \forall ij \in E' \\
y_{ij} \in \mathbb{B} \quad \forall (i, j) \in A \\
y'_{ij} \in \mathbb{B} \quad \forall (i, j) \in A^\neq \\
\theta_{ij} \in \mathbb{R}_+ \quad \forall i \in V, \forall j \in \tilde{V} \setminus \{i\} \\
B \in \mathbb{R}_+
\end{array}$$

3.4.1 Post-optimization procedure

After solving the MILP formulation of 1-R-RSP, we either have $B = 0$ or $B > 0$. The case where $B = 0$ occurs when all the hubs are certain, it is possible only when the number of certain nodes is at least three. The corresponding solution has a zero fixing cost because no hub can fail. If $B > 0$, then there exists at least one uncertain hub that incurs a maximum fixing cost when it fails. This hub is said to be *critical*, as it sets the numerical value of B . Non-critical uncertain hubs have a fixing cost that is strictly less than B , so there may be optimal solutions to 1-R-RSP in which some non-critical hubs have a non-minimum fixing cost. In order to avoid this, the post-optimization procedure described in Algorithm 5 is executed. It recomputes the value of all the θ decision variables, knowing \hat{y} and \hat{y}' , the numerical value of the y and y' variables in an optimal solution to the MILP formulation of 1-R-RSP.

For the sake of illustration, Figure 3.1 shows the nonzero x , y , x' and y' variables of an optimal solution to the MILP formulation of 1-R-RSP on a 10-node

Algorithm 5: A post-optimization procedure that sets the θ variables to their minimum value

```

1 foreach  $i \in V$  do
2   foreach  $j \in V$  do
3     if  $\hat{y}_{ij} = 1$  and  $i \neq j$  and  $j \in \tilde{V}$  then
4        $k \leftarrow 1$ 
5       while  $\hat{y}'_{ik} = 0$  do
6          $k \leftarrow k + 1$ 
7          $\theta_{ij} \leftarrow d'_{ik}$ 
8     else
9        $\theta_{ij} \leftarrow 0$ 

```

instance in which the depot is the only certain node. The x and y nonzero variables form a solution to RSP on that instance, it is shown in red. The plain blue edges are the backup edges. In particular, the backup edge (1,6) is used to fix the failure of hubs 3 and 10. The blue dashed arcs are the nonzero backup arcs in that solution. It can be seen that no backup arc is needed for terminals 5 and 9, as they are connected to a certain hub. All the other terminals are associated exactly one backup arc, to be used upon failure of the uncertain hub they are regularly connected to.

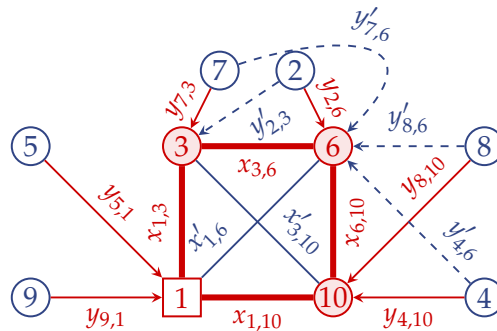


Figure 3.1: The nonzero x , x' , y and y' variables of a solution to 1-R-RSP

The numerical value of B in the optimal solution to 1-R-RSP shown in Figure 3.1 can also be deduced, as well as the minimum value for all the nonzero θ variables:

$$\begin{aligned} B &= \max \left(c'_{1,6} + d'_{7,6}, c'_{3,10} + d'_{2,3}, c'_{1,6} + d'_{4,6} + d'_{8,6} \right) \\ \theta_{2,6} &\geq d'_{2,3} \\ \theta_{4,10} &\geq d'_{4,6} \\ \theta_{7,3} &\geq d'_{7,6} \\ \theta_{8,10} &\geq d'_{8,6} \end{aligned}$$

The value of B is the maximum over 3 quantities, that represent the cost incurred by the failure of hubs 3, 6 and 10 respectively. $\theta_{2,6}$ is an upper bound on the cost of reconnecting terminal 2 when hub 6 fails. This cost may be strictly larger than $d'_{2,3}$ if hub 6 is non-critical, as long as $c'_{3,10} + d'_{2,3}$ is strictly less than B .

3.5 A Branch-and-Benders-cut decomposition of 1-R-RSP

The complicating variables are x and y because when they are known in advance, the subproblem becomes easy to solve and consists of computing the worst-case repairing cost among each hub failure of the ring. This means that the master problem is to solve RSP which is a hard problem.

3.5.1 Master problem

In a regular Benders decomposition approach, (3.11') is in the subproblem, so the master problem would return solutions in which terminals may possibly be connected to some other terminals. Hence the solutions of the master problem are not RSP solutions, and may be overly relaxed. To avoid this, we add (3.8) in the master problem which guarantees that the solutions to the master problem are

solutions to RSP.

$$\left\{ \begin{array}{l}
 \text{Minimize } \sum_{i \in V} d_i y_{ii} + \sum_{(i,j) \in A^\neq} d_{ij} y_{ij} + \sum_{ij \in E'} c_{ij} x_{ij} + \eta \\
 \sum_{i \in V} y_{ii} \geq 4 \\
 \sum_{ij \in \delta(i)} x_{ij} = 2y_{ii} \quad \forall i \in V' \setminus \{s, t\} \\
 x(\delta(S)) \geq 2 \sum_{j \in S} y_{ij} \quad \forall S \subset V' \setminus \{s, t\}, \forall i \in S \\
 \sum_{i \in V \setminus \{s\}} x_{si} = 1 \\
 \sum_{i \in V \setminus \{s\}} x_{it} = 1 \\
 y_{si} = 0 \quad \forall i \in V' \setminus \{s, t\} \\
 y_{tt} = 1 \\
 y_{ss} = 1 \\
 \sum_{j \in V} y_{ij} = 1 \quad \forall i \in V \\
 y_{jj} - x_{ij} \geq y_{ij} \quad \forall (i, j) \in A^\neq \\
 x_{ij} \in \mathbb{B} \quad \forall ij \in E' \\
 y_{ij} \in \mathbb{B} \quad \forall (i, j) \in A \\
 \eta \in \mathbb{R}_+
 \end{array} \right.$$

3.5.2 Benders subproblem

Assuming that the complicating variables are given as input parameters \hat{x} and \hat{y} , the primal subproblem can be formulated as follows:

$$\left\{ \begin{array}{l}
 \eta = \text{Minimize } FB \\
 x'_{ik} \geq \hat{x}_{ij} + \hat{x}_{jk} - 1 \quad \forall (i, j, k) \in \tilde{J} \\
 \sum_{(i,j) \in A^\neq} y'_{ij} = 1 - \hat{y}_{ii} - \sum_{j \in V \setminus \{\tilde{V} \cup \{i\}\}} \hat{y}_{ij} \quad \forall i \in V \\
 \theta_{ij} - \sum_{k \in V \setminus \{i, j\}} d'_{ik} y'_{ik} \geq \sum_{k \in V \setminus \{i, j\}} d'_{ik} (\hat{y}_{ij} - 1) \quad \forall i \in V, \forall j \in \tilde{V} \setminus \{i\} \\
 B - c'_{ik} x'_{ik} - \sum_{t \in V \setminus \{j\}} \theta_{ij} \geq c'_{ik} (\hat{x}_{ij} + \hat{x}_{jk} - 2) \quad \forall (i, j, k) \in \tilde{J} \\
 x'_{ij} + y'_{ij} \leq \hat{y}_{jj} - \hat{x}_{ij} - \hat{y}_{ij} \quad \forall (i, j) \in A^\neq \\
 x'_{ij} \in \mathbb{B} \quad (i, j) \in V \times V', i < j \\
 y'_{ij} \in \mathbb{B} \quad \forall (i, j) \in A^\neq \\
 \theta_{ij} \in \mathbb{R}_+ \quad \forall i \in V, \forall j \in \tilde{V} \setminus \{i\} \\
 B \in \mathbb{R}_+
 \end{array} \right.$$

Moreover, this integer linear program is easy to solve, and the optimal objective value is $B(\hat{x}, \hat{y}) = 0$ if the set $\{j : j \in \tilde{V}, \hat{y}_{jj} = 1\}$ is empty, otherwise:

$$B(\hat{x}, \hat{y}) = \min_{\substack{(i,j,k) \in \tilde{J} \\ \hat{x}_{ij} = \hat{x}_{jk} = 1}} \left(c'_{ik} + \sum_{\substack{i' \in V \setminus \{j\} \\ \hat{y}_{i'j} = 1}} \min_{\substack{k' \in V \setminus \{j\} \\ \hat{y}_{k'k'} = 1}} d'_{i'k'} \right)$$

A feasible solution to this integer linear program can be built as follows:

- x'_{ij} is initialized to zero for all $(i, j) \in A^\neq$.
- If $\sum_{i \in V} \hat{y}_{ii} = 4$, let s, a, b, c, t be the st -chain that represents the ring. If $s \in \tilde{V}$ or $b \in \tilde{V}$, then set x'_{ac} to one. If $a \in \tilde{V}$ or $c \in \tilde{V}$, then set x'_{sb} to one.
- Otherwise, *i.e.*, $\sum_{i \in V} \hat{y}_{ii} \geq 5$, then for all $(i, j, k) \in \tilde{J}$, if $\hat{x}_{ij} = \hat{x}_{jk} = 1$, then x'_{ik} is set to one.
- For all $i \in V$ such that $\hat{y}_{ii} = 1$, *i.e.*, node i is a hub, there is no backup arc leaving i , so y'_{ij} is set to zero for all $j \in V \setminus \{i\}$, and θ_{ij} is set to zero for all $j \in \tilde{V} \setminus \{i\}$.
- For all $i \in V$ such that $\hat{y}_{ii} = 0$, *i.e.*, node i is a terminal, there exists a unique hub $j \in V \setminus \{i\}$ such that $\hat{y}_{ij} = 1$. One of the following two sub-cases applies:
 - If $j \in V \setminus (\tilde{V} \cup \{i\})$, *i.e.*, node j is a certain hub, then j cannot fail and there is no backup arc leaving i , so y'_{ik} is set to zero for all $k \in V \setminus \{i\}$, and θ_{ik} is set to zero for all $k \in \tilde{V} \setminus \{i\}$.
 - If $j \in \tilde{V} \setminus \{i\}$, *i.e.*, node j is an uncertain hub, let k be a hub such that $d'_{ik} = \min_{\substack{q \in V \setminus \{j\} \\ \hat{y}_{qq} = 1}} d'_{iq}$, hence y'_{ik} is set to one, and y'_{iq} is set to zero for all $q \in V \setminus \{i, k\}$. Also, θ_{ij} is set to d'_{ik} as this is the cost incurred by reconnecting terminal i to hub k when hub j fails, and θ_{iq} is set to zero for all $q \in \tilde{V} \setminus \{i, j\}$.

Now, we consider the LP-relaxation of the primal of the Benders subproblem, where the upper bound on x' and y' are relaxed, because there exists an optimal solution to this LP that satisfies them:

$$\left\{ \begin{array}{ll} \eta = \text{Minimize } FB & \\ \sum_{(i,j) \in A^\neq} y'_{ij} = 1 - \hat{y}_{ii} - \sum_{j \in V \setminus \{\tilde{V} \cup \{i\}\}} \hat{y}_{ij} & \forall i \in V \quad [\alpha_i] \\ x'_{ik} \geq \hat{x}_{ij} + \hat{x}_{jk} - 1 & \forall (i,j,k) \in \tilde{J} \quad [\beta_{ijk}] \\ \theta_{ij} - \sum_{k \in V \setminus \{i,j\}} d'_{ik} y'_{ik} \geq \sum_{k \in V \setminus \{i,j\}} d'_{ik} (\hat{y}_{ij} - 1) & \forall i \in V, \forall j \in \tilde{V} \setminus \{i\} \quad [\gamma_{ij}] \\ B - c'_{ik} x'_{ik} - \sum_{t \in V \setminus \{j\}} \theta_{tj} \geq c'_{ik} (\hat{x}_{ij} + \hat{x}_{jk} - 2) & \forall (i,j,k) \in \tilde{J} \quad [\delta_{ijk}] \\ -x'_{ij} - y'_{ij} \geq \hat{x}_{ij} + \hat{y}_{ij} - \hat{y}_{jj} & \forall (i,j) \in A^\neq \quad [\epsilon_{ij}] \\ x'_{ij} \geq 0 & \forall ij \in E' \\ y'_{ij} \geq 0 & \forall (i,j) \in A^\neq \\ \theta_{ij} \geq 0 & \forall i \in V, \forall j \in \tilde{V} \setminus \{i\} \\ B \geq 0 & \end{array} \right.$$

The dual of this linear program is

$$\left\{ \begin{array}{ll} \eta = \text{Maximize } \sum_{i \in V} \left(1 - \hat{y}_{ii} - \sum_{j \in V \setminus \{\tilde{V} \cup \{i\}\}} \hat{y}_{ij} \right) \alpha_i + \sum_{(i,j,k) \in \tilde{J}} (\hat{x}_{ij} + \hat{x}_{jk} - 1) \beta_{ijk} & \\ + \sum_{(i,j) \in V \times \tilde{V} \setminus \{i\}} \left(\sum_{k \in V \setminus \{i,j\}} d'_{ik} (\hat{y}_{ij} - 1) \right) \gamma_{ij} + \sum_{(i,j,k) \in \tilde{J}} c'_{ik} (\hat{x}_{ij} + \hat{x}_{jk} - 2) \delta_{ijk} & \\ + \sum_{(i,j) \in A^\neq} (\hat{x}_{ij} + \hat{y}_{ij} - \hat{y}_{jj}) \epsilon_{ij} & \\ \sum_{(i,j,k) \in \tilde{J}} (\beta_{ijk} - c'_{ik} \delta_{ijk}) - \epsilon_{ik} - \epsilon_{ki} \leq 0 & \forall (i,k) \in V \times V, i < k \quad [x'_{ik}] \\ \sum_{(i,j,t) \in \tilde{J}} (\beta_{ijt} - c'_{it} \delta_{ijt}) \leq 0 & \forall i \in V \quad [x'_{it}] \\ \sum_{(i,j,k) \in \tilde{J}} \delta_{ijk} \leq F & [B] \\ \alpha_i - d'_{ij} \sum_{k \in \tilde{V} \setminus \{i,j\}} \gamma_{ik} - \epsilon_{ij} \leq 0 & \forall i \in V, \forall j \in V \setminus \{i\} \quad [y'_{ij}] \\ \gamma_{ij} - \sum_{(q,j,k) \in \tilde{J}} \delta_{qjk} \leq 0 & \forall i \in V, \forall j \in \tilde{V} \setminus \{i\} \quad [\theta_{ij}] \\ \alpha_i \in \mathbb{R} & \forall i \in V \\ \beta_{ijk} \geq 0 & \forall (i,j,k) \in \tilde{J} \\ \gamma_{ij} \geq 0 & \forall (i,j) \in V \times \tilde{V} \setminus \{i\} \\ \delta_{ijk} \geq 0 & \forall (i,j,k) \in \tilde{J} \\ \epsilon_{ij} \geq 0 & \forall (i,j) \in A^\neq \end{array} \right.$$

We propose an optimal solution to the dual, that is built as shown in Algorithm 7. To compute this solution, we need to determine the *critical triplet* $(i^*, j^*, k^*) \in \tilde{J}$ such that $\hat{x}_{i^*j^*} = \hat{x}_{j^*k^*} = 1$, and

$$c'_{i^*k^*} + \sum_{\substack{i \in V \setminus \{j^*\} \\ \hat{y}_{ij^*} = 1}} \min_{\substack{k \in V \setminus \{j^*\} \\ \hat{y}_{kk} = 1}} d'_{ik} = \min_{\substack{(i,j,k) \in \tilde{J} \\ \hat{x}_{ij} = \hat{x}_{jk} = 1}} \left(c'_{ik} + \sum_{\substack{i' \in V \setminus \{j\} \\ \hat{y}_{i'j} = 1}} \min_{\substack{k' \in V \setminus \{j\} \\ \hat{y}_{k'k'} = 1}} d'_{i'k'} \right)$$

Algorithm 6: CRITICALTRIPLET($n, V, \tilde{V}, V', E', c', d', \hat{x}, \hat{y}$)

 returns B and the critical triplet (i^*, j^*, k^*)

```

1 Let  $adj$  be an array of  $|V'|$  empty lists;
2 foreach  $(i, j) \in E'$  do
3   if  $\hat{x}_{ij} = 1$  then
4      $adj[i].append(j)$ ;
5      $adj[j].append(i)$ ;
6 foreach  $i \in V : \hat{y}_{ii} = 1$  do
7   if  $adj[i][1] > adj[i][2]$  then
8     Exchange  $adj[i][1]$  and  $adj[i][2]$ ;
9 Let  $costReconnection$  be a  $|V|$ -element array;
10 foreach  $i \in V$  do
11    $costReconnection[i] \leftarrow 0.0$ ;
12 foreach  $i \in V : \hat{y}_{ii} = 0$  do
13   /* Determining  $j$  such that  $\hat{y}_{ij} = 1$  */
14    $j \leftarrow 1$ ;
15   while  $\hat{y}_{ij} = 0$  do
16      $j \leftarrow j + 1$ ;
17   if  $j \in \tilde{V}$  then
18      $costReconnection[i] \leftarrow \infty$ ;
19     foreach  $k \in V \setminus \{j\} : \hat{y}_{kk} = 1$  do
20       if  $d'_{ik} < costReconnection[i]$  then
21          $costReconnection[i] \leftarrow d'_{ik}$ ;
22   /* Computing the critical triplet  $(i^*, j^*, k^*)$  */
23    $B \leftarrow 0.0$ ;
24    $(i^*, j^*, k^*) \leftarrow (-1, -1, -1)$ ;
25   foreach  $j \in \tilde{V} : \hat{y}_{jj} = 1$  do
26      $hubFixingCost \leftarrow c'_{adj[j][1]adj[j][2]}$ ;
27     foreach  $i \in V \setminus \{j\}$  do
28       if  $\hat{y}_{ij} = 1$  then
29          $hubFixingCost \leftarrow$ 
30            $hubFixingCost + costReconnection[i]$ ;
31   if  $hubFixingCost > B$  then
32      $B \leftarrow hubFixingCost$ ;
33      $(i^*, j^*, k^*) \leftarrow (adj[j][1], j, adj[j][2])$ ;
34 return  $B, i^*, j^*, k^*$ ;

```

Algorithm 7: Computation of an optimal solution to the subproblem's dual

```

1 Input:  $n, \tilde{V}, F, c', d', \hat{x}, \hat{y}$ 
   /* Initialization */
2 for  $i = 1$  to  $n$  do
3    $\alpha_i \leftarrow 0$ 
4 for  $i = 1$  to  $n$  do
5   for  $j = 1$  to  $n$  do
6     if  $j \neq i$  and  $j \in \tilde{V}$  then
7       for  $k = i + 1$  to  $n + 1$  do
8         if  $k \neq j$  then
9            $\beta_{ijk} \leftarrow 0$ 
10           $\delta_{ijk} \leftarrow 0$ 
11 for  $i = 1$  to  $n$  do
12   for  $j = 1$  to  $n$  do
13     if  $j \neq i$  and  $j \in \tilde{V}$  then
14        $\gamma_{ij} \leftarrow 0$ 
15 for  $i = 1$  to  $n$  do
16   for  $j = 1$  to  $n$  do
17     if  $j \neq i$  then
18        $\epsilon_{ij} \leftarrow 0$ 
   /* Setting the nonzero variables */
19  $B, i^*, j^*, k^* \leftarrow \text{CRITICALTRIPLET}(n, \tilde{V}, c', d', \hat{x}, \hat{y})$ 
20 for  $i = 1$  to  $n$  do
21   if  $i \neq j^*$  and  $\hat{y}_{ij^*} = 1$  then
22      $mind \leftarrow +\infty$ 
23     for  $k = 1$  to  $n$  do
24       if  $k \neq i$  and  $k \neq j^*$  and  $\hat{y}_{kk} = 1$  and  $d'_{ik} < mind$  then
25          $mind \leftarrow d'_{ik}$ 
26      $\alpha_i \leftarrow F \cdot mind$ 
27  $\beta_{i^*j^*k^*} = F \cdot c'_{i^*k^*}$ 
28  $\delta_{i^*j^*k^*} = F$ 
29 for  $i = 1$  to  $n$  do
30   if  $i \neq j^*$  and  $\hat{y}_{ij^*} = 1$  then
31      $\gamma_{ij^*} \leftarrow F$ 
32 for  $i = 1$  to  $n$  do
33   if  $i \neq j^*$  and  $\hat{y}_{ij^*} = 1$  then
34      $\epsilon_{ij^*} \leftarrow \alpha_i$ 
35     for  $j = 1$  to  $n$  do
36       if  $j \neq i$  and  $\hat{y}_{jj} = 0$  and  $\alpha_i > F \cdot d'_{ij}$  then
37          $\epsilon_{ij} \leftarrow \alpha_i - F \cdot d'_{ij}$ 
38 return  $\alpha, \beta, \delta, \gamma$ 

```

This is achieved by Algorithm 6. The solution returned by Algorithm 7 is feasible for the dual, and the corresponding objective value is $c'_{i^*k^*} + \sum_{\substack{i \in V \setminus \{j^*\} \\ \hat{y}_{ij^*}=1}} \min_{\substack{k \in V \setminus \{j^*\} \\ \hat{y}_{kk}=1}} d'_{ik}$, *i.e.*, it is the same as the one of the LP relaxation of the primal, and also the same as the one of the integer linear programming formulation of the Benders subproblem, which proves that both these solutions are optimal.

3.5.3 Benders optimality cut

The objective function of the dual of the subproblem is

$$\begin{aligned} \eta = & \sum_{i \in V} \left(1 - \hat{y}_{ii} - \sum_{j \in V \setminus \{\tilde{V} \cup \{i\}\}} \hat{y}_{ij} \right) \alpha_i + \sum_{(i,j,k) \in \tilde{f}} (\hat{x}_{ij} + \hat{x}_{jk} - 1) \beta_{ijk} \\ & + \sum_{(i,j) \in V \times \tilde{V} \setminus \{i\}} \left(\sum_{k \in V \setminus \{i,j\}} d'_{ik} (\hat{y}_{ij} - 1) \right) \gamma_{ij} + \sum_{(i,j,k) \in \tilde{f}} c'_{ik} (\hat{x}_{ij} + \hat{x}_{jk} - 2) \delta_{ijk} \\ & + \sum_{(i,j) \in A^\neq} (\hat{x}_{ij} + \hat{y}_{ij} - \hat{y}_{jj}) \epsilon_{ij} \end{aligned}$$

The solution of the dual of the LP relaxation is used and in the master problem, we should add the optimality cut:

$$\begin{aligned} \eta \geq & \sum_{\substack{i \in V \setminus \{j^*\} \\ \hat{y}_{ij^*}=1}} \left(1 - y_{ii} - \sum_{j \in V \setminus \{\tilde{V} \cup \{i\}\}} y_{ij} \right) \left(F \min_{\substack{k \in V \setminus \{i,j^*\} \\ \hat{y}_{kk}=1}} d'_{ik} \right) + (x_{i^*j^*} + x_{j^*k^*} - 1) (Fc'_{i^*k^*}) \\ & + \sum_{\substack{i \in V \setminus \{j^*\} \\ \hat{y}_{ij^*}=1}} \left(\sum_{k \in V \setminus \{i,j^*\}} d'_{ik} (y_{ij^*} - 1) \right) F + c'_{i^*k^*} (x_{i^*j^*} + x_{j^*k^*} - 2) F \\ & + \sum_{\substack{i \in V \setminus \{j^*\} \\ \hat{y}_{ij^*}=1}} \left[\sum_{\substack{j \in V \setminus \{i\} \\ \hat{y}_{jj}=0}} \left(F \cdot \min_{\substack{k \in V \setminus \{i,j^*\} \\ \hat{y}_{kk}=1}} d'_{ik} - F \cdot d'_{ij} \right) (x_{ij} + y_{ij} - y_{jj}) + \left(F \cdot \min_{\substack{k \in V \setminus \{i,j^*\} \\ \hat{y}_{kk}=1}} d'_{ik} \right) (x_{i^*j^*} + y_{ij^*} - y_{j^*j^*}) \right] \end{aligned}$$

Since $\eta = FB$, we can simplify this optimality cut to

$$\begin{aligned} B \geq & \sum_{\substack{i \in V \setminus \{j^*\} \\ \hat{y}_{ij^*}=1}} \left(1 - y_{ii} - \sum_{j \in V \setminus \{\tilde{V} \cup \{i\}\}} y_{ij} \right) \left(\min_{\substack{k \in V \setminus \{i,j^*\} \\ \hat{y}_{kk}=1}} d'_{ik} \right) + (2x_{i^*j^*} + 2x_{j^*k^*} - 3)c'_{i^*k^*} \\ & + \sum_{\substack{i \in V \setminus \{j^*\} \\ \hat{y}_{ij^*}=1}} \sum_{k \in V \setminus \{i,j^*\}} d'_{ik} (y_{ij^*} - 1) \\ & + \sum_{\substack{i \in V \setminus \{j^*\} \\ \hat{y}_{ij^*}=1}} \left[\sum_{\substack{j \in V \setminus \{i\} \\ \hat{y}_{jj}=0}} \left(\min_{\substack{k \in V \setminus \{i,j^*\} \\ \hat{y}_{kk}=1}} d'_{ik} - d'_{ij} \right) (x_{ij} + y_{ij} - y_{jj}) + \left(\min_{\substack{k \in V \setminus \{i,j^*\} \\ \hat{y}_{kk}=1}} d'_{ik} \right) (x_{i^*j^*} + y_{ij^*} - y_{j^*j^*}) \right] \end{aligned}$$

$$\begin{aligned}
B \geq & \sum_{\substack{i \in V \setminus \{j^*\} \\ \hat{y}_{ij^*} = 1}} \left(1 - y_{ii} + x_{ij^*} + y_{ij^*} - y_{j^*j^*} - \sum_{j \in V \setminus \{\bar{V} \cup \{i\}\}} y_{ij} \right) \left(\min_{\substack{k \in V \setminus \{i, j^*\} \\ \hat{y}_{kk} = 1}} d'_{ik} \right) + (2x_{i^*j^*} + 2x_{j^*k^*} - 3)c'_{i^*k^*} \\
& + \sum_{\substack{i \in V \setminus \{j^*\} \\ \hat{y}_{ij^*} = 1}} \sum_{k \in V \setminus \{i, j^*\}} d'_{ik} (y_{ij^*} - 1) \\
& + \sum_{\substack{i \in V \setminus \{j^*\} \\ \hat{y}_{ij^*} = 1}} \sum_{\substack{j \in V \setminus \{i\} \\ \hat{y}_{jj} = 0}} \left(\min_{\substack{k \in V \setminus \{i, j^*\} \\ \hat{y}_{kk} = 1}} d'_{ik} - d'_{ij} \right) (x_{ij} + y_{ij} - y_{jj}) \\
& \quad \min_{\substack{k \in V \setminus \{i, j^*\} \\ \hat{y}_{kk} = 1}} d'_{ik} > d'_{ij}
\end{aligned}$$

Note that now, the objective function of the master problem is exactly the same as the one of the MILP formulation (η has been replaced by FB), which implies that η is no longer a decision variable of the master problem. Instead, B is now part of the complicating variables, and is a decision variable of the master problem.

Algorithm 7 runs in cubic time because of the initialization phase (there is a cubic number of β_{ijk} variables for instance). But if we ignore initialization, the running time of Algorithm 7 is simply quadratic. So now, we can argue that the Benders optimality cut can be separated in quadratic time because:

- The last expression of the Benders optimality cut has at most two nested sums, so it requires to add $O(n^2)$ terms.
- What we need to build this optimality cut are the non-zero dual variables only, so we can drop the initialization phase of Algorithm 7, in practice. This is due to the fact that a unique β_{ijk} variable has a non-zero value (same thing for δ_{ijk}), this is the key for “killing” the cubic factor.

By contrast, the separation of the original Benders optimality cut requires making the sum of a cubic number of terms (the fact that the cardinality of \tilde{J} is in $O(n^3)$ is sufficient to show it). So we have an efficient separation algorithm based upon the computation of the critical triplet (in quadratic time), and Algorithm 7 which is quadratic also provided we skip the initialization phase.

3.6 Numerical Experiments

The ILP introduced in Section 3.4 as well as the Branch-and-Benders cut decomposition algorithm in Section 3.5, are implemented with the post-procedure optimization described in Subsection 3.4.1 and tested on a set of TSPLIB instances. Similarly to Section 2.7, these TSPLIB instances are a collection of $32 \times 3 \times 4 = 384$ TSP instances involving between 51 and 200 nodes. Let l_{ij} be the Euclidian distance between nodes i and j of V , we define the ring cost by $c_{ij} = \lceil \alpha l_{ij} \rceil$ and the star cost by $d_{ij} = \lceil (10 - \alpha) l_{ij} \rceil$ where α is in the set $\{3, 5, 7\}$, for all (i, j) in V^2 . c'_{ij}

is set to $0.01c_{ij}$ and is a cost per day in a one-year time horizon. d'_{ij} is set to $0.01d_{ij}$ and is also a cost per day in a one-year horizon. Opening costs are zero for these instances. F is expressed in days, set to 4 values, $F = 0$ which means there is no failures, $F = 7$ which is one week, $F = 31$ (one month) and $F = 183$ (6 months).

We use Julia v1.8.0, JuMP v1.2.1 and Gurobi v9.0.3 to solve all integer and linear programs. The same settings for Gurobi as in Section 2.7 are used. We enforce a time limit of one hour per instance. The computer used to run the numerical experiments is a DELL equipped with a Intel(R) Core(TM) i7-10610U CPU @ 1.80GHz and 16GBytes of RAM, under GNU/Linux. Table 3.1 present the results for the TSPLIB class. Two methods are tested, the ILP mentioned in Section 3.4 and the Branch-and-Benders scheme mentioned in Section 3.5. The columns of this table are as follows: **F** is the aforementioned parameter, it is the total duration of failures in days, **CPU** is the total CPU time in seconds, it includes the Brute Force Algorithm 1 CPU times which can be adapted for Resilient RSP. "TL" is indicated when the time limit of one hour is reached; **Gap** represents the relative optimality gap of both methods. Its formula is the same as in Table 2.2. Both **LB** and **UB** are also displayed in Table 3.1; **|H|** is the number of hubs in the best solution found. Note that since there is a single certain node in the considered instances, Algorithm 1 cannot find any ring with three hubs. **n_subtour** is the number of subtour elimination constraints, *i.e.*, feasibility cuts for Benders, and lazy constraints for ILP. Columns **Nodes** show the total number of explored nodes in the solver's branch-and-bound tree during the search. **CPU SP** is the CPU time in seconds for solving the Benders subproblems. **n_cut** gives the number of optimality cuts in the Benders decomposition, which is also the number of times the subproblem is solved. Each line in Table 3.1 presents an instance with the format "Instance name $n - \alpha$ " where n is the total number of nodes in the instance and α is the aforementioned parameter.

Some instances does not appear in Table 3.1, these are the instances where both ILP and Branch-and-Benders-cut algorithm run out of memory. When "NO SOLUTION FOUND" is printed, that means the algorithm could not find a solution within the one hour time limit. We noticed that ILP finds a valid solution without going out of memory on 64 out of 384 instances while Branch-and-Benders-cut algorithm finds a valid solution without going out of memory on 95 instances out of 384 instances. That means the Branch-and-Benders-cut algorithm is able to find solutions more often than the ILP model. Moreover, considering that an algorithm is better than another when both gaps are 0% and its CPU is lower or both gap or greater than 0% but its gap is lower then, when $F = 0$ (no failures), the Branch-and-Benders-cut algorithm is better on 13 instances while the ILP is better on three instances. When $F = 7$ (one week), the Branch-and-Benders-cut al-

gorithm is better on 31 instances while the *ILP* is better on three instances. When $F = 31$ (one month), the Branch-and-Benders-cut algorithm is better on 23 instances and the *ILP* is better on 6 instances. When $F = 183$ (six months), the Branch-and-Benders-cut algorithm is better on 11 instances and the *ILP* is better on 9 instances. We observe that the easiest instances for the Branch-and-Benders-cut are the ones with lower F , except for $F = 0$ (no failures). Moreover, when both algorithms find an optimal solution, the CPU times tend to be from 2 times faster up to 15 times faster for the Branch-and-Benders-cut algorithm. We observe that among the values of the parameter α , the easiest instances to solve are the ones with $\alpha = 3$ for the Branch-and-Benders-cut algorithm. We also observe that no instances with more than 105 nodes (lin-105) can be solved within the time limit for the Branch-and-Benders-cut algorithm while the *ILP* cannot solve instances bigger than 76 nodes (pr-76).

3.7 Resilient or Survivable Ring Star Problem?

In Chapter 2, we studied a Survivable variant of the Ring Star Problem and in Chapter 3 a Resilient variant of the same problem. In this subsection, we address the question of determining which of the two strategies leads to the smallest cost, as a function of the failures duration.

3.7.1 Properties

It is shown in this section that $g(F)$, the function that associates the optimal objective function value of 1-R-RSP(F) as a function of F , is continuous piecewise linear and concave.

For all $F \geq 0$, we denote by $\sigma(F)$ the set of all the optimal solutions to 1-R-RSP(F). The objective value of any feasible solution \mathcal{S} to 1-R-RSP(F) can be written as $K^{\mathcal{S}} + FB^{\mathcal{S}}$, where $K^{\mathcal{S}} = \sum_{ij \in E'} c_{ij}x_{ij} + \sum_{(i,j) \in A^{\neq}} d_{ij}y_{ij} + \sum_{i \in V} d_{ii}y_{ii}$ and $B^{\mathcal{S}} = B$ are nonnegative constants. For all $F_0 \geq 0$, an optimal solution \mathcal{S}_{F_0} to 1-R-RSP(F_0) is said to be B -optimal for 1-R-RSP(F_0) if it is such that $B^{\mathcal{S}_{F_0}} = \min_{\mathcal{S} \in \sigma(F_0)} B^{\mathcal{S}}$.

Observation 1: If solution \mathcal{S}_{F_0} is B -optimal for 1-R-RSP(F_0), then for all $\mathcal{S} \in \sigma(F_0)$, we have $K^{\mathcal{S}_{F_0}} + F_0B^{\mathcal{S}_{F_0}} = K^{\mathcal{S}} + F_0B^{\mathcal{S}}$. By definition of a B -optimal solution, we have $B^{\mathcal{S}_{F_0}} \leq B^{\mathcal{S}}$ for all $\mathcal{S} \in \sigma(F_0)$. So adding $B^{\mathcal{S}_{F_0}}(F - F_0) \leq B^{\mathcal{S}}(F - F_0)$ to the last equality yields $K^{\mathcal{S}_{F_0}} + FB^{\mathcal{S}_{F_0}} \leq K^{\mathcal{S}} + FB^{\mathcal{S}}$ for all $F \geq F_0$.

Lemma: For all $F_0 \geq 0$, there exists $\varepsilon > 0$ such that any B -optimal solution \mathcal{S}_{F_0} to

1-R-RSP(F_0) is an optimal solution to 1-R-RSP(F) for all $F \in [F_0, F_0 + \varepsilon]$.

Proof: If there does not exist any optimal solution \mathcal{S}' to 1-R-RSP(F) such that $K^{\mathcal{S}'} + FB^{\mathcal{S}'} < K^{\mathcal{S}_{F_0}} + FB^{\mathcal{S}_{F_0}}$ for any $F \geq F_0$, the lemma obviously holds. Hence, we now assume that there exists $F > F_0$ such that an optimal solution \mathcal{S}' to 1-R-RSP(F) satisfies $K^{\mathcal{S}'} + FB^{\mathcal{S}'} < K^{\mathcal{S}_{F_0}} + FB^{\mathcal{S}_{F_0}}$, this inequality is denoted by (A). Note that \mathcal{S}' cannot be optimal for 1-R-RSP(F_0) because no optimal solution \mathcal{S} to 1-R-RSP(F_0) can satisfy (A), by Observation 1. Consequently we have $K^{\mathcal{S}'} + F_0B^{\mathcal{S}'} > K^{\mathcal{S}_{F_0}} + F_0B^{\mathcal{S}_{F_0}}$ which can also be written as $K^{\mathcal{S}'} - K^{\mathcal{S}_{F_0}} > F_0(B^{\mathcal{S}_{F_0}} - B^{\mathcal{S}'})$. Using (A) we have $B^{\mathcal{S}_{F_0}} - B^{\mathcal{S}'} > \frac{1}{F}(K^{\mathcal{S}'} - K^{\mathcal{S}_{F_0}})$, hence $K^{\mathcal{S}'} - K^{\mathcal{S}_{F_0}} > \frac{F_0}{F}(K^{\mathcal{S}'} - K^{\mathcal{S}_{F_0}})$, so $(K^{\mathcal{S}'} - K^{\mathcal{S}_{F_0}})(F - F_0) > 0$. Since $F > F_0$, this implies $K^{\mathcal{S}'} > K^{\mathcal{S}_{F_0}}$. Now, using this inequality and (A), it follows that $B^{\mathcal{S}_{F_0}} > B^{\mathcal{S}'}$.

Hence we have $K^{\mathcal{S}_{F_0}} + (F_0 + \varepsilon)B^{\mathcal{S}_{F_0}} \leq K^{\mathcal{S}'} + (F_0 + \varepsilon)B^{\mathcal{S}'}$ for all $\varepsilon \leq \frac{K^{\mathcal{S}'} - K^{\mathcal{S}_{F_0}}}{B^{\mathcal{S}_{F_0}} - B^{\mathcal{S}'}}$, where $\frac{K^{\mathcal{S}'} - K^{\mathcal{S}_{F_0}}}{B^{\mathcal{S}_{F_0}} - B^{\mathcal{S}'}} > 0$. ■

Corollary: Let $F_1 > F_0$ be the largest value for which \mathcal{S}_{F_0} , a B -optimal solution to 1-R-RSP(F_0), is an optimal solution to 1-R-RSP(F_1). Since \mathcal{S}_{F_0} is no longer an optimal solution to 1-R-RSP(F) for $F > F_1$, it can be concluded by the previous lemma that it is not a B -optimal solution to 1-R-RSP(F_1). This means that any B -optimal solution to 1-R-RSP(F_1), denoted by \mathcal{S}_{F_1} is such that $B^{\mathcal{S}_{F_1}} < B^{\mathcal{S}_{F_0}}$ and $K^{\mathcal{S}_{F_0}} + F_1B^{\mathcal{S}_{F_0}} = K^{\mathcal{S}_{F_1}} + F_1B^{\mathcal{S}_{F_1}}$. This shows that $g(F)$, the function that associates the optimal objective function value of 1-R-RSP(F) as a function of F , is continuous, piecewise linear, and concave.

Consequently, $g(F)$ can be specified by an ordered sequence of q pairs $(F_0, \mathcal{S}_0), (F_1, \mathcal{S}_1), \dots, (F_{q-1}, \mathcal{S}_{q-1})$, where $F_0 = 0$, and $F_{k-1} < F_k$ for all $k \in \{1, \dots, q-1\}$. For all $k \in \{0, \dots, q-1\}$, solution \mathcal{S}_k is a B -optimal solution to 1-R-RSP(F_k) whose objective value is $K^k + F_kB^k$. An illustration of $g(F)$ when $q = 4$ is given in Figure 3.2. For all $k \in \{0, \dots, q-1\}$, solution \mathcal{S}_k is optimal for 1-R-RSP(F) for all $F \in [F_k, F_{k+1}]$. The slope of line segment k of $g(F)$ is B^k , and the corresponding value of K^k can be visualized at the intersection of the line segment with the line $F = 0$. Figure 3.2 also shows the interval upon which \mathcal{S}_k is optimal.

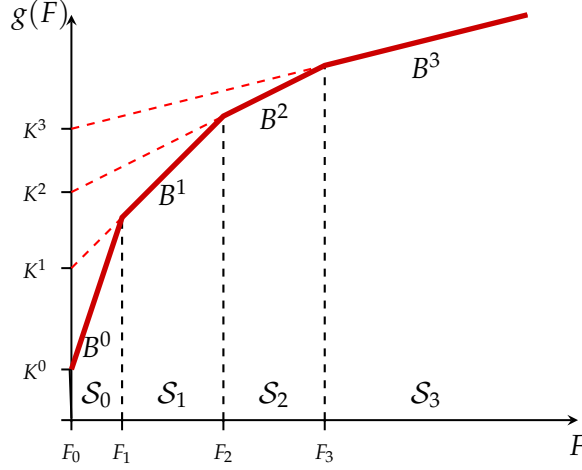


Figure 3.2: Illustration of the shape of $g(F)$ with $q = 4$

Property 3: Let $F_0 < F_1$. Let S_0 (resp. S_1) be an optimal solution to 1-R-RSP(F_0) (resp. 1-R-RSP(F_1)). If $B^{S_0} = B^{S_1}$, then S_0 is an optimal solution to 1-R-RSP(F) for all $F \in [F_0, F_1]$.

Proof: Property 3 is the consequence of the fact that $g(F)$ is concave. Indeed, for all $F \in (F_0, F_1)$, the optimal solution S_F to 1-R-RSP(F) is such that $g(F) = K^{S_F} + FB^{S_F}$, with $B^{S_0} \leq B^{S_F} \leq B^{S_1}$. Since $B^{S_0} = B^{S_1}$, we have $B^{S_F} = B^{S_0}$ for all $F \in [F_0, F_1]$, hence $S_F = S_0$.

3.7.2 Computational method to solve 1-R-RSP(F) on an interval

We wish to find an optimal solution to 1-R-RSP(F) for all $F \in [F_\ell, F_r]$. But we wish to avoid determining B -optimal solutions to 1-R-RSP, because this may be costly. In the sequel, if S is a solution to 1-R-RSP(F), $S.B$ denotes the associated value of B , and $S.K$ denotes K^S .

- Solve 1-R-RSP(F_ℓ), and obtain solution S_ℓ (so $B^{S_\ell} \leftarrow S_\ell.B$, and $K^{S_\ell} \leftarrow g(F_\ell) - F_\ell B^{S_\ell} = S_\ell.K$ are known)
- Solve 1-R-RSP(F_r), and obtain solution S_r (so $B^{S_r} \leftarrow S_r.B$, and $K^{S_r} \leftarrow g(F_r) - F_r B^{S_r} = S_r.K$ are known)
- Store (F_ℓ, S_ℓ)
- Store (F_r, S_r)
- Explore the interval $[F_\ell, F_r]$ by calling EXPLORE(F_ℓ, S_ℓ, F_r, S_r). This function recursively splits the interval to determine the different line segments of

$g(F)$

- Sort the stored pairs (F, \mathcal{S}) by increasing order of F . If we find two consecutive pairs with the same value of F , delete the one with the largest $B^{\mathcal{S}}$ value. If two consecutive pairs have the same B value, the pair associated with the largest F value should be deleted.

The pseudocode of $\text{EXPLORE}(F_\ell, \mathcal{S}_\ell, F_r, \mathcal{S}_r)$ is given in Algorithm 8, and the different situations that can occur in the course of the algorithm are illustrated by Figures 3.3 and 3.4. In both these figures, the objective value of solution \mathcal{S}_ℓ as a function of F , *i.e.*, $K^{\mathcal{S}_\ell} + FB^{\mathcal{S}_\ell}$ is show in blue, and the objective value of \mathcal{S}_r as a function of F is plotted in red.

Algorithm 8: $\text{EXPLORE}(F_\ell, \mathcal{S}_\ell, F_r, \mathcal{S}_r)$

```

1  $K^{\mathcal{S}_\ell} \leftarrow \mathcal{S}_\ell.K$ 
2  $B^{\mathcal{S}_\ell} \leftarrow \mathcal{S}_\ell.B$ 
3  $K^{\mathcal{S}_r} \leftarrow \mathcal{S}_r.K$ 
4  $B^{\mathcal{S}_r} \leftarrow \mathcal{S}_r.B$ 
5 if  $B^{\mathcal{S}_\ell} = B^{\mathcal{S}_r}$  then
6   return
7  $F_m \leftarrow \frac{K^{\mathcal{S}_r} - K^{\mathcal{S}_\ell}}{B^{\mathcal{S}_\ell} - B^{\mathcal{S}_r}}$ 
8 if  $F_m = F_\ell$  then
9   Store  $(F_\ell, \mathcal{S}_r)$ 
10  return
11 if  $F_m = F_r$  then
12   Store  $(F_\ell, \mathcal{S}_\ell)$ 
13  return
14  $\mathcal{S}_m \leftarrow \text{1-R-RSP}(F_m)$ 
15 if  $K^{\mathcal{S}_\ell} + F_m B^{\mathcal{S}_\ell} = K^{\mathcal{S}_m} + F_m B^{\mathcal{S}_m}$  then
16   Store  $(F_m, \mathcal{S}_r)$ 
17  return
18  $\text{EXPLORE}(F_\ell, \mathcal{S}_\ell, F_m, \mathcal{S}_m)$ 
19  $\text{EXPLORE}(F_m, \mathcal{S}_m, F_r, \mathcal{S}_r)$ 

```

Prior to each call to $\text{EXPLORE}(F_\ell, \mathcal{S}_\ell, F_r, \mathcal{S}_r)$, solution \mathcal{S}_ℓ is optimal for $\text{1-R-RSP}(F_\ell)$ and \mathcal{S}_r is optimal for $\text{1-R-RSP}(F_r)$. If $B^{\mathcal{S}_\ell} = B^{\mathcal{S}_r}$, 3 ensures that solution \mathcal{S}_ℓ (or \mathcal{S}_r) is an optimal solution to $\text{1-R-RSP}(F)$ for all $F \in [F_\ell, F_r]$, this situation is pictured in Figure 3.3, left, where the red line and the blue one overlap. Otherwise, $B^{\mathcal{S}_\ell} < B^{\mathcal{S}_r}$ and we compute F_m as the value of F for which the cost of solutions \mathcal{S}_ℓ and \mathcal{S}_r are equal, *i.e.*, $K^{\mathcal{S}_\ell} + F_m B^{\mathcal{S}_\ell} = K^{\mathcal{S}_r} + F_m B^{\mathcal{S}_r}$. If $F_m = F_\ell$ (see Figure 3.3, center), then solution \mathcal{S}_r is an optimal solution to $\text{1-R-RSP}(F)$ for all $F \in [F_\ell, F_r]$, and the pair (F_ℓ, \mathcal{S}_r) is stored. Symmetrically, if $F_m = F_r$ (see Figure 3.3, right), then solution \mathcal{S}_ℓ is an optimal solution to $\text{1-R-RSP}(F)$ for all $F \in [F_\ell, F_r]$

and the pair $(F_\ell, \mathcal{S}_\ell)$ is stored. Now, we assume that F_m is in (F_ℓ, F_r) , and we compute \mathcal{S}_m , an optimal solution to $1\text{-R-RSP}(F_m)$. If $K^{\mathcal{S}_\ell} + F_m B^{\mathcal{S}_\ell} = K^{\mathcal{S}_m} + F_m B^{\mathcal{S}_m}$ (see Figure 3.4, left), then we know that solution \mathcal{S}_ℓ is an optimal solution to $1\text{-R-RSP}(F)$ for all $F \in [F_\ell, F_m]$ by Property 3, hence we store the pair (F_m, \mathcal{S}_r) . If $K^{\mathcal{S}_\ell} + F_m B^{\mathcal{S}_\ell} > K^{\mathcal{S}_m} + F_m B^{\mathcal{S}_m}$ (see Figure 3.4, right), then solution \mathcal{S}_m has a slope $B^{\mathcal{S}_m} \in (B^{\mathcal{S}_\ell}, B^{\mathcal{S}_r})$, so the interval $[F_\ell, F_r]$ is halved, and the exploration is launched recursively on the intervals $[F_\ell, F_m]$ and $[F_m, F_r]$.

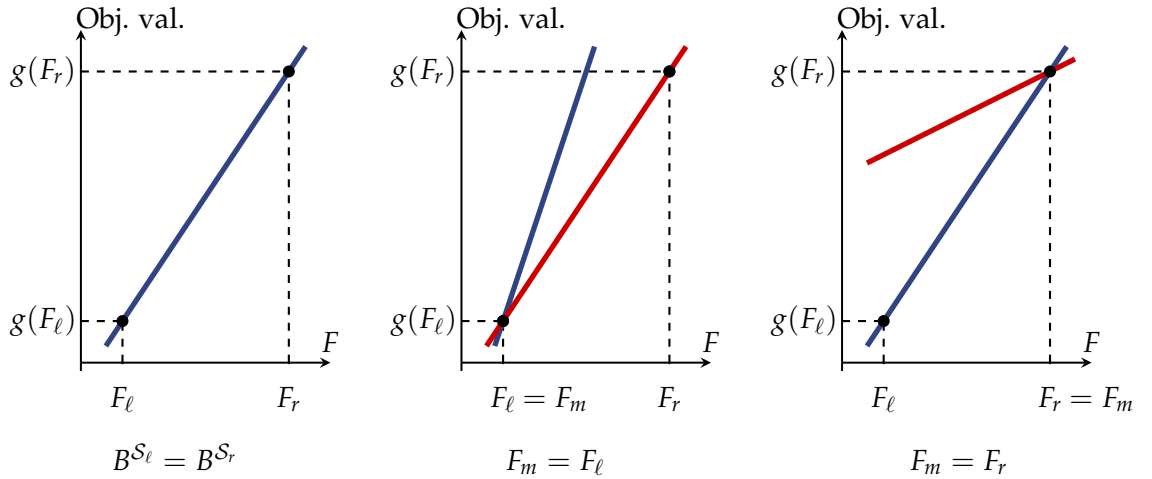


Figure 3.3: Illustration of the situations that can occur when $\text{EX-PLORE}(F_\ell, \mathcal{S}_\ell, F_r, \mathcal{S}_r)$ is run

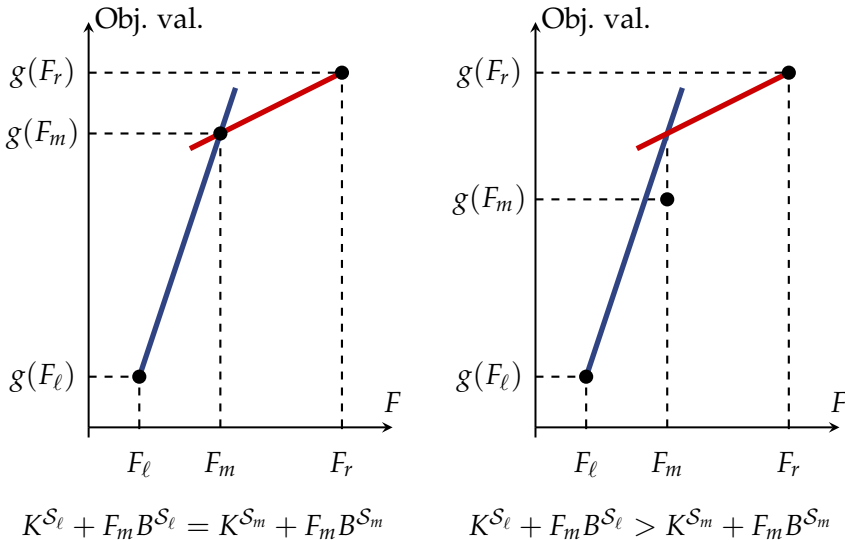


Figure 3.4: Illustration of the situations that can occur when $F_m \in (F_\ell, F_r)$ in $\text{EX-PLORE}(F_\ell, \mathcal{S}_\ell, F_r, \mathcal{S}_r)$

3.7.3 Solving 1-R-RSP(F) for all $F \geq F_\ell$

Let F_ℓ be a given nonnegative constant, and suppose that we want to solve 1-R-RSP(F) for all $F \geq F_\ell$. Note that such a problem may not be very relevant in practice, as we may assume that an upper bound for the total time during which a hub is down may not exceed the time horizon. We cannot apply the previous approach to address this problem, because $[F_\ell, +\infty)$ is not a closed interval. However, we may circumvent the issue by considering the solution \mathcal{S}_∞ , defined as follows:

- $B^{\mathcal{S}_\infty} = \min_{\mathcal{S} \in \sigma} B^{\mathcal{S}}$
- $K^{\mathcal{S}_\infty} = \min_{\substack{\mathcal{S} \in \sigma \\ B^{\mathcal{S}} = B^{\mathcal{S}_\infty}}} K^{\mathcal{S}}$

Where σ is the set of all the feasible solutions to 1-R-RSP(F) for some $F \geq 0$. By the corollary, $g(F)$ is continuous, linear piecewise and concave, so it follows that there exists $F_0 \geq 0$ such that the objective value of solution \mathcal{S}_∞ is less than or equal to the solution value to any feasible solution to 1-R-RSP(F), for all $F \geq F_0$. Solution \mathcal{S}_∞ can be computed by solving the following two successive variants of 1-R-RSP, in which there is no F input parameter:

- Obtain $B^{\mathcal{S}_\infty}$ by solving a variant of 1-R-RSP where the objective function is to minimize B
- Obtain $K^{\mathcal{S}_\infty}$ by solving a second variant of 1-R-RSP where the objective is to minimize $\sum_{ij \in E'} c_{ij}x_{ij} + \sum_{(i,j) \in A^\#} d_{ij}y_{ij} + \sum_{i \in V} d_{ii}y_{ii}$, and by enforcing the constraint $B = B^{\mathcal{S}_\infty}$.

It can be observed that if $|V \setminus \tilde{V}| \geq 3$, then \mathcal{S}_∞ is such that $B^{\mathcal{S}_\infty} = 0$. Indeed, any solution where the ring is made of certain hubs only is insensitive to failures. In such a case, there is no need to address the first variant of 1-R-RSP, since we already know that $B^{\mathcal{S}_\infty}$ is zero.

We can solve 1-R-RSP(F) for all $F \geq F_\ell$ as follows:

- Solve 1-R-RSP(F_ℓ), and obtain solution \mathcal{S}_ℓ (so $K^{\mathcal{S}_\ell}$ and $B^{\mathcal{S}_\ell}$ are known)
- Compute solution \mathcal{S}_∞ as shown above
- Store $(F_\ell, \mathcal{S}_\ell)$
- Explore the half-open interval $[F_\ell, +\infty)$ by calling EXPLORE+($F_\ell, K^{\mathcal{S}_\ell}, B^{\mathcal{S}_\ell}, K^{\mathcal{S}_\infty}, B^{\mathcal{S}_\infty}$)

The pseudocode of EXPLORE+($F_\ell, K^{\mathcal{S}_\ell}, B^{\mathcal{S}_\ell}, K^{\mathcal{S}_\infty}, B^{\mathcal{S}_\infty}$) is given in Algorithm 9

Algorithm 9: EXPLORE+($F_\ell, K^{\mathcal{S}_\ell}, B^{\mathcal{S}_\ell}, K^{\mathcal{S}_\infty}, B^{\mathcal{S}_\infty}$)

```

1 if  $B^{\mathcal{S}_\ell} = B^{\mathcal{S}_\infty}$  then
2   return
3  $F_m \leftarrow \frac{K^{\mathcal{S}_\infty} - K^{\mathcal{S}_\ell}}{B^{\mathcal{S}_\ell} - B^{\mathcal{S}_\infty}}$ 
4 if  $F_m = F_\ell$  then
5   Store ( $F_\ell, \mathcal{S}_\infty$ );
6   return
7  $\mathcal{S}_m \leftarrow 1\text{-R-RSP}(F_m)$ ;
8 if  $K^{\mathcal{S}_\ell} + F_m B^{\mathcal{S}_\ell} = K^{\mathcal{S}_m} + F_m B^{\mathcal{S}_m}$  then
9   Store ( $F_m, \mathcal{S}_\infty$ );
10  return
11 EXPLORE( $F_\ell, K^{\mathcal{S}_\ell}, B^{\mathcal{S}_\ell}, F_m, K^{\mathcal{S}_m}, B^{\mathcal{S}_m}$ )
12 EXPLORE+( $F_m, K^{\mathcal{S}_m}, B^{\mathcal{S}_m}, K^{\mathcal{S}_\infty}, B^{\mathcal{S}_\infty}$ )

```

3.7.4 Comparing the cost of 1-S-RSP and 1-R-RSP(F)

Now, we can compare an optimal solution of problem 1-S-RSP with the optimal solutions of 1-R-RSP(F) for F in some interval using a graphical approach. It suffices to plot $g(F)$ in the considered interval, as well as the horizontal line corresponding to the cost of an optimal solution to 1-S-RSP (because this cost is constant, and does not depend on F). There are two possibilities:

- $g(F)$ and the horizontal line intersect. In that case, the intersection occurs at F^* , and the solutions to 1-R-RSP(F) are less expensive for all $F \leq F^*$, whereas an optimal solution to 1-S-RSP is preferable for all $F \geq F^*$. This situation is illustrated in Figure 3.5, where the optimal objective value of 1-S-RSP is denoted by $f_{1\text{-S-RSP}}$.
- $g(F)$ and the horizontal line do not intersect. This can happen either if the values considered for F are too low, in which case no optimal solution to 1-S-RSP is competitive compared to optimal solutions to 1-R-RSP(F), or if the considered values for F are too large, in which case no optimal solution to 1-R-RSP(F) is competitive compared to an optimal solution to 1-S-RSP.

3.7.5 Numerical experiments to choose between Survivable or Resilient RSP

Some numerical experiments are reported in Table 3.2. We selected two instances with 10 and 12 nodes and displayed the CPU times for Resilient RSP and Survivable RSP solved by the Branch-and-Benders-cut approach, $|S_i|$ the number of optimal solutions to 1-R-RSP(F), the F values and F^* . We can see in this table that for the instance of 10 nodes, the user should choose the Resilient RSP below

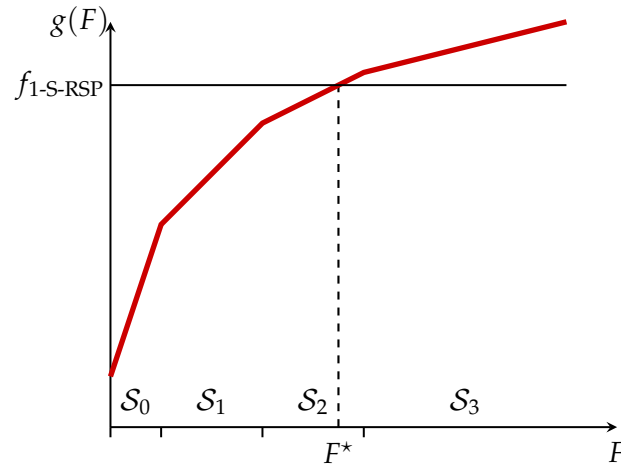


Figure 3.5: Comparing the optimal solutions of 1-R-RSP(F) and 1-S-RSP

$F^* = 569.76$ and Survivable RSP above. We see in Table 3.2 that the number of optimal solutions $|S_i|$ is equal to the number of F values. This means that the proposed approach is efficient in the sense that we solve the minimum number of instances of 1-R-RSP(F). Figures 3.6, 3.7, 3.8, 3.9 and 3.10 show the 5 solutions $S_i, i \in \{1, \dots, 5\}$ for the instance of 10 nodes. The critical hub which incurs the worst-case repairing cost among each hub failure of the ring is represented in green as well as its corresponding backup edge and the decision variable B . As expected, solution S_1 has a low cost when $F = 0$, but this cost increases significantly with F . Symmetrically, solution S_5 has a large cost, but it increases at a slow pace when F increases.

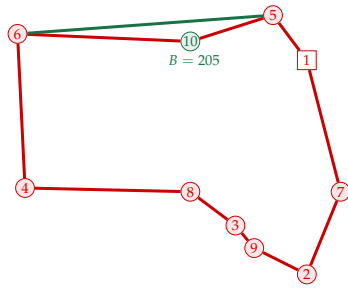


Figure 3.6: S_1 , a 10-node solution of the 1-R-RSP(F) with $\tilde{V} = V \setminus \{1\}$, $F = 0$ (no failures) and $g(F) = 908$

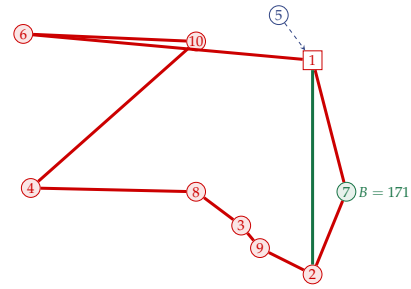


Figure 3.7: S_2 , a 10-node solution of the 1-R-RSP(F) with $\tilde{V} = V \setminus \{1\}$, $F = 779.4$ and $g(F) = 2505.8$

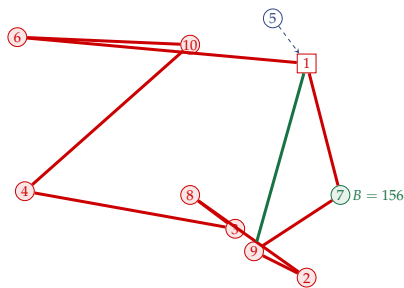


Figure 3.8: S_3 , a 10-node solution of the 1-R-RSP(F) with $\tilde{V} = V \setminus \{1\}$, $F = 933.3$ and $g(F) = 2769$

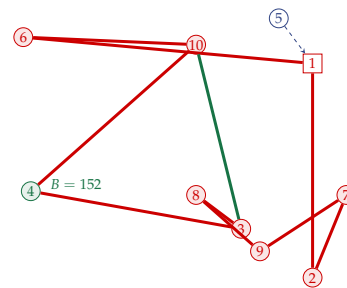


Figure 3.9: S_4 , a 10-node solution of the 1-R-RSP(F) with $\tilde{V} = V \setminus \{1\}$, $F = 1025$ and $g(F) = 2912$

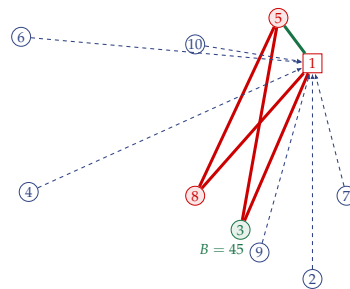


Figure 3.10: S_5 , a 10-node solution of the 1-R-RSP(F) with $\tilde{V} = V \setminus \{1\}$, $F = 1544.9$ and $g(F) = 3702.2$

Concluding remarks about this chapter are given in Chapter 4.

Table 3.1: TSPLIB instances (continued)

Instance $n-a$	ILP					Branch-and-Benders-cut												
	$n = V $	F	CPU	Gap	LB	UB	$ H $	$n_subtour$	Nodes	CPU	CPU SP	Gap	LB	UB	$ H $	$n_subtour$	n_cut	Nodes
kroA100-3	7 (one week)		OUT OF MEMORY						40.12	1.41	0%	64120.60	64120.60	100	1019	12	1666	
kroA100-3	31 (one month)		OUT OF MEMORY						72.49	2.98	0%	64698.55	64698.55	100	1376	28	1359	
kroA100-5	7 (one week)		OUT OF MEMORY						TL	0.54	NO SOLUTION FOUND	98483.50			8917	4	0	
kroA100-5	31 (one month)		OUT OF MEMORY						TL	0.55	NO SOLUTION FOUND	98482.00			8947	4	0	
kroA100-5	183 (six months)		OUT OF MEMORY						TL	0.34	NO SOLUTION FOUND	98520.50			9110	2	0	
kroA100-7	7 (one week)		OUT OF MEMORY						TL	0.33	NO SOLUTION FOUND	98676.11			7820	2	0	
kroA100-7	31 (one month)		OUT OF MEMORY						TL	0.33	NO SOLUTION FOUND	98588.38			7889	2	0	
kroA100-7	183 (six months)		OUT OF MEMORY						TL	0.32	NO SOLUTION FOUND	98676.58			7802	2	0	
kroB100-3	7 (one week)		OUT OF MEMORY						36.16	1.27	0%	66634.35	66634.35	100	530	10	1099	
kroB100-3	31 (one month)		OUT OF MEMORY						52.19	2.95	0%	67168.76	67168.76	100	757	26	1353	
kroB100-5	31 (one month)		OUT OF MEMORY						TL	1.17	NO SOLUTION FOUND	104452.00			9078	11	0	
kroB100-5	183 (six months)		OUT OF MEMORY						TL	1.25	NO SOLUTION FOUND	104399.90			9436	12	0	
kroB100-7	7 (one week)		OUT OF MEMORY						146.54	2.24	0%	118718.67	118718.67	47	1528	20	1379	
kroC100-3	7 (one week)		OUT OF MEMORY						22.69	1.06	0%	62465.97	62465.97	100	163	6	629	
kroC100-7	7 (one week)		OUT OF MEMORY						TL	0.34	NO SOLUTION FOUND	95471.74			8096	2	0	
kroC100-7	31 (one month)		OUT OF MEMORY						TL	0.33	NO SOLUTION FOUND	95440.01			7783	2	0	
kroC100-7	183 (six months)		OUT OF MEMORY						TL	0.34	NO SOLUTION FOUND	95440.01			7783	2	0	
kroD100-3	7 (one week)		OUT OF MEMORY						18.37	1.17	0%	64128.11	64128.11	100	119	9	172	
kroD100-3	31 (one month)		OUT OF MEMORY						25.85	2.62	0%	64713.60	64713.60	100	148	23	372	
kroD100-5	7 (one week)		OUT OF MEMORY						TL	0.33	NO SOLUTION FOUND	95806.38			8434	2	0	
kroD100-5	31 (one month)		OUT OF MEMORY						TL	0.38	NO SOLUTION FOUND	95791.67			8065	2	0	
kroD100-5	183 (six months)		OUT OF MEMORY						TL	0.34	NO SOLUTION FOUND	95794.00			8185	2	0	
kroD100-7	7 (one week)		OUT OF MEMORY						182.22	4.22	0%	117571.65	117571.65	48	1513	43	1607	
kroD100-7	31 (one month)		OUT OF MEMORY						2164.30	32.31	0%	119490.21	119490.21	53	2013	301	167663	
kroE100-3	7 (one week)		OUT OF MEMORY						53.12	1.34	0%	66437.39	66437.39	100	1264	11	1222	
kroE100-3	31 (one month)		OUT OF MEMORY						77.78	2.80	0%	67055.87	67055.87	100	1340	29	1807	
kroE100-5	183 (six months)		OUT OF MEMORY						TL	0.33	NO SOLUTION FOUND	102621.00			9126	2	0	
kroE100-7	7 (one week)		OUT OF MEMORY						94.10	2.00	0%	117103.25	117103.25	51	1708	18	1523	
kroE100-7	31 (one month)		OUT OF MEMORY						200.64	5.64	0%	119113.25	119113.25	51	1970	58	2998	
rd100-3	7 (one week)		OUT OF MEMORY						17.04	0.97	0%	23851.97	23851.97	100	127	8	85	
rd100-3	31 (one month)		OUT OF MEMORY						23.52	1.47	0%	24083.12	24083.12	100	243	11	253	
rd100-5	7 (one week)		OUT OF MEMORY						TL	0.38	NO SOLUTION FOUND	35610.00			7945	2	0	
rd100-5	31 (one month)		OUT OF MEMORY						TL	0.33	NO SOLUTION FOUND	35597.00			7826	2	0	
rd100-5	183 (six months)		OUT OF MEMORY						TL	0.41	NO SOLUTION FOUND	35599.00			7962	2	0	
rd100-7	7 (one week)		OUT OF MEMORY						168.53	2.59	0%	41213.96	41213.96	46	1521	24	1578	
rd100-7	31 (one month)		OUT OF MEMORY						863.01	18.82	0%	41832.19	41832.19	47	2040	183	32859	
e11101-3	7 (one week)		OUT OF MEMORY						13.25	0.49	0%	1968.55	1968.55	101	56	4	69	
e11101-3	31 (one month)		OUT OF MEMORY						18.18	1.44	0%	1984.15	1984.15	101	95	11	210	
e11101-3	183 (six months)		OUT OF MEMORY						432.02	20.39	0%	2082.95	2082.95	101	1709	169	15437	
e11101-5	7 (one week)		OUT OF MEMORY						28.27	1.06	0%	3039.23	3039.23	78	653	9	1033	
lin105-3	183 (six months)		OUT OF MEMORY						271.74	37.64	0%	46943.99	46943.99	105	2150	360	33070	
lin105-5	7 (one week)		OUT OF MEMORY						TL	0.36	NO SOLUTION FOUND	65083.67			8670	2	0	
lin105-5	31 (one month)		OUT OF MEMORY						TL	0.35	NO SOLUTION FOUND	65065.86			7928	2	0	
lin105-5	183 (six months)		OUT OF MEMORY						TL	0.35	NO SOLUTION FOUND	65077.50			8385	2	0	
lin105-7	7 (one week)		OUT OF MEMORY						TL	0.38	NO SOLUTION FOUND	73087.75			10716	2	0	
lin105-7	31 (one month)		OUT OF MEMORY						TL	0.36	NO SOLUTION FOUND	73073.70			10233	2	0	
pr107-5	183 (six months)		OUT OF MEMORY						TL	0.40	NO SOLUTION FOUND	143454.00			12456	2	0	
pr107-7	7 (one week)		OUT OF MEMORY						TL	0.50	NO SOLUTION FOUND	224795.14			8004	3	0	
pr107-7	31 (one month)		OUT OF MEMORY						TL	0.69	NO SOLUTION FOUND	224749.38			7611	3	0	
pr107-7	183 (six months)		OUT OF MEMORY						TL	0.50	NO SOLUTION FOUND	224766.64			7696	3	0	
gr120-5	7 (one week)		OUT OF MEMORY						TL	0.34	NO SOLUTION FOUND	7350.78			6693	1	0	
gr120-5	31 (one month)		OUT OF MEMORY						TL	0.33	NO SOLUTION FOUND	7349.00			6305	1	0	
gr120-5	183 (six months)		OUT OF MEMORY						TL	0.34	NO SOLUTION FOUND	7346.33			6202	1	0	
gr120-7	7 (one week)		OUT OF MEMORY						TL	0.57	NO SOLUTION FOUND	7452.32			5051	2	0	
gr120-7	31 (one month)		OUT OF MEMORY						TL	0.63	NO SOLUTION FOUND	7452.59			4469	2	0	
gr120-7	183 (six months)		OUT OF MEMORY						TL	0.57	NO SOLUTION FOUND	7458.37			4725	2	0	
pr124-5	7 (one week)		OUT OF MEMORY						TL	0.58	NO SOLUTION FOUND	271613.50			7142	2	0	
pr124-5	31 (one month)		OUT OF MEMORY						TL	0.58	NO SOLUTION FOUND	271573.62			6817	2	0	
pr124-5	183 (six months)		OUT OF MEMORY						TL	0.57	NO SOLUTION FOUND	271569.50			6684	2	0	
bier127-5	7 (one week)		OUT OF MEMORY						TL	0.69	NO SOLUTION FOUND	527657.56			7718	2	0	
bier127-5	183 (six months)		OUT OF MEMORY						TL	0.68	NO SOLUTION FOUND	527657.56			7720	2	0	
ch130-7	7 (one week)		OUT OF MEMORY						TL	0.95	NO SOLUTION FOUND	26315.37			6475	2	0	
ch130-7	31 (one month)		OUT OF MEMORY						TL	0.98	NO SOLUTION FOUND	26301.28			5981	2	0	
ch130-7	183 (six months)		OUT OF MEMORY						TL	0.91	NO SOLUTION FOUND	26308.19			6212	2	0	
pr136-5	7 (one week)		OUT OF MEMORY						TL	0.69	NO SOLUTION FOUND	449036.00			8565	2	0	
pr136-5	31 (one month)		OUT OF MEMORY						TL	0.96	NO SOLUTION FOUND	449023.17			8499	2	0	
pr136-5	183 (six months)		OUT OF MEMORY						TL	0.98	NO SOLUTION FOUND	449036.00			8563	2	0	

Table 3.1: TSPLIB instances (continued)

Instance $n-a$	ILP							Branch-and-Benders-cut										
	$n = V $	F	CPU	Gap	LB	UB	$ H $	n_{subtour}	Nodes	CPU	CPU SP	Gap	LB	UB	$ H $	n_{subtour}	n_{cut}	Nodes
gr137-3	7 (one week)									TL	0.70	NO SOLUTION FOUND	1109.44			5761	2	0
gr137-3	31 (one month)									TL	0.70	NO SOLUTION FOUND	1103.00			5916	2	0
gr137-3	183 (six months)									TL	0.71	NO SOLUTION FOUND	1103.00			5968	2	0
gr137-5	7 (one week)									TL	0.97	NO SOLUTION FOUND	1831.00			6147	3	0
gr137-5	31 (one month)									TL	0.98	NO SOLUTION FOUND	1830.50			5731	3	0
gr137-5	183 (six months)									TL	0.97	NO SOLUTION FOUND	1789.00			6579	3	0
gr137-7	7 (one week)									TL	0.99	NO SOLUTION FOUND	2256.88			6310	3	0
gr137-7	31 (one month)									TL	0.98	NO SOLUTION FOUND	2324.77			5648	3	0
gr137-7	183 (six months)									TL	0.70	NO SOLUTION FOUND	2309.88			5871	2	0
pr144-3	7 (one week)									TL	2.24	NO SOLUTION FOUND	142777.17			4076	4	0
pr144-5	7 (one week)									TL	1.86	NO SOLUTION FOUND	229849.50			4249	3	0
pr144-5	31 (one month)									TL	1.78	NO SOLUTION FOUND	229849.50			4233	3	0
pr144-5	183 (six months)									TL	1.72	NO SOLUTION FOUND	229834.50			4265	3	0
ch150-5	7 (one week)									TL	1.14	NO SOLUTION FOUND	29902.00			6100	2	0
ch150-5	31 (one month)									TL	1.13	NO SOLUTION FOUND	29891.00			6057	2	0
ch150-5	183 (six months)									TL	1.12	NO SOLUTION FOUND	29906.33			5886	2	0
ch150-7	7 (one week)									TL	1.14	NO SOLUTION FOUND	30741.90			5004	2	0
ch150-7	31 (one month)									TL	1.14	NO SOLUTION FOUND	30749.51			5221	2	0
ch150-7	183 (six months)									TL	1.13	NO SOLUTION FOUND	30729.73			4948	2	0
kroA150-5	7 (one week)									TL	1.36	NO SOLUTION FOUND	116620.00			5403	2	0
kroA150-5	31 (one month)									TL	1.39	NO SOLUTION FOUND	116620.00			5390	2	0
kroA150-5	183 (six months)									TL	1.39	NO SOLUTION FOUND	116620.00			5403	2	0
kroB150-5	7 (one week)									TL	1.37	NO SOLUTION FOUND	115046.87			5125	2	0
kroB150-5	31 (one month)									TL	2.99	NO SOLUTION FOUND	115047.39			5152	5	0
kroB150-5	183 (six months)									TL	2.92	NO SOLUTION FOUND	115049.37			5173	5	0
kroB150-7	7 (one week)									TL	1.40	NO SOLUTION FOUND	116568.17			5733	2	0
kroB150-7	183 (six months)									TL	1.40	NO SOLUTION FOUND	116478.55			5702	2	0
pr152-3	7 (one week)									TL	1.38	NO SOLUTION FOUND	216172.00			4699	2	0
pr152-3	31 (one month)									TL	1.40	NO SOLUTION FOUND	216154.81			4661	2	0
pr152-5	7 (one week)									TL	1.46	NO SOLUTION FOUND	259239.33			6034	2	0
pr152-5	31 (one month)									TL	1.53	NO SOLUTION FOUND	259161.17			5815	2	0
pr152-5	183 (six months)									TL	1.45	NO SOLUTION FOUND	259239.33			6043	2	0
pr152-7	31 (one month)									TL	3.94	NO SOLUTION FOUND	448642.29			3726	6	0
pr152-7	183 (six months)									TL	1.46	NO SOLUTION FOUND	447520.33			4375	2	0
u159-5	7 (one week)									TL	2.25	NO SOLUTION FOUND	197366.14			5333	4	0
u159-5	31 (one month)									TL	3.13	NO SOLUTION FOUND	197335.50			4916	4	0
u159-5	183 (six months)									TL	1.17	NO SOLUTION FOUND	198057.58			5582	2	0
d198-5	31 (one month)									TL	3.94	NO SOLUTION FOUND	69319.57			3965	3	0
d198-5	183 (six months)									TL	3.77	NO SOLUTION FOUND	69346.73			4252	3	0
d198-7	7 (one week)									TL	2.79	NO SOLUTION FOUND	68892.88			4114	2	0
d198-7	31 (one month)									TL	2.72	NO SOLUTION FOUND	68892.88			4117	2	0
d198-7	183 (six months)									TL	2.81	NO SOLUTION FOUND	68892.88			4123	2	0
kroA200-3	7 (one week)									TL	2.84	NO SOLUTION FOUND	83722.12			3323	2	0
kroA200-3	31 (one month)									TL	2.88	NO SOLUTION FOUND	83722.12			3350	2	0
kroA200-3	183 (six months)									TL	2.83	NO SOLUTION FOUND	83722.12			3350	2	0
kroA200-5	7 (one week)									TL	2.84	NO SOLUTION FOUND	127573.50			4522	2	0
kroA200-5	31 (one month)									TL	2.74	NO SOLUTION FOUND	127573.50			4525	2	0
kroA200-5	183 (six months)									TL	2.84	NO SOLUTION FOUND	127573.50			4528	2	0
kroA200-7	7 (one week)									TL	2.94	NO SOLUTION FOUND	130447.11			4836	2	0
kroA200-7	31 (one month)									TL	2.94	NO SOLUTION FOUND	130428.05			4592	2	0
kroA200-7	183 (six months)									TL	3.00	NO SOLUTION FOUND	130287.12			4947	2	0
kroB200-3	7 (one week)									TL	3.89	NO SOLUTION FOUND	87553.25			3180	3	0
kroB200-3	31 (one month)									TL	3.76	NO SOLUTION FOUND	87553.25			3202	3	0
kroB200-3	183 (six months)									TL	3.89	NO SOLUTION FOUND	87553.25			3191	3	0
kroB200-5	7 (one week)									TL	3.91	NO SOLUTION FOUND	127515.63			4212	3	0
kroB200-5	31 (one month)									TL	3.85	NO SOLUTION FOUND	127515.63			4213	3	0
kroB200-5	183 (six months)									TL	3.99	NO SOLUTION FOUND	127515.63			4212	3	0
kroB200-7	7 (one week)									TL	2.98	NO SOLUTION FOUND	130634.83			4205	2	0
kroB200-7	31 (one month)									TL	2.85	NO SOLUTION FOUND	130634.83			4204	2	0
kroB200-7	183 (six months)									TL	2.88	NO SOLUTION FOUND	130598.21			3997	2	0

Nb. nodes	$ \mathcal{S}_i $: Nb. opt. Solutions	F values	1-R-RSP(F) CPU Time	1-S-RSP CPU Time	F^*
10	5	0.0, 779.4, 933.3 1025.0, 1544.9	30.87s	3.28s	569.76
12	6	0.0, 0.96, 164.71 728.78, 1727.27, 2866.66	187.66s	286.96s	574.87

Table 3.2: Numerical experiments for comparing 1-R-RSP and 1-S-RSP

Chapter 4

Conclusions

4.1 Overview

In this thesis, we introduced a survivable and a resilient variant of the Ring Star Problem. The survivable variant called 1-S-RSP reduces to RSP when there are no uncertain nodes and is shown to be much harder to solve when the cardinality of the uncertain nodes grows. We introduced an ILP formulation to address this problem as well as a Branch-and-Benders-cut algorithm. We improved both ILP and Branch-and-Benders-cut and in particular, generalized the 2-opt heuristic from the traveling salesman problem to deal with uncertain hubs. We also introduce an instance transformation to enhance the master problem performance. Extensive numerical experiments have been carried out and the Branch-and-Benders-cut is shown to have a better performance for instances with a large number of nodes. Both methods have been tested on instances up to 200 nodes. The second problem we introduced, Resilient RSP, was modeled with an ILP and a Branch-and-Benders-cut model similarly to the first problem, the performance of the Branch-and-Benders-cut is shown to be much better than the ILP, solving instances up to 105 nodes. The practical question of determining which approach to use as a function of the duration of the failures has also been addressed. RSP has been studied since 1998 and exact and heuristics methods have been developed for this problem. Resilience and Survivability are concepts studied in many fields including combinatorial optimization and in this thesis, we filled the gap in the RSP literature by the study of the resilient and the survivable variants.

4.2 Limitations

For the Resilient variant of RSP, the Branch-and-Benders-cut scheme and the ILP can be improved. The Branch-and-Benders-cut scheme does not have an instance

transformation for its master problem and both models do not take advantage of a 2-opt heuristic adapted method. As a result, instances of more than 105 nodes are not solved by the Branch-and-Benders-cut scheme. Extensive numerical experiments should be carried out to deepen the analysis of these early results. Regarding the solution of 1-R-RSP(F) for all F , it is applicable only to small instances for which 1-R-RSP(F) can be solved to optimality in a reasonable amount of time. In addition, even if we know that the number of linear pieces in $g(F)$ is finite (because the number of distinct solutions to RSP is finite), determining the number of linear pieces of $g(F)$ is still an open question.

4.3 Future works

The following future work directions may be considered promising. First, any feasible solution found by ILP may be attempted for improvement by solving the subproblem of the Branch-and-Benders-cut algorithm. Moreover, other existing heuristic approaches may be used to try to improve the ring by changing some hubs, not just the edges that join them. In the same vein, existing heuristics or metaheuristics could be extended to 1-S-RSP and used to warm start ILP and the Branch-and-Benders-cut model. We could also develop heuristics for large instances. A more challenging research direction consists in searching for a fast way to generate optimality cuts before solving the Branch-and-Benders-cut model. Such valid inequalities may be obtained from feasible solutions to 1-S-RSP, but not all of them are useful. Another promising future work direction would be to decompose the subproblem of 1-S-RSP to $|V \setminus H| + 1$ subproblems and solve the dual for each of them. That would lead to $|V \setminus H| + 1$ optimality cuts that could be of better quality than the single optimality cut added in the Branch-and-Benders-cut algorithm presented in this thesis. Second, it may be relevant to adapt the st-chains formulation introduced in [32] for RSP to 1-S-RSP. This st-chains formulation has been adapted for 1-R-RSP in this thesis but the blossom inequalities have not been exploited. When we cannot afford to solve 1-R-RSP(F) many times, how can we estimate F^* ? A research direction may be to store and transfer all the subtour constraints (for the ILP model) and all the optimality cuts (for the Branch-and-Benders-cut scheme) to the next 1-R-RSP solution. We could also use a 1-R-RSP solution for the closest values of F to warm-start the solving process.

As shown in Table 2.2, the extra cost required to offer survivability is very large, which means that the proposed solution is adapted to situations where the failure of a hub is frequent. If it is not the case, it may be more relevant to favor solutions that are easy or inexpensive to fix. Finally, a natural research direction is to consider the situation where more than one hub can fail in both variants. For

the case of 1-S-RSP we would consider $k \in \mathbb{N}$ failing hubs and study k -S-RSP.

List of Figures

1.1	Ring Star network	16
1.2	2-edge connected star, dual homed network [31]	18
1.3	Classical Benders decomposition scheme	23
2.1	An 11-node solution of the 1-S-RSP with $\tilde{V} = V \setminus \{1, 11\}$	27
2.2	1-S-RSP instance with $\tilde{V} = \{2, 3, 4\}$ where the dashed backup edge is	28
2.3	Nonzero variables for a solution to an 11-node instance of 1-S-RSP	29
2.4	Branch-and-Benders-cut decomposition scheme	33
2.5	Illustration of edges before (left) and after (right) a 2-opt backup move	50
2.6	4 optimal RSP solutions with the same instance except the ring and star costs for 4 values of input parameter α	53
3.1	The nonzero x, x', y and y' variables of a solution to 1-R-RSP	67
3.2	Illustration of the shape of $g(F)$ with $q = 4$	80
3.3	Illustration of the situations that can occur when $\text{EXPLORE}(F_\ell, \mathcal{S}_\ell, F_r, \mathcal{S}_r)$ is run	82
3.4	Illustration of the situations that can occur when $F_m \in (F_\ell, F_r)$ in $\text{EXPLORE}(F_\ell, \mathcal{S}_\ell, F_r, \mathcal{S}_r)$	82
3.5	Comparing the optimal solutions of 1-R-RSP(F) and 1-S-RSP	85
3.6	S_1 , a 10-node solution of the 1-R-RSP(F) with $\tilde{V} = V \setminus \{1\}$, $F = 0$ (no failures) and $g(F) = 908$	86
3.7	S_2 , a 10-node solution of the 1-R-RSP(F) with $\tilde{V} = V \setminus \{1\}$, $F = 779.4$ and $g(F) = 2505.8$	86
3.8	S_3 , a 10-node solution of the 1-R-RSP(F) with $\tilde{V} = V \setminus \{1\}$, $F = 933.3$ and $g(F) = 2769$	86
3.9	S_4 , a 10-node solution of the 1-R-RSP(F) with $\tilde{V} = V \setminus \{1\}$, $F = 1025$ and $g(F) = 2912$	86
3.10	S_5 , a 10-node solution of the 1-R-RSP(F) with $\tilde{V} = V \setminus \{1\}$, $F = 1544.9$ and $g(F) = 3702.2$	86

List of Tables

2.1	Table of Notations	27
2.2	Impact of \tilde{V} 's cardinality	56
2.3	TSPLIB instances Class I	57
2.3	TSPLIB instances Class I (Continued)	58
2.4	Randomly generated instances Class II	59
2.5	Variation of the results when 2-opt backup is not used	59
3.1	TSPLIB instances	87
3.1	TSPLIB instances (continued)	88
3.1	TSPLIB instances (continued)	89
3.2	Numerical experiments for comparing 1-R-RSP and 1-S-RSP	90

Bibliography

- [1] S. Alumur and B.Y. Kara. "Network hub location problems: The state of the art". In: *European Journal of Operational Research* 190.1 (2008), pp. 1–21.
- [2] S.A. Alumur, J.F. Campbell, I. Contreras, B.Y. Kara, V. Marianov, and M.E. O'Kelly. "Perspectives on modeling hub location problems". In: *European Journal of Operational Research* 291.1 (2021), pp. 1–17.
- [3] Y. An, Y. Zhang, and B. Zeng. "The reliable hub-and-spoke design problem: Models and algorithms". In: *Transportation Research Part B: Methodological* 77 (2015), pp. 103–122.
- [4] R. Baldacci, M. Dell'Amico, and J.S. González. "The capacitated m-ring-star problem". In: *Operations Research* 55.6 (2007), pp. 1147–1162.
- [5] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski. *Robust optimization*. Vol. 28. Princeton university press, 2009.
- [6] A. Ben-Tal and A. Nemirovski. "Robust convex optimization". In: *Mathematics of Operations Research* 23.4 (1998), pp. 769–805.
- [7] A. Ben-Tal and A. Nemirovski. "Robust solutions of linear programming problems contaminated with uncertain data". In: *Mathematical Programming* 88 (2000), pp. 411–424.
- [8] A. Ben-Tal and A. Nemirovski. "Robust solutions of uncertain linear programs". In: *Operations Research letters* 25.1 (1999), pp. 1–13.
- [9] J. F. Benders. "Partitioning procedures for solving mixed-variables programming problems". In: *Numerische Mathematik* 4.1 (1962), pp. 238–252.
- [10] D. Berstimas and M. Sim. "The price of robustness". In: *Operations Research* 52.1 (2004), pp. 35–53.
- [11] J.A. Bloom. "Solving an electricity generating capacity expansion planning problem by generalized Benders' decomposition". In: *Operations Research* 31.1 (1983), pp. 84–100.

-
- [12] D.L. Bryan and M.E. O’Kelly. “Hub-and-spoke networks in air transportation: an analytical review”. In: *Journal of Regional Science* 39.2 (1999), pp. 275–295.
- [13] H.I. Calvete and J.A. Iranzo. “An efficient evolutionary algorithm for the ring star problem”. In: *European Journal of Operational Research* 231.1 (2013), pp. 22–33.
- [14] X. Chen, X. Hu, X. Jia, Z. Tang, C. Wang, and Y. Zhang. “Algorithms for the metric ring star problem with fixed edge-cost ratio”. In: *Journal of Combinatorial Optimization* 42.3 (2021), pp. 499–523.
- [15] G. Codato and M. Fischetti. “Combinatorial Benders’ cuts for mixed-integer linear programming”. In: *Operations Research* 54.4 (2006), pp. 756–766.
- [16] G.A. Croes. “A method for solving traveling-salesman problems”. In: *Operations Research* 6.6 (1958), pp. 791–812.
- [17] T.C.S. Dias, G.F. de Sousa Filho, E.M. Macambira, L.D.A.F. Cabral, and M.H.C. Fampa. “An efficient heuristic for the ring star problem”. In: *Experimental Algorithms: 5th International Workshop, WEA 2006, Cala Galdana, Menorca, Spain, May 24–27, 2006. Proceedings* 5. Springer. 2006, pp. 24–35.
- [18] L. El Ghaoui and H. Lebret. “Robust solutions to least-square problems to uncertain data matrices”. In: *SIAM Journal on Matrix Analysis and Applications* 18 (1997), pp. 1035–1064.
- [19] L. El Ghaoui, F. Oustry, and H. Lebret. “Robust solutions to uncertain semidefinite programs”. In: *SIAM Journal on Optimization* 9.1 (1998), pp. 33–52.
- [20] A. Fakhri, M. Ghatee, A. Fragkogios, and G.K.D. Saharidis. “Benders decomposition with integer subproblem”. In: *Expert Systems with Applications* 89 (2017), pp. 20–30.
- [21] R. Z. Farahani, E. Miandoabchi, W. Y. Szeto, and H. Rashidi. “A review of urban transportation network design problems”. In: *European Journal of Operational Research* 229.2 (2013), pp. 281–302.
- [22] R. Z. Farahani, M. SteadieSeifi, and N. Asgari. “Multiple criteria facility location problems: A survey”. In: *Applied Mathematical Modelling* 34.7 (2010), pp. 1689–1709. ISSN: 0307-904X. DOI: <https://doi.org/10.1016/j.apm.2009.10.005>. URL: <https://www.sciencedirect.com/science/article/pii/S0307904X09003242>.
- [23] B. Fortz and M. Poss. “An improved Benders decomposition applied to a multi-layer network design problem”. In: *Operations Research Letters* 37.5 (2009), pp. 359–364.

- [24] P. Fouilhoux, O. Ekin Karaan, A. R. Mahjoub, O. Özkök, and H. Yaman. "Survivability in hierarchical telecommunications networks". In: *Networks* 59.1 (2012), pp. 37–58.
- [25] A.M. Geoffrion. "Generalized Benders decomposition". In: *Journal of Optimization Theory and Applications* 10 (1972), pp. 237–260.
- [26] E. Gourdin, M. Labbé, and H. Yaman. "Telecommunication and location". In: *Location Problems in Telecommunications*. Ed. by Z. Drezner and H. Hamacher. Springer, 2002, pp. 275–305.
- [27] M. Grötschel, C. L. Monma, and M. Stoer. "Design of survivable networks". In: *Handbooks in Operations Research and Management Science*. Ed. by C.L. Monma M.O. Ball T.L. Magnanti and G.L. Nemhauser. Vol. 7. Elsevier, Amsterdam, 1995, pp. 617–671.
- [28] Y. Gu, X. Fu, Z. Liu, X. Xu, and A. Chen. "Performance of transportation network under perturbations: Reliability, vulnerability, and resilience". In: *Transportation Research Part E: Logistics and Transportation Review* 133 (2020), p. 101809.
- [29] C. S. Holling. "Resilience and stability of ecological systems". In: *Annual Review of Ecology and Systematics* 4.1 (1973), pp. 1–23.
- [30] C. Juvin, L. Houssin, and P. Lopez. "Logic-based Benders decomposition for the preemptive flexible job-shop scheduling problem". In: *Computers & Operations Research* 106156 (2023).
- [31] O. Ekin Karaşan, A. R. Mahjoub, O. Özkök, and H. Yaman. "Survivability in hierarchical telecommunications networks under dual homing". In: *INFORMS Journal on Computing* 26.1 (2014), pp. 1–15.
- [32] S. Kedad-Sidhoum and V.H. Nguyen. "An exact algorithm for solving the ring star problem". In: *Optimization* 59.1 (2010), pp. 125–140.
- [33] H. Kerivin and A. R. Mahjoub. "Design of survivable networks: A review". In: *Networks* 46 (2005), pp. 1–21.
- [34] A. Kershenbaum. *Telecommunications network design algorithms*. McGraw-Hill, Inc., 1993.
- [35] E. Keyvanshokoo, S.M. Ryan, and E. Kabir. "Hybrid robust and stochastic optimization for closed-loop supply chain network design using accelerated Benders decomposition". In: *European Journal of Operational Research* 249.1 (2016), pp. 76–92.

- [36] J. Khamphousone, A. Rossi, F. Castaño, and S. Toubaline. “Proc. 10th International Networks Optimization Conference (INOC, Aachen, Germany, June 7-10 2022)”. In: *Proc. 10th International Networks Optimization Conference (INOC, Aachen, Germany, June 7-10 2022)*. Aachen, Germany: OpenProceedings.org, 2022. URL: https://openproceedings.org/2022/conf/inoc/INOC_2022_paper_11.pdf.
- [37] J. G. Klincewicz. “Hub location in backbone/tributary network design: a review”. In: *Location Science* 6.1-4 (1998), pp. 307–335.
- [38] P. Kouvelis and G. Yu. *Robust discrete optimization and its applications*. Vol. 14. Springer Science & Business Media, 1996.
- [39] M. Labbé, G. Laporte, I.R. Martin, and J.J.S. González. “Locating median cycles in networks”. In: *European Journal of Operational Research* 160.2 (2005). Decision Support Systems in the Internet Age, pp. 457–470. ISSN: 0377-2217. DOI: <https://doi.org/10.1016/j.ejor.2003.07.010>. URL: <https://www.sciencedirect.com/science/article/pii/S0377221703005563>.
- [40] M. Labbé, G. Laporte, I.R. Martin, and J.J.S. González. “The ring star problem: Polyhedral analysis and exact algorithm”. In: *Networks* 43.3 (2004), pp. 177–189.
- [41] M. Labbé, H. Yaman, and É. Gourdin. “A branch and cut algorithm for hub location problems with single assignment”. In: *Mathematical Programming* 102 (2005), pp. 371–405.
- [42] P. Lammich and S.R. Sefidgar. “Formalizing the Edmonds-Karp algorithm”. In: *International Conference on Interactive Theorem Proving*. Springer. 2016, pp. 219–234.
- [43] T.-Y. Liao, T.-Y. Hu, and Y.-N. Ko. “A resilience optimization model for transportation networks under disasters”. In: *Natural Hazards* 93 (2018), pp. 469–489.
- [44] Z. Naji-Azimi, M. Salari, and P. Toth. “A heuristic procedure for the Capacitated m-Ring-Star problem”. In: *European Journal of Operational Research* 207.3 (2010), pp. 1227–1234. ISSN: 0377-2217. DOI: <https://doi.org/10.1016/j.ejor.2010.06.030>. URL: <https://www.sciencedirect.com/science/article/pii/S0377221710004698>.
- [45] M. E. O’Kelly. “A quadratic integer program for the location of interacting hub facilities”. In: *European Journal of Operational Research* 32.3 (1987), pp. 393–404. ISSN: 0377-2217. DOI: [https://doi.org/10.1016/S0377-2217\(87\)80007-3](https://doi.org/10.1016/S0377-2217(87)80007-3). URL: <https://www.sciencedirect.com/science/article/pii/S0377221787800073>.

-
- [46] C. Ortiz-Astorquiza, I. Contreras, and G. Laporte. “Multi-level facility location problems”. In: *European Journal of Operational Research* 267.3 (2018), pp. 791–805. ISSN: 0377-2217. DOI: <https://doi.org/10.1016/j.ejor.2017.10.019>. URL: <https://www.sciencedirect.com/science/article/pii/S0377221717309323>.
- [47] R. Rahmaniani, T.G. Crainic, M. Gendreau, and W. Rei. “The Benders decomposition algorithm: A literature review”. In: *European Journal of Operational Research* 259.3 (2017), pp. 801–817. ISSN: 0377-2217. DOI: <https://doi.org/10.1016/j.ejor.2016.12.005>. URL: <https://www.sciencedirect.com/science/article/pii/S0377221716310244>.
- [48] A. Rossi, E. Gurevsky, O. Battaia, and A. Dolgui. “Maximizing the robustness for simple assembly lines with fixed cycle time and limited number of workstations”. In: *Discrete Applied Mathematics* 208 (2016), pp. 123–136.
- [49] B. Roy. “Robustness in operational research and decision aiding: A multifaceted issue”. In: *European Journal of Operational Research* 200.3 (2010), pp. 629–638. ISSN: 0377-2217. DOI: <https://doi.org/10.1016/j.ejor.2008.12.036>. URL: <https://www.sciencedirect.com/science/article/pii/S0377221708010564>.
- [50] Ruprecht-Karls-Universität Heidelberg. *TSPLIB* 95. Available at <http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/>.
- [51] E. M. de Sá, I. Contreras, J.-F. Cordeau, R. Saraiva de Camargo, and G. de Miranda. “The hub line location problem”. In: *Transportation Science* 49.3 (2015), pp. 500–518.
- [52] D.B. Shmoys, É. Tardos, and K. Aardal. “Approximation algorithms for facility location problems”. In: *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*. 1997, pp. 265–274.
- [53] L. Simonetti, Y. Frota, and C.C. de Souza. “The ring-star problem: a new integer programming formulation and a branch-and-cut algorithm”. In: *Discrete Applied Mathematics* 159.16 (2011), pp. 1901–1914.
- [54] S. Wang, J. Liu, and Y. Jin. “A computationally efficient evolutionary algorithm for multiobjective network robustness optimization”. In: *IEEE Transactions on Evolutionary Computation* 25.3 (2021), pp. 419–432.
- [55] D. Weninger and L.A. Wolsey. “Benders algorithm with (mixed)-integer subproblems”. In: *Université catholique de Louvain* (2019).
- [56] J. Xu, S.Y. Chiu, and F. Glover. “Optimizing a ring-based private line telecommunication network using tabu search”. In: *Management Science* 45.3 (1999), pp. 330–345.

- [57] X. Zang, L. Jiang, B. Ding, and X. Fang. "A hybrid ant colony system algorithm for solving the ring star problem". In: *Applied Intelligence* 51 (2020), pp. 3789–3800.
- [58] X.-T. Zhang, H.-L. Bi, and Y. Wang. "A stochastic programming approach for resilient hub location in power projection network considering random hub failures". In: *Mathematical Problems in Engineering* 6517453 (2017).

RÉSUMÉ

Dans les réseaux de télécommunication, les réseaux de transport, la logistique et plusieurs autres domaines, la conception de réseaux induit de nombreux problèmes sous-jacents. Dans cette thèse, nous nous intéressons à la structure à deux niveaux composée d'une ossature et d'une partie tributaire. L'un des problèmes avec cette architecture est le problème Ring Star où un dépôt est défini à l'avance. Les nœuds sont sélectionnés ou non en tant qu'hubs et ces hubs et le dépôt sont reliés par un "Ring" tandis que les nœuds non sélectionnés forment la partie tributaire en étant connectés à un nœud du ring pour former la partie "Star". Nous introduisons deux variantes du problème Ring Star. Une variante à capacité de survie et une variante résiliente. Nous formulons un PLNE ainsi qu'un algorithme de Branch-and-Benders-cut pour les deux problèmes. Des propriétés sont étudiées, elles permettent des améliorations computationnelles mises en évidence par des expérimentations numériques.

MOTS CLÉS

Conception de réseaux, Problème Ring Star, Survivabilité, Résilience, Programmation Linéaire en Nombres Entiers, Décomposition de Benders, Branch-and-Benders-cut

ABSTRACT

In telecommunication networks, transportation networks, logistics, and many other fields, networks design is a vast subject inducing many underlying problems. In this thesis, we focus on the two-level structure, backbone and tributary architecture. One of the problems with such an architecture is the Ring Star Problem where there is a fixed depot. Nodes must then be selected as hubs or non-hubs. Hubs and the depot are linked by a Ring to form the backbone architecture while the non-hubs form the tributary architecture by being connected to the ring to form a star. In this thesis, we introduce two variants of the Ring Star Problem. A survivable variant and a resilient one. We formulate an Integer Linear Program and a Branch-and-Benders-cut algorithm for both problems. Some properties are studied, allowing to improve the computational performance, and numerical experiments are carried out.

KEYWORDS

Network Design, Ring Star Problem, Survivability, Resilience, Integer Linear Programming, Benders Decomposition, Branch-and-Benders-cut