ALGORITMOS DE OTIMIZAÇÃO PARA O PROBLEMA DO CAMINHO MÍNIMO ROBUSTO RESTRITO

LUCAS ASSUNÇÃO DE ALMEIDA

ALGORITMOS DE OTIMIZAÇÃO PARA O PROBLEMA DO CAMINHO MÍNIMO ROBUSTO RESTRITO

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais como requisito parcial para a obtenção do grau de Mestre em Ciência da Computação.

Orientador: Thiago Ferreira de Noronha
Coorientadora: Andréa Cynthia Santos

Belo Horizonte
Março de 2015

LUCAS ASSUNÇÃO DE ALMEIDA

OPTIMIZATION ALGORITHMS FOR THE RESTRICTED ROBUST SHORTEST PATH PROBLEM

Dissertation presented to the Graduate Program in Computer Science of the Universidade Federal de Minas Gerais — Departamento de Ciência da Computação in partial fulfillment of the requirements for the degree of Master in Computer Science.

Advisor: Thiago Ferreira de Noronha
Co-Advisor: Andréa Cynthia Santos

Belo Horizonte

March 2015

Almeida, Lucas Assunção de

A447o Optimization algorithms for the restricted robust shortest path problem / Lucas Assunção de Almeida. — Belo Horizonte, 2015
xxiv, 73 f. : il. ; 29cm

Dissertação (mestrado) — Universidade Federal de Minas Gerais — Departamento de Ciência da Computação
Orientador: Thiago Ferreira de Noronha
Coorientadora: Andréa Cynthia Santos

1. Computação — Teses. 2. Otimização Combinatória — Teses. I. Orientador. II. Coorientadora. III. Título.

CDU 519.6*61.(043)

UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

FOLHA DE APROVAÇÃO

Optimization algorithms for the restricted robust shortest path problem

LUCAS ASSUNCAO DE ALMEIDA

Dissertação defendida e aprovada pela banca examinadora constituída pelos Senhores:

PROF. THIAGO FERREIRA DE NORONHA - Orientador
Departamento de Ciência da Computação - UFMG

PROFA. ANDREA CYNTHIA SANTOS
Université de Tecnologie de Troyes

PROF. GERALDO ROBSON MATEUS
Departamento de Ciência da Computação - UFMG

PROF. RAFAEL CASTRO DE ANDRADE
Departamento de Estatística e Matemática Aplicada - UFC

PROF. RICARDO SARAIVA DE CAMARGO
Departamento de Engenharia de Produção - UFMG

Belo Horizonte, 09 de março de 2015.

À minha "mainha", cuja herança de lealdade e amor permeia todos os aspectos da minha vida.

"Deja que el alma
tenga la misma edad
que la edad del cielo"
(Jorge Drexler)

Abstract

This dissertation introduces the Restricted Robust Shortest Path problem (R-RSP), a robust optimization version of the Restricted Shortest Path problem (R-SP), a classical NP-hard problem. Given a digraph G, we associate a cost interval and a resource consumption value with each arc of G. R-RSP aims at finding a path from an origin to a destination vertices in G that satisfies a resource consumption constraint and minimizes a robust optimization criterion, called restricted robustness cost. This problem has practical applications, as routing electrical vehicles in urban areas, when one looks for a path from a location to another taking into account traffic jams and the vehicles' autonomy. R-RSP belongs to a new class of problems composed by robust optimization problems whose classical versions (*i.e.*, parameters known in advance) are already NP-hard. We refer to this class as *robust-hard* problems. Problems in this class are particularly challenging, as solely evaluating the cost of a solution requires solving a NP-hard problem, which corresponds to the classical counterpart of the problem considered. In this study, we discuss some theoretical aspects of R-RSP, including its computational complexity. Indeed, we show that both R-RSP and its decision version are NP-hard. We also derive a MILP formulation (with a polynomial number of variables and an exponential number of constraints) for R-RSP. Based on this formulation, we propose a heuristic method for R-RSP that consists in solving an approximate compact MILP formulation that uses dual information of the linear relaxation of R-SP. Moreover, a Benders-like decomposition approach is proposed to solve R-RSP at optimality. We also present some techniques to improve the convergence speed of the method by providing initial bounds, as well as by generating additional Benders cuts. Computational experiments show the effectiveness of the proposed algorithms. We highlight that the procedures to solve R-RSP presented in this dissertation are not limited to the referred problem, as they can be extended to other robust-hard problems.

Resumo

Introduzimos o problema do caminho mais curto restrito robusto (R-RSP, do inglês *restricted robust shortest path problem*), uma versão de otimização robusta do problema do caminho mais curto restrito (R-SP, do inglês *restricted shortest path problem*), um clássico problema NP-difícil. Dado um grafo orientado G, associamos um intervalo de custo e um valor de recurso a cada arco de G. R-RSP visa encontrar um caminho de menor custo de um vértice de origem a um de destino em G que satisfaz uma restrição de consumo máximo de recurso e minimiza um critério de otimização robusta, chamado de *desvio robusto restrito*. R-RSP tem aplicacões práticas, como em roteamento de veículos elétricos em áreas urbanas, quando se procura um caminho de uma localização a outra levando em consideração engarrafamentos e a autonomia energética do veículo. R-RSP pertence a uma nova classe de problemas composta por problemas de otimização robusta cujas versões de otimização clássica já são NP-difícil. Referimo-nos a essa classe como problemas *robusto-difícil*. Problemas nessa classe são particularmente desafiadores, uma vez que calcular o custo de uma solução envolve resolver um problema NP-difícil, que corresponde à versão clássica do problema considerado. Neste estudo, discutimos aspectos teóricos de R-RSP, incluindo sua complexidade computational. De fato, mostramos que tanto R-RSP como sua versão de decisão são NP-difícil. Derivamos uma formulação de programação inteira mista (com um número polinomial de variáveis e um número exponencial de restrições) para R-RSP. Baseando-nos nessa formulação, propomos uma heurística para R-RSP que consiste em resolver uma formulação de programação inteira mista que usa informação dual da relaxação linear de R-SP. Propomos ainda uma estratégia baseada em decomposição de Benders para resolver R-RSP na otimalidade. Apresentamos algumas técnicas para melhorar a velocidade de convergência do método a partir da obtenção de limites iniciais e da geração de cortes de Benders adicionais. Experimentos computacionais mostram a eficácia dos algoritmos propostos. Destacamos que os procedimentos para resolver R-RSP apresentados nesta dissertação não se limitam ao referido problema, já que eles podem ser estendidos a outros problemas robusto-difícil.

List of Figures

2.1 Example of an IRRSP instance, with origin $o = 0$ and destination $t = 3$. The notation $[l_{ij}, u_{ij}]$ gives the cost interval associated with each arc $(i, j) \in A$. 7

3.1 Example of a R-RSP instance, with origin $o = 0$ and destination $t = 3$. The notation $[l_{ij}, u_{ij}]\{d_{ij}\}$ means, respectively, the cost interval $[l_{ij}, u_{ij}]$ and the resource consumption $\{d_{ij}\}$ associated with each arc $(i, j) \in A$. 12

3.2 Scenario $s_{\tilde{p}}$ induced by the path $\tilde{p} = \{0, 1, 2, 3\}$ in the graph presented in Figure 3.1. For each arc $(i, j) \in A$, the notation $c_{ij}^{s_{\tilde{p}}}\{d_{ij}\}$ means, respectively, the arc cost $c_{ij}^{s_{\tilde{p}}}$ in the scenario $s_{\tilde{p}}$ and its resource consumption $\{d_{ij}\}$. . . . 13

3.3 Scenario $s_{\tilde{p}}$ induced by the path $\tilde{p} = \{0, 1, 2, 3\}$ in the graph presented in Figure 2.1. For each arc $(i, j) \in A$, it is shown the arc cost $c_{ij}^{s_{\tilde{p}}}$ in $s_{\tilde{p}}$. 16

3.4 Scenario $\bar{s}_{\tilde{p}}$ co-induced by the path $\tilde{p} = \{0, 1, 2, 3\}$ in the graph presented in Figure 3.1. For each arc $(i, j) \in A$, the notation $c_{ij}^{\bar{s}_{\tilde{p}}}\{d_{ij}\}$ denotes, respectively, the arc cost $c_{ij}^{\bar{s}_{\tilde{p}}}$ in the scenario $\bar{s}_{\tilde{p}}$ and its resource consumption $\{d_{ij}\}$. 22

6.1 An acyclic digraph with 3 layers of width 2. Here, $o = 0$ and $t = 7$. 44

6.2 A 2×4 grid digraph, with $o = 0$ and $t = 7$. 44

6.3 Summary of the results concerning the quality of the bounds obtained by the exact algorithms for the hardest grid digraph instances (instance set G-5x400-200-0.9). 58

6.4 Summary of the results concerning the quality of the bounds obtained by the exact algorithms for the hardest layered and acyclic digraph instances. 59

List of Tables

6.1 Computational results of LPH for the layered and acyclic digraph instances. 46
6.2 Computational results of LPH for the grid digraph instances. 47
6.3 Computational results of Standard Benders and Extended Benders for the layered and acyclic digraph instances, with 3600 seconds of time limit. . . 49
6.4 Computational results of Standard Benders and Extended Benders for the grid digraph instances, with 3600 seconds of time limit. 50
6.5 Algorithms obtained from coupling Standard Benders and Extended Benders with the warm start procedures. 50
6.6 Computational results of RS-Benders, HS-Benders and RS&HS-Benders for the layered and acyclic digraph instances, with 3600 seconds of time limit. 52
6.7 Computational results of RS-Benders, HS-Benders and RS&HS-Benders for the grid digraph instances, with 3600 seconds of time limit. 53
6.8 Computational results of Extended RS-Benders, Extended HS-Benders and Extended RS&HS-Benders for the layered and acyclic digraph instances, with 3600 seconds of time limit. 55
6.9 Computational results of Extended RS-Benders, Extended HS-Benders and Extended RS&HS-Benders for the grid digraph instances, with 3600 seconds of time limit. 56

A.1 Computational results of AMU for the layered and acyclic digraph instances. 70
A.2 Computational results of AMU for the grid digraph instances. 71
A.3 Average execution times (in seconds) of the warm start procedures for the grid digraph instances. 72
A.4 Average execution times (in seconds) of the warm start procedures for the layered and acyclic digraph instances. 73

List of Acronyms

RO *Robust Optimization*

LP *Linear Programming*

ILP *Integer Linear Programming*

MILP *Mixed Integer Linear Programming*

R-RSP *Restricted Robust Shortest Path problem*

R-SP *Restricted Shortest Path problem*

RSP *Robust Shortest Path problem*

IRRSP *Interval min-max Regret Robust Shortest Path problem*

AM *Algorithm Mean*

AU *Algorithm Upper*

AMU *Algorithm Mean Upper*

LPH *Linear Programming based Heuristic*

RS *Relaxation Start*

HS *Heuristic Start*

Contents

Abstract **xiii**

Resumo **xv**

List of Figures **xvii**

List of Tables **xix**

List of Acronyms **xxi**

1 Introduction **1**

2 Related works **5**

2.1 The restricted shortest path problem 5

2.2 The robust shortest path problem 6

2.3 Robust-hard problems 8

3 The restricted robust shortest path problem **11**

3.1 Problem definition and notation 11

3.2 Mathematical formulation 12

3.3 Theoretical results 16

3.4 Computational complexity 25

4 A linear programming based heuristic **29**

5 Benders-like decomposition approaches **33**

5.1 Standard Benders decomposition 33

5.2 Extended Benders decomposition 37

5.3 Warm start procedures 39

6 Computational experiments **43**

6.1 Benchmarks description 43
6.2 Results 45
6.2.1 LPH 45
6.2.2 Exact algorithms and warm start procedures 48
6.2.3 Discussion 58

7 Concluding remarks **61**

Bibliography **63**

Appendix A Additional tables **69**
A.1 AMU 69
A.2 Warm Start procedures 70

Chapter 1

Introduction

Different approaches can be applied to model problems under uncertain data. Stochastic programming [Spall, 2003], for example, models uncertainty through a probabilistic description and is mostly applied whenever the probability law associated with the uncertain data is known in advance. Alternatively, in Robust Optimization (RO) [Kouvelis and Yu, 1997], the variability of the data is represented by deterministic values in the context of *scenarios*. Here, a *scenario* corresponds to a parameters assignment, *i.e.*, a value is fixed for each parameter subject to uncertainty. In this sense, two main approaches are adopted to model RO problems: the *discrete scenarios model* and the *interval data model*. In the former, only a discrete set of possible scenarios is considered. In the latter, the uncertainty referred to a parameter is represented by a continuous interval of possible values. Differently from the discrete scenarios model, the infinite many possible scenarios that arise in the interval data model are not explicitly given. Nevertheless, in both models, a classical (*i.e.*, parameters known in advance) optimization problem takes place whenever a scenario is established.

The most commonly adopted RO criteria are the *absolute robustness* criterion, the *min-max regret* (also known as *robust deviation* criterion) and the *min-max relative regret* (also known as *relative robustness* criterion). The absolute robustness criterion is based on the anticipation of the worst possible conditions. Solutions for RO problems under such criterion tend to be conservative, as they optimize only a worst-case scenario. The min-max regret and the min-max relative regret are less conservative criteria and have been adopted in several works (*e.g.*, Averbakh [2005]; Montemanni [2006]; Montemanni et al. [2007]; Pereira and Averbakh [2011]; Coco et al. [2014a]). Intuitively speaking, the *regret* of a solution in a given scenario is the difference between the cost of such solution and the cost of an optimal one in the referred scenario. In this sense, the min-max regret criterion aims at minimizing (among all feasible solutions) the maximum (among all possible scenarios) regret. In turn, the *relative regret* of a solution in a given scenario consists of the corresponding regret normalized by the cost of an optimal solution for the scenario considered. We refer to Kouvelis and Yu [1997]; Coco et al. [2014b] for details on RO models and criteria.

Robust optimization versions of several combinatorial optimization problems have been studied in the literature, addressing, for example, uncertainties over costs. Handling uncertain costs brings an extra level of difficulty, such that even polynomially solvable problems become NP-hard in their corresponding robust versions (see, *e.g.*, [Aron and Hentenryck, 2004; Aissi et al., 2005; Kasperski, 2008]). A recent trend is to further investigate RO problems whose classical counterparts are already NP-hard. We refer to these problems as *robust-hard* problems. The RO problem introduced in this study, namely the Restricted Robust Shortest Path problem (R-RSP), belongs to this class of problems, along with the robust traveling salesman problem [Montemanni et al., 2007], the robust knapsack problem [Deineko and Woeginger, 2010], and the robust set covering problem [Pereira and Averbakh, 2013], among others [Deineko and Woeginger, 2010; Solano-Charris et al., 2014].

Robust-hard problems are particularly challenging, as solely evaluating the cost of a feasible solution requires solving a NP-hard problem, which corresponds to the classical counterpart of the RO problem considered. As far as we know, no work in the literature was able to provide a compact Integer Linear Programming (ILP) formulation (with a polynomial number of variables and constraints) for any robust-hard problem. In fact, the standard exact methods used to solve robust-hard problems are based on the generation of a possibly exponential number of cuts (see, *e.g.*, Montemanni et al. [2007]; Pereira and Averbakh [2013]). Since the generation of each cut is linked to the resolution of a NP-hard problem instance (precisely, the classical counterpart of the RO problem considered), these strategies present several challenges. For example, the size of the instances efficiently solvable is considerably smaller than the size of the instances solved in the corresponding classical counterparts. Innovative approaches are then required to better handle these problems.

In this study, we are interested in robust-hard optimization problems that consider the interval data model under the min-max regret. In particular, we focus on R-RSP, which is a RO version of the Restricted Shortest Path problem (R-SP), an extensively studied NP-hard problem [Garey and Johnson, 1979; Handler and Zang, 1980; Aneja et al., 1983; Beasley and Christofides, 1989; Hassin, 1992; Wang and Crowcroft, 1996].

R-SP is defined on a digraph $G = (V, A)$, where V is the set of vertices and A is the set of arcs. With each arc $(i, j) \in A$, we associate nonnegative values c_{ij} and

d_{ij}, which represent, respectively, the arc cost and the resource consumption incurred by using that arc. An origin and a destination vertices are also given, as well as a nonnegative value β, used to limit the resource consumed along a path (given by the summation of the resource consumption of each arc of the path). R-SP consists in finding a shortest path (with respect to arc costs) from the origin to the destination while not exceeding the limit β on the resource consumption. In R-RSP, the cost of the arcs are defined by intervals of values, and the problem is to find a path that minimizes the maximum regret (min-max regret) while satisfying the resource consumption limit.

R-RSP has applications in determining paths in urban areas, with a length (distance) constraint and uncertainties related to travel time. These uncertainties can happen due to traffic jams, bad weather conditions, etc. Here, uncertainties are represented by values in continuous intervals, which estimate the minimum and the maximum times to traverse each pathway. R-RSP can model situations involving electrical vehicles with a limited battery (energy) autonomy. A similar application arises in telecommunications field, when one wants to determine a path to send a data packet from an origin to a destination node in a network, while satisfying a Quality of Service (QoS) guarantee [Wang and Crowcroft, 1996; Apostolopoulos et al., 1998]. Also in this case, the transmission lines are subject to uncertain delays due to the varying traffic load on the network. To the best of our knowledge, R-RSP has not been studied in the literature, and this is also the first study to model and to propose procedures for solving it. Therefore, with respect to R-RSP, the contributions of this dissertation are twofold. First, we introduce R-RSP, along with a theoretical study and a Mixed Integer Linear Programming (MILP) formulation (with a polynomial number of variables and an exponential number of constraints) for the problem. Second, we propose a linear programming based heuristic and Benders-like decomposition algorithms (which combine speed-up techniques) to solve this formulation. We highlight that the procedures to solve R-RSP presented in this study are not limited to the referred problem, as they can be extended and applied to solve other robust-hard problems.

The remaining of this dissertation is organized as follows. In Chapter 2, we discuss related works by pointing out the main studies concerning R-SP and some variations of the Robust Shortest Path problem (RSP) [Gupta and Rosenhead, 1968; Rosenhead et al., 1972; Kouvelis and Yu, 1997; Zieliński, 2004; Averbakh, 2005]. Additionally, we present a literature review on robust-hard problems.

Chapter 3 is dedicated to introduce R-RSP and to study theoretical aspects regarding the problem. In Section 3.1, we give the formal definition of R-SP and, in Section 3.2, we present a mathematical formulation for the problem. Moreover, a theoretical study is conducted in Section 3.3, and the problem's computational complexity is discussed in Section 3.4. We show the NP-hardness of both R-RSP and the decision problem associated with it.

In Chapter 4, we propose a linear programming based heuristic to tackle robust-hard problems as a case study on solving R-RSP. It consists in solving a MILP formulation based on the dual of the linear relaxation of the classical optimization problem counterpart. Then, in Chapter 5, a Benders-like decomposition approach widely used to solve RO problems [Montemanni and Gambardella, 2005; Montemanni, 2006; Montemanni et al., 2007; Pereira and Averbakh, 2011, 2013] is adapted to R-RSP. In addition, we discuss some techniques able to improve the convergence speed of the method by providing initial bounds, as well as by generating additional Benders cuts. These techniques are coupled with the Benders-like decomposition approach in different manners to generate a total of eight exact algorithms.

Chapter 6 is dedicated to computational experiments. In Section 6.1, we introduce two benchmarks of instances inspired by the applications described in Chapter 1. Furthermore, in Section 6.2, we show the computational results concerning the heuristic and the exact algorithms on solving the two benchmarks of instances. Concluding remarks are provided in the last chapter.

Chapter 2

Related works

2.1 The restricted shortest path problem

R-SP is known to be NP-hard [Garey and Johnson, 1979; Handler and Zang, 1980] even for acyclic graphs [Bondy and Murty, 1976; Wang and Crowcroft, 1996]. The main exact algorithms to solve R-SP found in the literature can be classified into two groups: Lagrangian relaxation procedures and dynamic programming procedures. The former procedures use Lagrangian relaxation to handle ILP formulations for the problem (see, *e.g.*, Handler and Zang [1980]; Beasley and Christofides [1989]). In addition, preprocessing techniques have been presented in Aneja et al. [1983] and refined in Beasley and Christofides [1989]. These strategies identify arcs and vertices that cannot compose an optimal solution for R-SP through the analysis of the reduced costs related to the resolution of dual Lagrangian relaxations. More recently, Santos et al. [2007] proposed a path ranking approach that linearly combines the arc costs and the resource consumption values to generate a descent direction of search. In turn, the dynamic programming procedures for R-SP consist of label-setting and label-correcting algorithms, such as the one proposed in Joksch [1966] and further improved in Dumitrescu and Boland [2003] by the addition of preprocessing strategies. Recently, Zhu and Wilhelm [2012] developed a three-stage label-setting algorithm that runs in pseudo-polynomial time and stands among the state-of-the-art methods dedicated to R-SP, along with the algorithms presented in Santos et al. [2007] and in Dumitrescu and Boland [2003]. We refer to Pugliese and Guerriero [2013] for a survey of the main contributions with respect to exact methods to solve R-SP.

Although NP-hard, R-SP can be solved efficiently by some of the aforementioned procedures (particularly, the ones proposed in Santos et al. [2007]; Dumitrescu and Boland [2003]; Zhu and Wilhelm [2012]). Moreover, optimization softwares, as

CPLEX[1], are also competitive in handling reasonably-sized instances (with up to 3000 vertices) of the problem [Zhu and Wilhelm, 2012]. These evidences encourage the investigation of more realistic and complex models, such as the RO version of R-SP introduced in this study, namely R-RSP, which deals simultaneously with a limit on the resource consumption and with uncertain arc costs.

2.2 The robust shortest path problem

R-RSP is also related to RSP, a RO counterpart of the classical shortest path problem [Shimbel, 1953]. RSP is defined on a digraph whose arcs are subject to uncertain costs. Given an origin and a destination vertices, RSP consists in finding a path (from the origin to the destination) that minimizes a given RO criterion. RSP was introduced by Gupta and Rosenhead [1968], and, since then, some variants of the problem have been proposed by considering different RO models and criteria. In Kouvelis and Yu [1997], the problem was firstly modeled under the *interval data* and the *discrete scenarios* models. In addition, three RO criteria were applied to RSP in the study: the *absolute robustness* criterion, the *min-max regret* and the *min-max relative regret*. In the same study, the authors also proved the NP-hardness of several RO problems, including the interval data min-max relative regret version of RSP. In Coco et al. [2014a], the first MILP formulation for such version of the problem was presented, along with heuristic procedures for solving it.

R-RSP is particularly related to Interval min-max Regret Robust Shortest Path problem (IRRSP), the RSP version that considers the interval data model under the min-max regret. IRRSP is defined on a digraph $G = (V, A)$, where V is the set of vertices and A is the set of arcs. With each arc $(i, j) \in A$, we associate a cost interval $[l_{ij}, u_{ij}]$, where $l_{ij} \in \mathbb{Z}_+$ is the lower bound, and $u_{ij} \in \mathbb{Z}_+$ is the upper bound on this interval of cost, with $l_{ij} \leq u_{ij}$. An origin vertex o and a destination vertex t are also given. An example of an IRRSP instance is showed in Figure 2.1.

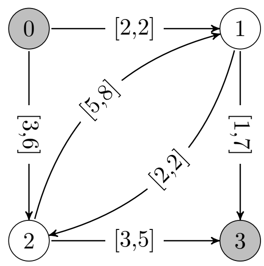
Let $\mathcal{P}$ be the set of all paths from o to t and $A[p]$ be the set of arcs that compose a path $p \in \mathcal{P}$.

Definition 1. A *scenario* s is an assignment of arc costs, *i.e.*, a cost $c_{ij}^s \in [l_{ij}, u_{ij}]$ is fixed $\forall\, (i, j) \in A$.

Let $\mathcal{S}$ be the set of all possible scenarios of G. The cost of a path $p \in \mathcal{P}$ in a scenario $s \in \mathcal{S}$ is given by $C_p^s = \sum_{(i,j)\in A[p]} c_{ij}^s$.

[1] http://www.ilog.com/products/cplex/

Figure 2.1: Example of an IRRSP instance, with origin $o = 0$ and destination $t = 3$. The notation $[l_{ij}, u_{ij}]$ gives the cost interval associated with each arc $(i, j) \in A$.



Definition 2. The *robust deviation* of a path $p \in \mathcal{P}$ in a scenario $s \in \mathcal{S}$, denoted by r_p^s, is the difference between the cost C_p^s of p in s and the cost of a minimum cost path $p^*(s) \in \mathcal{P}$ in s, *i.e.*, $r_p^s = C_p^s - C_{p^*(s)}^s$.

Definition 3. The *robustness cost* of a path $p \in \mathcal{P}$, denoted by R_p, is the maximum robust deviation of p among all possible scenarios, *i.e.*, $R_p = \max_{s \in \mathcal{S}} r_p^s$.

Definition 4. A path $p^* \in \mathcal{P}$ is said to be a *robust path* if it has the smallest robustness cost among all paths in $\mathcal{P}$, *i.e.*, $p^* = \arg\min_{p \in \mathcal{P}} R_p$.

Definition 5. IRRSP consists in finding a robust path.

IRRSP has been largely studied in the literature, and dedicated procedures have been developed, such as the *branch-and-bound* algorithm of Montemanni et al. [2004] and the Benders-like decomposition approach of Montemanni and Gambardella [2005]. Both algorithms are able to solve instances in random graphs with up to 4000 nodes and real instances with up to 2500 nodes. Moreover, preprocessing techniques able to identify arcs that cannot compose an optimal solution, namely *weak-arcs*, have been proposed by Karaşan et al. [2001] and later improved by Catanzaro et al. [2011].

In Kasperski et al. [2005], scenario-based procedures were proposed to solve IRRSP. The first, called Algorithm Mean (AM), returns a shortest path in the *median scenario*, where the cost of each arc $(i, j) \in A$ is set to its respective mean value, given by $\frac{(l_{ij}+u_{ij})}{2}$. According to Kasperski and Zieliński [2007], AM is a 2-approximation algorithm. The second procedure, called Algorithm Upper (AU), returns a shortest path in the *worst-case scenario*, where each arc cost is set to its corresponding upper bound. The third, called Algorithm Mean Upper (AMU), returns the best solution found by AM and AU. Therefore, AMU is also 2-approximative [Kasperski and Zieliński, 2007]. Recently, in Pérez et al. [2012], a *simulated annealing* heuristic inspired by Kirkpatrick et al. [1983] was proposed to solve IRRSP.

2.3 Robust-hard problems

As far as we know, Montemanni et al. [2006, 2007] were the first studies to address a robust-hard problem, *i.e.*, a RO version of a classical NP-hard problem. In such works, the interval min-max regret traveling salesman problem was investigated. In Montemanni et al. [2007], a mathematical formulation with an exponential number of constraints was proposed for the problem, along with exact approaches to solve it. Precisely, a *branch-and-bound* algorithm, a *branch-and-cut* algorithm and a Benders-like decomposition algorithm were proposed and computationally compared. The latter algorithm, which adapts a Benders-like decomposition approach widely used to solve RO problems (see, *e.g.*, Montemanni and Gambardella [2005]; Montemanni [2006]), outperforms the other exact algorithms on the benchmarks of instances considered. Montemanni et al. [2006] presented heuristic procedures for the problem. The heuristics use the idea proposed in Kasperski et al. [2005] and consist in solving the classical traveling salesman problem [Dantzig et al., 1954] in selected scenarios. In addition, the authors proposed preprocessing techniques able to identify edges that cannot compose an optimal solution tour.

In Pereira and Averbakh [2013], the interval min-max regret set covering problem was introduced. The authors proposed a mathematical formulation for the problem that is similar to the one proposed by Montemanni et al. [2007] for the interval min-max regret traveling salesman problem. As in Montemanni et al. [2007], the proposed formulation has an exponential number of constraints. The authors also adapted to the problem the standard Benders-like decomposition approach for RO problems (see, *e.g.*, Montemanni and Gambardella [2005]; Montemanni [2006]) and presented an extension of the method that aims at generating multiple Benders cuts per iteration of the decomposition. Moreover, the work presents an exact approach that uses Benders cuts in the context of a *branch-and-cut* framework. Such approach, as well as the extended Benders-like decomposition, outperforms the standard Benders-like decomposition on the instances tested. With respect to heuristic solutions, the same work [Pereira and Averbakh, 2013] proposed scenario-based procedures that find feasible solutions by solving the classical set covering problem [Revelle et al., 1970] in few selected scenarios. The authors also proposed a *genetic algorithm* and a *memetic algorithm* for the problem. The latter algorithm uses a Benders decomposition within the framework of the genetic algorithm to strengthen the crossover operator. According to the results, this hybrid heuristic outperforms the basic genetic algorithm and was able to find the same or better solutions than the *branch-and-cut* algorithm when given a same 900 seconds time limit.

A few works dealt with RO versions of the classical knapsack problem. For instance, Yu [1996] and Kouvelis and Yu [1997] addressed a version of the problem where the uncertainty over each item profit is represented by a discrete set of possible values. In these works, the absolute robustness criterion is considered. In Yu [1996], the author proved that this version of the problem is strongly NP-hard when the number of possible scenarios is unbounded and pseudo-polynomially solvable for a bounded number of scenarios. For the version of the robust knapsack problem that considers the min-max regret criterion under a discrete set of scenarios, Kouvelis and Yu [1997] provided a pseudo-polynomial algorithm for solving the problem when the number of possible scenarios is bounded. When the number os scenarios is unbounded, the problem becomes strongly NP-hard and there is no approximation scheme for it [Aissi et al., 2007, 2009].

Recently, in Deineko and Woeginger [2010], the computational complexity of the interval data min-max regret version of the knapsack problem was studied. The authors showed that the corresponding decision problem is complete for the complexity class $\sum_2^p$ [Meyer and Stockmeyer, 1972], *i.e.*, the problem is at the second level of the polynomial hierarchy. The polynomial hierarchy was established as an attempt to properly classify decision problems that appear to be harder than NP-complete problems. Such hierarchy contains complexity classes that generalize the classes P, NP and co-NP to oracle machines. Particularly, the complexity class $\sum_2^p$ lies one level above the class NP in the polynomial hierarchy. We refer to Papadimitriou [1994], Chapter 17, for details on the polynomial hierarchy.

Chapter 3

The restricted robust shortest path problem

In this chapter, we formally describe R-RSP and derive a MILP formulation for the problem, along with some theoretical results. Moreover, we discuss the computational complexity of both R-RSP and its corresponding decision version.

3.1 Problem definition and notation

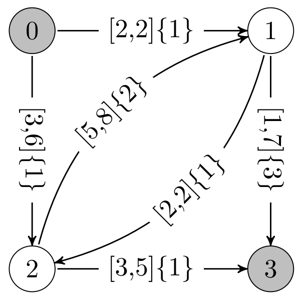
R-RSP is defined on a digraph $G = (V, A)$, where V is the set of vertices and A is the set of arcs. With each arc $(i, j) \in A$, we associate a resource consumption $d_{ij} \in \mathbb{R}_+$ and a cost interval $[l_{ij}, u_{ij}]$, where $l_{ij} \in \mathbb{Z}_+$ is the lower bound, and $u_{ij} \in \mathbb{Z}_+$ is the upper bound on this interval of cost, with $l_{ij} \leq u_{ij}$. The origin vertex o and the destination vertex t are also given, as well as a value $\beta \in \mathbb{R}_+$, parameter used to limit the resource consumed along a path from o to t, as discussed in the sequence. An example of a R-RSP instance is given in Figure 3.1.

Let $\mathcal{P}$ be the set of all paths from o to t and $A[p]$ be the set of arcs that compose a path $p \in \mathcal{P}$. The following definitions describe R-RSP formally.

Definition 6. A *scenario* s is an assignment of arc costs, *i.e.*, a cost $c^s_{ij} \in [l_{ij}, u_{ij}]$ is fixed $\forall\, (i, j) \in A$.

Let $\mathcal{S}$ be the set of all possible scenarios of G. The cost of a path $p \in \mathcal{P}$ in a scenario $s \in \mathcal{S}$ is given by $C^s_p = \sum_{(i,j)\in A[p]} c^s_{ij}$. Similarly, the resource consumption referred to a path $p \in \mathcal{P}$ is given by $D_p = \sum_{(i,j)\in A[p]} d_{ij}$. Also consider $\mathcal{P}(\beta) = \{p \in \mathcal{P}$

Figure 3.1: Example of a R-RSP instance, with origin $o = 0$ and destination $t = 3$. The notation $[l_{ij}, u_{ij}]\{d_{ij}\}$ means, respectively, the cost interval $[l_{ij}, u_{ij}]$ and the resource consumption $\{d_{ij}\}$ associated with each arc $(i, j) \in A$.



$\mid D_p \leq \beta\}$, the subset of paths in $\mathcal{P}$ whose resource consumptions are smaller than or equal to β.

Definition 7. A path $p^*(s, \beta) \in \mathcal{P}(\beta)$ is said to be a *β-restricted shortest path* in a scenario $s \in \mathcal{S}$ if it has the smallest cost in s among all paths in $\mathcal{P}(\beta)$, *i.e.*, $p^*(s, \beta) = \arg\min_{p \in \mathcal{P}(\beta)} C_p^s$.

Definition 8. The *β-restricted robust deviation* of a path $p \in \mathcal{P}(\beta)$ in a scenario $s \in \mathcal{S}$, denoted by $r_p^{(s,\beta)}$, is the difference between the cost C_p^s of p in s and the cost of a β-restricted shortest path $p^*(s, \beta) \in \mathcal{P}(\beta)$ in s, *i.e.*, $r_p^{(s,\beta)} = C_p^s - C_{p^*(s,\beta)}^s$.

Definition 9. The *β-restricted robustness cost* of a path $p \in \mathcal{P}(\beta)$, denoted by R_p^β, is the maximum β-restricted robust deviation of p among all possible scenarios, *i.e.*, $R_p^\beta = \max_{s \in \mathcal{S}} r_p^{(s,\beta)}$.

Definition 10. A path $p^* \in \mathcal{P}(\beta)$ is said to be a *β-restricted robust path* if it has the smallest β-restricted robustness cost among all paths in $\mathcal{P}(\beta)$, *i.e.*, $p^* = \arg\min_{p \in \mathcal{P}(\beta)} R_p^\beta$.

Definition 11. R-RSP consists in finding a β-restricted robust path.

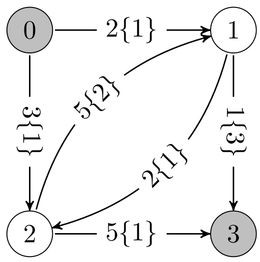
3.2 Mathematical formulation

The mathematical formulation for R-RSP proposed here makes use of Theorem 1 (presented below), which is derived from analogous results for other RO problems, including IRRSP (see, *e.g.*, Karaşan et al. [2001]; Yaman et al. [2001]; Montemanni et al. [2007]).

First, consider the scenario induced by a path $p \in \mathcal{P}$, denoted by s_p, for which each arc in $A[p]$ has its cost set to its upper bound, while all the remaining arcs of A have their cost values set to their corresponding lower bounds, *i.e.*, $c_{ij}^{s_p} = u_{ij} \; \forall \, (i, j) \in A[p]$

and $c_{ij}^{s_p} = l_{ij}\ \forall\,(i,j) \in A\backslash A[p]$. Figure 3.2 shows an example of the scenario $s_{\tilde{p}}$ induced by the path $\tilde{p} = \{0,1,2,3\}$ in the graph presented in Figure 3.1.

Figure 3.2: Scenario $s_{\tilde{p}}$ induced by the path $\tilde{p} = \{0,1,2,3\}$ in the graph presented in Figure 3.1. For each arc $(i,j) \in A$, the notation $c_{ij}^{s_{\tilde{p}}}\{d_{ij}\}$ means, respectively, the arc cost $c_{ij}^{s_{\tilde{p}}}$ in the scenario $s_{\tilde{p}}$ and its resource consumption $\{d_{ij}\}$.



Theorem 1. *For any value $\beta \in \mathbb{R}_+$ and any path $p \in \mathcal{P}(\beta)$, the β-restricted robust deviation of p is maximum in the scenario s_p induced by p.*

Proof. Consider a value $\beta \in \mathbb{R}_+$, a path $p \in \mathcal{P}(\beta)$, and let $s^* \in \mathcal{S}$ be the scenario in which the β-restricted robust deviation of p is maximum, *i.e.*, $R_p^\beta = r_p^{(s^*,\beta)}$. It holds that

$$r_p^{(s^*,\beta)} = C_p^{s^*} - C_{p^*(s^*,\beta)}^{s^*} = \sum_{(i,j)\in A[p]\backslash A[p^*(s^*,\beta)]} c_{ij}^{s^*} - \sum_{(i,j)\in A[p^*(s^*,\beta)]\backslash A[p]} c_{ij}^{s^*}. \tag{3.1}$$

Considering the definition of the scenario s_p induced by p, we have

$$\sum_{(i,j)\in A[p]\backslash A[p^*(s^*,\beta)]} c_{ij}^{s_p} \geq \sum_{(i,j)\in A[p]\backslash A[p^*(s^*,\beta)]} c_{ij}^{s^*}, \tag{3.2}$$

$$\sum_{(i,j)\in A[p^*(s^*,\beta)]\backslash A[p]} c_{ij}^{s_p} \leq \sum_{(i,j)\in A[p^*(s^*,\beta)]\backslash A[p]} c_{ij}^{s^*}. \tag{3.3}$$

Applying (3.2) and (3.3) in (3.1), we obtain

$$r_p^{(s^*,\beta)} \leq \sum_{(i,j)\in A[p]\backslash A[p^*(s^*,\beta)]} c_{ij}^{s_p} - \sum_{(i,j)\in A[p^*(s^*,\beta)]\backslash A[p]} c_{ij}^{s_p} = C_p^{s_p} - C_{p^*(s^*,\beta)}^{s_p}. \tag{3.4}$$

Since $p^*(s_p,\beta)$ is a β-restricted shortest path in s_p, it holds that

$$C^{s_p}_{p^*(s_p,\beta)} \leq C^{s_p}_{p^*(s^*,\beta)}. \tag{3.5}$$

Moreover, from (3.4) and (3.5), it can be deduced that

$$r_p^{(s^*,\beta)} \leq C_p^{s_p} - C^{s_p}_{p^*(s^*,\beta)} \leq C_p^{s_p} - C^{s_p}_{p^*(s_p,\beta)} = r_p^{(s_p,\beta)}. \tag{3.6}$$

Thus, the β-restricted robust deviation of p is maximum in the scenario s_p, *i.e.*, $R_p^\beta = r_p^{(s^*,\beta)} = r_p^{(s_p,\beta)}$. □

For example, consider the graph presented in Figure 3.1 and the scenario $s_{\tilde{p}}$ induced by the path $\tilde{p} = \{0, 1, 2, 3\}$ (as showed in Figure 3.2). Also let the resource limit be $\beta = 3$. According to Theorem 1, the β-restricted robustness cost of $\tilde{p}$ is given by $R^\beta_{\tilde{p}} = r_{\tilde{p}}^{(s_{\tilde{p}},\beta)} = C^{s_{\tilde{p}}}_{\tilde{p}} - C^{s_{\tilde{p}}}_{p^*(s_{\tilde{p}},\beta)} = (2+2+5) - (3+5) = 1$. Theorem 1 reduces R-RSP to finding a path $p^* \in \mathcal{P}(\beta)$ such that $p^* = \arg\min_{p\in\mathcal{P}} r_p^{(s_p,\beta)}$, *i.e.*, $p^* = \arg \min_{p\in\mathcal{P}(\beta)} \{C_p^{s_p} - C^{s_p}_{p^*(s_p,\beta)}\}$. However, notice that computing the β-restricted robustness cost of any feasible solution p for R-RSP implies finding a β-restricted shortest path in the scenario s_p induced by p. Therefore, solely evaluating the cost of a solution requires solving a R-SP instance, *i.e.*, a NP-hard problem.

Let us consider decision variables y on the choice of arcs belonging or not to a β-restricted robust path: $y_{ij} = 1$ if arc $(i,j) \in A$ belongs to the solution path; $y_{ij} = 0$, otherwise. Likewise, let the binary variables x identify a β-restricted shortest path in the scenario induced by the path defined by y, such that $x_{ij} = 1$ if an arc $(i,j) \in A$ belongs to this β-restricted shortest path, and $x_{ij} = 0$, otherwise. A non-linear compact formulation for R-RSP is given as follows:

$$\min_{y\in\mathcal{P}(\beta)} \left(\sum_{(i,j)\in A} u_{ij}y_{ij} - \min_{x\in\mathcal{P}(\beta)} \sum_{(i,j)\in A} (l_{ij} + (u_{ij} - l_{ij})y_{ij})x_{ij} \right). \tag{3.7}$$

In order to derive a MILP formulation from (3.7), we add a free variable ρ and linear constraints that explicitly bound ρ with respect to all the feasible paths that x can represent. The resulting formulation is provided from (3.8) to (3.15).

$$
\begin{aligned}
(\mathcal{F}) \quad \min \quad & \sum_{(i,j)\in A} u_{ij}y_{ij} - \rho && (3.8)\\
s.t. \quad & \sum_{j:(j,o)\in A} y_{jo} - \sum_{k:(o,k)\in A} y_{ok} = -1, && (3.9)\\
& \sum_{j:(j,i)\in A} y_{ji} - \sum_{k:(i,k)\in A} y_{ik} = 0 \quad \forall\, i \in V\backslash\{o,t\}, && (3.10)\\
& \sum_{j:(j,t)\in A} y_{jt} - \sum_{k:(t,k)\in A} y_{tk} = 1, && (3.11)\\
& \sum_{(i,j)\in A} d_{ij}y_{ij} \leq \beta, && (3.12)\\
& \rho \leq \sum_{(i,j)\in A} (l_{ij} + (u_{ij} - l_{ij})y_{ij})\bar{x}_{ij} \quad \forall\, \bar{x} \in \mathcal{P}(\beta), && (3.13)\\
& y_{ij} \in \{0,1\} \quad \forall\,(i,j) \in A, && (3.14)\\
& \rho \text{ free.} && (3.15)
\end{aligned}
$$

The flow conservation constraints (3.9)-(3.11), along with the domain constraints (3.14), ensure that y defines a path from the origin to the destination vertices. In fact, as pointed out in Karaşan et al. [2001], these constraints do not prevent the existence of additional cycles of cost zero disjoint from the solution path. Notice, however, that every arc (i,j) of such cycles necessarily has $l_{ij} = u_{ij} = 0$ and, thus, they do not modify the value of the objective function in (3.8) associated with y. Hence, these cycles are not considered in the remainder of this study.

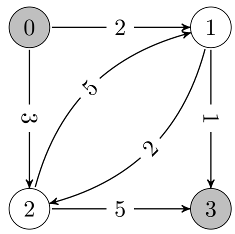
Constraint (3.12) limits the resource consumption of the path defined by y to be at most β, whereas constraints (3.13) guarantee that ρ does not exceed the value related to the inner minimization in (3.7). Note that, in (3.13), $\bar{x}$ is a constant vector, one for each path in $\mathcal{P}(\beta)$. Moreover, constraints (3.13) are tight whenever $\bar{x}$ identifies a β-restricted shortest path in the scenario induced by the path that y defines. Constraint (3.15) gives the domain of variable ρ.

The number of constraints (3.13) grows exponentially, since it corresponds to the number of paths in $\mathcal{P}(\beta)$. This fact discourages applying formulation $\mathcal{F}$ directly to solve large-sized R-RSP instances. However, this fomulation is suitable to be handled by decomposition methods, such as the Benders-like decomposition approach presented in the next section.

3.3 Theoretical results

In this section, we propose new IRRSP results and extend them to R-RSP. In addition, we derive some theoretical R-RSP results inspired by analogous results concerning IRRSP in the literature (see Karaşan et al. [2001]). We first highlight an IRRSP result used to reduce the number of scenarios that must be considered during the search of a robust path.

Figure 3.3: Scenario $s_{\tilde{p}}$ induced by the path $\tilde{p} = \{0, 1, 2, 3\}$ in the graph presented in Figure 2.1. For each arc $(i, j) \in A$, it is shown the arc cost $c_{ij}^{s_{\tilde{p}}}$ in $s_{\tilde{p}}$.



Theorem 2 (Karaşan et al. [2001])**.** *The robust deviation of a path p is maximum in the scenario s_p induced by p, where the costs of all arcs on p are in their corresponding upper bounds and the costs of all other arcs are in their lower bounds.*

Consider, in Figure 3.3, the scenario $s_{\tilde{p}}$ induced by the path $\tilde{p} = \{0, 1, 2, 3\}$ in the graph presented in Figure 2.1. According to Theorem 2, the robustness cost of $\tilde{p}$ is given by $R_{\tilde{p}} = r_{\tilde{p}}^{s_{\tilde{p}}} = C_{\tilde{p}}^{s_{\tilde{p}}} - C_{p^*(s_{\tilde{p}})}^{s_{\tilde{p}}} = (2 + 2 + 5) - (2 + 1) = 6$. Theorem 2 reduces IRRSP to finding a path $p^* \in \mathcal{P}$ such that $p^* = \arg\min_{p \in \mathcal{P}} r_p^{s_p} = \arg\min_{p \in \mathcal{P}} \{C_p^{s_p} - C_{p^*(s_p)}^{s_p}\}$.

Property 1. *Given two arbitrary sets Z_1 and Z_2, it holds that $Z_1 = (Z_1 \cap Z_2) \cup (Z_1 \backslash Z_2)$ and, likewise, that $Z_2 = (Z_1 \cap Z_2) \cup (Z_2 \backslash Z_1)$.*

Theorem 3. *For any non-elementary path $p \in \mathcal{P}$ containing at least one cycle, there is a corresponding elementary path $\tilde{p} \in \mathcal{P}$, such that $A[\tilde{p}] \subset A[p]$ and $r_{\tilde{p}}^{s_{\tilde{p}}} \leq r_p^{s_p}$.*

Proof. Consider a non-elementary path $p \in \mathcal{P}$ containing at least one cycle. Let $G[p]$ be the subgraph of G induced by the arcs in $A[p]$ and $\tilde{p}$ be an elementary path from o to t in $G[p]$. We have $A[\tilde{p}] \subset A[p]$ and, therefore, $\tilde{p} \in \mathcal{P}(\beta)$. By definition,

$$r_{\tilde{p}}^{s_{\tilde{p}}} = C_{\tilde{p}}^{s_{\tilde{p}}} - C_{p^*(s_{\tilde{p}})}^{s_{\tilde{p}}}. \tag{3.16}$$

Consider the set $\bar{A} = A[p] \backslash A[\tilde{p}]$ of the arcs in p which do not belong to $\tilde{p}$. As p is supposed to contain at least one cycle, we have $\bar{A} \neq \emptyset$. Besides, since $A[\tilde{p}] \subset A[p]$, the difference between scenarios s_p and $s_{\tilde{p}}$ consists of the cost values assumed by the arcs in $\bar{A}$. More precisely,

$$c_{ij}^{s_{\tilde{p}}} = c_{ij}^{s_p} \qquad \forall (i,j) \in A \backslash \bar{A}, \tag{3.17}$$
$$c_{ij}^{s_p} = u_{ij} \qquad \forall (i,j) \in \bar{A}, \tag{3.18}$$
$$c_{ij}^{s_{\tilde{p}}} = l_{ij} \qquad \forall (i,j) \in \bar{A}. \tag{3.19}$$

Then, we obtain

$$C_{\tilde{p}}^{s_{\tilde{p}}} = \sum_{(i,j)\in A[\tilde{p}]} u_{ij} = \sum_{(i,j)\in A[p]} u_{ij} - \sum_{(i,j)\in A[p]\backslash A[\tilde{p}]} u_{ij} = C_p^{s_p} - \sum_{(i,j)\in \bar{A}} u_{ij}. \tag{3.20}$$

Applying Property 1 to sets $A[p^*(s_{\tilde{p}})]$ and $A[p]$:

$$A[p^*(s_{\tilde{p}})] = \overbrace{(A[p^*(s_{\tilde{p}})] \cap A[p])}^{(a_1)} \cup (A[p^*(s_{\tilde{p}})] \backslash A[p]). \tag{3.21}$$

Since $A[\tilde{p}] \subset A[p]$ and $\bar{A} = A[p] \backslash A[\tilde{p}]$, it follows that

$$A[p] = A[\tilde{p}] \cup (A[p] \backslash A[\tilde{p}]) = A[\tilde{p}] \cup \bar{A}. \tag{3.22}$$

Therefore, expression (a_1) of (3.21) can be rewritten as

$$(A[p^*(s_{\tilde{p}})] \cap A[p]) = (A[p^*(s_{\tilde{p}})] \cap (A[\tilde{p}] \cup \bar{A})) =$$

$$(A[p^*(s_{\tilde{p}})] \cap A[\tilde{p}]) \cup (A[p^*(s_{\tilde{p}})] \cap \bar{A}). \tag{3.23}$$

Applying (3.21) and (3.23) to $C_{p^*(s_{\tilde{p}})}^{s_{\tilde{p}}}$ and $C_{p^*(s_{\tilde{p}})}^{s_p}$, we obtain:

$$C_{p^*(s_{\tilde{p}})}^{s_{\tilde{p}}} = \sum_{(i,j)\in A[p^*(s_{\tilde{p}})] \cap A[\tilde{p}]} c_{ij}^{s_{\tilde{p}}} + \sum_{(i,j)\in A[p^*(s_{\tilde{p}})] \cap \bar{A}} c_{ij}^{s_{\tilde{p}}} + \sum_{(i,j)\in A[p^*(s_{\tilde{p}})] \backslash A[p]} c_{ij}^{s_{\tilde{p}}}, \tag{3.24}$$
$$C_{p^*(s_{\tilde{p}})}^{s_p} = \sum_{(i,j)\in A[p^*(s_{\tilde{p}})] \cap A[\tilde{p}]} c_{ij}^{s_p} + \sum_{(i,j)\in A[p^*(s_{\tilde{p}})] \cap \bar{A}} c_{ij}^{s_p} + \sum_{(i,j)\in A[p^*(s_{\tilde{p}})] \backslash A[p]} c_{ij}^{s_p}. \tag{3.25}$$

Considering (3.17)-(3.19), (3.24) and (3.25), we deduce that the only difference between the cost of path $p^*(s_{\tilde{p}})$ in s_p and its cost in $s_{\tilde{p}}$ is given by the arcs which are simultaneously in $\bar{A}$ and in $A[p^*(s_{\tilde{p}})]$. Thus, expressions (3.24) and (3.25) can be reformulated as

$$C^{s_{\tilde{p}}}_{p^*(s_{\tilde{p}})} = \sum_{(i,j)\in A[p^*(s_{\tilde{p}})]\cap A[\tilde{p}]} u_{ij} + \sum_{(i,j)\in A[p^*(s_{\tilde{p}})]\cap \bar{A}} l_{ij} + \sum_{(i,j)\in A[p^*(s_{\tilde{p}})]\setminus A[p]} l_{ij}, \tag{3.26}$$

$$C^{s_p}_{p^*(s_{\tilde{p}})} = \sum_{(i,j)\in A[p^*(s_{\tilde{p}})]\cap A[\tilde{p}]} u_{ij} + \sum_{(i,j)\in A[p^*(s_{\tilde{p}})]\cap \bar{A}} u_{ij} + \sum_{(i,j)\in A[p^*(s_{\tilde{p}})]\setminus A[p]} l_{ij}. \tag{3.27}$$

Subtracting (3.27) from (3.26):

$$C^{s_{\tilde{p}}}_{p^*(s_{\tilde{p}})} - C^{s_p}_{p^*(s_{\tilde{p}})} = \sum_{(i,j)\in A[p^*(s_{\tilde{p}})]\cap \bar{A}} l_{ij} - \sum_{(i,j)\in A[p^*(s_{\tilde{p}})]\cap \bar{A}} u_{ij}. \tag{3.28}$$

Therefore,

$$C^{s_{\tilde{p}}}_{p^*(s_{\tilde{p}})} = C^{s_p}_{p^*(s_{\tilde{p}})} - \sum_{(i,j)\in A[p^*(s_{\tilde{p}})]\cap \bar{A}} (u_{ij} - l_{ij}). \tag{3.29}$$

Applying (3.20) and (3.29) in (3.16):

$$r^{(s_{\tilde{p}})}_{\tilde{p}} = C^{s_p}_p - \sum_{(i,j)\in \bar{A}} u_{ij} - \Bigg(C^{s_p}_{p^*(s_{\tilde{p}})} - \sum_{(i,j)\in A[p^*(s_{\tilde{p}})]\cap \bar{A}} (u_{ij} - l_{ij}) \Bigg) =$$

$$C^{s_p}_p - C^{s_p}_{p^*(s_{\tilde{p}})} + \sum_{(i,j)\in A[p^*(s_{\tilde{p}})]\cap \bar{A}} (u_{ij} - l_{ij}) - \sum_{(i,j)\in \bar{A}} u_{ij}. \tag{3.30}$$

One may note that

$$\sum_{(i,j)\in A[p^*(s_{\tilde{p}})]\cap \bar{A}} (u_{ij} - l_{ij}) - \sum_{(i,j)\in \bar{A}} u_{ij} \leq$$

$$\sum_{(i,j)\in \bar{A}} (u_{ij} - l_{ij}) - \sum_{(i,j)\in \bar{A}} u_{ij} \leq \sum_{(i,j)\in \bar{A}} (-l_{ij}). \tag{3.31}$$

Since $\bar{A} \subset A$ and $l_{ij} \geq 0$, $\forall (i,j) \in A$, it follows that $\sum\limits_{(i,j)\in \bar{A}} (-l_{ij}) \leq 0$ and, thus,

$$\sum_{(i,j)\in A[p^*(s_{\tilde{p}})]\cap \bar{A}} (u_{ij} - l_{ij}) - \sum_{(i,j)\in \bar{A}} u_{ij} \leq 0. \tag{3.32}$$

From (3.30) and (3.32),

$$r_{\tilde{p}}^{(s_{\tilde{p}})} \leq C_p^{s_p} - C_{p^*(s_{\tilde{p}})}^{s_p}. \tag{3.33}$$

As $p^*(s_p)$ is a path with the smallest cost in s_p among all the paths in $\mathcal{P}$, including $p^*(s_{\tilde{p}})$, it holds that $C_{p^*(s_{\tilde{p}})}^{s_p} \geq C_{p^*(s_p)}^{s_p}$. Thus,

$$r_{\tilde{p}}^{(s_{\tilde{p}})} \leq C_p^{s_p} - C_{p^*(s_{\tilde{p}})}^{s_p} \leq C_p^{s_p} - C_{p^*(s_p)}^{s_p}. \tag{3.34}$$

By definition, $r_p^{(s_p)} = C_p^{s_p} - C_{p^*(s_p)}^{s_p}$. Therefore, $r_{\tilde{p}}^{(s_{\tilde{p}})} \leq r_p^{(s_p)}$. □

Directly from Theorem 2 and Theorem 3, we obtain the following result.

Corollary 1. *There is an elementary path $p \in \mathcal{P}$ which is a robust path.*

We use the same mathematical argumentation of Theorem 3 to prove that the referred IRRSP result holds for R-RSP.

Theorem 4. *Given a value $\beta \in \mathbb{R}_+$, for any non-elementary path $p \in \mathcal{P}(\beta)$ containing at least one cycle, there is a corresponding elementary path $\tilde{p} \in \mathcal{P}(\beta)$, such that $A[\tilde{p}] \subset A[p]$ and $r_{\tilde{p}}^{(s_{\tilde{p}},\beta)} \leq r_p^{(s_p,\beta)}$.*

Proof. Consider a value $\beta \in \mathbb{R}_+$ and a non-elementary path $p \in \mathcal{P}(\beta)$ containing at least one cycle. Let $G[p]$ be the subgraph of G induced by the arcs in $A[p]$ and $\tilde{p}$ be an elementary path from o to t in $G[p]$. We have $A[\tilde{p}] \subset A[p]$ and, therefore, $\tilde{p} \in \mathcal{P}(\beta)$. By definition,

$$r_{\tilde{p}}^{(s_{\tilde{p}},\beta)} = C_{\tilde{p}}^{s_{\tilde{p}}} - C_{p^*(s_{\tilde{p}},\beta)}^{s_{\tilde{p}}}. \tag{3.35}$$

Consider the set $\bar{A} = A[p]\backslash A[\tilde{p}]$ of the arcs in p which do not belong to $\tilde{p}$. As p is supposed to contain at least one cycle, then $\bar{A} \neq \emptyset$. Moreover, since $A[\tilde{p}] \subset A[p]$, the difference between scenarios s_p and $s_{\tilde{p}}$ consists of the cost values assumed by the arcs in $\bar{A}$. More precisely,

$$c_{ij}^{s_{\tilde{p}}} = c_{ij}^{s_p} \qquad \forall (i,j) \in A \backslash \bar{A}, \tag{3.36}$$
$$c_{ij}^{s_p} = u_{ij} \qquad \forall (i,j) \in \bar{A}, \tag{3.37}$$
$$c_{ij}^{s_{\tilde{p}}} = l_{ij} \qquad \forall (i,j) \in \bar{A}. \tag{3.38}$$

It follows that

$$C_{\tilde{p}}^{s_{\tilde{p}}} = \sum_{(i,j)\in A[\tilde{p}]} u_{ij} = \sum_{(i,j)\in A[p]} u_{ij} - \sum_{(i,j)\in A[p]\backslash A[\tilde{p}]} u_{ij} = C_p^{s_p} - \sum_{(i,j)\in \bar{A}} u_{ij}. \tag{3.39}$$

Applying Property 1 to sets $A[p^*(s_{\tilde{p}}, \beta)]$ and $A[p]$:

$$A[p^*(s_{\tilde{p}}, \beta)] = \overbrace{(A[p^*(s_{\tilde{p}}, \beta)] \cap A[p])}^{(a_2)} \cup (A[p^*(s_{\tilde{p}}, \beta)] \backslash A[p]). \tag{3.40}$$

Since $A[\tilde{p}] \subset A[p]$ and $\bar{A} = A[p]\backslash A[\tilde{p}]$, it follows that

$$A[p] = A[\tilde{p}] \cup (A[p]\backslash A[\tilde{p}]) = A[\tilde{p}] \cup \bar{A}. \tag{3.41}$$

Therefore, expression (a_2) of (3.40) can be rewritten as

$$(A[p^*(s_{\tilde{p}}, \beta)] \cap A[p]) = (A[p^*(s_{\tilde{p}}, \beta)] \cap (A[\tilde{p}] \cup \bar{A})) =$$

$$(A[p^*(s_{\tilde{p}}, \beta)] \cap A[\tilde{p}]) \cup (A[p^*(s_{\tilde{p}}, \beta)] \cap \bar{A}). \tag{3.42}$$

Applying (3.40) and (3.42) to $C_{p^*(s_{\tilde{p}},\beta)}^{s_{\tilde{p}}}$ and $C_{p^*(s_{\tilde{p}},\beta)}^{s_p}$, we obtain:

$$C_{p^*(s_{\tilde{p}},\beta)}^{s_{\tilde{p}}} = \sum_{(i,j)\in A[p^*(s_{\tilde{p}},\beta)]\cap A[\tilde{p}]} c_{ij}^{s_{\tilde{p}}} + \sum_{(i,j)\in A[p^*(s_{\tilde{p}},\beta)]\cap \bar{A}} c_{ij}^{s_{\tilde{p}}} + \sum_{(i,j)\in A[p^*(s_{\tilde{p}},\beta)]\backslash A[p]} c_{ij}^{s_{\tilde{p}}}, \tag{3.43}$$
$$C_{p^*(s_{\tilde{p}},\beta)}^{s_p} = \sum_{(i,j)\in A[p^*(s_{\tilde{p}},\beta)]\cap A[\tilde{p}]} c_{ij}^{s_p} + \sum_{(i,j)\in A[p^*(s_{\tilde{p}},\beta)]\cap \bar{A}} c_{ij}^{s_p} + \sum_{(i,j)\in A[p^*(s_{\tilde{p}},\beta)]\backslash A[p]} c_{ij}^{s_p}. \tag{3.44}$$

Considering (3.36)-(3.38), (3.43) and (3.44), we deduce that the only difference between the cost of path $p^*(s_{\tilde{p}}, \beta)$ in s_p and its cost in $s_{\tilde{p}}$ is given by the arcs which are simultaneously in $\bar{A}$ and in $A[p^*(s_{\tilde{p}}, \beta)]$. Thus, expressions (3.43) and (3.44) can be reformulated as

$$C^{s_{\tilde{p}}}_{p^*(s_{\tilde{p}},\beta)} = \sum_{(i,j)\in A[p^*(s_{\tilde{p}},\beta)]\cap A[\tilde{p}]} u_{ij} + \sum_{(i,j)\in A[p^*(s_{\tilde{p}},\beta)]\cap \bar{A}} l_{ij} + \sum_{(i,j)\in A[p^*(s_{\tilde{p}},\beta)]\setminus A[p]} l_{ij}, \quad (3.45)$$

$$C^{s_p}_{p^*(s_{\tilde{p}},\beta)} = \sum_{(i,j)\in A[p^*(s_{\tilde{p}},\beta)]\cap A[\tilde{p}]} u_{ij} + \sum_{(i,j)\in A[p^*(s_{\tilde{p}},\beta)]\cap \bar{A}} u_{ij} + \sum_{(i,j)\in A[p^*(s_{\tilde{p}},\beta)]\setminus A[p]} l_{ij}. \quad (3.46)$$

Subtracting (3.46) from (3.45),

$$C^{s_{\tilde{p}}}_{p^*(s_{\tilde{p}},\beta)} - C^{s_p}_{p^*(s_{\tilde{p}},\beta)} = \sum_{(i,j)\in A[p^*(s_{\tilde{p}},\beta)]\cap \bar{A}} l_{ij} - \sum_{(i,j)\in A[p^*(s_{\tilde{p}},\beta)]\cap \bar{A}} u_{ij}. \quad (3.47)$$

Therefore,

$$C^{s_{\tilde{p}}}_{p^*(s_{\tilde{p}},\beta)} = C^{s_p}_{p^*(s_{\tilde{p}},\beta)} - \sum_{(i,j)\in A[p^*(s_{\tilde{p}},\beta)]\cap \bar{A}} (u_{ij} - l_{ij}). \quad (3.48)$$

Applying (3.39) and (3.48) in (3.35):

$$r^{(s_{\tilde{p}},\beta)}_{\tilde{p}} = C^{s_p}_{p} - \sum_{(i,j)\in \bar{A}} u_{ij} - \Big(C^{s_p}_{p^*(s_{\tilde{p}},\beta)} - \sum_{(i,j)\in A[p^*(s_{\tilde{p}},\beta)]\cap \bar{A}} (u_{ij} + l_{ij})\Big) =$$

$$C^{s_p}_{p} - C^{s_p}_{p^*(s_{\tilde{p}},\beta)} + \sum_{(i,j)\in A[p^*(s_{\tilde{p}},\beta)]\cap \bar{A}} (u_{ij} - l_{ij}) - \sum_{(i,j)\in \bar{A}} u_{ij}. \quad (3.49)$$

One may note that

$$\sum_{(i,j)\in A[p^*(s_{\tilde{p}},\beta)]\cap \bar{A}} (u_{ij} - l_{ij}) - \sum_{(i,j)\in \bar{A}} u_{ij} \leq$$

$$\sum_{(i,j)\in \bar{A}} (u_{ij} - l_{ij}) - \sum_{(i,j)\in \bar{A}} u_{ij} \leq \sum_{(i,j)\in \bar{A}} (-l_{ij}). \quad (3.50)$$

Since $\bar{A} \subset A$ and $l_{ij} \geq 0 \ \forall (i,j) \in A$, it follows that $\sum_{(i,j)\in \bar{A}} (-l_{ij}) \leq 0$ and, thus,

$$\sum_{(i,j)\in A[p^*(s_{\tilde{p}})]\cap\bar{A}} (u_{ij} - l_{ij}) - \sum_{(i,j)\in\bar{A}} u_{ij} \leq 0. \tag{3.51}$$

From (3.49) and (3.51),

$$r_{\tilde{p}}^{(s_{\tilde{p}},\beta)} \leq C_p^{s_p} - C_{p^*(s_{\tilde{p}},\beta)}^{s_p}. \tag{3.52}$$

As $p^*(s_p, \beta)$ is a path with the smallest cost in s_p among all the paths in $\mathcal{P}(\beta)$, including $p^*(s_{\tilde{p}}, \beta)$, it holds that $C_{p^*(s_{\tilde{p}},\beta)}^{s_p} \geq C_{p^*(s_p,\beta)}^{s_p}$. Thus,

$$r_{\tilde{p}}^{(s_{\tilde{p}},\beta)} \leq C_p^{s_p} - C_{p^*(s_{\tilde{p}},\beta)}^{s_p} \leq C_p^{s_p} - C_{p^*(s_p,\beta)}^{s_p}. \tag{3.53}$$

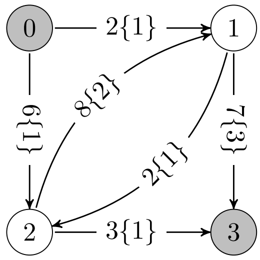
By definition, $r_p^{(s_p,\beta)} = C_p^{s_p} - C_{p^*(s_p,\beta)}^{s_p}$. Therefore, $r_{\tilde{p}}^{(s_{\tilde{p}},\beta)} \leq r_p^{(s_p,\beta)}$. □

Directly from Theorem 1 and Theorem 4, we obtain the following result.

Corollary 2. *For any $\beta \in \mathbb{R}_+$, there is an elementary path $p \in \mathcal{P}(\beta)$ which is β-restricted robust path.*

Consider the scenario co-induced by a path $p \in \mathcal{P}$, denoted by $\bar{s}_p$, for which each arc in $A[p]$ has its cost set to its lower bound, while all the remaining arcs of A have their cost values set to their corresponding upper bounds, *i.e*, $c_{ij}^{\bar{s}_p} = l_{ij}\ \forall (i,j) \in A[p]$ and $c_{ij}^{\bar{s}_p} = u_{ij}\ \forall (i,j) \in A \backslash A[p]$. Figure 3.4 shows an example of the scenario $\bar{s}_{\tilde{p}}$ co-induced by the path $\tilde{p} = \{0, 1, 2, 3\}$ in the graph presented in Figure 3.1.

Figure 3.4: Scenario $\bar{s}_{\tilde{p}}$ co-induced by the path $\tilde{p} = \{0, 1, 2, 3\}$ in the graph presented in Figure 3.1. For each arc $(i, j) \in A$, the notation $c_{ij}^{\bar{s}_{\tilde{p}}}\{d_{ij}\}$ denotes, respectively, the arc cost $c_{ij}^{\bar{s}_{\tilde{p}}}$ in the scenario $\bar{s}_{\tilde{p}}$ and its resource consumption $\{d_{ij}\}$.



Lemma 1. *Given a value $\beta \in \mathbb{R}_+$ and a path $p \in \mathcal{P}(\beta)$, if there is a path $p' \in \mathcal{P}(\beta)$ such that $C_{p'}^{\bar{s}_p} < C_p^{\bar{s}_p}$, then $C_{p'}^{s} < C_p^{s}$ in every scenario $s \in \mathcal{S}$.*

Proof. Consider a value $\beta \in \mathbb{R}_+$ and a path $p \in \mathcal{P}(\beta)$. Suppose that there is a path $p' \in \mathcal{P}(\beta)$ such that $C_{p'}^{\bar{s}_p} < C_p^{\bar{s}_p}$. Applying Property 1 to sets $A[p]$ and $A[p']$, we obtain

$$A[p] = (A[p] \cap A[p']) \cup (A[p] \backslash A[p']), \tag{3.54}$$

$$A[p'] = (A[p'] \cap A[p]) \cup (A[p'] \backslash A[p]). \tag{3.55}$$

Therefore, the costs $C_{p'}^{\bar{s}_p}$ and $C_p^{\bar{s}_p}$ can be expressed as

$$C_p^{\bar{s}_p} = \sum_{(i,j) \in A[p] \cap A[p']} c_{ij}^{\bar{s}_p} + \sum_{(i,j) \in A[p] \backslash A[p']} c_{ij}^{\bar{s}_p}, \tag{3.56}$$

$$C_{p'}^{\bar{s}_p} = \sum_{(i,j) \in A[p'] \cap A[p]} c_{ij}^{\bar{s}_p} + \sum_{(i,j) \in A[p'] \backslash A[p]} c_{ij}^{\bar{s}_p}. \tag{3.57}$$

Applying (3.56) and (3.57) to the assumption that $C_{p'}^{\bar{s}_p} < C_p^{\bar{s}_p}$, we have

$$\sum_{(i,j) \in A[p'] \cap A[p]} c_{ij}^{\bar{s}_p} + \sum_{(i,j) \in A[p'] \backslash A[p]} c_{ij}^{\bar{s}_p} < \sum_{(i,j) \in A[p] \cap A[p']} c_{ij}^{\bar{s}_p} + \sum_{(i,j) \in A[p] \backslash A[p']} c_{ij}^{\bar{s}_p}, \tag{3.58}$$

which implies

$$\sum_{(i,j) \in A[p'] \backslash A[p]} c_{ij}^{\bar{s}_p} < \sum_{(i,j) \in A[p] \backslash A[p']} c_{ij}^{\bar{s}_p}. \tag{3.59}$$

Let s be any scenario in $\mathcal{S}$. By the definition of the scenario $\bar{s}_p$, we have

$$\sum_{(i,j) \in A[p'] \backslash A[p]} c_{ij}^{\bar{s}_p} = \sum_{(i,j) \in A[p'] \backslash A[p]} u_{ij} \geq \sum_{(i,j) \in A[p'] \backslash A[p]} c_{ij}^{s}, \tag{3.60}$$

$$\sum_{(i,j) \in A[p] \backslash A[p']} c_{ij}^{\bar{s}_p} = \sum_{(i,j) \in A[p] \backslash A[p']} l_{ij} \leq \sum_{(i,j) \in A[p] \backslash A[p']} c_{ij}^{s}. \tag{3.61}$$

Using (3.60) and (3.61) in (3.59):

$$\sum_{(i,j)\in A[p']\setminus A[p]} c_{ij}^{s} \leq \sum_{(i,j)\in A[p']\setminus A[p]} c_{ij}^{\bar{s}_p} < \sum_{(i,j)\in A[p]\setminus A[p']} c_{ij}^{\bar{s}_p} \leq \sum_{(i,j)\in A[p]\setminus A[p']} c_{ij}^{s}. \tag{3.62}$$

By adding $\sum_{(i,j)\in A[p']\cap A[p]} c_{ij}^{s}$ to each side of the inequality (3.62) and by considering Property 1, we obtain:

$$C_{p'}^{s} = \sum_{(i,j)\in A[p']\setminus A[p]} c_{ij}^{s} + \sum_{(i,j)\in A[p']\cap A[p]} c_{ij}^{s} < \sum_{(i,j)\in A[p]\setminus A[p']} c_{ij}^{s} + \sum_{(i,j)\in A[p']\cap A[p]} c_{ij}^{s} = C_{p}^{s}.$$

□

Proposition 1. *Given a value $\beta \in \mathbb{R}_+$ and a path $p \in \mathcal{P}(\beta)$, if p is a β-restricted robust path, then there is a scenario $s \in \mathcal{S}$ in which p is a β-restricted shortest path.*

Proof. Consider a value $\beta \in \mathbb{R}_+$ and a path $p \in \mathcal{P}(\beta)$. We prove that the counterpositive holds, *i.e.*, if there is no scenario $s' \in \mathcal{S}$ in which p is a β-restricted shortest path, then p is not a β-restricted robust path. In this sense, let $p' \in \mathcal{P}(\beta)$ be a β-restricted shortest path in the scenario $\bar{s}_p$ co-induced by p and suppose that there is no scenario $s' \in \mathcal{S}$ (including $\bar{s}_p$) in which p is a β-restricted shortest path. From this assumption, it follows that $C_{p'}^{\bar{s}_p} < C_{p}^{\bar{s}_p}$ and, according to Lemma 1,

$$C_{p'}^{s} < C_{p}^{s} \qquad \forall s \in \mathcal{S}. \tag{3.63}$$

Let $p^*(s_{p'}, \beta) \in \mathcal{P}(\beta)$ be a β-restricted shortest path in the scenario $s_{p'}$ induced by p'. Considering (3.63) for the scenario $s_{p'}$ and subtracting the same value $C_{p^*(s_{p'},\beta)}^{s_{p'}}$ from each side of the resulting inequality, we obtain:

$$C_{p'}^{s_{p'}} - C_{p^*(s_{p'},\beta)}^{s_{p'}} < C_{p}^{s_{p'}} - C_{p^*(s_{p'},\beta)}^{s_{p'}}. \tag{3.64}$$

According to Theorem 1:

$$R_{p'}^{\beta} = r_{p'}^{(s_{p'},\beta)} = C_{p'}^{s_{p'}} - C_{p^*(s_{p'},\beta)}^{s_{p'}}, \tag{3.65}$$

$$R_{p}^{\beta} = r_{p}^{(s_p,\beta)} = C_{p}^{s_p} - C_{p^*(s_p,\beta)}^{s_p} \geq C_{p}^{s} - C_{p^*(s,\beta)}^{s} \qquad \forall s \in \mathcal{S}. \tag{3.66}$$

From (3.64)-(3.66), it follows that:

$$R_{p'}^{\beta} = C_{p'}^{s_{p'}} - C_{p^*(s_{p'},\beta)}^{s_{p'}} < C_{p}^{s_{p'}} - C_{p^*(s_{p'},\beta)}^{s_{p'}} \leq C_{p}^{s_{p}} - C_{p^*(s_{p},\beta)}^{s_{p}} = R_{p}^{\beta}. \quad (3.67)$$

Therefore, $R_{p'}^{\beta} < R_{p}^{\beta}$, which implies that p is not a β-restricted robust path.

□

3.4 Computational complexity

R-SP is as a special case of R-RSP, in which $l_{ij} = u_{ij} \; \forall \, (i, j) \in A$. Therefore, R-RSP, as R-SP, is NP-hard, even for acyclic digraphs [Wang and Crowcroft, 1996]. Furthermore, we show that the decision problem associated with R-RSP, namely Decision-R-RSP, is also NP-hard. Considering the R-RSP definitions presented in Section 3.1, Decision-R-RSP is defined as follows.

Input: A digraph $G = (V, A)$ such that each arc $(i, j) \in A$ is associated with a resource consumption $d_{ij} \in \mathbb{R}_+$ and with a cost interval $[l_{ij}, u_{ij}]$, where $l_{ij} \in \mathbb{Z}_+$ is the lower bound and $u_{ij} \in \mathbb{Z}_+$ is the upper bound, $l_{ij} \leq u_{ij}$. An origin $o \in V$ and a destination $t \in V$ are given, as well as a value $\beta \in \mathbb{R}_+$ and a value $k \in \mathbb{Z}_+$. For short, $\langle G, l, u, d, o, t, \beta, k \rangle$.

Question: Is there a path p from o to t in G such that $D_p \leq \beta$ and $R_p^{\beta} \leq k$?

We reduce the decision version of IRRSP, namely Decision-IRRSP, which is NP-complete [Zieliński, 2004], to Decision-R-RSP. Considering the IRRSP definitions presented in Section 2.2, Decision-IRRSP is defined as follows.

Input: A digraph $G = (V, A)$ such that each arc $(i, j) \in A$ is associated with a cost interval $[l_{ij}, u_{ij}]$, where $l_{ij} \in \mathbb{Z}_+$ is the lower bound and $u_{ij} \in \mathbb{Z}_+$ is the upper bound, $l_{ij} \leq u_{ij}$. An origin $o \in V$ and a destination $t \in V$ are given, as well as a value $k \in \mathbb{Z}_+$. For short, $\langle G, l, u, o, t, k \rangle$.

Question: Is there a path p from o to t in G such that $R_p \leq k$?

Consider the following claims.

Claim 1. *For any $\beta \in \mathbb{R}_+$ and any scenario $s \in S$, if there exists a β-restricted shortest path in s from o to t, then there is a β-restricted shortest path in s from o to t that is also an elementary path.*

Claim 2. *For any scenario $s \in S$, there is a minimum cost path in s from o to t that is an elementary path.*

Theorem 5. *Decision-IRRSP $\leq_p$ Decision-R-RSP.*

Proof. First, consider the function $f : \{\langle G, l, u, o, t, k\rangle\} \mapsto \{\langle G', l', u', d', o', t', \beta, k'\rangle\}$ that maps Decision-IRRSP instances to Decision-R-RSP instances through the following reduction:

$$G' = (V', A') = (V, A) = G, \tag{3.68}$$

$$l'_{ij} = l_{ij}, \quad \forall (i,j) \in A' = A, \tag{3.69}$$

$$u'_{ij} = u_{ij}, \quad \forall (i,j) \in A' = A, \tag{3.70}$$

$$d'_{ij} = 1, \quad \forall (i,j) \in A' = A, \tag{3.71}$$

$$o' = o, \tag{3.72}$$

$$t' = t, \tag{3.73}$$

$$\beta = |V'| - 1 = |V| - 1, \tag{3.74}$$

$$k' = k. \tag{3.75}$$

Notice that the asymptotic time complexity of such reduction is $O\big(|V| + |A|\big)$, and, thus, a Decision-IRRSP instance is polinomially reducible to a Decision-R-RSP instance. For the sake of simplicity, we consider $f(\langle G, l, u, o, t, k\rangle) = \langle G', l', u', d', o', t', \beta, k'\rangle = \langle G, l, u, d, o, t, |V|, k\rangle$ in the following. Now, we have to prove that Decision-IRRSP is satisfied for a given $\langle G, l, u, o, t, k\rangle$ if, and only if, Decision-R-RSP is satisfied for the corresponding $\langle G, l, u, d, o, t, |V|, k\rangle$.

$\Rightarrow$ Given a Decision-IRRSP instance $\langle G, l, u, o, t, k\rangle$, suppose that there exists a path p from o to t such that $R_p \leq k$. Two possibilities exist:

- If p is an elementary path, then it has at most $|V| - 1$ arcs. Notice that, from (3.74), $\beta = |V| - 1$. Moreover, from (3.71), $D_p \leq |V| - 1 = \beta$. Consider Claim 2 and let $p^*(s_p) \in \mathcal{P}$ be a minimum cost path in s_p that is also an elementary path. Also let $p^*(s_p, \beta) \in \mathcal{P}(\beta)$ be a β-restricted shortest path in s_p. We have that $D_{p^*(s_p)} \leq \beta$ and, moreover, that $C^{s_p}_{p^*(s_p,\beta)} = C^{s_p}_{p^*(s_p)}$. Considering Theorem 1 and Theorem 2, it follows that

$$R^{\beta}_p = r^{(s_p,\beta)}_p = C^{s_p}_p - C^{s_p}_{p^*(s_p,\beta)} = C^{s_p}_p - C^{s_p}_{p^*(s_p)} = r^{s_p}_p = R_p \leq k,$$

and, thus, p satisfies Decision-R-RSP for $\langle G, l, u, d, o, t, |V|, k\rangle$;

- If p is not an elementary path, then, from Theorem 3, there exists an elementary path $\tilde{p} \in \mathcal{P}$ in G such that $r_{\tilde{p}}^{s_{\tilde{p}}} \leq r_p^{s_p}$. Consider Claim 2 and let $p^*(s_{\tilde{p}}) \in \mathcal{P}$ be a minimum cost path in $s_{\tilde{p}}$ that is also elementary. From (3.74), consider $\beta = |V| - 1$ and let $p^*(s_{\tilde{p}}, \beta) \in \mathcal{P}(\beta)$ be a β-restricted shortest path in $s_{\tilde{p}}$. Notice that $D_{\tilde{p}} \leq |V| - 1$ and that $D_{p^*(s_{\tilde{p}})} \leq |V| - 1$. Then, it follows that $C_{p^*(s_{\tilde{p}},\beta)}^{s_{\tilde{p}}} = C_{p^*(s_{\tilde{p}})}^{s_{\tilde{p}}}$. Considering Theorem 1 and Theorem 2, we obtain

$$R_{\tilde{p}}^{\beta} = r_{\tilde{p}}^{(s_{\tilde{p}},\beta)} = C_{\tilde{p}}^{s_{\tilde{p}}} - C_{p^*(s_{\tilde{p}},\beta)}^{s_{\tilde{p}}} = C_{\tilde{p}}^{s_{\tilde{p}}} - C_{p^*(s_{\tilde{p}})}^{s_{\tilde{p}}} = r_{\tilde{p}}^{s_{\tilde{p}}} \leq r_p^{s_p} = R_p \leq k,$$

  and, thus, $\tilde{p}$ satisfies Decision-R-RSP for $\langle G, l, u, d, o, t, |V|, k \rangle$.

Therefore, if Decision-IRRSP is satisfied for a given $\langle G, l, u, o, t, k \rangle$, then Decision-R-RSP is also satisfied for the corresponding $\langle G, l, u, d, o, t, |V|, k \rangle$.

$\Leftarrow$ Now, take a Decision-R-RSP instance $\langle G, l, u, d, o, t, \beta, k \rangle$, with $\beta = |V| - 1$, and suppose that there exists a path p' from o to t in G such that $R_{p'}^{\beta} \leq k$. Consider Claim 1 and let $p^*(s_{p'}, \beta) \in \mathcal{P}(\beta)$ be a β-restricted shortest path in $s_{p'}$ that is also elementary. Moreover, consider Claim 2 and let $p^*(s_{p'}) \in \mathcal{P}$ be a minimum cost path in $s_{p'}$ that is also elementary. From (3.71) and (3.74), any elementary path from o to t in G belongs to $\mathcal{P}(\beta)$, which implies $C_{p^*(s_{p'})}^{s_{p'}} = C_{p^*(s_{p'},\beta)}^{s_{p'}}$. Considering Theorem 1 and Theorem 2, we obtain

$$R_{p'} = r_{p'}^{s_{p'}} = C_{p'}^{s_{p'}} - C_{p^*(s_{p'})}^{s_{p'}} = C_{p'}^{s_{p'}} - C_{p^*(s_{p'},\beta)}^{s_{p'}} = r_{p'}^{(s_{p'},\beta)} = R_{p'}^{\beta} \leq k.$$

Therefore, p' satisfies Desicion-RSP for $\langle G, l, u, o, t, k \rangle$. Moreover, if Decision-R-RSP is satisfied for a given $\langle G, l, u, d, o, t, |V|, k \rangle$, then Decision-IRRSP is also satisfied for the corresponding $\langle G, l, u, o, t, k \rangle$. $\square$

From Theorem 5, we obtain the computational complexity of Decision-R-RSP.

Corollary 3. *Decision-R-RSP is NP-hard.*

Whether Decision-R-RSP belongs or not to NP is still an open issue.

Chapter 4

A linear programming based heuristic

In this Chapter, we present (as a case study on solving R-RSP) a Linear Programming based Heuristic, namely LPH, to tackle robust-hard problems. It consists in solving a MILP formulation based on the dual of the linear relaxation of the classical optimization problem counterpart. In the case of R-RSP, the proposed heuristic uses dual information regarding the linear relaxation of R-SP to deal with the compact non-linear formulation (3.7), from which formulation $\mathcal{F}$, (3.8)-(3.15), is derived. For this purpose, consider the following R-SP ILP formulation used to compute a β-restricted shortest path $p^*(s,\beta) \in \mathcal{P}(\beta)$ in a scenario $s \in \mathcal{S}$. The binary variables x define $p^*(s,\beta)$, such that $x_{ij} = 1$ if an arc $(i,j) \in A$ belongs to $A[p^*(s,\beta)]$, and $x_{ij} = 0$, otherwise.

$$
\begin{aligned}
(\mathcal{I}) \quad \min \quad & \sum_{(i,j)\in A} c^s_{ij} x_{ij} && (4.1)\\
s.t. \quad & \sum_{j:(j,o)\in A} x_{jo} - \sum_{k:(o,k)\in A} x_{ok} = -1, && (4.2)\\
& \sum_{j:(j,i)\in A} x_{ji} - \sum_{k:(i,k)\in A} x_{ik} = 0 \quad \forall\, i \in V \backslash \{o,t\}, && (4.3)\\
& \sum_{j:(j,t)\in A} x_{jt} - \sum_{k:(t,k)\in A} x_{tk} = 1, && (4.4)\\
& \sum_{(i,j)\in A} d_{ij} x_{ij} \leq \beta, && (4.5)\\
& x_{ij} \in \{0,1\} \quad \forall\, (i,j) \in A. && (4.6)
\end{aligned}
$$

The objective function in (4.1) represents the cost, in the scenario s, of a path defined by x, while constraints (4.2)-(4.6) ensure that x identifies a path in $\mathcal{P}(\beta)$. Relaxing the integrality on x, we obtain the following Linear Programming (LP) formulation:

$$
\begin{aligned}
(\mathcal{L}) \quad \theta^{(s,\beta)} = \min \quad & \sum_{(i,j)\in A} c_{ij}^s x_{ij} && (4.7)\\
s.t. \quad & \text{Constraints (4.2)-(4.5)}, \\
& x_{ij} \geq 0 \quad \forall\,(i,j) \in A. && (4.8)
\end{aligned}
$$

The domain constraints $x_{ij} \leq 1 \; \forall\,(i,j) \in A$ were omitted from $\mathcal{L}$, since they are redundant. Let $\theta^{(s,\beta)}$ be the optimal value for the problem $\mathcal{L}$ in a scenario s. Observe that $\theta^{(s,\beta)}$ provides a lower bound on the solution of $\mathcal{I}$ in the scenario s. We can then define a new metric to evaluate the quality of a path in $\mathcal{P}(\beta)$.

Definition 12. The *β-heuristic robustness cost* of a path $p \in \mathcal{P}(\beta)$, denoted by H_p^β, is the difference between the cost $C_p^{s_p}$ of p in the scenario s_p induced by p and the relaxed cost $\theta^{(s_p,\beta)}$ in s_p, i.e., $H_p^\beta = C_p^{s_p} - \theta^{(s_p,\beta)}$.

Proposition 2. *For any path $p \in \mathcal{P}(\beta)$, the β-heuristic robustness cost of p gives an upper bound on the β-restricted robustness cost of p.*

Proof. Consider a path $p \in \mathcal{P}(\beta)$. According to Theorem 1, the β-restricted robustness cost of p is given by $R_p^\beta = r_p^{(s_p,\beta)} = C_p^{s_p} - C_{p^*(s_p,\beta)}^{s_p}$, where $p^*(s_p,\beta) \in \mathcal{P}(\beta)$ is a β-restricted shortest path in s_p. One may note that $\theta^{(s,\beta)} \leq C_{p^*(s,\beta)}^s$ for any scenario $s \in \mathcal{S}$, including s_p. Hence,

$$H_p^\beta = C_p^{s_p} - \theta^{(s_p,\beta)} \geq C_p^{s_p} - C_{p^*(s_p,\beta)}^{s_p} = R_p^\beta. \tag{4.9}$$

□

Definition 13. A path $\tilde{p}^* \in \mathcal{P}(\beta)$ is said to be a *β-heuristic robust path* if it has the smallest β-heuristic robustness cost among all the paths in $\mathcal{P}(\beta)$, *i.e.*, $\tilde{p}^* = \arg\min_{p\in\mathcal{P}(\beta)} H_p^\beta$.

The heuristic proposed in this work aims at finding a β-heuristic robust path and relies on the hypothesis that such a path is a near-optimal solution for R-RSP. The problem of finding a β-heuristic robust path can be modeled by adapting formulation (3.7). In this sense, the binary variables y now represent a β-heuristic robust path in

$\mathcal{P}(\beta)$. Furthermore, considering the scenario s_y induced by the path defined by y, the nested minimization in (3.7) is replaced by $\theta^{(s_y,\beta)}$. We obtain:

$$\min_{y\in\mathcal{P}(\beta)} \left(\sum_{(i,j)\in A} u_{ij}y_{ij} - \theta^{(s_y,\beta)} \right). \tag{4.10}$$

Given a scenario $s \in \mathcal{S}$, the value assumed by $\theta^{(s,\beta)}$ can be represented by the dual of $\mathcal{L}$, as follows:

$$(\tilde{\mathcal{L}}) \quad \theta^{(s,\beta)} = \max \quad \lambda_t - \lambda_o - \beta\mu \tag{4.11}$$

$$s.t. \quad \lambda_j \leq \lambda_i + c^s_{ij} + d_{ij}\mu \quad \forall\,(i,j) \in A, \tag{4.12}$$

$$\mu \geq 0, \tag{4.13}$$

$$\lambda_k \text{ free} \quad \forall\, k \in V. \tag{4.14}$$

The dual variables $\{\lambda_k : k \in V\}$ and μ are associated, respectively, with constraints (4.2)-(4.4) and with constraint (4.5) in the primal problem $\mathcal{L}$. Since $\tilde{\mathcal{L}}$ is a maximization problem, its objective function, along with (4.12)-(4.14), can be used to replace the relaxed cost $\theta^{(s_y,\beta)}$ in (4.10), thus deriving the following formulation:

$$\min_{y\in\mathcal{P}(\beta)} \left(\sum_{(i,j)\in A} u_{ij}y_{ij} - \overbrace{(\lambda_t - \lambda_o - \beta\mu)}^{\text{From (4.11)}} \right) \tag{4.15}$$

$$s.t. \quad \lambda_j \leq \lambda_i + l_{ij} + (u_{ij} - l_{ij})y_{ij} + d_{ij}\mu \quad \forall\,(i,j) \in A, \tag{4.16}$$

$$\mu \geq 0, \tag{4.17}$$

$$\lambda_k \text{ free} \quad \forall\, k \in V. \tag{4.18}$$

Notice that constraints (4.16) consider the cost of each arc $(i,j) \in A$ in the scenario s_y induced by the path identified by variables y, *i.e.*, the cost of each arc $(i,j) \in A$ is given by $l_{ij} + (u_{ij} - l_{ij})y_{ij}$. The domain constraints (4.17) and (4.18) related to $\tilde{\mathcal{L}}$ remain the same. Now, we give a MILP formulation for the problem of finding a β-heuristic robust path.

$$
\begin{aligned}
(\mathcal{H}) \quad \min \quad & \sum_{(i,j)\in A} u_{ij} y_{ij} - \lambda_t + \lambda_o + \beta\mu && (4.19)\\
s.t. \quad & \sum_{j:(j,o)\in A} y_{jo} - \sum_{k:(o,k)\in A} y_{ok} = -1, && (4.20)\\
& \sum_{j:(j,i)\in A} y_{ji} - \sum_{k:(i,k)\in A} y_{ik} = 0 \quad \forall\, i \in V \backslash \{o, t\}, && (4.21)\\
& \sum_{j:(j,t)\in A} y_{jt} - \sum_{k:(t,k)\in A} y_{tk} = 1, && (4.22)\\
& \sum_{(i,j)\in A} d_{ij} y_{ij} \leq \beta, && (4.23)\\
& \text{Constraints (4.16)-(4.18)}, && \\
& y_{ij} \in \{0,1\} \quad \forall\,(i,j) \in A. && (4.24)
\end{aligned}
$$

The objective function in (4.19) represents the β-heuristic robustness cost of the path defined by the variables y. Constraints (4.20)-(4.23) and (4.24) ensure that y belongs to $\mathcal{P}(\beta)$. Constraints (4.16)-(4.18) are the remaining restrictions related to $\tilde{\mathcal{L}}$.

The heuristic consists in solving the corresponding $\mathcal{H}$ problem in order to find a β-heuristic robust path $\tilde{p}^* \in \mathcal{P}(\beta)$. Note that $\tilde{p}^*$ is also a feasible solution path for R-RSP, and, according to Proposition 2, its β-heuristic robustness cost provides an upper bound on the solution of R-RSP. Such bound is improved by evaluating the actual β-restricted robustness cost of $\tilde{p}^*$. In this study, we solve $\mathcal{H}$ directly with an optimization solver.

Chapter 5

Benders-like decomposition approaches

Benders decomposition method was originally proposed by Benders [1962] (see also Geoffrion [1972]). Later on, methodologies able to improve such method were studied by McDaniel and Devine [1977]; Magnanti and Wong [1981]; Fischetti et al. [2010]. In this chapter, we adapt to R-RSP a Benders-like decomposition approach that is a state-of-the-art method used to solve RO problems [Montemanni and Gambardella, 2005; Montemanni, 2006; Montemanni et al., 2007; Pereira and Averbakh, 2011, 2013]. Furthermore, we discuss some techniques able to improve the convergence speed of the proposed exact method.

5.1 Standard Benders decomposition

As discussed in Section 3.2, formulation $\mathcal{F}$ has an exponential number of constraints (3.13), one for each path in $\mathcal{P}(\beta)$. Since several of these constraints are inactive at optimality, they can be generated on demand whenever they are violated. In this sense, given a set $\Gamma \subseteq \mathcal{P}(\beta)$, $\Gamma \neq \emptyset$, consider the *relaxed robustness cost* metric defined as follows.

Definition 14. A path $p^*(s, \Gamma) \in \Gamma$ is said to be a Γ-*relaxed shortest path* in a scenario $s \in \mathcal{S}$ if it has the smallest cost in s among all paths in Γ, *i.e.*, $p^*(s_p, \Gamma) = \arg\min_{p \in \Gamma} C_p^s$.

Definition 15. The Γ-*relaxed robustness cost* of a path $p \in \mathcal{P}(\beta)$, denoted by R_p^Γ, is the difference between the cost $C_p^{s_p}$ of p in the scenario s_p induced by p and the cost of a Γ-relaxed shortest path $p^*(s_p, \Gamma) \in \mathcal{P}(\beta)$ in s_p, *i.e.*, $R_p^\Gamma = C_p^{s_p} - C_{p^*(s_p,\Gamma)}^{s_p}$.

Proposition 3. *For any $\Gamma \subseteq \mathcal{P}(\beta)$, $\Gamma \neq \emptyset$, and any path $p \in \mathcal{P}(\beta)$, the Γ-relaxed robustness cost R_p^Γ of p gives a lower bound on the β-restricted robustness cost R_p^β of p.*

Proof. Consider a set $\Gamma \subseteq \mathcal{P}(\beta)$, $\Gamma \neq \emptyset$, and a path $p \in \mathcal{P}(\beta)$. According to Theorem 1, the β-restricted robustness cost of p is given by $R_p^\beta = r_p^{(s_p,\beta)} = C_p^{s_p} - C_{p^*(s_p,\beta)}^{s_p}$, where $p^*(s_p,\beta) \in \mathcal{P}(\beta)$ is a β-restricted shortest path in s_p. By definition, the Γ-relaxed robustness cost of p is given by $R_p^\Gamma = C_p^{s_p} - C_{p^*(s_p,\Gamma)}^{s_p}$, where $p^*(s_p,\Gamma) \in \mathcal{P}(\beta)$ is a Γ-relaxed shortest path in s_p. Notice that $C_{p^*(s_p,\beta)}^{s_p} \leq C_{p'}^{s_p}$ for all $p' \in \mathcal{P}(\beta)$, including $p^*(s_p,\Gamma)$. Therefore,

$$R_p^\Gamma = C_p^{s_p} - C_{p^*(s_p,\Gamma)}^{s_p} \leq C_p^{s_p} - C_{p^*(s_p,\beta)}^{s_p} = R_p^\beta. \tag{5.1}$$

□

Proposition 4. *If $\Gamma = \mathcal{P}(\beta)$, then, for any path $p \in \mathcal{P}(\beta)$, it holds that $R_p^\Gamma = R_p^\beta$.*

Proof. Consider a set $\Gamma = \mathcal{P}(\beta)$ and a path $p \in \mathcal{P}(\beta)$. In this case, a Γ-relaxed shortest path $p^*(s_p,\Gamma) \in \mathcal{P}(\beta)$ in s_p is also a β-restricted shortest path $p^*(s_p,\beta) \in \mathcal{P}(\beta)$ in s_p. Therefore, considering Theorem 1,

$$R_p^\Gamma = C_p^{s_p} - C_{p^*(s_p,\Gamma)}^{s_p} = C_p^{s_p} - C_{p^*(s_p,\beta)}^{s_p} = R_p^\beta. \tag{5.2}$$

□

Definition 16. A path $\bar{p}^* \in \mathcal{P}(\beta)$ is said to be a Γ-*relaxed robust path* if it has the smallest Γ-relaxed robustness cost among all the paths in Γ, *i.e.*, $\bar{p}^* = \arg\min_{p \in \Gamma} R_p^\Gamma$.

Considering the relaxed metric discussed above, we propose a Benders-like decomposition algorithm to solve $\mathcal{F}$. The algorithm, referred to as *Standard Benders*, is described in Algorithm 1. Let $\mathcal{P}(\beta)^\psi \subseteq \mathcal{P}(\beta)$ be the set of paths (Benders cuts) that are available at an iteration ψ, and $\mathcal{F}^\psi$ be a relaxed version of $\mathcal{F}$ in which constraints (3.13) are replaced by

$$\rho \leq \sum_{(i,j) \in A} (l_{ij} + (u_{ij} - l_{ij}) y_{ij}) \bar{x}_{ij} \quad \forall \bar{x} \in \mathcal{P}(\beta)^\psi. \tag{5.3}$$

Thus, the relaxed problem $\mathcal{F}^\psi$, called *master problem*, is defined by (3.8)-(3.12), (3.14), (3.15) and (5.3). One may observe that $\mathcal{F}^\psi$ is precisely the problem of finding a Γ-relaxed robust path, with $\Gamma = \mathcal{P}(\beta)^\psi$.

Let ub^{ψ} keep the best upper bound found (until an iteration ψ) on the solution of $\mathcal{F}$. Notice that, at the beginning of Standard Benders, $\mathcal{P}(\beta)^1$ contains the initial Benders cuts available, whereas ub^1 keeps the initial upper bound on the solution of $\mathcal{F}$. In this case, $\mathcal{P}(\beta)^1 = \emptyset$ and $ub^1 := +\infty$. At each iteration ψ, Standard Benders algorithm obtains a solution by solving a corresponding master problem $\mathcal{F}^{\psi}$ and seeks a constraint (3.13) that is most violated by this solution. One may observe that, initially, no constraint (5.3) is considered, since $\mathcal{P}(\beta)^1 = \emptyset$. An initialization step is then necessary to add at least one path to $\mathcal{P}(\beta)^1$, thus avoiding unbounded solutions during the first resolution of the master problem. In this sense, it is computed a β-restricted shortest path in the worst-case scenario s^u, with each arc in A having its cost value set to its upper bound, *i.e.*, $c_{ij}^{s^u} = u_{ij} \; \forall \, (i,j) \in A$ (Step I, Algorithm 1).

Algorithm 1: Standard Benders.

Input: Graph $G = (V, A)$, limit β, vertices o and t
Output: $(\bar{y}^*, R^*)$, where $\bar{y}^*$ is a β-restricted robust path, and R^* is its corresponding β-restricted robustness cost
$\psi := 1$; $ub^1 := +\infty$; $\mathcal{P}(\beta)^1 := \emptyset$;
Step I. (Initialization)
Find a β-restricted shortest path $\bar{x}^0 \in \mathcal{P}(\beta)$ in the scenario s^u;
$\mathcal{P}(\beta)^1 := \mathcal{P}(\beta)^1 \cup \{\bar{x}^0\}$;
Step II. (Master problem)
Solve the relaxed problem $\mathcal{F}^{\psi}$, obtaining a solution $(\bar{y}^{\psi}, \bar{\rho}^{\psi})$;
Step III. (Auxiliary problem)
Find a β-restricted shortest path $\bar{x}^{\psi} \in \mathcal{P}(\beta)$ in the scenario $s_{\bar{y}^{\psi}}$ induced by $\bar{y}^{\psi}$ and compute $R^{\beta}_{\bar{y}^{\psi}} = C^{s_{\bar{y}^{\psi}}}_{\bar{y}^{\psi}} - C^{s_{\bar{y}^{\psi}}}_{\bar{x}^{\psi}}$, the β-restricted robustness cost of $\bar{y}^{\psi}$;
Step IV. (Stopping condition)
$lb^{\psi} := \sum_{(i,j)\in A} u_{ij}\bar{y}^{\psi}_{ij} - \bar{\rho}^{\psi}$;
if $lb^{\psi} \geq R^{\beta}_{\bar{y}^{\psi}}$ **then**
    $\bar{y}^* := \bar{y}^{\psi}$;
    $R^* := R^{\beta}_{\bar{y}^{\psi}}$;
    Return $(\bar{y}^*, R^*)$;
end
else
    $ub^{\psi} := \min\{ub^{\psi}, R^{\beta}_{\bar{y}^{\psi}}\}$;
    $ub^{\psi+1} := ub^{\psi}$;
    $\mathcal{P}(\beta)^{\psi+1} := \mathcal{P}(\beta)^{\psi} \cup \{\bar{x}^{\psi}\}$;
    $\psi := \psi + 1$;
    Go to Step II;
end

After the initialization step, the iterative procedure takes place. At each iteration ψ, the corresponding relaxed problem $\mathcal{F}^{\psi}$ is solved (Step II, Algorithm 1), obtaining a solution $(\bar{y}^\psi, \bar{\rho}^\psi)$. For the sake of simplicity, the notation $\bar{y}^\psi$ is used in the sequel meaning the actual path in $\mathcal{P}(\beta)$ that this solution represents. Then, the algorithm checks if $(\bar{y}^\psi, \bar{\rho}^\psi)$ violates any constraint (3.13) of the original problem $\mathcal{F}$, *i.e.*, if there is a constraint (5.3) that should have been considered in $\mathcal{F}^\psi$ and was not. For this purpose, it is solved an *auxiliary problem* that computes $R^\beta_{\bar{y}^\psi}$ (the actual β-restricted robustness cost of $\bar{y}^\psi$) by finding a β-restricted shortest path $\bar{x}^\psi$ in the scenario $s_{\bar{y}^\psi}$ induced by $\bar{y}^\psi$ (see Step III, Algorithm 1).

Let $lb^\psi = \sum_{(i,j)\in A} u_{ij}\bar{y}^\psi_{ij} - \bar{\rho}^\psi$ be the value of the objective function in (3.8) related to the solution $(\bar{y}^\psi, \bar{\rho}^\psi)$ of the current master problem $\mathcal{F}^\psi$. Notice that, considering $\Gamma = \mathcal{P}(\beta)^\psi$, lb^ψ corresponds to the Γ-relaxed robustness cost of $\bar{y}^\psi$. Thus, according to Proposition 3, lb^ψ gives a lower bound on the solution of $\mathcal{F}$. Moreover, since $\bar{y}^\psi$ is a path in $\mathcal{P}(\beta)$, its β-restricted robustness cost $R^\beta_{\bar{y}^\psi}$ gives an upper bound on the solution of $\mathcal{F}$. Accordingly, if lb^ψ reaches $R^\beta_{\bar{y}^\psi}$, the algorithm stops. Otherwise, ub^ψ and $ub^{\psi+1}$ are both set to the best upper bound found by the algorithm until iteration ψ. In addition, a new constraint (5.3) is generated from $\bar{x}^\psi$ and added to $\mathcal{F}^{\psi+1}$ by setting $\mathcal{P}(\beta)^{\psi+1} := \mathcal{P}(\beta)^\psi \cup \{\bar{x}^\psi\}$ (see Step IV of Algorithm 1). One may observe that the algorithm stops when the value $\bar{\rho}^\psi$ corresponds to the cost of $\bar{x}^\psi$ in the scenario $s_{\bar{y}^\psi}$, *i.e.*, the optimal solution for $\mathcal{F}^\psi$ is also feasible (and, therefore, optimal) for $\mathcal{F}$.

Notice that the auxiliary problems that arise in Standard Benders (as well as the initialization step) involve solving R-SP instances. Then, any exact algorithm for R-SP (see, *e.g.*, Santos et al. [2007]; Dumitrescu and Boland [2003]; Zhu and Wilhelm [2012]) can be applied to solve these problems. Indeed, an optimization solver may be used to directly handle an ILP formulation for R-SP. The convergence of Standard Benders is ensured by Proposition 4 and the following results.

Lemma 2. *If $\mathcal{F}$ is feasible, then every R-SP problem that arises in Standard Benders is also feasible.*

Proof. Assume that $\mathcal{F}$ is feasible. Then, we must have $\mathcal{P}(\beta) \neq \emptyset$. This implies the existence of a β-restricted shortest path in every scenario $s \in S$, and, thus, any R-SP problem that arises while executing Algorithm 1 is also feasible. □

Proposition 5. *At each iteration $\psi \geq 1$ of Algorithm 1, if the stopping condition is not satisfied, then the resolution of the corresponding auxiliary problem leads to a new path $\bar{x}^\psi \in \mathcal{P}(\beta)\backslash\mathcal{P}(\beta)^\psi$.*

Proof. Consider an iteration $\psi \geq 1$ of Algorithm 1 and assume, by contradiction, that (I) the stopping condition is not satisfied, and (II) the resolution of the auxiliary problem of iteration ψ does not lead to a path in $\mathcal{P}(\beta)\backslash\mathcal{P}(\beta)^{\psi}$. Let $(\bar{y}^{\psi}, \bar{\rho}^{\psi})$ be the solution obtained from the resolution of the corresponding master problem $\mathcal{F}^{\psi}$. From Lemma 2, the auxiliary problem of iteration ψ is feasible, and, thus, its resolution leads to a β-restricted shortest path $\bar{x}^{\psi} \in \mathcal{P}(\beta)$ in the scenario $s_{\bar{y}^{\psi}}$ induced by $\bar{y}^{\psi}$. From assumption (I), we must have $lb^{\psi} < R^{\beta}_{\bar{y}^{\psi}}$. Considering Theorem 1 and letting $\Gamma = \mathcal{P}(\beta)^{\psi}$, we have that $lb^{\psi} = R^{\Gamma}_{\bar{y}^{\psi}}$, and, moreover,

$$C^{s_{\bar{y}^{\psi}}}_{\bar{y}^{\psi}} - C^{s_{\bar{y}^{\psi}}}_{p^*(s_{\bar{y}^{\psi}},\Gamma)} = R^{\Gamma}_{\bar{y}^{\psi}} = lb^{\psi} < R^{\beta}_{\bar{y}^{\psi}} = C^{s_{\bar{y}^{\psi}}}_{\bar{y}^{\psi}} - C^{s_{\bar{y}^{\psi}}}_{p^*(s_{\bar{y}^{\psi}},\beta)}, \tag{5.4}$$

where $p^*(s_{\bar{y}^{\psi}}, \Gamma)$ is a Γ-relaxed shortest path in $s_{\bar{y}^{\psi}}$ and $p^*(s_{\bar{y}^{\psi}}, \beta)$ is a β-restricted shortest path in $s_{\bar{y}^{\psi}}$. From (5.4), we obtain

$$C^{s_{\bar{y}^{\psi}}}_{p^*(s_{\bar{y}^{\psi}},\Gamma)} > C^{s_{\bar{y}^{\psi}}}_{p^*(s_{\bar{y}^{\psi}},\beta)}. \tag{5.5}$$

Notice that, since $\bar{x}^{\psi}$ is also a β-restricted shortest path in $s_{\bar{y}^{\psi}}$, it follows, from (5.5), that

$$C^{s_{\bar{y}^{\psi}}}_{p^*(s_{\bar{y}^{\psi}},\Gamma)} > C^{s_{\bar{y}^{\psi}}}_{p^*(s_{\bar{y}^{\psi}},\beta)} = C^{s_{\bar{y}^{\psi}}}_{\bar{x}^{\psi}}. \tag{5.6}$$

Furthermore, as $p^*(s_{\bar{y}^{\psi}}, \Gamma)$ is a Γ-relaxed shortest path in $s_{\bar{y}^{\psi}}$, and, from assumption (II), $\bar{x}^{\psi}$ belongs to $\mathcal{P}(\beta)^{\psi} = \Gamma$, we also have that

$$C^{s_{\bar{y}^{\psi}}}_{p^*(s_{\bar{y}^{\psi}},\Gamma)} \leq C^{s_{\bar{y}^{\psi}}}_{\bar{x}^{\psi}}, \tag{5.7}$$

which, considering (5.6), is a contradiction. □

5.2 Extended Benders decomposition

The extended version of Standard Benders here presented, called *Extended Benders*, uses additional information obtained while solving each master problem to generate, whenever possible, more than a single Benders cut per iteration of the algorithm. Precisely, whenever a new incumbent solution is found in the process of solving a master problem, such solution is stored to be later used in the generation of Benders cuts. This idea was suggested in Fischetti et al. [2010] and firstly applied to a robust-hard problem in Pereira and Averbakh [2013]. Extended Benders is described in Algorithm 2.

After the initialization (Step I, Algorithm 2), which is done as in Standard Benders, an improved iterative procedure takes place. At each iteration ψ, the corresponding $\mathcal{F}^{\psi}$ problem is solved (as a master problem) to obtain a solution $(\bar{y}^{\psi}, \bar{\rho}^{\psi})$, and all the incumbent solution vectors $\bar{y}$ found along the process are stored in a set Π^{ψ} (Step II, Algorithm 2). Notice that, in this case, an incumbent solution consists of the best

Algorithm 2: Extended Benders.

Input: Graph $G = (V, A)$, limit β, vertices o and t
Output: $(\bar{y}^*, R^*)$, where $\bar{y}^*$ is a β-restricted robust path, and R^* is its corresponding β-restricted robustness cost
$\psi := 1$; $ub^1 := +\infty$; $\mathcal{P}(\beta)^1 := \emptyset$;
Step I. (Initialization)
Find a β-restricted shortest path $\bar{x}^0 \in \mathcal{P}(\beta)$ in the scenario s^u;
$\mathcal{P}(\beta)^1 := \mathcal{P}(\beta)^1 \cup \{\bar{x}^0\}$;
Step II. (Master problem)
Solve the relaxed problem $\mathcal{F}^\psi$, obtaining a solution $(\bar{y}^\psi, \bar{\rho}^\psi)$, and store in Π^ψ all the incumbent (integer and feasible) solution vectors $\bar{y}$ found along the process;
Step III. (Auxiliary problem)
Find a β-restricted shortest path $\bar{x}^\psi \in \mathcal{P}(\beta)$ in the scenario $s_{\bar{y}^\psi}$ induced by $\bar{y}^\psi$ and compute $R^\beta_{\bar{y}^\psi}$, the β-restricted robustness cost of $\bar{y}^\psi$;
Step IV. (Stopping condition)
$lb^\psi := \sum_{(i,j)\in A} u_{ij}\bar{y}^\psi_{ij} - \bar{\rho}^\psi$;
if $lb^\psi \geq R^\beta_{\bar{y}^\psi}$ **then**
  $\bar{y}^* := \bar{y}^\psi$;
  $R^* := R^\beta_{\bar{y}^\psi}$;
  Return $(\bar{y}^*, R^*)$;
end
else
  $\mathcal{C}^\psi := \{\bar{x}^\psi\}$;
  $ub^\psi := \min\{ub^\psi, R^\beta_{\bar{y}^\psi}\}$;
  forall $\bar{y} \in \Pi^\psi$ **do**
    Find a β-restricted shortest path $\bar{x} \in \mathcal{P}(\beta)$ in the scenario $s_{\bar{y}}$ induced by $\bar{y}$ and compute $R_{\bar{y}}$, the β-restricted robustness cost of $\bar{y}$;
    $\mathcal{C}^\psi := \mathcal{C}^\psi \cup \{\bar{x}\}$;
    $ub^\psi := \min\{ub^\psi, R_{\bar{y}}\}$;
  end
  $ub^{\psi+1} := ub^\psi$;
  $\mathcal{P}(\beta)^{\psi+1} := \mathcal{P}(\beta)^\psi \cup \mathcal{C}^\psi$;
  $\psi := \psi + 1$;
  Go to Step II;
end

feasible solution known at a given point of the resolution of $\mathcal{F}^\psi$. Moreover, since only the incumbent solution values referred to y variables are stored in Π^ψ, any solution in this set represents a path in $\mathcal{P}(\beta)$.

As in Standard Benders, an auxiliary problem is then solved in order to compute the β-restricted robustness cost $R^\beta_{\bar{y}^\psi}$ of the path identified by $\bar{y}^\psi$. This is done by finding a β-restricted shortest path $\bar{x}^\psi$ in the scenario $s_{\bar{y}^\psi}$ induced by $\bar{y}^\psi$ (Step III, Algorithm 2). If the stopping condition is satisfied, *i.e.*, the current lower bound (given by $lb^{\psi} = \sum_{(i,j)\in A} u_{ij}\bar{y}^{\psi}_{ij} - \bar{\rho}^{\psi}$) reaches the upper bound provided by $R^{\beta}_{\bar{y}^{\psi}}$, then $\bar{y}$ is an optimal solution for $\mathcal{F}$ and the algorithm stops. Otherwise, let $\mathcal{C}^{\psi}$ be the set of Benders cuts generated at iteration ψ. Initially, $\mathcal{C}^{\psi} := \{\bar{x}^{\psi}\}$. For each $\bar{y} \in \Pi^{\psi}$, a β-restricted shortest path $\bar{x}$ in the scenario induced by $\bar{y}$ is found (by solving a R-SP problem) and added to $\mathcal{C}^{\psi}$. Then, new constraints (5.3) are generated from $\mathcal{C}^{\psi}$ and added to $\mathcal{F}^{\psi+1}$ by setting $\mathcal{P}(\beta)^{\psi+1} := \mathcal{P}(\beta)^{\psi} \cup \mathcal{C}^{\psi}$. In addition, the β-restricted robustness costs referred to $\bar{y}^{\psi}$ and to the solutions in Π^{ψ} are used to update ub^{ψ}, the best upper bound found until iteration ψ (see Step IV, Algorithm 2).

5.3 Warm start procedures

We present three warm start procedures able to further improve the performance of the Benders decompositions previously discussed by providing initial Benders cuts referred to constraints (5.3), as well as initial upper bounds. The first procedure, whose idea was suggested in McDaniel and Devine [1977], consists in solving a linearly relaxed version of $\mathcal{F}$, namely $\tilde{\mathcal{F}}$, in which constraints (3.14) are replaced by

$$0 \leq y_{ij} \leq 1 \quad \forall\,(i,j) \in A. \tag{5.8}$$

$\tilde{\mathcal{F}}$ is then defined by (3.8)-(3.13), (3.15) and (5.8). This problem is solved via a slightly modified Standard Benders. In this case, the master problem of any iteration ψ of the decomposition is a LP problem defined by (3.8)-(3.12), (3.15), (5.3) and (5.8). Therefore, a solution vector $\bar{y}^{\psi}$ referred to a master problem is not necessarily a binary vector and may not represent a path in $\mathcal{P}(\beta)$. Moreover, the scenario $s_{\bar{y}^{\psi}}$ induced by $\bar{y}^{\psi}$ is now defined as $c^{s_{\bar{y}^{\psi}}}_{ij} = l_{ij} + (u_{ij} - l_{ij})\bar{y}^{\psi}_{ij}$ $\forall (i,j) \in A$, which allows the existence of arc costs that are neither the lower nor the upper bounds of the corresponding cost intervals. Likewise, the cost of $\bar{y}^{\psi}$ in a scenario $s \in \mathcal{S}$ is now given by $C^{s}_{\bar{y}^{\psi}} = \sum_{(i,j)\in A} c^{s}_{ij}\bar{y}^{\psi}_{ij}$. The auxiliary problems remain providing valid Benders cuts by solving R-SP problems. These cuts are stored along the process of solving $\tilde{\mathcal{F}}$ to be later used to initialize either Standard Benders or Extended Benders while solving $\mathcal{F}$. We refer to the procedure described above as *Relaxation Start* (RS).

The second procedure, called *Heuristic Start* (HS), uses the heuristic LPH presented in Chapter 4. Precisely, the β-restricted robustness cost $R^{\beta}_{\tilde{p}^*}$ of the β-heuristic robust path $\tilde{p}^* \in \mathcal{P}(\beta)$ found by the heuristic gives an initial upper bound on the solution of $\mathcal{F}$. Furthermore, the β-restricted shortest path $p^*(s_{\tilde{p}^*}, \beta) \in \mathcal{P}(\beta)$ obtained while computing $R_{\tilde{p}^*}$ is stored as a Benders cut.

The third procedure, namely *Extended Heuristic Start* (Extended HS), adapts LPH to retrieve information able to provide additional Benders cuts in the same way as in Extended Benders. Such procedure can be seen as an extension of HS and is described in Algorithm 3. First, the heuristic problem $\mathcal{H}$ is solved, obtaining a β-heuristic robust path represented by $\bar{y}^*$. In addition, all the incumbent solutions found along the process are stored in a set Π, initially empty. As in LPH, problem $\mathcal{H}$ can be directly handled with an optimization solver. The β-restricted robustness cost referred to $\bar{y}^*$ is then computed by finding a β-restricted shortest path $\bar{x}^*$ in the scenario $s_{\bar{y}^*}$ induced by $\bar{y}^*$ (Step I, Algorithm 3). Let $\mathcal{C}$ be the set of valid Benders cuts obtained by the procedure. Also let ub^* keep the smallest β-restricted robustness cost found so far. At this point of the execution, $\mathcal{C} := \{\bar{x}^*\}$ and $ub^* := R^{\beta}_{\bar{y}^*}$. For each $\bar{y} \in \Pi$, a β-restricted shortest path $\bar{x}$ in the scenario induced by $\bar{y}$ is found (by solving a R-SP problem) and added to $\mathcal{C}$. In addition, the β-restricted robustness costs referred to the solutions in Π are computed and used to update ub^* (see Step II, Algorithm 3).

Algorithm 3: Extended HS.

Input: Graph $G = (V, A)$, distance limit β, vertices o and t
Output: $(\mathcal{C}, ub^*)$, where $\mathcal{C}$ is the set of available Benders cuts, and ub^* is the smallest β-restricted robustness cost found

$\Pi := \emptyset$;
Step I. (Heuristic initialization)
Solve the heuristic problem $\mathcal{H}$, obtaining a solution $\bar{y}^*$, and store in Π all the incumbent solution vectors $\bar{y}$ found along the process;
Find a β-restricted shortest path $\bar{x}^* \in \mathcal{P}(\beta)$ in the scenario $s_{\bar{y}^*}$ induced by $\bar{y}^*$ and compute $R^{\beta}_{\bar{y}^*}$, the β-restricted robustness cost of $\bar{y}^*$;
Step II. (Additional cuts generation)
$\mathcal{C} := \{\bar{x}^*\}$;
$ub^* := R^{\beta}_{\bar{y}^*}$;
forall $\bar{y} \in \Pi$ **do**
  Find a β-restricted shortest path $\bar{x} \in \mathcal{P}(\beta)$ in the scenario $s_{\bar{y}}$ induced by $\bar{y}$ and compute $R^{\beta}_{\bar{y}}$, the β-restricted robustness cost of $\bar{y}$;
  $\mathcal{C} := \mathcal{C} \cup \{\bar{x}\}$;
  $ub^* := \min\{ub^*, R^{\beta}_{\bar{y}}\}$;
end
Return $(\mathcal{C}, ub^*)$;

Notice that all of these procedures provide at least one Benders cut to be added to $\mathcal{P}(\beta)$[1], the set of initial cuts available to Standard Benders (and to Extended Benders). The initialization step (Step I of Algorithms 1 and 2) can then be skipped whenever these procedures are adopted. In the next chapter, we describe and compare, in terms of computational performance, some variants of Standard and Extended Benders obtained from coupling the speed-up procedures discussed above with the Benders-like decomposition approaches.

Chapter 6

Computational experiments

In this chapter, we introduce two benchmarks of R-RSP instances and use them to evaluate, out of computational experiments, the effectiveness of LPH. We also compare the quality of the solutions obtained by Standard Benders and Extended Benders while solving these benchmarks of instances. In addition, we analyse the impact of the warm start procedures presented in Section 5.3 on the performance of both Standard and Extended Benders. For this purpose, we couple these procedures with the Benders-like decomposition approaches in different manners, as detailed in the sequel (see Section 6.2.2).

All the codes are in C++, along with the optimization solver ILOG CPLEX 12.5 under default parameter settings. The computational experiments were performed on a 64 bits Intel® Xeon® E5405 machine with 2.0 GHz of clock and 7.0 GB of RAM memory, under Linux operating system. Whenever a R-SP problem had to be solved, we used CPLEX to handle the ILP formulation $\mathcal{I}$ directly. We also used CPLEX to solve each master problem of the Benders-like decompositions. Every exact algorithm tested was set to run for up to 3600 seconds.

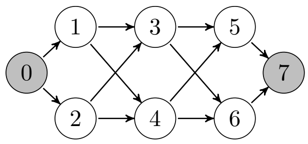
6.1 Benchmarks description

Due to the lack of R-RSP instances in the literature, we generated two benchmarks of instances inspired by the applications described in Section 1. These benchmarks were adapted from two sets of RSP instances. Karaşan instances [Karaşan et al., 2001] resemble telecommunication networks, while Coco instances [Coco et al., 2014a] resemble urban transportation networks.

Karaşan instances have been largely used in experiments concerning RSP [Karaşan et al., 2001; Montemanni and Gambardella, 2004; Montemanni et al., 2004;

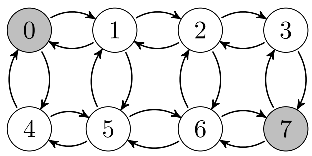
Montemanni and Gambardella, 2005; Coco et al., 2014a]. They consist of layered [Sugiyama et al., 1981] and acyclic [Bondy and Murty, 1976] digraphs. In these digraphs, each of the κ layers has the same number ω of vertices. There is an arc from every vertex in a layer $b \in \{1, \dots, \kappa - 1\}$ to every vertex in the adjacent layer $b + 1$. Moreover, there is an arc from the origin vertex o to every vertex in the first layer, and an arc from every vertex in the layer κ to the destination vertex t. These instances are named K-v-Φ_{max}-δ-ω, with $0 < \delta < 1$, where v is the number of vertices (aside from o and t) and Φ_{max} is an integer value. For each arc $(i, j) \in A$, a random integer value Φ_{ij} is uniformly chosen in the range $[1, \Phi_{max}]$. Afterwards, random values l_{ij} and u_{ij} are uniformly selected, respectively, in the ranges $[(1 - \delta) \cdot \Phi_{ij}, (1 + \delta) \cdot \Phi_{ij}]$ and $[l_{ij}, (1 + \delta) \cdot \Phi_{ij}]$. Note that Φ plays the role of a base case scenario from which the uncertainty is generated. Figure 6.1 shows an example of an acyclic digraph with 3 layers of width 2.

Figure 6.1: An acyclic digraph with 3 layers of width 2. Here, $o = 0$ and $t = 7$.



Coco instances consist of grid digraphs based on $n \times m$ matrices, where n is the number of rows and m is the number of columns. Each matrix cell corresponds to a vertex in the graph, and there are two bidirectional arcs between each pair of vertices whose respective matrix cells are adjacent. The origin o is defined as the upper left vertex, and the destination t is defined as the lower right vertex. These instances are named G-$n{\times}m$-Φ_{max}-δ, with $0 < \delta < 1$, where Φ_{max} is an integer value. Given Φ_{max} and δ values, the cost intervals are generated as in the Karaşan instances. Figure 6.2 gives an example of a grid digraph.

Figure 6.2: A 2×4 grid digraph, with $o = 0$ and $t = 7$.



For all instances, the resource consumption associated with each arc is given by a random integer value uniformly selected in the interval $(0, 10]$. The small size of the interval allows the generation of instances in which most of the arcs are candidate to appear in an optimal solution, increasing the number of feasible solutions. The symmetry with respect to arc resource consumptions was preserved, *i.e.*, we considered $d_{ij} = d_{ji}$ for any pair of adjacent vertices i and j such that $(i, j) \in A$ and $(j, i) \in A$.

The resource consumption limit β of a given instance was computed as follows. Consider the set $\mathcal{P}$ of all the paths from o to t, and let $\bar{p} \in \mathcal{P}$ be a shortest path in terms of resource consumption, *i.e.*, $\bar{p} = \arg\min_{p \in \mathcal{P}} D_p$. We set $\beta = 1.1 \cdot D_{\bar{p}}$, which means that is given a 10% tolerance with respect to the minimum resource consumption $D_{\bar{p}}$. This way, the resource consumption limit is tighter.

6.2 Results

We generated Karaşan and Coco instances of 1000 and 2000 vertices (approximately), with $\Phi_{max} \in \{20, 200\}$, $\delta \in \{0.5, 0.9\}$ and $\omega \in \{5, 10, 25\}$. Considering these values, a set of 10 instances was generated for each possible parameter configuration. In summary, 480 instances were used in the experiments of LPH and the exact algorithms.

6.2.1 LPH

Computational experiments were carried out in order to evaluate if the proposed heuristic efficiently finds optimal or near-optimal solutions for the two benchmarks of instances described above. Results for Karaşan and Coco instances are reported in Tables 6.1 and 6.2, respectively. The first column displays the name of each set of 10 instances. The second column shows the number of instances solved at optimality by any of the exact algorithms (detailed in Section 6.2.2) within 3600 seconds of execution. The third and fourth columns show, respectively, the average and the standard deviation (over the 10 instances) of the relative optimality gaps given by $100 \cdot \frac{UB^* - LB^*}{UB^*}$. LB^* and UB^* are, respectively, the best lower and upper bounds obtained by any exact algorithm for a given instance. The fifth column displays the average processing time (in seconds) of LPH. The sixth column shows the average (over the 10 instances) of the heuristic optimality gaps given by $100 \cdot \frac{UB_{lph} - LB^*}{UB_{lph}}$, where UB_{lph} is the β-restricted robustness cost of the solution obtained by LPH for a given instance. The standard deviation of these gaps is given in the last column. Notice that LB^* remains the best lower bound obtained by any exact algorithm within 3600 seconds of execution and, therefore, it might not correspond to the cost of an optimal solution. Thus, the aforementioned heuristic gaps may overestimate the actual gaps between the cost of the heuristic solution and the cost of an optimal one.

Regarding Karaşan instances (Table 6.1), it can be seen that the average optimality gaps referred to the solutions provided by LPH are at most 3.46% for the instances

Table 6.1: Computational results of LPH for the layered and acyclic digraph instances.

	Exact algorithms			LPH		
Test set	#opt	AvgGAP(%)	StDev(%)	Time(s)	AvgGAP(%)	StDev(%)
K-1000-20-0.5-5	10	0.00	0.00	11.17	0.00	0.00
K-1000-20-0.9-5	2	3.58	2.90	31.66	3.46	2.78
K-1000-200-0.5-5	10	0.00	0.00	9.29	0.16	0.28
K-1000-200-0.9-5	5	1.36	1.64	23.22	1.36	1.64
K-1000-20-0.5-10	10	0.00	0.00	12.29	0.07	0.22
K-1000-20-0.9-10	10	0.00	0.00	19.32	0.00	0.00
K-1000-200-0.5-10	10	0.00	0.00	11.34	0.31	0.97
K-1000-200-0.9-10	10	0.00	0.00	26.81	0.05	0.15
K-1000-20-0.5-25	10	0.00	0.00	23.65	0.00	0.00
K-1000-20-0.9-25	10	0.00	0.00	29.57	0.24	0.76
K-1000-200-0.5-25	10	0.00	0.00	25.71	0.00	0.00
K-1000-200-0.9-25	10	0.00	0.00	33.15	0.00	0.00
K-2000-20-0.5-5	0	8.61	3.86	74.73	8.35	3.72
K-2000-20-0.9-5	0	14.90	2.60	288.43	14.09	2.25
K-2000-200-0.5-5	0	8.30	2.27	79.02	8.00	2.19
K-2000-200-0.9-5	0	15.52	2.59	557.90	14.54	2.59
K-2000-20-0.5-10	8	0.57	1.34	104.45	0.58	1.22
K-2000-20-0.9-10	1	3.46	2.99	292.36	3.23	2.69
K-2000-200-0.5-10	8	0.29	0.84	126.97	0.28	0.84
K-2000-200-0.9-10	1	2.32	1.99	238.33	2.18	1.86
K-2000-20-0.5-25	10	0.00	0.00	141.14	0.00	0.00
K-2000-20-0.9-25	10	0.00	0.00	263.55	0.07	0.22
K-2000-200-0.5-25	10	0.00	0.00	144.20	0.04	0.13
K-2000-200-0.9-25	10	0.00	0.00	249.86	0.02	0.05
Average		2.46	0.96		2.38	1.02

with 1000 vertices (see K-1000-20-0.9-5), while those of the exact algorithms are up to 3.58% for the same instances. Moreover, the average optimality gaps referred to the solutions provided by LPH are at most 14.54% for the instances with 2000 vertices (see K-2000-200-0.9-5), whereas those of the exact algorithms are up to 15.52% for the same instances. In fact, the average optimality gap of LPH over all Karaşan instances (2.38%) is slightly tighter than that of the exact algorithms (2.46%). It can also be observed that, for a same number of vertices, the smaller the value of ω is, the larger the average gaps achieved by the heuristic (as well as by the exact algorithms) are. For the hardest instances (with $\omega = 5$), the average gaps of the solutions provided by LPH are smaller than or equal to those of the exact algorithms (except for K-1000-200-0.5-5). Notice that the larger optimality gaps observed for the more challenging instances do not necessarily indicate that LPH is ineffective in solving these instances. In fact, the larger gaps may be due to insufficient quality of the lower bounds provided by the exact algorithms. With respect to computational time effort, LPH required at most 33.15 seconds of execution (on average) for the instances with 1000 vertices

Table 6.2: Computational results of LPH for the grid digraph instances.

	Exact algorithms			LPH		
Test set	#opt	AvgGAP(%)	StDev(%)	Time(s)	AvgGAP(%)	StDev(%)
G-32x32-20-0.5	10	0.00	0.00	4.37	0.22	0.70
G-32x32-20-0.9	10	0.00	0.00	5.59	0.69	1.67
G-32x32-200-0.5	10	0.00	0.00	4.44	3.11	4.72
G-32x32-200-0.9	10	0.00	0.00	5.92	0.00	0.00
G-20x50-20-0.5	10	0.00	0.00	4.43	0.42	1.34
G-20x50-20-0.9	10	0.00	0.00	5.22	0.76	1.10
G-20x50-200-0.5	10	0.00	0.00	4.93	0.78	2.45
G-20x50-200-0.9	10	0.00	0.00	5.88	0.58	0.74
G-5x200-20-0.5	10	0.00	0.00	11.61	0.10	0.31
G-5x200-20-0.9	10	0.00	0.00	20.83	0.12	0.16
G-5x200-200-0.5	10	0.00	0.00	9.45	0.29	0.66
G-5x200-200-0.9	10	0.00	0.00	23.46	0.12	0.35
G-44x44-20-0.5	10	0.00	0.00	14.65	0.81	1.42
G-44x44-20-0.9	10	0.00	0.00	17.04	0.55	1.04
G-44x44-200-0.5	10	0.00	0.00	16.18	0.00	0.00
G-44x44-200-0.9	10	0.00	0.00	21.61	0.13	0.22
G-20x100-20-0.5	10	0.00	0.00	21.97	0.00	0.00
G-20x100-20-0.9	10	0.00	0.00	31.32	0.61	1.13
G-20x100-200-0.5	10	0.00	0.00	21.77	0.25	0.66
G-20x100-200-0.9	10	0.00	0.00	28.07	0.21	0.37
G-5x400-20-0.5	7	0.50	0.85	68.95	0.75	1.10
G-5x400-20-0.9	1	3.33	2.32	169.93	3.15	2.21
G-5x400-200-0.5	4	1.92	2.09	88.28	1.95	2.05
G-5x400-200-0.9	0	5.45	2.00	279.05	5.28	1.95
Average		0.47	0.30		0.87	1.10

(see K-1000-200-0.9-25) and at most 557.90 seconds (on average) for the instances with 2000 vertices (see K-2000-200-0.9-5).

Regarding Coco instances (Table 6.2), it can be seen that the average optimality gaps referred to the solutions provided by LPH are at most 5.28% (see G-5x400-200-0.9), while those of the exact algorithms are up to 5.45% for the same instances. The average optimality gap of the exact algorithms over all Coco instances is very small (0.47%), while that of LPH is close to this value (0.87%). It can also be observed that the instances based on 5x400 grids are harder to solve than the other grid instances. Moreover, the average relative gaps referred to the solutions provided by LPH are close to those of the exact algorithms for the hardest grid instances. The results indicate that LPH is an effective heuristic to solve Coco instances. With respect to computational time effort, LPH required, on average, at most 279.05 seconds of execution (see G-5x400-200-0.9).

6.2.2 Exact algorithms and warm start procedures

The first experiment was performed in order to compare the quality of the solutions obtained by Standard Benders and Extended Benders for the two benchmarks and, thus, check if generating additional Benders cuts as in Extended Benders leads to better quality bounds. Results for Karaşan and Coco instances are reported in Tables 6.3 and 6.4, respectively. The first column displays the name of each set of 10 instances. For each algorithm, the "#opt" column displays the number of instances solved at optimality within 3600 seconds of execution. The average processing time (in seconds) spent in solving these instances is reported in the next column in a row. If no instance in the set was solved at optimality, this entry is filled with a dash. For each set of instances, it is also reported the average and the standard deviation (over the 10 instances) of the relative optimality gaps given by $100 \cdot \frac{UB_b - LB_b}{UB_b}$, where LB_b and UB_b are, respectively, the best lower and upper bounds obtained by the corresponding algorithm within the time limit. The last row shows, for each algorithm, the average of the optimality gaps over all instances considered and the average of the standard deviations referred to each set of instances.

Regarding Karaşan instances (Table 6.3), it can be seen that the average optimality gaps referred to the solutions provided by Standard Benders are up to 7.54% for the instances with 1000 vertices (see K-1000-20-0.9-5), while those of Extended Benders are at most 4.35% for the same instances. Moreover, the average optimality gaps referred to the solutions provided by Standard Benders are up to 25.90% for the instances with 2000 vertices (see K-2000-200-0.9-5), whereas those of Extended Benders are at most 21.41% for the same instances. Indeed, the average gaps of the solutions provided by Extended Benders are smaller than or equal to those of Standard Benders for all sets of instances. In addition, Extended Benders was able to solve at optimality seven more instances than Standard Benders (three from K-1000-200-0.9-5, two from K-1000-20-0.5-5, one from K-1000-200-0.5-5 and one from K-2000-200-0.5-10). We also notice that, for a same number of vertices, the smaller the value of ω is, the larger the average gaps achieved by both algorithms are.

With respect to Coco instances (Table 6.4), the average optimality gaps referred to the solutions provided by Standard Benders are up to 11.69% (see G-5x400-200-0.9), while those of Extended Benders are at most 7.92% for the same instances. The average optimality gap of Standard Benders over all Coco instances is very small (1.18%), while that of Extended Benders is even tighter (0.66%). Indeed, the average relative gaps referred to the solutions provided by Extended Benders are smaller than or equal to those of Standard Benders for all sets of instances. Notice that the instances based

Table 6.3: Computational results of Standard Benders and Extended Benders for the layered and acyclic digraph instances, with 3600 seconds of time limit.

	Standard Benders				**Extended Benders**			
			GAP				GAP	
Test set	#opt	Time(s)	Avg(%)	StDev(%)	#opt	Time(s)	Avg(%)	StDev(%)
K-1000-20-0.5-5	8	1612.77	0.33	0.74	10	1364.84	0.00	0.00
K-1000-20-0.9-5	2	2228.30	7.54	5.33	2	1183.74	4.35	3.73
K-1000-200-0.5-5	9	989.83	0.02	0.07	10	721.04	0.00	0.00
K-1000-200-0.9-5	1	2035.99	4.98	3.34	4	1867.38	1.88	2.36
K-1000-20-0.5-10	10	103.66	0.00	0.00	10	170.91	0.00	0.00
K-1000-20-0.9-10	10	247.38	0.00	0.00	10	272.06	0.00	0.00
K-1000-200-0.5-10	10	55.76	0.00	0.00	10	151.68	0.00	0.00
K-1000-200-0.9-10	10	458.15	0.00	0.00	10	424.64	0.00	0.00
K-1000-20-0.5-25	10	23.40	0.00	0.00	10	40.88	0.00	0.00
K-1000-20-0.9-25	10	41.63	0.00	0.00	10	71.75	0.00	0.00
K-1000-200-0.5-25	10	24.38	0.00	0.00	10	49.46	0.00	0.00
K-1000-200-0.9-25	10	52.57	0.00	0.00	10	98.19	0.00	0.00
K-2000-20-0.5-5	0	-	14.72	5.74	0	-	10.80	4.83
K-2000-20-0.9-5	0	-	25.08	2.66	0	-	20.21	2.67
K-2000-200-0.5-5	0	-	14.60	3.09	0	-	11.75	2.63
K-2000-200-0.9-5	0	-	25.90	3.45	0	-	21.41	3.70
K-2000-20-0.5-10	8	1538.56	1.01	2.39	8	1539.18	0.65	1.58
K-2000-20-0.9-10	0	-	7.57	4.00	0	-	4.61	4.14
K-2000-200-0.5-10	4	970.60	1.59	2.35	5	1681.78	0.58	1.34
K-2000-200-0.9-10	0	-	6.17	2.40	0	-	3.47	2.55
K-2000-20-0.5-25	10	196.35	0.00	0.00	10	300.64	0.00	0.00
K-2000-20-0.9-25	10	470.72	0.00	0.00	10	541.09	0.00	0.00
K-2000-200-0.5-25	10	173.27	0.00	0.00	10	366.05	0.00	0.00
K-2000-200-0.9-25	10	658.04	0.00	0.00	10	753.94	0.00	0.00
Average			4.56	1.48			3.32	1.23

on 5x400 grids are harder than the other grid instances. Extended Benders was able to solve at optimality more instances than Standard Benders, specially for the sets of hardest instances (see, for example, G-5x400-20-0.5 and G-5x400-200-0.5). The results suggest that solely generating additional Benders cuts referred to incumbent solutions (as in Extended Benders) improves the overall quality of the bounds obtained for both benchmarks.

The second experiment was performed in order to evaluate the impact of the warm start procedures discussed in Section 5.3 on the quality of the solutions obtained by Standard Benders and Extended Benders. Table 6.5 describes a total of six algorithms obtained from coupling the warm start procedures with these Benders-like decomposition approaches in different manners. The first column displays the name of the resulting algorithm, whereas the second, third and fourth columns indicate which warm start procedures are used in the corresponding algorithm. The last two columns

Table 6.4: Computational results of Standard Benders and Extended Benders for the grid digraph instances, with 3600 seconds of time limit.

	Standard Benders				Extended Benders			
			GAP				GAP	
Test set	#opt	Time(s)	Avg(%)	StDev(%)	#opt	Time(s)	Avg(%)	StDev(%)
G-32x32-20-0.5	10	10.46	0.00	0.00	10	27.18	0.00	0.00
G-32x32-20-0.9	10	11.70	0.00	0.00	10	36.58	0.00	0.00
G-32x32-200-0.5	10	9.42	0.00	0.00	10	30.81	0.00	0.00
G-32x32-200-0.9	10	13.14	0.00	0.00	10	42.57	0.00	0.00
G-20x50-20-0.5	10	8.83	0.00	0.00	10	18.78	0.00	0.00
G-20x50-20-0.9	10	11.82	0.00	0.00	10	33.47	0.00	0.00
G-20x50-200-0.5	10	8.55	0.00	0.00	10	35.59	0.00	0.00
G-20x50-200-0.9	10	17.01	0.00	0.00	10	57.07	0.00	0.00
G-5x200-20-0.5	10	423.59	0.00	0.00	10	297.39	0.00	0.00
G-5x200-20-0.9	9	704.22	0.28	0.89	10	741.82	0.00	0.00
G-5x200-200-0.5	10	167.59	0.00	0.00	10	187.41	0.00	0.00
G-5x200-200-0.9	8	1110.11	0.22	0.55	10	669.96	0.00	0.00
G-44x44-20-0.5	10	32.16	0.00	0.00	10	91.02	0.00	0.00
G-44x44-20-0.9	10	39.06	0.00	0.00	10	141.27	0.00	0.00
G-44x44-200-0.5	10	31.24	0.00	0.00	10	119.11	0.00	0.00
G-44x44-200-0.9	10	49.78	0.00	0.00	10	176.38	0.00	0.00
G-20x100-20-0.5	10	44.76	0.00	0.00	10	134.42	0.00	0.00
G-20x100-20-0.9	10	108.33	0.00	0.00	10	258.88	0.00	0.00
G-20x100-200-0.5	10	61.86	0.00	0.00	10	214.04	0.00	0.00
G-20x100-200-0.9	10	84.77	0.00	0.00	10	240.37	0.00	0.00
G-5x400-20-0.5	0	-	2.82	2.00	5	2788.59	0.80	1.23
G-5x400-20-0.9	0	-	7.87	3.98	1	3239.56	4.38	2.95
G-5x400-200-0.5	1	2963.11	5.47	4.38	4	2593.59	2.72	2.85
G-5x400-200-0.9	0	-	11.69	3.47	0	-	7.92	3.26
Average			1.18	0.64			0.66	0.43

especify which Benders-like decomposition approach is adopted along with the warm start procedures.

Table 6.5: Algorithms obtained from coupling Standard Benders and Extended Benders with the warm start procedures.

	Warm Sart Procedure			Benders Decomposition Algorithm	
Algorithm	RS	HS	Extended HS	Standard Benders	Extended Benders
RS-Benders	×			×	
HS-Benders		×		×	
RS&HS-Benders	×	×		×	
Extended RS-Benders	×				×
Extended HS-Benders			×		×
Extended RS&HS-Benders	×		×		×

Tables 6.6 and 6.7 display the results concerning the first three algorithms in

Table 6.5, the ones that couple the warm start procedures with Standard Benders. Tables 6.8 and 6.9 show the results concerning the last three algorithms in Table 6.5, which couple the warm start procedures with Extended Benders. The first column displays the name of each set of 10 instances. For each algorithm, the "#opt" column displays the number of instances solved at optimality within 3600 seconds of execution. The average processing time (in seconds) spent in solving these instances is reported in the next column in a row. If no instance in the set was solved at optimality, this entry is filled with a dash. Notice that the processing time referred to solving a given instance includes the time spent by the warm start procedures. For each set of instances, it is also reported the average and the standard deviation (over the 10 instances) of the relative optimality gaps given by $100 \cdot \frac{UB_b - LB_b}{UB_b}$. LB_b and UB_b are, respectively, the best lower and upper bounds obtained by the corresponding algorithm within the time limit. The last row shows, for each algorithm, the average of the optimality gaps over all instances considered and the average of the standard deviations referred to each set of instances.

Table 6.6: Computational results of RS-Benders, HS-Benders and RS&HS-Benders for the layered and acyclic digraph instances, with 3600 seconds of time limit.

	RS-Benders				**HS-Benders**				**RS&HS-Benders**			
			GAP				GAP				GAP	
Test set	#opt	Time(s)	Avg(%)	StDev(%)	#opt	Time(s)	Avg(%)	StDev(%)	#opt	Time(s)	Avg(%)	StDev(%)
K-1000-20-0.5-5	8	1573.25	0.22	0.53	7	1414.10	0.28	0.48	8	1629.03	0.17	0.37
K-1000-20-0.9-5	2	2364.28	7.40	5.04	2	2359.35	5.90	3.88	2	2427.22	5.78	3.69
K-1000-200-0.5-5	10	1212.22	0.00	0.00	9	966.39	0.10	0.33	10	1213.41	0.00	0.00
K-1000-200-0.9-5	1	1747.17	5.17	3.56	1	2045.70	3.94	2.58	1	1951.49	3.73	2.60
K-1000-20-0.5-10	10	114.73	0.00	0.00	10	119.12	0.00	0.00	10	113.23	0.00	0.00
K-1000-20-0.9-10	10	258.50	0.00	0.00	10	271.99	0.00	0.00	10	272.50	0.00	0.00
K-1000-200-0.5-10	10	61.16	0.00	0.00	10	66.56	0.00	0.00	10	74.57	0.00	0.00
K-1000-200-0.9-10	10	391.97	0.00	0.00	10	469.29	0.00	0.00	10	442.88	0.00	0.00
K-1000-20-0.5-25	10	32.75	0.00	0.00	10	43.06	0.00	0.00	10	54.56	0.00	0.00
K-1000-20-0.9-25	10	47.29	0.00	0.00	10	66.11	0.00	0.00	10	76.51	0.00	0.00
K-1000-200-0.5-25	10	32.62	0.00	0.00	10	46.18	0.00	0.00	10	54.95	0.00	0.00
K-1000-200-0.9-25	10	52.09	0.00	0.00	10	76.80	0.00	0.00	10	79.98	0.00	0.00
K-2000-20-0.5-5	0	-	13.94	5.11	0	-	10.65	3.77	0	-	10.17	3.84
K-2000-20-0.9-5	0	-	20.09	2.46	0	-	18.30	2.75	0	-	15.83	2.23
K-2000-200-0.5-5	0	-	13.88	3.35	0	-	10.96	2.29	0	-	10.30	2.38
K-2000-200-0.9-5	0	-	22.08	3.23	0	-	18.87	3.04	0	-	16.10	2.62
K-2000-20-0.5-10	8	1382.02	1.03	2.28	8	1570.92	0.93	2.17	8	1402.06	0.94	2.11
K-2000-20-0.9-10	0	-	6.41	4.33	0	-	6.44	3.71	0	-	5.33	3.23
K-2000-200-0.5-10	4	867.18	1.39	2.25	4	997.00	1.28	1.77	4	907.11	1.13	1.68
K-2000-200-0.9-10	0	-	5.42	3.14	0	-	5.10	2.48	0	-	4.20	2.23
K-2000-20-0.5-25	10	204.66	0.00	0.00	10	323.48	0.00	0.00	10	355.93	0.00	0.00
K-2000-20-0.9-25	10	315.05	0.00	0.00	10	714.59	0.00	0.00	10	554.43	0.00	0.00
K-2000-200-0.5-25	10	189.81	0.00	0.00	10	293.97	0.00	0.00	10	318.33	0.00	0.00
K-2000-200-0.9-25	10	443.44	0.00	0.00	10	873.65	0.00	0.00	10	662.04	0.00	0.00
Average			4.04	1.47			3.45	1.22			3.07	1.12

Table 6.7: Computational results of RS-Benders, HS-Benders and RS&HS-Benders for the grid digraph instances, with 3600 seconds of time limit.

	RS-Benders				**HS-Benders**				**RS&HS-Benders**			
			GAP				GAP				GAP	
Test set	#opt	Time(s)	Avg(%)	StDev(%)	#opt	Time(s)	Avg(%)	StDev(%)	#opt	Time(s)	Avg(%)	StDev(%)
G-32x32-20-0.5	10	15.07	0.00	0.00	10	13.37	0.00	0.00	10	18.10	0.00	0.00
G-32x32-20-0.9	10	18.10	0.00	0.00	10	16.05	0.00	0.00	10	23.00	0.00	0.00
G-32x32-200-0.5	10	16.38	0.00	0.00	10	12.20	0.00	0.00	10	18.98	0.00	0.00
G-32x32-200-0.9	10	18.31	0.00	0.00	10	15.83	0.00	0.00	10	22.45	0.00	0.00
G-20x50-20-0.5	10	13.03	0.00	0.00	10	11.56	0.00	0.00	10	15.91	0.00	0.00
G-20x50-20-0.9	10	18.38	0.00	0.00	10	15.90	0.00	0.00	10	21.90	0.00	0.00
G-20x50-200-0.5	10	12.14	0.00	0.00	10	11.30	0.00	0.00	10	15.82	0.00	0.00
G-20x50-200-0.9	10	21.96	0.00	0.00	10	18.99	0.00	0.00	10	26.11	0.00	0.00
G-5x200-20-0.5	10	438.82	0.00	0.00	10	422.14	0.00	0.00	10	395.55	0.00	0.00
G-5x200-20-0.9	9	723.82	0.28	0.89	9	745.53	0.24	0.77	9	672.59	0.26	0.83
G-5x200-200-0.5	10	158.83	0.00	0.00	10	168.12	0.00	0.00	10	164.71	0.00	0.00
G-5x200-200-0.9	7	791.80	0.25	0.50	7	758.44	0.20	0.43	7	722.27	0.21	0.44
G-44x44-20-0.5	10	47.17	0.00	0.00	10	41.31	0.00	0.00	10	56.35	0.00	0.00
G-44x44-20-0.9	10	53.50	0.00	0.00	10	49.97	0.00	0.00	10	71.33	0.00	0.00
G-44x44-200-0.5	10	48.96	0.00	0.00	10	43.35	0.00	0.00	10	60.96	0.00	0.00
G-44x44-200-0.9	10	66.32	0.00	0.00	10	62.63	0.00	0.00	10	85.82	0.00	0.00
G-20x100-20-0.5	10	72.12	0.00	0.00	10	59.44	0.00	0.00	10	86.54	0.00	0.00
G-20x100-20-0.9	10	110.83	0.00	0.00	10	127.95	0.00	0.00	10	148.63	0.00	0.00
G-20x100-200-0.5	10	86.12	0.00	0.00	10	72.09	0.00	0.00	10	90.85	0.00	0.00
G-20x100-200-0.9	10	79.23	0.00	0.00	10	94.49	0.00	0.00	10	108.14	0.00	0.00
G-5x400-20-0.5	0	-	2.89	2.04	0	-	2.22	1.54	1	3602.02	2.07	1.43
G-5x400-20-0.9	0	-	7.35	3.64	0	-	6.09	3.03	0	-	5.56	2.83
G-5x400-200-0.5	1	3228.53	5.29	4.12	1	3134.86	4.21	3.43	1	3454.40	3.87	3.03
G-5x400-200-0.9	0	-	10.96	2.87	0	-	8.72	2.35	0	-	7.95	2.06
Average			1.13	0.59			0.90	0.48			0.83	0.44

With respect to Karaşan instances (Table 6.6), the average optimality gaps referred to the solutions provided by RS-Benders, HS-Benders and RS&HS-Benders are up to, respectively, 7.40%, 5.90% and 5.78% for the instances with 1000 vertices (see K-1000-20-0.9-5), whereas those of Standard Benders are up to 7.54% for the same instances (see Table 6.3, K-1000-20-0.9-5). For the instances with 2000 vertices, the average optimality gaps referred to the solutions provided by RS-Benders, HS-Benders and RS&HS-Benders are up to, respectively, 22.08%, 18.87% and 16.10% (see K-2000-200-0.9-5), whereas those of Standard Benders are up to 25.90% for the same instances (see Table 6.3, K-2000-200-0.9-5). Notice that the average optimality gap of RS&HS-Benders over all Karaşan instances (3.07%) is the smallest among the three algorithms considered, followed by that of HS-Benders (3.45%). In fact, the average gaps of the solutions provided by RS&HS-Benders are smaller than or equal to those of the other two algorithms for all sets of instances. Also notice that, for a same number of vertices, the smaller the value of ω is, the harder to solve the instances are.

Regarding Coco instances (Table 6.7), the average optimality gaps referred to the solutions provided by RS-Benders, HS-Benders and RS&HS-Benders are up to, respectively, 10.96%, 8.72% and 7.95% (see K-5x400-200-0.9). Recall that the average optimality gaps referred to the solutions provided by Standard Benders are up to 11.69% for the same instances (see Table 6.4, K-5x400-200-0.9). Also observe that the instances based on 5x400 grids remain harder to solve than the other grid instances. The average optimality gap of RS&HS-Benders over all Coco instances (0.83%) is the smallest among the three algorithms considered, followed by that of HS-Benders (0.90%). In addition, RS&HS-Benders was always able to provide tighter average optimality gaps for the hardest instances (5x400 grids).

Table 6.8: Computational results of Extended RS-Benders, Extended HS-Benders and Extended RS&HS-Benders for the layered and acyclic digraph instances, with 3600 seconds of time limit.

	Extended RS-Benders				**Extended HS-Benders**				**Extended RS&HS-Benders**			
			GAP				GAP				GAP	
Test set	#opt	Time(s)	Avg(%)	StDev(%)	#opt	Time(s)	Avg(%)	StDev(%)	#opt	Time(s)	Avg(%)	StDev(%)
K-1000-20-0.5-5	10	1402.40	0.00	0.00	10	1320.75	0.00	0.00	10	1334.42	0.00	0.00
K-1000-20-0.9-5	2	1150.47	4.36	3.56	2	1107.51	3.60	2.94	2	1112.97	3.67	2.88
K-1000-200-0.5-5	10	693.81	0.00	0.00	10	737.45	0.00	0.00	10	697.54	0.00	0.00
K-1000-200-0.9-5	3	1261.35	1.79	2.26	4	2026.45	1.42	1.68	3	1318.72	1.52	1.74
K-1000-20-0.5-10	10	159.41	0.00	0.00	10	170.67	0.00	0.00	10	166.75	0.00	0.00
K-1000-20-0.9-10	10	248.15	0.00	0.00	10	277.77	0.00	0.00	10	264.57	0.00	0.00
K-1000-200-0.5-10	10	131.18	0.00	0.00	10	171.17	0.00	0.00	10	138.48	0.00	0.00
K-1000-200-0.9-10	10	388.30	0.00	0.00	10	444.51	0.00	0.00	10	408.31	0.00	0.00
K-1000-20-0.5-25	10	39.64	0.00	0.00	10	57.86	0.00	0.00	10	58.61	0.00	0.00
K-1000-20-0.9-25	10	62.95	0.00	0.00	10	91.70	0.00	0.00	10	86.18	0.00	0.00
K-1000-200-0.5-25	10	41.42	0.00	0.00	10	68.28	0.00	0.00	10	65.13	0.00	0.00
K-1000-200-0.9-25	10	72.12	0.00	0.00	10	134.55	0.00	0.00	10	107.50	0.00	0.00
K-2000-20-0.5-5	0	-	10.06	4.83	0	-	9.11	3.94	0	-	8.66	3.75
K-2000-20-0.9-5	0	-	16.97	2.50	0	-	16.53	2.89	0	-	14.92	2.61
K-2000-200-0.5-5	0	-	10.35	3.16	0	-	8.67	2.23	0	-	8.50	2.46
K-2000-200-0.9-5	0	-	17.88	3.09	0	-	17.47	3.18	0	-	15.58	2.63
K-2000-20-0.5-10	8	1446.07	0.63	1.40	8	1586.70	0.58	1.35	8	1560.16	0.63	1.41
K-2000-20-0.9-10	1	3586.15	3.96	3.46	0	-	3.82	3.02	0	-	3.75	3.00
K-2000-200-0.5-10	8	2337.12	0.43	1.22	6	2158.96	0.53	1.10	6	1972.94	0.33	0.83
K-2000-200-0.9-10	0	-	2.72	2.44	1	3600.63	2.76	2.05	1	3398.78	2.46	2.00
K-2000-20-0.5-25	10	311.08	0.00	0.00	10	404.45	0.00	0.00	10	395.10	0.00	0.00
K-2000-20-0.9-25	10	417.52	0.00	0.00	10	763.11	0.00	0.00	10	653.37	0.00	0.00
K-2000-200-0.5-25	10	296.03	0.00	0.00	10	474.49	0.00	0.00	10	416.05	0.00	0.00
K-2000-200-0.9-25	10	564.58	0.00	0.00	10	946.03	0.00	0.00	10	789.25	0.00	0.00
Average			2.88	1.16			2.69	1.02			2.50	0.97

Table 6.9: Computational results of Extended RS-Benders, Extended HS-Benders and Extended RS&HS-Benders for the grid digraph instances, with 3600 seconds of time limit.

	Extended RS-Benders				**Extended HS-Benders**				**Extended RS&HS-Benders**			
			GAP				GAP				GAP	
Test set	#opt	Time(s)	Avg(%)	StDev(%)	#opt	Time(s)	Avg(%)	StDev(%)	#opt	Time(s)	Avg(%)	StDev(%)
G-32x32-20-0.5	10	25.26	0.00	0.00	10	29.37	0.00	0.00	10	28.40	0.00	0.00
G-32x32-20-0.9	10	32.97	0.00	0.00	10	34.57	0.00	0.00	10	32.79	0.00	0.00
G-32x32-200-0.5	10	28.83	0.00	0.00	10	32.15	0.00	0.00	10	30.28	0.00	0.00
G-32x32-200-0.9	10	38.98	0.00	0.00	10	41.91	0.00	0.00	10	40.24	0.00	0.00
G-20x50-20-0.5	10	19.43	0.00	0.00	10	21.11	0.00	0.00	10	21.45	0.00	0.00
G-20x50-20-0.9	10	32.76	0.00	0.00	10	42.10	0.00	0.00	10	36.39	0.00	0.00
G-20x50-200-0.5	10	26.78	0.00	0.00	10	33.11	0.00	0.00	10	30.85	0.00	0.00
G-20x50-200-0.9	10	47.10	0.00	0.00	10	52.38	0.00	0.00	10	45.62	0.00	0.00
G-5x200-20-0.5	10	311.47	0.00	0.00	10	323.36	0.00	0.00	10	298.33	0.00	0.00
G-5x200-20-0.9	10	736.58	0.00	0.00	9	468.19	0.02	0.06	9	425.86	0.02	0.06
G-5x200-200-0.5	10	169.03	0.00	0.00	10	195.68	0.00	0.00	10	178.08	0.00	0.00
G-5x200-200-0.9	10	699.52	0.00	0.00	10	714.88	0.00	0.00	10	675.15	0.00	0.00
G-44x44-20-0.5	10	79.63	0.00	0.00	10	101.89	0.00	0.00	10	89.42	0.00	0.00
G-44x44-20-0.9	10	116.86	0.00	0.00	10	150.13	0.00	0.00	10	120.80	0.00	0.00
G-44x44-200-0.5	10	106.70	0.00	0.00	10	130.85	0.00	0.00	10	112.61	0.00	0.00
G-44x44-200-0.9	10	153.18	0.00	0.00	10	182.59	0.00	0.00	10	166.50	0.00	0.00
G-20x100-20-0.5	10	129.51	0.00	0.00	10	147.21	0.00	0.00	10	143.08	0.00	0.00
G-20x100-20-0.9	10	227.48	0.00	0.00	10	280.96	0.00	0.00	10	242.72	0.00	0.00
G-20x100-200-0.5	10	195.65	0.00	0.00	10	217.25	0.00	0.00	10	197.53	0.00	0.00
G-20x100-200-0.9	10	190.06	0.00	0.00	10	260.44	0.00	0.00	10	255.36	0.00	0.00
G-5x400-20-0.5	6	2850.81	0.81	1.36	7	2988.21	0.52	0.89	6	2961.06	0.52	0.83
G-5x400-20-0.9	1	3034.43	4.53	3.40	1	2923.29	3.57	2.37	1	2847.87	3.42	2.44
G-5x400-200-0.5	4	2501.02	2.50	2.59	4	2530.38	2.12	2.24	4	2503.07	1.94	2.11
G-5x400-200-0.9	0	-	7.56	2.94	0	-	5.81	2.28	0	-	5.47	2.01
Average			0.64	0.43			0.50	0.33			0.47	0.31

With respect to Karaşan instances (Table 6.8), the average optimality gaps referred to the solutions provided by Extended RS-Benders, Extended HS-Benders and Extended RS&HS-Benders are up to, respectively, 4.36%, 3.60% and 3.67% for the instances with 1000 vertices (see K-1000-20-0.9-5), whereas those of Extended Benders are up to 4.35% for the same instances (see Table 6.3, K-1000-20-0.9-5). For the instances with 2000 vertices, the average optimality gaps referred to the solutions provided by Extended RS-Benders, Extended HS-Benders and Extended RS&HS-Benders are up to, respectively, 17.88%, 17.47% and 15.58% (see K-2000-200-0.9-5), whereas those of Extended Benders are up to 21.41% for the same instances (see Table 6.3, K-2000-200-0.9-5). Moreover, the average optimality gap of Extended RS&HS-Benders over all Karaşan instances (2.50%) is the smallest among the three algorithms considered, followed by that of Extended HS-Benders (2.69%).

Regarding Coco instances (Table 6.9), the average optimality gaps referred to the solutions provided by Extended RS-Benders, Extended HS-Benders and Extended RS&HS-Benders are up to, respectively, 7.56%, 5.81% and 5.47% (see K-5x400-200-0.9). Recall that the average optimality gaps referred to the solutions provided by Extended Benders are up to 7.92% for the same instances (see Table 6.4, K-5x400-200-0.9). Also in this case, notice that the instances based on 5x400 grids remain harder to solve than the other grid instances. Moreover, the average optimality gap of Extended RS&HS-Benders over all Coco instances (0.47%) is the smallest among the algorithms considered, followed by that of Extended HS-Benders (0.50%). Indeed, the average gaps of the solutions provided by Extended RS&HS-Benders are smaller than or equal to those of the other algorithms in Table 6.9 for all sets of instances, except for G-5x200-20-0.9.

Figures 6.3 and 6.4 summarize the results concerning the quality of the bounds obtained by the exact algorithms for the hardest instance sets considered. Precisely, Figure 6.3 displays, for each algorithm, the average and the standard deviation of the relative optimality gaps referred to the instance set G-5x400-200-0.9, whereas Figure 6.4 shows those values referred to the instance sets K-1000-20-0.9-5 and K-2000-20-0.9-5. Notice that Extended RS&HS-Benders clearly outperforms, on average, the other algorithms in solving the three instance sets highlighted. In fact, when compared to Standard Benders, Extended RS&HS-Benders achieved an improvement of 53.21% in terms of average gaps for G-5x400-200-0.9. With respect to K-1000-20-0.9-5 and K-2000-20-0.9-5, that improvement was of 51.32% and 39.84%, respectively.

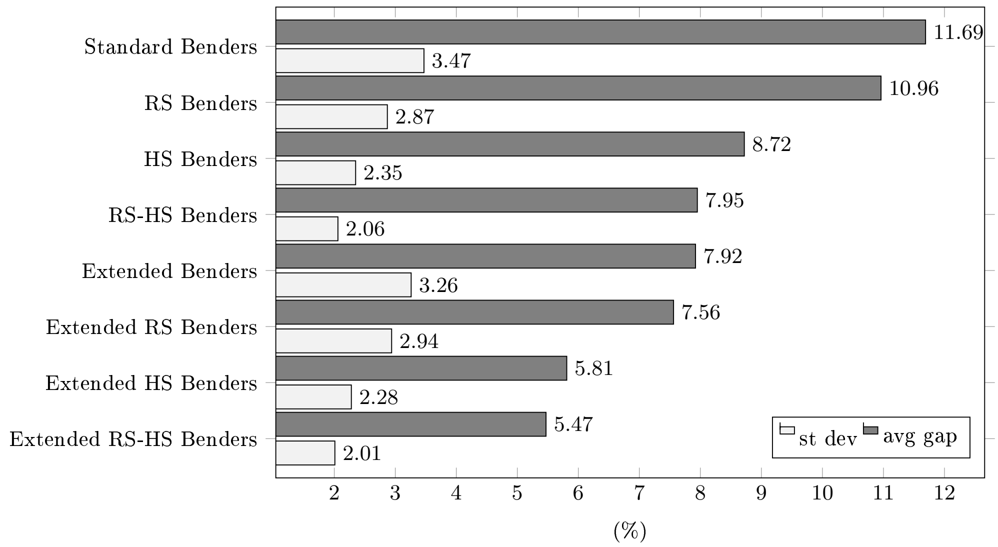
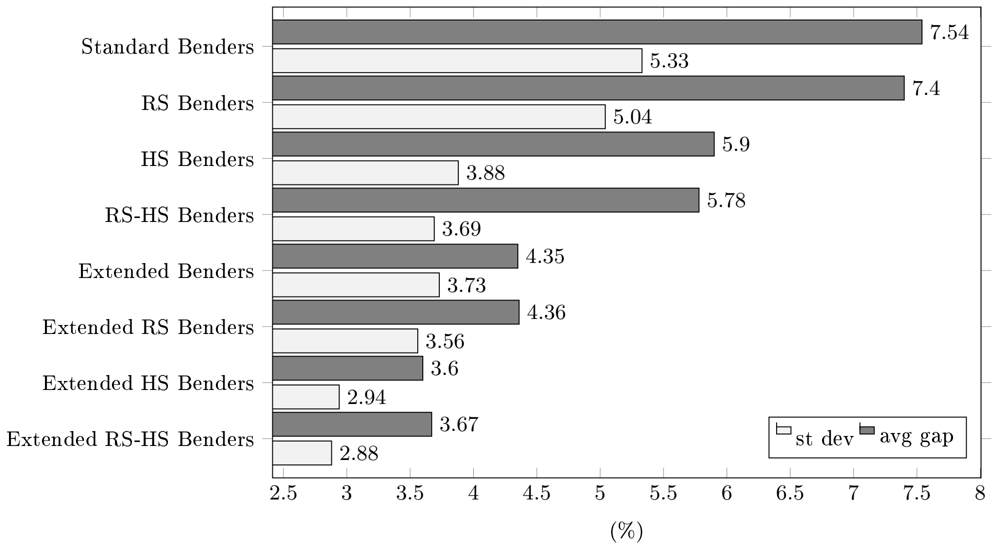


Figure 6.3: Summary of the results concerning the quality of the bounds obtained by the exact algorithms for the hardest grid digraph instances (instance set G-5x400-200-0.9).

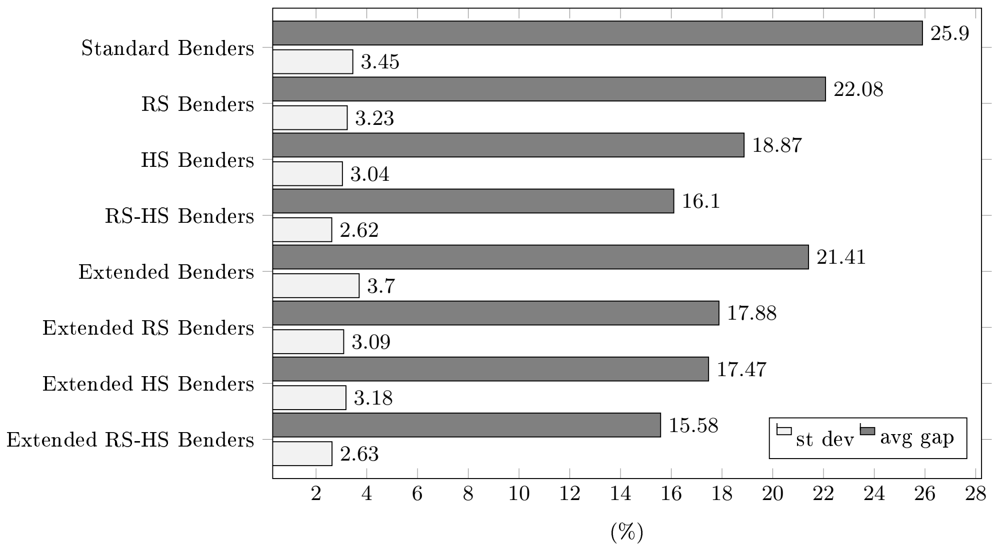
6.2.3 Discussion

From the results, LPH was able to provide optimal or near optimal solutions for most of the instance sets considered. Regarding the test sets for which not all optimality certificates were available, the average optimality gaps of the solutions obtained by LPH are close to the best among those of the exact methods. In fact, the average gap of LPH over all layered and acyclic instances was only 2.38% (see Table 6.1). Likewise, the average gaps of LPH over all grid instances was only 0.87% (see Table 6.2).

According to the results, Extended RS&HS-Benders outperforms (in terms of average optimality gaps achieved) the other exact algorithms presented in this study for both benchmarks of instances considered. Coupling the warm start procedures here discussed with the Benders-like decomposition approaches was successful in tightening the optimality gaps referred to the solutions obtained for the two benchmarks. Moreover, Extended Benders achieved, on average, better optimality gaps than Standard Benders for all sets of instances considered. The results suggest that generating Benders cuts referred to incumbent solutions (as in Extended Benders) is able to improve the overall quality of the bounds obtained. The results also indicate that the initial upper bounds provided by LPH play an important role in tightening the average optimality gaps referred to the exact algorithms that use HS and Extended HS as warm start procedures. Considering the test conditions established, none of the exact algorithms stood out in terms of time efficiency. On the other hand, we highlight that the average



(a) Instance set K-1000-20-0.9-5



(b) Instance set K-2000-20-0.9-5

Figure 6.4: Summary of the results concerning the quality of the bounds obtained by the exact algorithms for the hardest layered and acyclic digraph instances.

execution times of LPH was remarkably smaller than those of the exact algorithms for all instance sets considered.

We used different parameter combinations in the generation of the test instances. Notice that, considering a same number of vertices in a layered and acyclic digraph instance, a smaller width (given by the number ω of vertices per layer) implies more layers between the origin and the destination vertices. In this sense, the results regarding Karaşan instances suggest that the exact algorithms (as well as LPH) benefit from the degrowth of the networks' number of layers. In fact, instances generated under $\omega = 5$

(specially the ones with 2000 vertices) are the hardest ones (among Karaşan instances) for all algorithms. Furthermore, instances based on 5x400 grids are the hardest ones (among Coco instances) for all the algorithms considered.

For the hardest instances, the results also suggest that networks generated under a higher δ value (in particular, $\delta = 0.9$) tend to become even more difficult to be solved by any of the exact algorithms tested. A possible explanation for this behavior is that higher δ values might increase the occurrence of overlapping cost intervals, as pointed out in other RO studies in which the uncertainty is generated in a similar manner [Karaşan et al., 2001; Pereira and Averbakh, 2013]. From the results, it is not clear how the variation of Φ_{max} (parameter used to define the case base scenario Φ, from which the uncertainty is generated) interferes in the performance of the algorithms analysed.

Chapter 7

Concluding remarks

In this study, we formally defined a new robust-hard problem, namely the Restricted Robust Shortest Path problem (R-RSP). Furthermore, some theoretical aspects of the problem were discussed, including its computational complexity. Indeed, we showed that both R-RSP and its decision version are NP-hard. We derived a MILP formulation (with a polynomial number of variables and an exponential number of constraints) for R-RSP. Based on this formulation, we proposed a heuristic method, namely LPH, that uses dual information of R-RSP's classical optimization problem counterpart. Furthermore, we adapted to R-RSP a standard Benders-like decomposition approach that is a state-of-the-art method to solve RO problems. In addition, we discussed some techniques able to improve the convergence speed of the exact method by providing initial bounds, as well as by generating additional Benders cuts. These techniques were coupled with the Benders-like decomposition approach in different manners to generate a total of eight exact algorithms.

We introduced two benchmarks of instances adapted from the literature of a well-known RO problem related to R-RSP. The first benchmark consists of layered and acyclic digraphs, whereas the second one is based on grid digraphs. These benchmarks were used in computational experiments in order to evaluate the effectiveness of the proposed algorithms. With respect to LPH, the results indicate that the heuristic is effective in solving the two benchmarks of instances considered. In fact, the average gap of LPH over all layered and acyclic instances was only 2.38%. Likewise, the average gaps of LPH over all grid instances was only 0.87%. Moreover, the average processing times of LPH were remarkably smaller than those of the exact algorithms for both benchmarks considered. These results point out to the fact that the proposed heuristic approach may also be efficient in finding good quality solutions for other robust-hard problems.

With respect to the exact algorithms, the results of the experiments suggest that solely generating Benders cuts referred to incumbent solutions is able to improve the overall quality of the bounds obtained by the standard Benders-like decomposition algorithm for both benchmarks. Additional experiments were conducted in order to evalutate the impact of the speed-up techniques on the quality of the bounds obtained by the Benders-like decomposition algorithms. For all of the test sets considered, the improved algorithms were able to tighten the average optimality gaps obtained by the standard algorithm within 3600 seconds of execution. In addition, new optimality certificates were found for some of the more challenging sets of instances. Particularly, Extended RS&HS-Benders outperforms (in terms of average optimality gaps achieved) the other exact algorithms presented in this study for both benchmarks of instances considered.

This study opens several directions for further research on the topic. For instance, local search strategies, as Local Branching [Fischetti and Lodi, 2003], can be added to our heuristic method in order to further improve the quality of the solutions obtained. Recall that, in this study, the MILP heuristic problems that arise in LPH are directly handled with an optimization solver. Investigating alternative ways to solve such problems might lead to even more efficient procedures. Research can also be done in order to apply the proposed heuristic, as well as the improved exact methods, to solve other robust-hard problems. Further research may also investigate complexity issues concerning robust-hard problem in general. Such study is particularly relevant, as it may enable the development of more tractable mathematical formulations for this class of problems.

Bibliography

Aissi, H., Bazgan, C., and Vanderpooten, D. (2005). Complexity of the min-max and min-max regret assignment problems. *Operations Research Letters*, 33(6):634–640.

Aissi, H., Bazgan, C., and Vanderpooten, D. (2007). Approximation of min-max and min-max regret versions of some combinatorial optimization problems. *European Journal of Operational Research*, 179(2):281–290.

Aissi, H., Bazgan, C., and Vanderpooten, D. (2009). Min-max and min-max regret versions of combinatorial optimization problems: A survey. *European Journal of Operational Research*, 197(2):427–438.

Aneja, Y. P., Aggarwal, V., and Nair, K. P. K. (1983). Shortest chain subject to side constraints. *Networks*, 13:295–302.

Apostolopoulos, G., Guérin, R., Kamat, S., and Tripathi, S. K. (1998). Quality of Service based routing: A performance perspective. *SIGCOMM Computer Communication Review*, 28(4):17–28.

Aron, I. D. and Hentenryck, P. V. (2004). On the complexity of the robust spanning tree problem with interval data. *Operations Research Letters*, 32(1):36–40.

Averbakh, I. (2005). Computing and minimizing the relative regret in combinatorial optimization with interval data. *Discrete Optimization*, 2(4):273–287.

Beasley, J. and Christofides, N. (1989). An algorithm for the resource constrained shortest path problem. *Networks*, 19:379–394.

Benders, J. F. (1962). Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4(1):238–252.

Bondy, J. A. and Murty, U. S. R. (1976). *Graph Theory with Applications*. Elsevier, New York.

Catanzaro, D., Labbé, M., and Salazar-Neumann, M. (2011). Reduction approaches for robust shortest path problems. *Computers & Operations Research*, 38:1610–1619.

Coco, A. A., Júnior, J. C. A., Noronha, T. F., and Santos, A. C. (2014a). An integer linear programming formulation and heuristics for the minmax relative regret robust shortest path problem. *Journal of Global Optimization*, 60(2):265–287.

Coco, A. A., Solano-Charris, E. L., Santos, A. C., Prins, C., and Noronha, T. F. (2014b). Robust optimization criteria: state-of-the-art and new issues. Technical report, Université de Technologie de Troyes, Troyes, France.

Dantzig, G. B., Fulkerson, D. R., and Johnson, S. M. (1954). Solution of a large-scale traveling-salesman problem. *Operations Research*, 2:393–410.

Deineko, V. G. and Woeginger, G. J. (2010). Pinpointing the complexity of the interval min-max regret knapsack problem. *Discrete Optimization*, 7(4):191–196.

Dumitrescu, I. and Boland, N. (2003). Improved preprocessing, labeling and scaling algorithms for the weight-constrained shortest path problem. *Networks*, 42(3):135–153.

Fischetti, M. and Lodi, A. (2003). Local branching. *Mathematical Programming*, 98(1-3):23–47.

Fischetti, M., Salvagnin, D., and Zanette, A. (2010). A note on the selection of Benders' cuts. *Mathematical Programming*, 124(1-2):175–182.

Garey, M. R. and Johnson, D. S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA.

Geoffrion, A. M. (1972). Generalized Benders decomposition. *Journal of Optimization Theory and Applications*, 10(4):237–260.

Gupta, S. K. and Rosenhead, J. (1968). Robustness in sequential investment decisions. *Management Science*, 15(2):B–18–B–29.

Handler, G. and Zang, I. (1980). A dual algorithm for the constrained shortest path problem. *Networks*, 10:293–310.

Hassin, R. (1992). Approximation schemes for the restricted shortest path problem. *Mathematics of Operations Research*, 17(1):36–42.

Joksch, H. (1966). The shortest route problem with constraints. *Journal of Mathematical Analysis and Applications*, 14(2):191–197.

Karaşan, O. E., Pinar, M. Ç., and Yaman, H. (2001). The robust shortest path problem with interval data. Technical report, Bilkent University, Ankara, Turkey.

Kasperski, A. (2008). *Discrete Optimization with Interval Data: Minmax Regret and Fuzzy Approach (Studies in Fuzziness and Soft Computing)*. Springer Berlin.

Kasperski, A., Kobylański, P., Kulej, M., and Zieliński, P. (2005). *Minimizing maximal regret in discrete optimization problems with interval data, in: Issues in Soft Computing. Decisions and Operations Research, O. Hryniewicz, J. Kacprzyk, D. Kuchta (eds.)*. Akademicka Oficyna Wydawnicza EXIT, Warszawa.

Kasperski, A. and Zieliński, P. (2007). On the existence of an FPTAS for minmax regret combinatorial optimization problems with interval data. *Operations Research Letters*, 35(4):525–532.

Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220(4598):671–680.

Kouvelis, P. and Yu, G. (1997). *Robust Discrete Optimization and Its Applications*. Kluwer Academic Publishers, Boston.

Magnanti, T. L. and Wong, R. T. (1981). Accelerating Benders decomposition: Algorithmic enhancement and model selection criteria. *Operations Research*, 29(3):464–484.

McDaniel, D. and Devine, M. (1977). A modified Benders' partitioning algorithm for mixed integer programming. *Management Science*, 24(3):312–319.

Meyer, A. R. and Stockmeyer, L. J. (1972). The equivalence problem for regular expressions with squaring requires exponential space. In *Proceedings of the 13th Annual Symposium on Switching and Automata Theory (Swat 1972)*, pages 125–129. IEEE Computer Society.

Montemanni, R. (2006). A Benders decomposition approach for the robust spanning tree problem with interval data. *European Journal of Operational Research*, 174(3):1479–1490.

Montemanni, R., Barta, J., and Gambardella, L. (2006). Heuristic and preprocessing techniques for the robust traveling salesman problem with interval data. Technical report, Dalle Molle Institute for Artificial Intelligence.

Montemanni, R., Barta, J., and Gambardella, L. M. (2007). The robust traveling salesman problem with interval data. *Transportation Science*, 41:366–381.

Montemanni, R. and Gambardella, L. M. (2004). An exact algorithm for the robust shortest path problem with interval data. *Computers & Operations Research*, 31(10):1667–1680.

Montemanni, R. and Gambardella, L. M. (2005). The robust shortest path problem with interval data via Benders decomposition. *4OR*, 3:315–328.

Montemanni, R., Gambardella, L. M., and Donati, A. V. (2004). A branch and bound algorithm for the robust shortest path problem with interval data. *Operations Research Letters*, 32(3):225–232.

Papadimitriou, C. H. (1994). *Computational Complexity*. Addison-Wesley.

Pereira, J. and Averbakh, I. (2011). Exact and heuristic algorithms for the interval data robust assignment problem. *Computers & Operations Research*, 38(8):1153–1163.

Pereira, J. and Averbakh, I. (2013). The robust set covering problem with interval data. *Annals of Operations Research*, 207(1):217–235.

Pérez, F., Astudillo, C., Bardeen, M., and Candia-Véjar, A. (2012). A simulated annealing approach for the minmax regret path problem. In *Proceedings of the Congresso Latino Americano de Investigación Operativa (CLAIO)-Simpósio Brasileiro de Pesquisa Operacional (SBPO) 2012. Rio de Janeiro, Brazil.*

Pugliese, L. D. P. and Guerriero, F. (2013). A survey of resource constrained shortest path problems: Exact solution approaches. *Networks*, 62(3):183–200.

Revelle, C., Marks, D., and Liebman, J. C. (1970). An analysis of private and public sector location models. *Management Science*, 16(11):692–707.

Rosenhead, M. J., Elton, M., and Gupta, S. K. (1972). Robustness and optimality as criteria for strategic decisions. *Operational Research Quarterly*, 23:413–430.

Santos, L., Coutinho-Rodrigues, J., and Current, J. R. (2007). An improved solution algorithm for the constrained shortest path problem. *Transportation Research Part B: Methodological*, 41(7):756–771.

Shimbel, A. (1953). Structural parameters of communication networks. *The bulletin of mathematical biophysics*, 15(4):501–507.

Solano-Charris, E. L., Prins, C., and Santos, A. C. (2014). Heuristic approaches for the robust vehicle routing problem. In *Combinatorial Optimization*, Lecture Notes in Computer Science, pages 384–395. Springer International Publishing.

Spall, J. C. (2003). *Introduction to Stochastic Search and Optimization: Estimation, Simulation and Control*. Wiley, New York.

Sugiyama, K., Tagawa, S., and Toda, M. (1981). Methods for visual understanding of hierarchical system structures. *IEEE Transactions on Systems, Man & Cybernetics*, 11(2):109–125.

Wang, Z. and Crowcroft, J. (1996). Quality-of-Service routing for supporting multimedia applications. *IEEE on Selected Areas in Communications*, 14(7):1228–1234.

Yaman, H., Karaşan, O. E., and Pinar, M. Ç. (2001). The robust spanning tree problem with interval data. *Operations Research Letters*, 29:31–40.

Yu, G. (1996). On the max-min 0-1 knapsack problem with robust optimization applications. *Operations Research*, 44(2):407–415.

Zhu, X. and Wilhelm, W. E. (2012). A three-stage approach for the resource-constrained shortest path as a sub-problem in column generation. *Computers & Operations Research*, 39(2):164–178.

Zieliński, P. (2004). The computational complexity of the relative robust shortest path problem with interval data. *European Journal of Operational Research*, 158:570–576.

Appendix A

Additional tables

A.1 AMU

We adapted to R-RSP the scenario-based heuristic AMU proposed in Kasperski et al. [2005] for IRRSP. In the case of R-RSP, AMU consists in finding β-restricted shortest paths in two specific scenarios: (I) the *median scenario*, where the cost of each arc $(i, j) \in A$ is set to its respective mean value, given by $\frac{(l_{ij}+u_{ij})}{2}$, and (II) the *worst-case scenario*, where each arc cost is set to its corresponding upper bound. Then, the heuristic computes the β-restricted robustness cost of each of these two paths and returns the smallest value, along with its corresponding path.

With respect to AMU, the results for Karaşan and Coco instances are reported in Tables A.1 and A.2, respectively. The first column displays the name of each set of 10 instances. The second column shows the number of instances solved at optimality by any of the exact algorithms proposed in this study, within 3600 seconds of execution. The third and fourth columns show, respectively, the average and the standard deviation (over the 10 instances) of the relative optimality gaps given by $100 \cdot \frac{UB^{*}-LB^{*}}{UB^{*}}$. LB^{*} and UB^{*} are, respectively, the best lower and upper bounds obtained by any exact algorithm for a given instance. The fifth column displays the average processing time (in seconds) of AMU. The sixth column shows the average (over the 10 instances) of the heuristic optimality gaps given by $100 \cdot \frac{UB_{amu}-LB^{*}}{UB_{amu}}$, where UB_{amu} is the β-restricted robustness cost of the solution obtained by AMU for a given instance. The standard deviation of these gaps is given in the last column. Notice that LB^{*} remains the best lower bound obtained by any exact algorithm within 3600 seconds of execution and, therefore, it might not correspond to the cost of an optimal solution. Thus, the aforementioned heuristic gaps may overestimate the actual gaps between the cost of the heuristic solution and the cost of an optimal one.

Table A.1: Computational results of AMU for the layered and acyclic digraph instances.

	Exact algorithms			AMU		
Test set	#opt	AvgGAP(%)	StDev(%)	Time(s)	AvgGAP(%)	StDev(%)
K-1000-20-0.5-5	10	0.00	0.00	4.09	4.10	3.59
K-1000-20-0.9-5	2	3.58	2.90	4.07	7.83	3.81
K-1000-200-0.5-5	10	0.00	0.00	4.09	3.68	2.18
K-1000-200-0.9-5	5	1.36	1.64	4.03	7.45	2.46
K-1000-20-0.5-10	10	0.00	0.00	5.37	3.06	2.70
K-1000-20-0.9-10	10	0.00	0.00	5.40	4.52	3.20
K-1000-200-0.5-10	10	0.00	0.00	5.19	1.57	1.84
K-1000-200-0.9-10	10	0.00	0.00	5.18	4.50	2.98
K-1000-20-0.5-25	10	0.00	0.00	9.48	3.12	4.52
K-1000-20-0.9-25	10	0.00	0.00	9.66	2.35	3.69
K-1000-200-0.5-25	10	0.00	0.00	9.39	1.27	2.71
K-1000-200-0.9-25	10	0.00	0.00	9.50	1.13	1.89
K-2000-20-0.5-5	0	8.61	3.86	14.61	12.31	4.04
K-2000-20-0.9-5	0	14.90	2.60	14.68	17.66	1.88
K-2000-200-0.5-5	0	8.30	2.27	14.38	11.56	2.72
K-2000-200-0.9-5	0	15.52	2.59	14.43	18.70	3.02
K-2000-20-0.5-10	8	0.57	1.34	17.67	4.08	3.26
K-2000-20-0.9-10	1	3.46	2.99	17.41	6.90	3.57
K-2000-200-0.5-10	8	0.29	0.84	17.00	3.39	2.22
K-2000-200-0.9-10	1	2.32	1.99	17.08	6.31	3.27
K-2000-20-0.5-25	10	0.00	0.00	28.57	5.98	7.21
K-2000-20-0.9-25	10	0.00	0.00	28.44	1.46	1.54
K-2000-200-0.5-25	10	0.00	0.00	28.87	1.53	2.11
K-2000-200-0.9-25	10	0.00	0.00	28.04	2.70	2.76
Average		2.46	0.96		5.71	3.05

A.2 Warm Start procedures

Tables A.3 and A.4 display the average execution times referred to the warm start procedures for Coco and Karaşan instances, respectively. For each test set of 10 instances, the average execution times of RS, HS and Extended HS are given in seconds.

Table A.2: Computational results of AMU for the grid digraph instances.

	Exact algorithms			AMU		
Test set	#opt	AvgGAP(%)	StDev(%)	Time(s)	AvgGAP(%)	StDev(%)
G-32x32-20-0.5	10	0.00	0.00	4.35	4.16	9.20
G-32x32-20-0.9	10	0.00	0.00	4.53	5.38	7.49
G-32x32-200-0.5	10	0.00	0.00	4.43	1.56	2.72
G-32x32-200-0.9	10	0.00	0.00	4.30	3.90	5.89
G-20x50-20-0.5	10	0.00	0.00	3.75	2.71	5.94
G-20x50-20-0.9	10	0.00	0.00	3.89	1.80	2.93
G-20x50-200-0.5	10	0.00	0.00	4.13	2.54	5.09
G-20x50-200-0.9	10	0.00	0.00	4.03	2.33	4.43
G-5x200-20-0.5	10	0.00	0.00	3.80	5.29	4.89
G-5x200-20-0.9	10	0.00	0.00	3.77	4.40	1.69
G-5x200-200-0.5	10	0.00	0.00	3.83	4.82	3.57
G-5x200-200-0.9	10	0.00	0.00	3.86	7.81	3.87
G-44x44-20-0.5	10	0.00	0.00	13.27	0.97	1.92
G-44x44-20-0.9	10	0.00	0.00	13.21	3.92	8.06
G-44x44-200-0.5	10	0.00	0.00	12.69	5.05	4.02
G-44x44-200-0.9	10	0.00	0.00	12.45	2.70	3.23
G-20x100-20-0.5	10	0.00	0.00	14.01	2.81	5.26
G-20x100-20-0.9	10	0.00	0.00	13.84	5.41	4.98
G-20x100-200-0.5	10	0.00	0.00	14.45	1.64	1.97
G-20x100-200-0.9	10	0.00	0.00	13.59	3.37	3.27
G-5x400-20-0.5	7	0.50	0.85	13.27	5.50	3.36
G-5x400-20-0.9	1	3.33	2.32	13.21	8.43	3.30
G-5x400-200-0.5	4	1.92	2.09	13.26	6.77	3.12
G-5x400-200-0.9	0	5.45	2.00	13.11	10.52	3.70
Average		0.47	0.30		4.33	4.33

Table A.3: Average execution times (in seconds) of the warm start procedures for the grid digraph instances.

	Warm start procedure		
Test set	RS	HS	Extended HS
G-32x32-20-0.5	1.07	4.37	7.50
G-32x32-20-0.9	1.08	5.59	9.74
G-32x32-200-0.5	1.07	4.44	7.14
G-32x32-200-0.9	1.06	5.92	10.28
G-20x50-20-0.5	0.97	4.43	6.81
G-20x50-20-0.9	0.95	5.22	7.83
G-20x50-200-0.5	1.06	4.93	8.24
G-20x50-200-0.9	1.05	5.88	9.30
G-5x200-20-0.5	0.97	11.61	16.92
G-5x200-20-0.9	0.93	20.83	28.86
G-5x200-200-0.5	0.98	9.45	15.71
G-5x200-200-0.9	0.96	23.46	31.34
G-44x44-20-0.5	3.15	14.65	26.40
G-44x44-20-0.9	3.25	17.04	25.21
G-44x44-200-0.5	3.22	16.18	26.88
G-44x44-200-0.9	3.14	21.61	34.58
G-20x100-20-0.5	3.66	21.97	35.34
G-20x100-20-0.9	3.40	31.32	54.08
G-20x100-200-0.5	3.56	21.77	35.71
G-20x100-200-0.9	3.43	28.07	43.04
G-5x400-20-0.5	3.30	68.95	93.90
G-5x400-20-0.9	3.31	169.93	207.78
G-5x400-200-0.5	3.28	88.28	116.40
G-5x400-200-0.9	3.30	279.05	325.02

Table A.4: Average execution times (in seconds) of the warm start procedures for the layered and acyclic digraph instances.

	Warm start procedure		
Test set	RS	HS	Extended HS
K-1000-20-0.5-5	1.02	11.17	15.15
K-1000-20-0.9-5	1.02	31.66	39.88
K-1000-200-0.5-5	1.04	9.29	14.31
K-1000-200-0.9-5	1.02	23.22	30.74
K-1000-20-0.5-10	1.36	12.29	17.14
K-1000-20-0.9-10	1.34	19.32	25.96
K-1000-200-0.5-10	1.31	11.34	14.43
K-1000-200-0.9-10	1.32	26.81	34.01
K-1000-20-0.5-25	2.34	23.65	28.49
K-1000-20-0.9-25	2.40	29.57	35.54
K-1000-200-0.5-25	2.30	25.71	31.75
K-1000-200-0.9-25	2.26	33.15	38.71
K-2000-20-0.5-5	3.69	74.73	97.20
K-2000-20-0.9-5	3.72	288.43	323.54
K-2000-200-0.5-5	3.62	79.02	113.26
K-2000-200-0.9-5	3.60	557.90	586.53
K-2000-20-0.5-10	4.46	104.45	127.27
K-2000-20-0.9-10	4.39	292.36	333.73
K-2000-200-0.5-10	4.27	126.97	148.77
K-2000-200-0.9-10	4.30	238.33	268.74
K-2000-20-0.5-25	7.22	141.14	163.03
K-2000-20-0.9-25	7.02	263.55	286.09
K-2000-200-0.5-25	7.14	144.20	167.14
K-2000-200-0.9-25	6.98	249.86	273.17