



## Uma Heurística baseada em Programação Linear para o Problema do Caminho Mínimo Robusto Restrito

**Lucas Assunção**

Departamento de Ciência da Computação – Universidade Federal de Minas Gerais  
Belo Horizonte – MG – Brazil  
lucas-assuncao@ufmg.br

**Thiago F. Noronha**

Departamento de Ciência da Computação – Universidade Federal de Minas Gerais  
Belo Horizonte – MG – Brazil  
tfn@dcc.ufmg.br

**Andréa Cynthia Santos**

ICD-LOSI – Université de Technologie de Troyes  
Troyes Cedex – France  
andrea.duhamel@utt.fr

**Rafael Andrade**

Departamento de Estatística e Matemática Aplicada – Universidade Federal do Ceará  
Fortaleza – CE – Brazil  
rca@lia.ufc.br

### RESUMO

Tratamos o problema do caminho mínimo robusto restrito, uma versão de otimização robusta do problema do caminho mínimo restrito, clássico problema  $\mathcal{NP}$ -difícil. Os arcos estão associados a um intervalo de custo e a um valor de comprimento. O objetivo é encontrar um caminho de um vértice de origem a um de destino respeitando um limite máximo de comprimento e minimizando um critério robusto chamado de custo de robustez restrito. Um importante resultado teórico que diminui o espaço de busca em diversos problemas de otimização robusta é generalizado para o novo problema. Apresentamos uma heurística baseada em programação linear, juntamente com experimentos computacionais que atestam sua eficácia na resolução de instâncias adaptadas da literatura. **PALAVRAS CHAVE.** Caminho mínimo robusto restrito, Otimização robusta, Heurística, Programação linear inteira.

### ABSTRACT

We study the restricted robust shortest path problem, a robust optimization version of the restricted shortest path problem, classical  $\mathcal{NP}$ -hard problem. The arcs are associated with cost intervals and with a length value. The goal is to find a path connecting a source vertex to a target one respecting a maximum length constraint and minimizing a robust criterion called restricted robustness cost. An important theoretical result which reduces the search space on many robust optimization problems is generalized to this new problem. We present a heuristic based on linear programming, along with computational experiments that show its effectiveness on solving instances adapted from the literature. **KEYWORDS.** Restricted robust shortest path, Robust optimization, Heuristic, Integer linear programming.

## 1. Introdução

O problema do caminho mínimo com recursos limitados (RCSP, do inglês *resource constrained shortest path*) (Garey and Johnson, 1979; Handler and Zang, 1980) é uma variação do problema clássico do caminho mínimo entre dois vértices (SP, do inglês *shortest path problem*) (Shimbel, 1955). São dados um grafo direcionado  $G = (V, A)$ ,  $z$  limites de recurso, de  $\beta^{(1)}$  a  $\beta^{(z)}$ , além de um vértice de origem e um de destino. Cada arco  $(i, j) \in A$  apresenta um valor de custo  $c_{ij}$  e consome  $w_{ij}^{(m)}$  unidades de recurso,  $m$  de 1 a  $z$ . Tanto os valores de custo como os de recursos são não-negativos e aditivos ao longo do caminho. O objetivo é encontrar um caminho do vértice de origem ao de destino, com menor custo possível, que respeite as restrições de recurso impostas pelos limites  $\beta^{(1)}$  a  $\beta^{(z)}$ .

Uma versão simplificada de RCSP, em que apenas um recurso é considerado, ou seja,  $z = 1$ , é conhecida na literatura (Handler and Zang, 1980) por problema do caminho mínimo restrito (R-SP, do inglês *restricted shortest path*). Observe que, tanto em SP como em RCSP (e R-SP), cada arco apresenta um custo fixo. Quando se considera um fator de incerteza nos valores de custo associados a esses arcos, surgem novas versões do problema que podem ser modeladas e resolvidas através da otimização robusta (Kouvelis and Yu, 1997; Dias and Clímaco, 2000; Karaşan et al., 2001; Coco et al., 2014).

O problema do caminho mínimo robusto (RSP, do inglês *robust shortest path problem*), na versão com intervalos de custos (Dias and Clímaco, 2000; Karaşan et al., 2001), é definido para grafos orientados  $G = (V, A)$ , onde  $V$  representa o conjunto de vértices e  $A$  o conjunto de arcos. Cada arco  $(i, j) \in A$  está associado a um intervalo de custo  $[l_{ij}, u_{ij}]$ , onde  $l_{ij} \in \mathbb{Z}$  é o limite inferior, e  $u_{ij} \in \mathbb{Z}$  é o limite superior, com  $0 \leq l_{ij} \leq u_{ij}$ . São dados também um vértice de origem e um vértice de destino pertencentes a  $V$ .

Considere ainda o conjunto  $\mathcal{P}$  de todos os caminhos do vértice de origem ao de destino em  $G$  e o conjunto  $A[p]$  dos arcos de um caminho  $p \in \mathcal{P}$ . Podemos definir formalmente RSP através das definições a seguir.

**Definição 1.** Um cenário  $s$  é uma atribuição de valores de custo  $c_{ij}^s \in [l_{ij}, u_{ij}]$  para todo arco  $(i, j) \in A$ .

Seja  $\mathcal{S}$  o conjunto de todos os possíveis cenários de  $G$ . O custo de um caminho  $p \in \mathcal{P}$  em  $s \in \mathcal{S}$  é dado por  $C_p^s = \sum_{(i,j) \in A[p]} c_{ij}^s$ .

**Definição 2.** O desvio de robustez de um caminho  $p \in \mathcal{P}$  em um cenário  $s \in \mathcal{S}$ , denotado por  $r_p^s$ , é a diferença entre o custo  $C_p^s$  de  $p$  em  $s$  e o custo de um caminho mínimo  $p^*(s) \in \mathcal{P}$  em  $s$ , isto é,  $r_p^s = C_p^s - C_{p^*(s)}^s$ .

**Definição 3.** O custo de robustez de um caminho  $p \in \mathcal{P}$ , denotado por  $R_p$ , é definido como o maior desvio de robustez de  $p$  dentre todos os cenários em  $\mathcal{S}$ , isto é,  $R_p = \max_{s \in \mathcal{S}} r_p^s$ .

**Definição 4.** Um caminho  $p \in \mathcal{P}$  é dito um caminho robusto se ele possui o menor custo de robustez dentre todos os caminhos em  $\mathcal{P}$ , isto é,  $p = \arg \min_{p' \in \mathcal{P}} R_{p'}$ .

**Definição 5.** RSP consiste em encontrar um caminho robusto.

Define-se o cenário induzido por um caminho  $p \in \mathcal{P}$ , denotado por  $s_p$ , como o cenário em que cada arco em  $A[p]$  tem o valor de custo fixado no seu limite superior, enquanto todos os demais arcos têm os valores de custo fixados nos seus respectivos limites inferiores, isto é,  $c_{ij}^{s_p} = u_{ij} \forall (i, j) \in A[p]$  e  $c_{ij}^{s_p} = l_{ij} \forall (i, j) \in A \setminus A[p]$ .

**Observação 1.** (Karaşan et al., 2001) O desvio de robustez de todo caminho  $p \in \mathcal{P}$  é máximo no cenário  $s_p$ .

A Observação 1 reduz a quantidade de cenários que precisam ser considerados na busca por um caminho robusto, uma vez que, dado um caminho  $p \in \mathcal{P}$ , o custo de robustez de  $p$  consiste no desvio de robustez  $r_p^{s_p}$  de  $p$  em  $s_p$ , ou seja,  $R_p = r_p^{s_p}$ .

O problema do caminho mínimo robusto restrito (R-RSP, do inglês *restricted robust shortest path problem*) é definido para grafos orientados  $G = (V, A)$ , onde  $V$  representa o conjunto de vértices e  $A$  o conjunto de arcos. Cada arco  $(i, j) \in A$  está associado a um valor de comprimento  $d_{ij} \in \mathbb{R}$ , com  $d_{ij} > 0$ , e a um intervalo de custo  $[l_{ij}, u_{ij}]$ , onde  $l_{ij} \in \mathbb{Z}$  é o limite inferior, e  $u_{ij} \in \mathbb{Z}$  é o limite superior, com  $0 \leq l_{ij} \leq u_{ij}$ . São dados ainda um valor  $\beta \in \mathbb{R}$ , com  $\beta > 0$ , um vértice de origem e um vértice de destino pertencentes a  $V$ . A Figura 1 mostra um exemplo de instância de R-RSP.

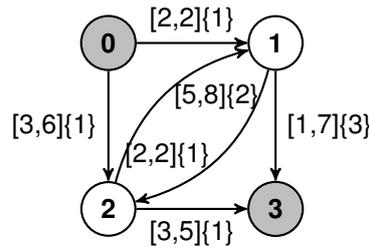


Figura 1. Exemplo de instância de R-RSP. Foi utilizada a notação  $[l_{ij}, u_{ij}]\{d_{ij}\}$  para representar o intervalo de custo e o valor de comprimento associados a cada arco  $(i, j) \in A$ . Os vértices 0 e 3 são, respectivamente, a origem e o destino.

Considere as definições relativas a RSP, o conjunto  $\mathcal{S}$  de todos os possíveis cenários e o conjunto  $\mathcal{P}$  dos caminhos do vértice de origem ao de destino em  $G$ . O comprimento de um caminho  $p \in \mathcal{P}$  é dado por  $D_p = \sum_{(i,j) \in A[p]} d_{ij}$ , e  $\mathcal{P}(\beta) \subseteq \mathcal{P}$  é o conjunto constituído pelos caminhos  $p \in \mathcal{P}$  de comprimento inferior ou igual a  $\beta$ , isto é,  $\mathcal{P}(\beta) = \{p \in \mathcal{P} \mid D_p \leq \beta\}$ . As definições a seguir descrevem formalmente R-RSP.

**Definição 6.** Um caminho  $p^*(s, \beta) \in \mathcal{P}(\beta)$  é dito um *caminho mínimo  $\beta$ -restrito* em um cenário  $s \in \mathcal{S}$  se ele possui o menor custo em  $s$  dentre todos os caminhos em  $\mathcal{P}(\beta)$ , isto é,  $p^*(s, \beta) = \arg \min_{p' \in \mathcal{P}(\beta)} C_{p'}^s$ .

**Definição 7.** O *desvio de robustez  $\beta$ -restrito* de um caminho  $p \in \mathcal{P}(\beta)$  em um cenário  $s \in \mathcal{S}$ , denotado por  $r_p^{(s, \beta)}$ , é a diferença entre o custo  $C_p^s$  de  $p$  em  $s$  e o custo de um caminho mínimo  $\beta$ -restrito  $p^*(s, \beta) \in \mathcal{P}(\beta)$  em  $s$ , isto é,  $r_p^{(s, \beta)} = C_p^s - C_{p^*(s, \beta)}^s$ .

**Definição 8.** O *custo de robustez  $\beta$ -restrito* de um caminho  $p \in \mathcal{P}(\beta)$ , denotado por  $R_p^\beta$ , é definido como o maior desvio de robustez  $\beta$ -restrito de  $p$  dentre todos os cenários possíveis, isto é,  $R_p^\beta = \max_{s \in \mathcal{S}} r_p^{(s, \beta)}$ .

**Definição 9.** Um caminho  $p \in \mathcal{P}(\beta)$  é dito um *caminho robusto  $\beta$ -restrito* se ele possui o menor custo de robustez  $\beta$ -restrito dentre todos os caminhos em  $\mathcal{P}(\beta)$ , isto é,  $p = \arg \min_{p' \in \mathcal{P}(\beta)} R_{p'}^\beta$ .

**Definição 10.** R-RSP consiste em encontrar um caminho robusto  $\beta$ -restrito.

R-RSP é uma generalização do problema do caminho mínimo restrito (R-SP), apresentando ainda forte relação com o problema do caminho mínimo robusto (RSP). Até onde sabemos, R-RSP é um problema inédito na literatura, apresentando aplicação em problemas de roteamento.

Um exemplo de aplicação consiste na busca por uma rota que minimize o tempo total de percurso entre duas localidades. Condições climáticas, congestionamentos, acidentes e diversos outros fatores influenciam no tempo necessário para percorrer as vias disponíveis. Essa incerteza pode ser representada por intervalos de valores, estimando-se o tempo mínimo e o tempo máximo necessários para percorrer cada via. Com o surgimento

de veículos automotores movidos a fontes energéticas sustentáveis, como carros elétricos, torna-se necessário estabelecer um limite na distância total a ser percorrida pelo veículo, de forma a não ultrapassar a autonomia proporcionada pela carga energética disponível, uma vez que o número de pontos de abastecimento adequados a esses veículos tende a ser limitado. Dessa forma, R-RSP pode ser utilizado para modelar caminhos que visem minimizar o tempo total de percurso, quaisquer que sejam as condições das vias, respeitando ainda uma restrição de distância máxima para a rota.

## 2. Trabalhos Relacionados

A complexidade de RCSP é estudada em (Garey and Johnson, 1979), onde os autores mostram que o problema é  $\mathcal{NP}$ -completo. Em (Handler and Zang, 1980), é atestado que, mesmo para a versão de RCSP que considera a restrição de um único recurso (R-SP), o problema continua  $\mathcal{NP}$ -completo. Em (Wang and Crowcroft, 1996), é mostrado que aciclicidade não impede que RCSP se mantenha  $\mathcal{NP}$ -completo.

Foram propostos algoritmos exatos para solucionar RCSP utilizando relaxação linear, como os apresentados em (Handler and Zang, 1980) e (Beasley and Christofides, 1989). Procedimentos de pré-processamento foram introduzidos por (Aneja et al., 1983) e aprimorados em (Beasley and Christofides, 1989). Esses procedimentos fazem uso de custos reduzidos obtidos pela solução do dual Lagrangeano para eliminar arcos e vértices que não podem fazer parte da solução ótima. Um algoritmo pseudo-polinomial para RCSP é apresentado em (Hassin, 1992), juntamente com um algoritmo  $\epsilon$ -aproximativo que executa em tempo  $\mathcal{O}(|V||A|/\epsilon)$ .

Em (Karaşan et al., 2001), o problema do caminho mínimo robusto com intervalos de custo é estudado. São apresentados resultados que reduzem a quantidade de cenários a serem considerados no problema, juntamente com uma formulação de programação inteira mista. São ainda desenvolvidas estratégias de pré-processamento que identificam arcos que não podem fazer parte da solução ótima em grafos planares, acíclicos e em camadas. A conjectura de que essa versão do problema do caminho mínimo robusto é  $\mathcal{NP}$ -difícil só é provada em (Zieliński, 2004).

Um método de resolução exata para o problema foi proposto em (Montemanni and Gambardella, 2004), onde os autores desenvolveram um algoritmo baseado em ranqueamento de caminhos. Em seguida, um algoritmo *branch-and-bound* (R. Montemanni and Donati, 2004) foi apresentado. Foi elaborada ainda uma decomposição de Benders em (Montemanni and Gambardella, 2005). Os algoritmos apresentados por Montemanni et al. representaram uma significativa evolução na resolução do problema do caminho mínimo robusto, sobretudo em relação ao tamanho das instâncias resolvidas. Recentemente, o trabalho de (Catanzaro et al., 2011) surgiu como uma extensão de (Karaşan et al., 2001), aperfeiçoando informações que determinam se um arco está ou não no caminho ótimo robusto e, assim, propondo novas estratégias de pré-processamento.

Versões robustas de vários problemas clássicos de otimização combinatória vêm sendo trabalhadas na literatura (Kasperski, 2008). A incerteza de custos nesses novos problemas dificulta sua resolução de tal forma que, em geral, problemas polinomiais na versão clássica passam a pertencer à classe  $\mathcal{NP}$ -difícil nas versões robustas correspondentes (Montemanni and Gambardella, 2005; Montemanni, 2006; Pereira and Averbakh, 2011). Recentemente, começaram a ser trabalhadas versões robustas de problemas que, na versão clássica, já são  $\mathcal{NP}$ -difícil. Em (Montemanni et al., 2007), por exemplo, o problema do caixeiro viajante robusto é estudado. Os autores apresentam três métodos exatos de resolução do problema: um algoritmo *branch-and-bound*, um algoritmo *branch-and-cut* e

uma decomposição de Benders. Em (Pereira and Averbakh, 2013), a versão robusta do problema de cobertura de conjuntos é estudada. São apresentados uma decomposição de Benders, um algoritmo *branch-and-cut*, além de um algoritmo genético e um algoritmo híbrido, combinando o esquema de decomposição de Benders com o algoritmo genético.

### 3. Formulação Matemática

Introduzimos uma formulação matemática para R-RSP baseada na proposição apresentada a seguir, que generaliza o resultado citado na Observação 1, reduzindo a quantidade de cenários que precisam ser considerados durante a busca por um caminho robusto  $\beta$ -restrito.

**Proposição 1.** Para qualquer valor  $\beta \in \mathbb{R}$  tal que  $\beta > 0$ , o desvio de robustez  $\beta$ -restrito de todo caminho  $p \in \mathcal{P}(\beta)$  é máximo no cenário  $s_p$  induzido por  $p$ .

*Demonstração.* Considere um valor  $\beta \in \mathbb{R}$ , com  $\beta > 0$ , um caminho  $p \in \mathcal{P}(\beta)$ , e seja  $s^* \in \mathcal{S}$  o cenário para o qual o desvio de robustez  $\beta$ -restrito de  $p$  é máximo, isto é,  $R_p^\beta = r_p^{(s^*, \beta)}$ . Pela definição de  $s_p$ , temos

$$\sum_{(i,j) \in A[p] \setminus A[p^*(s^*, \beta)]} c_{ij}^{s_p} \geq \sum_{(i,j) \in A[p] \setminus A[p^*(s^*, \beta)]} c_{ij}^{s^*} \quad (1)$$

$$\sum_{(i,j) \in A[p^*(s^*, \beta)] \setminus A[p]} c_{ij}^{s_p} \leq \sum_{(i,j) \in A[p^*(s^*, \beta)] \setminus A[p]} c_{ij}^{s^*} \quad (2)$$

Segue, então

$$R_p^\beta = r_p^{(s^*, \beta)} = C_p^{s^*} - C_{p^*(s^*, \beta)}^{s^*} = \sum_{(i,j) \in A[p] \setminus A[p^*(s^*, \beta)]} c_{ij}^{s^*} - \sum_{(i,j) \in A[p^*(s^*, \beta)] \setminus A[p]} c_{ij}^{s^*} \quad (3)$$

Aplicando-se (1) e (2) em (3)

$$R_p^\beta \leq \sum_{(i,j) \in A[p] \setminus A[p^*(s^*, \beta)]} c_{ij}^{s_p} - \sum_{(i,j) \in A[p^*(s^*, \beta)] \setminus A[p]} c_{ij}^{s_p} = C_p^{s_p} - C_{p^*(s^*, \beta)}^{s_p} \quad (4)$$

Como  $p^*(s_p, \beta)$  é um caminho mínimo  $\beta$ -restrito em  $s_p$ , temos

$$C_{p^*(s_p, \beta)}^{s_p} \leq C_{p^*(s^*, \beta)}^{s_p} \quad (5)$$

A partir de (4) e (5), obtemos

$$R_p^\beta \leq C_p^{s_p} - C_{p^*(s^*, \beta)}^{s_p} \leq C_p^{s_p} - C_{p^*(s_p, \beta)}^{s_p} = r_p^{(s_p, \beta)} \quad (6)$$

Portanto, o desvio de robustez  $\beta$ -restrito de  $p$  é máximo no cenário  $s_p$ , como queríamos provar.  $\square$

A Proposição 1 resume R-RSP a encontrar um caminho  $p$  tal que  $p = \arg \min_{p' \in \mathcal{P}} r_{p'}^{(s_{p'}, \beta)}$  ou, de forma equivalente,  $p = \arg \min_{p' \in \mathcal{P}(\beta)} \{C_{p'}^{s_{p'}} - C_{p^*(s_{p'}, \beta)}^{s_{p'}}\}$ . Considere as variáveis binárias  $y$ , que identificam os arcos que estão presentes em um caminho  $p \in \mathcal{P}(\beta)$  solução de R-RSP:  $y_{ij} = 1$  se o arco  $(i, j) \in A[p]$ ;  $y_{ij} = 0$  caso contrário. Analogamente, as variáveis binárias  $x$  identificam um caminho mínimo  $\beta$ -restrito  $p^*(s_p, \beta) \in \mathcal{P}(\beta)$  no

cenário induzido pelo caminho  $p$  definido por  $y$ :  $x_{ij} = 1$  se o arco  $(i, j) \in A[p^*(s_p, \beta)]$ ;  $x_{ij} = 0$  caso contrário. Uma formulação não-linear compacta para R-RSP é apresentada:

$$\min_{y \in \mathcal{P}(\beta)} \left( \sum_{(i,j) \in A} u_{ij} y_{ij} - \min_{x \in \mathcal{P}(\beta)} (l_{ij} + (u_{ij} - l_{ij}) y_{ij}) x_{ij} \right) \quad (7)$$

Para que a formulação (7) seja expressa na forma de um problema de programação linear inteira mista, é necessário que o operador de minimização aninhado, assim como a não-linearidade existente entre as variáveis  $x$  e  $y$ , sejam eliminados. Com esse objetivo, adicionamos uma variável irrestrita  $\rho$ , que passa a representar o valor do operador de minimização aninhado. Adicionamos ainda novas restrições lineares associadas a  $\rho$ . A formulação resultante, após a reorganização dos termos, é mostrada a seguir:

$$(R-RSP-F) \min \quad \sum_{(i,j) \in A} u_{ij} y_{ij} - \rho \quad (8)$$

$$s.a. \quad \rho \leq \sum_{(i,j) \in A} l_{ij} x_{ij} + \sum_{(i,j) \in A} ((u_{ij} - l_{ij}) y_{ij}) x_{ij} \quad \forall x \in \mathcal{P}(\beta) \quad (9)$$

$$\sum_{(j,i) \in A} y_{ji} - \sum_{(i,k) \in A} y_{ik} = -1 \quad \text{se } i \text{ é a origem} \quad (10)$$

$$\sum_{(j,i) \in A} y_{ji} - \sum_{(i,k) \in A} y_{ik} = 1 \quad \text{se } i \text{ é o destino} \quad (11)$$

$$\sum_{(j,i) \in A} y_{ji} - \sum_{(i,k) \in A} y_{ik} = 0 \quad \forall i \text{ intermediário} \quad (12)$$

$$\sum_{(i,j) \in A} d_{ij} y_{ij} \leq \beta \quad (13)$$

$$y_{ij} \in \{0, 1\} \quad \forall (i, j) \in A \quad (14)$$

$$\rho \text{ irrestrito} \quad (15)$$

As restrições (14) e (15) definem o domínio das variáveis. As restrições de fluxo (10)-(12) asseguram que  $y$  representa um caminho em  $\mathcal{P}$ , enquanto a restrição (13) limita o comprimento desse caminho ao máximo valor permitido  $\beta$ . Conjuntamente, as restrições (10)-(14) garantem que  $y$  representa um caminho em  $\mathcal{P}(\beta)$ . O conjunto de restrições (9) substitui o operador de minimização aninhado de (7) e incorpora o resultado obtido na Proposição 1 para o cálculo do custo de robustez  $\beta$ -restrito do caminho definido por  $y$ .

Note que o número de restrições (9) pode ser muito grande, uma vez que é proporcional à quantidade de caminhos presentes em  $\mathcal{P}(\beta)$ . Logo, em instâncias de R-RSP de dimensões reais, a utilização da formulação R-RSP-F diretamente como foi apresentada pode resultar em um modelo intratável.

#### 4. Heurística Baseada em Programação Linear

Apresentamos a seguir uma heurística baseada em programação linear que é constituída a partir da formulação para R-RSP mostrada em (7). Considere a formulação de programação linear inteira para o cálculo de um caminho mínimo  $\beta$ -restrito  $p^*(s, \beta) \in \mathcal{P}(\beta)$  em um cenário  $s \in \mathcal{S}$ . As variáveis binárias  $x$  identificam esse caminho, de tal forma que:  $x_{ij} = 1$  se o arco  $(i, j) \in A[p^*(s, \beta)]$ ;  $x_{ij} = 0$  caso contrário.

$$(R-SP-F) \min \sum_{(i,j) \in A} c_{ij}^s x_{ij} \quad (16)$$

$$s.a. \sum_{(j,i) \in A} x_{ji} - \sum_{(i,k) \in A} x_{ik} = -1 \quad \text{se } i \text{ é a origem} \quad (17)$$

$$\sum_{(j,i) \in A} x_{ji} - \sum_{(i,k) \in A} x_{ik} = 1 \quad \text{se } i \text{ é o destino} \quad (18)$$

$$\sum_{(j,i) \in A} x_{ji} - \sum_{(i,k) \in A} x_{ik} = 0 \quad \forall i \text{ intermediário} \quad (19)$$

$$\sum_{(i,j) \in A} d_{ij} x_{ij} \leq \beta \quad (20)$$

$$x_{ij} \in \{0, 1\} \quad \forall (i, j) \in A. \quad (21)$$

As restrições (17)-(19) e (21) garantem que  $x$  representa um caminho do vértice de origem ao de destino, enquanto (20) limita o comprimento desse caminho ao máximo valor permitido  $\beta$ . Conjuntamente, (17)-(21) asseguram que  $x$  representa um caminho em  $\mathcal{P}(\beta)$ . Relaxando a restrição de integralidade de  $x$ , obtemos a seguinte formulação de programação linear.

$$(R-SP-F-relax) \theta^{(s,\beta)} = \min \sum_{(i,j) \in A} c_{ij}^s x_{ij} \quad (22)$$

$$s.a. \sum_{(j,i) \in A} x_{ji} - \sum_{(i,k) \in A} x_{ik} = -1 \quad \text{se } i \text{ é a origem} \quad (23)$$

$$\sum_{(j,i) \in A} x_{ji} - \sum_{(i,k) \in A} x_{ik} = 1 \quad \text{se } i \text{ é o destino} \quad (24)$$

$$\sum_{(j,i) \in A} x_{ji} - \sum_{(i,k) \in A} x_{ik} = 0 \quad \forall i \text{ intermediário} \quad (25)$$

$$\sum_{(i,j) \in A} d_{ij} x_{ij} \leq \beta \quad (26)$$

$$x_{ij} \geq 0 \quad \forall (i, j) \in A. \quad (27)$$

Considere  $\theta^{(s,\beta)}$  o custo de uma solução ótima para a relaxação (R-SP-F-relax) em um cenário  $s$ . Observe que (R-SP-F-relax) se diferencia de (R-SP-F) apenas pela substituição da restrição (21) por (27). Portanto,  $\theta^{(s,\beta)}$  representa um limite inferior para (R-SP-F) no cenário  $s$ . Definimos uma nova métrica para avaliar a qualidade de um caminho em  $\mathcal{P}(\beta)$ .

**Definição 11.** O custo de robustez  $\beta$ -heurístico de um caminho  $p \in \mathcal{P}(\beta)$ , denotado por  $H_p^\beta$ , é a diferença entre o custo  $C_p^{s_p}$  de  $p$  no cenário  $s_p$  induzido por  $p$  e o custo relaxado  $\theta^{(s_p,\beta)}$  em  $s_p$ , isto é,  $H_p^\beta = C_p^{s_p} - \theta^{(s_p,\beta)}$ .

**Definição 12.** Um caminho  $p \in \mathcal{P}(\beta)$  é dito um caminho robusto  $\beta$ -heurístico se ele possui o menor custo de robustez  $\beta$ -heurístico dentre todos os caminhos em  $\mathcal{P}(\beta)$ , isto é,  $p = \arg \min_{p' \in \mathcal{P}(\beta)} H_{p'}^\beta$ .

Derivamos a seguinte relação entre as métricas custo de robustez  $\beta$ -restrito e custo de robustez  $\beta$ -heurístico.

**Proposição 2.** Para todo caminho  $p \in \mathcal{P}(\beta)$ , o custo de robustez  $\beta$ -heurístico de  $p$  é um limite superior para o custo de robustez  $\beta$ -restrito de  $p$ .

*Demonstração.* Considere um caminho  $p \in \mathcal{P}(\beta)$ . Pela Proposição 1, o custo de robustez  $\beta$ -restrito de  $p$  é dado por  $R_p^\beta = r_p^{(s_p, \beta)} = C_p^{s_p} - C_{p^*(s_p, \beta)}^{s_p}$ , onde  $p^*(s_p, \beta) \in \mathcal{P}(\beta)$  é um caminho mínimo  $\beta$ -restrito em  $s_p$ . Note que  $\theta^{(s, \beta)} \leq C_{p^*(s, \beta)}^s$  para todo cenário  $s \in \mathcal{S}$ , inclusive  $s_p$ , logo:

$$H_p^\beta = C_p^{s_p} - \theta^{(s_p, \beta)} \geq C_p^{s_p} - C_{p^*(s_p, \beta)}^{s_p} = R_p^\beta \quad (28)$$

□

A heurística aqui proposta se baseia na busca por um caminho robusto  $\beta$ -heurístico, problema que pode ser modelado adaptando a formulação (7): as variáveis binárias  $y$  passam a representar um caminho robusto  $\beta$ -heurístico  $p' \in \mathcal{P}(\beta)$ , e o operador de minimização aninhado é substituído por  $\theta^{(s_{y'}, \beta)}$ . Considerando, por simplificação,  $s_y$  o cenário induzido pelo caminho identificado pelas variáveis  $y$ , obtemos:

$$\min_{y \in \mathcal{P}(\beta)} \left( \sum_{(i,j) \in A} u_{ij} y_{ij} - \theta^{(s_y, \beta)} \right) \quad (29)$$

Dado um cenário  $s \in \mathcal{S}$ , o valor do operador  $\theta^{(s, \beta)}$  pode ser representado a partir do problema dual de (R-SP-F-relax), a saber:

$$\text{(Dual R-SP-F-relax)} \theta^{(s, \beta)} = \max \quad \lambda_{destino} - \lambda_{origem} - \beta \mu \quad (30)$$

$$s.a. \quad \lambda_j \leq \lambda_i + c_{ij}^s + d_{ij} \mu \quad \forall (i, j) \in A \quad (31)$$

$$\mu \geq 0 \quad (32)$$

$$\lambda_k \text{ irrestrito} \quad \forall k \in V. \quad (33)$$

As variáveis duais  $\{\lambda_k : k \in V\}$  e  $\mu$  estão associadas às restrições do problema primal (R-SP-F-relax). Por se tratar de um problema de maximização, (Dual R-SP-F-relax) é utilizado, então, para substituir o valor do custo relaxado  $\theta^{(s_y, \beta)}$  em (29).

$$\min_{y \in \mathcal{P}(\beta)} \left( \sum_{(i,j) \in A} u_{ij} y_{ij} - \overbrace{(\lambda_{destino} - \lambda_{origem} - \beta \mu)}^{\text{(Dual R-SP-F-relax)}} \right) \quad (34)$$

$$s.a. \quad \lambda_j \leq \lambda_i + l_{ij} + (u_{ij} - l_{ij}) y_{ij} + d_{ij} \mu \quad \forall (i, j) \in A \quad (35)$$

$$\mu \geq 0 \quad (36)$$

$$\lambda_k \text{ irrestrito} \quad \forall k \in V. \quad (37)$$

Observe que as restrições (35) passam a considerar o custo de cada arco  $(i, j) \in A$  no cenário induzido pelo caminho identificado pelas variáveis  $y$ , ou seja, o custo de cada

arco  $(i, j) \in A$  é dado por  $l_{ij} + (u_{ij} - l_{ij})y_{ij}$ . As restrições de domínio das variáveis duais de (Dual R-SP-F-relax) são mantidas inalteradas. Chegamos, por fim, à formulação que serve de base para a heurística.

$$(R-RSP-H) \min \sum_{(i,j) \in A} u_{ij}y_{ij} - \lambda_{destino} + \lambda_{origem} + \beta\mu \quad (38)$$

$$s.a. \quad \lambda_j \leq \lambda_i + l_{ij} + (u_{ij} - l_{ij})y_{ij} + d_{ij}\mu \quad \forall (i, j) \in A \quad (39)$$

$$\sum_{(j,i) \in A} y_{ji} - \sum_{(i,k) \in A} y_{ik} = -1 \quad \text{se } i \text{ é a origem} \quad (40)$$

$$\sum_{(j,i) \in A} y_{ji} - \sum_{(i,k) \in A} y_{ik} = 1 \quad \text{se } i \text{ é o destino} \quad (41)$$

$$\sum_{(j,i) \in A} y_{ji} - \sum_{(i,k) \in A} y_{ik} = 0 \quad \forall i \text{ intermediário} \quad (42)$$

$$\sum_{(i,j) \in A} d_{ij}y_{ij} \leq \beta \quad (43)$$

$$y_{ij} \in \{0, 1\} \quad \forall (i, j) \in A \quad (44)$$

$$\mu \geq 0 \quad (45)$$

$$\lambda_k \text{ irrestrito} \quad \forall k \in V. \quad (46)$$

As restrições (40) a (44) impõem as condições que fazem o caminho identificado pelas variáveis  $y$  pertencer a  $\mathcal{P}(\beta)$ . Sejam  $y'$  uma solução viável para (R-RSP-H) e  $p' \in \mathcal{P}(\beta)$  o caminho identificado por  $y'$ . Note que  $p'$  também é um caminho viável para o problema R-RSP. Pela Proposição 2, o custo de robustez  $\beta$ -heurístico de  $p'$  fornece um limite superior para R-RSP, que pode ser aperfeiçoado pelo cálculo do custo de robustez  $\beta$ -restrito de  $p'$ . Os passos da heurística são apresentados no Algoritmo 1.

**Passo I.** Resolve o problema (R-RSP-H) com um *solver* de otimização, estabelecendo um limite de tempo  $\delta$ ;  
**Passo II.** Recupera a melhor solução viável  $y^\delta$  obtida pelo *solver*;  
**Passo III.** Calcula o custo de robustez  $\beta$ -restrito do caminho identificado por  $y^\delta$ .

#### Algoritmo 1: Heurística para R-RSP

O custo computacional da heurística é determinado pela constante  $\delta$ , que limita o tempo de execução do *solver* na resolução de (R-RSP-H). Apesar de o cálculo do custo de robustez  $\beta$ -restrito da solução  $y^\delta$  encontrada pela heurística depender da resolução de uma instância do problema do caminho mínimo restrito, que é  $\mathcal{NP}$ -difícil, esse passo não é necessário, uma vez que  $y^\delta$  já está disponível no **Passo II** da heurística.

## 5. Experimentos Computacionais

A implementação, em linguagem C++, utilizou o *solver* ILOG CPLEX 12.5<sup>1</sup> em duas configurações: a primeira, chamada de LPH<sub>1</sub>, usa os parâmetros padrões; a segunda, chamada de LPH<sub>2</sub>, desabilita o pré-processamento e atribui *feasibility* ao parâmetro

<sup>1</sup><http://www.ilog.com/products/cplex/>

*emphasis*. Os experimentos computacionais foram realizados numa máquina Intel Core i5 1.70GHz/5.7 GiB, no Linux, estabelecendo um limite de tempo  $\delta = 120s$ . As instâncias testadas são inspiradas nas existentes na literatura de RSP: instâncias de Karaşan (Karaşan et al., 2001), e instâncias de Coco (Coco et al., 2012).

As instâncias apresentadas em (Karaşan et al., 2001) são compostas por grafos em camadas (Sugiyama et al., 1981) e acíclicos (Bondy and Murty, 1976), tendo sido usadas em diversos experimentos computacionais na literatura de RSP (Karaşan et al., 2001; Montemanni and Gambardella, 2004; R. Montemanni and Donati, 2004; Montemanni and Gambardella, 2005). Cada uma das  $\kappa$  camadas desses grafos apresenta um mesmo número  $\omega$  de vértices. De cada vértice de uma camada  $b \in \{1, \dots, \kappa - 1\}$  parte um arco para cada vértice da camada adjacente  $b+1$ . Além disso, existe um arco do vértice de origem até cada vértice da primeira camada e um arco de cada vértice da camada  $\kappa$  até o vértice de destino. A nomenclatura das instâncias segue o formato  $K-v-\Phi_{max}-\Delta-\varpi-\omega$ , com  $0 < \Delta < 1$ , onde  $v$  é a quantidade de vértices do grafo,  $\Phi_{max}$  é um valor inteiro e  $\varpi$  distingue instâncias diferentes geradas com a mesma configuração de parâmetros. O intervalo de custo  $[l_{ij}, u_{ij}]$  associado a cada arco  $(i, j) \in A$  é obtido gerando aleatoriamente um inteiro  $\Phi_{ij}$  em  $[1, \Phi_{max}]$  e selecionando, também aleatoriamente,  $l_{ij}$  em  $[(1 - \Delta) \cdot \Phi_{ij}, (1 + \Delta) \cdot \Phi_{ij}]$  e  $u_{ij}$  em  $[l_{ij}, (1 + \Delta) \cdot \Phi_{ij}]$ . Foram considerados valores fixos  $\Phi_{max} = 200$  e  $\Delta = 0.9$ .

As instâncias apresentadas em (Coco et al., 2012) são grafos em *grid* gerados a partir de matrizes  $n \times m$ , onde  $n$  é o número de linhas e  $m$  é o número de colunas. Cada célula da matriz representa um vértice do grafo, e existem dois arcos bidirecionais entre cada par de vértices cujas células correspondentes da matriz são adjacentes. O nome das instâncias segue o formato  $G-n \times m - \varpi$ , onde  $\varpi$  diferencia as instâncias geradas com os mesmos valores de  $n$  e  $m$ . Os intervalos de custo dos arcos foram gerados da mesma maneira que nas instâncias de Karaşan, com os mesmos valores estabelecidos para  $\theta_{max}$  e  $\Delta$ . Exemplos de ambos os tipos de instâncias são mostrados na Figura 2.

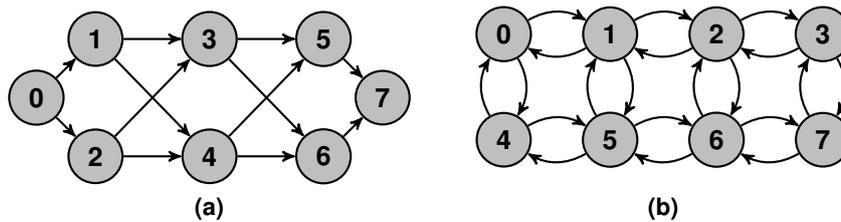


Figura 2. Exemplos de instâncias utilizadas nos experimentos computacionais. (a) Instância de Karaşan (Karaşan et al., 2001), com 3 camadas de largura 2. (b) Instância de Coco (Coco et al., 2012), com *grid*  $2 \times 4$ . Em ambos os casos, os vértices 0 e 7 são, respectivamente, a origem e o destino.

Em ambos os tipos de instâncias, o valor de comprimento  $d_{ij}$  associado a um arco  $(i, j) \in A$  foi selecionado aleatoriamente no intervalo  $(0, 10]$ . Seja  $p^*$  um caminho de menor comprimento, isto é,  $p^* = \arg \min_{p' \in \mathcal{P}} D_{p'}$ , o limite de comprimento  $\beta$  de cada instância foi estabelecido como  $\beta = 1.1 \cdot D_{p^*}$ , ou seja, é dada uma tolerância de 10% com relação ao caminho de menor comprimento.

O certificado de otimalidade de cada instância foi obtido a partir da adaptação de um esquema de decomposição de Benders amplamente utilizado para resolver problemas de otimização robusta (Montemanni and Gambardella, 2005; Montemanni, 2006; Montemanni et al., 2007; Pereira and Averbakh, 2011, 2013). Devido à restrição de espaço, os

detalhes desse método foram omitidos neste trabalho.

Instance	LPH <sub>1</sub>				LPH <sub>2</sub>			
	$t_1$ (s)	$GAP_1$ (%)	$t_2$ (s)	$GAP_2$ (%)	$t_1$ (s)	$GAP_1$ (%)	$t_2$ (s)	$GAP_2$ (%)
g_6x60_a	<b>0.80</b>	1.81	<b>0.90</b>	<b>0.0</b>	0.99	1.81	1.09	<b>0.0</b>
g_7x70_a	2.09	3.45	2.27	<b>0.0</b>	<b>1.77</b>	3.45	<b>1.95</b>	<b>0.0</b>
g_8x80_a	3.58	0.66	3.81	<b>0.0</b>	<b>3.13</b>	0.66	<b>3.37</b>	<b>0.0</b>
g_9x90_a	<b>5.93</b>	6.59	<b>6.76</b>	<b>0.0</b>	6.01	6.59	6.84	<b>0.0</b>
g_10x100_a	<b>21.49</b>	1.62	<b>22.20</b>	<b>0.0</b>	25.85	1.62	26.58	<b>0.0</b>
g_6x60_b	0.46	14.79	0.59	<b>0.0</b>	<b>0.41</b>	14.79	<b>0.53</b>	<b>0.0</b>
g_7x70_b	1.63	5.97	1.90	0.72	<b>1.37</b>	5.97	<b>1.64</b>	0.72
g_8x80_b	3.80	3.34	4.10	1.63	<b>3.28</b>	3.34	<b>3.59</b>	1.63
g_9x90_b	7.12	3.85	7.62	1.48	<b>6.92</b>	3.85	<b>7.41</b>	1.48
g_10x100_b	<b>13.53</b>	1.44	<b>14.44</b>	0.72	21.85	1.44	22.50	0.72
K-1000-200-0.9-a-5	<b>20.64</b>	0.0	<b>21.54</b>	<b>0.0</b>	22.87	<b>0.0</b>	23.63	<b>0.0</b>
K-1000-200-0.9-a-10	18.19	0.91	19.37	<b>0.0</b>	<b>15.72</b>	0.91	<b>16.94</b>	<b>0.0</b>
K-1000-200-0.9-a-25	<b>26.37</b>	1.33	<b>28.28</b>	<b>0.0</b>	29.65	1.33	31.63	<b>0.0</b>
K-1000-200-0.9-a-50	<b>22.64</b>	3.01	<b>26.80</b>	<b>0.0</b>	24.48	3.01	28.53	<b>0.0</b>
K-1000-200-0.9-a-100	<b>54.66</b>	0.97	<b>60.57</b>	<b>0.0</b>	120.00	0.97	128.33	<b>0.0</b>
K-1000-200-0.9-b-25	<b>28.85</b>	1.95	<b>30.62</b>	<b>0.0</b>	19.16	1.95	21.11	<b>0.0</b>
K-1000-200-0.9-b-50	34.50	1.83	38.07	<b>0.0</b>	<b>31.35</b>	1.83	<b>35.13</b>	<b>0.0</b>
K-1000-200-0.9-b-100	<b>39.16</b>	1.31	<b>44.73</b>	<b>0.0</b>	51.50	1.31	57.46	<b>0.0</b>

**Tabela 1. Resultados computacionais para instâncias de Karaşan (Karaşan et al., 2001) e instâncias de Coco (Coco et al., 2012).**

A Tabela 1 detalha os resultados obtidos para as instâncias testadas utilizando as configurações LPH<sub>1</sub> e LPH<sub>2</sub>. Os valores  $t_1$  e  $t_2$  representam, respectivamente, o tempo gasto sem e com a etapa de cálculo do custo de robustez  $\beta$ -restrito da solução encontrada. Os valores  $GAP_1$  e  $GAP_2$  são dados por  $GAP_i = 100 \cdot \frac{UB_i - R^*}{UB_i}$ ,  $i \in \{1, 2\}$ , onde  $R^*$  é o custo de robustez  $\beta$ -restrito ótimo da instância, e  $UB_1$  e  $UB_2$  são, respectivamente, os custos de robustez  $\beta$ -heurístico e  $\beta$ -restrito da solução encontrada.

Quanto à qualidade das soluções, ambas as configurações obtiveram os mesmos resultados, embora, em geral, LPH<sub>1</sub> tenha obtido menor tempo de execução nas instâncias de Karaşan (Karaşan et al., 2001), e LPH<sub>2</sub> nas de Coco (Coco et al., 2012). A heurística foi capaz de encontrar a solução ótima na maioria das instâncias testadas, em especial nas de Karaşan (Karaşan et al., 2001). Comparando os valores de  $GAP_1$  e  $GAP_2$ , verificamos que o cálculo do custo robusto  $\beta$ -restrito da solução encontrada pela heurística refina bastante o limite superior. Estamos atualmente investigando instâncias mais difíceis para o problema, que serão abordadas em trabalhos futuros.

## 6. Agradecimentos

Este trabalho foi parcialmente financiado pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), pela Fundação de Amparo à Pesquisa do Estado de Minas Gerais, Brasil (FAPEMIG) e pela Coordenação de Aperfeiçoamento de Pessoal de Nível Superior, Brasil (CAPES).

## Referências

- Aneja, Y. P., Aggarwal, V., and Nair, K. P. K. (1983). Shortest chain subject to side constraints. *Networks*, 13:295–302.
- Beasley, J. and Christofides, N. (1989). An algorithm for the resource constrained shortest path problem. *Networks*, 19:379–394.
- 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.** (2012). A biased random-key genetic algorithm for the robust shortest path problem. In *Global Optimization Workshop. GOW 2012*, Natal, Brazil.
- Coco, A. A., Júnior, J. C. A., Noronha, T. F., and Santos, A. C.** (2014). An integer linear programming formulation and heuristics for the minmax relative regret robust shortest path problem. *Journal of Global Optimization*, pages 1–23.
- Dias, L. C. and Clímaco, J. N.** (2000). Shortest path problems with partial information: Models and algorithms for detecting dominance. *European Journal Of Operational Research*, 121(1):16–31.
- 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.
- 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.
- 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.
- Kouvelis, P. and Yu, G.** (1997). *Robust Discrete Optimization and Its Applications*. Kluwer Academic Publishers, Boston.
- 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. 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 and 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.
- 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.
- R. Montemanni, L. M. G. 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.
- Shimbel, A.** (1955). Structure in communication nets. In *Proceedings of the Symposium on Information Networks (New York, 1954)*, pages 199–203.
- 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.
- Zieliński, P.** (2004). The computational complexity of the relative robust shortest path problem with interval data. *European Journal of Operational Research*, 158:570–576.