

## FORMULAÇÃO E HEURÍSTICA PARA O PROBLEMA DA ÁRVORE DE CAMINHOS MAIS CURTOS ROBUSTA

**Iago A. de Carvalho**

Departamento de Ciência da Computação - Universidade Federal de Minas Gerais  
Belo Horizonte - MG - Brasil  
iagoac@ufmg.br

**Marcelo A. Gomes**

Departamento de Ciência da Computação - Universidade Federal de Minas Gerais  
Belo Horizonte - MG - Brasil  
marcelog@dcc.ufmg.br

**Thiago F. Noronha**

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

**Christophe Duhamel**

LIMOS, ISIMA - Université Blaise Pascal  
Clermont-Ferrand - França  
christophe.duhamel@isima.fr

**Luiz F. M. Vieira**

Departamento de Ciência da Computação - Universidade Federal de Minas Gerais  
Belo Horizonte - MG - Brasil  
lfvieira@dcc.ufmg.br

### RESUMO

*IPv6 Low Wireless Personal Area Networks (6LoWPAN)* é a mais promissora tecnologia para a implementação da Internet das Coisas (IdC). Para que a IdC torne-se uma realidade, protocolos eficientes precisam ser resilientes a variações na qualidade da transmissão, devido a constantes mudanças nos enlaces. O mais promissor destes protocolos é o *IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL)*. Nós estendemos este protocolo considerando a incerteza na qualidade dos enlaces. O problema de roteamento do RPL é modelado como um problema de otimização robusta derivado do Problema da Árvore de Caminhos Mais Curtos, denominado Árvore de Caminhos Mais Curtos Robusta. Nós propomos uma heurística para este problema, além de uma formulação por Programação Linear Inteira Mista produziu soluções ótimas para todas as instâncias com até 100 vértices em menos de 8 segundos. A heurística proposta encontrou um *gap* menor que 8%, em média, para instâncias com até 200 vértices.

**PALAVRAS CHAVE.** Otimização robusta. Internet das Coisas. RPL.

**Área Principal:** Otimização combinatória, Programação matemática, Outras aplicações em PO

**ABSTRACT**

IPv6 Low Wireless Personal Area Networks (6LoWPAN) is the most promising technology for implementing the Internet of Things (IoT). In order to IoT become a reality, routing protocols have to be resilient to variations in transmission quality, due to constant changes in the network surrounding. The most promising of these protocols is the IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL). We extend the RPL protocol considering the uncertainty in the link quality. RPL routing problem is modelled as a robust optimization variant of the Shortest Path Tree Problem, called Robust Shortest Path Tree. We propose an heuristic for this problem, and a Mixed Integer Linear Programming for this problem that gives the optimal solution for all instances with up to 100 nodes in less than 8 seconds. The proposed heuristic algorithm had an average gap smaller than 8% on networks with 200 nodes.

**KEYWORDS.** Robust optimization. Internet of Things. RPL.

**Main area:** Combinatorial optimization, Mathematical programming, Other applications in OR



## 1. Introdução

Atualmente, há um crescente aumento no número de dispositivos conectados a Internet, como computadores, sensores, *smartphones*, eletrodomésticos, dentre outros. Este crescente conjunto introduz um novo paradigma no cenário da moderna comunicação sem fio. Estes dispositivos comunicam-se entre si e colaboram com seus vizinhos para alcançar objetivos comuns, formando a Internet das Coisas (IdC) [Giusto et al.(2010)]. Para que a IdC se torne uma realidade, muitos desafios ainda precisam ser superados, como o design de protocolos eficientes que resultam em redes seguras e de baixo consumo energético. Além disto, estes protocolos precisam ser resilientes a grandes variações na qualidade da transmissão, devido a constantes mudanças no enlace da rede.

A principal solução para este desafio são as IPv6 Low Wireless Personal Area Networks (6LoWPAN) [Shelby and Bormann(2011)], redes caracterizadas por baixos recursos, tanto em termos de capacidade de computação quanto em capacidade energética. Cada vértice em uma rede 6LoWPAN representa um dispositivo na IdC. Estes vértices são interconectados por enlaces potencialmente com baixa qualidade de comunicação e taxas de perdas elevadas. Um erro em um único enlace pode afetar muitos outros, fazendo com que a rede se torne ineficiente ou até mesmo desconectada [Winter(2012)].

Vários protocolos de roteamento para 6LoWPAN foram desenvolvidos na tentativa de superar estas dificuldades. O mais promissor destes protocolos é o *IPv6 Routing Protocol for Low-Power and Lossy Networks* (RPL) [Winter(2012)]. Primeiramente, o RPL constrói um Grafo Acíclico Direto Orientado (GADO) para um vértice central da rede *s*, chamado *sink*, a partir de todos os outros vértices na rede que servem a mesma aplicação que o *sink*. Este GADO é construído levando em consideração o alcance de transmissão dos sensores, bem como a distância entre os sensores na rede. Cada aplicativo de rede tem seu próprio GADO e o roteamento de mensagens para cada aplicação é feita de forma independente. Toda a comunicação entre os sensores é realizada através deste GADO e retransmitida pelo *sink*. Pode existir um número exponencial de rotas entre cada vértice e o *sink* e a eficiência da rede depende da qualidade das rotas escolhidas.

O RPL utiliza uma estimativa da qualidade dos enlaces para determinar as rotas entre o *sink* e os outros vértices da rede. É permitido ao administrador da rede escolher qualquer métrica para estimar a qualidade dos enlaces, como a taxa de transferência, a taxa de perda de pacotes, dentre outras. Contudo, o protocolo sempre faz o roteamento do pacote atual de acordo com observações passadas da métrica escolhida, cujos valores podem ter sido modificados desde a última vez que o RPL atualizou a métrica determinada. Como redes 6LoWPAN são caracterizadas por grandes variações na qualidade do enlace, qualquer estimativa errada da qualidade do enlace usada pelo RPL pode resultar em uma má performance de roteamento.

Neste trabalho, o problema de roteamento do protocolo RPL é estendido ao considerarmos a incerteza associada na estimativa da qualidade dos enlaces. O problema de roteamento do RPL é modelado como um problema de otimização robusta derivado do Problema da Árvore de Caminhos Mais Curtos (ACMC) [Salazar-Neumann(2007)], o Problema da Árvore de Caminhos Mais Curtos Robusta (ACMC-R). Então, uma heurística é proposta para este problema, que pode ser implementada no protocolo RPL. Além disto, uma formulação por Programação Linear Inteira Mista (PLIM) para a ACMC-R é proposta. A formulação PLIM foi resolvida a partir da aplicação de um algoritmo baseado no método *branch-and-bound* de forma a avaliar a qualidade de nossa heurística. Experimentos computacionais realizados mostram que o pior caso da heurística proposta é, no máximo, duas vezes o valor da solução ótima, porém sua performance média é muito próxima a solução ótima.

O restante deste artigo está organizado da seguinte forma. Na Seção 2, o problema da ACMC-R é definido. Problemas relacionados são discutidos na Seção 3. A formulação PLIM é apresentada na Seção 4. Logo após, um algoritmo heurístico é proposto na Seção 5. Em sequência, resultados computacionais são reportados na Seção 6. Finalmente, as considerações finais são reportadas na última Seção.

## 2. Definição do problema

Dado um digrafo conectado  $G = (V, A)$  com um conjunto  $V$  de vértices e um conjunto  $A$  de arcos. Cada arco  $(i, j) \in A$  é associado a um custo  $c_{ij} \in \mathbb{R}$ . Além disso, sejam  $n = |V|$  e  $m = |A|$  respectivamente o número total de vértices e o número total de arcos de  $G$ . O problema do Caminho Mais Curto (CMC) consiste em encontrar o caminho mais curto entre um vértice de origem  $s \in V$  para um vértice de destino  $t \in V$  sendo o custo total do caminho o mínimo possível. Uma solução existe se e somente se não ocorre a formação de nenhum ciclo de peso negativo no caminho entre  $s$  e  $t$ . Existem algoritmos de tempo polinomial que resolvem o CMC, como o algoritmo de Dijkstra [Dijkstra(1959)] e o algoritmo de Bellman-Ford [Bellman(1956)]. Uma extensão deste problema é o problema da Árvore de Caminhos Mais Curtos (ACMC) [Wu and Chao(2004)], que consiste em encontrar uma árvore que contenha o caminho mais curto de  $s$  para todos os outros vértices em  $V$ . Este problema também pode ser resolvido pelo algoritmo de Dijkstra [Cormen et al.(2009)]. O problema de roteamento do RPL definido em [Vasseur et al.(2011), Winter(2012)] consiste em uma versão distribuída do algoritmo de Dijkstra para a ACMC, em que os vértices em  $V$  são associados a dispositivos na IdC, os arcos em  $A$  são associados aos enlaces e o custo  $c_{ij}$  corresponde ao valor da métrica utilizada para estimar a qualidade do enlace.

Os métodos mais utilizados para resolver problemas de otimização sob incerteza são a programação estocástica [Spall(2003)] e a otimização robusta [Ben-Tal and Nemirovski(2002), Kouvelis and Yu(1997)]. A programação estocástica é aplicada principalmente quando uma lei de probabilidade associada a incerteza é conhecida antecipadamente. Um ponto falho desta abordagem consiste no fato de que pode ser difícil definir a lei de probabilidade associada a incerteza dos dados, ou algum erro pode ocorrer na estimação dos parâmetros. Pode-se referir a [Nie and Wu(2009), Bertsekas and Tsitsiklis(1991)] para trabalhos dedicados ao problema do caminho mais curto estocástico, que também estende o CMC, uma vez que ele minimiza o custo total esperado.

Neste trabalho, foi focada a otimização robusta. Ela é uma alternativa a programação estocástica onde a variabilidade dos dados é representada por valores determinísticos. Neste trabalho, foram focados modelos de otimização robusta onde os dados incertos podem ser modelados em um intervalo de valores possíveis. Nos referimos ao livro [Kouvelis and Yu(1997)] para outros modelos de otimização robusta.

O problema do Caminho Mais Curto Robusto (CMC-R) é uma generalização do CMC, onde o custo  $c_{ij}$  de cada arco  $(i, j) \in A$  é definido em um intervalo  $[l_{ij}, u_{ij}]$ , com  $l_{ij}, u_{ij} \in \mathbb{Z}$ , sendo  $u_{ij} \geq l_{ij} \geq 0$ , para todo  $(i, j) \in A$  [Karaşan et al.(2001)]. Existem diferentes versões do CMC-R com intervalo de dados na literatura. Elas diferem uma da outra pelo critério de otimização utilizado [Aissi et al.(2009), Averbakh(2005), Kasperski et al.(2005), Montemanni and Gambardella(2005)a, Candia-Véjar et al.(2011), Montemanni and Gambardella(2005)b, Montemanni et al.(2004)].

A versão mais estudada do CMC-R usa o critério *minmax regret* [Kouvelis and Yu(1997), Aissi et al.(2009), Candia-Véjar et al.(2011)] e é chamado *minmax regret CMC-R* [Coco et al.(2014), Kouvelis and Yu(1997), Montemanni et al.(2004), Montemanni and Gambardella(2005)b]. Seja  $P \subseteq A$  um caminho com origem em  $s$  e destino  $t$  em  $G$ . O *regret* de  $P$  no cenário  $r$  (também referido como o *desvio robusto* de  $P$  em  $r$ ) é definido como a diferença entre o custo de  $P$  em  $r$  e o custo do caminho mais curto  $S^r$  de  $s$  a  $t$  em  $r$ . Em outras palavras, o *desvio robusto* de  $P$  em  $r$  é o *regret* de se usar  $P$  ao invés de  $S^r$  caso o cenário  $r$  ocorra. O *custo robusto* de  $P$  é definido como o maior desvio robusto de  $P$  dentre todos os cenários. O *minmax regret CMC-R* consiste em encontrar o caminho  $P^*$  de  $s$  a  $t$  com o menor custo robusto. Este problema é demonstrado ser NP-Difícil mesmo em digrafos acíclicos [Kouvelis and Yu(1997)].

O problema da Árvore de Caminhos Mais Curtos Robusta (ACMC-R) é uma generalização do CMC-R que foi proposto em [Salazar-Neumann(2007)]. Seja  $G = (V, A)$  um digrafo conectado, onde  $V$  é um conjunto de vértices e  $A$  é um conjunto de arcos. Cada arco  $(i, j) \in A$  é associado a um intervalo de custos  $[l_{ij}, u_{ij}]$ , com  $l_{ij}, u_{ij} \in \mathbb{R}_+$  e  $u_{ij} \geq l_{ij} \geq 0$ , para todo  $(i, j) \in A$ . Seja

também  $s \in V$  o vértice raiz e  $V' = V \setminus \{s\}$ . Em nossa modelagem para o problema de roteamento do RPL, os vértices em  $V$  são associados a sensores, os arcos em  $A$  são associados aos enlaces,  $l_{ij}$  corresponde ao menor valor observado pela métrica de qualidade do enlace em um determinado intervalo de tempo, e  $u_{ij}$  corresponde ao maior valor observado por esta mesma métrica, durante o mesmo intervalo.

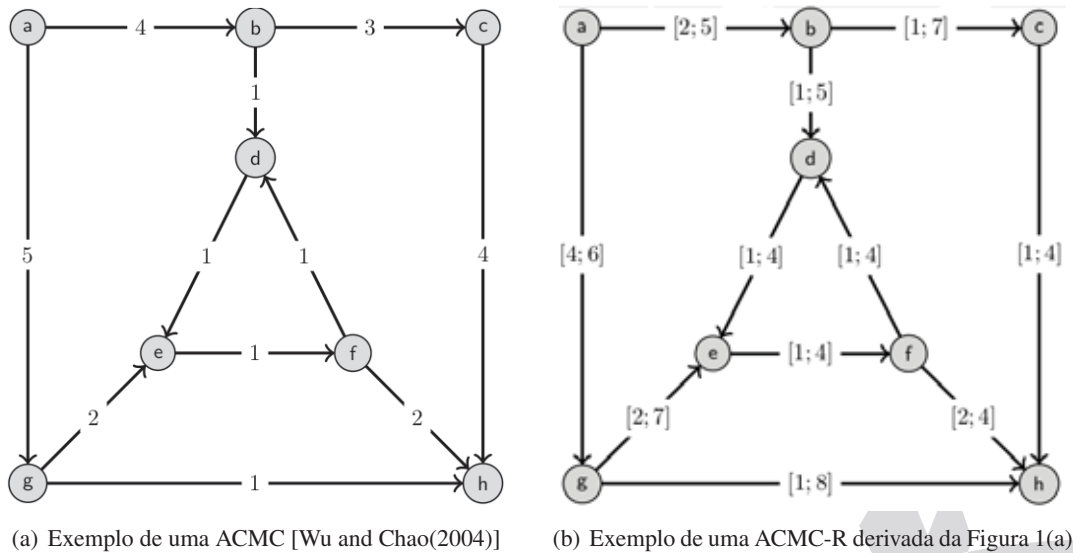


Figura 1: Exemplos de Árvore de Caminhos mais Curtos e Árvore de Caminhos Mais Curtos Robusta

**Definição** Um cenário  $r$  é uma realização do custo dos arcos  $c_{ij}^r \in [l_{ij}, u_{ij}]$  para todo arco  $(ij) \in A$ .

Seja  $\mathcal{T}$  o conjunto de todas as árvores geradoras com raiz em  $s$  de  $G$ . Seja também  $p_i^t$  um caminho do vértice  $s$  para o vértice  $i \in V'$  induzido pela árvore  $t \in \mathcal{T}$ . Além disto, seja  $c^r(p_i^t) = \sum_{(i,j) \in A[p_i^t]} c_{ij}^r$  o custo de  $p_i^t$  no cenário  $r$ , e  $c^r(p_i^*)$  o custo do caminho mais curto  $p_i^*$  de  $s$  a  $i$  no cenário  $r$ .

**Definição** O *desvio robusto* de um caminho  $p_i^t$  induzido por  $t \in \mathcal{T}$  no cenário  $r$  (também referido como o *regret* de  $p_i^t \in r$ ) é definido como  $d^r(p_i^t) = c^r(p_i^t) - c^r(p_i^*)$ , isto é, a diferença entre o custo de  $p_i^t \in r$  e o melhor caminho possível de  $s$  para  $i$  no cenário  $r$ .

Seja  $A[p_i^t] \in A$  o conjunto de arcos que compõe o caminho  $p_i^t$ , e  $\mathcal{R}$  o conjunto de todos os possíveis cenários em  $G$ .

**Lema 2.1** O desvio robusto de  $p_i^t$  é o máximo, dentre todos os cenários em  $\mathcal{R}$ , no cenário  $r(p_i^t) \in \mathcal{R}$ , de tal forma que  $c_{ij}^{r(p_i^t)} = u_{ij}$ , para todo  $(i, j) \in A[p_i^t]$ , e  $c_{ij}^{r(p_i^t)} = l_{ij}$ , para todo  $(i, j) \in A \setminus A[p_i^t]$ .

**Prova** Disponível em [Kouvelis and Yu(1997)].

Quanto menor o desvio robusto de  $p_i^t$ , melhor é o caminho para enviar mensagens entre  $s$  e  $i$ . Além disto, quanto menor é o valor de  $d^r(p_i^t)(p_i^t)$ , mais robusto é o caminho em relação a variações na métrica de qualidade do enlace.

**Definição** O *custo robusto* de  $t \in \mathcal{T}$  é definido como  $R_t = \sum_{i \in V'} d^r(p_i^t)(p_i^t)$ , ou seja, a soma do máximo desvio robusto de todos os caminhos entre  $s$  e todo vértice  $i \in V'$ .

Quanto menor é o custo robusto de uma árvore de comunicação em uma 6LoWPAN, mais eficiente e mais confiável é a rede.

**Definição** Uma árvore  $t^* \in \mathcal{T}$  é dita ser *robusta* se ela possui o menor custo robusto dentre todas as árvores em  $\mathcal{T}$ .

Portanto, uma ACMC-R pode ser definida como uma árvore geradora de  $G$  com raiz em  $s$ , isto é,  $t^* = \arg \min_{t \in \mathcal{T}} R_t$ .

### 3. Trabalhos relacionados

O *survey* [Aissi et al.(2009)] é dedicado a resultados sobre a complexidade computacional de vários problemas de otimização robusta, enquanto o *survey* [Gabrel et al.(2013)] é dedicado ao CMC-R, tanto com dados intervalares como com cenários discretos. Uma visão global sobre problemas de otimizações robusta e suas aplicações, bem como modelos matemáticos para diferentes versões do CMC-R pode ser encontrado em [Kouvelis and Yu(1997)]. Um estudo sobre algoritmos exatos, aproximativos e heurísticos para uma série de problemas de otimização, incluindo o CMC-R, é encontrado em [Candia-Véjar et al.(2011)].

O *minmax regret* CMC-R foi demonstrado ser NP-Difícil em [Kouvelis and Yu(1997)]. Uma formulação PLIM para este problema foi introduzida em [Karaşan et al.(2001)]. Um algoritmo de *branch-and-bound* baseado na relaxação combinatória das restrições robustas foi desenvolvido em [Montemanni et al.(2004)]. Em [Montemanni and Gambardella(2005)b], este trabalho foi estendido pelo uso de um algoritmo de decomposição de Benders baseado na mesma relaxação. Ambos os algoritmos resolveram instâncias em grafos aleatórios com até 4.000 vértices e instâncias reais de até 2.500 vértices. Uma série de heurísticas para o CMC-R foram comparadas em [Coco et al.(2014)], sendo que um algoritmo heurístico 2-aproximativo proposto em [Kasperski and Zieliński(2006)] obteve um *gap* relativo de 76%, em média, para instâncias com até 1500 vertices.

Os trabalhos [Averbakh and Lebedev(2004), Escoffier et al.(2008)] investigaram instâncias que podem ser solucionadas em tempo polinomial ou pseudo-polinomial em problemas de otimização robusta clássicos. Um algoritmo pseudo-polinomial é proposto em [Kasperski and Zieliński(2006)] para o *minmax regret* CMC-R. Ele trabalha com multidigraphos série-paralelo e tem uma complexidade computacional igual a  $O(m \cdot |S^u|^2)$ , sendo  $|S^u|$  o número de arcos no caminho mais curto de  $s$  para  $t$  no cenário onde o custo dos arcos são setados para  $u_{ij}$ . Algoritmos 2-aproximativos para uma larga classe de problemas de otimização robusta *minmax regret* são apresentados em [Conde(2012)]. Para o *minmax regret* CMC-R, um algoritmo 2-aproximativo com complexidade igual a  $O(m + n \log n)$  é proposto em [Kasperski and Zieliński(2006)]. Uma versão aperfeiçoada deste algoritmo é desenvolvida em [Kasperski and Zieliński(2007)] para multidigraphos série-paralelos com fator de aproximação igual a  $(1 + \epsilon)$  e complexidade de  $O(q(n, n/\epsilon))$ , onde  $q$  é uma função polinomial bivariada e  $\epsilon$  é um valor no intervalo  $[0, 1]$ .

Nos trabalhos [Karaşan et al.(2001), Catanzaro et al.(2011)] são propostas técnicas de pré-processamento para o *minmax regret* CMC-R. A ideia geral é remover arcos dominados, chamados *weak-arcs*, ou seja, arcos que não pertencem a solução ótima. Um algoritmo para eliminar estes *weak-arcs* é introduzido em [Karaşan et al.(2001)] para tipos de grafos particulares, como por exemplo grafos acíclicos, planares e em camadas. Em [Catanzaro et al.(2011)], o autor estende a técnica de pré-processamento proposta em [Karaşan et al.(2001)] eliminando vértices e aplicando outras estratégias para eliminar os *weak-arcs*. O procedimento de eliminação de vértices tem ordem de complexidade igual a  $O(n^3)$ , e elimina vértices que não pertencem a nenhum caminho mais curto entre o vértice de origem e o vértice de destino. Além disso, os autores apresentam estratégias usando árvores geradoras mínimas para detectar *weak-arcs* com uma complexidade computacional de  $O(m \cdot n^2)$ .

O problema da APMC-R foi provado ser NP-Difícil no trabalho [Salazar-Neumann(2007)]. Técnicas de pré-processamento e uma formulação PLIM foram propostos neste mesmo trabalho, realizando uma generalização de técnicas de pré-processamento propostas em [Karaşan et al.(2001), Catanzaro et al.(2011)] para o CMC-R.

#### 4. Formulação PLIM para a APMC-R

A formulação PLIM é definida com variáveis de decisão  $z_{ij} = 1$  se o arco  $(i, j) \in A$  pertence a árvore robusta, e  $z_{ij} = 0$  caso contrário. Além disso, variáveis auxiliares  $y_{ij}^k \in \{0, 1\}$  mantêm o caminho entre  $s$  e  $k$  induzido pela árvore definida pelas variáveis  $z_{ij}$ . Mais além, variáveis  $x_i^k \geq 0$  mantêm o custo do caminho mais curto de  $s$  para  $i \in V'$  no cenário induzido pelo caminho de  $s$  para  $k$  (definido pelas variáveis  $y_{ij}^k$ ). Portanto, o caminho mais curto de  $s$  para  $k$  neste cenário é mantido nas variáveis  $x_k^k$ . A correspondente formulação PLIM é definida pelas equações (1) a (9).

$$\min \sum_{k \in V'} \left[ \sum_{(i,j) \in A} (u_{ij} y_{ij}^k) - x_k^k \right] \quad (1)$$

$$\sum_{(j,l) \in A} y_{jl}^k - \sum_{(i,j) \in A} y_{ij}^k = \begin{cases} 1, & \text{if } j = s \\ -1, & \text{if } j = k \\ 0, & \text{caso contrário} \end{cases}, \quad \forall j \in V, k \in V' \quad (2)$$

$$x_j^k \leq x_i^k + l_{ij} + (u_{ij} - l_{ij}) y_{ij}^k, \quad \forall (i, j) \in A, k \in V' \quad (3)$$

$$y_{ij}^k \leq z_{ij}, \quad \forall (i, j) \in A, k \in V' \quad (4)$$

$$\sum_{(i,j) \in A} z_{ij} = n - 1 \quad (5)$$

$$x_s^k = 0, \quad k \in V' \quad (6)$$

$$x_i^k \geq 0, \quad \forall i \in V', k \in V' \quad (7)$$

$$y_{ij}^k \in \{0, 1\}, \quad \forall (i, j) \in A, k \in V' \quad (8)$$

$$z_{ij} \in \{0, 1\}, \quad \forall (i, j) \in A \quad (9)$$

A função objetivo (1) minimiza a soma do desvio robusto máximo para todo  $k \in V'$ . As restrições (2) são as clássicas restrições de conservação de fluxo e asseguram a conectividade de cada caminho de  $s$  para  $k$ . As desigualdades (3) constroem um *link* entre as variáveis  $x$  e  $y$  e impõe o valor correto para  $x$ . O custo do caminho entre o vértice  $s$  e um vértice  $j$  no cenário  $k$  é calculado como o custo do caminho entre  $s$  e um vértice  $i$  adjacente a  $j$  e pertencente a seu fecho transitivo inverso acrescido de  $u_{ij}$  caso  $y_{ij}^k = 1$  ou  $l_{ij}$  caso  $y_{ij}^k = 0$ . As desigualdades (4) constroem um *link* entre as variáveis  $y$  e  $z$  e estabelece que o caminho do vértice raiz  $s$  para todos os vértices  $k \in V'$  estejam na árvore. As restrições (5), juntamente as restrições (2) e (4), determinam que as variáveis  $z$  induzam uma árvore. As restrições (6) quebram a simetria nas variáveis  $x$ , atribuindo para zero o valor do caminho mais curto de  $s$  para ele mesmo, para todo  $k \in V'$ . O domínio das variáveis  $x$ ,  $y$  e  $z$  são definidos por (7), (8) e (9), respectivamente. Nenhum resultado experimental foi reportado em [Salazar-Neumann(2007)] para as técnicas de pré-processamento ou para esta formulação proposta. Na Sessão 6 os resultados para esta formulação são reportados e utilizados para assegurar a qualidade de nossa heurística. A formulação utiliza  $2mn + m$  variáveis e  $m^2 + 3mn$  restrições.

## 5. Heurística Average Median para a ACMC-R

Heurísticas desenvolvidas para outros problemas de otimização robusta em redes podem ser estendidas para a ACMC-R. Uma heurística proposta em [Kasperski and Zieliński(2006)] para o CMC-R pode ser aplicada a ACMC-R. Esta heurística, denominada Average Median (AM), fixa o custo de cada arco em seu respectivo valor médio (isto é,  $(u_{ij} + l_{ij})/2$ ) e então retorna o caminho mais curto neste cenário.

Uma heurística para a ACMC-R é proposta, estendendo a heurística AM proposta em [Kasperski and Zieliński(2006)]. O custo de todos os arcos são fixados em seus respectivos valores médios e então a árvore de caminhos mais curtos neste cenário é retornada. Esta heurística pode ser implementada de forma distribuída, pois trata-se de um algoritmo de caminho mais curto em grafos [Chandy and Misra(1982), Humblet(1991), Träff(1995)].

## 6. Resultados computacionais

Experimentos computacionais foram realizados em um CPU Intel Xeon E5645 com clock de 2.4 GHz e 32 GB de RAM, executando o sistema operacional Linux. A versão 12.5 do ILOG CPLEX foi utilizada com seus parâmetros padrões para resolver a formulação PLIM definida entre as equações (1) e (9). A heurística AM foi implementada em C++ e compilada com o GNU GCC versão 4.7.3. O tempo de execução do CPLEX foi limitado em 7200 segundos.

As instâncias são relativas aos grafos de Karaşan [Karaşan et al.(2001)] previamente utilizados em [Coco et al.(2014)]. Grafos de Karaşan baseiam-se em uma topologia acíclica e em camadas [Bondy and Murty(1976), Sugiyama et al.(1981)], com  $M$  camadas, cada uma contendo  $W$  vértices. Cada par de camadas consecutivas  $m$  e  $m + 1$ ,  $m \in \{1, \dots, M - 1\}$  define um digrafo bipartido completo: cada vértice  $i \in m$  é conectado a todo vértice  $j \in m + 1$  por um arco. Além disso, o vértice fonte  $s$  é conectado a todos os vértices da camada 1 por um arco e o vértice *sink*  $t$  recebe um arco de todos os vértices da camada  $M$ , como demonstrado na Figura 2. Cada instância é referenciada como K-n-a-b-x-w, onde  $n$  simboliza o número de vértices e  $w$  se refere ao número de vértices por camada. Os parâmetros  $a$  e  $b$  são utilizados para criar os intervalos  $[l_{ij}; u_{ij}]$ . Para cada arco  $(i, j) \in A$ , um valor  $c \in [1; a]$  é gerado aleatoriamente utilizando uma distribuição uniforme. Então,  $l_{ij}$  e  $u_{ij}$  são gerados aleatoriamente nos intervalos  $[(1 - b)c; (1 + b)c]$  e  $[l_{ij}; (1 + b)c]$  respectivamente. O parâmetro  $x$  refere-se unicamente a identificação da instância. Grafos de Karaşan reivindicam representar redes de telecomunicações reais e são usados em experimentos computacionais em [Coco et al.(2014), Karaşan et al.(2001), Montemanni and Gambardella(2005)b, Montemanni et al.(2004), Pérez et al.(2012)].

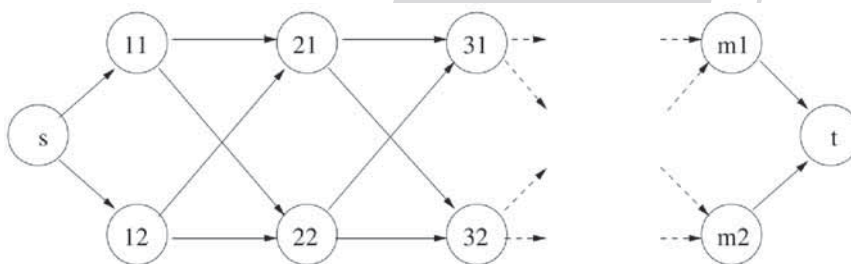


Figura 2: Um grafo de Karaşan com  $M = m$  camadas e largura  $W = 2$  [Karaşan et al.(2001)]

As Tabelas 1 e 2 relatam os resultados do CPLEX utilizando a formulação proposta e da heurística AM. O nome da instância é relatado na primeira coluna. A segunda e a terceira colunas correspondem respectivamente ao *lower bound* (lb) e ao *upper bound* (ub) computados pelo CPLEX. A quarta coluna representa o *gap* relativo, computado como  $\text{gap} = (\text{ub} - \text{lb})/\text{lb}$ , seguido do tempo de CPU (t) em segundos e o número de vértices (vértices) avaliados na árvore de *branch and bound*. O *gap* relativo ao *lower bound* da heurística AM (gap) e seu tempo de execução (t) em segundos são reportados respectivamente nas duas últimas colunas. A última linha



das Tabelas 1 e 2 relatam o valor médio de *gap* e desvio padrão para o *gap* do CPLEX e do AM, bem como o valor médio do tempo de execução em segundos.

instância	CPLEX					AM	
	lb	ub	gap	t	vértices	gap	t
K-100-200-0.9-a-2	22,00	22,00	0,00	0,72	1	0,00	0,01
K-100-200-0.9-b-2	83,00	83,00	0,00	0,80	1	0,00	0,01
K-100-200-0.9-a-5	228,00	228,00	0,00	2,74	3	0,00	0,02
K-100-200-0.9-b-5	370,00	370,00	0,00	4,58	3	0,26	0,01
K-100-200-0.9-a-10	39,00	39,00	0,00	4,07	1	2,50	0,02
K-100-200-0.9-b-10	27,00	27,00	0,00	4,16	1	0,00	0,02
K-100-200-0.9-a-25	14,00	14,00	0,00	2,73	1	0,00	0,04
K-100-200-0.9-b-25	32,00	32,00	0,00	7,55	1	3,03	0,03
K-100-200-0.9-a-50	14,00	14,00	0,00	2,52	1	6,67	0,05
K-100-200-0.9-b-50	9,00	9,00	0,00	2,50	1	0,00	0,04
média			0,00 ± 0,00	3,23		1,24 ± 2,10	0,02

Tabela 1: Resultados do CPLEX e AM para instâncias de Karaşan [Coco et al.(2014)] com 100 vértices

instância	CPLEX					AM	
	lb	ub	gap	t	vértices	gap	t
K-200-200-0.9-a-2	157.650,24	169.332,00	6,90	7200	1.577	6,90	0,02
K-200-200-0.9-b-2	123.798,07	128.538,00	3,69	7200	4.051	4,35	0,02
K-200-200-0.9-a-5	24.060,13	25.953,00	7,29	7200	1.013	8,43	0,05
K-200-200-0.9-b-5	14.602,57	15.394,00	5,14	7200	599	7,05	0,02
K-200-200-0.9-a-10	7.319,27	8.251,00	11,29	7200	350	18,17	0,10
K-200-200-0.9-b-10	6.655,22	7.120,00	6,53	7200	520	12,41	0,10
K-200-200-0.9-a-25	1.975,04	2.037,00	3,04	7200	1.005	5,63	0,21
K-200-200-0.9-b-25	2.296,83	2.382,00	3,58	7200	1.230	6,82	0,21
K-200-200-0.9-a-50	1.050,00	1.050,00	0,00	370	974	2,86	0,31
K-200-200-0.9-b-50	895,00	895,00	0,00	302	1.010	4,48	0,34
média			4,74 ± 3,28	5.827,2		7,71 ± 4,28	0,13

Tabela 2: Resultados do CPLEX e AM para instâncias de Karaşan [Coco et al.(2014)] com 200 vértices

Como pode ser visto na Tabela 1, instâncias com 100 vértices podem ser resolvidas em poucos segundos com o CPLEX. A heurística AM é duas ordens de magnitude mais rápida, produzindo soluções próximas ao ótimo, exceto para 3 instâncias, principalmente para K-100-200-0.9-a-50. A situação torna-se diferente para instâncias com 200 vértices. O CPLEX só foi capaz de resolver 2 instâncias em menos de duas horas. O *gap* para as 8 instâncias restantes não é superior a 12%. A heurística ainda computa soluções em menos de 1 segundo, mas o *gap* atinge um valor significativamente maior (aproximadamente 7,7% na media contra um valor de aproximadamente 1,2% para instâncias de 100 vértices).

## 7. Conclusões

Neste trabalho, nós propusemos uma formulação por Programação Linear Inteira Mista para a APMC-R bem como uma heurística eficiente que pode ser aplicada a redes 6LoWPAN. Os experimentos foram executados baseados nos clássicos grafos de Karaşan. O CPLEX resolveu todas as instâncias com 100 vértices e duas instâncias com 200 vértices em menos de 7200 segundos. Deste modo, o limite em tamanho para o CPLEX com este modelo parece estar entre 100 e 200 vértices. Por outro lado, o *gap* relativo para as instâncias de 200 vértices é bastante baixo,

alcançando 4,7% em média. Isto pode deixar espaço para melhorias como métodos de plano de corte. A heurística AM obteve uma boa performance, sendo que seu tempo de CPU nunca excedeu 1 segundo e seu *gap* relativo permaneceu limitado.

Estes resultados indicam que o uso da APMC-R para definir a árvore de roteamento em redes 6LoWPAN podem resultar em redes eficientes, resilientes e confiáveis, minimizando o pior cenário da rede através do critério de robustez e tornando as redes mais tolerantes a falhas, devido a otimização levar em conta a variabilidade nos enlaces da rede.

### Referências

- 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.
- Averbakh, I.** (2005). Computing and minimizing the relative regret in combinatorial optimization with interval data. *Discrete Optimization*, 2:273–287.
- Averbakh, I. and Lebedev, V.** (2004). Interval data minmax regret network optimization problems. *Discrete Applied Mathematics*, 138:289–301.
- Bellman, R.** (1956). On a routing problem. Technical report, DTIC Document.
- Ben-Tal, A. and Nemirovski, A.** (2002). Robust optimization—methodology and applications. *Mathematical Programming*, 92(3):453–480.
- Bertsekas, D. P. and Tsitsiklis, J. N.** (1991). An analysis of stochastic shortest path problems. *Mathematics of Operations Research*, 16:580–595.
- Bondy, J. A. and Murty, U. S. R.** (1976). *Graph theory with applications*, volume 290. Macmillan London.
- Candia-Véjar, A., Álvarez-Miranda, E., and Maculan, N.** (2011). Minmax regret combinatorial optimization problems: an algorithmic perspective. *RAIRO-Operation Reserach*, 45:101–129.
- Catanzaro, D., Labbé, M., and Salazar-Neumann, M.** (2011). Reduction approaches for robust shortest path problems. *Computers & Operations Research*, 38:1610–1619.
- Chandy, K. M. and Misra, J.** (1982). Distributed computation on graphs: Shortest path algorithms. *Communications of the ACM*, 25(11):833–837.
- 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*, 60(2):265–287.
- Conde, E.** (2012). On a constant factor approximation for minmax regret problems using a symmetry point scenario. *European Journal of Operational Research*, 219:452–457.
- Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C.** (2009). Single-source shortest paths. In *Introduction to Algorithms*, pages 580–619. MIT Press Cambridge.
- Dijkstra, E. W.** (1959). A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271.
- Escoffier, B., Monnot, J., and Spanjaard, O.** (2008). Some tractable instances of interval data minmax regret problems. *Operations Research Letters*, 36:424–429.

- Gabrel, V., Murat, C., and Wu, L.** (2013). New models for the robust shortest path problem: complexity, resolution and generalization. *Annals of Operations Research*, 207:97–120.
- Giusto, D., Lera, A., Morabito, G., and Atzori, L.** (2010). *The Internet of Things*. Springer.
- Humblet, P. A.** (1991). Another adaptive distributed shortest path algorithm. *IEEE transactions on communications*, 39(6):995–1003.
- Karaşan, O. E., Yaman, H., and Pinar, M. C.** (2001). The robust shortest path problem with interval data. Technical report, Bilkent University, Department of Industrial Engineering.
- Kasperski, A., Kobylański, P., Kulej, M., and Zieliński, P.** (2005). *Minimizing maximal regret in discrete optimization problems with interval data*, pages 193–208. Akademicka Oficyna Wydawnicza EXIT, Warszawa.
- Kasperski, A. and Zieliński, P.** (2006). An approximation algorithm for interval data minmax regret combinatorial optimization problems. *Information Processing Letters*, 97(5):177–180.
- 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.
- Kouvelis, P. and Yu, G.** (1997). *Robust discrete optimization and its applications*, volume 14 of *Nonconvex optimization and its applications*. Springer.
- Montemanni, R. and Gambardella, L. M.** (2005a). A branch and bound algorithm for the robust spanning tree problem with interval data. *European Journal of Operational Research*, 161(3):771–779.
- Montemanni, R. and Gambardella, L. M.** (2005b). The robust shortest path problem with interval data via benders decomposition. *4OR*, 3(4):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.
- Nie, Y. and Wu, X.** (2009). Shortest path problem considering on-time arrival probability. *Transportation Research Part B*, 43:597–613.
- Pérez, F., Astudillo, C. A., 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)*.
- Salazar-Neumann, M.** (2007). The robust shortest path and the single-source shortest path problems: interval data. In *Annales Du LAMSADE*.
- Shelby, Z. and Bormann, C.** (2011). *6LoWPAN: The wireless embedded Internet*, volume 43 of *Wiley Series in Communications Networking & Distributed Systems*. John Wiley & Sons.
- Spall, J. C.** (2003). *Introduction to Stochastic Search and Optimization*. Wiley.
- Sugiyama, K., Tagawa, S., and Toda, M.** (1981). Methods for visual understanding of hierarchical system structures. *Systems, Man and Cybernetics, IEEE Transactions on*, 11(2):109–125.
- Träff, J. L.** (1995). An experimental comparison of two distributed single-source shortest path algorithms. *Parallel Computing*, 21(9):1505–1532.

**Vasseur, J., Agarwal, N., Hui, J., Shelby, Z., Bertrand, P., and Chauvenet, C.** (2011). Rpl: The ip routing protocol designed for low power and lossy networks. *Internet Protocol for Smart Objects (IPSO) Alliance*.

**Winter, T.** (2012). Rpl: Ipv6 routing protocol for low-power and lossy networks. Technical report, Internet Engineering Task Force.

**Wu, B. Y. and Chao, K.** (2004). Shortest-paths trees. In *Spanning Trees and Optimization Problems*, pages 23–40. CRC Press.

