

# Heurística para o problema de construção de Árvore de Caminhos Mais Curtos Robusta para a Internet das Coisas

Iago A. Carvalho<sup>1</sup>, Marcelo A. Gomes<sup>1</sup>, Thiago F. Noronha<sup>1</sup>,  
Christophe Duhamel<sup>2</sup>, Luiz F. M. Vieira<sup>1</sup>

<sup>1</sup>Departamento de Ciência da Computação - Universidade Federal de Minas Gerais  
Belo Horizonte - MG - Brasil

<sup>2</sup>LIMOS, ISIMA - Université Blaise Pascal  
Clermont-Ferrand, France

iagoac@ufmg.br, {marcelog, tfn, lfvieira}@dcc.ufmg.br,

christophe.duhamel@isima.fr

**Abstract.** *In order to Internet of Things (IoT) become a reality, many challenges still need to be outweighed. Efficient protocols have to be specially resilient to high variations in transmission quality, due to constant changes in the network surrounding, which is characteristic of IoT. The most promising of these protocols is the IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL). In this paper, we extend the RPL protocol in order to consider the uncertainty in the link quality. We model the RPL routing problem as a robust optimization variant of the Shortest Path Tree Problem, called Robust Shortest Path Tree (RSPT), in which the cost of each arc is defined by an interval of feasible values, instead of a single value. Then, we propose a heuristic that can be implemented as an extension of RPL protocol, providing a communication tree 13 % more efficient.*

**Resumo.** *Para que a Internet das Coisas (IdC) torne-se uma realidade, muitos desafios ainda necessitam ser superados. Protocolos eficientes necessariamente precisam ser resilientes a variações na qualidade da transmissão, devido a constantes mudanças nos enlaces da rede, uma característica da IdC. O mais promissor destes protocolos é o IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL). Neste trabalho nós estendemos o protocolo RPL de forma a considerar a incerteza na qualidade dos enlaces. O problema de roteamento do protocolo 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 (ACMC-R), na qual cada arco é definido em um intervalo de valores factíveis ao invés de um único valor. Então, nós propomos uma heurística que pode ser implementada como uma extensão do protocolo RPL, obtendo uma árvore de comunicações até 13% mais eficiente.*

## 1. Introdução

Hoje em dia, existe 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 entorno da rede.

A principal solução para este desafio é o IPv6 Low Wireless Personal Area Networks (6LoWPAN) [Shelby and Bormann 2011], que são 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 constroi um Grafo Acíclico Direto de Orientado (GADDO) 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 GADDO é construído levando em consideração o alcance de transmissão dos sensores e a distância entre os sensores na rede. Cada aplicativo de rede tem seu próprio GADDO e o roteamento de mensagens para cada aplicação é feita de forma independente. Toda a comunicação entre os sensores é realizada através deste GADDO e é retransmitida pelo *sink*. Pode existir um número exponencialmente grande 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 de 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 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 [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. Experimentos computacionais realizados mostram que a heurística proposta cria um GADDO de melhor qualidade do que o atual protocolo RPL, quando este observa a qualidade dos enlaces em um determinado instante de tempo.

O restante deste artigo está organizado da seguinte forma. Na Seção 2, nós definimos a ACMC-R. Problemas relacionados são discutidos na Seção 3. Um algoritmo

heurístico é proposto na Seção 4. Em sequência, resultados computacionais são reportados na Seção 5. 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, deixe que  $n = |V|$  e  $m = |A|$  sejam respectivamente o número total de vértices e o número total de arcos de  $G$ . O problema do Caminho Mais Curto (CMC) consiste em encontrar um caminho entre um vértice de origem  $s \in V$  para um vértice de destino  $t \in V$  sendo que o custo total do caminho é o mínimo possível. Uma solução existe se não existe 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. O problema de roteamento do RPL definido em [Vasseur et al. 2011, Winter 2012] consiste em uma implementação distribuída do algoritmo de Dijkstra para a ACMC, onde 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.

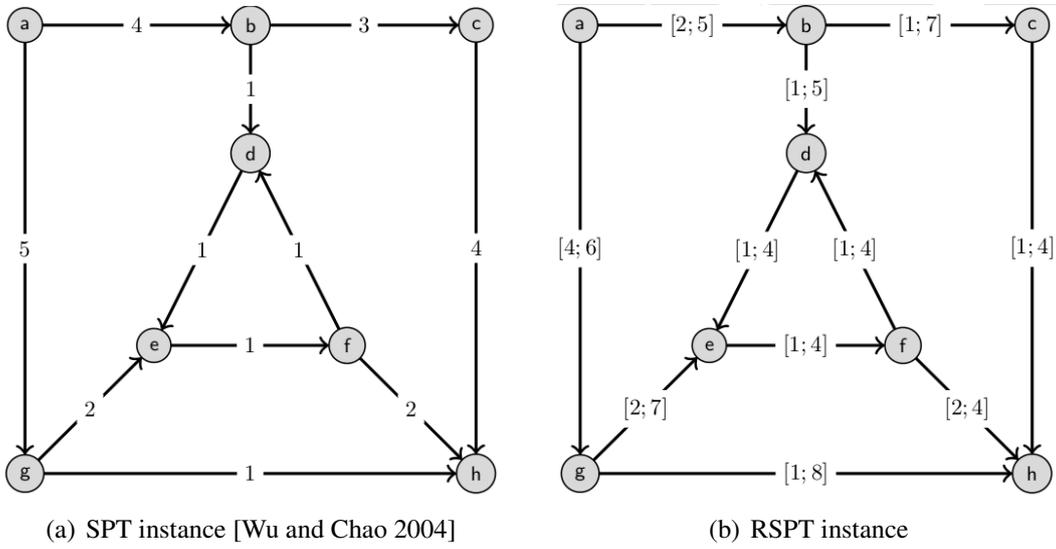
Este trabalho foca a otimização robusta, um modelo de otimização sob incerteza no qual a variabilidade dos dados é representada por valores determinísticos. Neste trabalho, foram focados modelos de otimização robusta onde os dados incertos podem ser modelados em um intervalo de valores possíveis. Nos 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 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  $u_{ij} \geq l_{ij} > 0$ , para todo  $(i, j) \in A$  [Karaşan et al. 2001]. Existem diferentes versões do CMC-R com intervalo de datas na literatura. Elas diferem uma da outra pelo critério de otimização utilizado [Aissi et al. 2009, Averbakh 2005, Candia-Véjar et al. 2011, Kasperski et al. 2005, Montemanni and Gambardella 2005a, Montemanni and Gambardella 2005b, 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 chamada *minmax regret CMC-R* [Kouvelis and Yu 1997, Montemanni et al. 2004, Montemanni and Gambardella 2005b, Coco et al. 2014]. Seja  $P \subseteq A$  um caminho com origem em  $s$  e destino  $t$  em  $G$ . Além disto, seja um cenário  $r$  uma realização do custo dos arcos  $c_{ij}^r \in [l_{ij}, u_{ij}]$  para cada arco  $(i, j) \in A$ . 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, ou seja, o caminho de  $s$  a  $t$  para o qual a máxima diferença entre  $P$  e  $S^r$  é a menor possível, para todos os cenários  $r \in \mathcal{R}$ . 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  $u_{ij} \geq l_{ij} > 0$ , 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 dispositivos na IdC, os arcos em  $E$  são associados aos enlaces,  $l_{ij}$  corresponde ao menor valor observado pela métrica de qualidade do enlace, e  $u_{ij}$  corresponde ao maior valor observado por esta mesma métrica.



**Figura 1. Examples of the Shortest Path Tree and the RSPT**

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  $A[p_i^t] \in A$  o conjunto de arcos que compõe o caminho  $p_i^t$ , e  $\mathcal{R}$  o conjunto de todos os possíveis cenários em  $G$  e  $c^r(p_i^t) = \sum_{(i,j) \in A[p_i^t]} c_{ij}^r$  o custo de  $p_i^t$  no cenário  $r$ , e  $c^r(p_i^*)$  o custo do caminho mais curto 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  $r$ .

**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 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

Não existem trabalhos na literatura para o problema variante da ACMC estudado neste trabalho. Entretanto, existem diversos trabalhos sobre o CMC-R.

O *minmax regret* CMC-R foi demonstrado ser NP-Difícil em [Kouvelis and Yu 1997]. Vários trabalhos resolvem o CMC-R por métodos exatos. Uma formulação por Programação Linear Inteira Mista (PLIM) para este problema foi introduzida em [Karaşan et al. 2001]. Em [Montemanni et al. 2004], os autores propuseram um algoritmo *branch-and-bound* baseado na relaxação combinatória das restrições robustas. Este trabalho foi estendido em [Montemanni and Gambardella 2005b] 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 trabalhos reportam algoritmos heurísticos desenvolvidos para o CMC-R. 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  $O(m + n \log n)$  no pior caso 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  $O(q(n, n/\epsilon))$  no pior caso, onde  $q$  é uma função polinomial bivariada e  $\epsilon$  é um valor no intervalo  $[0, 1]$ . Uma série de heurísticas para o CMC-R foram desenvolvidas e comparadas em [Coco et al. 2014], sendo que o algoritmo 2-aproximativo de [Kasperski and Zieliński 2006] obteve um *gap* relativo de 76 %, na média, para instâncias com até 1,500 vértices.

### 4. 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 determinística 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.

Neste trabalho, propomos uma heurística para a ACMC-R, estendendo a heurística AM proposta em [Kasperski and Zieliński 2006]. O custo de todos os arcos são fixados em seus valores médios e então a árvore de caminhos mais curtos neste cenário

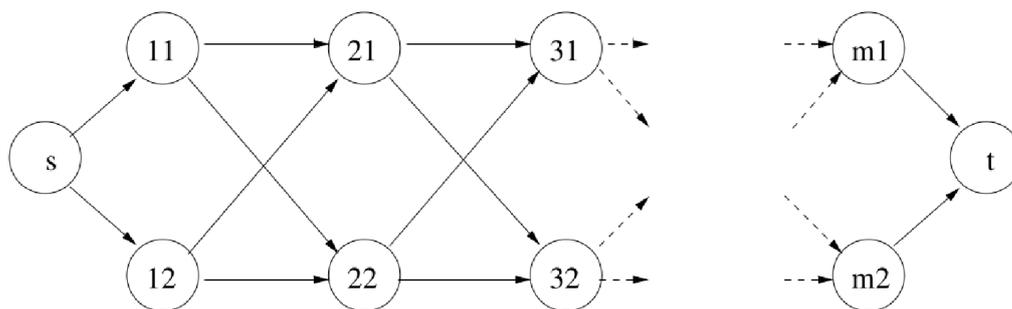
é retornada. Esta heurística pode ser implementada de forma distribuída, pois trata-se de um algoritmo de caminho mais curto em grafos [Chandy and Misra 1982, Träff 1995, Humblet 1991].

Na próxima Seção, esta heurística proposta será comparada com o atual protocolo RPL, de modo a avaliar sua performance quando aplicada a construção de uma árvore de comunicações para dispositivos na IdC.

## 5. Experimentos 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 heurística AM foi implementada em C++ e compilada com o GNU GCC versão 4.7.3.

As instâncias são baseadas nos grafos de Karaşan [Karaşan et al. 2001] previamente utilizados em [Coco et al. 2014]. Grafos de Karaşan baseiam-se em uma topologia acíclica [Bondy and Murty 1976] e em camadas [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 referida como K-n-a-b-x-w, onde  $n$  é 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 randomicamente utilizando uma distribuição uniforme. Então,  $l_{ij}$  e  $u_{ij}$  é gerado randomicamente 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 propõe-se a representar redes de telecomunicações reais e são usados em experimentos computacionais em [Coco et al. 2014, Karaşan et al. 2001, Montemanni and Gambardella 2005b, 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]**

De maneira a comprovar a eficiência da heurística AM, as Tabelas 1 e 2 relatam resultados computacionais comparando a heurística proposta com os GADDOs cujos pesos dos arcos foram definidos aleatoriamente dentro do intervalo  $[l_{ij}; u_{ij}]$ , para todo arco  $(i, j) \in A$ . A primeira coluna contém o nome de cada instância. A segunda, terceira e quarta colunas representam o *regret* mínimo, médio e máximo obtidos a partir de 100 GADDOs aleatórios. A quinta coluna relata o coeficiente de variação (cv) calculado como

o desvio padrão dos experimentos dividido por seu valor médio. A sexta coluna relata o tempo de execução (s) em segundos da construção destes 100 cenários aleatórios. A sexta coluna. A sétima e a oitava colunas relatam, respectivamente, o *regret* encontrado pela heurística AM e seu tempo de execução. A última coluna relata o desvio relativo percentual (% dev) dos GADDOs gerados aleatoriamente em comparação com a heurística AM, calculado como (média - AM)/média.

Como pode ser visto na Tabela 1, instâncias com 100 vértices podem ser resolvidas em poucos segundos com a heurística AM. A heurística encontra soluções 13 % melhores que o RPL, em média. Além disto, a solução encontrada pela AM é melhor do que 100 execuções do RPL para 4 das 10 instâncias testadas. Já para instâncias com 200 vértices, como demonstrado na Tabela 2, a heurística AM encontra uma solução melhor que 100 execuções do RPL para todas as instâncias executadas, obtendo um *regret* 17% menor que o RPL, em média.

| instancia          | RPL    |        |        |      |       | AM     |       |       |
|--------------------|--------|--------|--------|------|-------|--------|-------|-------|
|                    | mínimo | médio  | máximo | cv % | t (s) | regret | t (s) | % dev |
| K-100-200-0.9-a-2  | 22,00  | 24,76  | 30,00  | 11   | 0,55  | 22,00  | 0,01  | 11    |
| K-100-200-0.9-b-2  | 83,00  | 94,77  | 108,00 | 9    | 0,56  | 83,00  | 0,01  | 12    |
| K-100-200-0.9-a-5  | 228,00 | 239,96 | 285,00 | 4    | 1,16  | 228,00 | 0,02  | 4     |
| K-100-200-0.9-b-5  | 370,00 | 406,51 | 455,00 | 8    | 1,12  | 371,00 | 0,01  | 8     |
| K-100-200-0.9-a-10 | 42,00  | 45,52  | 61,00  | 11   | 2,09  | 40,00  | 0,02  | 12    |
| K-100-200-0.9-b-10 | 27,00  | 31,50  | 59,00  | 26   | 2,16  | 27,00  | 0,02  | 14    |
| K-100-200-0.9-a-25 | 15,00  | 17,42  | 23,00  | 15   | 3,70  | 14,00  | 0,04  | 19    |
| K-100-200-0.9-b-25 | 34,00  | 41,63  | 55,00  | 12   | 3,66  | 33,00  | 0,03  | 20    |
| K-100-200-0.9-a-50 | 14,00  | 16,33  | 27,00  | 15   | 4,08  | 15,00  | 0,05  | 11    |
| K-100-200-0.9-b-50 | 10,00  | 11,23  | 17,00  | 20   | 3,98  | 9,00   | 0,04  | 19    |
| média              |        |        |        |      | 13    | 2,30   | 0,02  | 13    |

**Tabela 1. Resultados do RPL e AM para instâncias de Karaşan [Karaşan et al. 2001, Coco et al. 2014] com 100 vértices**

| instancia          | RPL        |            |            |      |       | AM         |       |       |
|--------------------|------------|------------|------------|------|-------|------------|-------|-------|
|                    | mínimo     | médio      | máximo     | cv % | t (s) | regret     | t (s) | % dev |
| K-200-200-0.9-a-2  | 179.826,00 | 209.874,20 | 251.680,00 | 7    | 2,25  | 169.349,00 | 0,02  | 19    |
| K-200-200-0.9-b-2  | 130.899,00 | 151.155,10 | 186.857,00 | 7    | 2,30  | 129.435,00 | 0,02  | 14    |
| K-200-200-0.9-a-5  | 27.005,00  | 33.505,96  | 50.013,00  | 15   | 4,94  | 26.770,00  | 0,05  | 20    |
| K-200-200-0.9-b-5  | 16.463,00  | 18.251,54  | 25.236,00  | 8    | 4,89  | 15.711,00  | 0,02  | 13    |
| K-200-200-0.9-a-10 | 8.771,00   | 10.368,89  | 22.062,00  | 14   | 8,71  | 8.495,00   | 0,10  | 18    |
| K-200-200-0.9-b-10 | 7.576,00   | 8.510,08   | 11.666,00  | 8    | 9,08  | 7.484,00   | 0,10  | 12    |
| K-200-200-0.9-a-25 | 2.174,00   | 2.522,94   | 3.182,00   | 7    | 19,86 | 2.093,00   | 0,21  | 17    |
| K-200-200-0.9-b-25 | 2.568,00   | 2.924,32   | 3.375,00   | 6    | 19,44 | 2.465,00   | 0,22  | 15    |
| K-200-200-0.9-a-50 | 1.175,00   | 1.362,05   | 1.702,00   | 6    | 29,26 | 1.081,00   | 0,31  | 20    |
| K-200-200-0.9-b-50 | 1.014,00   | 1.124,22   | 1.263,00   | 5    | 30,83 | 879,00     | 0,34  | 20    |
| média              |            |            |            |      | 8     | 12,67      | 0,13  | 17    |

**Tabela 2. Resultados do RPL e AM para instâncias de Karaşan [Karaşan et al. 2001, Coco et al. 2014] com 200 vértices**

## 6. Conclusões

Neste trabalho, nós propusemos uma heurística para a construção de uma Árvore de Caminhos Mais Curtos Robusta para a Internet das Coisas, que pode facilmente ser aplicada a redes 6LoWPAN. Experimentos computacionais foram executados baseados nos clássicos grafos de Karaşan. Para a construção de uma árvore de comunicações, a heurística AM proposta apresentou resultados 13 % melhores do que 100 execuções do atual protocolo RPL para 14 das 20 instâncias propostas, quando comparado o valor de *regret* das soluções encontrada, sendo que seu tempo computacional nunca excedeu 1 segundo.

Nossos resultados indicam que a geração da Árvore de Caminhos Mais Curtos Robusta através da heurística proposta consiste em uma abordagem mais eficiente para a geração de árvores de roteamento em redes 6LoWPAN, uma tecnologia promissora para a implementação da Internet das Coisas.

## 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.
- Bellman, R. (1956). On a routing problem. Technical report, DTIC Document.
- 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 Research*, 45:101–129.
- 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.
- Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271.
- 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., Kobylanński, P., Kulej, M., and Zieliński, P. (2005). *Minimizing maximal regret in discrete optimization problems with interval data*, pages 193–208. Akademia 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.
- 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.
- 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.