Heurística de Refinamento para o Problema de Caminho Mais Curto Robusto

Amadeu A. Coco
DCC – Universidade Federal de Minas Gerais/ICD-LOSI – Université de Tecnologie de Troyes
CEP – 31270-010 – Belo Horizonte – MG – Brasil/CS 42060 - 10004 Troyes – France
amadeuac@dcc.ufmg.br

Thiago F. Noronha
DCC – Universidade Federal de Minas Gerais
CEP – 31270-010 – Belo Horizonte – MG – Brasil
tfn@dcc.ufmg.br

Andréa Cynthia Santos
ICD-LOSI – Université de Tecnologie de Troyes
CS 42060 - 10004 Troyes – France
andrea_cynthia.duhamel@utt.fr

RESUMO

O Problema de Caminho Mais Curto (SP, do inglês *Shortest Path*) consiste em encontrar um caminho a partir de um vértice de origem $s \in V$ para um vértice de destino $t \in V$, onde o custo total é mínimo. SP modela problemas práticos e teóricos. Porém, diversas aplicações para o SP dependem de dados não conhecidos à priori. O Problema do Caminho Mais Curto Robusto (RSP, do inglês *Robust Shortest Path*) é uma generalização do SP. No RSP, o custo de cada arco é definido por um intervalo de possíveis valores para o custo do arco. O objetivo do RSP é encontrar o caminho entre s e t que possui o menor custo robusto relativo. Este problema é conhecido como *minmax relative regret* RSP e é NP-Difícil. Neste trabalho é proposto uma heurística de refinamento para o *minmax relative regret* RSP. Os resultados dos experimentos computacionais mostram que a heurística de refinamento proposta obteve resultados melhores do que as heurísticas da literatura.

PALAVRAS CHAVE. Caminho Mais Curto Robusto, Otimização Robusta, Heurísticas.

ABSTRACT

The well-known Shortest Path problem (SP) consists in finding a shortest path from a source node $s \in V$ to a destination node $t \in V$ such that the total cost is minimized. The SP models practical and theoretical problems. However, several shortest path applications rely on uncertain data. The Robust Shortest Path problem (RSP) is a generalization of SP. In the former, the cost of each arc is defined by an interval of possible values for the arc cost. The objective is to find a path that minimizes the maximum relative regret cost from s to t. This problem is known as the *minmax relative regret* RSP and is NP-Hard. In this study, a new heuristic for the *minmax relative regret* RSP is proposed. The computational

experiments show that the proposed heuristic produces better results than the heuristics found in the literature.

KEYWORDS. Robust Shortest Path, Robust Optimization, Heuristics.

1. Introdução

Seja $G = (V, A)$ um grafo direcionado, no qual V é um conjunto de vértices e A é um conjunto de arcos, onde cada arco $(i, j) \in A$ está associado a um peso $c_{ij} \in \mathbb{R}$. Dados $\{s, t\} \in V$, o Problema do Caminho Mais Curto (SP, do inglês *Shortest Path Problem*) consiste em encontrar um caminho entre o vértice de origem s e o vértice de destino t em G cujo somatório dos custos associados aos arcos no caminho seja mínimo. O SP é estudado desde a década de 50 (Shimbel, 1955) e sua complexidade computacional é $\theta(|V| \cdot \log |V|)$ (Cormen et al., 2009). Quando o custo associado aos arcos é positivo, o algoritmo mais utilizado para sua resolução é o de Dijkstra (Dijkstra, 1959). Quando o custo associado aos arcos pode ser negativo (sem ciclos de custo negativo), o algoritmo mais utilizado na literatura é o de Bellmann-Ford (Bellman, 1958) com complexidade $\mathcal{O}(|V|^2 \cdot \log |V|)$ (Cormen et al., 2009).

SP modela diversos problemas de interesse prático e teórico (Gallo e Pallottino, 1986). Entretanto, neste problema cada arco $(i, j) \in A$ está associado a um custo fixo. Dessa forma, vários problemas de caminho mais curto encontrados no mundo real são dificilmente modelados via SP, pois os dados deste tipo de problema frequentemente estão sujeitos a incertezas.

Algumas estratégias podem ser aplicadas para resolução de problemas nos quais os dados estão sujeitos a incertezas. As modelagens mais comuns na literatura são a programação estocástica (Spall, 2003) e a otimização robusta (Ben-Tal e Nemirovski, 2002; Kouvelis e Yu, 1997). Neste trabalho, o foco está nos modelos de otimização robusta, onde as incertezas são modeladas por um intervalo de possíveis valores. Para o leitor que quiser estudar outras formas de modelagens é sugerido o livro de Kouvelis e Yu (1997). O Problema de Caminho mais Curto Robusto (RSP, do inglês *Robust Shortest Path Problem*) é uma generalização do SP, onde o custo de cada arco $(i, j) \in A$ é definido por um intervalo $[l_{ij}, u_{ij}]$, com $l_{ij}, u_{ij} \in \mathbb{N}$, onde $u_{ij} \geq l_{ij} \geq 0$, $\forall (i, j) \in A$ (Karasan et al., 2001). Vale salientar que na literatura existem diferentes versões de RSP com a incerteza modelada de forma intervalar, esses trabalhos se diferem entre si pelo critério de otimização utilizado (Aissi et al., 2009; Averbakh, 2005; Candia-Véjar et al., 2011; Kasperski et al., 2005; Montemanni e Gambardella, 2005; Montemanni et al., 2004).

A versão mais estudada do RSP utiliza o critério *minmax regret* e é chamada de MR-RSP. Seja $P \subseteq A$ um caminho que parte de um vértice de origem s até um vértice de destino t em G. O *arrependimento* de P no cenário r, também chamado de *desvio robusto* de P em r, é definido como a diferença entre o custo de P no cenário r e o custo do caminho mais curto S^r entre s e t em r. Ou seja, o *desvio robusto* de P em r é o custo extra pago na utilização de P ao invés de S^r, caso o cenário r ocorra. O *custo robusto* de P é o maior desvio robusto de P sobre todos os cenários. Portanto, o MR-RSP consiste em encontrar o caminho P^* de s para t com o menor custo robusto. Este problema é NP-Difícil mesmo para grafos acíclicos (Kouvelis e Yu, 1997).

Este trabalho é dedicado ao *minmax relative regret* RSP (MRR-RSP) proposto por (Kouvelis e Yu, 1997; Averbakh, 2005). Seja $P \subseteq A$ um caminho que parte de um vértice de origem s até um vértice de destino t em G. O *arrependimento relativo* de P no cenário r, também chamado de *desvio robusto relativo* de P em r, é definido como $(cost(P, r) - cost(S^r, r))/cost(S^r, r)$, onde $custo(P, r)$ é o custo de P em r e $custo(S^r, r)$ é o custo do caminho mais curto S^r entre s e t em r. O *custo robusto relativo* de P é o maior desvio

de P sobre todos os cenários. O MRR-RSP consiste em encontrar o caminho P^* entre s e t com o menor custo robusto relativo. Este problema é NP-Difícil mesmo para grafos acíclicos (Averbakh, 2005). Apesar do MRR-RSP se diferenciar do MR-RSP apenas na função objetivo, ele possui maior complexidade de resolução, pois sua função objetivo é não linear.

O *custo robusto relativo* é uma métrica melhor que o *custo robusto*, porque o arrependimento de usar P ao invés de S^r é normalizado pelo custo de S^r. Por exemplo, dados dois caminhos P' e P'' e dois cenários r' e r'' onde $custo(P', r') = 11$, $custo(S^{r'}, r') = 1$, $custo(P'', r'') = 100$ e $custo(S^{r''}, r'') = 90$. De acordo com o critério *minmax regret*, o arrependimento de P' em r' $(11 - 1 = 10)$ é o mesmo que o de P'' in r'' $(100 - 90 = 10)$. Porém, pode-se verificar que o custo de P' é 10 vezes maior que o custo de $S^{r'}$, enquanto o custo de P'' é apenas 11% maior que o custo de $S^{r''}$. Portanto, de acordo com critério *minmax relative regret*, o arrependimento de P' em r' $((11 - 1)/1 = 10)$ é muito maior que o arrependimento de P'' em r'' $((100 - 90)/90 = 0.11)$.

O restante deste trabalho está organizado da seguinte forma. Na Seção 2, são apresentadas as referências bibliográficas relativas ao RSP. Na Seção 3, é apresentada a heurística de refinamento proposta neste trabalho. Na Seção 4, são apresentados os experimentos computacionais. Por fim, na Seção 5, são apresentadas as conclusões do trabalho.

2. Trabalhos Relacionados

O problema de Caminho Mais Curto Robusto (RSP) é estudado desde o fim da década de 60 (Gupta e Rosenhead, 1968; Rosenhead et al., 1972). Kouvelis e Yu (1997) sugeriram a utilização da otimização robusta no auxílio à tomada de decisões quando o conhecimento sobre o estado atual de um problema não pode ser determinado. Kouvelis e Yu (1997), também, apresentaram pela primeira vez as modelagens intervalar e por cenários discretos, além dos critérios de avaliação absoluto, arrependimento absoluto e arrependimento relativo. Por fim, foi provada que a complexidade computacional de problemas de otimização robusta nos quais a modelagem é intervalar e o critério de avaliação é arrependimento absoluto, como o MR-RSP, é NP-Difícil.

Karasan et al. (2001) trabalharam com o MR-RSP. Nesse trabalho, foi proposta a primeira formulação baseada em programação inteira mista para MR-RSP. Além disto, Karasan et al. (2001) introduziram o conceito de arco fraco (do inglês, *weak-arc*). Arco Fraco é aquele arco do grafo que jamais estará em uma solução ótima do problema. Baseado nisso, foi proposto um algoritmo de pré-processamento para o MR-RSP. Porém, o algoritmo funciona apenas para grafos acíclicos, planares ou em camadas.

Montemanni et al. (2004) propuseram algoritmos exatos para MR-RSP. Montemanni et al. (2004) desenvolveram um algoritmo baseado na técnica *Branch-and-Bound* que encontra soluções ótimas em um tempo menor que aquele encontrado pela formulação matemática de Karasan et al. (2001) implementada no *solver* CPLEX[1]. Montemanni e Gambardella (2005) aprimoraram os resultados de Montemanni et al. (2004) com a utilização de técnicas de pré-processamento baseadas naquelas propostas por Karasan et al. (2001). Montemanni e Gambardella (2005) propuseram um algoritmo baseado na decomposição de *Benders*, que obteve resultados semelhantes aos de Montemanni e Gambardella (2005). Os algoritmos apresentados nesses trabalhos representaram uma evolução signifi-

[1] http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/

cativa na resolução do MR-RSP, pois foram os primeiros algoritmos exatos apresentados para resolução do problema quando as incertezas são modeladas por um intervalo de possíveis valores. Os autores destes artigos resolveram instâncias com até 4.000 vértices e 160.000 arcos.

Averbakh (2005) trabalhou com a versão robusta de problemas de otimização combinatória, onde a modelagem dos cenários é intervalar e o critério *minmax relative regret*. Para esta classe de problemas, o autor propôs encontrar soluções que minimizam a diferença relativa do pior caso. Foi proposto, também, uma formulação genérica não linear para esta versão de problemas. Neste mesmo trabalho, também, foi provado que a complexidade deste critério de otimização robusta é Fortemente NP-Difícil, ou seja, não existem algoritmos aproximativos para esta classe. Por fim, foi apresentado o problema de sequenciamento robusto em uma máquina, que na otimização combinatória é polinomial e no critério *relative minmax regret* é NP-Difícil.

Kasperski e Zieliński (2006) propuseram três algoritmos para problemas de otimização robusta com modelagem intervalar, dentre os quais está o MR-RSP. Estas heurísticas podem ser aplicadas diretamente para MRR-RSP. A primeira, chamada AU, recebe como entrada um grafo $G = (V, A)$, os limites inferior l_{ij} e superior u_{ij} para o custo de cada arco $(i, j) \in A$, além do vértice de origem $s \in V$ e do vértice de destino $t \in V$. Em seguida, fixa-se o cenário r^u, onde o custo de todos os arcos estão em seu respectivo valor máximo, ou seja, $c^u_{ij} = u_{ij}, \forall (i, j) \in A$. Em seguida, calcula-se o caminho mais curto P^u entre s e t no cenário r^u, utilizando o algoritmo de Dijkstra. Por fim, o custo robusto de P^u é calculado e esta solução é retornada. A complexidade assintótica desta heurística no pior caso é $O(|V| \cdot \log |V|)$, assumindo-se que $|V| \cdot \log |V| \geq |A|$. A segunda heurística, chamada AM, recebe os mesmos parâmetros de entrada que AU. Primeiramente, fixa-se o cenário r^m, onde o custo de todos os arcos estão em seu respectivo valor médio, ou seja, $c^m_{ij} = (l_{ij} + u_{ij})/2, \forall (i, j) \in A$. Em seguida, calcula-se o caminho mais curto P^m entre s e t no cenário r^m, utilizando algoritmo de Dijkstra. Por fim, o custo robusto de P^m é calculado e esta solução é retornada. A complexidade desta heurística no pior caso é $O(|V| \cdot \log |V|)$, assumindo-se que $|V| \cdot \log |V| \geq |A|$. A terceira, chamada AMU, executa AM e AU e retorna a melhor solução encontrada dentre os 2 algoritmos. Vale salientar que AM e AMU são algoritmos aproximativos de fator 2 para MR-RSP, porém devido a conjectura de (Averbakh, 2005), elas não são 2-aproximativas para MRR-RSP.

Pérez et al. (2012) propuseram uma heurística baseada em *Simulated Annealing*, inspirada no trabalho de Kirkpatrick et al. (1983), para o MR-RSP. Esta heurística é chamada aqui de SA-RSP. Os parâmetros de SA-RSP são (i) a temperatura inicial $t_0 \in \mathbb{R}$, (ii) a temperatura final $t_f \in \mathbb{R}$, (iii) a taxa de decrescimento da temperatura β (onde $0 < \beta < 1$) e (iv) um valor $\lambda \in \mathbb{N}$ que determina o número de soluções aceitas antes da redução da temperatura. Inicialmente, SA-RSP executa a heurística AU para obter uma solução inicial. A seguir, os quatro procedimentos a seguir são executados a cada iteração. Primeiro, seleciona-se aleatoriamente um subconjunto $A' \subset A$. A seguir, executa-se a heurística AU no grafo $G' = (V, A')$ para que uma solução candidata P' seja obtida. No terceiro passo, P' é aceita se $custo(\mathcal{P}') < custo(\mathcal{P})$ ou se $e^{\frac{-\Delta}{t}} < \theta$, onde $custo(\mathcal{P})$ e $custo(\mathcal{P}')$ são os custos dos caminhos $\mathcal{P}$ e $\mathcal{P}'$, respectivamente, $\Delta = cost(\mathcal{P}') - cost(\mathcal{P})$, $\theta \in [0, 1[$ é um número gerado aleatoriamente e t é a temperatura corrente do *simulated annealing*. No quarto passo, $t = t * \beta$, se λ soluções foram aceitas desde a última atualização. Por fim, o algoritmo para se $t \leq t_f$, e a melhor solução encontra é retornada. Os resultados de Pérez

et al. (2012) indicam uma melhoria média de 0.3% em relação aos resultados apresentados por Kasperski e Zieliński (2006). Neste trabalho, foram resolvidas instâncias com até 20.000 vértices em tempos inferiores a 1240 segundos.

Recentemente, Coco et al. (2014) desenvolveram duas heurísticas para o MRR-RSP. A primeira, chamada de PM-RSP, foi baseada na metaheurística *Pilot Method* (Duin e Voss, 1999). O pseudo-código de PM-RSP é apresentado no Algoritmo 1. A heurística recebe como entrada um grafo $G = (V, A)$, o *lower bound* l_{ij}, o *upper bound* u_{ij}, para o custo de cada arco $(i, j) \in A$, a origem $s \in V$ e o destino $t \in V$. A solução parcial P' e a melhor solução viável conhecida são inicializadas na linha 2. O laço entre as linhas 4-13 é executado enquanto $\mathcal{P}'$ não é um caminho entre s e t, ou seja, enquanto t não é inserido em $\mathcal{P}'$. Na linha 4, identifica-se o último vértice u inserido em $\mathcal{P}'$. O laço das linhas 6 a 11 é executado para cada vértice $v \in \delta^+(u)$, onde $\delta^+(u)$ é o conjunto de vértices sucessores do vértice u. Na linha 7, o algoritmo de Dijkstra é aplicado para obter o caminho mais curto $\mathcal{P}_v$ entre v e t no cenário r^m. A seguir, os caminhos $\mathcal{P}'$ (entre s e u) e $\mathcal{P}_v$ (entre v e t) são concatenados na linha 8, formando o caminho $\mathcal{P}'_v$ (entre s e t). O custo robusto relativo de $\mathcal{P}'_v$ é utilizado como o custo guloso da inserção do vértice v na solução $v^* \in \delta^+(u)$ com o menor custo guloso e seu respectivo caminho $\mathcal{P}'_{v^*}$ são atualizados na linha 10. A seguir, v^* é inserido no fim de $\mathcal{P}'$ na linha 11. O algoritmo PM-RSP retorna a melhor solução encontrada pela heurística $\mathcal{P}^*$, que não é necessariamente $\mathcal{P}'$. Por fim, $\mathcal{P}^*$ é atualizado nas linhas 12-13 e retornado na linha 14. A complexidade de PM-RSP no pior caso é $\mathcal{O}(|A| \cdot |V| \cdot \log |V|)$.

Algoritmo 1: Pilot Method (PM)

Entrada: $G = (V, A)$, s, t
Saída: P^*

```
1  início
2      P' ← s e P* ← ∅
3      P'_v ← ∅
4      enquanto P' não é um caminho entre s e t faça
5          Seja u o último vértice inserido em P'
6          para cada v ∈ δ+(u) faça
7              P_v ← Dijkstra (v, t, G, r^m)
8              P'_v ← Concatenar (P', P_v))
9              se custo (P'_v) < custo(P'_{v*}) or P'_{v*} = ∅ então
10                 v* ← v and P'_{v*} ← P'_v
11             Insira v* no fim de P'
12         se custo(P'_{v*}) < custo(P*) então
13             P* ← P'_{v*}
14     retorna P*;
```

A segunda heurística desenvolvida por (Coco et al., 2014) foi baseada na metaheurística *Biased Random Key Genetic Algorithm* (BRKGA) (Gonçalves e Resende, 2011). O BRKGA proposto para o MMR-RSP, chamado de BRKGA-RSP, evolui uma população de cromossomos que consiste em vetores de números reais no intervalo $[0, 1]$, que são gerados de forma aleatória na população inicial. O *fitness* do cromossomo é dado pelo custo da so-

lução encontrada por uma heurística de decodificação que recebe como entrada os valores chave dos vetores de números reais e retorna uma solução viável com seu respectivo custo. Cada cromossomo p possui uma chave $k_{ij}^p \in [0,1]$ para cada arco $(i,j) \in A$ e a heurística de decodificação, chamada de FSH, realiza os seguintes passos. Primeiro, a heurística fixa o custo de todos os arcos para $c_{ij}^p = l_{ij} + (u_{ij} - l_{ij}) \cdot k_{ij}^p$, para todo $(i,j) \in A$, ou seja, o cenário induzido pelas chaves de p. Este cenário é chamado de r^p. A seguir, executa-se o algoritmo de Dijkstra para calcular o caminho mais curto $\mathcal{P}^p$ entre s e t no cenário r^p. Por fim, o custo robusto relativo de $\mathcal{P}^p$ é utilizado como *fitness* do cromossomo.

A cada nova geração, a população é particionada em dois conjuntos: TOP e $REST$. Consequentemente, o tamanho da população é $|TOP| + |REST|$. As melhores soluções são guardadas em TOP enquanto o resto é colocado em $REST$. Os cromossomos em TOP são copiados para a próxima geração. Os novos mutantes gerados de forma aleatória semelhante a população original, são colocados em um conjunto chamado BOT, onde $BOT \subset REST$. Os elementos restantes da nova população são obtidos por operação de cruzamento entre dois parentes, um escolhido aleatoriamente em TOP e outro em $REST$. Dessa forma, $|REST| - |BOT|$ soluções são geradas via cruzamento. O algoritmo para quando o número máximo de gerações é atingido. Os tamanhos dos conjuntos TOP, $REST$ e BOT são parâmetros que foram ajustados por Coco et al. (2014).

Até onde sabemos, não existem na literatura de MRR-RSP trabalhos que propõem heurísticas que melhorem os resultados encontrados por uma outra heurística. Desta forma, este trabalho propõe a primeira heurística de refinamento para MRR-RSP com o intuito de resolvê-lo de forma mais eficiente.

3. Heurística de Refinamento

Nesta seção, é proposta uma heurística de refinamento para resolver o MRR-RSP. Esta heurística, chamada HR-RSP, realiza uma busca no espaço de caminhos com o objetivo de guardar informações de caminhos encontrados anteriormente e utilizá-las de forma a encontrar novos caminhos. Na Seção 4, a heurística HR-RSP é comparada com AMU (Kasperski e Zieliński, 2006), SA-RSP (Pérez et al., 2012) e BRKGA-RSP (Coco et al., 2014). Heurísticas de busca no espaço de caminhos foram utilizadas com sucesso no OSPF-IS (Fortz e Thorup, 2003) e no Problema de Seleção de Sensores (Joshi e Boyd, 2009).

Neste trabalho, a heurística FSH, proposta por (Coco et al., 2014), é refinada para permitir uma pertubação controlada nos valores dos intervalos dos arcos. A heurística modificada proposta, chama-se FSHM. A cada iteração de FSHM, executa-se o algoritmo de (Dijkstra, 1959) juntamente com o cálculo do custo robusto relativo (Karasan et al., 2001), entretanto o valor dos intervalos a cada iteração é diferente. A cada iteração: (i) o intervalo de todos os arcos $(i,j) \in A|P$ são fixados em $[l_{ij}, m_{ij}]$, onde $m_{ij} = (l_{ij} + u_{ij})/2$, (ii) o intervalo de 10% dos arcos $(i,j) \in P$ é fixado em m_{ij} e, (iii) os outros arcos $(i,j) \in P$ tem seus intervalos fixados em $[m_{ij}, u_{ij}]$. O objetivo desta heurística de refinamento é permitir que o espaço de caminhos possa ser explorado de forma eficiente.

O Algoritmo 2 apresenta um pseudocódigo para HR-RSP. A heurística HR-RSP recebe como entrada um grafo $G = (V, A)$, os limites inferior l_{ij} e superior u_{ij} para o custo de cada arco $(i,j) \in A$, além do vértice de origem $s \in V$ e do vértice de destino $t \in V$. Na Linha 2, executa-se AMU e a solução encontrada por AMU é inserida em P^*. Na Linha

3, o contador i, que representa o número de iterações ocorridas desde a última melhora no custo de P^*, é inicializado com valor 0. Repete-se o laço das linhas 4-10 enquanto o valor de i for menor que valor máximo de iterações sem melhora da solução P^*. Na Linha 5, executa-se FSHM e insere-se a solução encontrada em P'. Na Linha 6, verifica-se se o custo robusto relativo de P' é menor que o de P^*. Caso $Custo(P') < Custo(P^*)$, P^* recebe a solução P' na Linha 7 e na Linha 8, reinicializa-se i é com o valor 0, pois P^* foi modificado. Caso $Custo(P') \geq Custo(P^*)$, incrementa-se o valor do contador na Linha 9. Por fim, retorna-se P^* na Linha 11.

Algoritmo 2: Heurística de Refinamento HR-RSP

Entrada: $G = (V, A)$, s, t, $NumIteracoesSemMelhora$
Saída: P^*, Melhor solução encontrada

```
1  início
2      P* ← AMU(G, s, t)
3      i ← 0;
4      enquanto i < NumIteracoesSemMelhora faça
5          P' ← FSHM(G, s, t, P*)
6          se CustoRobusto (P') < CustoRobusto(P*) então
7              P* ← P'
8              i ← 0;
9          senão
10             i++
11     retorna P*;
```

4. Experimentos Computacionais

Nesta seção são descritos os experimentos computacionais que comparam a heurística de refinamento desenvolvida na seção anterior com as principais heurísticas encontradas na literatura (Kasperski e Zieliński, 2006; Pérez et al., 2012; Coco et al., 2014). As heurísticas AMU, SA-RSP, BRKGA-RSP, PM-RSP e HR-RSP foram implementadas em C++ e compiladas na versão 4.6.3 do Compilador Linux/GNU. A máquina de testes possui quatro processadores Intel Xeon CPU 2.00 GHz e 16GB de memória RAM, entretanto, apenas um processador foi utilizado. Dois conjuntos de instâncias foram utilizados nos experimentos computacionais: grafos de Karasan (Karasan et al., 2001) e instâncias *grid* (Coco et al., 2014). Em ambos os conjuntos de instâncias, os intervalos foram gerados tal como sugerido em Karasan et al. (2001). O intervalo $[l_{ij}, u_{ij}]$ de custo dos arcos (i, j) foi gerado tal como descrito em (Montemanni et al., 2004). Para cada arco, é gerado um número aleatório $c_{ij} \in [1, 200]$ e seu intervalo é definido pela seleção aleatória de l_{ij} em $U[(1-d) \cdot c_{ij}, (1+d) \cdot c_{ij}]$ e u_{ij} em $U[l_{ij}+1, (1+d) \cdot c_{ij}]$, onde $d = 0.9$ para todas as instâncias deste grupo. Por fim, vale salientar que os grafos de Karasan possuem 1002 vértices e os grafos *grid* possuem entre 360 e 1000 vértices.

O experimento a seguir apresenta uma comparação entre as heurísticas AMU (Kasperski e Zieliński, 2006), SA-RSP (Pérez et al., 2012), PM-RSP (Coco et al., 2014) e BRKGA-RSP (Coco et al., 2014) e HR-RSP, proposta neste trabalho. A Tabela 1 apresenta os resultados para grafos de Karasan enquanto a Tabela 2 apresenta os resultados para grafos *grid*. O valor dos parâmetros da heurística SA-RSP utilizados neste experimento foram

os mesmos de Pérez et al. (2012). Já o valor dos parâmetros das heurísticas PM-RSP e BRKGA-RSP utilizados neste experimento foram os mesmos de Coco et al. (2014). A heurística HR-RSP termina sua execução após 1000 iterações sem que haja melhora na solução corrente. A primeira coluna mostra o nome de cada instância. As colunas 2 e 3 apresentam, respectivamente, o custo e o tempo de processamento (em segundos) de AMU. As colunas 4 e 5 apresentam, respectivamente, o custo médio de 20 execuções e o tempo de processamento (em segundos) de SA-RSP. As colunas 6 e 7 apresentam, respectivamente, o custo e o tempo de processamento (em segundos) de PM-RSP. As colunas 8 e 9 mostram o custo médio de 20 execuções e o tempo de processamento (em segundos) de BRKGA-RSP, respectivamente. Por fim, nas duas últimas colunas, são mostrados respectivamente o custo médio de 20 execuções e o tempo de processamento (em segundos) de HR-RSP. Vale salientar que em SA-RSP, BRKGA-RSP e HR-RSP o valor das sementes foi definido entre 1 e 20. Pode-se observar que a heurística de refinamento HR-RSP obteve, na média, resultados melhores que as heurísticas AMU (Kasperski e Zieliński, 2007), SA-RSP (Pérez et al., 2012) e PM-RSP (Coco et al., 2014), além de resultados um pouco piores em relação aos apresentados por BRKGA-RSP (Coco et al., 2014). Além disso, pode-se observar que em relação aos tempos de processamento, na média, HR-RSP foi um pouco mais lento que AMU e PM-RSP, apresentou um desempenho semelhante a SA-RSP e foi muito mais rápido que BRKGA-RSP. Dessa forma, pode-se verificar que apesar de apresentar resultados um pouco piores, HR-RSP apresenta uma viabilidade prática maior que BRKGA-RSP, pois os tempos de execução de HR-RSP são muito menores. Por fim, de acordo com (Coco et al., 2014), pode-se verificar que todos os algoritmos encontraram a solução ótima nas instâncias K-1000-200-0.9-a-50, K-1000-200-0.9-b-50 e K-1000-200-0.9-a-100, que é um indicativo de que estas instâncias são de fácil resolução.

	AMU		SA-RSP		PM-RSP		BRKGA-RSP (V3)		HR-RSP	
Instância	Regret (%)	t(s)	Regret (%)	t(s)	Regret (%)	t(s)	Regret (%)	t(s)	Regret (%)	t(s)
K-1000-200-0.9-a-5	64.07	0.01	64.07	4.15	**61.36**	2.80	61.49	281.74	62.09	11.90
K-1000-200-0.9-b-5	74.13	0.01	78.12	3.90	68.86	2.83	**67.58**	281.83	69.26	11.89
K-1000-200-0.9-a-10	96.94	0.01	96.94	5.18	99.78	3.50	**93.38**	380.48	93.81	11.63
K-1000-200-0.9-b-10	82.56	0.01	82.56	6.48	77.23	3.46	**76.74**	380.81	76.80	8.29
K-1000-200-0.9-a-25	66.90	0.01	66.90	18.99	67.36	5.64	**65.07**	711.62	**65.07**	9.26
K-1000-200-0.9-b-25	78.57	0.01	78.54	12.47	80.14	5.64	**77.93**	711.75	**77.93**	9.70
K-1000-200-0.9-a-50	**68.09**	0.04	**68.09**	16.51	**68.09**	8.19	**68.09**	1339.12	**68.09**	13.66
K-1000-200-0.9-b-50	**65.91**	0.03	**65.91**	16.38	**65.91**	8.17	**65.91**	1337.78	**65.91**	13.59
K-1000-200-0.9-a-100	**84.21**	0.05	**84.21**	16.92	**84.21**	13.15	**84.21**	2593.59	**84.21**	22.59
K-1000-200-0.9-b-100	100.00	0.05	90.75	16.98	**90.00**	12.98	**90.00**	2553.54	**90.00**	23.25
K-1500-200-0.9-a-5	56.68	0.01	56.90	6.46	53.31	8.56	**52.85**	809.79	54.43	27.09
K-1500-200-0.9-b-5	47.87	0.01	48.61	6.36	46.74	8.54	**45.17**	809.43	45.63	28.55
K-1500-200-0.9-a-10	60.52	0.02	60.52	9.56	58.19	10.02	**57.53**	1030.40	57.82	22.05
K-1500-200-0.9-b-10	59.20	0.02	59.38	9.47	**56.39**	10.03	**56.39**	1030.28	56.53	18.95
K-1500-200-0.9-a-25	53.88	0.04	55.84	27.80	**52.88**	14.79	**52.88**	1842.47	**52.88**	19.36
K-1500-200-0.9-b-25	89.34	0.04	94.90	30.34	**87.57**	14.74	**87.57**	1843.16	**87.57**	19.89
K-1500-200-0.9-a-50	63.46	0.05	62.41	46.32	**59.26**	20.76	**59.26**	3463.09	**59.26**	33.00
K-1500-200-0.9-b-50	79.59	0.06	84.08	56.14	**75.51**	20.62	**75.51**	3469.26	**75.51**	25.24
K-1500-200-0.9-a-100	**81.25**	0.1	82.06	50.48	**81.25**	32.85	**81.25**	6304.12	**81.25**	39.19
K-1500-200-0.9-b-100	**105.88**	0.11	107.35	38.39	**105.88**	32.67	**105.88**	6331.89	**105.88**	39.26
Média	76.36	0.03	74.41	19.96	71.99	12.00	71.23	1875.31	71.50	20.42

Tabela 1. Comparação entre a heurística de refinamento proposta com as heurísticas da literatura para grafos de Karasan

5. Conclusão

Neste trabalho, foi abordado o Caminho Mais Curto Robusto com critério *Minmax Relative Regret* (MRR-RSP). Este problema consiste em encontrar o caminho com menor custo robusto relativo entre os vértices s e t em um grafo $G = (V, A)$. Desta forma, foi proposta uma heurística de refinamento (HR-RSP) para resolver o MRR-RSP. Além disso, a heurística desenvolvida foi comparada com os trabalhos da literatura (AMU, SA-RSP,PM-RSP e BRKGA-RSP) e apresentou soluções melhores que AMU, além de um

	AMU		SA-RSP		PM-RSP		BRKGA-RSP (V3)		HR-RSP	
Instância	Regret (%)	t(s)	Regret (%)	t(s)	Regret (%)	t(s)	Regret (%)	t(s)		
G_6x60_a	44.24	0.01	49.45	4.98	43.03	0.11	42.47	20.53	**42.45**	1.09
G_6x60_b	54.09	0.01	54.09	5.18	53.83	0.10	**50.05**	20.35	50.47	1.32
G_7x70_a	61.00	0.01	61.00	9.60	59.68	0.18	**58.78**	37.67	58.92	1.78
G_7x70_b	54.95	0.01	57.42	9.42	53.96	0.16	**53.51**	37.66	53.55	1.66
G_8x80_a	49.46	0.01	49.46	16.97	48.02	0.32	**46.66**	69.24	46.74	1.95
G_8x80_b	54.31	0.01	54.31	16.61	**52.75**	0.26	**52.75**	65.50	52.82	1.64
G_9x90_a	51.38	0.01	57.83	34.78	50.77	0.47	**50.53**	111.81	50.69	3.24
G_9x90_b	56.04	0.01	56.04	28.95	52.32	0.45	**51.86**	110.10	51.87	3.22
G_10x100_a	59.22	0.01	69.39	53.94	57.30	0.73	**57.09**	182.42	57.14	4.88
G_10x100_b	49.50	0.01	54.10	65.48	47.50	0.66	**47.44**	176.70	48.83	4.65
Média	53.42	0.01	56.31	24.59	51.92	0.34	51.11	83.20	51.35	2.54

Tabela 2. Comparação entre a heurística de refinamento proposta com as heurísticas da literatura para grafos *grid*

custo-benefício melhor que SA-RSP, PM-RSP e BRKGA-RSP, pois ela obteve melhores soluções dentro da mesma grandeza de tempo, como pode ser observado tanto no SA-RSP quanto no PM-RSP, ou soluções um pouco piores em um tempo computacional muito menor, como pode ser observado BRKGA-RSP.

Como trabalhos futuros, sugere-se o estudo de algoritmos exatos e híbridos para esta versão do problema e a adaptação e aplicação da heurística proposta em outros problemas de otimização robusta no qual o critério utilizado é *minmax relative regret* e em outros critérios de solução para o RSP.

Referências

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

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

Bellman, R. (1958). On a routing problem. *Quarterly of Applied Mathematics*, 16:87–90.

Ben-Tal, A. e Nemirovski, A. (2002). Robust optimization – methodology and applications. *Mathematical Programming*, 92:453–480.

Candia-Véjar, A., Álvarez-Miranda, E., e Maculan, N. (2011). Minmax regret combinatorial optimization problems: an algorithmic perspective. *RAIRO-Operation Reserach*, 45:101–129.

Coco, A. A., Junior, J. C. A., Noronha, T. F., e Santos, A. C. (2014). A linear integer programming formulation and heuristics for the minmax relative regret robust shortest path problem. To appear in Journal of Global Optimization. DOI: 10.1007/s10898-014-0187-x.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., e Stein, C. (2009). *Introduction to Algorithms*. The MIT Press, 3rd edition.

Dijkstra, E. W. (1959). A note on two problems in connection with graphs. *Numerische Mathematik*, 1:269–271.

Duin, C. e Voss, S. (1999). The Pilot Method: A strategy for heuristic repetition with application to the Steiner problem in Graphs. *Networks*, 34:181–191.

Fortz, B. e Thorup, M. (2003). Robust optimization of OSPF/IS-IS weights. In *In Proc. International Network Optimization Conference*, pages 225–230.

Gallo, G. e Pallottino, S. (1986). Shorthest path methods: A unifying approach. *Mathematical Progamming Study*, 26:38–64.

Gonçalves, J. F. e Resende, M. G. C. (2011). Biased random-key genetic algorithms for combinatorial optimization. *Journal of Heuristics*, 17:487–525.

Gupta, S. K. e Rosenhead, J. (1968). Robustness in sequential investment decisions. *Management science*, 15:B18–B29.

Joshi, S. e Boyd, S. (2009). Sensor selection via convex optimization. *IEEE Transactions on Signal Processing*, 57:451–462.

Karasan, O. E., Yaman, H., e Ç. Pinar, M. (2001). The robust shortest path problem with interval data. Technical report, Bilkent University, Department of Industrial Engineering.

Kasperski, A., Kobylański, P., Kulej, M., e Zieliński, P. (2005). *Minimizing maximal regret in discrete optimization problems with interval data*, pages 193–208. Akademicka Oficyna Wydawnicza EXIT, Warszawa.

Kasperski, A. e Zieliński, P. (2006). An approximation algorithm for interval data minmax regret combinatorial optimization problems. *Information Processing Letters*, 97:177–180.

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

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

Kouvelis, P. e Yu, G. (1997). *Robust discrete optimization and its applications*. Kluver Academic Publishers.

Montemanni, R. e Gambardella, L. M. (2005a). A branch and bound algorithm for the robust spanning tree problem with interval data. *European Journal of Operational Research*, 161:771–779.

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

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

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

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

Shimbel, A. (1955). Structure in communication nets. In *Proceedings of the symposium on information Networks, New York.*, pages 199–203.

Spall, J. C. (2003). *Introduction to stochastic search and optimization*. Wiley.