

Algoritmo baseado em BRKGA para o Problema de Árvore Geradora Mínima Robusta

Amadeu A. Coco

Departamento de Ciência da Computação – Universidade Federal de Minas Gerais
CEP – 31270-010 – Belo Horizonte – MG – Brasil
amadeuac@dcc.ufmg.br

Thiago F. Noronha

Departamento de Ciência da Computação – Universidade Federal de Minas Gerais
CEP – 31270-010 – Belo Horizonte – MG – Brasil
tfn@dcc.ufmg.br

Andréa Cynthia Santos

ICD-LOSI, Université de Technologie de Troyes
CS 42060 – 10004 – Troyes – France
andrea.duhamel@utt.fr

RESUMO

O problema de árvore geradora mínima (MST, do inglês *Minimum Spanning Tree*) consiste em encontrar uma árvore geradora de um grafo $G = (V, E)$, onde o custo total é mínimo. Porém, diversas aplicações para o MST dependem de dados não conhecidos à priori. O Problema da Árvore Geradora Mínima Robusta generaliza o MST de forma que os dados incertos podem ser modelados como um intervalo ou como um conjunto de cenários discretos e o objetivo é encontrar a árvore com o menor custo robusto. Este trabalho propõe um algoritmo genético para o problema da Árvore Geradora Mínima Robusta Intervalar com Arrependimento Absoluto. Os resultados dos experimentos computacionais mostram que o algoritmo genético com chaves aleatórias viciadas proposto neste trabalho obteve resultados melhores do que as heurísticas da literatura.

PALAVRAS CHAVE. Árvore Geradora Mínima Robusta, Otimização Robusta, Algoritmo Genético, Metaheurísticas.

ABSTRACT

The well-known Minimum Spanning Tree problem (MST) consists in finding a spanning tree such that the total cost is minimized. However, the data of several MST applications are uncertain. The Robust Minimum Spanning Tree problem (RMST) generalizes the MST once the uncertain data can be modeled with costs as an interval or else by a set of discrete scenarios and the objective is find the tree with the minimum robust cost. This paper propose a new genetic algorithm for the minmax regret RMST. The computational experiments show that the biased random-key genetic algorithm proposed in this work produce better results than the heuristics proposed in the literature.

KEYWORDS. Robust Minimum Spaning Tree, Robust Optimization, Genetic Algorithm, Metaheuristics.

1. Introdução

Seja $G = (V, E)$ um grafo não direcionado, no qual V é um conjunto de vértices e E é um conjunto de arestas, onde cada aresta $\{i, j\} \in E$ está associada a um peso $c_{ij} \in \mathbb{R}$. Dada uma árvore como o subgrafo S de G formado por todos os vértices de G e um subconjunto $E' \subseteq E$ de arestas, onde $|E'| = |V| - 1$, o Problema da Árvore Geradora Mínima (MST, do inglês *Minimum Spanning Tree*) consiste em encontrar uma árvore cujo somatório dos custos associados as arestas seja mínimo. O MST é estudado desde a década de 20 (Boruvka, 1926) e sua complexidade computacional é $\theta(|E| \cdot \log |V|)$ (Cormen et al., 2009). Os algoritmos mais utilizados para a resolução do MST são os algoritmos de Kruskal (Kruskal, 1956) e Prim (Prim, 1957). O primeiro possui complexidade $\theta(|E| \cdot \log |V|)$ e o segundo $\theta(|E| + |V| \cdot \log |V|)$.

MST modela diversos problemas de interesse prático e teórico (Bazlamaççi e Hindi, 2001). Entretanto, neste problema cada aresta $\{i, j\} \in E$ está associada a um custo fixo, fato que impede que vários problemas encontrados no mundo real possam ser tratados e fornecer boas soluções quando modelados via MST, pois frequentemente os dados deste tipo de problema estão sujeitos a incertezas.

O problema da árvore geradora mínima robusta (RMST) é a versão de MST no qual existe incerteza nos valores das arestas $\{i, j\} \in E$. As principais formas de modelar esta incerteza encontradas na literatura do RMST são: *modelagem por intervalos* (Yaman et al., 2001; Kasperski e Zieliński, 2006; Montemanni e Gambardella, 2005) e a *modelagem por cenários discretos* (Aissi et al., 2009; Kouvelis e Yu, 1997). Na modelagem por intervalos, cada aresta $\{i, j\} \in E$ pode assumir qualquer valor real no intervalo $[l_{ij}, u_{ij}]$, independentemente dos valores assumidos pelas outras arestas em E , desta forma, infinitos cenários são considerados. Já na modelagem por cenários discretos, um conjunto finito R de cenários é definido, onde cada cenário $r \in R$ consiste em uma atribuição única de c_{ij}^r para o custo de cada aresta $\{i, j\} \in E$.

Os principais critérios de avaliação utilizados para RMST são (i) *absoluto* (Kozina e Perepelitsa, 1994; Kouvelis e Yu, 1997), (ii) *arrepentimento absoluto* (Averbakh e Lebedev, 2004; Kasperski e Zieliński, 2006; Kozina e Perepelitsa, 1994; Yaman et al., 2001) e (iii) *arrepentimento relativo* (Averbakh, 2005). No critério *absoluto* a solução ótima é dada pela árvore que apresenta a melhor solução no pior caso. No critério de avaliação *arrepentimento absoluto*, dado $T \subseteq E$ uma árvore em G , o *desvio robusto* de T no cenário r é definido como a diferença entre (i) o custo da árvore T no cenário r e (ii) o custo da árvore geradora mínima no cenário r , ou seja, o desvio robusto de T em r é o quanto o custo de T seria pior que o custo da melhor árvore possível caso o cenário r ocorresse. Denota-se por r^T o cenário onde observa-se o pior desvio robusto para a árvore T onde r^T é gerado fixando-se $c_{ij}^r = u_{ij}, \forall \{i, j\} \in T$, onde u_{ij} é o valor máximo da aresta, e $c_{ij}^r = l_{ij}, \forall \{i, j\} \in E \setminus T$, onde l_{ij} é o valor mínimo da aresta (Yaman et al., 2001). O *custo robusto* de uma árvore T é definido como o *desvio robusto* de T em r^T , sendo assim, o problema consiste em encontrar a árvore com o menor *custo robusto* dentre todas as árvores geradoras do grafo. Já no critério de avaliação *arrepentimento relativo*, seja $T \subseteq E$ uma árvore em G , o *desvio robusto relativo* de T no cenário r é definido por $\frac{a-b}{b}$, onde a denota o custo da árvore T no cenário r e b denota o custo da árvore geradora mínima no cenário r^T , ou seja, o *desvio robusto relativo* de T em r é o quanto T seria pior, relativamente a melhor árvore possível caso o cenário r ocorresse. O *custo robusto relativo* de uma árvore T é definido como o desvio robusto relativo de T em r^T , sendo assim, o problema consiste em

encontrar a árvore com o menor *custo robusto relativo* dentre todas as árvores geradoras de G (Yaman et al., 2001).

As seis versões mais estudadas na literatura de RMST são as seguintes: RMST intervalar Absoluto (RMST-IA) (Kouvelis e Yu, 1997), RMST intervalar com arrependimento absoluto (RMST-IR) (Karasan et al., 2001; Montemanni et al., 2004), RMST intervalar com arrependimento relativo (RMST-IRR) (Averbakh, 2005), RMST discreto Absoluto (RMST-DA) (Kouvelis e Yu, 1997), RMST discreto com arrependimento absoluto (RMST-DR) (Averbakh e Lebedev, 2004), RMST discreto com arrependimento relativo (RMST-DRR) (Averbakh, 2005). As características destes problemas são resumidas na Tabela 1. Na primeira coluna é mostrado o tipo de modelagem de cenário. Nas colunas seguintes são mostrados os nomes da versão de acordo com a estratégia de solução utilizada.

Modelagem	Critério Absoluto	Critério Arrependimento Absoluto	Critério Arrependimento Relativo
Contínuo	RMST-IA	RMST-IR	RMST-IRR
Discreto	RMST-DA	RMST-DR	RMST-DRR

Tabela 1. Diversas versões conhecidas para o Problema de Árvore Geradora Mínima Robusta (RMST) mostradas na literatura.

RMST-IA é polinomial (Kouvelis e Yu, 1997) e pode ser resolvida com a utilização do algoritmo de Kruskal. Porém, todas as outras versões do RMST são NP-Difíceis (Kouvelis e Yu, 1997). Este trabalho foca principalmente no RMST-IR tal como definido em Yaman et al. (2001); Montemanni e Gambardella (2005), pois não existe nenhuma meta-heurística para esta versão do problema e os algoritmos exatos disponíveis não conseguem resolver instâncias com mais de 40 vértices.

A Figura 1 apresenta um exemplo de RMST-IR para um grafo com quatro vértices e seis arestas. Cada aresta $\{i, j\} \in E$ está associada a um intervalo $[l_{ij}, u_{ij}]$. Por exemplo, o custo da aresta $\{0, 2\}$ pode assumir qualquer valor real entre $l_{02} = 4$ e $u_{02} = 6$. A árvore com menor custo robusto no grafo da Figura 1 é a árvore $T^* = \{\{0, 1\}, \{0, 3\}, \{2, 3\}\}$. O custo de T^* no cenário r^{T^*} , onde r^{T^*} é o cenário induzido pela árvore T^* , ou seja, o cenário onde a aresta $\{i, j\}$ é fixada em u_{ij} caso $\{i, j\} \in T^*$ e fixado em l_{ij} caso contrário, é $u_{01} + u_{03} + u_{23} = 2 + 2 + 7 = 11$. O custo da árvore geradora mínima é $T' = \{\{0, 1\}, \{0, 2\}, \{0, 3\}\}$ no cenário r^{T^*} é $u_{01} + l_{02} + u_{03} = 2 + 2 + 4 = 8$. Portanto, o custo robusto de T^* é $11 - 8 = 3$.

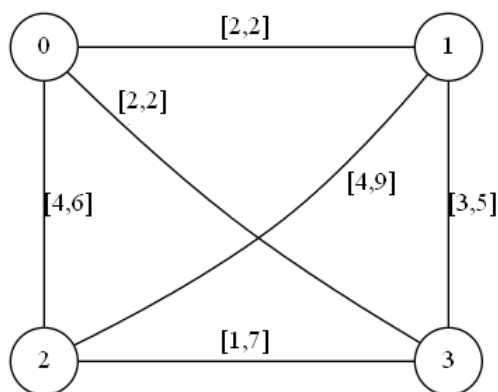


Figura 1. Exemplo de grafo para o problema de árvore geradora mínima robusta.

O restante deste trabalho está organizado da seguinte forma. Na Seção 2, são apresentadas as referências bibliográficas para o RMST. Na Seção 3, é mostrada a formulação de Yaman et al. (2001) para o RMST-IR. Na Seção 4, é apresentada uma heurística baseada em algoritmo genético com chaves aleatórias viciadas (BRKGA, do inglês *Biased Random Key Genetic Algorithm* para o RMST-IR. Na Seção 5, são apresentados os experimentos computacionais. Por fim, na Seção 6 são apresentadas as conclusões do trabalho.

2. Trabalhos Relacionados

Otimização Robusta é estudada 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, tais como o RST-IR, é NP-Difícil.

Kozina e Perepelitsa (1994) propuseram o RMST. Nesse trabalho, os autores propuseram a transformação do RMST-IR em um problema de otimização multiobjetivo onde a solução ótima do problema é dada por um conjunto de Pareto. Por fim, os autores concluíram que a complexidade do problema de otimização multi-objetivo é não polinomial.

Karasan et al. (2001) introduziram o conceito de arco fraco (do inglês, *weak-arc*) para problemas de otimização robusta que envolvem grafos, tais como o RMST. Arco Fraco é aquele que jamais estará em uma solução ótima de um problema de otimização robusta.

Yaman et al. (2001) apresentaram uma formulação baseada em programação inteira mista para RMST-IR. Por fim, os autores utilizaram o conceito de Arco Fraco (Karasan et al., 2001) em técnicas de pré-processamento para o RMST-IR e demonstraram, através de experimentos, que tais técnicas reduzem o tempo de processamento da implementação da formulação no *solver* CPLEX em cerca de 50%. A formulação de Yaman et al. (2001) é apresentada na Seção 3.

Aron e Hentenryck (2004) provaram que o RST na modelagem intervalar é NP-difícil utilizando a conjectura de Kouvelis e Yu (1997). Os autores também mostraram que o problema permanece NP-Difícil em grafos completos ou quando os intervalos de todas as arestas estão entre $[0, 1]$. Os resultados demonstrados por Aron e Hentenryck (2004) indicam que a dificuldade do RST deriva-se da topologia do grafo e da estrutura dos custos dos intervalos, o que sugere novas direções no desenvolvimento de algoritmos.

Averbakh e Lebedev (2004) provaram que problemas de otimização robusta com modelagem intervalar e critério de solução arrependimento absoluto são NP-difíceis quando o custo das arestas varia somente em $\{0, 1\}$. Além disso, os autores provaram que esta classe de problemas são polinomiais no caso em que o número de parâmetros com valores incertos é limitado pelo logaritmo de uma função polinomial em relação ao número total de parâmetros com incertezas.

Montemanni e Gambardella (2005) propuseram um algoritmo baseado na técnica *branch-and-bound* (B&B) para o RMST-IR. O B&B apresentado em Montemanni e Gambardella (2005) utiliza as técnicas de pré-processamento de Yaman et al. (2001), junta-

mente com regras de poda da árvore de *branching* propostas pelos autores. Segundo os autores, o algoritmo desenvolvido por eles é aproximadamente 210 vezes mais rápido que os algoritmos propostos em (Aron e Hentenryck, 2004; Yaman et al., 2001).

Averbakh (2005) estendeu o trabalho de Kouvelis e Yu (1997) e trabalhou com a versão robusta de problemas de otimização combinatória, onde a modelagem dos cenários é intervalar e o critério de avaliação da solução é dada por arrependimento relativo. 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. Nesse mesmo trabalho, também, foi provado que a complexidade desta versão de problemas robustos é Fortemente NP-Difícil, ou seja, não existem algoritmos aproximativos para esta versão. Por fim, foi apresentado o problema de sequenciamento robusto em uma máquina, que na versão clássica é polinomial e na versão robusta com arrependimento relativo é NP-Difícil.

Kasperski e Zieliński (2006) propuseram dois algoritmos aproximativos com fator de aproximação 2 para problemas de otimização robusta com modelagem intervalar, dentre os quais o RMST-IR. A primeira, chamada aqui de U-KZ, recebe como entrada um grafo $G = (V, E)$ e os limites inferior l_{ij} e superior u_{ij} para o custo de cada aresta $\{i, j\} \in E$. Em seguida, fixa-se o cenário r^u , onde o custo de todas as arestas estão em seu respectivo valor máximo, ou seja, $c_{ij}^u = u_{ij}, \forall \{i, j\} \in E$. Em seguida, calcula-se a árvore geradora mínima T^u no cenário r^u , utilizando o algoritmo de Kruskal. Por fim, o custo robusto de T^u é calculado e esta solução é retornada. A complexidade assintótica desta heurística no pior caso é $O(|E| \cdot \log |V|)$, assumindo-se que $|V| \cdot \log |V| \geq |E|$. A segunda heurística, chamada aqui de M-KZ, recebe os mesmos parâmetros de entrada que U-KZ. Primeiramente, fixa-se o cenário r^m , onde o custo de todas as arestas estão em seu respectivo valor médio, ou seja, $c_{ij}^m = (l_{ij} + u_{ij})/2, \forall \{i, j\} \in E$. Em seguida, calcula-se a árvore geradora mínima T^m no cenário r^m , utilizando algoritmo de Kruskal. Por fim, o custo robusto de T^m é calculado e esta solução é retornada. A complexidade desta heurística no pior caso é $O(|E| \cdot \log |V|)$, assumindo-se que $|V| \cdot \log |V| \geq |E|$. Vale salientar que, ambas as heurísticas são 2-aproximadas para o RMST-IR.

Candia-Véjar et al. (2011) trabalharam com complexidade de algoritmos de problemas de otimização robusta com modelagem intervalar. Os autores partiram de definições e propriedades comuns aos problemas da classe propostas em Averbakh e Lebedev (2004). A seguir foram apresentadas as complexidades da versão *minmax* com arrependimento absoluto dos problemas de p-medianas, caminho mais curto, árvore geradora mínima, assinalamento, programação linear e caixeiro viajante. Em seguida, os autores apresentaram as formulações matemáticas destes problemas e por fim foram descritos algoritmos aproximativos e suas respectivas provas.

Kasperski e Zieliński (2011) modelaram o RST por cenários discretos nas versões absoluta e com arrependimento absoluto. Nesse trabalho, os autores estudam a complexidade e a existência de algoritmos aproximativos para as versões do RST citadas. Por fim, os autores propuseram um algoritmo aproximativo randomizado, com base na formulação do problema onde o fator aproximativo é igual a $\mathcal{O}(\log^2 n)$

Coco et al. (2012) desenvolveram uma heurística baseada em Algoritmos Genéticos com Chaves Aleatórias Viciadas (Gonçalves et al., 2009; Gonçalves e Resende, 2010) para o Problema de Caminho Mais Curto Robusto Intervalar com Arrependimento Relativo. O

BRKGA proposto por Coco et al. (2012) foi a primeira metaheurística do gênero proposta para um problema de otimização robusta e obteve resultados melhores que as heurísticas propostas por (Kasperski e Zieliński, 2006).

Até onde sabemos, não existem na literatura trabalhos que propõem heurísticas baseadas em algoritmos genéticos para o RMST-IR. Sendo assim, este trabalho propõe o primeiro algoritmo genético para o problema com o intuito de encontrar soluções de qualidade para instâncias maiores de RMST-IR em tempo polinomial.

3. Formulação de Yaman et al. (2001)

Uma formulação por programação linear inteira mista para o RMST-IR foi proposta por Yaman et al. (2001). Existem cinco conjuntos de variáveis na formulação. Para cada aresta $\{i, j\} \in E$, são definidas variáveis $x_{ij} \in \{0, 1\}$ de forma que $x_{ij} = 1$ se a aresta $\{i, j\}$ faz parte da árvore geradora mínima robusta, e $y_{ij} = 0$ em caso contrário. As variáveis $f_{ij} \in \mathbb{R}, \forall (i, j) \in A$, onde $A = \{(i, j), (j, i) | \{i, j\} \in E\}$, representam o fluxo que passa por um arco entre o vértice i e o vértice j . As variáveis restantes $\alpha_i^k \in \mathbb{R}, \forall i, k \in V \setminus \{0\}$, $\sigma_{ij}^k \in \mathbb{R}, \forall (i, j) \in A$ e $k \in V \setminus \{0\}$ e $\mu \in \mathbb{R}$ definem o dual de uma formulação com fluxo multi-produto para o MST, onde os fluxos são originados no nó 0 (Yaman et al., 2001). Vale salientar que as variáveis duais são utilizadas na formulação para calcular a árvore geradora mínima induzida pelo cenário r^T . A formulação é definida pelas equações (1)-(13).

$$\min \sum_{(i,j) \in A} u_{ij} x_{ij} - \sum_{k \in V \setminus \{0\}} (\alpha_k^k - \alpha_0^k) - (|V| - 1)\mu \quad (1)$$

Sujeito a:

$$\sigma_{ij}^k \geq \alpha_j^k - \alpha_i^k \quad \forall (i, j) \in A, \forall k \in V \setminus \{0\} \quad (2)$$

$$\sum_{k \in V \setminus \{0\}} \sigma_{ij}^k + \mu \leq l_{ij} + (u_{ij} - l_{ij})x_{ij} \quad \forall \{i, j\} \in E \quad (3)$$

$$\sum_{k \in V \setminus \{0\}} \sigma_{ji}^k + \mu \leq l_{ij} + (u_{ij} - l_{ij})x_{ij} \quad \forall \{i, j\} \in E \quad (4)$$

$$\sum_{(i,j) \in A} f_{ij} - \sum_{(j,i) \in A} f_{ji} = \begin{cases} |V| - 1; & \text{se } i = 0. \\ -1; & \forall i \in V \setminus \{0\} \end{cases} \quad (5)$$

$$f_{ij} \leq (|V| - 1)x_{ij} \quad \forall \{i, j\} \in E, \quad (6)$$

$$f_{ji} \leq (|V| - 1)x_{ij} \quad \forall \{i, j\} \in E, \quad (7)$$

$$\sum_{\{i,j\} \in E} x_{ij} = |V| - 1 \quad (8)$$

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

$$f_{ij} \geq 0 \quad \forall (i, j) \in A, \quad (10)$$

$$\sigma_{ij}^k \geq 0 \quad \forall (i, j) \in A, \forall k \in V \setminus \{0\} \quad (11)$$

$$\alpha_i^k \geq 0 \quad \forall i \in V, \forall k \in V \setminus \{0\} \quad (12)$$

$$\mu \text{ irrestrito} \quad (13)$$

A função objetivo (1) minimiza o custo robusto da árvore geradora mínima definida pelas variáveis x . As equações (2)-(4) definem o dual do problema de árvore geradora mínima clássica em um cenário induzido pelas variáveis x (Yaman et al., 2001), além disso elas garantem a ligação entre as variáveis x e as variáveis σ e α . As equações (5)-(8) definem as restrições de conservação de fluxo da árvore. As restrições (9)-(13) definem os domínios das variáveis x, f, α, σ e μ .

O modelo definido em (Yaman et al., 2001) foi resolvido de forma eficiente pelo *solver* comercial CPLEX para grafos completos com até 25 vértices em 7 horas e meia.

4. Biased Random Key Genetic Algorithm

Uma extensão das heurísticas de Kasperski e Zieliński (2006), apresentada em Coco et al. (2012) é denominada *Fixed Scenario Heuristic* (FSH). Ela recebe como entrada um grafo $G = (V, E)$ e os limites inferior l_{ij} e superior u_{ij} para o custo de cada aresta $\{i, j\} \in E$. Além dos dados do problema, FSH também recebe como parâmetro um possível cenário r' , ou seja, uma fixação de valores para o custo $c'_{ij} \in [l_{ij}, u_{ij}]$ para cada aresta $(i, j) \in A$. FSH inicialmente calcula a árvore geradora mínima T' no cenário r' , utilizando algoritmo de Kruskal. Em seguida, o custo robusto de T' é calculado e T' é retornado como solução da heurística. A complexidade assintótica, no pior caso, desta heurística é $\theta(|E| \cdot \log |V|)$, assumindo-se que $|V| \cdot \log |V| \geq |E|$. Para cada cenário r' passado como parâmetro para FSH, esta pode retornar uma solução diferente para RMST-IR. Portanto, a heurística FSH explora o espaço de cenários. Vale salientar que nem todas as soluções são induzidas por algum cenário.

Para se explorar o espaço de cenários de forma eficiente, é proposta uma heurística baseada em Algoritmos Genéticos com Chaves Aleatórias (BRKGA, do inglês *Biased Random-Key Genetic Algorithm*) (Gonçalves e Resende, 2010). A motivação para aplicar esta metaheurística para RMST-IR é a sua aplicação bem sucedida para vários outros problemas de otimização combinatória como problemas de trânsito (Buriol et al., 2010; Reis et al., 2011), problemas de sequenciamento (Gonçalves et al., 2009, 2010), Problema do Carregamento de Contêineres (Gonçalves e Resende, 2012), Problema de Roteamento e Atribuição do Comprimento de Onda em Redes (Noronha et al., 2011) e Problema de Caminho Mais Curto Robusto (Coco et al., 2012).

Em BRKGAs, os cromossomos consistem em vetores de números reais com valores no intervalo $[0, 1]$. Cada elemento do vetor é chamado de *chave* e seu valor é concebido aleatoriamente na população inicial. A adaptabilidade do cromossomo é definida pelo custo da solução fornecida por uma heurística de decodificação. Além dos dados da instância, esta última também recebe como um de seus dados de entrada o vetor de chaves do cromossomo e devolve uma solução viável para o problema. O custo desta solução é usado como o valor de adaptabilidade do cromossomo.

Na heurística BRKGA-RMST, cada cromossomo da população representa um possível cenário. Ou seja, um cromossomo p contém uma chave $k_{ij}^p \in [0, 1]$ para cada aresta $\{i, j\} \in E$. Desta forma, dado um cromossomo p e um cenário r^p induzido pelo cromossomo p , o custo de cada aresta $\{i, j\} \in E$ em r^p é fixado em $c_{ij}^p = l_{ij} + (u_{ij} - l_{ij}) \cdot k_{ij}^p$. A adaptabilidade de p é definida como o custo robusto da árvore retornada pela heurística FSH quando esta é executada a partir do cenário r^p . Sendo assim, a complexidade assintótica no pior caso de avaliar um cromossomo é $\theta(|E| \cdot \log |V|)$.

O algoritmo de recombinação ou cruzamento utilizado neste algoritmo genético foi proposto em (Spears e DeJong, 1991). Dados dois cromossomos c_1 e c_2 , um novo cromossomo c_{new} é gerado da seguinte forma. O valor de cada uma das chaves do novo cromossomo c_{new} é herdado aleatoriamente da chave correspondente em c_1 com probabilidade p_{rec} e da chave correspondente em c_2 com probabilidade $1 - p_{rec}$.

A Figura 2 apresenta um exemplo do cruzamento de dois cromossomos c_1 e c_2 . As chaves herdadas pelo cromossomo c_{new} estão em destaque. Para definir o valor de cada chave do cromossomo c_{new} é gerado um número aleatório. Neste exemplo, são gerados cinco números aleatórios $\omega = [0, 72; 0, 18; 0, 93; 0, 48; 0, 04]$. Quando $\omega_i \leq 0, 5$, a chave i é herdada do cromossomo c_1 , caso contrário, a chave i é herdada do cromossomo c_2 .

$$\begin{aligned} C_1 &= [0,36 \mathbf{0,42} \ 0,12 \mathbf{0,78} \ \mathbf{0,04}] \\ C_2 &= [\mathbf{0,27} \ 0,81 \ \mathbf{0,99} \ 0,14 \ 0,17] \\ C_{new} &= [0,27 \ 0,42 \ 0,99 \ 0,78 \ 0,04] \end{aligned}$$

Figura 2. Cruzamento de dois cromossomos no algoritmo genético.

Este algoritmo genético não faz uso de operadores tradicionais de mutação. Ao invés disso, a cada geração novos cromossomos são gerados aleatoriamente, tal como na população inicial.

A Figura 3 mostra a evolução da população do algoritmo genético entre duas gerações. A cada nova geração a população corrente é ordenada e particionada em três conjuntos: *TOP*, *MID* e *BOT*. Os cromossomos mais aptos são mantidos no conjunto *TOP*, os menos aptos no conjunto *BOT* e os restantes no conjunto *MID*. Os cromossomos no conjunto *TOP* são copiados para a próxima geração e aqueles em *BOT* são substituídos por novos cromossomos gerados aleatoriamente. Os cromossomos no conjunto *MID* são substituídos por novos cromossomos obtidos pelo cruzamento de dois cromossomos, sendo que o primeiro é escolhido aleatoriamente do conjunto *TOP* e o segundo do conjunto $REST = MID \cup BOT$. O tamanho da população e a cardinalidade dos conjuntos *TOP*, *MID* e *BOT* são parâmetros que devem ser ajustados.

O BRKGA-RMST é descrito no Algoritmo 1. Na Linha 2, a população inicial é gerada utilizando-se vetores com chaves aleatórias. O laço das linhas 3-14 é executado enquanto a condição de parada não é satisfeita. Na Linha 4, ordena-se a população de acordo com o custo robusto relativo. Na Linha 5, divide-se a população em três grupos, de acordo com o custo robusto relativo, chamados de *TOP*, *MID* e *BOT*. No laço das linhas 6 e 7, copia-se o grupo *TOP* para a próxima geração. No laço das linhas 8-11, realiza-se o cruzamento, onde os cromossomos da porção *MID* da próxima geração são gerados. No laço das linhas 12-14 são gerados, de forma aleatória, os novos elementos da porção *BOT* da próxima geração. Por fim, na linha 15, retorna-se a solução com o melhor custo robusto.

5. Experimentos Computacionais

Nesta seção, os experimentos computacionais que comparam a heurística desenvolvida na seção anterior com as heurísticas de Kasperski e Zieliński (2006) e o algoritmo de *branch-and-bound* obtido pela programação da formulação de Yaman et al. (2001) no CPLEX são descritos. As heurísticas M-KZ, U-KZ e BRKGA-RMST foram implementadas em C++ e compiladas na versão 4.3.3 do Compilador Linux/GNU. O modelo de

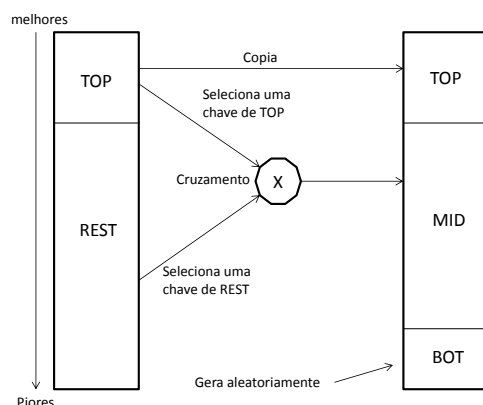


Figura 3. Transição entre gerações sucessivas no Algoritmo Genético com Chaves Aleatórias.

Yaman et al. (2001) foi executado no *solver* IBM Ilog CPLEX Versão 12.5. Os parâmetros do BRKGA-RMST foram definidos de forma semelhante a (Coco et al., 2012), onde (i) o tamanho da população p foi fixado em 100 cromossomos, (ii) os conjuntos $|TOP|$ e $|BOT|$ foram definidos como $|TOP| = 0,2 \times p$ e $|BOT| = 0,1 \times p$, (iii) o valor de p_{rec} foi fixado em 0,5 e, (iv) a execução do BRKGA foi encerrada após $|V|^2$ gerações. A máquina de testes possui quatro processadores Intel Xeon CPU 2.00 GHz e 16GB de memória RAM, entretanto, apenas um processador foi utilizado. As instâncias utilizadas nos experimentos computacionais consistem em grafos completos semelhantes aos propostos por Yaman et al. (2001). O intervalo $[l_{ij}, u_{ij}]$ de custo das arestas $\{i, j\} \in E$ foi gerado da seguinte forma. Para cada aresta, é gerado um número aleatório $R \in \{1, 2, 3, 4\}$. Se $R = 1$, o intervalo da aresta é definido pela seleção aleatória de l_{ij} em $U[1, 5]$ e u_{ij} em $U[l_{ij}, 5]$. Se $R = 2$, o intervalo da aresta é definido pela seleção aleatória de l_{ij} em $U[10, 15]$ e u_{ij} em $U[l_{ij}, 15]$. Se $R = 3$, o intervalo da aresta é definido pela seleção aleatória de l_{ij} em $U[21, 25]$ e u_{ij} em $U[l_{ij}, 25]$. Por fim, se $R = 4$, o intervalo da aresta é definido pela seleção aleatória de l_{ij} em $U[31, 35]$ e u_{ij} em $U[l_{ij}, 35]$. Por fim, vale salientar que se $R_1 \neq R_2$ a interseção dos intervalos de duas arestas é vazio.

O experimento a seguir apresenta uma comparação entre a formulação de Yaman et al. (2001) e as heurísticas M-KZ (Kasperski e Zieliński, 2006), U-KZ (Kasperski e Zieliński, 2006) e BRKGA-RMST. A Tabela 2 apresenta os resultados dos experimentos desta comparação. Para cada instância, o tempo de execução do CPLEX foi fixado em 4 horas (14400 segundos). A primeira coluna mostra o nome de cada instância. A Coluna 2 mostra o valor do limite inferior (LB, do inglês *Lower Bound*) e a Coluna 3 exibe o valor do limite superior (UB, do inglês *Upper Bound*) obtido pelo CPLEX. A Coluna 4 apresenta o número de nós da árvore de *Branch-and-Bound* (BB) avaliados, enquanto a Coluna 5 mostra o *gap* de cada instância, dado por $\frac{UB-LB}{UB}$. As colunas 6 e 7 mostram, respectivamente, o custo e o tempo de processamento da heurística M-Kasperski. Já as colunas 8 e 9 mostram, respectivamente, o custo e o tempo de processamento da heurística U-Kasperski. Por fim, as três últimas colunas mostram o custo médio (após 20 execuções),

Algoritmo 1: Random Key Genetic Algorithm Robust Minimum Spanning Tree (BRKGA-RMST).

Entrada: $G = (V, E), l_{ij}, u_{ij}$
Saída: T^* , Melhor solução encontrada

- 1 **início**
- 2 Gerar uma população inicial de vetores de chaves aleatórias;
- 3 **enquanto** (*Condição de Parada não satisfeita*) **faça**
- 4 Ordenar a população em ordem crescente de custo robusto;
- 5 Dividir a população nos grupos TOP, MID e BOT;
- 6 **para cada elemento da População TOP faça**
- 7 Copiar para a próxima geração;
- 8 **para** i de 1 até $|MID|$ **faça**
- 9 Selecione aleatoriamente um cromossomo c^t de TOP e outro c^r de
 REST = MID \cup BOT;
- 10 $c^f \leftarrow$ Cruzamento (c^t, c^r)
- 11 Copiar c^f para a próxima geração
- 12 **para** i de 1 até $|BOT|$ **faça**
- 13 Gerar um novo cromossomo c^a aleatoriamente
- 14 Copiar c^a para a próxima geração
- 15 **retorna** melhor solução de TOP

o desvio padrão (em %) e o tempo de processamento (em segundos) de BRKGA-RMST, respectivamente. O valor das sementes de BRKGA-RMST foi definido entre 1 e 20. A menor solução encontrada para cada instância está destacada em negrito. Pode-se observar que a formulação de Yaman et al. (2001) encontrou a solução ótima nos grafos com 20 vértices. Já nas instâncias com 30 e 40 vértices o CPLEX encontrou soluções viáveis, porém não chegou a otimalidade, enquanto nas instâncias de 50 vértices o CPLEX não encontrou soluções viáveis. Já em relação as heurísticas, pode-se observar que BRKGA-RMST encontrou soluções melhores que M-KZ e U-KZ em todas as instâncias, porém em tempo de execução maior. O tempo de execução de BRKGA-RMST é maior que o de M-KZ e U-KZ, porque o BRKGA-RMST avalia $|V|^2 \cdot p$ árvores do grafo, enquanto as heurísticas de Kasperski e Zieliński (2006) avaliam apenas uma árvore. Por fim, vale salientar que BRKGA-RMST encontrou o custo robusto ótimo nas instâncias de 20 vértices.

Instância	CPLEX (4h)				M-KZ		U-KZ		BRKGA-RMST		
	LB	UB	Nós	GAP (%)	UB	T (s)	UB	T (s)	UB	STDEV (%)	T (s)
grafo20a	978,00	978,00	19967	0,00%	1104,00	0,01	1461,00	0,01	978,00	0,00	6,01
grafo20b	853,00	853,00	16752	0,00%	954,00	0,01	1008,00	0,01	853,00	0,00	6,02
grafo20c	1099,00	1099,00	10049	0,00%	1223,00	0,01	1565,00	0,01	1099,00	0,00	6,02
grafo30a	1914,19	2919,00	25263	34,42%	3220,00	0,01	3356,00	0,01	2904,00	0,00	30,44
grafo30b	2250,49	3131,00	83555	28,12%	3533,00	0,01	4032,00	0,01	3131,00	0,00	30,38
grafo30c	1847,56	3544,00	22315	47,87%	3797,00	0,01	3957,00	0,01	3529,60	4,92	30,38
grafo40a	1915,46	4913,00	2119	61,01%	4607,00	0,01	4840,00	0,01	4307,00	6,16	99,55
grafo40b	2075,923	5265,00	2494	60,57%	5021,00	0,01	5325,00	0,01	4767,00	29,73	99,97
grafo40c	1283,3579	4858,00	2352	73,58%	4287,00	0,01	4494,00	0,01	4018,00	0,00	99,95
grafo50a	-2552,7599	5.53830e+09	1	100,00%	5029,00	0,01	4888,00	0,01	4668,35	1,57	255,96
grafo50b	-1.11826e+08	5.47944e+09	1	100,00%	5373,00	0,01	5915,00	0,01	5015,75	5,76	255,88
grafo50c	-1.14093e+08	5.59058e+09	1	100,00%	6099,00	0,01	6184,00	0,01	5690,00	0,00	256,65

Tabela 2. Comparação entre o modelo de Yaman et al. (2001), as heurísticas de (Kasperski e Zieliński, 2006) e BRKGA-RMST

6. Conclusão

Neste trabalho, foi abordado o Problema de Árvore Gerada Mínima Intervalar com Arrependimento Absoluto (RMST-IR). Este problema consiste em encontrar a árvore com menor custo robusto em um grafo $G = (V, E)$. Desta forma, foi proposto um algoritmo genético (BRKGA-RMST) para resolver o RMST-IR. Além disso, o algoritmo desenvolvido foi comparado com a formulação de Yaman et al. (2001) e as heurísticas de Kasperski e Zieliński (2006) e apresentou resultados equivalentes ou melhores em relação aos algoritmos da literatura.

Como trabalhos futuros, sugere-se o estudo de heurísticas construtivas, buscas locais e algoritmos híbridos para esta versão do problema, além da derivação de desigualdades válidas para a formulação de Yaman et al. (2001).

Referências

- Boruvka, O.** (1926). About a certain minimal problem. *German Summary*, 3:37–58.
- Kruskal, J. B.** (1956). On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical Society*, 7:48–50.
- Prim, R. C.** (1957). Shortest connection networks and some generalizations. *Bell System Technical Journal*, 36:1389–1401.
- Gupta, S. K. e Rosenhead, J.** (1968). Robustness in sequential investment decisions. *Management science*, 15:B18–B29.
- Rosenhead, M. J., Elton, M., e Gupta, S. K.** (1972). Robustness and optimality as criteria for strategic decisions. *Operational Research Quarterly*, 23:413–430.
- Spears, W. e DeJong, K.** (1991). On the virtues of parameterized uniform crossover. In Belew, R. e Booker, L., editors, *Proceedings of the Fourth International Conference on Genetic Algorithms*, pages 230–236, San Mateo, Italy.
- Kozina, G. L. e Perepelitsa, V. A.** (1994). Interval spanning trees problem: Solvability and computational complexity. *Interval Computations*, pages 42–50.
- Kouvelis, P. e Yu, G.** (1997). *Robust discrete optimization and its applications*. Kluwer Academic Publishers.
- Bazlamaççi, C. F. e Hindi, K. S.** (2001). Minimum-weight spanning tree algorithms a survey and empirical study. *Computers & Operations Research*, 28:767–85.
- 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.
- Yaman, H., Karasan, O. E., e Ç. Pinar, M.** (2001). The robust spanning tree problem with interval data. *Operations Research Letters*, 29:31–40.
- 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 Research Letters*, 32:225–232.
- Averbakh, I. e Lebedev, V.** (2004). Interval data minmax regret network optimization problems. *Discrete Applied Mathematics*, 138:289–301.
- Aron, I. D. e Hentenryck, P. V.** (2004). On the complexity of the robust spanning tree problem with interval data. *Operations Research Letters*, 32:36–40.
- Montemanni, R. e Gambardella, L. M.** (2005). A branch and bound algorithm for the robust spanning tree problem with interval data. *European Journal of Operational Research*, 161:771–779.

- Averbakh, I.** (2005). Computing and minimizing the relative regret in combinatorial optimization with interval data. *Discrete Optimization*, 2:273–287.
- 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.** (2006). The robust shortest path problem in series–parallel multidigraphs with interval data. *Operations Reserach Letters*, 34:69–76.
- Gonçalves, J. F., Mendes, J. J. M., e Resende, M. G. C.** (2009). A random key based genetic algorithm for the resource constrained project scheduling problems. *Computers & Operations Research*, 36:92–109.
- Cormen, T. H., Leiserson, C. E., Rivest, R. L., e Stein, C.** (2009). *Introduction to Algorithms*. The MIT Press, 3rd edition.
- 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.
- Buriol, L. S., Hirsch, M. J., Querido, T., Pardalos, P. M., Resende, M. G. C., e Ritt, M.** (2010). A biased random-key genetic algorithm for road congestion minimization. *Optimization Letters*, 4:619–633.
- Gonçalves, J. F., Resende, M. G. C., e Mendes, J. J. M.** (2010). A biased random-key genetic algorithm with forward-backward improvement for the resource constrained project scheduling problem. *Journal of Heuristics*, 5:467–486.
- Gonçalves, J. F. e Resende, M. G. C.** (2010). Biased random-key genetic algorithms for combinatorial optimization. *Journal of Heuristics*, 17:487–525.
- Reis, R., Ritt, M., Buriol, L. S., e Resende, M. G. C.** (2011). A biased random-key genetic algorithm for OSPF and DEFT routing to minimize network congestion. *International Transactions in Operation Reserach*, 18:401–423.
- Noronha, T. F., Resende, M. G. C., e Ribeiro, C. C.** (2011). A biased random-key genetic algorithm for routing and wavelength assignment. *Journal of Global Optimization*, 50:503–518.
- Candia-Véjar, A., Álvarez-Miranda, E., e Maculan, N.** (2011). Minmax regret combinatorial optimization problems: an algorithmic perspective. *RAIRO-Operation Reserach*, 45:101–129.
- Kasperski, A. e Zieliński, P.** (2011). On the approximability of robust spanning tree problems. *Theoretical Computer Science*, 412:365–374.
- Gonçalves, J. F. e Resende, M. G. C.** (2012). A parallel multi-population biased random-key genetic algorithm for a container loading problem. *Computers & Operations Research*, 29:179–190.
- Coco, A. A., Noronha, T. F., e Santos, A. C.** (2012). A biased random-key genetic algorithm for the robust shortest path problem. In *Proceedings of the global optimization workshop*, pages 53–56, Natal, Brazil.