



UNIVERSIDADE FEDERAL DO CEARÁ
CENTRO DE CIÊNCIAS
DEPARTAMENTO DE ESTATÍSTICA E MATEMÁTICA APLICADA
CURSO DE GRADUAÇÃO EM MATEMÁTICA INDUSTRIAL

GABRIEL HELLEN DE SOUSA

**UMA ABORDAGEM DE PROGRAMAÇÃO MATEMÁTICA PARA O NÚMERO DE
ENVOLTÓRIA DE UM GRAFO**

FORTALEZA

2018

GABRIEL HELLEN DE SOUSA

UMA ABORDAGEM DE PROGRAMAÇÃO MATEMÁTICA PARA O NÚMERO DE
ENVOLTÓRIA DE UM GRAFO

Trabalho de Conclusão de Curso apresentado ao
Curso de Graduação em Matemática Industrial
do Centro de Ciências da Universidade Federal
do Ceará, como requisito parcial à obtenção do
grau de bacharel em Matemática Industrial.

Orientador: Prof. Dr. Manoel Bezerra
Campêlo Neto

Coorientador: Prof. Dr. Júlio César Silva
Araújo

FORTALEZA

2018

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

- S696a Sousa, Gabriel Hellen de.
Uma Abordagem de Programação Matemática para o Número de Envoltória de um Grafo / Gabriel Hellen de Sousa. – 2018.
58 f. : il.
- Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Centro de Ciências, Curso de Matemática Industrial, Fortaleza, 2018.
Orientação: Prof. Dr. Manoel Bezerra Campêlo Neto.
Coorientação: Prof. Dr. Júlio César Silva Araújo.
1. Convexidade. 2. Número de Envoltória. 3. Heurística. I. Título.

CDD 510

GABRIEL HELLEN DE SOUSA

UMA ABORDAGEM DE PROGRAMAÇÃO MATEMÁTICA PARA O NÚMERO DE
ENVOLTÓRIA DE UM GRAFO

Trabalho de Conclusão de Curso apresentado ao
Curso de Graduação em Matemática Industrial
do Centro de Ciências da Universidade Federal
do Ceará, como requisito parcial à obtenção do
grau de bacharel em Matemática Industrial.

Aprovada em:

BANCA EXAMINADORA

Prof. Dr. Manoel Bezerra Campêlo
Neto (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. Júlio César Silva Araújo (Coorientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. Rafael Castro de Andrade
Universidade Federal do Ceará (UFC)

Prof. Dr. Ronan Pardo Soares
Universidade Federal do Ceará (UFC)

Aos meus pais, que sempre me incentivaram e investiram na minha educação. À comunidade científica, para que este trabalho possa ser usado como referência para auxiliar outros pesquisadores na mesma área de pesquisa.

AGRADECIMENTOS

Ao Professor Manoel Bezerra Campêlo Neto, por me orientar em minha monografia.

Ao Professor Júlio César Silva Araújo, por me coorientar em minha monografia.

Aos Professores Rafael Castro de Andrade e Ronan Pardo Soares, por aceitarem o convite para serem avaliadores da minha monografia.

Aos meus pais, que sempre investiram na minha educação, esperando um futuro promissor para o seu filho. Sem eles, não estaria na posição que me encontro hoje.

Ao CNPq, pelo financiamento da bolsa de iniciação científica durante os últimos dois anos.

Obrigado a todos.

“Nossa maior fraqueza está em desistir. O caminho mais certo de vencer é tentar mais uma vez.”

(Thomas Edison)

RESUMO

O número de envoltória (*hull number*) de um grafo $G = (V, E)$ não-orientado, onde V é o conjunto de vértices e E , o conjunto de arestas, consiste no menor número de vértices que, inicialmente contaminados, conseguem, iterativamente, contaminar todo o grafo. Os tipos de contaminação (convexidade) estudados neste trabalho foram geodésica, P_3 e P_3^* . Determinar o número de envoltória é um problema *NP*-Difícil, mesmo para classes de grafos como bipartidos. Neste trabalho, foram estudados e implementados modelos matemáticos e heurísticas para o problema. Para resolução dos modelos utilizou-se o CPLEX, acoplado à linguagem de programação C++, e para a heurística utilizou-se uma implementação na mesma linguagem. Os grafos usados como instâncias de teste foram bipartidos e arbitrários, criados a partir de dois parâmetros: quantidade de vértices e um fator de probabilidade para definir a existência das arestas. Os resultados apresentados por cada modelo e pela heurística, referentes ao tempo de execução e à solução, foram armazenados e comparados.

Palavras-chave: Convexidade. Número de Envoltória. Heurística.

ABSTRACT

The hull number of an undirected graph $G = (V, E)$, where V is the set of vertices and E is the set of edges, consists of the smallest number of vertices that, initially contaminated, can iteratively contaminate the whole graph. The types of contamination (convexity) studied in this work were geodetic, P_3 , and P_3^* . Determining the hull number is an *NP*-hard problem, even for bipartite graphs. In this work, mathematical models and heuristics for the problem were studied and implemented. To solve the models, the CPLEX was used, coupled with the C++ programming. The same language was used to code the heuristic. The graphs used as test instances were bipartite and arbitrary graphs, created from two parameters: the number of vertices and a probability factor to define the existence of edges. The results presented by each model and the heuristic, related to the execution time and solution quality, were stored and compared.

Keywords: Convexity. Hull number. Heuristic.

LISTA DE FIGURAS

| | |
|--|----|
| Figura 1 – Exemplo de um grafo G | 15 |
| Figura 2 – Contra-exemplo para Propriedade 2 em P_3 | 22 |
| Figura 3 – Blocos de um Grafo | 23 |
| Figura 4 – Árvore Bloco-Articulação | 23 |
| Figura 5 – Exemplo em que $hn(G)$ aumenta com adição de aresta | 25 |
| Figura 6 – Exemplo que $hn(G)$ diminui com adição de aresta na convexidade Geodésica | 25 |
| Figura 7 – Exemplo em que $hn(G)$ diminui com adição de aresta na convexidade P_3^* | 26 |
| Figura 8 – Exemplo em que $hn(G)$ aumenta com adição de vértice | 26 |
| Figura 9 – Exemplo em que $hn(G)$ diminui com adição de vértice | 27 |
| Figura 10 – Contra-exemplo para monotonicidade do número de envoltória com relação ao tempo de contaminação | 28 |
| Figura 11 – $hn_0(G) = 12, S_0(G) = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$ | 28 |
| Figura 12 – $hn_1(G) = 8, S_1(G) = \{1, 4, 5, 8, 9, 10, 11, 12\}$ | 29 |
| Figura 13 – $hn_2(G) = 8, S_2(G) = \{1, 4, 5, 8, 9, 10, 11, 12\}$ | 29 |
| Figura 14 – $hn_3(G) = 7, S_3(G) = \{4, 5, 8, 9, 10, 11, 12\}$ | 29 |
| Figura 15 – Tempo de computação do modelo coconvexo em grafos bipartidos | 52 |
| Figura 16 – Tempo de computação do modelo coconvexo em grafos arbitrários | 53 |
| Figura 17 – Tempo de computação do modelo coconvexo em função de p para grafos Bipartidos | 53 |
| Figura 18 – Tempo de computação do modelo coconvexo em função de p para grafos arbitrários | 54 |

LISTA DE TABELAS

| | |
|--|----|
| Tabela 1 – Principais convexidades de caminhos na literatura | 20 |
| Tabela 2 – Resultados da heurística para convexidades de intervalo em grafos Bipartidos | 45 |
| Tabela 3 – Resultados da heurística para convexidades de intervalo em grafos Arbitrários | 45 |
| Tabela 4 – Heurística Geral × Específica para grafos bipartidos | 46 |
| Tabela 5 – Heurística Geral × Específica para grafos arbitrários | 46 |
| Tabela 6 – Resultados dos modelos para grafos bipartidos | 47 |
| Tabela 7 – Resultados dos modelos para grafos arbitrários | 48 |
| Tabela 8 – Decomposição em blocos em grafos Bipartidos | 49 |
| Tabela 9 – Decomposição em blocos em grafos Arbitrários | 49 |
| Tabela 10 – MIP-Start para grafos Bipartidos | 50 |
| Tabela 11 – MIP-Start para grafos Arbitrários | 50 |
| Tabela 12 – Variantes do modelo passo de contaminação em grafos Bipartidos | 51 |
| Tabela 13 – Variantes do modelo passo de contaminação em grafos Arbitrários | 51 |

SUMÁRIO

| | | |
|--------------|--|-----------|
| 1 | INTRODUÇÃO | 13 |
| 1.1 | Motivações | 13 |
| 1.2 | Preliminares | 13 |
| 1.3 | Estruturação da Monografia | 16 |
| 2 | CONVEXIDADE EM GRAFOS | 18 |
| 2.1 | Convexidade | 18 |
| 2.2 | Convexidade de Intervalo | 19 |
| 2.3 | Convexidade de Caminhos | 20 |
| 2.4 | Propriedades | 21 |
| 2.4.1 | <i>Vértices folhas</i> | 21 |
| 2.4.2 | <i>Vértices simpliciais</i> | 22 |
| 2.4.3 | <i>Articulações e Blocos</i> | 22 |
| 2.4.4 | <i>Adição ou remoção de arestas</i> | 24 |
| 2.4.5 | <i>Adição ou Remoção de Vértices</i> | 26 |
| 2.4.6 | <i>Passo de Contaminação</i> | 27 |
| 3 | CONTAMINAÇÃO | 30 |
| 3.1 | Caminhos Mínimos | 30 |
| 3.2 | Vizinhança | 32 |
| 3.3 | Algoritmo de Contaminação | 32 |
| 3.4 | Decomposição em blocos | 33 |
| 4 | HEURÍSTICA | 36 |
| 4.1 | Para convexidades de intervalo | 36 |
| 4.2 | Específica para Convexidade Geodésica | 37 |
| 5 | FORMULAÇÕES | 38 |
| 5.1 | Passo de Contaminação | 38 |
| 5.2 | Coconvexo | 41 |
| 6 | EXPERIMENTOS COMPUTACIONAIS | 43 |
| 6.1 | Instâncias de Teste | 43 |
| 6.2 | Testes com Heurística | 44 |
| 6.2.1 | <i>Testes nas instâncias padrão</i> | 44 |

| | | |
|---------|--|----|
| 6.2.2 | <i>Heurística para Convexidades de Intervalo × Específica para Convexidade Geodésica</i> | 45 |
| 6.3 | Testes com os Modelos | 47 |
| 6.3.1 | <i>Passo de Contaminação × coconvexo</i> | 47 |
| 6.3.1.1 | <i>Testes com modelos básicos</i> | 47 |
| 6.3.1.2 | <i>Testes com Decomposição em Blocos</i> | 48 |
| 6.3.1.3 | <i>Testes com MIP-Start</i> | 49 |
| 6.3.2 | <i>Variantes Passo de Contaminação</i> | 50 |
| 6.3.3 | <i>Aprofundamento coconvexo</i> | 52 |
| 6.3.3.1 | <i>Testes em função do número de vértices</i> | 52 |
| 6.3.3.2 | <i>Testes em função da densidade do grafo</i> | 53 |
| 7 | CONSIDERAÇÕES FINAIS | 55 |
| | REFERÊNCIAS | 57 |

1 INTRODUÇÃO

Este trabalho é fruto de dois anos de pesquisa na área de convexidade em grafos, mais especificamente, na determinação do número de envoltória de um grafo. Parte dos resultados aqui apresentados já foram aceitos e divulgados em três eventos científicos, sob os títulos "Métodos para determinação do Número de Envoltória Geodésico de um Grafo" (CAMPELO *et al.*, 2018), "Calculando o Número de Envoltória nas Convexidades P_3 e P_3^* " (ARAÚJO *et al.*, 2018a) e "Comparação de Heurísticas para o Número de Envoltória nas Convexidades P_3 e P_3^* " (SOUSA *et al.*, 2018).

1.1 Motivações

Em uma monografia, espera-se um estudo detalhado sobre um determinado assunto, podendo ser composta apenas por uma revisão bibliográfica do tema tratado ou por novas contribuições para a área da pesquisa. Neste trabalho, estamos propondo novas modelagens matemáticas e heurísticas polinomiais para o problema de encontrar o número de envoltória de um grafo, em três diferentes convexidades.

Nos três casos, o problema a ser tratado é conhecidamente *NP*-difícil. Grosseiramente falando, isso significa que não há garantia da existência de algoritmo para solução do problema, cujo tempo de execução cresça polinomialmente com o tamanho do grafo de entrada. Tal fato justifica a abordagem do problema com heurísticas (para obter soluções viáveis rapidamente) e métodos de programação matemática (para obter soluções exatas).

1.2 Preliminares

Este trabalho, está alicerçado em duas grandes áreas de estudo: otimização matemática e teoria dos grafos.

Genericamente, problemas de otimização consistem em determinar, dentre um conjunto de soluções possíveis, que atendem às condições do problema, uma que seja a melhor, segundo algum critério que ranqueia as soluções viáveis.

Quando nos deparamos com um problema desse tipo, intuitivamente procuramos formas de tentar solucioná-lo. Às vezes, o fazemos de forma desorganizada e não conseguimos resultados. Uma solução para esses casos pode ser encontrada com programação matemática, onde modelamos e organizamos os dados do problema, por meio de variáveis que representam

as características dele, de modo que ele fique mais simples de ser estudado. Modelos de programação matemática são compostos por três partes: variáveis, restrições e função objetivo.

As variáveis de um modelo matemático são elementos de decisão, podendo determinar uma condição ou quantificação de algum fator, este último podendo ser ou não particionado. Restrições determinam limitações ou imposições para alguma característica do problema, a serem descritas por expressões matemáticas em termos das variáveis. Por fim, temos a função objetivo do modelo, que define o critério que queremos otimizar, dadas as variáveis e as restrições estabelecidas. O objetivo pode ser de dois tipos: minimização ou maximização.

A solução de um modelo matemático é a melhor atribuição de valores às suas variáveis, considerando a função objetivo, dentro dos limites definidos pelas restrições. A depender das características do modelo, são usados diferentes métodos de otimização, para sua resolução. Em nosso estudo, o problema é tratado por meio de modelagens de programação linear-inteira (PLI), onde cada uma das funções que definem as restrições e a função objetivo são lineares, e os valores que cada variável pode assumir pertence ao conjunto dos números inteiros. Uma formulação genérica para modelos PLI é:

$$\begin{aligned} \min \quad & c^t x \\ \text{s.a:} \quad & Ax \leq b \\ & x \geq 0 \\ & x \in \mathbb{Z}^n \end{aligned}$$

Nessa descrição c é um vetor de coeficientes, com $c^t x$ representando a função objetivo, onde \min indica que a mesma é de minimização. $A_{m \times n}$ é a matriz de coeficientes das n variáveis, nas m restrições do modelo. Para transformarmos esse problema em maximização basta pré-multiplicarmos a função objetivo por -1 . $Ax \leq b$ representa o conjunto de restrições do modelo e o vetor x compreende as variáveis. Por simplicidade, estamos supondo variáveis não negativas.

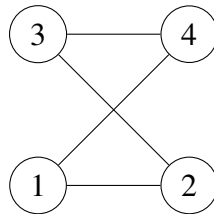
Modelos matemáticos, apesar de representarem problemas de otimização de modo preciso, podem demandar grande esforço computacional para serem solucionadas. Nesse contexto, podemos propor métodos heurísticos de resolução, que não garantem que a solução obtida seja ótima, mas, em geral, nos retornam uma solução rapidamente.

Neste trabalho, as modelagens matemáticas e heurísticas são aplicadas em grafos,

que descrevem instâncias do problema tratado. Nesse caso, variáveis podem representar, por exemplo, vértices de um grafo G ; e restrições determinam condições que esses vértices devem atender.

Um grafo G simples é um par ordenado (V, E) , onde V é o conjunto de vértices e E , o conjunto de arestas, que representam uma relação entre pares de vértices. A Figura 1 ilustra a representação gráfica de um grafo, onde $V = \{1, 2, 3, 4\}$ e $E = \{(1, 2), (2, 3), (3, 4), (1, 4)\}$.

Figura 1 – Exemplo de um grafo G



Fonte: elaborado pelo autor.

Um dos problemas fundamentais de otimização é a determinação de um caminho de menor comprimento em um grafo G . Um i, j -caminho em um grafo não-direcionado é uma sequência de vértices distintos $V' = \langle v_0, v_1, \dots, v_{k-1}, v_k \rangle$, com $v_0 = i$ e $v_k = j$, onde $v_l \in V(G)$, $l \in \{0, \dots, k\}$ e, para todo par (v_l, v_{l+1}) , a aresta $(v_l, v_{l+1}) \in E(G)$. O comprimento desse caminho é k , ou seja, a quantidade de arestas. Por exemplo, na Figura 1 $\langle 1, 2, 3, 4 \rangle$ é um 1,4-caminho de comprimento 3.

Em nosso caso, estudamos grafos simples não-direcionados, não ponderados e conexos, com pelo menos 3 vértices. Um grafo conexo é tal que, partindo de qualquer vértice, podemos alcançar qualquer outro, por meio de uma sequência de vértices e arestas (induzidas pelos pares de vértices consecutivos na sequência). Em outras palavras, para todo $i, j \in V(G)$ existe um caminho entre i e j . Por exemplo, o grafo da Figura 1 é conexo.

Existem diferentes tipos de grafos: completo, árvore, bipartido, trivial, etc. Um grafo completo, por exemplo, é tal que, para qualquer par $i, j \in V(G)$, temos $(i, j) \in E(G)$.

Um dos principais tipos de grafo usados neste trabalho é o grafo bipartido. Ele se caracteriza por podermos dividir o conjunto de vértices $V(G)$ em dois subconjuntos V_1 e V_2 , tais que $V_1 \cup V_2 = V(G)$ e $V_1 \cap V_2 = \emptyset$, e toda aresta liga algum vértice de V_1 a algum vértice de V_2 . Mais definições sobre grafos podem ser encontradas em West (2000).

Quando esse caminho possui o menor comprimento possível, dentre todos os i, j -caminhos, ele é dito um i, j -caminho mínimo. Se atribuirmos pesos às arestas de G , o peso do caminho é a soma dos pesos de suas arestas. Nesse caso, definimos um i, j -caminho com relação

ao peso. O caso de pesos unitários é equivalente àquele sem pesos.

Uma aresta entre dois vértices não consecutivos em um caminho é dita uma corda. Chamamos um i, j -caminho de caminho induzido em G quando ele não possuir cordas. Na Figura 1 todos os caminhos são induzidos, porém, se adicionarmos a aresta $(2,4)$, o caminho $\langle 1,2,3,4 \rangle$ deixa de ser induzido.

Uma classe de problemas combinatórios que tem recebido bastante atenção recentemente compreende o que se chama genericamente de problemas de contaminação ou infecção ou convexidade em grafos. Há vários tipos de contaminação considerados na literatura, que podem ser vistos como um processo iterativo, onde, a partir de um subconjunto de vértices inicialmente contaminados, passo a passo, outros também o serão. Por exemplo, podemos considerar que dois vértices contaminados infectam, no passo seguinte, todos aqueles que estão em algum caminho mínimo entre eles (PELAYO, 2013), ou todos aqueles que são simultaneamente vizinhos dos dois (PENSO *et al.*, 2015), etc. Também são estudadas na literatura variações em que vértices contaminados podem se tornar “sadios”, segundo alguma condição pré-determinada (DOURADO *et al.*, 2012). Nesse contexto, um dos problemas fundamentais consiste em determinar o menor número de vértices que, inicialmente contaminados, infectam, iterativamente, todo o grafo em algum momento. Esse menor valor é conhecido como número de envoltória (EVERETT; SEIDMAN, 1985).

Embora os problemas de convexidade venham sendo bastante estudados na última década, que conheçamos, não havia, até a apresentação dos nossos trabalhos Campelo *et al.* (2018), Araújo *et al.* (2018a), Sousa *et al.* (2018), formulação de programação linear-inteira proposta na literatura, nem estudos que proponham e avaliem o comportamento de heurísticas polinomiais para obtenção do número de envoltória de um grafo. Há, entretanto, a implementação de um algoritmo exponencial para obtenção do número de envoltória em uma das convexidades estudadas neste trabalho no SageMath (COHEN, 2018), que sugere uma formulação de programação linear-inteira com um número exponencial de restrições. Vale ressaltar que, embora a sua formulação seja facilmente obtida a partir da implementação no SageMath, acreditamos que a mesma não foi explicitamente proposta na literatura.

1.3 Estruturação da Monografia

Esta monografia está estruturada em três partes principais: definição do problema, métodos de resolução e resultados computacionais e considerações finais.

No Capítulo 2, definimos noções de convexidade de maneira geral e, posteriormente, aplicada em grafos. Apresentamos algumas das principais produções sobre o problema na literatura, bem como mostramos algumas propriedades específicas sobre cada convexidade em alguns tipos de grafos especiais. No Capítulo 3, relacionamos o processo de convexificação com um processo de contaminação ou infecção. Adicionalmente, apresentamos alguns algoritmos que auxiliam em tal processo, dependendo da convexidade estudada.

Após a apresentação do problema, mostramos os métodos propostos para resolução dele. No Capítulo 4, propomos duas heurísticas polinomiais para determinar o número de envoltória de um grafo: uma geral, para diferentes convexidades de caminhos, e outra mais específica para convexidade geodésica. Ambas são baseadas em estratégias gulosas.

Depois das heurísticas, apresentamos duas modelagens matemáticas no Capítulo 5. A primeira formulação de programação linear-inteira possui uma quantidade polinomial de variáveis e restrições, o que a torna compacta. A segunda é baseada na proposta do algoritmo exponencial em Cohen (2018). Essa também possui um número polinomial (na verdade linear) de variáveis, entretanto tem a desvantagem de possuir um número exponencial de restrições.

No Capítulo 6, apresentamos os resultados de testes computacionais sobre instâncias correspondentes a grafos bipartidos e grafos arbitrários, tomados aleatoriamente. Comparamos então as soluções ótimas obtidas pela implementação, no CPLEX, das formulações propostas com as soluções obtidas pelas heurísticas. Avaliamos também o desempenho dos modelos com respeito ao tamanho do grafo e sua densidade.

Finalmente, no Capítulo 7, exibimos conclusões sobre este trabalho e apresentamos novos caminhos para o desenvolvimento da pesquisa sobre esse parâmetro.

2 CONVEXIDADE EM GRAFOS

Neste capítulo, apresentamos noções básicas de convexidade e apresentamos o conceito de convexidade em grafos. Apresentamos uma breve revisão bibliográfica sobre trabalhos na área de convexidade em grafos. Após, mostramos a noção de convexidade de caminhos e algumas propriedades que cada convexidade estudada apresenta.

2.1 Convexidade

A noção de convexidade tem sua origem na geometria euclidiana, na qual o menor caminho entre dois pontos é dado pelo segmento de reta entre eles. Nesse contexto, um conjunto é dito convexo quando o segmento (menor caminho) entre quaisquer dois pontos do conjunto está nele contido; e tal segmento é dito a envoltória convexa dos dois pontos. Recursivamente, definimos a envoltória convexa de um conjunto finito S como a união das envoltórias convexas de um ponto arbitrário $x \in S$ e todo ponto y que pertence a envoltória convexa de $S \setminus \{x\}$.

De forma mais geral, um *espaço de convexidade* é um par ordenado $(\mathcal{V}, \mathcal{C})$, onde \mathcal{V} é um conjunto finito não vazio e \mathcal{C} é uma família de subconjuntos de \mathcal{V} , chamados de *conjuntos convexas*, satisfazendo:

(C1) $\emptyset, \mathcal{V} \in \mathcal{C}$

(C2) $C \cap C' \in \mathcal{C}$, para todos $C, C' \in \mathcal{C}$

Os complementos de conjuntos convexas (em relação a \mathcal{V}) são chamados de coconvexos.

A seguir temos um exemplo de um espaço de convexidade atendendo (C1) e (C2):

Ex: $(\mathcal{V}, \mathcal{C}) = (\{1, 2, 3\}, \{\emptyset, \{1, 2\}, \{2, 3\}, \{2\}, \{1, 2, 3\}\})$

Dado um subconjunto $C \subseteq \mathcal{V}$, a *envoltória convexa de C* (com respeito a $(\mathcal{V}, \mathcal{C})$) é o único conjunto minimal (com respeito à inclusão) $C' \in \mathcal{C}$ que contém C e é denotado por $\text{hullset}_{(\mathcal{V}, \mathcal{C})}(C)$, ou simplesmente, $hs_{(\mathcal{V}, \mathcal{C})}(C)$. Se $hs_{(\mathcal{V}, \mathcal{C})}(C) = \mathcal{V}$, então C é chamado um *conjunto de envoltória* de $(\mathcal{V}, \mathcal{C})$. O *número de envoltória de \mathcal{V} com respeito a \mathcal{C}* é a cardinalidade de um conjunto de envoltória mínimo, denotado por $hn_{(\mathcal{V}, \mathcal{C})}$. Essas noções remontam aos trabalhos de Farber e Jamison (1986), Duchet (1988), Everett e Seidman (1985).

No exemplo anterior, podemos observar que $hs_{(\mathcal{V}, \mathcal{C})}(\{3\})$ é o *conjunto convexo* $\{2, 3\}$; Logo, o conjunto $\{3\}$ não é um *conjunto de envoltória* para esse *espaço de convexidade*. Entretanto, se olharmos para $\{1, 3\}$, $hs_{(\mathcal{V}, \mathcal{C})}(\{1, 3\}) = \{1, 2, 3\}$; então $\{1, 3\}$ é um *conjunto de*

envoltória. Dessa forma, deduzimos $hn_{(\mathcal{V}, \mathcal{C})} \leq 2$. Por outro lado, para cada conjunto unitário, temos $|hs_{(\mathcal{V}, \mathcal{C})}(\{1\})| = 2$, $|hs_{(\mathcal{V}, \mathcal{C})}(\{2\})| = 1$, $|hs_{(\mathcal{V}, \mathcal{C})}(\{3\})| = 2$. Como nenhum conjunto unitário é de envoltória, então $hn_{(\mathcal{V}, \mathcal{C})} \geq 2$. Logo $hn_{(\mathcal{V}, \mathcal{C})} = 2$.

Destacamos duas propriedades básicas de um espaço de convexidade:

1. $S \subseteq S' \subseteq \mathcal{V} \Rightarrow hs_{(\mathcal{V}, \mathcal{C})}(S) \subseteq hs_{(\mathcal{V}, \mathcal{C})}(S')$: Como $S \subseteq S' \subseteq hs_{(\mathcal{V}, \mathcal{C})}(S') \in \mathcal{C}$, temos que $hs_{(\mathcal{V}, \mathcal{C})}(S)$ é subconjunto de $hs_{(\mathcal{V}, \mathcal{C})}(S')$.
2. $C \neq \emptyset$ é coconvexo, e S é conjunto de envoltória $\Rightarrow S \cap C \neq \emptyset$: se $S \cap C = \emptyset$, então $S \subseteq \mathcal{V} \setminus C$ e, portanto, $hs_{(\mathcal{V}, \mathcal{C})}(S) \subseteq \mathcal{V} \setminus C \neq \mathcal{V}$, absurdo pois $hs_{(\mathcal{V}, \mathcal{C})}(S) = \mathcal{V}$.

2.2 Convexidade de Intervalo

Quando se trata de convexidade em grafos, a grande maioria das convexidades definidas na literatura toma \mathcal{V} como o conjunto de vértices e a família \mathcal{C} como subconjuntos de vértices que são pontos fixos de uma função de intervalo. Dado um grafo $G = (V, E)$, uma *função de intervalo* é uma função $I : 2^{V(G)} \rightarrow 2^{V(G)}$ tal que: **(i)** $I(S) \supseteq S$, para todo $S \subseteq V(G)$ e **(ii)** $I(S') \subseteq I(S)$, para todo $S' \subseteq S$. Os vértices em $I(S) \setminus S$ são ditos *infectados* ou *gerados* por S .

Proposição: Seja $G = (V, E)$ um grafo e $I : 2^{V(G)} \rightarrow 2^{V(G)}$ uma função de intervalo, então $(\mathcal{V}, \mathcal{C})$ é espaço de convexidade, para $\mathcal{V} = V(G)$ e $\mathcal{C} = \{S \subseteq V : I(S) = S\} \cup \{\emptyset\}$.

Prova. Devemos mostrar que $(\mathcal{V}, \mathcal{C})$ satisfaz (C1) e (C2). Trivialmente $\{\emptyset\} \in \mathcal{C}$. Além disso, $V \in \mathcal{C}$ pois $V \subseteq I(V) \subseteq V$, assim obtemos (C1). Tome agora $S, S' \in \mathcal{C}$. Para obter (C2) devemos mostrar que $I(S \cap S') = S \cap S'$. Por (i), $S \cap S' \subseteq I(S \cap S')$. Por (ii), $I(S \cap S') \subseteq I(S) \cap I(S')$. Mas $S \cap S' \subseteq I(S) \cap I(S')$, pois $S, S' \in \mathcal{C}$. Logo o resultado segue. ■

Mostramos que os pontos fixos de uma função de intervalo definem uma convexidade em G , dita convexidade de intervalo.

Vários parâmetros em grafos relacionados à convexidade têm sido explorados na literatura. O número de Radon, número de Carathéodory, número de envoltória, o número de convexidade, o número de intervalo e tempo máximo de infecção são alguns dos parâmetros de convexidade estudados. O número de envoltória é o parâmetro em que mais estamos interessados nesta monografia. Estudos mais aprofundados sobre os outros parâmetros podem ser encontrados em Marcilon (2017).

Há várias convexidades de intervalo estudadas na literatura. Aqui estudamos convexidades de intervalo específicas, chamadas de convexidades de caminhos, mais especificamente as convexidades geodésica, P_3 e P_3^* .

2.3 Convexidade de Caminhos

Seja uma função de intervalo $I : 2^{V(G)} \rightarrow 2^{V(G)}$, que, portanto, define uma convexidade em G . Dizemos que essa função de intervalo denota uma convexidade de caminho se, para todo $S \subseteq V$, e para todo $k \in I(S) \setminus S$, existem $i, j \in S$, tal que k está em um tipo pré-determinado (como os descritos a seguir) de i, j -caminho, ou seja, a cada iteração a função I agrega vértices que estão nesse tipo de i, j -caminho. As convexidades de caminho se diferenciam exatamente pela característica específica que define esse tipo de caminho. Em Thompson (2017), são listadas as principais convexidades de caminhos da literatura. Na Tabela 1 a seguir estão os nomes das principais convexidades e sua caracterização para cada i, j -caminho:

Tabela 1 – Principais convexidades de caminhos na literatura

| Convexidade | Característica do i, j -caminho |
|---------------|---|
| Geodésica | Caminhos mínimos |
| Monofônica | Caminhos induzidos |
| g^3 | Caminhos mínimos com comprimento no mínimo três |
| m^3 | Caminhos induzidos com comprimento no mínimo três |
| g_k | caminhos mínimos de comprimento no máximo k |
| P_3 | Caminhos de comprimento dois |
| P_3^* | Caminhos induzidos de comprimento dois |
| triangle-path | Caminhos permitindo cordas de comprimento dois |
| total | Caminhos permitindo cordas de comprimento três |
| detour | Caminhos máximos no grafo |
| all-path | Todos os caminhos |

Fonte: adaptado de Thompson (2017).

Para este trabalho focamo-nos em três convexidades: geodésica, P_3 e P_3^* . A seguir, particularizamos, para essas convexidades, alguns dos conceitos apresentados.

Na convexidade geodésica, dizemos que um subconjunto $S \subseteq V(G)$ é *convexo*, se, para todos $u, v \in S$, os vértices em qualquer u, v -geodésico (u, v -caminho mínimo) também pertencem a S . Para a convexidade P_3 mudamos, nessa definição, de caminho mínimo para caminho de tamanho 2. Em P_3^* , temos uma restrição a mais que P_3 , onde os caminhos devem ser induzidos, ou seja, não pode haver aresta entre duas extremidades.

Note que, para as convexidades P_3 e P_3^* , em vez de considerarmos caminhos, podemos olhar para vizinhança dos vértices. Denotaremos $N_G(v)$ a vizinhança aberta de um vértice $v \in V(G)$, ou seja, $N_G(v) = \{k \in V(G) : (v, k) \in E(G)\}$. Lembre que, na convexidade P_3 , um vértice k é contaminado por i e j , já contaminados, se k está em um i, j -caminho de comprimento 2. Note que essa condição é equivalente a dizer que $(i, k), (k, j) \in E(G)$, ou seja,

$k \in N_G(i) \cap N_G(j)$. Assim, podemos construir o caminho $i \rightarrow k \rightarrow j$, com comprimento 2. Em outras palavras, ele será contaminado sempre que dois de seus vizinhos estejam contaminados. A mesma abordagem pode ser utilizada em P_3^* , se $(i, j) \notin E(G)$; caso contrário, o caminho induzido entre i e j seria a própria aresta (i, j) .

O problema de determinar o número de envoltória geodésico de um grafo é *NP*-difícil, mesmo para grafos cubos parciais (ALBENQUE; KNAUER, 2016) e grafos cordais (BESSY *et al.*, 2017). Um *cubo parcial* é um subgrafo isométrico do hipercubo e, portanto, o problema é *NP*-difícil para grafos bipartidos. Uma demonstração independente desse fato pode ser encontrada em Araújo *et al.* (2013). Também para o número de envoltória geodésico, há na literatura diversos limitantes superiores, algoritmos polinomiais e algoritmos parametrizados para diversas classes de grafos, além de demonstrações que tais algoritmos paramétricos não existem, sob a hipótese de que $FPT \neq W[1]$ (veja alguns exemplos em Dourado *et al.* (2009), Araújo *et al.* (2013), Araújo *et al.* (2016), Marcilon (2017)). Em Cohen (2018) há a implementação de um algoritmo exponencial que calcula esse parâmetro. Para as convexidades P_3 e P_3^* , determinar o número de envoltória também é um problema *NP*-difícil para grafos bipartidos. Em Araújo *et al.* (2018b) há a demonstração desse fato para essas duas convexidades.

2.4 Propriedades

Nesta seção, apresentaremos algumas propriedades para convexidades de caminhos, em particular para as três convexidades estudadas. Tais propriedades sugerem estratégias de pré-processamento ou decomposição do problema, que poderão ser utilizados nos métodos propostos mais à frente. Vamos denotar de $G + \{v\}$ e $G + (i, j)$ como, respectivamente, o grafo G com a adição do vértice v e o grafo G com a adição da aresta (i, j) .

2.4.1 Vértices folhas

Um vértice v é chamado de vértice folha em G se $|N_G(v)| = 1$, ou seja, v tem grau 1. Observe que qualquer vértice folha nunca estará em um i, j -caminho em G , o que nos leva a seguinte propriedade:

Propriedade 1: Todo vértice folha pertence a qualquer conjunto de envoltória, em qualquer convexidade de caminho.

2.4.2 Vértices simpliciais

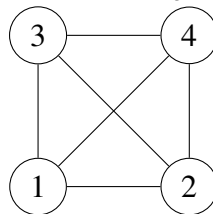
Um vértice v é chamado de vértice simplicial em G se $N_G(v)$ é uma clique. Uma clique C em G é tal que: $C \subseteq V(G)$ e para todo $i, j \in C$ existe a aresta $(i, j) \in E(G)$. Em particular um vértice folha é simplicial. Observe que, se v é vértice simplicial, ele nunca está em um i, j -caminho mínimo de G . De fato, suponha, por absurdo, que exista um i, j -caminho mínimo, em número de arestas, com a seguinte sequência de vértices $X = \{x_0, x_1, \dots, x_{a-1}, x_a, x_{a+1}, \dots, x_k\}$, onde $x_0 = i, x_k = j$ e $x_a = v$, com v simplicial. Podemos construir um caminho X' da seguinte forma $X' = \{x_0, x_1, \dots, x_{a-1}, x_{a+1}, \dots, x_k\}$, pois $x_{a-1}, x_{a+1} \in N_G(x_a)$ e, por conseguinte, a aresta $(x_{a-1}, x_{a+1}) \in E(G)$. Logo $|X'| < |X|$, um absurdo pois $|X|$ é mínimo.

A constatação acima leva-nos à seguinte propriedade:

Propriedade 2: Todo vértice simplicial pertence a qualquer conjunto de envoltória mínimo nas convexidades de caminhos mínimos, em particular geodésica e P_3^* .

Para a convexidade P_3 , onde os caminhos devem ter tamanho 2, mas não obrigatoriamente mínimos, existem contra exemplos para esse resultado estabelecido para as duas outras convexidades.

Figura 2 – Contra-exemplo para Propriedade 2 em P_3



Fonte: elaborado pelo autor.

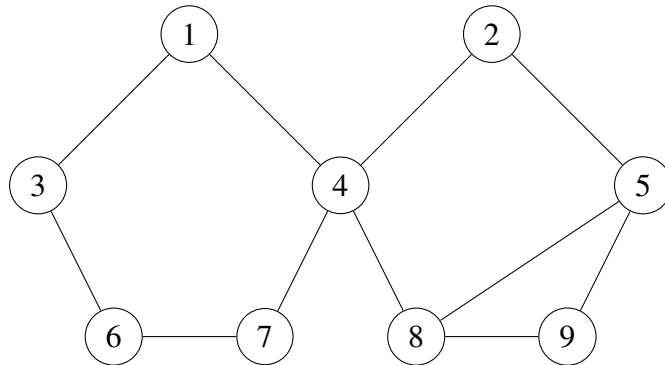
No grafo da Figura 2, observe que $hn(G)$ nas convexidades geodésica e P_3^* é 4, conforme a propriedade 2. Para a convexidade P_3 basta, entretanto, selecionarmos qualquer par de vértices, que estes conseguirão contaminar os dois vizinhos restantes, mostrando que a propriedade demonstrada anteriormente não pode ser aplicada à convexidade P_3 .

2.4.3 Articulações e Blocos

Um subgrafo de G que é conexo e maximal, com respeito a propriedade de ser conexo, é dito uma *componente* (conexa). Dizemos que um vértice $v \in V(G)$ é um *vértice de corte* ou *articulação*, se $G - v$ tem mais componentes que G . Em particular, se G é conexo, v é vértice de corte se, e somente se, $G - v$ é desconexo. Um grafo conexo sem vértices de

corte é chamado de *bloco*. Chamamos um subgrafo B de G de *bloco de G* , se B é um bloco e maximal com respeito a essa propriedade, ou seja, B é conexo, e para todo $v \in B$, temos que $B - v$ é conexo, mas para todo $v \in V \setminus B$, $B \cup \{v\}$ é desconexo ou tem vértice de corte. No grafo da Figura 3 temos dois blocos B_1 e B_2 , onde $V(B_1) = \{1, 3, 4, 6, 7\}$ e $V(B_2) = \{2, 4, 5, 8, 9\}$. Ele possui apenas a articulação 4.

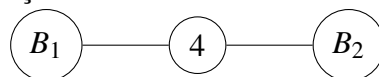
Figura 3 – Blocos de um Grafo



Fonte: elaborado pelo autor.

A decomposição em blocos de um grafo consiste em dividi-lo em seus blocos, gerando uma árvore, chamada árvore de bloco-articulação. Essa árvore tem um vértice para cada bloco e um vértice para cada articulação, bem como uma aresta entre um bloco e uma articulação se esta pertence àquele. A Figura 4 ilustra a árvore bloco-articulação correspondente ao grafo da Figura 3. Observe que uma folha de uma árvore bloco-articulação corresponde a um bloco com uma única articulação, por isso chamado bloco-folha. Por outro lado, toda articulação define um vértice interno da árvore, pois está na interseção de pelo menos dois blocos. Na verdade, os vizinhos da articulação na árvore são exatamente os blocos a que ela pertence.

Figura 4 – Árvore Bloco-Articulação



Fonte: elaborado pelo autor.

Todo caminho em G pode ser descrito por uma sequência de vértices, que não se repetem na sequência, do tipo: $\langle u_1^0, u_2^0, \dots, u_{k_1}^0, v_1, u_1^2, u_2^2, \dots, u_{k_2}^2, v_2, \dots, v_p, u_1^p, u_2^p, \dots, u_{k_p}^p \rangle$, onde $u_1^i, u_2^i, \dots, u_{k_i}^i$ são vértices de um mesmo bloco B_i , para $i = \{0, \dots, p\}$, e B_i e B_{i+1} são blocos adjacentes que se interceptam na articulação v_i , para $\{i = 1, \dots, p-1\}$. Tal caminho tem um caminho correspondente $(B_0, v_1, B_1, v_2, \dots, v_p, B_p)$ na árvore bloco-articulação, o único caminho entre B_0 e B_p na árvore. Sendo assim, um u, v -caminho em G , com u e v pertencendo a blocos

diferentes, digamos B e B' , passa necessariamente por todas as articulações de G que estão no único caminho entre B e B' na árvore. Essas observações nos levam à seguinte propriedade:

Propriedade 3: Na convexidade geodésica, todo conjunto de envoltória mínimo não contém articulação e contém pelo menos um vértice de cada bloco-folha.

Prova. Seja S^* , com $|S^*| = hn(G)$, um conjunto de envoltória. Se G não possuir articulação, o resultado é trivial. Admita então que v seja vértice de corte em G . Seja B o conjunto de vértices de um bloco-folha da árvore bloco-articulação de G e v o único vértice de articulação em B . Primeiro mostramos que S^* contém um vértice de $B \setminus v$.

Note que $G \setminus (B \setminus v)$ é convexo, pois quaisquer dois vértices nesse conjunto não atingem $B \setminus v$. Logo, $B \setminus v$ é coconvexo e, portanto, $S^* \cap (B \setminus v) \neq \emptyset$.

Considere agora um vértice de articulação u de G . Então existem dois blocos folha B e B' tais que u pertence ao único caminho mínimo entre B e B' na árvore bloco-articulação. Tome $w \in B \cap S^*$ e $w' \in B' \cap S^*$. Logo, u está em todo caminho em w e w' em G . Em outras palavras, w e w' infectam u na convexidade geodésica. Por conseguinte $u \notin S^*$. ■

Seguindo argumentos similares ao da prova da Propriedade 3, Araújo *et al.* (2013) obtém um resultado mais abrangente. Para isso, os autores definem o número de envoltória generalizado. Dado um subconjunto $S \subseteq V$, seja $hn(G, S)$ o tamanho do menor conjunto S' tal que $S \cup S'$ é conjunto de envoltória. Note que $hn(G, \emptyset) = hn(G)$.

Teorema 2.4.1 ((ARAÚJO *et al.*, 2013)) *Seja G um grafo e G_1, G_2, \dots, G_n seus blocos. Para todo $i \leq n$, seja $S_i \subseteq V(G_i)$ o conjunto de vértices de corte de G em G_i , então:*

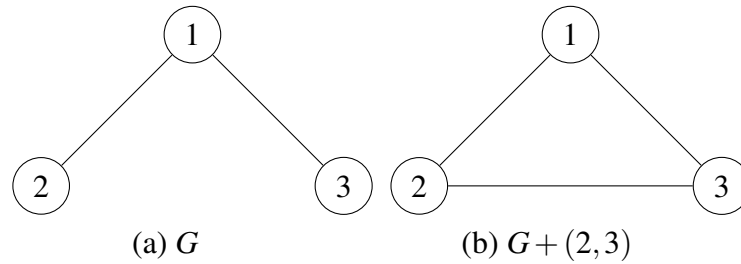
$$hn(G) = \sum_{i \leq n} hn(G_i, S_i).$$

O Teorema 2.4.1 nos mostra que podemos decompor o grafo em todos os seus blocos e resolver o problema em cada um separadamente, fixando os vértices de corte pertencentes ao conjunto de envoltória mínimo de cada bloco. Consideramos que uma articulação faz parte do conjunto de envoltória apenas quando solucionarmos o problema em cada bloco, pois vimos pela Propriedade 3 que articulações não fazem parte de um conjunto de envoltória mínimo.

2.4.4 Adição ou remoção de arestas

Alterações em $E(G)$ não determinam variação específica em $hn(G)$ nas convexidades P_3^* e Geodésica, ou melhor, a inclusão ou exclusão de uma aresta pode fazer o número de envoltória se manter igual, aumentar ou diminuir. Observe o exemplo da Figura 5.

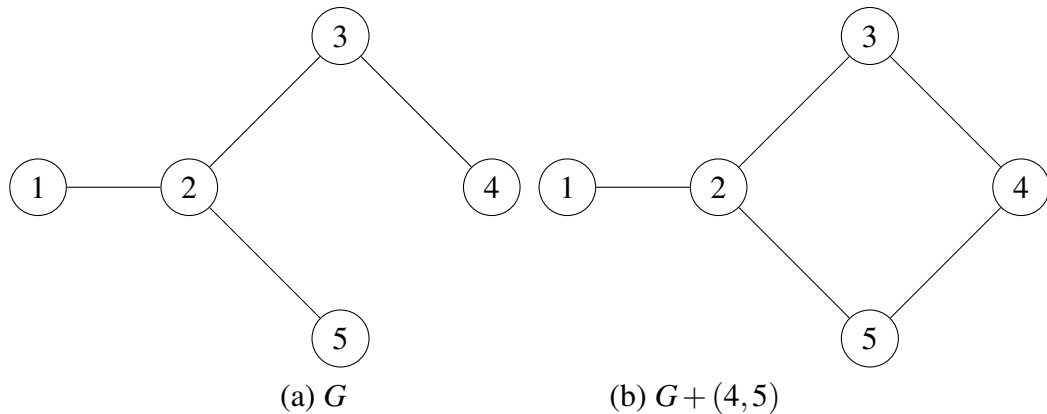
Figura 5 – Exemplo em que $hn(G)$ aumenta com adição de aresta



Fonte: elaborado pelo autor.

Tanto na convexidade geodésica, como na P_3^* , temos que $hn(G) = 2$, no grafo original da Figura 5(a), onde $S^* = \{2, 3\}$. Ao adicionarmos a aresta (2,3) na Figura 5(b), passamos a ter $hn(G + (2,3)) = 3$. Note que $hn(G)$ aumentou com a adição de uma aresta. Todavia, isso nem sempre acontece. A adição de arestas em G , em vez de aumentar $hn(G)$, pode diminuir, como na Figura 6.

Figura 6 – Exemplo que $hn(G)$ diminui com adição de aresta na convexidade Geodésica



Fonte: elaborado pelo autor.

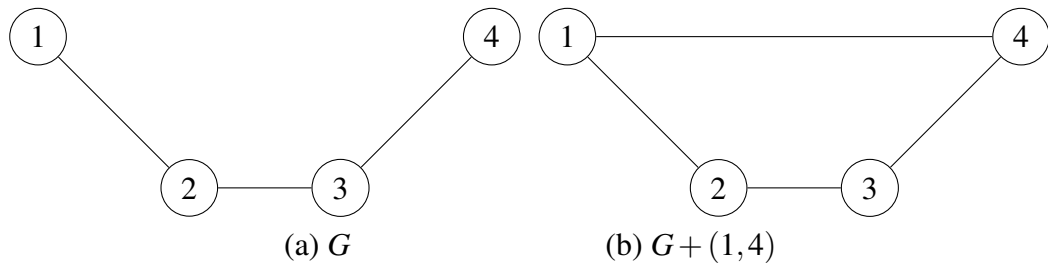
Na Figura 6(a), temos que $hn(G) = 3$, com $S^* = \{1, 4, 5\}$. Adicionando a aresta (4,5)(Figura 6(b)), temos que $hn(G) = 2$, com $S^* = \{1, 4\}$, na convexidade geodésica, ou seja, nesse caso $hn(G)$ diminuiu com a adição de uma aresta. Também para convexidade P_3^* , o exemplo da Figura 7 mostra que $hn(G)$ pode diminuir com a adição de uma aresta.

Na Figura 7(a), temos que $hn(G) = 3$, com $S^* = \{1, 3, 4\}$, na convexidade P_3^* . Adicionando a aresta (1,4)(Figura 7(b)), temos $hn(G) = 2$, para $S^* = \{1, 3\}$. Ou seja, a adição da aresta (1,4) diminuiu $hn(G)$.

Por outro lado, vale a seguinte propriedade:

Propriedade 4: Na convexidade P_3 , a adição (respectivamente remoção) de uma aresta não pode

Figura 7 – Exemplo em que $hn(G)$ diminui com adição de aresta na convexidade P_3^*



Fonte: elaborado pelo autor.

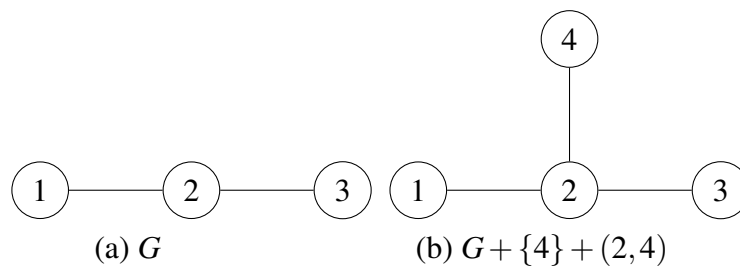
aumentar (respectivamente reduzir) o número de envoltória.

Prova. Seja S^* um conjunto de envoltória mínimo em uma convexidade de caminhos não induzidos em um grafo G . Como todos os caminhos de G permanecem em $G + (u, v)$, S^* continua conjunto de envoltória neste grafo modificado. Logo, $hn(G + (u, v)) \leq hn(G)$. Isto também demonstra o caso de remoção de aresta. ■

2.4.5 Adição ou Remoção de Vértices

De modo similar à adição ou remoção de arestas, a adição ou remoção de vértices pode tanto aumentar quanto diminuir $hn(G)$. No Figura 8(a) observe que $hn(G) = 2$ nas convexidades geodésica, P_3 e P_3^* . Ao adicionarmos o vértice 4 e a aresta $(2,4)$ (Figura 8(b)), $hn(G) = 3$.

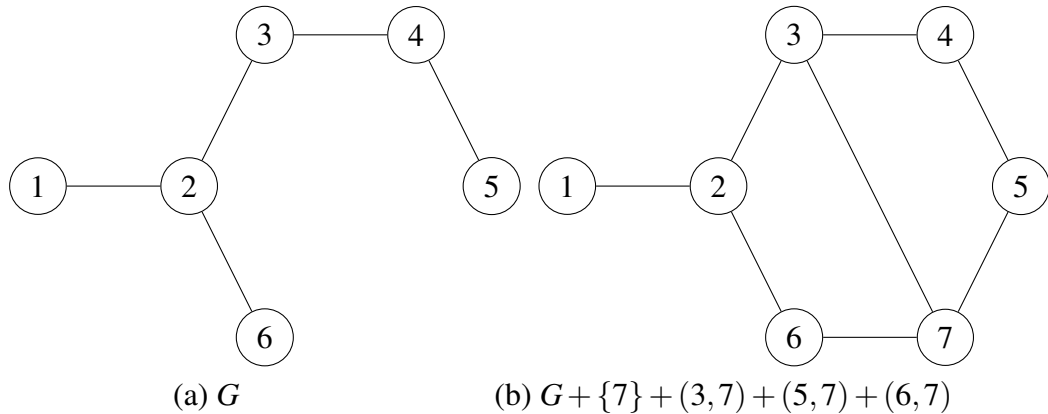
Figura 8 – Exemplo em que $hn(G)$ aumenta com adição de vértice



Fonte: elaborado pelo autor.

Observe agora o exemplo da Figura 9. Temos que $hn(G) = 3$, na convexidade geodésica com $S^* = \{1, 5, 6\}$, e $hn(G) = 4$, nas convexidades P_3 e P_3^* , com $S^* = \{1, 4, 5, 6\}$. Com a adição do vértice 7, temos que $hn(G) = 2$, na convexidade geodésica, com $S^* = \{1, 5\}$, e $hn(G) = 3$, nas convexidades P_3 e P_3^* , com $S^* = \{1, 4, 6\}$.

Figura 9 – Exemplo em que $hn(G)$ diminui com adição de vértice



Fonte: elaborado pelo autor.

2.4.6 Passo de Contaminação

O processo de contaminação ou infecção é realizado iterativamente, através da função de intervalo. Formalmente, dado um inteiro p não negativo, denote por I^p a aplicação p vezes (passos) da função de intervalo I , ou seja, $I^0(S) = I(S)$ e $I^p(S) = I(I^{p-1}(S))$. Então $S_p(G)$ denota um elemento do conjunto $\operatorname{argmin}\{|S| : I^p(S) = V(G)\}$ e $hn_p(G) = |S_p(G)|$. Evidentemente, a partir de um conjunto S de vértices inicialmente contaminados, pode-se infectar mais vértices quando se permite realizar mais passos de contaminação. Dessa maneira, podemos pensar em definir uma variante do problema onde se deseja encontrar o menor conjunto de vértices $S_p(G)$ que contamine todo o grafo em até p passos, para $p \in \{0, 1, \dots, n-2\}$, $n = |V|$. Denotamos por $hn_p(G)$ a cardinalidade de $S_p(G)$. Conforme observado, temos que:

$$n = hn_0(G) \geq hn_1(G) \geq \dots \geq hn_{n-2}(G) = hn(G)$$

Note que precisamos de no máximo $n-2$ passos para obter $hn(G)$. Observe também que pelo menos 2 vértices precisam ser inicialmente infectados.

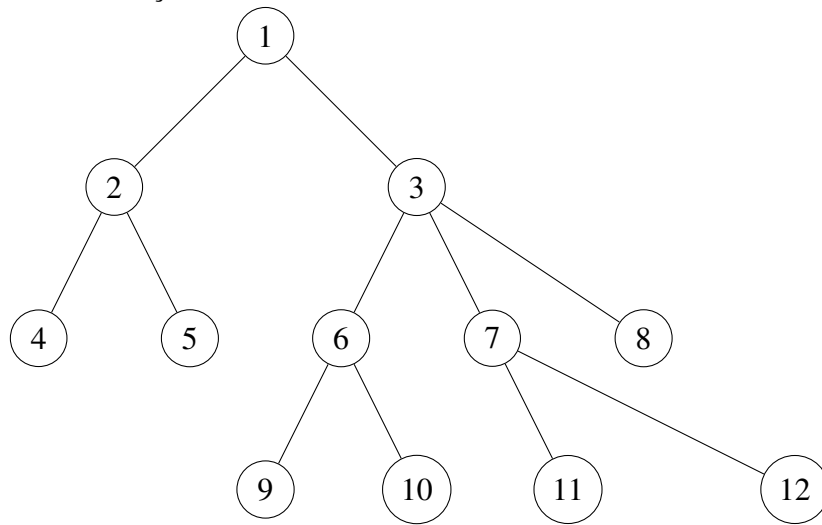
Poder-se-ia então conjecturar que, se $hn_p(G) = hn_{p+1}(G)$, para algum $p \in \{0, 1, \dots, n-3\}$, então teríamos $hn_p(G) = hn(G)$.

Na Figura 10 mostramos um contra-exemplo, para essa conjectura, para as convexidades P_3 e P_3^* .

Observe que $hn(G) = 7$. Basta selecionarmos os 7 vértices folhas que conseguiremos gerar todo o grafo. Nas Figuras 11 a 14, mostraremos o comportamento de $hn_p(G)$ ao fixarmos P em 0, 1, 2 e 3, respectivamente. Os vértices de $S_p(G)$ estão destacados em cor mais escura.

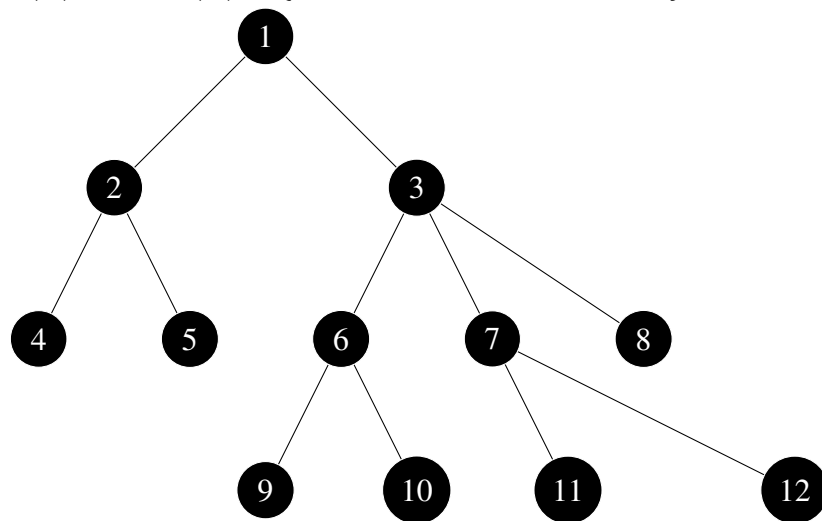
Vemos que, se $P = 1$ e $P = 2$, então $hn_p(G) = 8$, mas $hn_3(G) = hn(G) = 7$. Com

Figura 10 – Contra-exemplo para monotonicidade do número de envoltória com relação ao tempo de contaminação



Fonte: elaborado pelo autor.

Figura 11 – $hn_0(G) = 12$, $S_0(G) = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$

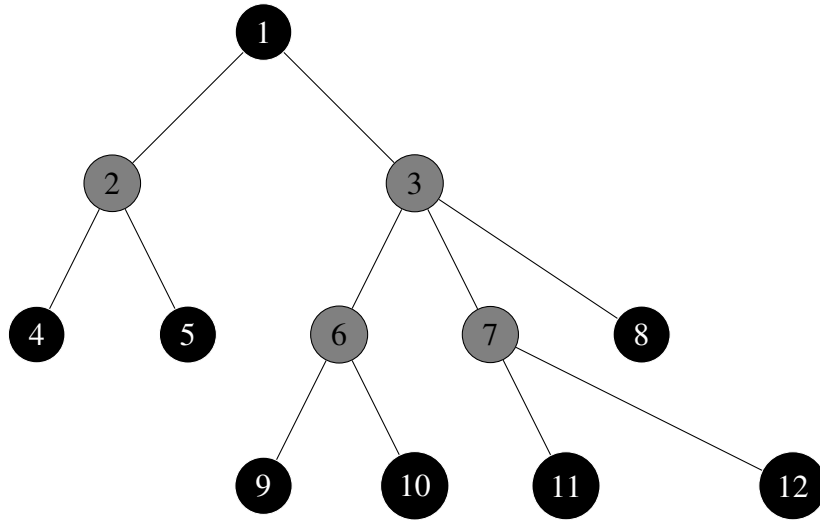


Fonte: elaborado pelo autor.

esse exemplo, podemos observar que, por mais que $hn_p(G)$ não mude para valores consecutivos de p , não podemos afirmar que a solução encontrada é a solução ótima.

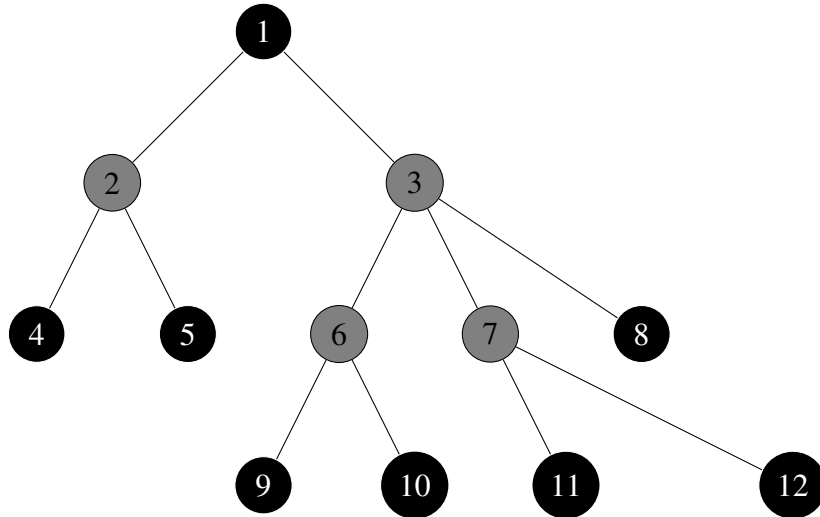
Para a convexidade geodésica, não conseguimos encontrar um contra-exemplo para esta mesma propriedade, do mesmo modo, que não chegamos a mostrar que ela é verdadeira.

Figura 12 – $hn_1(G) = 8, S_1(G) = \{1, 4, 5, 8, 9, 10, 11, 12\}$



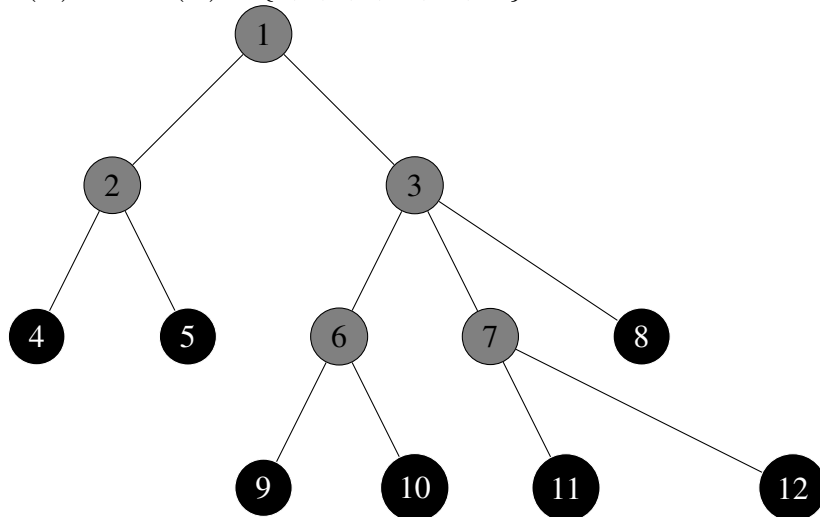
Fonte: elaborado pelo autor.

Figura 13 – $hn_2(G) = 8, S_2(G) = \{1, 4, 5, 8, 9, 10, 11, 12\}$



Fonte: elaborado pelo autor.

Figura 14 – $hn_3(G) = 7, S_3(G) = \{4, 5, 8, 9, 10, 11, 12\}$



Fonte: elaborado pelo autor.

3 CONTAMINAÇÃO

O processo de convexidade de intervalo em grafos também é equivalentemente tratado como um processo de contaminação ou infecção. Quando dizemos que um par de vértices i e j geram um certo vértice k em uma convexidade, também podemos dizer que o vértice k foi infectado ou contaminado pelos vértices i e j . Veja que esse processo pode modelar na prática questões como a determinação da menor quantidade de indivíduos para disseminar informações em uma rede social, ou a menor quantidade de indivíduos para disseminar uma doença ou epidemia em um local, por exemplo.

Tome o cenário a seguir que remete a um exemplo prático de aplicação da convexidade P_3 :

"Considere a situação em que uma pessoa é convencida a fazer parte de um grupo (ou comprar um produto, ou visitar um site) sempre que pelo menos k de seus amigos também fizerem parte deste grupo. O caso em que $k = 2$ é modelado fielmente na convexidade P_3 . Chamemos de convertidos aqueles indivíduos que fazem parte do grupo. Estes podem ter sido adicionados ao grupo em um momento inicial (após alguma campanha publicitária, por exemplo) ou podem ter entrado após constatarem a presença de amigos no grupo. Neste contexto, o número de envoltória equivale ao problema de encontrar a menor quantidade de pessoas que devem ser convencidas inicialmente a fazer parte do grupo de forma que, uma vez que tempo suficiente tenha transcorrido, todos os membros da rede social sejam convertidos (SANTOS, 2013)".

Neste capítulo, propomos algoritmos que representam e auxiliam esse processo de contaminação sobre as três convexidades estudadas: geodésica, P_3 e P_3^* .

3.1 Caminhos Mínimos

Vimos no Capítulo 1 que um caminho de menor comprimento possível em um grafo é conhecido como caminho mínimo. O problema de determinar um i, j -caminho mínimo é fácil de se resolver em grafos com ponderação não negativa nas arestas, ou seja, existem algoritmos polinomiais para encontrar caminhos mínimos em um grafo sob essas condições. Alguns exemplos de algoritmos polinomiais são os algoritmos de Dijkstra, Bellman-Ford, Floyd-Warshall, etc, que podem ser encontrados em Cormen *et al.* (2009). Como os grafos utilizados em nosso trabalho tem peso 1 em todas as arestas, a utilização desses algoritmos é viável.

Para convexidade geodésica, necessitamos determinar não apenas o custo de um caminho mínimo entre cada par de vértices de G , como também os vértices internos a todos esses caminhos mínimos, pois se dois vértices estão contaminados em algum instante, eles infectam todos os vértices internos em qualquer caminho mínimo entre os dois.

O clássico algoritmo de Floyd-Warshall (FLOYD, 1962; WARSHALL, 1962) consegue determinar o custo de todos os caminhos mínimos entre todos os pares de vértices de um grafo. Vamos modificá-lo para determinar também todos os vértices internos em algum caminho mínimo entre pares de vértices de G .

Para a entrada do algoritmo, tem-se uma matriz $D_{n \times n}$, $n = |V(G)|$, onde a posição D_{ij} representa o custo da aresta ij , caso exista, e infinito caso contrário. Ao final do procedimento, cada posição D_{ij} representa o custo de um i, j -caminho mínimo. Como estamos interessados não apenas nos custos mas também nos próprios caminhos mínimos, adicionamos uma matriz ao algoritmo, $H_{n \times n}$, onde cada posição H_{ij} armazenará os vértices que estão internos a algum i, j -caminho mínimo, para cada par $i, j \in V(G)$. Em outros termos H_{ij} é o conjunto de vértices infectados por i e j na convexidade geodésica, ou ainda, a função de intervalo correspondente aplicada a $\{i, j\}$. Para tanto, usamos o Algoritmo 1.

Algoritmo 1: Floyd-Warshall-Modificado($D_{n \times n}$)

```

1: for  $i \leftarrow 1$  to  $n$  do
2:   for  $j \leftarrow 1$  to  $n$  do
3:      $H_{ij} \leftarrow \emptyset$ 
4:     for  $k \leftarrow 1$  to  $n$  do
5:       if  $D_{ij} > D_{ik} + D_{kj}$  then
6:          $D_{ij} \leftarrow D_{ik} + D_{kj}$ 
7:          $H_{ij} \leftarrow H_{ik} \cup H_{kj} \cup \{k\}$ 
8:       else
9:         if  $D_{ij} = D_{ik} + D_{kj}$  then
10:           $H_{ij} \leftarrow H_{ij} \cup H_{ik} \cup H_{kj} \cup \{k\}$ 
11:        end if
12:      end if
13:    end for
14:  end for
15: end for
16: return  $D_{n \times n}, H_{n \times n}$ 

```

3.2 Vizinhança

Diferentemente da convexidade geodésica, nas convexidades P_3 e P_3^* , dois vértices infectados i e j , em determinado instante, contaminam todos os seus vizinhos em comum, independente da existência ou não da aresta (i, j) (para P_3), ou contaminam tais vizinhos apenas quando não existe a aresta (i, j) (para P_3^*). Da mesma forma do algoritmo anterior, podemos gerar a matriz $H_{n \times n}$ para essas duas convexidades em tempo polinomial, onde H_{ij} novamente é o conjunto de vértices infectados por i e j .

Para determinarmos certo conjunto H_{ij} , basta olharmos para a vizinhança de i e de j . Para todo vértice k , tal que $(i, k), (k, j) \in E(G)$, adicionamos a H_{ij} o vértice k em P_3 , e só adicionamos em P_3^* se $(i, j) \notin E(G)$. No Algoritmo 2 temos um pseudo-código do procedimento.

Algoritmo 2: Vizinhança(G)

```

1: for  $i \leftarrow 1$  to  $n$  do
2:   for  $j \leftarrow 1$  to  $n$  do
3:      $H_{ij} \leftarrow \emptyset$ 
4:     for  $k \leftarrow 1$  to  $n$  do
5:       if  $(i, k), (k, j) \in E(G)$  then
6:          $H_{ij} \leftarrow H_{ij} \cup \{k\}$ 
7:       end if
8:     end for
9:   end for
10: end for
11: return  $H_{n \times n}$ 

```

O Algoritmo 2 determina a matriz $H_{n \times n}$ para a convexidade P_3 . Caso o interesse esteja na convexidade P_3^* , basta alterarmos a condição do passo 5 para:

$$(u, k), (k, v) \in E(G) \text{ e } (u, v) \notin E(G) \quad (3.1)$$

3.3 Algoritmo de Contaminação

Em nossa heurística e em um dos modelos, precisamos calcular a envoltória convexa de um dado conjunto S de vértices de um grafo G , para, por exemplo, verificar se ele é um conjunto de envoltória. Em Dourado *et al.* (2009), é mencionado um algoritmo polinomial para calcular tal envoltória. A partir dessa sugestão, foi implementado um algoritmo de contaminação, que recebe como entrada um subconjunto S , correspondendo àqueles já contaminados, e um

tipo de convexidade, representada pela função de intervalo, e retorna todo o conjunto resultante da contaminação (na convexidade pré-determinada), aplicada a S . No Algoritmo 3 temos o pseudo-código do procedimento.

Algoritmo 3: Algoritmo de Contaminação (S, I)

```

1:  $S' \leftarrow S$ 
2:  $U \leftarrow \emptyset$ 
3: while  $U \neq S'$  do
4:    $U \leftarrow \{I(\{i, j\}) : \{i, j\} \in S'\}$ 
5:    $S' \leftarrow S' \cup U$ 
6: end while
7: return  $S'$ 

```

Observe que a função I atua sempre sob um par de vértices (i, j) . Essa função determina a convexidade estudada, sendo definida nas três convexidades como:

- Geodésica: $I(i, j)$ retorna os vértices internos em quaisquer caminhos mínimos entre i e j
- P_3 : $I(i, j)$ retorna todos os vértices simultaneamente vizinhos de i e j
- P_3^* : $I(i, j)$ retorna todos os vértices simultaneamente vizinhos de i e j , se $(i, j) \notin E(G)$.

Vale ressaltar que os algoritmos 1 e 2 servem de base para esse Algoritmo de Contaminação. Mais especificamente $I(\{i, j\})$ é o conjunto H_{ij} , onde H é a matriz retornada pelo Algoritmo 1 (convexidade geodésica), Algoritmo 2 (convexidade P_3) e Algoritmo 2 com condição (3.1) (convexidade P_3^*).

3.4 Decomposição em blocos

O Teorema 2.4.1 estabelece que podemos determinar o número de envoltória de G a partir do número de envoltória generalizado de seus blocos. Este resultado sugere uma decomposição do problema, segundo a árvore bloco-articulação do grafo.

Há diferentes algoritmos para determinar a árvore bloco-articulação de um grafo. Em geral, eles se baseiam em uma busca em profundidade. O algoritmo clássico foi proposto em Hopcroft e Tarjan (1973). Neste trabalho, usamos a implementação disponível em Geek (2018).

A ideia geral do algoritmo é efetuar uma busca em profundidade (DFS), mantendo as seguintes informações:

- a profundidade de cada vértice na árvore DFS (ou seja, o passo em que é visitado pela primeira vez);

- o ponto baixo de cada vértice v , ou seja, a menor das profundidades dos vizinhos de todos os descendentes de v (incluindo o próprio v) na árvore; o ponto baixo de v pode ser calculado após visitar todos os descendentes de v , como o mínimo entre a profundidade de v , a profundidade de todos os vizinhos de v (diferentes do pai de v) e o ponto baixo dos filhos de v na árvore.

Quando u não for o vértice raiz, ele será uma articulação se, e somente se, possuir filho v tal que o ponto baixo de v seja maior ou igual à profundidade de u . Nesse caso, um bloco é dado por u e a subárvore de v , incluindo o próprio v . No caso em que u for a raiz, ele é uma articulação se, e somente se, possuir pelo menos dois filhos. Nesse caso, a subárvore de cada filho, junto com a raiz, induz um bloco.

Um pseudo código desse procedimento pode ser visto no Algoritmo 4, que usa a função recursiva descrita no Algoritmo 5. Além de determinarmos as articulações, também identificamos todas as arestas de cada bloco. Uma pilha é usada para manter essas arestas.

Algoritmo 4: Decomposição em Blocos(G)

```

1:  pilha ←  null
2:  nblocos ← 0 // Quantidade de blocos
3: for  $i \leftarrow 1$  to  $n$  do
4:    $prof_i \leftarrow -1$  // Profundidade de  $i$ 
5:    $pb_i \leftarrow -1$  // Ponto baixo
6:    $pai_i \leftarrow -1$  // Profundidade do pai de  $i$ 
7: end for
8: for  $i \leftarrow 1$  to  $n$  do
9:   if  $prof_i = -1$  then
10:     Bloco( $i, 0, prof, pb, pai, nblocos$ )
11:   end if
12:   if  pilha  $\langle \rangle$   null then
13:     Remova as arestas da pilha, formando bloco
14:      $nblocos \leftarrow nblocos + 1$ 
15:   end if
16: end for

```

Algoritmo 5: Bloco($u, pf, prof, pb, pai, pilha, nblocos$)

```

1:  $prof_u \leftarrow pb_u$ 
2:  $pb_u \leftarrow pf$ 
3:  $nfilhos \leftarrow 0$  // Número de filhos
4: for  $v \in N(u)$  do
5:   if  $prof_v = -1$  then
6:      $nfilhos \leftarrow nfilhos + 1$ 
7:      $pai_v \leftarrow u$ 
8:     Empilhe  $(u, v)$  em pilha
9:     Bloco( $v, prof_u + 1, prof, pb, pai, pilha, nblocos$ )
10:     $pb_u \leftarrow \min(pb_u, pb_v)$ 
11:    if ( $prof_u = 0$  and  $nfilhos > 1$ ) or ( $prof_u > 0$  and  $pb_v \geq prof_u$ ) then
12:      Identifique  $u$  como articulação
13:      Desempilhe todas as arestas até  $(u, v)$ , formando bloco
14:       $nblocos \leftarrow nblocos + 1$ 
15:    end if
16:  else
17:    if  $v \neq pai_u$  then
18:       $pb_u = \min(pb_u, prof_v)$ 
19:      if  $prof_v < prof_u$  then
20:        Empilhe  $(u, v)$  em pilha
21:      end if
22:    end if
23:  end if
24: end for

```

4 HEURÍSTICA

Modelos matemáticos nem sempre são a estratégia mais rápida para resolução de problemas de otimização. Apesar de nos retornarem a solução ótima de determinado problema, às vezes podem exigir muito tempo computacional para serem solucionados. Quando essa situação ocorre, geralmente temos interesse na utilização de métodos heurísticos, que, apesar de não garantirem encontrar a melhor solução possível, retornam uma solução rapidamente e, dependendo da qualidade do método, podem retornar soluções bem próximas à solução ótima do problema. Neste capítulo, são propostas duas heurísticas gulosas: uma geral, que pode ser utilizada para diferentes tipos de convexidades de caminhos, e uma específica para convexidade geodésica. Ambas as heurísticas são baseadas em estratégias gulosas.

4.1 Para convexidades de intervalo

A heurística consiste em uma estratégia gulosa, que constrói um conjunto de envoltória S^* iterativamente. A cada iteração busca sempre contaminar a maior quantidade de vértices possível no grafo, ao adicionar um novo vértice ao conjunto S^* . Para inicializar o processo, adicionamos a esse conjunto S^* um par de vértices, a partir do qual, aplicado o processo de contaminação e fixada uma convexidade, será contaminado o maior número de vértices possível no grafo, no instante posterior. Aplicamos o algoritmo de contaminação a esse par de vértices e obtemos a envoltória convexa C desse conjunto inicial S^* . Caso o conjunto C não seja o próprio conjunto de vértices do grafo, então, nesse instante, o conjunto S^* ainda não é de envoltória. Assim, precisamos adicionar um vértice $k \in V(G) - C$ ao conjunto S^* . A escolha desse vértice é semelhante à inicial do método. Precisamente, determinamos o par (k, l) , com $l \in C$ e $k \in V \setminus C$, que contaminará o maior número de vértices, no próximo instante. Voltamos ao processo de contaminação, determinando a nova envoltória convexa de C , e continuamos até que o conjunto C seja o próprio $V(G)$. No Algoritmo 6, apresentamos um pseudocódigo para esse procedimento, que recebe como entrada um grafo G e uma determinada convexidade I .

Vale lembrar que, para as convexidades geodésica, P_3 e P_3^* , o conjunto $I(i, j)$ é dado pela entrada H_{ij} da matriz H , retornada pelos Algoritmos 1 ou 2.

Algoritmo 6: Heurística-Geral (G, I)

```

1: Encontre  $\{i, j\} \subset V(G)$ , tal que  $|I(i, j)|$  é máxima em  $G$ 
2:  $S^* \leftarrow \{i, j\}$ 
3:  $C \leftarrow$  Algoritmo de Contaminação( $S^*, I$ )// Algoritmo 3
4: while  $C \neq V(G)$  do
5:   Encontre  $k \in V(G) - C$  e  $l \in C$ , tais que  $|I(k, l)|$  é máxima
6:    $S^* \leftarrow S^* \cup \{k\}$ 
7:    $C \leftarrow$  Algoritmo de Contaminação( $S^*, I$ )// Algoritmo 3
8: end while
9: return  $|S^*|$ 

```

4.2 Específica para Convexidade Geodésica

Na convexidade geodésica temos que dois vértices i e j , contaminados em determinado instante, contaminam quaisquer vértices em todos os i, j -caminhos mínimos no instante posterior. Na heurística apresentada anteriormente, nosso interesse era adicionar ao conjunto S^* um vértice $k \in V(G) - C$, tal que, para algum $l \in C$, a quantidade de vértices nos vários k, l -caminhos mínimos é máxima no grafo G .

Propomos agora uma ideia gulosa diferente para essa convexidade, onde buscamos adicionar a S^* os vértices que são extremidades dos maiores caminhos mínimos em G . Para tal, alteramos o passo 1 do Algoritmo 6, onde agora encontramos os vértices (i, j) , tais que $dist_G(i, j)$ é máxima em G , e alteramos o passo 5, de tal modo que adicionamos em S^* o vértice k , onde $dist_G(k, C)$ é máxima em G . Nessa descrição, $dist_G(i, j)$ representa o comprimento de um i, j -caminho mínimo em G , e $dist_G(k, C)$ representa o mínimo entre $dist_G(k, l)$, para todo $l \in C$. O pseudo-código desse novo procedimento está descrito no Algoritmo 7.

Algoritmo 7: Heurística Específica-Convexidade Geodésica (G)

```

1: Encontre  $\{i, j\} \subset V(G)$ , tal que  $dist_G(i, j)$  é máxima em  $G$ 
2:  $S^* \leftarrow \{i, j\}$ 
3:  $C \leftarrow$  Algoritmo de Contaminação( $S^*, 'Geodésica'$ )
4: while  $C \neq V(G)$  do
5:   Encontre  $k \in V(G) - C$ ,  $l \in C$ , tais que  $\min_{l \in C} dist_G(k, l)$  é máxima
6:    $S^* \leftarrow S^* \cup \{k\}$ 
7:    $C \leftarrow$  Algoritmo de Contaminação( $S^*, 'Geodésica'$ )
8: end while
9: return  $|S^*|$ 

```

5 FORMULAÇÕES

Neste capítulo, apresentaremos dois modelos de programação linear inteira para determinar o número de envoltória de um grafo. O primeiro é compacto (possui um número polinomial de variáveis e restrições); o segundo possui um número linear de variáveis, porém exponencial de restrições, em termos do número $n = |V(G)|$ de vértices do grafo.

5.1 Passo de Contaminação

O primeiro modelo, denominado passo de contaminação, é baseado no tempo para contaminação, ou seja, número de passos necessários para a infecção de todo o grafo. Observe que, se o grafo possuir pelo menos três vértices e não for o grafo completo, então $hn(G) \leq n - 1$ e, em no máximo $n - 2$ iterações, os demais vértices serão infectados. Defina:

- $P = n - 2$, em que P é o tempo máximo para contaminar todo o grafo;
- Variáveis:

$$x_i^p = \begin{cases} 1, & \text{se } i \in V \text{ está contaminado no passo } p = 0, 1, \dots, P; \\ 0, & \text{caso contrário.} \end{cases}$$

Como o nosso objetivo é determinar o menor número de vértices que, inicialmente contaminados, infectam iterativamente todo o grafo, então queremos que as variáveis x_i^0 somem o mínimo possível. Portanto, temos a seguinte função objetivo, que queremos minimizar:

$$\sum_{i \in V} x_i^0$$

Para modelar a contaminação de um vértice k em determinado instante $p + 1$, temos duas possibilidades: ou ele está contaminado desde o início; ou no instante p dois vértices i e j estão contaminados e k está em algum i, j -caminho. As características desse caminho são determinadas pela convexidade escolhida. Logo, temos a seguinte restrição:

$$x_k^{p+1} \leq \sum_{\{i,j\} \in H_k} x_i^p x_j^p + x_k^0, \quad \forall k \in V, p = 0, 1, \dots, P - 1 \quad (5.1)$$

onde H_k é o conjunto contendo todos os pares de vértices $\{i, j\}$, tais que k está em algum i, j -caminho. Tais conjuntos podem ser determinados a partir da saída do Algoritmo 1 (convexidade geodésica) ou Algoritmo 2 (convexidades P_3 e P_3^*).

Mais precisamente $H_k = \{\{i, j\} : k \in H_{ij}\}$, onde H é a matriz gerada por esses algoritmos.

Queremos propor um modelo de programação linear-inteira, então não podemos ter termos do tipo $x_i^p x_j^p$, como aparecem nas restrições (5.1). Para linearizarmos essa expressão, criaremos uma nova variável y_{ij}^p , que represente esse produto, onde:

$$y_{ij}^p = \begin{cases} 1, & \text{se } i, j \in V \text{ estão contaminados no passo } p = 0, 1, \dots, P; \\ 0, & \text{caso contrário.} \end{cases}$$

Em outros termos, queremos $y_{ij}^p = x_i^p x_j^p$. Adicionamos à princípio as seguintes restrições para relacionar as variáveis x_i^p e x_j^p com y_{ij}^p :

$$y_{ij}^p \leq x_i^p, y_{ij}^p \leq x_j^p \quad \forall i, j \in V, p = 0, 1, \dots, P \quad (5.2)$$

$$y_{ij}^p - x_i^p - x_j^p + 1 \geq 0 \quad \forall i, j \in V, p = 0, 1, \dots, P \quad (5.3)$$

Por causa da função objetivo, podemos remover esse segundo grupo de restrição, e mesmo assim termos garantia da continuidade da relação $y_{ij}^p = x_i^p x_j^p$ na solução ótima. Ao final, alteramos a restrição que modela a contaminação de um vértice k em certo instante $p + 1$ da seguinte forma:

$$x_k^{p+1} \leq \sum_{\{i,j\} \in H_k} y_{ij}^p + x_k^0, \quad \forall k \in V, p = 0, 1, \dots, P - 1.$$

A garantia de que o conjunto contaminado inicialmente é de envoltória equivale a pedir que o processo de infecção definido acima resulte em todos os vértices contaminados no último instante. Em outros termos, devemos ter:

$$x_i^P = 1, \quad \forall i \in V.$$

Resumindo, temos o seguinte modelo:

$$\min \sum_{i \in V} x_i^0 \quad (5.4)$$

$$\text{s.a: } x_i^P = 1, \quad \forall i \in V. \quad (5.5)$$

$$y_{ij}^p \leq x_i^p, y_{ij}^p \leq x_j^p, \quad \forall i, j \in V, p = 0, 1, \dots, P. \quad (5.6)$$

$$x_k^{p+1} \leq \sum_{\{i,j\} \in H_k} y_{ij}^p + x_k^0, \quad \forall k \in V, p = 0, 1, \dots, P - 1. \quad (5.7)$$

$$x_i^p \in \{0, 1\}, \quad \forall i \in V, p = 0, 1, \dots, P. \quad (5.8)$$

$$y_{ij}^p \in \{0, 1\}, \quad \forall i, j \in V, p = 0, 1, \dots, P. \quad (5.9)$$

Vale mencionar que as restrições $x_i^{p+1} \geq x_i^p$, para todo $i \in V$ e todo $p = 0, \dots, P-1$, que garantiriam que cada vértice permaneceria contaminado a partir do primeiro instante em que fosse contaminado, não são necessárias para a corretude do modelo. De fato, se alguma solução viável do modelo acima tiver $x_i^p = 1$ e $x_i^{p+1} = 0$, podemos trocar $x_i^{p+1} = 1$ e obter outra solução viável de igual valor. Destacamos ainda que a integralidade das variáveis y pode ser descartada do modelo. De fato, se (x, y) é uma solução viável que satisfaz as restrições do modelo, então existe solução (x, y') , com $y \in \mathbb{B}^{|E|^p}$, de mesmo valor.

O modelo acima tem um número cúbico, com relação ao número de vértices do grafo dado como entrada, de variáveis e de restrições. Os valores das variáveis x_i^0 indicarão ao final da resolução do modelo se o vértice i participa de um conjunto de envoltória mínimo do grafo e o número de envoltória será a soma de todas essas variáveis.

A implementação desse modelo é direta, dados os métodos mostrados anteriormente. Todos os componentes do modelo (função objetivo, restrições e variáveis) são de ordem polinomial, de modo que podemos gerá-lo em tempo polinomial.

Na verdade, a partir das propriedades 1, 2 e 3, podemos fixar inicialmente as seguintes variáveis: $x_i^p = 1$, para todo vértice i simplicial e para todo p , $x_i^p = 1$, para todo vértice i folha e para todo p (convexidade geodésica e P_3^*), $x_i^0 = 0$, para todo vértice de articulação, bem como adicionar as restrições $\sum_{i \in B} x_i^0 \geq 1$, para todo bloco folha não trivial (na convexidade geodésica). Alternativamente às duas últimas restrições, o Teorema 2.4.1 estabelece que podemos aplicar o modelo apenas em cada bloco B do grafo. Neste caso, devemos fazer $x_i^0 = 1$, para toda articulação de B e remover essas variáveis da função objetivo. Onde B^* é o conjunto de vértices que não são articulação de B . Precisamente o modelo aplicado a um bloco B é:

$$\min \sum_{i \in B^*} x_i^0 \tag{5.10}$$

$$\text{s.a: } x_i^0 = 1, \quad \forall i \in B - B^*. \tag{5.11}$$

$$x_i^{|B|} = 1, \quad \forall i \in B. \tag{5.12}$$

$$y_{ij}^p \leq x_i^p, y_{ij}^p \leq x_j^p, \quad \forall i, j \in B, p = 0, 1, \dots, |B^*|. \tag{5.13}$$

$$x_k^{p+1} \leq \sum_{\{i,j\} \in H_k} y_{ij}^p + x_k^0, \quad \forall k \in B, p = 0, 1, \dots, |B^*| - 1. \tag{5.14}$$

$$x_i^p \in \{0, 1\}, \quad \forall i \in B, p = 0, 1, \dots, |B^*|. \tag{5.15}$$

$$y_{ij}^p \in \{0, 1\}, \quad \forall i, j \in B, p = 0, 1, \dots, |B^*| - 2. \tag{5.16}$$

5.2 Coconvexo

Dado um grafo G , relembramos que um conjunto $S \subseteq V(G)$ é *coconvexo* se $V(G) \setminus S$ é convexo. Pela propriedade (ii) de espaços de convexidade, deduzimos que pelo menos um vértice de cada coconvexo S precisa estar infectado inicialmente; do contrário, como a envoltória de $V(G) \setminus S$ é o próprio conjunto, nenhum dos vértices de S se tornaria contaminado. Na verdade, essa condição necessária também é suficiente para definir um conjunto de envoltória. Em outros termos, podemos deduzir que S' é um conjunto de envoltória se, e somente se, para todo conjunto coconvexo S , temos que $S' \cap S \neq \emptyset$ (DOURADO *et al.*, 2009). Ou seja, um conjunto de envoltória pode ser visto como um transversal da família de coconvexos do grafo dado. Esse é o elemento chave da formulação a seguir, onde usamos as seguintes variáveis:

$$x_i = \begin{cases} 1, & \text{se } i \in V(G) \text{ é selecionado para o conjunto de envoltória;} \\ 0, & \text{caso contrário.} \end{cases}$$

A função objetivo, que queremos minimizar, é trivialmente dada por:

$$\sum_{i \in V} x_i.$$

Para a descrição das restrições, defina $CC(G)$ como a família de conjuntos coconvexos de G . Pela observação supramencionada, temos:

$$\sum_{i \in S} x_i \geq 1, \quad \forall S \in CC(G).$$

Logo, o modelo completo é:

$$\min \sum_{i \in V} x_i \tag{5.17}$$

$$\text{s.a: } \sum_{i \in S} x_i \geq 1 \quad \forall S \in CC(G) \tag{5.18}$$

$$x_i \in \{0, 1\} \quad \forall i \in V(G) \tag{5.19}$$

Vale mencionar que o método implementado em Cohen (2018) resolve implicitamente esse modelo. Em outras palavras, ele pode ser visto como a aplicação do método de planos-de-corte à formulação acima.

Pode-se diferenciar o modelo co-convexo do anterior tanto em número de variáveis como em número de restrições. Este apresenta uma vantagem quanto ao número de variáveis, que é linear em função do número de vértices, enquanto no anterior tal quantidade era cúbica.

Entretanto, o modelo coconvexo apresenta uma desvantagem: um número exponencial de restrições, que, na prática, não podem ser fornecidas diretamente ao solver.

Para a implementação do modelo (5.17)-(5.19), usamos a seguinte estratégia. Geramos inicialmente apenas uma pequena parte das restrições, ou simplesmente a restrição $\sum_{i \in V} x_i \geq 1$. Então, usamos o CPLEX para resolução do modelo relaxado. Aqui, o termo relaxado se refere à adição de apenas um subconjunto das restrições (5.18) junto com a integralidade das variáveis. Verificamos se a solução inteira produzida define um conjunto de envoltória. Para tal, utilizamos o algoritmo de contaminação (Algoritmo 3) proposto anteriormente. Caso afirmativo, temos uma solução ótima. Do contrário, o subconjunto de vértices não contaminados é coconvexo e leva a uma restrição violada. Essa é acrescentada ao modelo relaxado corrente e o processo, repetido. Note que a integralidade não é relaxada em nenhum momento. Um pseudocódigo desse procedimento é apresentado no Algoritmo 8.

Algoritmo 8: Resolução coconvexo

```

1:  $C \leftarrow \emptyset$ 
2: repeat
3:    $S \leftarrow V(G) \setminus C$ 
4:   Adicione ao modelo a restrição (5.18) relativa a  $S$ 
5:   Resolva o Modelo
6:    $S' \leftarrow$  Vértices indicados pela solução
7:    $C \leftarrow$  Algoritmo de Contaminação( $S'$ )
8: until  $C = V(G)$ 
9: return  $S$ 

```

Similarmente ao apresentado para o modelo passo de contaminação, podemos também fixar variáveis no modelo coconvexo, a partir das propriedades 1, 2 e 3: $x_i = 1$, para todo vértice i folha, $x_i = 1$, para todo vértice simplicial (convexidades geodésica e P_3^*). Também podemos adicionar as restrições $\sum_{i \in B} x_i \geq 1$, para todo bloco folha não trivial (convexidade geodésica). Alternativamente, podemos usar o Teorema 2.4.1 e restringir o modelo (5.17) - (5.19) a cada bloco do grafo, fixando em 1 as variáveis associadas às articulações e removendo-as da função objetivo.

6 EXPERIMENTOS COMPUTACIONAIS

Neste capítulo, iremos apresentar os resultados computacionais obtidos pelos métodos anteriormente propostos. Para os modelos matemáticos apresentaremos o tempo necessário para resolução. Já para a heurística, estaremos mais interessados no desvio percentual da solução encontrada para a solução ótima.

6.1 Instâncias de Teste

Para os testes computacionais, foram utilizados grafos gerados aleatoriamente, arbitrários ou bipartidos. Note que o estudo nessa segunda classe é relevante, uma vez que mesmo em bipartidos a determinação do número de envoltória é *NP*-difícil nas convexidades estudadas.

Para gerar cada instância, usamos dois parâmetros de entrada: a quantidade de vértices $n \in \mathbb{N}^*$; e uma probabilidade $0 \leq p \leq 1$ relativa à ocorrência de aresta. Para gerar grafos bipartidos, separamos o conjunto de vértices em dois subconjuntos, A com $\lceil n/2 \rceil$ vértices e B com os $\lfloor n/2 \rfloor$ restantes; em seguida, de modo a garantir a conexidade do grafo, criamos uma árvore geradora qualquer com esse conjunto de vértices (mantendo a propriedade de ser bipartido com as partes A e B). Posteriormente, para todo par de vértices $i, j \in V(G)$, tais que $i \in A$ e $j \in B$, se a aresta (i, j) não for uma das que formaram a árvore inicialmente, geramos um número aleatório entre 0 e 1. Caso esse número seja menor que o fator de probabilidade p , então adicionamos a aresta (i, j) . Note que, quanto maior esse fator p , maior será a quantidade de arestas do grafo, o que se reflete diretamente na densidade do mesmo. Para gerarmos grafos aleatórios quaisquer, seguimos o mesmo procedimento anterior, sendo que formamos uma árvore geradora qualquer com os n vértices e, depois, geramos uma aresta entre cada par $i, j \in V(G)$, com probabilidade p .

Para cada um dos dois tipos de grafos considerados, geramos dois grupos de instâncias. O primeiro, que chamaremos instâncias padrão, possuem $n \in \{10, 20, 40, 80\}$ e $p \in \{0.2, 0.5, 0.9\}$. Para cada par (n, p) foram geradas 5 instâncias, totalizando 60. Elas serão usadas em todos os experimentos, com exceção daqueles apresentados na Subseção 6.6.3, quando avaliamos o desempenho do modelo coconvexo em função de n e p . Nesse caso, usamos instâncias maiores ou mais esparsas. Precisamente, geramos 5 instâncias para cada combinação $n \in \{10, 20, \dots, 250\}$ e $p \in \{0.2, 0.5, 0.9\}$, totalizando 375 instâncias, assim como 5 instâncias com $n=100$ para cada $p \in \{0.01, 0.02, 0.03, \dots, 0.20\}$, totalizando 100 instâncias.

Os modelos foram resolvidos com uso do CPLEX (IBM, Versão 12.6.1). A implementação dos métodos descritos ao longo do texto foi toda feita na linguagem de programação C++, por meio do compilador QT-Creator, versão 3.0.1. Os experimentos foram realizados em um processador Intel Celeron(R) CPU N2830 com 2.16 GHz, 4 GB RAM e sistema operacional Ubuntu versão 17.04.

Para cada teste, na coluna relativa ao tempo de execução do método avaliado, a ocorrência do símbolo '-' significa que o método utilizou o tempo limite que determinamos, que foi de 1000 segundos, e não conseguiu encontrar a solução.

Comparamos os desempenhos dos dois modelos matemáticos e da heurística. Os resultados obtidos estão descritos nas seções a seguir.

6.2 Testes com Heurística

Nesta seção, apresentamos os resultados obtidos pela heurística. Os primeiros testes foram feitos com conjunto de instâncias que denominamos instâncias padrão. Observamos, principalmente, o desvio médio percentual da solução ótima em cada um dos testes e também contabilizamos os tempos para cada teste.

6.2.1 Testes nas instâncias padrão

Nas tabelas 2 e 3, estão contabilizados os resultados obtidos pela heurística (Algoritmo 6) para as convexidades de intervalo descritas no Capítulo 4. A primeira tabela refere-se aos grafos bipartidos e, a segunda, a grafos arbitrários. As colunas n e p indicam o número de vértices do grafo e o fator de probabilidade, para ocorrência de arestas. Para cada convexidade, as colunas 'm o', 'm h' e 't' correspondem a, respectivamente, a média das soluções ótimas, a média das soluções obtidas pela heurística e média do tempo por ela gasto. Cada linha corresponde à média de cinco instâncias geradas com um mesmo par (n, p) . Também calculamos o desvio percentual, dado por $\frac{mh-mo}{mo} \times 100$, e apresentamos o desvio médio.

A partir dos resultados obtidos, contidos nas Tabelas 2 e 3, observamos que, nas três convexidades, as heurísticas apresentaram um bom desempenho. Além de encontrarem a solução ótima na maior parte das instâncias, apresentaram um baixo desvio percentual médio. Em média, as heurísticas erraram em menos de um vértice para cada grupo de instâncias, o que mostra que o erro é pequeno para o número de envoltória.

Tabela 2 – Resultados da heurística para convexidades de intervalo em grafos Bipartidos

| Teste | n | p | Geodésica | | | P3 | | | P3* | | |
|-------|----|-----|---------------|-----|--------|---------------|-----|--------|---------------|-----|--------|
| | | | m o | m h | t | m o | m h | t | m o | m h | t |
| 1 | 10 | 0,9 | 2 | 2 | 0,0028 | 2 | 2 | 0,0004 | 2 | 2 | 0,0004 |
| 2 | 10 | 0,5 | 2 | 2,2 | 0,0022 | 2,8 | 2,8 | 0,0003 | 2,8 | 2,8 | 0,0003 |
| 3 | 10 | 0,2 | 3,4 | 3,8 | 0,0022 | 3,6 | 4,4 | 0,0004 | 3,6 | 4,4 | 0,0003 |
| 4 | 20 | 0,9 | 2 | 2 | 0,0205 | 2 | 2 | 0,0022 | 2 | 2 | 0,0018 |
| 5 | 20 | 0,5 | 2 | 2 | 0,015 | 2 | 2 | 0,0014 | 2 | 2 | 0,0012 |
| 6 | 20 | 0,2 | 3 | 3,8 | 0,0108 | 3,2 | 4 | 0,0012 | 3,2 | 4 | 0,0008 |
| 7 | 40 | 0,9 | 2 | 2 | 0,1775 | 2 | 2 | 0,0085 | 2 | 2 | 0,0090 |
| 8 | 40 | 0,5 | 2 | 2 | 0,1132 | 2 | 2 | 0,0054 | 2 | 2 | 0,0052 |
| 9 | 40 | 0,2 | 2 | 2,8 | 0,0658 | 2,8 | 2,8 | 0,0038 | 2,8 | 2,8 | 0,0028 |
| 10 | 80 | 0,9 | 2 | 2 | 1,6544 | 2 | 2 | 0,0455 | 2 | 2 | 0,0489 |
| 11 | 80 | 0,5 | 2 | 2 | 0,9695 | 2 | 2 | 0,0255 | 2 | 2 | 0,0258 |
| 12 | 80 | 0,2 | 2 | 2 | 0,4876 | 2 | 2 | 0,0145 | 2 | 2 | 0,0106 |
| | | | Desvio: 7,37% | | | Desvio: 4,29% | | | Desvio: 4,29% | | |

Fonte: elaborado pelo autor.

Tabela 3 – Resultados da heurística para convexidades de intervalo em grafos Arbitrários

| Teste | n | p | Geodésica | | | P3 | | | P3* | | |
|-------|----|-----|---------------|-----|--------|---------------|-----|--------|---------------|-----|--------|
| | | | m o | m h | t | m o | m h | t | m o | m h | t |
| 1 | 10 | 0,9 | 2 | 2 | 0,0028 | 2 | 2 | 0,0025 | 2 | 2 | 0,0027 |
| 2 | 10 | 0,5 | 2,6 | 2,8 | 0,0023 | 2 | 2 | 0,0024 | 2,6 | 2,8 | 0,0025 |
| 3 | 10 | 0,2 | 3,8 | 3,8 | 0,0022 | 2,4 | 2,8 | 0,0025 | 4,2 | 4,8 | 0,0021 |
| 4 | 20 | 0,9 | 2 | 2 | 0,0206 | 2 | 2 | 0,0231 | 2 | 2 | 0,0239 |
| 5 | 20 | 0,5 | 2 | 2 | 0,0147 | 2 | 2 | 0,0201 | 2 | 2 | 0,0232 |
| 6 | 20 | 0,2 | 2,4 | 3,2 | 0,0109 | 2,2 | 2,4 | 0,0153 | 3 | 3,2 | 0,0211 |
| 7 | 40 | 0,9 | 2 | 2 | 0,1742 | 2 | 2 | 0,1721 | 2 | 2 | 0,1821 |
| 8 | 40 | 0,5 | 2 | 2 | 0,1127 | 2 | 2 | 0,1129 | 2 | 2 | 0,1257 |
| 9 | 40 | 0,2 | 2 | 2,2 | 0,0662 | 2 | 2 | 0,0523 | 2,2 | 2,2 | 0,0426 |
| 10 | 80 | 0,9 | 2 | 2 | 1,6531 | 2 | 2 | 1,5132 | 2 | 2 | 1,6284 |
| 11 | 80 | 0,5 | 2 | 2 | 0,9748 | 2 | 2 | 1,0192 | 2 | 2 | 0,9993 |
| 12 | 80 | 0,2 | 2 | 2 | 0,4877 | 2 | 2 | 0,5271 | 2 | 2 | 0,7623 |
| | | | Desvio: 4,25% | | | Desvio: 2,14% | | | Desvio: 2,38% | | |

Fonte: elaborado pelo autor.

6.2.2 Heurística para Convexidades de Intervalo × Específica para Convexidade Geodésica

Nas tabelas 4 e 5 estamos comparando o desempenho da heurística para convexidades de intervalo (Algoritmo 6), aplicada a convexidade geodésica, e a heurística específica para essa convexidade (Algoritmo 7). As colunas identificadas por HI e Tempo HI se referem à solução e tempo, respectivamente, da primeira, enquanto HG e Tempo HG se referem às correspondentes para segunda. A Tabela 4 apresenta os resultados par grafos bipartidos e a Tabela 5 para grafos

arbitrários. Cada linha de cada tabela corresponde à média de 5 instâncias.

As heurísticas mostraram um comportamento similar, tanto com relação ao tempo de execução, quanto às soluções encontradas. Entretanto, pode-se notar que a heurística HG apresentou soluções um pouco melhores que a heurística HI, em média.

Tabela 4 – Heurística Geral × Específica para grafos bipartidos

| Teste | n | p | Ótimo | HI | Tempo HI | HG | Tempo HG |
|-------|-----|-----|-------|---------------|----------|---------------|----------|
| 1 | 10 | 0,9 | 2 | 2 | 0,0028 | 2 | 0,0032 |
| 2 | 10 | 0,5 | 2 | 2,2 | 0,0022 | 2,4 | 0,0023 |
| 3 | 10 | 0,2 | 3,4 | 3,8 | 0,0022 | 3,8 | 0,0022 |
| 4 | 20 | 0,9 | 2 | 2 | 0,0205 | 2 | 0,0208 |
| 5 | 20 | 0,5 | 2 | 2 | 0,015 | 2 | 0,0148 |
| 6 | 20 | 0,2 | 3 | 3,8 | 0,0108 | 3,4 | 0,0109 |
| 7 | 40 | 0,9 | 2 | 2 | 0,1775 | 2 | 0,1758 |
| 8 | 40 | 0,5 | 2 | 2 | 0,1132 | 2 | 0,1145 |
| 9 | 40 | 0,2 | 2 | 2,8 | 0,0658 | 2,8 | 0,0668 |
| 10 | 80 | 0,9 | 2 | 2 | 1,6544 | 2 | 1,6724 |
| 11 | 80 | 0,5 | 2 | 2 | 0,9695 | 2 | 0,9976 |
| 12 | 80 | 0,2 | 2 | 2 | 0,4876 | 2 | 0,4931 |
| | | | | Desvio: 7,37% | | Desvio: 7,09% | |

Fonte: elaborado pelo autor.

Tabela 5 – Heurística Geral × Específica para grafos arbitrários

| Teste | n | p | Ótimo | HI | Tempo HI | HG | Tempo HG |
|-------|-----|-----|-------|---------------|----------|---------------|----------|
| 1 | 10 | 0,9 | 2 | 2 | 0,0028 | 2 | 0,0022 |
| 2 | 10 | 0,5 | 2,6 | 2,8 | 0,0023 | 2,6 | 0,0023 |
| 3 | 10 | 0,2 | 3,8 | 3,8 | 0,0022 | 3,8 | 0,0021 |
| 4 | 20 | 0,9 | 2 | 2 | 0,0206 | 2 | 0,0123 |
| 5 | 20 | 0,5 | 2 | 2 | 0,0147 | 2 | 0,0142 |
| 6 | 20 | 0,2 | 2,4 | 3,2 | 0,0109 | 3,2 | 0,0111 |
| 7 | 40 | 0,9 | 2 | 2 | 0,1742 | 2 | 0,0851 |
| 8 | 40 | 0,5 | 2 | 2 | 0,1127 | 2 | 0,1023 |
| 9 | 40 | 0,2 | 2 | 2,2 | 0,0662 | 2,2 | 0,0658 |
| 10 | 80 | 0,9 | 2 | 2 | 1,6531 | 2 | 0,7938 |
| 11 | 80 | 0,5 | 2 | 2 | 0,9748 | 2 | 0,8512 |
| 12 | 80 | 0,2 | 2 | 2 | 0,4877 | 2 | 0,3904 |
| | | | | Desvio: 4,25% | | Desvio: 3,61% | |

Fonte: elaborado pelo autor.

6.3 Testes com os Modelos

Nesta seção, apresentaremos os resultados obtidos pelas modelagens matemáticas, apresentadas no Capítulo 5. Inicialmente iremos comparar as eficiências dos dois modelos, usando as instâncias padrão. Após, avaliaremos algumas variações de ambos, procurando incrementar seu desempenho quanto à modelagem, além de considerarmos instâncias com características variadas em relação à n e p .

6.3.1 Passo de Contaminação \times coconvexo

6.3.1.1 Testes com modelos básicos

Inicialmente, comparamos o desempenho dos dois modelos nas mesmas instâncias usadas com as heurísticas. O único pré-processamento feito foi a fixação de variáveis que correspondiam a vértices k tais que $H_k = \emptyset$. Esses vértices só podem ser infectados por si mesmos, ou seja, devem ser escolhidos inicialmente. Os resultados obtidos, na convexidade geodésica, estão representados nas tabelas 6 e 7, respectivamente, para grafos bipartidos e arbitrários. 't p', 't c' e 'o' representam, respectivamente, o tempo do modelo passo de contaminação em segundos, o tempo do modelo co-convexo em segundos e a solução ótima. Cada linha corresponde a média dos resultados de 5 instâncias.

Tabela 6 – Resultados dos modelos para grafos bipartidos

| Teste | V | p | Geodésica | | | P3 | | | P3* | | |
|-------|----|-----|-----------|--------|-----|----------|--------|------|----------|--------|------|
| | | | t p | t c | o | t p | t c | o | t p | t c | o |
| 1 | 10 | 0,9 | 0,6930 | 0,0047 | 2 | 3,3862 | 0,0025 | 2 | 3,4023 | 0,0026 | 2 |
| 2 | 10 | 0,5 | 0,4579 | 0,0189 | 2 | 3,4579 | 0,1069 | 2,8 | 3,5329 | 0,1061 | 2,8 |
| 3 | 10 | 0,2 | 0,2206 | 0,1202 | 3,4 | 1,0482 | 0,3153 | 3,6 | 1,0217 | 0,3375 | 3,6 |
| 4 | 20 | 0,9 | 21,3609 | 0,0231 | 2 | 145,4570 | 0,0034 | 2 | 142,3291 | 0,0038 | 2 |
| 5 | 20 | 0,5 | 49,4237 | 0,0169 | 2 | 80,8663 | 0,0029 | 2 | 81,7386 | 0,0034 | 2 |
| 6 | 20 | 0,2 | 468,9021 | 0,1118 | 3 | 827,8213 | 1,1275 | 3,2 | 832,9299 | 1,1529 | 3,2 |
| 7 | 40 | 0,9 | - | 0,1760 | 2 | - | 0,0066 | 2 | - | 0,0112 | 2 |
| 8 | 40 | 0,5 | - | 0,1136 | 2 | - | 0,0096 | 2 | - | 0,0074 | 2 |
| 9 | 40 | 0,2 | - | 0,0881 | 2 | - | 0,0963 | 2,8 | - | 0,0992 | 2,8 |
| 10 | 80 | 0,9 | - | 1,6577 | 2 | - | 0,0446 | 2 | - | 0,0522 | 2 |
| 11 | 80 | 0,5 | - | 0,9693 | 2 | - | 0,0246 | 2 | - | 0,0293 | 2 |
| 12 | 80 | 0,2 | - | 0,4872 | 2 | - | 0,0119 | 2 | - | 0,0138 | 2 |
| Média | | | 90,1764 | 0,3156 | 2,2 | 177,0061 | 0,1460 | 2,37 | 177,4924 | 0,1516 | 2,37 |

Fonte: elaborado pelo autor.

A partir dos dados apresentados nas tabelas 6 e 7, observamos que o modelo co-convexo foi mais rápido, em média, que o modelo passo de contaminação em todos os testes

Tabela 7 – Resultados dos modelos para grafos arbitrários

| Teste | V | p | Geodésica | | | P3 | | | P3* | | |
|-------|----|-----|-----------|--------|------|----------|--------|------|---------|--------|------|
| | | | t p | t c | o | t p | t c | o | t p | t c | o |
| 1 | 10 | 0,9 | 1,2107 | 0,0667 | 2 | 5,2044 | 0,0031 | 2 | 0,8934 | 0,0265 | 2 |
| 2 | 10 | 0,5 | 3,3397 | 0,3083 | 2,6 | 6,3281 | 0,0025 | 2 | 2,4003 | 0,0677 | 2,6 |
| 3 | 10 | 0,2 | 5,1678 | 1,1811 | 3,8 | 7,9933 | 0,0455 | 2,4 | 1,3909 | 0,5048 | 4,2 |
| 4 | 20 | 0,9 | 51,5048 | 0,1224 | 2 | 62,9604 | 0,0054 | 2 | 24,3481 | 0,0507 | 2 |
| 5 | 20 | 0,5 | 294,8680 | 0,2665 | 2 | 38,6230 | 0,0042 | 2 | 61,9071 | 0,0043 | 2 |
| 6 | 20 | 0,2 | 365,4821 | 0,4546 | 2,4 | 544,8800 | 0,0043 | 2,2 | - | 0,2595 | 3 |
| 7 | 40 | 0,9 | - | 1,0032 | 2 | - | 0,0224 | 2 | - | 0,2679 | 2 |
| 8 | 40 | 0,5 | - | 2,2261 | 2 | - | 0,0125 | 2 | - | 0,0162 | 2 |
| 9 | 40 | 0,2 | - | 2,6820 | 2 | - | 0,0064 | 2 | - | 0,0293 | 2,2 |
| 10 | 80 | 0,9 | - | 2,6574 | 2 | - | 0,1337 | 2 | - | 0,1270 | 2 |
| 11 | 80 | 0,5 | - | 7,0337 | 2 | - | 0,0641 | 2 | - | 0,0725 | 2 |
| 12 | 80 | 0,2 | - | 9,9799 | 2 | - | 0,0231 | 2 | - | 0,0227 | 2 |
| Média | | | 120,2621 | 2,3318 | 2,23 | 110,0245 | 0,0272 | 2,05 | 18,1880 | 0,1208 | 2,33 |

Fonte: elaborado pelo autor.

apresentados, tanto em grafos arbitrários quanto bipartidos. O modelo passo de contaminação não conseguiu encontrar a solução ótima para instâncias a partir de 40 vértices, mostrando que o modelo coconvexo é mais adequado para instâncias maiores. Além disso, podemos notar que o tempo computacional para o modelo coconvexo é maior, para a convexidade geodésica, em grafos arbitrários, em comparação ao tempo para grafos bipartidos, com os mesmos parâmetros n e p .

6.3.1.2 Testes com Decomposição em Blocos

Como vimos, o Teorema 2.4.1 estabelece que podemos determinar o número de envoltória de G a partir do número de envoltória generalizado de seus blocos, na convexidade geodésica. Nos testes apresentados nas tabelas 8 e 9, fazemos a decomposição em blocos do grafo (instância) e resolvemos o problema em cada um dos blocos separadamente, e combinamos a solução. As colunas identificadas por 'p' e 'Dec p' se referem aos tempos, respectivamente, do modelo passo de contaminação e decomposição em blocos com modelo passo de contaminação em cada bloco separadamente. Similarmente, 'c' e 'Dec c' referem-se aos resultados correspondentes para o modelo co-convexo.

Observamos que, para as instâncias padrão, o uso da decomposição em blocos não contribuiu para melhoria de desempenho dos modelos. Na maior parte dos testes, eles demandaram maior tempo computacional para a solução ótima que as versões básicas avaliadas na subseção anterior. Esse resultado indica que, para essas instâncias, a decomposição em blocos não consegue reduzir o esforço computacional para cada modelo encontrar o número de

Tabela 8 – Decomposição em blocos em grafos Bipartidos

| Teste | n | p | p | c | Dec p | Dec c | Solução |
|-------|-----|-----|----------|--------|---------|---------|---------|
| 1 | 10 | 0,9 | 0,6930 | 0,0047 | 0,9612 | 0,0057 | 2 |
| 2 | 10 | 0,5 | 0,4579 | 0,0189 | 0,7531 | 0,0189 | 2 |
| 3 | 10 | 0,2 | 0,2206 | 0,1202 | 0,3590 | 0,1956 | 3,4 |
| 4 | 20 | 0,9 | 21,3609 | 0,0231 | 43,3316 | 0,1071 | 2 |
| 5 | 20 | 0,5 | 49,4237 | 0,0169 | 80,2749 | 0,1192 | 2 |
| 6 | 20 | 0,2 | 468,9021 | 0,1118 | - | 0,2714 | 3 |
| 7 | 40 | 0,9 | - | 0,1760 | - | 0,5947 | 2 |
| 8 | 40 | 0,5 | - | 0,1136 | - | 0,5644 | 2 |
| 9 | 40 | 0,2 | - | 0,0881 | - | 0,0918 | 2 |
| 10 | 80 | 0,9 | - | 1,6577 | - | 2,3729 | 2 |
| 11 | 80 | 0,5 | - | 0,9693 | - | 1,1001 | 2 |
| 12 | 80 | 0,2 | - | 0,4872 | - | 0,9725 | 2 |
| Média | | | 90,1764 | 0,3156 | 25,1360 | 0,5345 | 2,2 |

Fonte: elaborado pelo autor.

Tabela 9 – Decomposição em blocos em grafos Arbitrários

| Teste | n | p | p | c | Dec p | Dec c | Solução |
|-------|-----|-----|----------|--------|----------|---------|---------|
| 1 | 10 | 0,9 | 1,2107 | 0,0667 | 1,0121 | 0,0645 | 2 |
| 2 | 10 | 0,5 | 3,3397 | 0,3083 | 4,6277 | 0,4291 | 2,6 |
| 3 | 10 | 0,2 | 5,1678 | 1,1811 | 6,9923 | 1,2008 | 3,8 |
| 4 | 20 | 0,9 | 51,5048 | 0,1224 | 82,5791 | 0,1392 | 2 |
| 5 | 20 | 0,5 | 294,8680 | 0,2665 | 558,9902 | 0,2781 | 2 |
| 6 | 20 | 0,2 | 365,4821 | 0,4546 | 708,6827 | 0,4412 | 2,4 |
| 7 | 40 | 0,9 | - | 1,0032 | - | 1,3184 | 2 |
| 8 | 40 | 0,5 | - | 2,2261 | - | 2,2555 | 2 |
| 9 | 40 | 0,2 | - | 2,6820 | - | 2,7165 | 2 |
| 10 | 80 | 0,9 | - | 2,6574 | - | 2,7281 | 2 |
| 11 | 80 | 0,5 | - | 7,0337 | - | 8,9213 | 2 |
| 12 | 80 | 0,2 | - | 9,9799 | - | 13,1789 | 2 |
| Média | | | 120,2621 | 2,3318 | 227,1473 | 2,8059 | 2,23 |

Fonte: elaborado pelo autor.

envoltória.

6.3.1.3 Testes com MIP-Start

Uma ideia interessante para reduzir o esforço computacional necessário para resolver modelos de programação inteira é fornecer uma solução viável, a ser usada para antecipar podas na árvore de enumeração do Branch-and-Bound. O CPLEX permite a aplicação dessa estratégia por meio da função MIP-Start. Em nosso caso, encontramos uma solução viável, usando a heurística de convexidade de intervalo, nas três convexidades, e a informamos ao CPLEX como

ponto viável. Os resultados obtidos com essa estratégia estão apresentados nas Tabelas 10 e 11, respectivamente, para grafos bipartidos e arbitrários.

Tabela 10 – MIP-Start para grafos Bipartidos

| teste | n | p | Geodésica | | | | P_3 | | | | P_3^* | | | |
|-------|----|-----|-----------|---------|--------|--------|----------|---------|--------|--------|----------|----------|--------|--------|
| | | | t p | mip p | co | mip co | t p | mip p | co | mip co | t p | mip p | co | mip co |
| 1 | 10 | 0,9 | 0,693 | 0,659 | 0,0047 | 0,009 | 3,3862 | 3,5552 | 0,0025 | 0,0039 | 3,4023 | 3,4134 | 0,0026 | 0,0037 |
| 2 | 10 | 0,5 | 0,4579 | 0,4697 | 0,0189 | 0,0084 | 3,4579 | 3,449 | 0,1069 | 0,1038 | 3,5329 | 3,5872 | 0,1061 | 0,1278 |
| 3 | 10 | 0,2 | 0,2206 | 0,2092 | 0,1202 | 0,1286 | 1,0482 | 0,5691 | 0,3153 | 0,3514 | 1,0217 | 0,6558 | 0,3375 | 0,3389 |
| 4 | 20 | 0,9 | 21,3609 | 20,3726 | 0,0231 | 0,045 | 145,4570 | 109,439 | 0,0034 | 0,0075 | 142,3291 | 109,2481 | 0,0038 | 0,0062 |
| 5 | 20 | 0,5 | 49,4237 | 51,8415 | 0,0169 | 0,0328 | 80,8663 | 92,5191 | 0,0029 | 0,0064 | 81,7386 | 92,0148 | 0,0034 | 0,0046 |
| 6 | 20 | 0,2 | 468,9021 | 852,057 | 0,1118 | 0,1141 | 827,8213 | - | 1,1275 | 1,2073 | 832,9299 | - | 1,1529 | 1,2048 |
| 7 | 40 | 0,9 | - | - | 0,176 | 0,3545 | - | - | 0,0066 | 0,0192 | - | - | 0,0112 | 0,0207 |
| 8 | 40 | 0,5 | - | - | 0,1136 | 0,2297 | - | - | 0,0096 | 0,013 | - | - | 0,0074 | 0,0132 |
| 9 | 40 | 0,2 | - | - | 0,0881 | 0,1556 | - | - | 0,0963 | 0,1066 | - | - | 0,0992 | 0,1044 |
| 10 | 80 | 0,9 | - | - | 1,6577 | 3,3499 | - | - | 0,0446 | 0,0922 | - | - | 0,0522 | 0,1028 |
| 11 | 80 | 0,5 | - | - | 0,9693 | 1,9662 | - | - | 0,0246 | 0,0507 | - | - | 0,0293 | 0,0551 |
| 12 | 80 | 0,2 | - | - | 0,4872 | 0,9877 | - | - | 0,0119 | 0,0278 | - | - | 0,0138 | 0,0251 |

Fonte: elaborado pelo autor.

Tabela 11 – MIP-Start para grafos Arbitrários

| teste | n | p | Geodésica | | | | P_3 | | | | P_3^* | | | |
|-------|----|-----|-----------|---------|--------|--------|----------|---------|--------|--------|---------|---------|--------|--------|
| | | | t p | mip p | co | mip co | t p | mip p | co | mip co | t p | mip p | co | mip co |
| 1 | 10 | 0,9 | 1,2107 | 0,3713 | 0,0667 | 0,0075 | 5,2044 | 4,5361 | 0,0031 | 0,0047 | 0,8934 | 1,0608 | 0,0265 | 0,0264 |
| 2 | 10 | 0,5 | 3,3397 | 0,4689 | 0,3083 | 0,0725 | 6,3281 | 5,6655 | 0,0025 | 0,0041 | 2,4003 | 2,2084 | 0,0677 | 0,0629 |
| 3 | 10 | 0,2 | 5,1678 | 0,2184 | 1,1811 | 0,286 | 7,9933 | 9,3657 | 0,0455 | 0,0502 | 1,3909 | 1,4906 | 0,5048 | 0,4814 |
| 4 | 20 | 0,9 | 51,5048 | 15,4919 | 0,1224 | 0,0273 | 62,9604 | 64,878 | 0,0054 | 0,0099 | 24,3481 | 25,3484 | 0,0507 | 0,0402 |
| 5 | 20 | 0,5 | 294,868 | 26,5832 | 0,2665 | 0,0318 | 38,6230 | 37,0306 | 0,0042 | 0,0074 | 61,9071 | 35,5473 | 0,0043 | 0,0062 |
| 6 | 20 | 0,2 | 365,4821 | 121,89 | 0,4546 | 0,1103 | 544,8800 | - | 0,0043 | 0,0069 | - | 56,5751 | 0,2595 | 0,2471 |
| 7 | 40 | 0,9 | - | - | 1,0032 | 0,1727 | - | - | 0,0224 | 0,0448 | - | - | 0,2679 | 0,2961 |
| 8 | 40 | 0,5 | - | - | 2,2261 | 0,2083 | - | - | 0,0125 | 0,024 | - | - | 0,0162 | 0,0211 |
| 9 | 40 | 0,2 | - | - | 2,682 | 0,1468 | - | - | 0,0064 | 0,0122 | - | - | 0,0293 | 0,0336 |
| 10 | 80 | 0,9 | - | - | 2,6574 | 1,6082 | - | - | 0,1337 | 0,2755 | - | - | 0,1270 | 0,1862 |
| 11 | 80 | 0,5 | - | - | 7,0337 | 1,6931 | - | - | 0,0641 | 0,1315 | - | - | 0,0725 | 0,1171 |
| 12 | 80 | 0,2 | - | - | 9,9799 | 0,7883 | - | - | 0,0231 | 0,0488 | - | - | 0,0227 | 0,0424 |

Fonte: elaborado pelo autor.

Observamos que utilizar a função MIP-Start do CPLEX reduziu o tempo para algumas instâncias, mas também aumentou o tempo de resolução de algumas outras. Em todo caso, mesmo fornecendo uma solução de boa qualidade, o modelo passo de contaminação não conseguiu encontrar soluções para instâncias a partir de 40 vértices, da mesma forma que a resolução sem utilizar MIP-Start. Para o modelo coconvexo o resultado foi similar, o tempo de computação cresceu para a maioria das instâncias, com exceção, para grafos arbitrários, na convexidade geodésica.

6.3.2 Variantes Passo de Contaminação

Na tentativa de melhorar o desempenho do modelo passo de contaminação, consideramos aqui algumas variantes dele. Na primeira delas avaliamos o modelo quadrático, onde em

uma variação montamos o modelo quadrático, inicialmente proposto com as restrições (5.1), e em outra relaxamos as variáveis y , ou seja, retiramos as restrições de integralidade desta variável e priorizamos o branch nas variáveis x_i^0 , pois essas irão definir efetivamente o *hull set*. Note que a integralidade das variáveis x leva à integralidade das variáveis y . Os resultados obtidos, com relação ao tempo, estão nas Tabelas 12 e 13. As colunas indexadas por 't p', 't quad' e 'ycont', se referem ao tempo de computação, respectivamente, para o modelo básico, o modelo quadrático e o modelo com as variáveis y contínuas (priorizando o branch em x_i^0). Cada linha da matriz corresponde a uma média de 5 instâncias.

Tabela 12 – Variantes do modelo passo de contaminação em grafos Bipartidos

| teste | n | p | Geodésica | | | P3 | | | P3* | | |
|-------|-----|-----|-----------|--------|--------|---------|---------|--------|---------|---------|--------|
| | | | t p | t quad | ycont | t p | t quad | ycont | t p | t quad | ycont |
| 1 | 10 | 0,9 | 0,693 | 3,9748 | 0,7519 | 3,3862 | 14,4438 | 0,9921 | 3,4023 | 14,3142 | 0,9664 |
| 2 | 10 | 0,5 | 0,4579 | 1,9497 | 0,7438 | 3,4579 | 5,6372 | 1,3442 | 3,5329 | 5,6636 | 1,3197 |
| 3 | 10 | 0,2 | 0,2206 | 0,6546 | 0,4801 | 1,0482 | 0,7239 | 0,6178 | 1,0217 | 0,726 | 0,6077 |
| 4 | 20 | 0,9 | 21,3609 | - | - | 145,457 | - | - | 142,329 | - | - |
| 5 | 20 | 0,5 | 49,4237 | - | - | 80,8663 | - | - | 81,7386 | - | - |
| 6 | 20 | 0,2 | 468,902 | - | - | 827,821 | - | - | 832,929 | - | - |
| 7 | 40 | 0,9 | - | - | - | - | - | - | - | - | - |
| 8 | 40 | 0,5 | - | - | - | - | - | - | - | - | - |
| 9 | 40 | 0,2 | - | - | - | - | - | - | - | - | - |
| 10 | 80 | 0,9 | - | - | - | - | - | - | - | - | - |
| 11 | 80 | 0,5 | - | - | - | - | - | - | - | - | - |
| 12 | 80 | 0,2 | - | - | - | - | - | - | - | - | - |

Fonte: elaborado pelo autor.

Tabela 13 – Variantes do modelo passo de contaminação em grafos Arbitrários

| teste | n | p | Geodésica | | | P3 | | | P3* | | |
|-------|-----|-----|-----------|--------|--------|---------|----------|----------|---------|--------|--------|
| | | | t p | t quad | ycont | t p | t quad | ycont | t p | t quad | ycont |
| 1 | 10 | 0,9 | 1,2107 | 0,4313 | 0,4016 | 5,2044 | 225,6291 | 139,8951 | 0,8934 | 0,7148 | 0,6681 |
| 2 | 10 | 0,5 | 3,3397 | 0,4238 | 0,5635 | 6,3281 | 80,1723 | 9,9103 | 2,4003 | 1,6162 | 0,9451 |
| 3 | 10 | 0,2 | 5,1678 | 0,5344 | 0,1852 | 7,9933 | 4,0028 | 1,1445 | 1,3909 | 1,142 | 0,8386 |
| 4 | 20 | 0,9 | 51,5048 | - | - | 62,9604 | - | - | 24,3481 | - | - |
| 5 | 20 | 0,5 | 294,868 | - | - | 38,6230 | - | - | 61,9071 | - | - |
| 6 | 20 | 0,2 | 365,482 | - | - | 544,880 | - | - | - | - | - |
| 7 | 40 | 0,9 | - | - | - | - | - | - | - | - | - |
| 8 | 40 | 0,5 | - | - | - | - | - | - | - | - | - |
| 9 | 40 | 0,2 | - | - | - | - | - | - | - | - | - |
| 10 | 80 | 0,9 | - | - | - | - | - | - | - | - | - |
| 11 | 80 | 0,5 | - | - | - | - | - | - | - | - | - |
| 12 | 80 | 0,2 | - | - | - | - | - | - | - | - | - |

Fonte: elaborado pelo autor.

Pelas Tabelas 12 e 13, vemos que as variações propostas para o modelo passo de

contaminação, pela tentativa de aumentar a quantidade das instâncias solucionadas, não surtiram o efeito desejado. Na verdade, houve uma piora: instâncias com 20 vértices que já estavam sendo solucionadas pela versão original, passaram a extrapolar o limite de tempo.

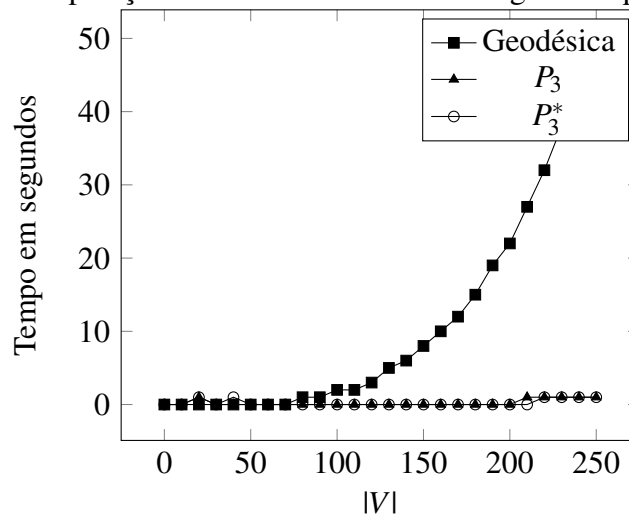
6.3.3 Aprofundamento coconvexo

Agora iremos aprofundar a avaliação do modelo coconvexo, que se mostrou o mais eficiente para as instâncias padrão. Iremos testar o comportamento desse modelo para instâncias com maior quantidade de vértices. Também iremos fazer um estudo do comportamento do modelo, na segunda variante, variando o parâmetro p no intervalo $[0,01;0,2]$. Nos experimentos anteriores, observamos que o modelo tem maior dificuldade para resolver instâncias esparsas.

6.3.3.1 Testes em função do número de vértices

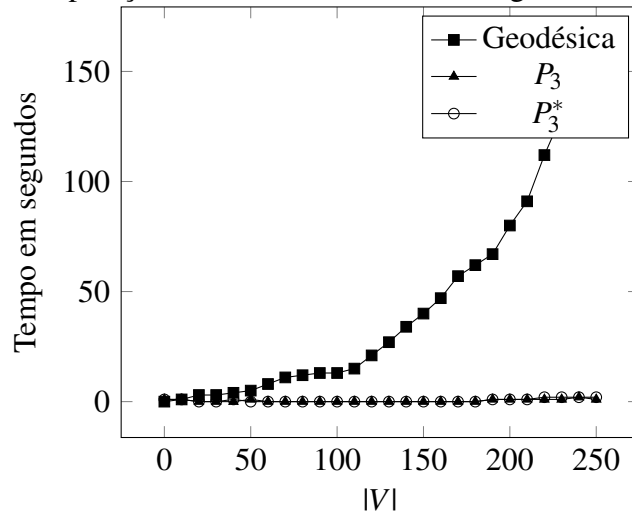
Nos gráficos das Figuras 15 e 16 apresentamos o comportamento do tempo demandado pelo modelo co-convexo com o crescimento de n . Cada ponto no gráfico corresponde à média de 5 instâncias. Observamos que o tempo demandado, tanto para grafos bipartidos, quanto para arbitrários, teve um crescimento similar. Notamos um crescimento exponencial do esforço computacional para a convexidade geodésica, à medida que o número de vértices do grafo aumenta. Já para as convexidades P_3 e P_3^* , o tempo permaneceu praticamente constante.

Figura 15 – Tempo de computação do modelo coconvexo em grafos bipartidos



Fonte: elaborado pelo autor.

Figura 16 – Tempo de computação do modelo coconvexo em grafos arbitrários

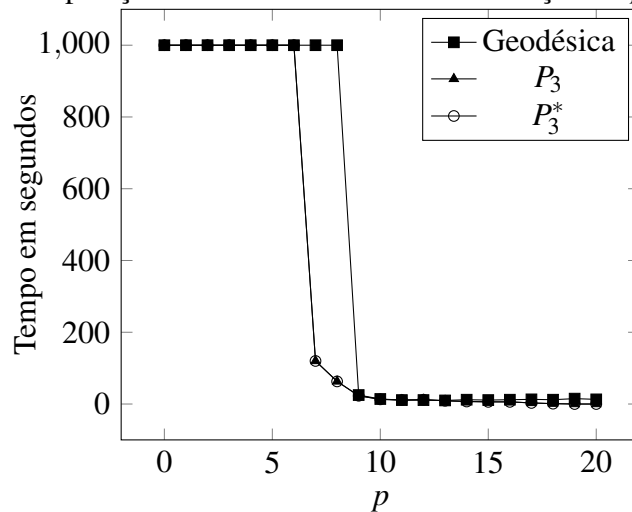


Fonte: elaborado pelo autor.

6.3.3.2 Testes em função da densidade do grafo

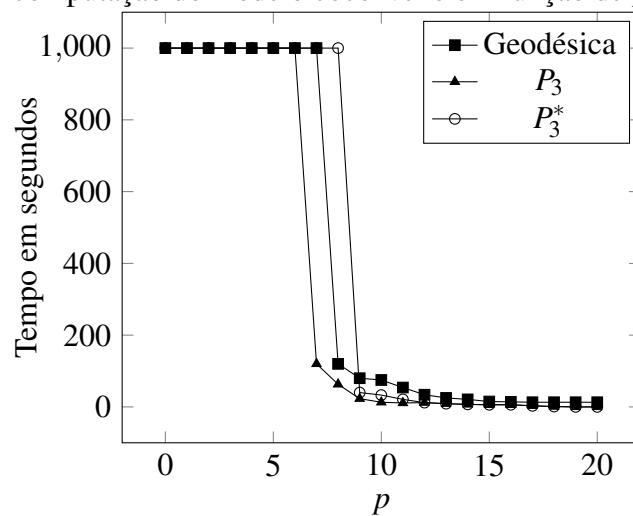
Nos experimentos com instâncias padrão, observamos que o modelo coconvexo, indica que ele demanda mais tempo de computação para grafos gerados com p bem pequeno. Para estudar melhor essa característica, ilustramos nos gráficos das figuras 17 e 18 o tempo de resolução desse modelo quando variamos o parâmetro p no intervalo $[0.01, 0.2]$, para instâncias com $|V|= 100$ fixo. O eixo x corresponde a $p * 100$.

Figura 17 – Tempo de computação do modelo coconvexo em função de p para grafos Bipartidos



Fonte: elaborado pelo autor.

Figura 18 – Tempo de computação do modelo coconvexo em função de p para grafos arbitrários



Fonte: elaborado pelo autor.

Observamos que uma mudança brusca ocorre no tempo demandado pelo modelo coconvexo, para instâncias com p bem próximo a 0.08. Instâncias com p bem próximo ou menor que 0.08 não conseguiram ser resolvidas dentro do tempo limite. Esse comportamento pode ser explicado, pelo fato de que, em grafos esparsos, o tamanho de conjuntos coconvexos é bem maior, gerando restrições mais fracas para o modelo coconvexo. O maior tamanho dos coconvexos nos grafos esparsos deve-se ao menor número de possibilidades de infecção. Com relação às convexidades, notamos que, na convexidade geodésica, o modelo ultrapassou o tempo limite em mais casos que nas convexidades P_3 e P_3^* para grafos bipartidos. Já para grafos arbitrários, a relação de ordem mudou: foram solucionadas menos instâncias em P_3^* , seguida da convexidade geodésica e P_3 , nessa ordem.

7 CONSIDERAÇÕES FINAIS

Neste trabalho, estudamos o problema de determinar o número de envoltória de um grafo nas convexidades geodésica, P_3 e P_3^* . Usamos uma abordagem de programação matemática, ainda inexplorada nesse tema. Apresentamos e comparamos dois modelos de programação inteira e duas heurísticas. Apresentamos uma extensão dos resultados obtidos em Araújo *et al.* (2018a), Campelo *et al.* (2018) e Sousa *et al.* (2018), além de novos resultados para o problema estudado.

A partir dos experimentos mostrados, observa-se que o modelo co-convexo foi mais rápido em quase todas as instâncias que o modelo passo de contaminação, em cada classe de grafos. Isto sugere o modelo co-convexo como o mais adequado para encontrar o conjunto de envoltória para as convexidades estudadas, pois, além de encontrar soluções ótimas, em média, mais rapidamente que o passo de contaminação, não ultrapassou o limite de tempo proposto nos testes, a menos de grafos bastante esparsos, com p abaixo de 0,1. Para problemas reais, onde a quantidade de vértices e arestas do grafo é da ordem de centenas ou milhares, esse modelo parece ser mais adequado, que o modelo passo de contaminação.

Apesar de utilizarmos variações do modelo passo de contaminação, como o modelo quadrático, por exemplo, não conseguimos melhoras significativas no tempo de computação. Na melhor configuração que conseguimos para o modelo, houve instâncias com 40 vértices que não puderam ser solucionadas dentro do tempo limite.

Também foi observado que as heurísticas propostas encontraram a solução ótima em quase todas as instâncias, apresentando um baixo desvio percentual médio da solução ótima. Esse resultado indica que elas podem ser boas alternativas quando queremos utilizar métodos de resolução, que não precisam garantir otimalidade. Além disso, quando comparamos a heurística para convexidade de intervalo, na convexidade geodésica, e a específica para essa convexidade, elas apresentaram desempenhos similares.

Com a utilização da decomposição em blocos para a convexidade geodésica e MIP-Start, não obtivemos resultados promissores com os modelos, nas instâncias testadas, com exceção da utilização de MIP-Start na convexidade geodésica para grafos arbitrários. Em quase todos os testes, a utilização desses pré-processamentos aumentaram o tempo para identificação da solução ótima.

Como trabalhos futuros, há pelo menos duas direções para pesquisa. A primeira compreende um estudo mais aprofundado das formulações apresentadas neste trabalho, em busca de se aplicar métodos de programação inteira mais robustos para resolução de instâncias maiores,

no modelo passo de contaminação, e resolução de instâncias com p relativamente pequeno, no modelo coconvexo. Em particular, pode-se explorar a estrutura dos grafos de entrada. Uma segunda linha de pesquisa consiste em considerar os diversos outros parâmetros de convexidade (número de intervalo, número de Carathéodory, número de Radon, número de Helly, posto, etc.), nas diversas convexidades conhecidas na literatura (Geodésica, P_3 , P_3^* , monofônica, de caminhos triangulados, etc.), com a devida modelagem em programação matemática e a consequente análise de resultados nas mais diversas instâncias.

REFERÊNCIAS

- ALBENQUE, M.; KNAUER, K. Convexity in partial cubes: The hull number. **Discrete Mathematics**, v. 339, n. 2, p. 866 – 876, 2016. ISSN 0012-365X.
- ARAÚJO, J.; CAMPOS, V.; GIROIRE, F.; NISSE, N.; SAMPAIO, L.; SOARES, R. On the hull number of some graph classes. **Theoretical Computer Science**, v. 475, n. Supplement C, p. 1 – 12, 2013. ISSN 0304-3975.
- ARAÚJO, J.; MOREL, G.; SAMPAIO, L.; SOARES, R.; WEBER, V. Hull number: P5-free graphs and reduction rules. **Discrete Applied Mathematics**, v. 210, p. 171 – 175, 2016. ISSN 0166-218X.
- ARAÚJO, J. C. S.; CAMPELO, M. B.; SOUSA, G. H. Calculando o número de envoltória nas convexidades p_3 e p_3^* . 3^o Encontro de Teoria da Computação, p. 1–4, 2018.
- ARAÚJO, R. T.; SAMPAIO, R. M.; SANTOS, V. F.; SZWARCFITER, J. L. The convexity of induced paths of order three and applications: Complexity aspects. **Discrete Applied Mathematics**, p. 33–42, 2018.
- BESSY, S.; DOURADO, M. C.; PENSO, L. D.; RAUTENBACH, D. The geodetic hull number is hard for chordal graphs. **Electronic Notes in Discrete Mathematics**, v. 62, p. 291 – 296, 2017. ISSN 1571-0653.
- CAMPELO, M. B.; ARAÚJO, J. C. S.; SOUSA, G. H. Métodos para determinação do número de envoltória geodésico de um grafo. L Simpósio Brasileiro de Pesquisa Operacional, p. 1–9, 2018.
- COHEN, N. **Convexity properties of graphs on SageMath**. 2018. Disponível em: <http://doc.sagemath.org/html/en/reference/graphs/sage/graphs/convexity_properties.html>.
- CORMEN, T. H.; LEISERSON, C. E.; RIVEST, R. L.; STEIN, C. **Introduction to Algorithms, Third Edition**. 3rd. ed. [S.l.]: The MIT Press, 2009. ISBN 0262033844, 9780262033848.
- DOURADO, M. C.; GIMBEL, J. G.; KRATOCHVÍL, J.; PROTTI, F.; SZWARCFITER, J. L. On the computation of the hull number of a graph. **Discrete Mathematics**, v. 309, n. 18, p. 5668 – 5674, 2009. ISSN 0012-365X.
- DOURADO, M. C.; PENSO, L. D.; RAUTENBACH, D.; SZWARCFITER, J. L. Reversible iterative graph processes. **Theoretical Computer Science**, v. 460, p. 16 – 25, 2012. ISSN 0304-3975.
- DUCHET, P. Convex sets in graphs, ii. minimal path convexity. **Journal of Combinatorial Theory, Series B**, v. 44, n. 3, p. 307 – 316, 1988. ISSN 0095-8956.
- EVERETT, M. G.; SEIDMAN, S. B. The hull number of a graph. **Discrete Mathematics**, v. 57, n. 3, p. 217 – 223, 1985. ISSN 0012-365X.
- FARBER, M.; JAMISON, R. E. Convexity in graphs and hypergraphs. **SIAM J. Algebraic Discrete Methods**, v. 7, p. 433–444, July 1986. ISSN 0196-5212.
- FLOYD, R. W. Algorithm 97: Shortest path. **Commun. ACM**, v. 5, n. 6, jun. 1962. ISSN 0001-0782.

- GEEK. **Biconnected Components**. 2018. Disponível em: <<https://www.geeksforgeeks.org/biconnected-components/>>.
- HOPCROFT, J.; TARJAN, R. Algorithm 447: efficient algorithms for graph manipulation. **Communications of the ACM**, v. 16, p. 372 – 378, 1973.
- MARCILON, T. B. **Resultados no Tempo Máximo e no Número de Envoltória nas Convexidades P_3 e Geodésica**. Tese (Doutorado) — Universidade Federal do Ceará, 2017.
- PELAYO, I. **Geodesic Convexity in Graphs**. [S.l.]: Springer New York, 2013. (SpringerBriefs in Mathematics). ISBN 9781461486992.
- PENSO, L. D.; PROTTI, F.; RAUTENBACH, D.; SOUZA, U. dos S. Complexity analysis of p_3 -convexity problems on bounded-degree and planar graphs. **Theoretical Computer Science**, v. 607, p. 83 – 95, 2015. ISSN 0304-3975.
- SANTOS, V. F. dos. **Convexidades em Grafos: Intermediações, Parâmetros e Conversões**. Tese (Doutorado) — Universidade Federal do Rio de Janeiro, 2013.
- SOUSA, G. H.; CAMPELO, M. B.; ARAÚJO, J. C. S. Comparação de heurísticas para o número de envoltória nas convexidades p_3 e p_3^* . II Encontro Regional de Matemática Aplicada e Computacional, p. 1–3, 2018.
- THOMPSON, J. V. C. **Partições e Convexidades de Caminhos em Grafos**. Tese (Doutorado) — Universidade Federal Fluminense, 2017.
- WARSHALL, S. A theorem on boolean matrices. **J. ACM**, v. 9, n. 1, p. 11–12, jan. 1962. ISSN 0004-5411.
- WEST, D. B. **Introduction to Graph Theory**. 2. ed. [S.l.]: Prentice Hall, 2000. ISBN 0130144002.