

Recoloração convexa de grafos: algoritmos e poliedros

Phablo Fernando Soares Moura

DISSERTAÇÃO APRESENTADA
AO
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA
DA
UNIVERSIDADE DE SÃO PAULO
PARA
OBTENÇÃO DO TÍTULO
DE
MESTRE EM CIÊNCIAS

Programa: Ciência da Computação
Orientadora: Prof^a. Dr^a. Yoshiko Wakabayashi

Durante o desenvolvimento deste trabalho o autor recebeu auxílio financeiro do
CNPq (Proc. 132998/2011-4)

São Paulo, novembro de 2013

Recoloração convexa de grafos: algoritmos e poliedros

Esta versão da dissertação contém as correções e alterações sugeridas pela Comissão Julgadora durante a defesa da versão original do trabalho, realizada em 07/08/2013. Uma cópia da versão original está disponível no Instituto de Matemática e Estatística da Universidade de São Paulo.

Comissão Julgadora:

- Prof^a. Dr^a. Yoshiko Wakabayashi (orientadora) – IME-USP
- Prof. Dr. Carlos Eduardo Ferreira – IME-USP
- Prof. Dr. Manoel Bezerra Campêlo Neto – UFC

Agradecimentos

Em primeiro lugar, agradeço a Deus que tanto me deu forças para superar as dificuldades e esteve comigo em cada passo desta caminhada. “Eis que envio um anjo à frente de vocês para protegê-los por todo o caminho e fazê-los chegar ao lugar que preparei” (Êxodo 23:20).

Aos meus pais, que sempre trabalharam muito para dar a mim, a melhor educação. Um agradecimento muito especial também às minhas irmãs, Flávia e Fabiana. Obrigado a Bruno e a Michelle por todo o apoio e torcida pelo meu sucesso. Jamais teria conseguido sem o apoio e o carinho da minha família. Este trabalho e todos os outros que virão são dedicados a vocês.

Agradeço a minha orientadora, professora Yoshiko que acreditou em meu trabalho, paciência e que foi muito paciente durante todo este tempo. Ao longo desses anos, a minha admiração e respeito, por você, só aumentaram. Muito obrigado pela dedicação, conselhos e ensinamentos.

Ao professor Manoel que me acompanha desde o início da graduação, conduzido os primeiros passos da minha vida acadêmica e que muito ensinou daquilo que hoje sei. Além de professor, é um amigo para toda vida.

Muitos cooperaram para que eu pudesse chegar até aqui. Alguns agradecimentos, a uma pequena parcela dessas pessoas:

Obrigado ao meu amigo e irmão para todas as horas, Maurício, pois ainda que geograficamente distantes, estamos sempre juntos e nossa amizade é cada vez mais forte. Obrigado a toda sua família: Maurício (o pai), Suelly, Marlon e Suellen. Vocês me fazem sentir parte dessa família.

Aos meus amigos da Computação da UFC, Adriano, Rogério, Leví, Samuel e Nicolás. Um obrigado especial aos outros dos “Sete Anões”, Márcio, Arthur, Vinícius, Danusa, Lívia e Tici.

Agradeço também aos professores do ParGO, Ricardo, Rafael, Cláudia e Carlos.

A Júlio (Sr. Araújo), Karol (Sra. Araújo) e ao Ronan, por terem me recebido tão bem durante o meu estágio. Vocês fizeram a minha “aventura” em terras francesas muito mais simples e feliz. Obrigado!

Aos meus colegas de república, Ana, George, Léo e Allan. Mais do que colegas, vocês são amigos que muito ajudaram a superar a saudade da minha família e da minha terra.

Aos colegas e amigos do NUMEC, Schultz, Renato, Fábio, Schouery, Márcio e Álvaro. Obrigado aos cearenses do IME-USP, Guilherme, Bustamante, Phillipe, P.V. e Josefran.

Vocês ajudaram muito quanto à minha adaptação na “terra da garoa”.

Um agradecimento especial à Karla e ao Alexandre que contribuíram enormemente para a realização deste trabalho.

Resumo

Moura, P. F. S. **Recoloração convexa de grafos: algoritmos e poliedros**. 2013. 75 f. Dissertação (Mestrado) - Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2013.

Neste trabalho, estudamos o problema da recoloração convexa de grafos, denotado por RC. Dizemos que uma coloração dos vértices de um grafo G é convexa se, para cada cor atribuída d , os vértices de G com a cor d induzem um subgrafo conexo. No problema RC, é dado um grafo G e uma coloração de seus vértices, e o objetivo é recolorir o menor número possível de vértices de G tal que a coloração resultante seja convexa. A motivação para o estudo deste problema surgiu em contexto de árvores filogenéticas. Sabe-se que este problema é \mathcal{NP} -difícil mesmo quando G é um caminho. Mostramos que o problema RC parametrizado pelo número de mudanças de cor é $\mathcal{W}[2]$ -difícil mesmo se a coloração inicial usa apenas duas cores. Além disso, provamos alguns resultados sobre a inaproximabilidade deste problema. Apresentamos uma formulação inteira para a versão com pesos do problema RC em grafos arbitrários, e então a especializamos para o caso de árvores. Estudamos a estrutura facial do politopo definido como a envoltória convexa dos pontos inteiros que satisfazem as restrições da formulação proposta, apresentamos várias classes de desigualdades que definem facetas e descrevemos os correspondentes algoritmos de separação. Implementamos um algoritmo *branch-and-cut* para o problema RC em árvores e mostramos os resultados computacionais obtidos com uma grande quantidade de instâncias que representam árvores filogenéticas reais. Os experimentos mostram que essa abordagem pode ser usada para resolver instâncias da ordem de 1500 vértices em 40 minutos, um desempenho muito superior ao alcançado por outros algoritmos propostos na literatura.

Palavras-chave: coloração convexa, faceta, formulação linear inteira, estudo poliédrico, *branch-and-cut*, árvore filogenética, inaproximabilidade.

Abstract

Moura, P. F. S. **Convex recoloring of graphs: algorithms and polyhedra**. 2013. 75 f. Dissertação (Mestrado) - Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2013.

In this work we study the convex recoloring problem of graphs, denoted by CR. We say that a vertex coloring of a graph G is convex if, for each assigned color d , the vertices of G with color d induce a connected subgraph. In the CR problem, given a graph G and a coloring of its vertices, we want to find a recoloring that is convex and minimizes the number of recolored vertices. The motivation for investigating this problem has its roots in the study of phylogenetic trees. It is known that this problem is \mathcal{NP} -hard even when G is a path. We show that the problem CR parameterized by the number of color changes is $\mathcal{W}[2]$ -hard even if the initial coloring uses only two colors. Moreover, we prove some inapproximation results for this problem. We also show an integer programming formulation for the weighted version of this problem on arbitrary graphs, and then specialize it for trees. We study the facial structure of the polytope defined as the convex hull of the integer points satisfying the restrictions of the proposed ILP formulation, present several classes of facet-defining inequalities and the corresponding separation algorithms. We also present a branch-and-cut algorithm that we have implemented for the special case of trees, and show the computational results obtained with a large number of instances. We considered instances which are real phylogenetic trees. The experiments show that this approach can be used to solve instances up to 1500 vertices in 40 minutes, comparing favorably to other approaches that have been proposed in the literature.

Keywords: convex coloring, facet, integer linear programming, polyhedral study, branch-and-cut, phylogenetic tree, inapproximability.

Sumário

Lista de Figuras	ix
Lista de Tabelas	xi
1 Introdução	1
1.1 O problema de recoloração convexa (RC)	2
1.2 Resultados conhecidos	3
1.3 Contribuições e organização do trabalho	5
2 Preliminares	7
2.1 Teoria dos grafos	7
2.2 Teoria poliédrica	9
2.3 Algoritmos de aproximação	10
3 Complexidade e algoritmos de aproximação	13
3.1 Complexidade parametrizada	13
3.2 Inaproximabilidade do RC	17
3.3 Uma 4-aproximação para o problema RC em árvores	20
4 Poliedros de recoloração convexa	25
4.1 Poliedro do problema RC	25
4.1.1 Desigualdades válidas	27
4.2 Poliedro do problema RC em árvores	31
4.2.1 Desigualdades válidas	32
5 Algoritmos de separação	39
5.1 Desigualdades de convexidade-estrela	39
5.2 Desigualdades de convexidade-árvore	43
6 Experimentos computacionais	49
6.1 Instâncias obtidas de árvores filogenéticas reais	49
6.2 Detalhes da implementação	51
6.3 Resultados computacionais	52

7 Conclusão	57
--------------------	-----------

Referências Bibliográficas	59
-----------------------------------	-----------

Lista de Figuras

1.1	Exemplos de colorações convexa e não-convexa.	2
1.2	Exemplos de colorações parciais.	2
3.1	Grafo G do Teorema 1. A cor T corresponde ao verde e a cor F ao vermelho.	15
3.2	Árvore parcialmente colorida com as cores a e b	23
4.1	Sinais dos coeficientes da desigualdade (4.4) quando G é uma árvore.	29
4.2	Sinais dos coeficientes da desigualdade (4.9) obtida a partir de P_{uw} , onde $u = u_1$ e $w = u_{t+1}$	33
4.3	Exemplo de desigualdade do Teorema 9.	36
5.1	Contração da sub-árvore T'_w no vértice \dot{w}	47

Lista de Tabelas

6.1 Resultados computacionais do *branch-and-cut* – Parte 1. 54
6.2 Resultados computacionais do *branch-and-cut* – Parte 2. 55

Capítulo 1

Introdução

Uma *coloração* de um grafo $G = (V, E)$ é uma função C que atribui a cada vértice uma cor, i.e. $C : V \rightarrow \mathcal{C}$, onde \mathcal{C} é o conjunto de cores. Observamos que não nos referimos aqui a *coloração própria*, no sentido clássico da teoria dos grafos, onde é exigido que $C(u) \neq C(v) \forall uv \in E$. Tratamos aqui de uma coloração arbitrária, sem nenhuma restrição. Para cada cor $c \in \mathcal{C}$, a *classe de cor c* é o conjunto dos vértices de G que possuem essa cor, i.e. $C^{-1}(c) := \{v \in V(G) \mid C(v) = c\}$.

Um *grafo colorido* é um par (G, C) que consiste de um grafo G e de uma coloração C desse (veja figura 1.1(a)). Uma coloração C de G é dita *convexa* se, para cada cor $c \in \mathcal{C}$, a classe de cor c induz um subgrafo conexo de G , i.e. $G[C^{-1}(c)]$ é conexo (veja figura 1.1(b)). Uma *coloração parcial* de um grafo G é uma função $C : V \rightarrow \mathcal{C} \cup \{\emptyset\}$, onde \emptyset denota a ausência de cor. Dessa maneira, se $C(v) = \emptyset$ então v é considerado *incolor*. Como anteriormente, um par (G, C) de um grafo G e uma coloração parcial C é chamado *grafo parcialmente colorido*. Neste trabalho, *estender* uma coloração parcial significa atribuir cor a cada vértice sem cor e manter as cores dos vértices já coloridos. Uma coloração parcial é dita *convexa* se ela pode ser estendida para uma coloração (total) convexa (veja a figura 1.2(b)). A figura 1.2(a) apresenta um exemplo de grafo parcialmente colorido tal que a coloração parcial não é convexa. Note que não é possível estender essa coloração de forma que cada classe de cor induza um subgrafo conexo. Uma *coloração boa* C de G é uma coloração parcial convexa tal que, para cada $c \in \mathcal{C}$, o conjunto de vértices que possuem a cor c induzem um subgrafo conexo de G . Note que toda coloração boa pode ser estendida para uma coloração convexa

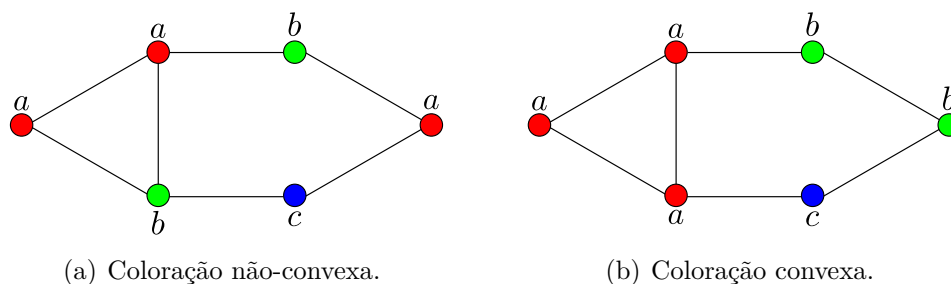
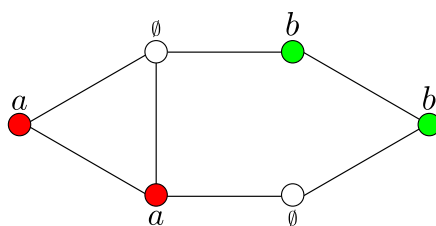
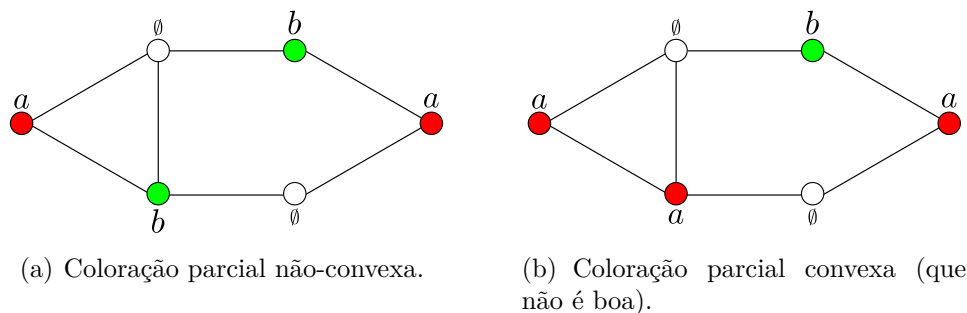


Figura 1.1: Exemplos de colorações convexas e não-convexas.



(c) Coloração (parcial convexa) boa.

Figura 1.2: Exemplos de colorações parciais.

em tempo $\mathcal{O}(|V| + |E|)$. É suficiente fazer uma busca em profundidade a partir de cada vértice incolor até encontrar o primeiro vértice colorido e usar a cor desse para colorir os vértices percorridos na busca.

Dado um grafo parcialmente colorido (G, C) , qualquer coloração C' de G é chamada de *recoloração*. O conjunto dos *vértices recoloridos* por uma recoloração C' de C é denotado por $R_C(C') := \{v \in V \mid C(v) \neq \emptyset \text{ e } C(v) \neq C'(v)\}$. Por simplicidade, denotamos tal conjunto por $R(C')$, quando a coloração C estiver clara no contexto.

1.1 O problema de recoloração convexa (RC)

Quando uma coloração de um dado grafo não é convexa, é desejável saber o quão distante de uma coloração convexa ela está. Essa noção de proximidade entre duas colorações é

capturada pelo conceito de *distância de recoloração*. Tal distância é definida como o menor número de mudanças de cor necessárias para transformar uma dada coloração em uma que seja convexa. Mais formalmente, dado um grafo colorido (G, C) , a versão mais simples do problema de *recoloração convexa* consiste em encontrar uma recoloração convexa C' de G que minimize $|R_C(C')|$. Quando pesos são atribuídos aos vértices, uma versão ponderada da distância de recoloração é definida de forma similar. Formalmente, definimos esse problema da seguinte forma:

Problema 1. RECOLORAÇÃO CONVEXA (RC)

Dados um grafo parcialmente colorido (G, C) e uma função de pesos $w : V \rightarrow \mathbb{Q}_{\geq}$, buscamos uma recoloração parcial convexa C' que minimize a distância de recoloração ponderada pela função w , i.e. $\sum_{v \in R_C(C')} w(v)$.

Algumas observações devem ser feitas sobre o problema RC. Uma vez que G é conexo, podemos assumir que a recoloração C' é definida sobre $\mathcal{C} \cup \{\emptyset\}$, isto é, o mesmo conjunto de cores inicial. Além disso, podemos (e vamos) assumir que o peso de todo vértice incolor é 0. Note que, para cada coloração parcial convexa que não é boa, existe uma coloração boa de mesmo valor. Logo, no restante do texto, consideraremos que no problema RC estamos interessados apenas em encontrar colorações boas. Se a função w atribui apenas pesos 1 ou 0 aos vértices de G , então temos um caso particular do problema RC denominado RC-BIN. Mais formalmente, o problema RC-BIN é o problema RC restrito às instâncias (G, C, w) tais que, para cada $v \in V$, $w(v) = 1$, se $C(v) \neq \emptyset$, e $w(v) = 0$, caso contrário. Por simplicidade, denotamos por (G, C) uma instância do RC-BIN.

1.2 Resultados conhecidos

O problema de recoloração convexa foi introduzido por Moran e Snir [MS05] em 2005 no contexto de árvores filogenéticas. Uma *árvore filogenética* é uma árvore que representa o curso de evolução de um certo conjunto de espécies. As folhas dessa árvore representam espécies existentes. Os vértices internos correspondem a espécies hipoteticamente extintas. Uma *característica* é um atributo compartilhado por todas as espécies do conjunto em questão. Cada espécie pode apresentar um *estado* diferente para uma determinada característica.

Mais formalmente, se V é o conjunto de espécies, então uma característica é uma função $C : V \rightarrow \mathcal{C}$, onde \mathcal{C} é o conjunto de todos os possíveis estados dessa característica.

Moran e Snir [MS08] provaram que o problema de recoloração convexa e algumas de suas variantes são \mathcal{NP} -difíceis mesmo para caminhos. Esses autores mostraram uma redução do problema de 3-satisfatibilidade para a versão de decisão do problema de recoloração convexa em caminhos. Em 2009, Kanj e Kratsch [KK09] mostraram que a versão de decisão do problema de recoloração convexa em caminhos é \mathcal{NP} -completa mesmo se cada cor aparece no máximo duas vezes. A redução exibida por esses autores, a partir do problema do conjunto estável máximo, é muito mais simples do que a prova apresentada por Moran e Snir [MS08]. Recentemente, Campêlo et al. [CHSW13] mostraram que o problema RC é \mathcal{NP} -difícil em grafos grade mesmo se a coloração inicial usa apenas duas cores. Eles também mostraram que o problema RC em grafos bipartidos com n vértices e duas cores não pode ser aproximado por um fator $o(\log n)$.

Uma 2-aproximação para o problema de recoloração convexa em caminhos foi mostrada em 2007 por Moran e Snir [MS07]. Em 2011, Lima e Wakabayashi [LW11] mostraram uma $\frac{3}{2}$ -aproximação para o problema de recoloração convexa restrito a caminhos tais que cada cor aparece no máximo duas vezes. Em 2008, Bar-Yeuda et al. [BYFR08] mostraram uma $(2 + \varepsilon)$ -aproximação para a recoloração convexa de árvores. Ainda em 2008, Kammer e Tholey apresentaram uma $(2 + \varepsilon)$ -aproximação para o problema em grafos com largura arbórea limitada.

Dado um grafo parcialmente colorido (G, C) , dizemos que uma cor $c \in \mathcal{C}$ é *ruim*, se a classe de cor c induz um subgrafo desconexo de G , mais formalmente, o subgrafo $G[C^{-1}(c)]$ possui mais de uma componente. Um algoritmo FPT para a recoloração convexa de árvores com complexidade $\mathcal{O}(nk^* \Delta^{k^*+2})$, onde k^* é a quantidade de cores ruins, foi proposto por Moran e Snir [MS08]. Por simplicidade, na notação de complexidade, escrevemos $\mathcal{O}^*(f(p))$ em vez de $\mathcal{O}(f(p)n^\ell)$, onde $f : \mathbb{N} \rightarrow \mathbb{N}$ é uma função que independe de n , p é o parâmetro do problema considerado e ℓ é uma constante que independe de n e p . Em 2007, Razgon [Raz07] exibiu um algoritmo FPT para o problema RC-BIN em árvores e parametrizado pelo número de trocas de cores cuja complexidade é $\mathcal{O}^*(256^p)$, onde p é o parâmetro do problema. Outros resultados sobre a complexidade parametrizada do problema de recoloração convexa podem

sem encontrados em Bodlaender et al. [BFL⁺11].

Chor et al. [CFR⁺07] generalizaram o conceito de coloração convexa visto anteriormente, permitindo que cada classe de cor induza no máximo r componentes conexos, onde r é um inteiro positivo. Chamamos de r -EXTCOL o problema de decidir se é possível estender a coloração inicial para uma coloração total tal que cada classe de cor induza um subgrafo que possui no máximo r componentes. Esses autores mostraram que 1-EXTCOL é \mathcal{NP} -completo mesmo quando a coloração inicial usa apenas duas cores. Para esse problema parametrizado em p (número de vértices sem cor), foi apresentado um algoritmo FPT com complexidade $\mathcal{O}^*(8^p)$. Ponta, Hüffner e Niedermeier [PHN08] reduziram o tempo de execução desse algoritmo para $\mathcal{O}^*(4^p)$. Para $r \geq 2$, o problema r -EXTCOL parametrizado pela largura arbórea do grafo é $\mathcal{W}[1]$ -difícil [CFR⁺07]. Algumas aplicações para esse tipo de coloração incluem problemas em redes de interação de proteínas, em redes filogenéticas e de roteamento [CFR⁺07, KT08].

1.3 Contribuições e organização do trabalho

No capítulo seguinte, apresentamos a notação e conceitos básicos sobre teoria poliédrica, teoria dos grafos e algoritmos de aproximação que foram usados neste trabalho. No Capítulo 3, mostramos uma redução polinomial do problema 3-SAT para uma versão de decisão do problema RC descrita em [CFR⁺07]. A partir de uma redução similar a essa, provamos que o problema RC-BIN parametrizado pelo número de trocas de cores é $\mathcal{W}[2]$ -difícil mesmo se a coloração inicial usa apenas duas cores. Além disso, apresentamos alguns resultados sobre a inaproximabilidade do problema RC. Ainda nesse capítulo, mostramos a 4-aproximação para esse problema em árvores proposta por Moran e Snir [MS07]. No Capítulo 4, apresentamos uma formulação com variáveis binárias para o problema RC e sua especialização para o caso onde o grafo é uma árvore. Definimos os poliedros associados a essas formulações e apresentamos várias classes de desigualdades não-triviais que definem facetas. No Capítulo 5, mostramos algoritmos de separação para cada uma das classes de desigualdades não-triviais que definem facetas do poliedro associado ao problema RC em árvores. As provas de corretude e uma discussão geral sobre a complexidade de cada um desses algoritmos também estão nesse capítulo. Um algoritmo do tipo *Branch-and-Cut* e os

resultados computacionais para o problema RC em árvores são mostrados no Capítulo 6. As considerações finais são apresentadas no Capítulo 7.

Capítulo 2

Preliminares

Neste texto, utilizamos notação e definições consistentes com as que são geralmente usadas em teoria dos grafos, teoria poliédrica e algoritmos de aproximação. Apesar disso, para que o texto fique auto-contido, apresentamos a seguir alguns conceitos e a notação que adotamos. O leitor familiarizado com a notação e conceitos dessas áreas pode ignorar este capítulo sem qualquer prejuízo.

A terminologia adotada para grafos, a menos de pequenas variações, é a mesma utilizada por Diestel [Die10] e Bondy e Murty [BM08]. Mais detalhes sobre teoria poliédrica podem ser encontrados em Ferreira e Wakabayashi [FW96], e em Nemhauser e Wolsey [NW99]. Em Carvalho et al. [dCCD⁺01] e Vazirani [Vaz01] podem ser encontradas mais informações sobre algoritmos de aproximação.

2.1 Teoria dos grafos

Dado um grafo G , denotamos por $V(G)$ e $E(G)$ seus conjuntos de vértices e arestas, respectivamente. A notação simplificada V e E é preferida quando o grafo G é claro no contexto. O complemento de G é denotado por $\overline{G} = (V, \overline{E})$.

Dizemos que dois vértices u e v em G são *adjacentes* ou *vizinhos*, se existe uma aresta em G incidente a ambos; denotamos tal aresta por uv . O conjunto dos vizinhos de um vértice $v \in V(G)$ é denotado por $N(v)$. De modo mais geral, para $W \subseteq V(G)$, denotamos por $N(W)$ o conjunto dos vizinhos em $V(G) \setminus W$ dos vértices em W , i.e. $N(W) := \bigcup_{v \in W} N(v) \setminus W$. O

complemento de $N(W) \cup W$ é chamado de *anti-vizinhança* de W e é denotado por $\overline{N}(W)$. Para $W \subseteq V(G)$, $\partial(W)$ denota o conjunto de arestas em G que têm exatamente uma das extremidades em W , i.e. $\partial(W) := \{uv \in E(G) \mid u \in W \text{ e } v \notin W\}$.

Se G e G' são dois grafos tais que $V(G') \subseteq V(G)$ e $E(G') \subseteq E(G)$, então G' é dito um *subgrafo* de G e escrevemos $G' \subseteq G$. Se $G' \subseteq G$ e G' contém todas as arestas $uv \in E(G)$ com $u, v \in V(G')$, então G' é um *subgrafo induzido* de G ; dizemos que $V(G')$ induz G' em G e escrevemos $G' := G[V(G')]$. Assim, se $U \subseteq V(G)$, então $G[U]$ denota o subgrafo de G cujo conjunto de vértices é U e cujas arestas são precisamente as arestas de G com ambas as extremidades em U . Se $W \subseteq V(G)$, então $G - W$ denota o subgrafo obtido a partir de G removendo-se W , i.e. $G - W = G[V(G) \setminus W]$. Por simplicidade, se v é um vértice de G , então escrevemos $G - v$ para denotar o subgrafo $G - \{v\}$.

Um *caminho* é um grafo não-vazio P tal que $V(P) = \{v_0, v_1, \dots, v_{n-1}\}$ e $E(P) = \{v_0v_1, v_1v_2, \dots, v_{n-2}v_{n-1}\}$, onde v_i são todos distintos. Dizemos que os vértices v_0 e v_{n-1} são *ligados* por P , e os chamamos de *extremidades* de P . Os vértices v_1, v_2, \dots, v_{n-2} são os vértices *internos* de P . O número de arestas de um caminho é o seu *tamanho* ou *comprimento*. Frequentemente, nos referimos a um caminho pela sequência natural de seus vértices, i.e. $P = v_0v_1 \dots v_{n-1}$, e dizemos que P é um caminho *de* v_0 *a* v_{n-1} . Para inteiros $0 \leq i \leq j \leq n-1$, denotamos por Pv_i o subcaminho de P que vai de v_0 a v_i , isto é, $Pv_i := v_0 \dots v_i$. Analogamente, escrevemos $v_iP := v_i \dots v_{n-1}$ e $v_iPv_j := v_i \dots v_j$. Usamos uma notação intuitiva similar para a concatenação de caminhos, por exemplo, se a união $Pu \cup uQv \cup vR$ de três caminhos é também um caminho, então denotamos este por $PuQvR$. Dado um grafo G e dois caminhos $P = v_0 \dots v_k \subseteq G$ e $P' = z_0 \dots z_l \subseteq G$, denotamos por PP' o caminho em G que é a concatenação de P e P' , se $V(P) \cap V(P') = \emptyset$ e $v_kz_0 \in E$. O grafo $(V(P), E(P) \cup \{v_0v_{n-1}\})$, onde $P = v_0 \dots v_{n-1}$ é um caminho, é chamado de *circuito*.

Um grafo G é dito *conexo* se existe um caminho entre quaisquer dois vértices em $V(G)$. Uma *árvore* é um grafo conexo e sem circuitos.

2.2 Teoria poliédrica

A menos de menção em contrário, os vetores aqui considerados são vetores-coluna. Assim, se x é um vetor, então x^T denota seu transposto. O *produto interno* de dois vetores $x, y \in \mathbb{R}^n$ é o escalar $x^T y := \sum_{i=1}^n x_i y_i$. Um vetor que possui todas as componentes iguais a 1 é denotado por $\mathbf{1}$. Um vetor de zeros é denotado por $\mathbf{0}$. Um vetor $x \in \mathbb{R}^n$ é referido como *unitário* se existe $1 \leq i \leq n$, tal que $x_i = 1$ e $x_j = 0 \forall j \neq i$. Utilizamos a notação e^i para designar o vetor unitário cuja componente 1 está na posição i .

Dado um subconjunto $S \subseteq \mathbb{R}^n$, um vetor $x \in \mathbb{R}^n$ é uma *combinação linear* dos vetores x^1, x^2, \dots, x^t em S se $x = \sum_{i=1}^t \lambda_i x^i$ para algum $(\lambda_1, \lambda_2, \dots, \lambda_t) \in \mathbb{R}^t$. Uma tal combinação linear é chamada *afim*, se $\lambda_1 + \dots + \lambda_t = 1$; *cônica*, se $\lambda_1, \dots, \lambda_t \geq 0$; *convexa*, se for afim e cônica. Para um conjunto não-vazio $S \subseteq \mathbb{R}^n$, denotamos por $\text{lin}(S)$ o *fecho linear* dos elementos de S , i.e. o conjunto de todos os vetores que são combinação linear de um número finito de vetores em S . Analogamente, definimos os fechos *afim*, *cônico* e *convexo*, denotados por $\text{afim}(S)$, $\text{cone}(S)$ e $\text{conv}(S)$, respectivamente. O $\text{conv}(S)$ é também conhecido como a *envoltória convexa* de S .

Um conjunto de vetores $x^1, \dots, x^k \in \mathbb{R}^n$ é *linearmente independente* se a única solução de $\sum_{i=1}^k \lambda_i x^i = 0$ é $\lambda = \mathbf{0}$. Um conjunto de vetores $x^1, \dots, x^k \in \mathbb{R}^n$ é *afim independente* se a única solução de $\sum_{i=1}^k \lambda_i x^i = 0$, $\sum_{i=1}^k \lambda_i = 0$ é $\lambda = \mathbf{0}$.

Dado $S \subseteq \mathbb{R}^n$, o *posto* de S , denotado por $\text{posto}(S)$, é a cardinalidade de um maior subconjunto de S que é linearmente independente. Analogamente, o *posto-afim* de S , denotado por $\text{posto-afim}(S)$, é a cardinalidade de um maior subconjunto de S afim independente. É conhecido que para todo $S \subseteq \mathbb{R}^n$, se $\mathbf{0} \in \text{afim}(S)$, então $\text{posto-afim}(S) = \text{posto}(S) + 1$; caso contrário $\text{posto-afim}(S) = \text{posto}(S)$. A *dimensão* $\text{dim}(S)$ de S é definida como $\text{dim}(S) := \text{posto-afim}(S) - 1$. Dizemos que S tem *dimensão plena*, se $\text{dim}(S) = n$.

O posto de uma matriz $A \in \mathbb{R}^{m \times n}$, denotado por $\text{posto}(A)$, é o posto do conjunto de vetores-coluna de A , que prova-se ser igual ao posto do conjunto de vetores-linha de A . Dizemos que A tem *posto-linha completo*, se $\text{posto}(A) = m$; e tem *posto-coluna completo* se $\text{posto}(A) = n$.

Um poliedro $\mathcal{P} \subseteq \mathbb{R}^n$ é um conjunto de vetores que satisfazem um número finito de

desigualdades lineares, i.e. $\mathcal{P} = \{x \in \mathbb{R}^n \mid Ax \leq b\}$, onde (A, b) é uma matriz $m \times (n+1)$. Ao longo deste texto, consideramos somente os poliedros tais que a matriz (A, b) que define cada um deles possui apenas valores racionais. Uma desigualdade $\lambda x \leq \lambda_0$ [ou (λ, λ_0)] é chamada *desigualdade válida* para \mathcal{P} se ela é satisfeita por todos os vetores em \mathcal{P} , i.e. $\lambda x \leq \lambda_0 \forall x \in \mathcal{P}$. Se (λ, λ_0) é uma desigualdade válida para \mathcal{P} e $F = \{x \in \mathcal{P} \mid \lambda x = \lambda_0\}$, então F é chamado de *face* de \mathcal{P} . Uma face F de \mathcal{P} é chamada de *faceta* de \mathcal{P} , se $\dim(F) = \dim(\mathcal{P}) - 1$.

2.3 Algoritmos de aproximação

Um problema de otimização tem três ingredientes principais: um conjunto de *instâncias*, um conjunto $\text{sol}(I)$ de *soluções viáveis* para cada instância I , e uma função que atribui um número $\text{val}(S)$ a cada solução viável S . O número $\text{val}(S)$ é o *valor* de S . Quando o conjunto $\text{sol}(I)$ das soluções viáveis associado a uma instância I é vazio, dizemos que a instância é *inválida*; caso contrário, a instância é *viável*.

Um problema *de minimização* está interessado nas soluções viáveis de valor mínimo, enquanto um problema *de maximização* está interessado nas soluções viáveis de valor máximo. Quando uma dessas alternativas — mínimo ou máximo — está subentendida, dizemos simplesmente *valor ótimo* e problema *de otimização*. Uma solução viável cujo valor é ótimo é chamada *solução ótima*. O valor de qualquer das soluções ótimas de uma instância I será denotado por $\text{opt}(I)$. Portanto,

$$\text{opt}(I) := \text{val}(S^*),$$

onde S^* é uma solução ótima de I . É claro que esse número só está definido se a instância I é viável.

Considere um problema de otimização em que $\text{val}(S) > 0$ para toda solução viável S de qualquer instância do problema. Seja \mathcal{A} um algoritmo que, para toda instância viável I do problema, devolve uma solução viável $\mathcal{A}(I)$ de I . Se o problema é de minimização e

$$\text{val}(\mathcal{A}(I)) \leq \alpha \text{opt}(I) \tag{2.1}$$

para toda instância I , dizemos que \mathcal{A} é uma α -*aproximação* para o problema. O fator α pode

ser uma função cujo valor depende de I . Dizemos que $\alpha = \alpha(I)$ é a *razão de aproximação* do algoritmo. É claro que $\alpha \geq 1$, uma vez que o problema é de minimização. No caso de problemas de maximização, basta refazer a definição com

$$\text{val}(\mathcal{A}(I)) \geq \alpha \text{opt}(I)$$

no lugar da desigualdade 2.1. É claro que nesse caso $0 < \alpha \leq 1$. Um *algoritmo de aproximação* é uma α -aproximação para algum α . Uma 1-aproximação para um problema de otimização é um *algoritmo exato* para o problema.

Observe que um algoritmo \mathcal{A} é uma α -aproximação para um problema de minimização (maximização) se α é uma delimitação superior (inferior) para a razão entre $\text{val}(\mathcal{A}(I))$ e $\text{opt}(I)$ para uma instância arbitrária I do problema. Como o valor de $\text{opt}(I)$ é em geral tão difícil de calcular quanto uma solução ótima do problema, para demonstrar que um algoritmo é uma α -aproximação é essencial ter boas delimitações para o valor de $\text{opt}(I)$. No contexto de algoritmos de aproximação, sempre supomos que estamos lidando com instâncias I tais que $\text{opt}(I) > 0$.

A cada instância I de um dado problema está associado um número natural $\langle I \rangle$ que chamamos *tamanho* da instância. Podemos imaginar que as instâncias, bem como as soluções viáveis, são cadeias de caracteres; nesse caso, $\langle I \rangle$ é o comprimento da cadeia de caracteres I . Um algoritmo \mathcal{A} para o problema é *polinomial* se existe um polinômio p tal que o consumo de tempo do algoritmo é limitado por $p(\langle I \rangle)$ para cada instância I . O conceito de algoritmo polinomial deve ser entendido como uma formalização da ideia de algoritmo eficiente. Se um problema é \mathcal{NP} -difícil, então é improvável que exista um algoritmo polinomial exato para o problema.

Capítulo 3

Complexidade e algoritmos de aproximação

Neste capítulo, demonstramos resultados sobre a complexidade computacional e a intratabilidade parametrizada do problema RC. Na Seção 3.1, apresentaremos o problema de extensão de coloração parcial e a prova de NP-completude obtida por Chor et al. [CFR⁺07]. A partir de uma redução FPT de um problema de satisfatibilidade de fórmulas FNC, provamos que o problema RC-BIN parametrizado pelo número de trocas de cores é $\mathcal{W}[2]$ -difícil usando uma redução similar à que será apresentada no Teorema 1. A Seção 3.2 contém dois resultados interessantes sobre a inaproximabilidade do problema RC. Por fim, na Seção 3.3, apresentaremos a 4-aproximação para árvores obtida por Moran & Snir [MS07], embora sejam conhecidos algoritmos de aproximação para o problema RC em árvores com razão $(2 + \varepsilon)$ [BYFR08] e 3 [MS07]. Optamos por apresentar a 4-aproximação devido a sua simplicidade e clareza na prova da sua razão de aproximação.

3.1 Complexidade parametrizada

Problema 2. r -EXTENSÃO DE COLORAÇÃO (r -EXTCOL).

A entrada consiste de um grafo $G = (V, E)$ e uma coloração parcial $C : V \rightarrow \mathcal{C} \cup \{\emptyset\}$ de G . É possível estender a coloração C para uma coloração total $C' : V \rightarrow \mathcal{C}$ tal que cada classe de cor induz no máximo r componentes? Mais formalmente, existe recoloração C' tal que

para cada $v \in V$, se $C(v) \neq \emptyset$, então $C'(v) = C(v)$; e para cada $d \in \mathcal{C}$, o grafo $G[C^{-1}(d)]$ tem no máximo r componentes?

Teorema 1 ([CFR⁺07]). *O problema 1-EXTCOL é \mathcal{NP} -completo mesmo quando a coloração inicial usa apenas duas cores.*

Prova. Para provar esse resultado, mostraremos uma redução do problema de 3-SAT para o problema 1-EXTCOL. Sejam $Q = \{Q_1, Q_2, \dots, Q_\ell\}$ o conjunto de cláusulas e $X = \{x_1, x_2, \dots, x_t\}$ o conjunto de variáveis de uma instância qualquer do 3-SAT. A partir dessa instância, construímos um grafo $G = (V, E)$ parcialmente colorido pela função $C : V \rightarrow \{T, F, \emptyset\}$ da seguinte forma. Para cada variável x_i , associamos um *gadget* como o da Figura 3.1(a). Para cada cláusula Q_j , associamos um vértice v_j em G tal que $C(v_j) = T$. Fazemos v_j adjacente a α_i^+ se, e somente se, o literal x_i pertence a Q_j ; e v_j adjacente a α_i^- se, e somente se, o literal $\neg x_i$ pertence a Q_j , onde $i \in \{1, 2, \dots, t\}$. O grafo G possui ainda dois vértices γ^T e γ^F tais que $C(\gamma^T) = T$, $C(\gamma^F) = F$, e γ^T, γ^F são adjacentes aos vértices α_i^+ , α_i^- para todo $i \in \{1, 2, \dots, t\}$. Na Figura 3.1(b) exibimos o grafo obtido a partir da cláusula $(x_1 \vee \neg x_2 \vee x_3)$.

Seja (Q, X) uma instância “sim” do problema 3-SAT e considere uma atribuição de valores *verdadeiro* e *falso* às variáveis em X tal que cada cláusula em Q seja verdadeira. Definimos uma coloração $C' : V \rightarrow \{T, F\}$ que estende C da seguinte maneira. Para cada variável $x_i \in X$, fazemos $C'(\alpha_i^+) := T$, $C'(\alpha_i^-) := F$, se x_i tem valor verdadeiro, e $C'(\alpha_i^+) := F$, $C'(\alpha_i^-) := T$, se x_i é falsa. Note que cada cláusula Q_j contém um literal (positivo ou negativo) de alguma variável x_i cujo valor faz com que essa cláusula seja verdadeira. Se o literal $x_i \in Q_j$, então o vértice v_j de G é adjacente a α_i^+ e não-adjacente a α_i^- . Se o literal $\neg x_i \in Q_j$, então v_j é adjacente a α_i^- e não-adjacente a α_i^+ . Uma vez que a atribuição de valores satisfaz cada cláusula, temos que na coloração C' todos os vértices em $\{v_1, v_2, \dots, v_\ell\}$ estão na componente conexa que contém o vértice γ^T no subgrafo induzido pela classe de cor T . Além disso, é fácil ver que para cada $i \in \{1, 2, \dots, t\}$, o vértice β_i^T está na mesma componente conexa que contém o vértice γ^T no subgrafo induzido pela classe de cor T , e que tal subgrafo tem uma única componente conexa. Por fim, note que, para cada $i \in \{1, 2, \dots, t\}$, temos que $\beta_i^F, \alpha_i^*, \gamma^F$ formam um caminho monocromático de cor F , onde

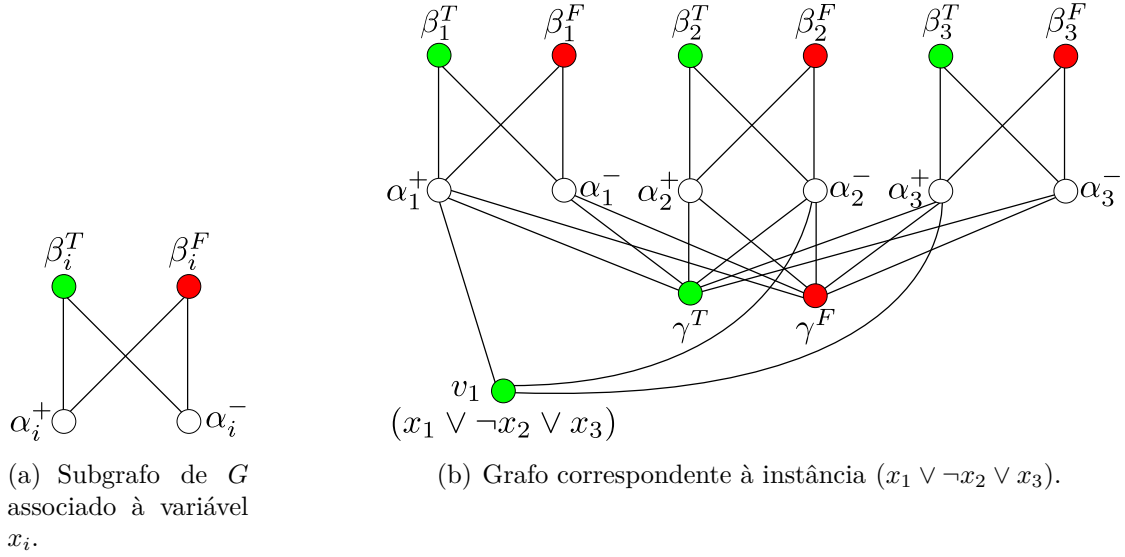


Figura 3.1: Grafo G do Teorema 1. A cor T corresponde ao verde e a cor F ao vermelho.

$\alpha_i^* = \alpha_i^+$, se x_i tem valor falso, e $\alpha_i^* = \alpha_i^-$, se x_i tem valor verdadeiro. Portanto, a coloração total C' é convexa e (G, C') é uma instância “sim” de 1-EXTCOL.

Suponha que C é uma coloração parcial convexa e considere uma coloração total convexa C' que estende C . Note que C' atribui cores diferentes aos vértices α_i^+ e α_i^- para cada $i \in \{1, 2, \dots, t\}$, caso contrário ou β_i^T ou β_i^F definiria uma componente trivial do grafo induzido pela sua classe de cor. Para cada $i \in \{1, 2, \dots, t\}$, atribuímos a x_i o valor verdadeiro, se $C'(\alpha_i^+) = T$, e o valor falso, caso contrário. Uma vez que C' é convexa, temos que, para cada $j \in \{1, 2, \dots, \ell\}$, o vértice v_j é adjacente a um vértice α_i^* colorido com a cor T , onde ou $\alpha_i^* = \alpha_i^+$ ou $\alpha_i^* = \alpha_i^-$. Logo, a atribuição de valor às variáveis em X acima definida satisfaz toda cláusula em Q . Portanto, (Q, X) é uma instância “sim” do problema 3-SAT. \square

Observamos que, na prova acima, o grafo G construído é bipartido. Assim, podemos afirmar que o problema 1-EXTCOL é \mathcal{NP} -completo mesmo para grafos bipartidos cuja coloração inicial usa apenas duas cores.

O problema RC-BIN é o problema RC restrito as instâncias (G, C, w) tais que, para cada $v \in V$, $w(v) = 1$, se $C(v) \neq \emptyset$, e $w(v) = 0$, caso contrário. A seguir, definimos uma versão do RC-BIN parametrizada pelo número de vértices recoloridos. No Teorema 2, apresentamos uma redução FPT de uma versão parametrizada do problema de satisfatibilidade de fórmulas na forma normal conjuntiva (FNC).

Problema 3. RECOLORAÇÃO CONVEXA COM PARÂMETRO p (RC-PARAM).

Dados uma instância (G, C) do problema RC-BIN e um parâmetro $p \in \mathbb{Z}_{\geq}$, existe uma recoloração convexa de (G, C) que troca a cor de no máximo p vértices?

Problema 4. SATISFATIBILIDADE FNC LIMITADO COM PARÂMETRO p (BOUNDED-CNF-SAT).

Dados uma fórmula \mathcal{F} em FNC e um parâmetro $p \in \mathbb{Z}_{>}$, é possível satisfazer a fórmula \mathcal{F} atribuindo-se a no máximo p variáveis o valor verdadeiro?

Teorema 2. *O problema RC-PARAM é $\mathcal{W}[2]$ -difícil mesmo se a coloração inicial usa apenas duas cores.*

Prova. A seguir, mostraremos uma redução FPT do problema BOUNDED-CNF-SAT, que é $\mathcal{W}[2]$ -completo [DMS11], para o problema RC-PARAM. Essa redução é uma simples adaptação da redução mostrada na prova do Teorema 1.

Seja (\mathcal{F}, p) uma instância do problema BOUNDED-CNF-SAT, onde $p \in \mathbb{Z}_{>}$ é um parâmetro. Embora as cláusulas em \mathcal{F} não tenham um tamanho fixo, podemos gerar um grafo parcialmente colorido $(G = (V, E), C)$ da mesma maneira que na prova do Teorema 1 apenas acrescentando arestas extras. Para cada $v \in V$ tal que $C(v) \neq \emptyset$, adicionamos um conjunto $S(v)$ de p novos vértices ao grafo G e fazemos cada vértice em $S(v)$ adjacente a v . Além disso, para cada vértice α_i^+ , onde $i \in \{1, 2, \dots, t\}$ e t é o número de variáveis da fórmula \mathcal{F} , adicionamos um novo vértice u_i e uma aresta $u_i\alpha_i^+$ ao grafo G . Seja G' o grafo obtido após esse procedimento e N o seu número de vértices. Note que $N \leq n + np$, onde n é o número de vértices de G . Definimos a coloração $C' : V(G') \rightarrow \{T, F, \emptyset\}$ como uma extensão da coloração C tal que $C'(v) := C(z)$ para todo $z \in \hat{V}$ e $v \in S(z)$, onde $\hat{V} := \{z \in V \mid C(z) \neq \emptyset\}$, e $C'(u_i) := F$ para todo $i \in \{1, 2, \dots, t\}$. Claramente, a instância $((G', C'), p)$ de RC-PARAM pode ser computada em tempo FPT no tamanho de \mathcal{F} .

Suponha que a fórmula \mathcal{F} pode ser satisfeita atribuindo-se o valor verdadeiro a no máximo p variáveis e considere uma tal atribuição de valores às variáveis de \mathcal{F} . Considere o mapeamento dos valores verdadeiro e falso às variáveis de \mathcal{F} para atribuir as cores T e F aos vértices $\{\alpha_1^+, \alpha_1^-, \dots, \alpha_t^+, \alpha_t^-\}$ de G' como descrito na prova do Teorema 1. Dessa forma, podemos obter uma recoloração convexa C'' de (G', C') que troca exatamente as cores dos

vértices u_i tais que x_i tem valor verdadeiro, onde $i \in \{1, 2, \dots, t\}$. Claramente, C'' é uma recoloração convexa de (G', C') que muda a cor de no máximo p vértices.

Seja C''' uma recoloração convexa de (G', C') . Se C''' troca a cor de um vértice $v \in \hat{V}$, então a cor de todo vértice em $S(v)$ também é trocada. Além disso, observe que se a cor de algum vértice $z \in S(v)$, onde $v \in \hat{V}$, é modificada, então C''' também troca a cor de algum vértice em \hat{V} , pois C' utiliza apenas duas cores e cada uma delas aparece em pelo menos um vértice do conjunto V . Em ambos os casos, a recoloração C''' muda a cor de mais de p vértices. Portanto, se C''' troca a cor de no máximo p vértices em G' , então ela recolora no máximo p vértices do conjunto $\{u_1, u_2, \dots, u_t\}$. Note que $C'''(\alpha_i^+) = T$ se, e somente se, $C'''(u_i) = T$. Dessa forma, considere o mapeamento da coloração C''' de G' para uma atribuição de valores verdadeiro e falso às variáveis de \mathcal{F} como descrito na prova do Teorema 1. Uma vez que no máximo p vértices no conjunto $\{\alpha_1^+, \alpha_2^+, \dots, \alpha_t^+\}$ possuem a cor V na coloração C''' , temos que a fórmula \mathcal{F} é satisfeita atribuindo-se o valor verdadeiro a no máximo p variáveis. \square

3.2 Inaproximabilidade do RC

Teorema 3. *Sejam $I' = (G', C', w)$ uma instância do problema RC com duas cores, onde G' é um grafo com N vértices, e $f : \mathbb{N} \rightarrow \mathbb{R}$ uma função da forma $f(x) = 2^{\text{poly}(x)}$, onde $\text{poly}(x)$ é um polinômio em x . Não existe $f(N)$ -aproximação para o problema RC, a menos que $\mathcal{P} = \mathcal{NP}$.*

Prova. Seja $I = (G, C)$ uma instância do problema 1-EXTCOL formada por um grafo com n vértices $G = (V, E)$ e uma coloração parcial $C : V \rightarrow \{T, F, \emptyset\}$ de G . Suponha que o número de vértices sem cor seja estritamente menor do que n e que as cores T e F apareçam em pelo menos um vértice do grafo. Seja G' um grafo com N vértices obtido a partir de G da forma descrita a seguir. Para cada $v \in V$ tal que $C(v) = \emptyset$, adicionamos dois novos vértices v_T, v_F , e duas arestas vv_T, vv_F ao grafo G . Seja ainda C' uma coloração de G' tal que, para cada $v \in V$, $C'(v) = \emptyset$, $C'(v_T) = T$ e $C'(v_F) = F$, se $C(v) = \emptyset$, e $C'(v) = C(v)$, caso contrário. A partir de I , construímos uma instância $I' = (G', C', w)$ do problema RC tal que, para cada $v \in V$, $w(v) = 0$ e $w(v_T) = w(v_F) = 1$, se $C(v) = \emptyset$, e $w(v) = Nf(N)$,

caso contrário. Claramente, temos que $\text{opt}(I') > 0$.

Suponha que exista uma $f(N)$ -aproximação polinomial \mathcal{A} para o problema RC, onde I é uma instância com duas cores desse e N o número de vértices no grafo de I . Considere a instância I' de RC acima construída. Seja $\mathcal{A}(I')$ a solução obtida pelo algoritmo \mathcal{A} para essa instância I' e seja $\text{val}(\mathcal{A}(I'))$ o seu valor. Note que a representação binária dos pesos dados pela função w tem tamanho $\mathcal{O}(N \log(Nf(N)))$, donde segue que a instância I' tem tamanho polinomial em n . Logo, a solução $\mathcal{A}(I')$ pode ser calculada em tempo polinomial em n . Se $\text{val}(\mathcal{A}(I')) < Nf(N)$, então a solução devolvida por \mathcal{A} não muda a cor de nenhum vértice colorido no grafo G (i.e. $v \in V$ tal que $C(v) \neq \emptyset$). Portanto, I é uma instância “sim” do problema 1-EXTCOL. Se $\text{val}(\mathcal{A}(I')) \geq Nf(N)$, então $\text{opt}(I') \geq N$, pois $Nf(N) \leq \text{val}(\mathcal{A}(I')) \leq f(N) \text{opt}(I')$. Logo, qualquer solução ótima da instância I' tem que recolorir algum vértice de G cuja cor inicial é T ou F . Portanto, I é uma instância “não” do problema 1-EXTCOL. Assim, temos que a resposta para a instância I é “sim” se, e somente se, $\text{val}(\mathcal{A}(I')) < Nf(N)$. Logo, o algoritmo \mathcal{A} resolve o problema 1-EXTCOL (que é \mathcal{NP} -completo) em tempo polinomial, o que é um absurdo, a menos que $\mathcal{P} = \mathcal{NP}$. \square

Seja $I = (G, C)$ uma instância do problema RC-BIN, onde G é um grafo com n vértices. Kammer e Tholey [KT12] provaram que se a coloração C atribui uma mesma cor a no máximo dois vértices, então não existe $(1 - o(1)) \ln \ln n$ -aproximação para I , a menos que $\mathcal{NP} \subseteq \text{DTIME}(n^{\mathcal{O}(\log \log n)})$. No teorema a seguir, mostramos um resultado similar a esse para o caso em que C é uma coloração que utiliza apenas duas cores.

Teorema 4. *Seja $I' = (G', C')$ uma instância do problema RC-BIN com duas cores, onde G' é um grafo com N vértices. Para qualquer número real ϵ , $0 < \epsilon < 1$, não existe uma $N^{1-\epsilon}$ -aproximação para RC-BIN, a menos que $\mathcal{P} = \mathcal{NP}$.*

Prova. Dado $\epsilon \in \mathbb{R}$, $0 < \epsilon < 1$, tome $\ell \in \mathbb{Z}$ tal que $\frac{\ell}{\ell+2} \geq 1 - \epsilon$, e considere a função $f(x) = x^\ell$. Tome $I = (G, C)$ uma instância do problema 1-EXTCOL, onde $G = (V, E)$ é um grafo com n vértices e $C : V \rightarrow \{T, F, \emptyset\}$ é uma coloração com duas cores. Suponha que G tenha algum vértice colorido, ou seja, o número de vértices sem cor é menor do que n . A partir da instância I definimos uma instância $I' = (G', C')$ do problema RC-BIN da

maneira descrita a seguir. Seja $\hat{V} := \{z \in V \mid C(z) \neq \emptyset\}$. Para cada $v \in \hat{V}$, adicionamos um conjunto $S(v)$ de $nf(n) - 1$ novos vértices ao grafo G e fazemos cada vértice em $S(v)$ adjacente a v . Além disso, para cada $v \in V$ tal que $C(v) = \emptyset$, adicionamos dois novos vértices v_T, v_F , e duas arestas vv_T, vv_F ao grafo G . Seja G' o grafo obtido após esse procedimento e N o seu número de vértices. Claramente, temos que $\text{opt}(I') > 0$. Note que $N \leq n + n(nf(n) - 1) = n^2f(n) = n^{\ell+2}$. Portanto, o grafo G' tem tamanho polinomial em n . Além disso, observe que $N^{\frac{\ell}{\ell+2}} \leq n^\ell = f(n)$. Logo, $N^{1-\epsilon} \leq N^{\frac{\ell}{\ell+2}} \leq f(n)$. Definimos a coloração $C' : V(G') \rightarrow \{T, F, \emptyset\}$ como uma extensão da coloração C tal que $C'(v) := C(u) \forall u \in \hat{V}$, $\forall v \in S(u)$, e $C'(v_T) := T, C'(v_F) := F \forall v \in V \setminus \hat{V}$.

Suponha que exista uma $N^{1-\epsilon}$ -aproximação polinomial \mathcal{A} para o problema RC-BIN, onde N é o número de vértices do grafo de uma instância com duas cores. Uma vez que G' tem tamanho polinomial em n , a recoloração convexa $\mathcal{A}(I')$ é calculada em tempo também polinomial em n . Note que se a cor de um vértice $v \in \hat{V}$ é trocada na recoloração convexa $\mathcal{A}(I')$, então a cor de todo vértice em $S(v)$ também é trocada. Além disso, observe que se a cor de algum vértice $z \in S(v)$, onde $v \in \hat{V}$, é modificada, então a recoloração convexa $\mathcal{A}(I')$ também troca a cor de algum vértice em \hat{V} , pois C' utiliza apenas duas cores e cada uma delas aparece em pelo menos um vértice do conjunto V . Em qualquer desses casos, temos que a solução $\mathcal{A}(I')$ é tal que $\text{val}(\mathcal{A}(I')) \geq nf(n)$. Vamos analisar dois casos. Se $\text{val}(\mathcal{A}(I')) < nf(n)$, então a solução $\mathcal{A}(I')$ não troca a cor de qualquer vértice em $\hat{V} \cup \bigcup_{v \in \hat{V}} S(v)$. Portanto, a coloração C de G pode ser estendida para uma coloração convexa total, ou seja, I é uma instância “sim” do problema 1-EXTCOL. Se $\text{val}(\mathcal{A}(I')) \geq nf(n)$, então temos que $nf(n) \leq \text{val}(\mathcal{A}(I')) \leq N^{1-\epsilon} \text{opt}(I')$. Uma vez que $N^{1-\epsilon} \leq f(n)$, temos que $\text{opt}(I') \geq n$. Portanto, em toda solução ótima de I' , algum vértice $v \in \hat{V}$ tem a sua cor trocada. Dessa forma, I é uma instância “não” do problema 1-EXTCOL. Assim, temos que a resposta para a instância I é “sim” se, e somente se, $\text{val}(\mathcal{A}(I')) < nf(n)$. Logo, o algoritmo \mathcal{A} resolve o problema 1-EXTCOL em tempo polinomial, o que é um absurdo, a menos que $\mathcal{P} = \mathcal{NP}$. \square

3.3 Uma 4-aproximação para o problema RC em árvores

Primeiramente, apresentaremos uma 3-aproximação polinomial para o problema RC restrito a caminhos. A 4-aproximação para o problema RC em árvores proposta por Moran & Snir [MS07] é uma simples generalização do algoritmo para caminhos.

Seja (T, C, w) uma instância do problema RC, onde $T = (V, E)$ é uma árvore, $C : V \rightarrow \mathcal{C}$ uma coloração de T e $w : V \rightarrow \mathbb{Q}_{\geq}$ uma função de pesos. Definimos o conjunto suporte(w) := $\{v \in V \mid w(v) > 0\}$. Um conjunto X de vértices de T é uma *cobertura convexa* (ou simplesmente *cobertura*) para a árvore colorida (T, C) , se a coloração parcial C_X é convexa, onde $C_X(v) := \emptyset$, se $v \in X$, e $C_X(v) := C(v)$, caso contrário. Ou seja, um subconjunto $X \subseteq V$ é dito uma cobertura para (T, C) , se C pode ser transformada em uma coloração convexa recolorindo apenas vértices em X . Portanto, se C' é uma recoloração convexa de (T, C) , então $R_C(C')$, o conjunto de vértices recoloridos por C' , é uma cobertura para (T, C) com custo $w(R_C(C')) := \sum_{v \in R_C(C')} w(v)$. Além disso, decidir se um subconjunto $X \subseteq V(T)$ é uma cobertura para (T, C) e, caso seja uma cobertura, construir uma recoloração total convexa C' de C tal que $R_C(C') \subseteq X$, pode ser feito em $\mathcal{O}(nk)$, onde $n = |V(T)|$ e $k = |\mathcal{C}|$. Para decidirmos se C_X é coloração convexa de T , fixamos uma cor $c \in \mathcal{C}$ e um vértice v que possua essa cor em C_X , e fazemos uma busca a partir de v a fim de obter uma subárvore maximal T' tal que cada um de seus vértices ou possui a cor c ou é incolor. Se T' contém todos os vértices de T que possuem a cor c , então colorimos com c os vértices incolores de T' e repetimos o processo para a próxima cor em \mathcal{C} . Caso contrário, a coloração C_X não é convexa e, portanto, X não é uma cobertura para (T, C) . Logo, o problema RC em árvores é polinomialmente equivalente a encontrar uma cobertura X tal que $w(X)$ é minimizado.

Os algoritmos apresentados nesta seção utilizam a técnica chamada *proporção local*, que é também utilizada em algoritmos de aproximação para problemas de cobertura de vértices, e.g. [BBF99]. A seguir, descreveremos brevemente essa técnica.

A entrada do problema é uma tripla $(V, \Sigma \subseteq 2^V, w : V \rightarrow \mathbb{Q}_{\geq})$ e o objetivo é encontrar um subconjunto $X \in \Sigma$ tal que $w(X)$ é minimizado, i.e. $w(X) = \text{opt}(V, \Sigma, w) = \min_{Y \in \Sigma} w(Y)$. No nosso contexto, V é o conjunto de vértices da árvore T e Σ é o conjunto de cobertu-

ras de (T, C) . O princípio da técnica de proporção local é baseado na seguinte observação apresentada por Bar-Yehuda [BY00]:

Observação 1. *Para quaisquer funções $w_1 : V \rightarrow \mathbb{Q}_{\geq}$ e $w_2 : V \rightarrow \mathbb{Q}_{\geq}$, temos que*

$$\text{opt}(V, \Sigma, w_1) + \text{opt}(V, \Sigma, w_2) \leq \text{opt}(V, \Sigma, w_1 + w_2).$$

Dada a função de pesos inicial w , definimos duas outras funções w_1, w_2 tais que $w_1 + w_2 = w$ e $|\text{suporte}(w_1)| < |\text{suporte}(w)|$. De maneira geral, uma r -aproximação que faz uso da técnica de proporção local consiste em executar o algoritmo recursivamente para (V, Σ, w_1) afim de encontrar uma solução X tal que possamos garantir que $w_2(X) \leq r \text{opt}(V, \Sigma, w_2)$. Dessa forma, a Observação 1 permite concluir que X é uma r -aproximação para $(V, \Sigma, w_1 + w_2 = w)$. A propriedade original, apresentada pela primeira vez em [BYE85], que foi usada para garantir que $w_2(X) \leq r \text{opt}(V, \Sigma, w_2)$ é a r -efetividade. Uma função w_2 é dita r -efetiva se $w_2(Y) \leq r \text{opt}(V, \Sigma, w_2)$ para todo $Y \in \Sigma$. Note que se $V \in \Sigma$, então para mostrar que a propriedade anterior é válida basta provar que $w_2(V) \leq r \text{opt}(V, \Sigma, w_2)$.

Teorema 5 ([BYE85]). *Seja $X \in \Sigma$ tal que $w_1(X) \leq r \text{opt}(V, \Sigma, w_1)$. Se w_2 é r -efetiva, então $w(X) = w_1(X) + w_2(X) \leq r \text{opt}(V, \Sigma, w)$.*

Os algoritmos apresentados a seguir são aplicações do Teorema 5. Mostraremos que o Algoritmo 1 é uma 3-aproximação para recoloração convexa de caminhos e o Algoritmo 2 é uma 4-aproximação para recoloração convexa de árvores.

Note que uma coloração parcial de um caminho não é convexa se, e somente se, existem os vértices como descritos na Linha 5 do Algoritmo 1. É fácil ver que qualquer cobertura para $(P, C_{V \setminus \text{suporte}(w_2)})$ tem que conter pelo menos um dos vértices de qualquer tripla descrita na Linha 5. Logo, temos que $\text{opt}(P, C_{V \setminus \text{suporte}(w_2)}, w_2) \geq \varepsilon$. Portanto, podemos concluir que w_2 é 3-efetiva, pois $w_2(V) = 3\varepsilon \leq 3 \text{opt}(P, C_{V \setminus \text{suporte}(w_2)}, w_2)$.

O Algoritmo 1 não serve para gerar uma solução para o problema RC em árvores, pois em uma árvore podem não existir os vértices como descritos na Linha 5 mesmo se $V \setminus \text{suporte}(w)$ não é uma cobertura. A Figura 3.2 apresenta uma árvore na qual o Algoritmo 1 falha. No Algoritmo 2, a Linha 5 generaliza a ideia da Linha 5 do Algoritmo 1 de forma que os pares de vértices descritos existem em toda coloração parcial não-convexa de uma árvore.

Algoritmo 1 3-APROXCAMINHOS

Entrada: Um caminho $P = (V, E)$, uma coloração $C : V \rightarrow \mathcal{C}$ e pesos $w : V \rightarrow \mathbb{Q}_{\geq}$.**Saída:** Cobertura X .

```

1: se  $V \setminus \text{suporte}(w)$  é uma cobertura então
2:    $X \leftarrow V \setminus \text{suporte}(w)$ 
3:   devolva  $X$ 
4: senão
5:   Encontre 3 vértices distintos  $x, y, z \in \text{suporte}(w)$  tais que  $C(x) = C(z) \neq C(y)$  e
      $y \in V(P_{xz})$ 
6:    $\varepsilon \leftarrow \min\{w(x), w(y), w(z)\}$ 
7:   para todo  $v \in V$  faça
8:     se  $v \in \{x, y, z\}$  então
9:        $w_2(v) \leftarrow \varepsilon$ 
10:    senão
11:       $w_2(v) \leftarrow 0$ 
12:    fim se
13:  fim para
14:   $w_1 \leftarrow w - w_2$ 
15:  devolva 3-APROXCAMINHOS( $P, C_{V \setminus \text{suporte}(w_1)}, w_1$ )
16: fim se

```

Algoritmo 2 4-APROXÁRVORES

Entrada: Uma árvore $T = (V, E)$, uma coloração $C : V \rightarrow \mathcal{C}$ e pesos $w : V \rightarrow \mathbb{Q}_{\geq}$.**Saída:** Cobertura X .

```

1: se  $V \setminus \text{suporte}(w)$  é uma cobertura então
2:    $X \leftarrow V \setminus \text{suporte}(w)$ 
3:   devolva  $X$ 
4: senão
5:   Encontre 2 pares de vértices (não necessariamente distintos)  $x_1, x_2, y_1, y_2 \in$ 
      $\text{suporte}(w)$  tais que  $C(x_1) = C(x_2) \neq C(y_1) = C(y_2)$  e  $V(P_{x_1x_2}) \cap V(P_{y_1y_2}) \neq \emptyset$ 
6:    $\varepsilon \leftarrow \min\{w(x_1), w(x_2), w(y_1), w(y_2)\}$ 
7:   para todo  $v \in V$  faça
8:     se  $v \in \{x_1, x_2, y_1, y_2\}$  então
9:        $w_2(v) \leftarrow \varepsilon$ 
10:    senão
11:       $w_2(v) \leftarrow 0$ 
12:    fim se
13:  fim para
14:   $w_1 \leftarrow w - w_2$ 
15:  devolva 4-APROXÁRVORES( $T, C_{V \setminus \text{suporte}(w_1)}, w_1$ )
16: fim se

```

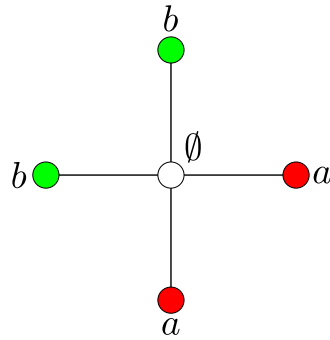


Figura 3.2: *Árvore parcialmente colorida com as cores a e b .*

O Algoritmo 2 é correto, pois se não existem dois pares como descritos na Linha 5, então o conjunto $V \setminus \text{suporte}(w)$ é uma cobertura de (T, C) . A partir de uma análise análoga à do Algoritmo 1, podemos concluir que a função w_2 é 4-efetiva. Logo, esse algoritmo devolve uma cobertura com peso no máximo $4 \text{opt}(T, C, w)$.

Capítulo 4

Poliedros de recoloração convexa

Neste capítulo, apresentamos uma formulação linear inteira para o problema RC em grafos arbitrários e sua especialização para árvores. Além disso, mostramos várias classes de desigualdades que induzem facetos dos poliedros associados. Os resultados relativos ao poliedro geral são apresentados na Seção 4.1 e os resultados para árvores na Seção 4.2.

4.1 Poliedro do problema RC

Seja (G, C, w) uma tripla, onde $G = (V, E)$ é um grafo conexo, $C : V \rightarrow \mathcal{C} \cup \{\emptyset\}$ uma coloração parcial de G e $w : V \rightarrow \mathbb{Q}_{\geq}$ uma função que associa pesos aos vértices. Suponha $n := |V| \geq 3$ e $k := |\mathcal{C}| \geq 2$. Para vértices distintos, não-adjacentes u e v , denote por $\Gamma(u, v)$ a coleção de conjuntos minimais de vértices que separam u e v , i.e. os conjuntos minimais de vértices S tais que u e v pertencem a componentes diferentes de $G - S$. Um tal conjunto S é chamado de corte que separa u e v .

A seguir, descrevemos uma formulação linear para o problema RC com a instância (G, C, w) como entrada. Nessa formulação, que denominamos \mathcal{R} , para cada $v \in V$ e $c \in \mathcal{C}$, temos uma variável binária $x_{v,c}$ com o seguinte significado:

$$x_{v,c} = \begin{cases} 1, & \text{se o vértice } v \text{ possui a cor } c, \\ 0, & \text{caso contrário.} \end{cases}$$

Para expressar a função objetivo, considere constantes $w_{v,c}$ para todo $v \in V$ e todo $c \in \mathcal{C}$

definidas a seguir:

$$w_{v,c} = \begin{cases} w(v), & \text{se } C(v) = c, \\ 0, & \text{caso contrário.} \end{cases}$$

$$\begin{aligned} (\mathcal{R}) \quad & \max \sum_{v \in V} \sum_{c \in \mathcal{C}} w_{v,c} x_{v,c} \\ & \text{s.a. } \sum_{c \in \mathcal{C}} x_{v,c} \leq 1 \quad \forall v \in V, \end{aligned} \quad (4.1)$$

$$x_{u,c} + x_{v,c} - \sum_{z \in Z} x_{z,c} \leq 1 \quad \forall uv \notin E, Z \in \Gamma(u, v), c \in \mathcal{C}, \quad (4.2)$$

$$x_{v,c} \in \{0, 1\} \quad \forall v \in V, \forall c \in \mathcal{C}. \quad (4.3)$$

Dada $C' : V \rightarrow \mathcal{C} \cup \{\emptyset\}$ uma recoloração boa de (G, C) , definimos o vetor $\chi(C') \in \{0, 1\}^{nk}$ da seguinte forma:

$$\chi(C')_{v,c} = \begin{cases} 1, & \text{se } C'(v) = c, \\ 0, & \text{caso contrário.} \end{cases}$$

Seja $\chi[A] := \{\chi(C') \in \{0, 1\}^{nk} \mid C' \in A\}$, onde A é um conjunto de colorações parciais de G . Definimos $\mathcal{Q}(G)$ o poliedro associado ao problema de recoloração convexa como:

$$\mathcal{Q}(G) := \text{conv}\{\chi(C') \in \{0, 1\}^{nk} \mid C' \text{ é uma coloração boa de } G\}.$$

Proposição 1. $\mathcal{Q}(G) = \text{conv}\{x \in \mathbb{R}^{nk} \mid x \text{ satisfaz as restrições 4.1, 4.2 e 4.3 de } \mathcal{R}\}$.

Prova. Considere uma coloração boa C' de G e seja $x' := \chi(C')$. As restrições 4.1 e 4.3 são claramente satisfeitas por x' . Tome $u, v \in V$ tal que $uv \notin E$, $Z \in \Gamma(u, v)$ e $c \in \mathcal{C}$. Podemos assumir $x'_{u,c} = x'_{v,c} = 1$, caso contrário a desigualdade 4.2 é trivialmente satisfeita por x' . Uma vez que C' é uma coloração boa, existe um caminho monocromático (de cor c) entre u e v com ao menos 3 vértices (pois $uv \notin E$). Além disso, dado que Z é um corte de vértices que separa u e v , pelo menos um dos vértices internos desse caminho monocromático tem que estar em Z . Logo, $\sum_{z \in Z} x'_{z,c} \geq 1$, e, portanto, x' satisfaz 4.2.

Seja $x' \in \mathbb{R}^{nk}$ um vetor que satisfaz 4.1, 4.2 e 4.3 da formulação \mathcal{R} . A partir de 4.1 e

4.3, podemos concluir que x' é o vetor de incidência de uma coloração parcial, digamos C' . Suponha que C' não é uma coloração boa de G . Então, existem dois vértices não-adjacentes u e v aos quais C' atribui a cor c e tal que, em cada caminho entre esses vértices, existe um vértice que não recebe a cor c da coloração C' . Seja S o conjunto de vértices do subgrafo maximal monocromático de (G, C') que contém v . Note que $Z = N(S)$ é um corte de vértices que separa u e v , e todos os vértices em Z não possuem a cor c . Portanto, o vetor x' viola a desigualdade 4.2 com respeito a u , v , c e Z , uma contradição. \square

Observação 2. *Seja x^* uma solução ótima de \mathcal{R} e denote por C^* a coloração correspondente. Claramente, C^* é uma solução ótima para o problema RC, e a distância de recoloração de (G, C, w) é $\sum_{v \in V} w(v) - \sum_{v \in V} \sum_{c \in \mathcal{C}} w_{v,c} x_{v,c}^*$. Note que se a função w atribui peso 1 aos vértices com cor e peso 0 aos vértices sem cor, então minimizar o número de trocas de cor é equivalente a maximizar o número de vértices que mantêm a sua cor inicial.*

4.1.1 Desigualdades válidas

Dados $X, Y \subseteq V$, tais que $X \cap Y = \emptyset$, e $c, d \in \mathcal{C}$ definimos a coloração $C_{Y,d}^{X,c} : V \rightarrow \mathcal{C} \cup \{\emptyset\}$ de $G = (V, E)$ como sendo:

$$C_{Y,d}^{X,c}(z) = \begin{cases} c, & \text{se } z \in X \\ d, & \text{se } z \in Y \\ \emptyset, & \text{caso contrário.} \end{cases}$$

Seja ainda a coloração $C^{X,c} : V \rightarrow \mathcal{C} \cup \{\emptyset\}$ definida da seguinte forma:

$$C^{X,c}(z) = \begin{cases} c, & \text{se } z \in X \\ \emptyset, & \text{caso contrário.} \end{cases}$$

Por simplicidade, usamos a notação $C^{v,c}$ em vez de $C^{X,c}$, se $X = \{v\}$. Ademais, escrevemos $C^{G',c}$ como uma notação simplificada de $C^{V(G'),c}$, onde G' é um grafo.

Proposição 2. *O politopo $\mathcal{Q}(G)$ possui dimensão plena, i.e. $\dim(\mathcal{Q}(G)) = nk$.*

Prova. Considere o conjunto de colorações $A = \{C^{v,c} \mid v \in V \text{ e } c \in \mathcal{C}\}$. Claramente,

cada coloração parcial em A é convexa e boa. Logo, o conjunto de vetores $\chi[A]$ está contido em $\mathcal{Q}(G)$. Note que os vetores em $\chi[A]$ são exatamente as colunas da matriz identidade de ordem nk , portanto, existem nk vetores linearmente independentes que pertencem a $\mathcal{Q}(G)$. Note que $\mathbf{0} \in \mathcal{Q}(G)$, e portanto $\dim(\mathcal{Q}(G)) = \text{posto-afim}(\mathcal{Q}(G)) - 1 = \text{posto}(\mathcal{Q}(G)) = nk$. \square

Proposição 3. *Para cada $v \in V$ e $c \in \mathcal{C}$, $x_{v,c} \geq 0$ define faceta de $\mathcal{Q}(G)$.*

Prova. Tome um vértice $v \in V$ e uma cor $c \in \mathcal{C}$. Para mostrar que $x_{v,c} \geq 0$ define faceta é suficiente exibir $nk - 1$ vetores linearmente independentes em $\hat{F} = \{x \in \mathcal{Q}(G) \mid x_{v,c} = 0\}$. Considere a coleção de colorações parciais $A = \{C^{z,d} \mid z \in V \setminus \{v\} \text{ e } d \in \mathcal{C}\} \cup \{C^{v,d} \mid d \in \mathcal{C} \setminus \{c\}\}$. Claramente, cada coloração parcial em A atribui cor a exatamente um vértice e não atribui cor c ao vértice v , portanto, $\chi[A] \subseteq \hat{F}$. Observe que $\chi[A]$ é um subconjunto dos vetores-coluna da matriz identidade de ordem nk e que $|\chi[A]| = nk - 1$. Portanto, \hat{F} contém $nk - 1$ vetores linearmente independentes. \square

Em cada um dos teoremas e proposições a seguir, para provar que uma face $\hat{F} = \{x \in \mathcal{Q}(G) \mid \hat{\lambda}^T x = \hat{\lambda}_0\}$ é faceta de $\mathcal{Q}(G)$, mostramos que se uma face $F = \{x \in \mathcal{Q}(G) \mid \lambda^T x = \lambda_0\}$ de $\mathcal{Q}(G)$ contém \hat{F} então existe $a \in \mathbb{R}_{\geq}$ tal que $\lambda = a\hat{\lambda}$ e $\lambda_0 = a\hat{\lambda}_0$.

Proposição 4. *Para cada $v \in V$, $\sum_{c \in \mathcal{C}} x_{v,c} \leq 1$ define faceta de $\mathcal{Q}(G)$.*

Prova. Considere um vértice $v \in V$ e uma face $F = \{x \in \mathcal{Q}(G) \mid \lambda x = \lambda_0\}$ de $\mathcal{Q}(G)$ tal que $\hat{F} = \{x \in \mathcal{Q}(G) \mid \hat{\lambda}^T x = \hat{\lambda}_0\} \subseteq F$, onde a desigualdade $\hat{\lambda}^T x \leq \hat{\lambda}_0$ corresponde a $\sum_{c \in \mathcal{C}} x_{v,c} \leq 1$. Primeiramente, tomamos $\dot{c}, \ddot{c} \in \mathcal{C}$, onde $\dot{c} \neq \ddot{c}$, e definimos os conjuntos de colorações $A_1 = \{C_{u,d}^{v,\dot{c}} \mid u \in V \setminus \{v\}, d \in \mathcal{C} \setminus \{\dot{c}\}\} \cup \{C^{v,\dot{c}}\}$ e $A_2 = \{C_{u,\dot{c}}^{v,\ddot{c}} \mid u \in V \setminus \{v\}\} \cup \{C^{v,\ddot{c}}\}$. Observe que todas as colorações em $A_1 \cup A_2$ são colorações boas de G e dessa forma $\chi[A_1 \cup A_2] \subseteq \mathcal{Q}(G)$. Note ainda que o vértice v recebe exatamente uma cor em cada coloração de $A_1 \cup A_2$, assim, $\chi[A_1 \cup A_2] \subseteq \hat{F}$. Logo, essas colorações mostram que $\lambda_{u,c} = 0 \forall u \in V \setminus \{v\} \forall c \in \mathcal{C}$.

Por fim, considere o conjunto $A_3 = \{C^{v,c} \mid c \in \mathcal{C}\}$. Claramente, $\chi[A_3] \subseteq \hat{F}$. Logo, $\lambda_{v,c} = \lambda_0 \forall c \in \mathcal{C}$. Note que $\lambda_0 \geq 0$, caso contrário $\lambda^T x \leq \lambda_0 < 0 \forall x \in \mathcal{Q}(G)$, absurdo, pois

$\mathbf{0} \in \mathcal{Q}(G)$. Portanto, $\lambda = a\hat{\lambda}$ onde $a := \lambda_0$, dessa forma, \hat{F} é faceta de $\mathcal{Q}(G)$. \square

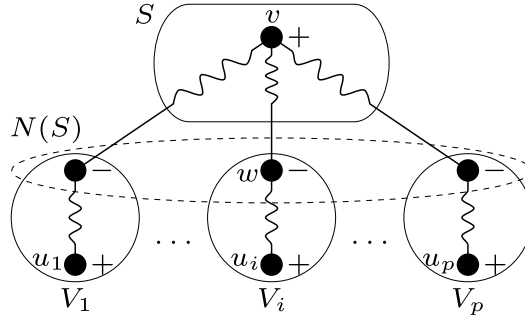


Figura 4.1: Sinais dos coeficientes da desigualdade (4.4) quando G é uma árvore.

Teorema 6. *Sejam $S \subset V$ um conjunto não-vazio tal que $G[S]$ é conexo e, além disso, a partição $\{V_1, V_2, \dots, V_p\}$ de $V \setminus S$ induzida pelas componentes conexas $G[V_i]$ de $G - S$ é tal que $V_i \setminus N(S) \neq \emptyset$ para todo $i \in \{1, 2, \dots, p\}$. Para cada $c \in \mathcal{C}$, $v \in S$ e $u_i \in V_i \setminus N(S)$ onde $i \in \{1, 2, \dots, p\}$, a desigualdade*

$$\sum_{i=1}^p x_{u_i, c} - \sum_{w \in N(S)} x_{w, c} + x_{v, c} \leq 1 \quad (4.4)$$

define faceta de $\mathcal{Q}(G)$, se para cada $i \in \{1, \dots, p\}$ e cada $w \in V_i \cap N(S)$ existe um caminho entre u_i e w tal que seus vértices internos pertencem a $V_i \setminus N(S)$.

Prova. Sejam $S, \{V_1, V_2, \dots, V_p\}, v \in S, c \in \mathcal{C}$, e $(u_1, \dots, u_p) \in (V_1 \setminus N(S)) \times \dots \times (V_p \setminus N(S))$ tais como no enunciado do teorema, e suponha que as outras hipóteses sobre G são também satisfeitas. É fácil ver que a desigualdade do Teorema 6 é válida para $\mathcal{Q}(G)$, pois cada caminho monocromático entre u_i e qualquer outro vértice que não está em V_i tem que conter pelo menos um vértice do conjunto $V_i \cap N(S)$ (veja Figura 4.1).

Considere uma face $F = \{x \in \mathcal{Q}(G) \mid \lambda x = \lambda_0\}$ de $\mathcal{Q}(G)$ tal que $\hat{F} = \{x \in \mathcal{Q}(G) \mid \hat{\lambda}^T x = \hat{\lambda}_0\} \subseteq F$, onde a desigualdade $\hat{\lambda}^T x \leq \hat{\lambda}_0$ corresponde a $\sum_{i=1}^p x_{u_i, c} - \sum_{z \in N(S)} x_{z, c} + x_{v, c} \leq 1$.

Seja $A_1^i = \{C_{z, d}^{v, c} \mid z \in V_i \text{ e } d \in \mathcal{C} \setminus \{c\}\} \cup \{C^{v, c}\}$ um conjunto de colorações, onde $i \in \{1, 2, \dots, p\}$. Note que cada coloração em A_1^i atribui a cor c somente ao vértice v , a cor d , diferente de c , a no máximo um vértice de V_i e não atribui qualquer cor aos demais. Dessa maneira, é fácil ver que $\chi[\cup_{i=1}^p A_1^i] \subseteq \hat{F}$. Logo, a partir dos vetores das colorações em $\cup_{i=1}^p A_1^i$ obtemos que $\lambda_{z, d} = 0 \forall i \in \{1, \dots, p\}, \forall z \in V_i$ e $\forall d \in \mathcal{C} \setminus \{c\}$.

Considere agora o conjunto de colorações $A_2 = \{C_{z,d}^{u_1,c} \mid z \in S \text{ e } d \in \mathcal{C} \setminus \{c\}\} \cup \{C^{u_1,c}\}$. Cada coloração de A_2 atribui a cor c ao vértice u_1 , a cor d , diferente de c , a no máximo um vértice de S e nenhuma cor a todos os outros vértices. Logo, temos $\chi[A_2] \subseteq \hat{F}$. A partir desses vetores, obtemos que $\lambda_{z,d} = 0 \forall z \in S \text{ e } \forall d \in \mathcal{C} \setminus \{c\}$. Uma vez que $G[S]$ é conexo, existe um caminho em S entre v e qualquer vértice $z \in S \setminus \{v\}$. Denote um tal caminho por P_{vz} . É fácil ver que as colorações parciais $C^{P_{vz},c}$ e $C^{P_{vz}-z,c}$ são boas, pois P_{vz} e $P_{vz} - z$ são subgrafos conexos de G . Dessa maneira, temos que $\chi(C^{P_{vz},c}), \chi(C^{P_{vz}-z,c}) \in \hat{F}$. Portanto, $\lambda_{z,c} = 0 \forall z \in S \setminus \{v\}$.

Para cada $i \in \{1, \dots, p\}$, se $V_i \setminus (N(S) \cup \{u_i\}) \neq \emptyset$, então considere um vértice z nesse conjunto. Se existe algum caminho entre u_i e z que passe apenas em vértices de $V_i \setminus N(S)$ então defina $P_{u_i z}$ como sendo este caminho e considere as colorações $C^{P_{u_i z},c}$ e $C^{P_{u_i z}-z,c}$. Caso contrário, deduzimos da hipótese que, para algum $w \in N(S) \cap V_i$, existem caminhos $P_{u_i w}$ de u_i até w e P_{zw} de z até w tais que os vértices internos de ambos os caminhos pertencem a $V_i \setminus N(S)$. Então, basta tomar w como sendo o vértice em $V_i \cap N(S)$ mais próximo de z em algum caminho de u_i até z na componente $G[V_i]$. Nessa situação, seja $P_{u_i z} = P_{u_i w} P_{zw}$ e considere as colorações $C^{P_{vw} P_{u_i z},c}$ e $C^{P_{vw} P_{u_i z}-z,c}$, onde P_{vw} é um caminho de v até w tal que todos os vértices internos pertencem a S . No primeiro caso é fácil ver que as colorações consideradas atribuem cor c a um caminho cujos vértices internos pertencem a $V_i \setminus N(S)$ e tal que uma das extremidades é u_i , logo, $\chi(C^{P_{u_i z},c}), \chi(C^{P_{u_i z}-z,c}) \in \hat{F}$. No segundo caso, colorimos com a cor c os vértices de um caminho de v até w , onde os vértices internos pertencem a S , e também os vértices de um caminho de u_i até z que passa por w e por nenhum outro vértice de $V_i \cap N(S)$. Dessa forma, temos que $\chi(C^{P_{vw} P_{u_i z},c}), \chi(C^{P_{vw} P_{u_i z}-z,c}) \in \hat{F}$. A partir dos vetores dessas colorações, obtemos $\lambda_{z,c} = 0 \forall i \in \{1, 2, \dots, p\} \forall z \in V_i \setminus (N(S) \cup \{u_i\})$.

Seja $A_3 = \{C^{u_i,c} \mid i \in \{1, \dots, p\}\}$. Claramente, cada coloração parcial em A_3 é boa, logo, $\chi[A_3] \subseteq \hat{F}$. A partir desses vetores obtemos que $\lambda_{u_i,c} = \lambda_0 \forall i \in \{1, \dots, p\}$. É fácil ver que a coloração $C^{v,c}$ é boa e, portanto, $\chi(C^{v,c}) \in \hat{F}$. Logo, $\lambda_{v,c} = \lambda_0$. Para cada $i \in \{1, \dots, p\}$, seja $P_{u_i z}$ um caminho entre u_i e z tal que todos os vértices internos pertencem a $V_i \setminus N(S)$ e $z \in V_i \cap N(S)$. Dado que $G[S]$ é conexo, existe um caminho entre v e w tal que todos seus vértices estão em S e $wz \in \partial(S)$. Seja P_{vw} tal caminho. Observe que $P_{vw} P_{u_i z}$ induz um subgrafo conexo de G , logo, $C^{P_{vw} P_{u_i z},c}$ é uma coloração parcial convexa boa e $\chi(C^{P_{vw} P_{u_i z},c}) \in \hat{F}$.

Dessa forma, temos $\lambda_{v,c} + \lambda_{z,c} + \lambda_{u_i,c} = \lambda_0$ o que implica $\lambda_{z,c} = -\lambda_0$. Portanto, $\lambda_{z,c} = -\lambda_0$ $\forall i \in \{1, \dots, p\} \forall z \in V_i \cap N(S)$. Por fim, observe que $\lambda_0 \geq 0$, caso contrário $\lambda^T x < 0$ $\forall x \in \mathcal{Q}(G)$, o que é um absurdo, pois $\mathbf{0} \in \mathcal{Q}(G)$. Portanto, $\lambda^T = a\hat{\lambda}^T$, onde $a := \lambda_0$. \square

Quando $p = 1$, a desigualdade anterior é equivalente à desigualdade 4.2 de \mathcal{R} e, portanto, pode ser separada em tempo polinomial encontrando-se um corte mínimo de vértices.

4.2 Poliedro do problema RC em árvores

Nesta seção, consideramos o problema de recoloração convexa restrito a árvores, o qual denotamos por RCA. Uma instância desse problema é uma tripla (T, C, w) , onde $T = (V, E)$ é uma árvore, $C : V \rightarrow \mathcal{C} \cup \{\emptyset\}$ é uma coloração, $n := |V|$ e $k := |\mathcal{C}|$. Os outros parâmetros e convenções são análogos aos que foram definidos para o problema RC. Adicionalmente, para vértices distintos u, v em T , denotamos por P_{uv} o caminho entre u e v na árvore T .

Uma formulação linear inteira \mathcal{T} pode ser obtida a partir de \mathcal{R} substituindo a desigualdade (2) por

$$x_{u,c} + x_{v,c} - x_{w,c} \leq 1, \quad \text{for } u, v \in V, w \in V(P_{uv}) \setminus \{u, v\}, c \in \mathcal{C}.$$

Assim, podemos definir a formulação \mathcal{T} para o problema RCA da seguinte forma:

$$\begin{aligned} (\mathcal{T}) \quad & \max \sum_{v \in V} \sum_{c \in \mathcal{C}} w_{v,c} x_{v,c} \\ & \text{s.a.} \sum_{c \in \mathcal{C}} x_{v,c} \leq 1 \quad \forall v \in V \end{aligned} \tag{4.5}$$

$$x_{u,c} - x_{v,c} + x_{w,c} \leq 1 \quad \forall u, w \in V, \forall v \in V(P_{uw}) \setminus \{u, w\}, \forall c \in \mathcal{C}, \tag{4.6}$$

$$x_{v,c} \in \{0, 1\} \quad \forall v \in V, \forall c \in \mathcal{C}. \tag{4.7}$$

Definimos $\mathcal{P}(T)$ o poliedro associado ao problema de recoloração convexa como:

$$\mathcal{P}(T) := \text{conv}\{\chi(C') \mid C' \text{ é uma coloração boa de } T\}.$$

Uma vez que \mathcal{T} é um caso particular de \mathcal{R} , os Corolários 1 e 2 seguem diretamente das Proposições 1 e 2.

Corolário 1. $\mathcal{P}(T) = \text{conv}\{x \in \mathbb{R}^{nk} \mid x \text{ satisfaz as restrições 4.5, 4.6 e 4.7 de } \mathcal{T}\}.$

Corolário 2. *O politopo $\mathcal{P}(T)$ tem dimensão plena, i.e. $\dim(\mathcal{P}(T)) = nk$.*

4.2.1 Desigualdades válidas

A partir das Proposições 3 e 4 apresentadas na seção 4.1, podemos obter facilmente os Corolários 3 e 4, respectivamente.

Corolário 3. *Para cada $v \in V$ e $c \in \mathcal{C}$, $x_{v,c} \geq 0$ define faceta de $\mathcal{P}(T)$.*

Corolário 4. *Para cada $v \in V$, $\sum_{c \in \mathcal{C}} x_{v,c} \leq 1$ define faceta de $\mathcal{P}(T)$.*

O Teorema 7 é exatamente o Teorema 6 especializado para o caso em que G é uma árvore.

Teorema 7. DESIGUALDADES DE CONVEXIDADE-ESTRELA. *Seja $S \subset V$ um conjunto não-vazio tal que $T[S]$ é conexo e, além disso, a partição $\{V_1, V_2, \dots, V_p\}$ de $V \setminus S$ induzida pelas sub-árvores $T[V_i]$ de $T - S$ é tal que $|V_i| \geq 2$ para todo $i \in \{1, 2, \dots, p\}$. Para cada $c \in \mathcal{C}$, $v \in S$ e $u_i \in V_i \setminus N(S)$ onde $i \in \{1, 2, \dots, p\}$, a desigualdade*

$$\sum_{i=1}^p x_{u_i,c} - \sum_{w \in N(S)} x_{w,c} + x_{v,c} \leq 1 \quad (4.8)$$

define faceta do politopo $\mathcal{P}(T)$.

Nos teoremas a seguir, apresentamos uma classe não-trivial de facetas que consiste em uma generalização da desigualdade 4.6. Embora o Teorema 9 seja uma generalização do Teorema 8, decidimos manter a prova desse por ser de mais fácil compreensão e porque a Proposição 5 apresenta a demonstração da validade das desigualdades desse teorema por meio do processo de combinação e arredondamento a partir das desigualdades 4.6.

Dados vértices distintos $u, w \in V$, definimos uma ordem total \prec dos elementos em $V(P_{uw})$ de acordo com a ordem em que cada vértice aparece no caminho P_{uw} a partir do vértice u .

Proposição 5. Para cada $c \in \mathcal{C}$, $u, w \in V$ e $\{u_1, v_1, u_2, v_2, \dots, u_t, v_t, u_{t+1}\} \subseteq V(P_{uw})$, tal que $t \in \mathbb{Z}_{>}$, $u = u_1$, $w = u_{t+1}$ e $u_1 \prec v_1 \prec \dots \prec u_t \prec v_t \prec u_{t+1}$, a desigualdade

$$\sum_{i=1}^t x_{u_i, c} - \sum_{i=1}^t x_{v_i, c} + x_{w, c} \leq 1$$

é válida para o politopo $\mathcal{P}(T)$.

Prova. Tome quaisquer $c \in \mathcal{C}$, $u, w \in V$ e $\{u = u_1, v_1, u_2, v_2, \dots, u_t, v_t, u_{t+1} = w\} \subseteq V(P_{uw})$, tal que $t \in \mathbb{Z}_{>}$ e $u_1 \prec v_1 \prec \dots \prec u_t \prec v_t \prec u_{t+1}$. A demonstração será por indução sobre t . Se $t = 1$, então a proposição é verdadeira trivialmente, pois basta observar que as desigualdades desse caso são exatamente as desigualdades 4.6 que definem o politopo $\mathcal{P}(T)$.

Se $t > 1$, então suponha que para todo inteiro $1 \leq t' < t$ a seguinte afirmação é verdadeira: Para cada $d \in \mathcal{C}$, $p, r \in V$ e $\{p = p_1, q_1, p_2, q_2, \dots, p_{t'}, q_{t'}, p_{t'+1} = r\} \subseteq V(P_{pr})$, tal que $p_1 \prec q_1 \prec \dots \prec p_{t'} \prec q_{t'} \prec p_{t'+1}$, a desigualdade $\sum_{i=1}^{t'} x_{p_i, c} - \sum_{i=1}^{t'} x_{q_i, c} + x_{r, c} \leq 1$ é válida para $\mathcal{P}(T)$. Por hipótese indutiva, as desigualdades $\sum_{i=1}^{t-1} x_{u_i, c} - \sum_{i=1}^{t-1} x_{v_i, c} + x_{u_t, c} \leq 1$ (a), $\sum_{i=2}^t x_{u_i, c} - \sum_{i=2}^t x_{v_i, c} + x_{w, c} \leq 1$ (b) e $x_{u, c} - x_{v_t, c} + x_{w, c} \leq 1$ (c) são válidas para $\mathcal{P}(T)$. Ademais, a desigualdade $-x_{v_1, c} \leq 0$ (d) é claramente válida. Portanto, a partir da soma das desigualdades (a), (b), (c) e (d) obtemos $\sum_{i=1}^t x_{u_i, c} - \sum_{i=1}^t x_{v_i, c} + x_{w, c} \leq \frac{3}{2}$. Uma vez que o lado esquerdo da última desigualdade é inteiro, a desigualdade $\sum_{i=1}^t x_{u_i, c} - \sum_{i=1}^t x_{v_i, c} + x_{w, c} \leq 1$ é válida para $\mathcal{P}(T)$. \square

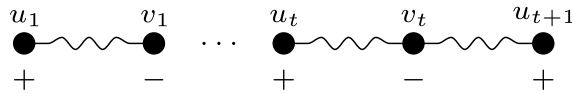


Figura 4.2: Sinais dos coeficientes da desigualdade (4.9) obtida a partir de P_{uw} , onde $u = u_1$ e $w = u_{t+1}$.

Teorema 8. DESIGUALDADES DE CONVEXIDADE-CAMINHO. Para cada $c \in \mathcal{C}$, $u, w \in V$ e $\{u_1, v_1, u_2, v_2, \dots, u_t, v_t, u_{t+1}\} \subseteq V(P_{uw})$ tal que $t \in \mathbb{Z}_{>}$, $u = u_1$, $w = u_{t+1}$ e $u_1 \prec v_1 \prec \dots \prec u_t \prec v_t \prec u_{t+1}$, a desigualdade

$$\sum_{i=1}^t x_{u_i, c} - \sum_{i=1}^t x_{v_i, c} + x_{w, c} \leq 1 \quad (4.9)$$

define faceta do politopo $\mathcal{P}(T)$.

Prova. Tome uma cor $c \in \mathcal{C}$, vértices $u, w \in V$, $S = \{u = u_1, v_1, u_2, v_2, \dots, u_t, v_t, u_{t+1} = w\} \subseteq V(P_{uw})$, tal que $t \in \mathbb{Z}_{>}$ e $u_1 \prec v_1 \prec \dots \prec u_t \prec v_t \prec u_{t+1}$, e uma face $F = \{x \in \mathcal{P}(T) \mid \lambda x = \lambda_0\}$ de $\mathcal{P}(T)$ tal que $\hat{F} = \{x \in \mathcal{P}(T) \mid \hat{\lambda}^T x = \hat{\lambda}_0\} \subseteq F$, onde a desigualdade $\hat{\lambda}^T x \leq \hat{\lambda}_0$ corresponde a $\sum_{i=1}^t x_{u_i, c} - \sum_{i=1}^t x_{v_i, c} + x_{w, c} \leq 1$. Os coeficientes da desigualdade associada à P_{uw} são mostrados na Figura 4.2. Para cada $z \in V \setminus S$, definimos S_z como o conjunto de vértices da componente de $T - z$ tal que $|S_z \cap S|$ é ímpar. Seja $A_1 = \{C_{z,d}^{S_z, c}, C^{S_z, c} \mid z \in V \setminus S \text{ e } d \in \mathcal{C}\}$ um conjunto de colorações de T . Observe que, para cada $z \in V \setminus S$, $T[S_z]$ e $T[S_z \cup \{z\}]$ são grafos conexos, $|S_z \cap S|$ é ímpar e os vértices em $S_z \cap S$ são consecutivos na ordem \prec , portanto, $\chi(C_{z,d}^{S_z, c}), \chi(C^{S_z, c}) \in \hat{F}$. Logo, a partir das colorações em A_1 temos que $\lambda_{z,d} = 0 \forall z \in V \setminus S$ e $\forall d \in \mathcal{C}$.

Considere os conjuntos de colorações $A_2 = \{C_{z,d}^{w,c} \mid z \in S \setminus \{w\} \text{ e } d \in \mathcal{C} \setminus \{c\}\} \cup \{C^{w,c}\}$ e $A_3 = \{C_{w,d}^{u,c} \mid d \in \mathcal{C} \setminus \{c\}\} \cup \{C^{u,c}\}$. Note que cada coloração em A_2 atribui a cor c apenas ao vértice w e que cada uma em A_3 atribui a cor c somente ao vértice u . Logo, $\chi[A_2 \cup A_3] \subseteq \hat{F}$. Os vetores das colorações em A_2 mostram que $\lambda_{z,d} = 0 \forall z \in S \setminus \{w\} \forall d \in \mathcal{C} \setminus \{c\}$ e a partir das colorações em A_3 obtemos $\lambda_{w,d} = 0 \forall d \in \mathcal{C} \setminus \{c\}$. Portanto, $\lambda_{z,d} = 0 \forall z \in S \forall d \in \mathcal{C} \setminus \{c\}$.

Sejam $A_4 = \{C^{u_i, c} \mid i \in \{1, 2, \dots, t+1\}\}$ e $A_5 = \{C^{P_{u_i u_{i+1}}, c} \mid i \in \{1, 2, \dots, t\}\}$. Observe que cada coloração em A_4 atribui a cor c a exatamente um vértice u_i , para algum $i \in \{1, 2, \dots, t+1\}$ e, portanto, $\chi[A_4] \subseteq \hat{F}$. Assim, é fácil ver que $\lambda_{u_i, c} = \lambda_0 \forall i \in \{1, 2, \dots, t+1\}$. Para cada $i \in \{1, 2, \dots, t\}$, $P_{u_i u_{i+1}}$ é o caminho entre os vértices u_i e u_{i+1} em T . Note que $V(P_{u_i u_{i+1}}) \cap S = \{u_i, v_i, u_{i+1}\}$, logo, $\chi(C^{P_{u_i u_{i+1}}, c}) \in \hat{F}$. Portanto, obtemos $\lambda_{u_i, c} + \lambda_{v_i, c} + \lambda_{u_{i+1}, c} = \lambda_0$ o que implica $\lambda_{v_i, c} = -\lambda_0 \forall i \in \{1, 2, \dots, t\}$. Note que $a := \lambda_0 \geq 0$, pois $\mathbf{0} \in \mathcal{P}(T)$ e $\lambda^T x \leq \lambda_0$ é uma desigualdade válida para $\mathcal{P}(T)$. Portanto, \hat{F} é faceta de $\mathcal{P}(T)$, pois $\lambda = a\hat{\lambda}$ e $a \geq 0$. \square

Antes de demonstrar o resultado principal dessa seção, apresentaremos algumas definições utilizadas na sua prova. Seja $T' \subseteq T$ uma árvore enraizada em r e $f : V(T') \rightarrow \mathbb{Z}_{\geq}$ uma função que associa a cada vértice a quantidade de filhos em T' . A partir de T' , denotamos por $W \subseteq V(T')$ o conjunto formado pelos vértices que possuem mais de um filho em T' , pelas folhas e pela raiz. Mais formalmente, $W := \{v \in V(T') \mid f(v) > 1 \text{ ou } f(v) = 0 \text{ ou } v = r\}$.

Chamamos os vértices em W de *fixos*. A partir desse conjunto, definimos a função $p : W \rightarrow W$ de tal forma que $p(r) = r$ e $p(v)$ é o ancestral de v com maior profundidade na árvore T' e que está em W , se $v \in V \setminus \{r\}$. Para cada $v \in W$, seja $D_v \subseteq W$ o conjunto dos vértices de W com profundidade maior que a de v e alcançáveis a partir de v por caminhos cujos vértices internos pertencem a $V(T') \setminus W$, mais formalmente, $D_v := \{u \in W \mid p(u) = v\}$. É fácil ver que $|D_v| = f(v) \forall v \in W$. Para cada $v \in W \setminus \{r\}$, tomamos um conjunto $W_v \subseteq V(P_{uv}) \setminus \{u, v\}$ tal que $|W_v|$ é par e $u := p(v)$. Suponha que os vértices em W_v sejam ordenados na sequência em que aparecem no caminho P_{uv} a partir de v , assim um vértice $w \in W_v$ é dito *positivo* (*negativo*) se existe um número ímpar (par) de vértices em W_v que o precedem nesse ordem. Dessa forma, podemos particionar o conjunto W_v em $W_v^+ := \{w \in W_v \mid w \text{ é positivo}\}$ e $W_v^- := \{w \in W_v \mid w \text{ é negativo}\}$.

Agora, estamos prontos para definir as desigualdades da forma $R(r) \leq 1$, denominadas *Desigualdades de convexidade-árvore* (associada à árvore T' com raiz r). Para cada $v \in W$, atribuímos coeficiente $1 - f(v)$ à variável $x_{v,c}$, se v não é folha de T' ; e coeficiente 1 à variável corresponde, caso contrário. Além disso, atribuímos coeficiente 1 e -1 às variáveis associadas aos vértices pertencentes à $\bigcup_{v \in W \setminus \{r\}} W_v^+$ e $\bigcup_{v \in W \setminus \{r\}} W_v^-$, respectivamente. Na Figura 4.3 apresentamos uma árvore T' com raiz r que define a desigualdade $R(r) = -2x_{r,c} - x_{v_2,c} + x_{v_3,c} - x_{v_4,c} + x_{v_7,c} + x_{v_8,c} + x_{w_+,c} - x_{w_-,c} - 2x_{v_5,c} + x_{v_9,c} + x_{v_{10},c} + x_{v_{11},c} + x_{v_6,c} \leq 1$. Observe na Figura 4.3(a) que $W = \{v_1, v_2, \dots, v_{11}\}$, $p(v_2) = p(v_5) = p(v_6) = r$ e $p(v_4) = v_2$. A Figura 4.3(b) apresenta os coeficientes da desigualdade $R(r) \leq 1$.

No próximo teorema, usamos uma definição recursiva dessa classe de desigualdades.

Teorema 9. DESIGUALDADES DE CONVEXIDADE-ÁRVORE. *Sejam $T' \subseteq T$ uma árvore enraizada em r , W o conjunto dos vértices fixos e $c \in \mathcal{C}$ uma cor. Para cada coleção $\{W_v\}_{v \in W \setminus \{r\}}$ tal que $W_v \subseteq V(P_{p(v)v})$ e $|W_v|$ é par para todo $v \in W \setminus \{r\}$, temos que a desigualdade*

$$R(r) \leq 1 \tag{4.10}$$

define faceta de $\mathcal{P}(T)$, se T' tem mais de uma folha, onde R é uma função recursiva sobre

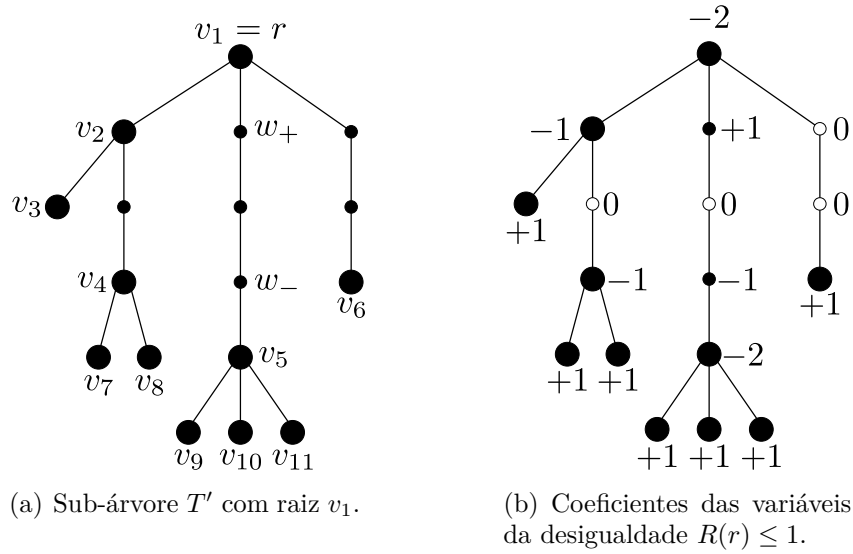


Figura 4.3: Exemplo de desigualdade do Teorema 9.

o conjunto W definida da seguinte maneira:

$$R(v) = \begin{cases} x_{v,c}, & \text{se } v \text{ é folha de } T', \\ \sum_{u \in D_v} \left(R(u) + \sum_{w \in W_u^+} x_{w,c} - \sum_{w \in W_u^-} x_{w,c} \right) - (f(v) - 1)x_{v,c}, & \text{caso contrário.} \end{cases}$$

Prova. Por indução na profundidade dos vértices em W , mostraremos que $R(v) \leq 1$ para todo $v \in W$ e toda coloração boa de T . Se v é uma folha de T' , então o resultado segue trivialmente. Se v não é folha de T' , então suponha que $R(u) \leq 1$ para todo $u \in W$ com profundidade maior que a de v e para toda coloração boa de T . Por hipótese indutiva, temos que $R(u) \leq 1$ para cada $u \in D_v$. Logo, para qualquer coloração boa de T podemos concluir que $R(u) + \sum_{w \in W_u^+} x_{w,c} - \sum_{w \in W_u^-} x_{w,c} \leq 1$, pois os vértices positivos e negativos de W_u se alternam no caminho P_{uv} e $|W_u^+| = |W_u^-| \forall u \in D_v$. Em qualquer coloração boa de T , se ao menos dois vértices em componentes distintas de $T'_v - v$ têm a cor c , então v também possui a cor c , onde T'_v é a subárvore de T' com raiz em v . Portanto, a desigualdade $R(v) = \sum_{u \in D_v} (R(u) + \sum_{w \in W_u^+} x_{w,c} - \sum_{w \in W_u^-} x_{w,c}) - (f(v) - 1)x_{v,c} \leq 1$ é satisfeita para qualquer coloração boa de T . Ademais, note que para qualquer $v \in W$ temos $R(v) = 1$ quando a coloração boa de T é $C^{T'_v, c}$.

Considere uma face $F = \{x \in \mathcal{P}(T) \mid \lambda x = \lambda_0\}$ de $\mathcal{P}(T)$ tal que $\hat{F} = \{x \in \mathcal{P}(T) \mid \hat{\lambda}^T x = \hat{\lambda}_0\} \subseteq F$, onde a desigualdade $\hat{\lambda}^T x \leq \hat{\lambda}_0$ corresponde a $R(r) \leq 1$. Para cada $z \in V(T) \setminus V(T')$,

seja v o vértice mais próximo de z em T' e considere as colorações $C_{P_{vz-v,d}}^{T',c}$, $C_{P_{vz-v-z,d}}^{T',c}$, onde d é uma cor qualquer em \mathcal{C} . Claramente, essas são colorações boas de T e em cada uma delas temos $R(r) = 1$. Portanto, os vetores $\chi(C_{P_{vz-v,d}}^{T',c}), \chi(C_{P_{vz-v-z,d}}^{T',c}) \in F$. Logo, podemos concluir que $\lambda_{z,d} = 0 \forall z \in V(T) \setminus V(T')$ e $\forall d \in \mathcal{C}$. Considere duas folhas distintas v, v' de T' . Sejam $A_1 := \{C_{z,d}^{v,c} \mid z \in V(T') \setminus \{v\} \text{ e } d \in \mathcal{C} \setminus \{c\}\} \cup \{C^{v,c}\}$ e $A_2 := \{C_{v,d}^{v',c} \mid d \in \mathcal{C} \setminus \{c\}\} \cup \{C^{v',c}\}$ conjuntos de colorações de T . Observe que cada coloração em A_1 (em A_2) é boa e atribui a cor c exclusivamente ao vértice v (ao vértice v'). Uma vez que v e v' são folhas de T' , temos $R(r) = 1$ para qualquer coloração de $A_1 \cup A_2$ e, portanto, $\chi[A_1 \cup A_2] \subseteq F$. Logo, podemos concluir que $\lambda_{z,d} = 0 \forall z \in V(T')$ e $\forall d \in \mathcal{C} \setminus \{c\}$. Considere agora dois vértices $v \in W \setminus \{r\}$ e $z \in V(P_{p(v)}) \setminus W_v$. Se $|V(P_{zv}) \cap W_v|$ é par, então $C_{P_{zv},c}^{T',c}$ e $C_{P_{zv-z},c}^{T',c}$ são colorações boas de T . Note que em ambas colorações temos $R(v) = 1$, pois todos os vértices da subárvore T'_v recebem a cor c . Além disso, o caminho P_{zv} contém a mesma quantidade de vértices positivos e negativos o que implica $R(r) = R(v) = 1$ em ambas colorações. Logo, os vetores $\chi(C_{P_{zv},c}^{T',c})$ e $\chi(C_{P_{zv-z},c}^{T',c})$ pertencem à face F . Se $|V(P_{zv}) \cap W_v|$ é ímpar, então seja v' o vértice positivo tal que $V(P_{v'z}) \cap W_v = \{v'\}$. Uma vez que $|W_v|$ é par, o vértice v' existe. É fácil ver que as colorações $C^{P_{v'z},c}$ e $C^{P_{v'z-z},c}$ são boas e que em ambas temos $R(r) = R(p(v)) = 1$. Logo, os vetores $\chi(C^{P_{v'z},c})$ e $\chi(C^{P_{v'z-z},c})$ pertencem à face F . A partir dos vetores das colorações de ambos os casos podemos concluir que $\lambda_{z,c} = 0 \forall z \in V(T') \setminus (W \cup \bigcup_{v \in W \setminus \{r\}} W_v)$.

Dado um vértice $v \in W$, observe que $C^{T',c}$ é uma coloração boa tal que $R(r) = R(v) = 1$. Portanto, o vetor dessa coloração pertence a F o que implica $\sum_{z \in T'_v} \lambda_{z,c} = \lambda_0$. Para mostrar os coeficientes não-nulos da desigualdade (λ, λ_0) , observe que $A_3 := \{C^{z,c} \mid z \in \bigcup_{v \in W \setminus \{r\}} W_v^+\}$ é um conjunto de colorações boas de T tais que $R(r) = 1$. Logo, temos $\chi[A_3] \subseteq F$ o que implica $\lambda_{z,c} = \lambda_0 \forall z \in \bigcup_{v \in W \setminus \{r\}} W_v^+$.

Seja v um vértice em W . Se v é uma folha, então o vetor da coloração $C^{v,c}$ mostra que $\lambda_{v,c} = \lambda_0$. Se v não é folha, então considere os vértices $u \in D_v$ e $w \in W_u^-$. Se existem vértices $w', w'' \in W_u^+$ tais que o caminho $P_{w'w''}$ contém w e nenhum outro vértice de W_u^- , então $C^{P_{w'w''},c}$ é uma coloração boa de T tal que $R(r) = 1$. Caso contrário, existe um caminho $P_{uw'}$ em T' que contém w e nenhum outro vértice de W_u^- , onde $w' \in W_u^+$. Dessa forma, a coloração $C_{T'_u}^{P_{uw'},c}$ é boa e $R(r) = 1$, pois a coloração de T'_u com c garante $R(u) = 1$ e $P_{uw'}$ contém exatamente dois vértices em W_u , um positivo e outro negativo. No primeiro caso,

temos $\lambda_{w',c} + \lambda_{w,c} + \lambda_{w'',c} = \lambda_0$, no segundo, obtemos que $\sum_{z \in T'_u} \lambda_{z,c} + \lambda_{w,c} + \lambda_{w',c} = \lambda_0$. Uma vez que $\sum_{z \in T'_u} \lambda_{z,c} = \lambda_0$ e $\lambda_{w',c} = \lambda_{w'',c} = \lambda_0$, obtemos $\lambda_{w,c} = -\lambda_0$ em ambos os casos. Portanto, podemos concluir que $\lambda_{z,c} = -\lambda_0 \forall z \in \bigcup_{v \in W \setminus \{r\}} W_v^-$. Logo, temos que $\sum_{z \in T'_v} \lambda_{z,c} = \lambda_{v,c} + \sum_{u \in D_v} (\sum_{z \in T'_u} \lambda_{z,c} + \sum_{z \in W_u^+} \lambda_{z,c} - \sum_{z \in W_u^-} \lambda_{z,c}) = \lambda_{v,c} + \sum_{u \in D_v} \lambda_0$, onde a última igualdade segue porque $|W_u^+| = |W_u^-|$. Dessa forma, obtemos $\lambda_{v,c} = -(f(v) - 1)\lambda_0$. Note que $a := \lambda_0 \geq 0$, pois $\mathbf{0} \in \mathcal{P}(T)$ e $\lambda^T x \leq \lambda_0$ é uma desigualdade válida para $\mathcal{P}(T)$. Portanto, \hat{F} é faceta de $\mathcal{P}(T)$, pois $\lambda = a\hat{\lambda}$ e $a \geq 0$. \square

No Capítulo 5, mostraremos algoritmos de separação com tempo de execução $\mathcal{O}(kn)$ para cada uma das classes de desigualdades não-triviais apresentadas nesta seção.

Capítulo 5

Algoritmos de separação

Neste capítulo, apresentamos algoritmos de separação polinomiais para as classes de desigualdades 4.8, 4.9 e 4.10. Seja x um vetor pertencente a uma relaxação de $\mathcal{P}(T)$. Na Seção 5.1, mostramos um algoritmo que computa um corte de arestas de peso máximo em uma árvore obtida a partir de T cujas arestas são ponderadas usando os valores no vetor x . Na Seção 5.2, apresentamos um algoritmo para separação das desigualdades de convexidade-caminho e um para as desigualdades de convexidade-árvore.

Sejam $T = (V, E)$ uma árvore, $C : V \rightarrow \mathcal{C} \cup \{\emptyset\}$ uma coloração parcial de T , $n := |V|$ e $k := |\mathcal{C}|$. Neste capítulo, suponha uma ordem total qualquer sobre o conjunto de vértices V . Para quaisquer vértice $u, v \in V$, denotamos por T^u a árvore T enraizada no vértice u e T_v^u a sub-árvore de T^u enraizada no vértice v . Frequentemente, usamos a notação T' para nos referir à árvore T enraizada em um vértice qualquer.

5.1 Desigualdades de convexidade-estrela

Teorema 10. *As desigualdades de convexidade-estrela 4.8 podem ser separadas em $\mathcal{O}(kn)$.*

Prova. Para cada $c \in \mathcal{C}$ e cada $v \in V$, buscamos um conjunto S que contenha v e tal que $T[S]$ é conexo, e vértices $u_i \in V_i \setminus N(S)$ que maximizem o lado esquerdo da desigualdade $\sum_{i=1}^p x_{u_i, c} + x_{v, c} - \sum_{w \in N(S)} x_{w, c} \leq 1$, onde x é um vetor de uma relaxação de $\mathcal{P}(T)$. Note que para a desigualdade 4.8 sempre temos $|N(S)| = p$.

Fixos um vértice r e uma cor c , denotamos por T' a árvore T com raiz em r . Atribuímos

custos às arestas de T' da seguinte maneira. Para cada $z \in V(T')$, e cada w em $N(z)$ tal que w é filho de z , e não é uma folha, denotamos por w^* um vértice na subárvore $T'_w \subseteq T'$ com raiz em w tal que $x_{w^*,c} = \max\{x_{y,c} \mid y \in V(T'_w) \setminus \{w\}\}$. Note que se z é escolhido para estar em S , e $w \in \bar{S}$, então w^* faz o papel de algum vértice u_i . Sejam T'' a super-árvore obtida a partir de T' pela adição de um novo vértice ℓ' para cada folha ℓ de T' e uma aresta $\ell'\ell$, e $c' : E(T'') \rightarrow \mathbb{Q}$ uma função de pesos das suas arestas definida a seguir. Para cada aresta $\ell'\ell \in E(T'') \setminus E$, definimos $c'(\ell'\ell) = 0$. Atribua à cada aresta $zw \in E$, onde z é pai de w e w não é folha de T' , o peso $c'(zw) := x_{w^*,c} - x_{w,c}$. Se w é uma folha de T' , então definimos $c'(zw) = -\infty$. Queremos encontrar em T'' , com a função de pesos nas arestas c' , um corte de arestas com valor máximo, digamos $[S, \bar{S}]$, tal que $v \in S$ e $T''[S]$ é conexo. Observe que o conjunto de arestas $E(T'') \setminus E(T')$ define um corte trivial $[S', \bar{S}']$ de T'' com valor zero, onde $S' = V(T')$.

O algoritmo de separação percorre os vértices de T'' em ordem não-crescente de profundidade a partir dos vértices de maior profundidade. Em cada passo, esse algoritmo considera um vértice $v \in V(T'')$ para decidir se a aresta uv , onde u é pai de v em T'' , deve (ou não) ser adicionada ao conjunto de arestas E' , o corte de arestas corrente. Logo, uv é adicionada ao corte corrente se, e somente se, $c'(uv) \geq c'(E' \cap E(T''_v))$, onde $c'(E' \cap E(T''_v)) := \sum_{e \in E' \cap E(T''_v)} c'(e)$. Nesse caso, o corte de arestas ao fim desse passo é $(E' \setminus E(T''_v)) \cup \{uv\}$. Claramente, todo o procedimento descrito para a árvore T'' pode ser feito em tempo $\mathcal{O}(n)$.

Afirmção 10.1. *O algoritmo de separação é correto.*

Prova. Para provar que o nosso algoritmo de separação é correto, mostraremos que o corte de arestas E' calculado por ele tem valor máximo. Considere um corte de arestas máximo E^* de T'' com a menor quantidade possível de arestas e tal que nenhum par de arestas nesse corte pertence a um mesmo caminho entre a raiz e alguma folha de T'' .

Para cada aresta $e \in E(T'')$, dizemos que e é uma aresta *usada* se essa aresta faz parte do corte de arestas corrente em algum passo do algoritmo. Note que se $E^* \subset E'$, então duas arestas de E' estão em um mesmo caminho entre a raiz e alguma folha. Uma contradição à forma como o corte E' é construído. Logo, considere uma aresta $uy \in E^* \setminus E'$ e suponha que a profundidade de u seja maior que a de y . Se uy não é usada pelo algoritmo de separação,

então no passo em que o vértice u é examinado, o algoritmo decide não adicionar uy ao corte corrente. Logo, $c'(uy) < c'(E'' \cap E(T''_u))$, onde E'' é o corte gerado pelo nosso algoritmo logo antes de o vértice u ser visitado. Dessa forma, o conjunto $[S', \bar{S}'] := (E^* \setminus \{uy\}) \cup (E'' \cap E(T''_u))$ é um corte tal que $v \in S'$ e $T''[S']$ é conexo. Uma contradição, pois $c'([S', \bar{S}']) > c'(E^*)$. Portanto, toda aresta em $E^* \setminus E'$ é usada pelo algoritmo de separação. Dessa forma, temos que para cada $uy \in E^* \setminus E'$, onde u tem profundidade maior que y , existe uma aresta $zw \in E'$ tal que $E(P_{ur}) \cap E' = \{zw\}$. Suponha que z tem profundidade maior que w e observe que $zw \notin E^*$. Logo, cada aresta em $E^* \cap E(T''_z)$ foi usada em algum passo pelo algoritmo de separação e substituída por zw no corte final. Portanto, $c'(zw) \geq c'(E^* \cap E(T''_z))$. Uma vez que E^* é um corte ótimo, obtemos a igualdade na última desigualdade, caso contrário poderíamos proceder de maneira análoga à anterior para construir uma solução com valor maior do que $c'(E^*)$. Pela minimalidade de E^* , temos que $E^* \cap E(T''_z) = \{uy\}$. Portanto, podemos concluir que $c'(E') = c'(E^*)$. \square

Para armazenar o corte de valor máximo na árvore T enraizada em cada um de seus vértices, podemos utilizar uma estrutura de dados que consiste em um vetor D de tamanho n indexado pelo conjunto de vértices V e tal que, para cada $v \in V$, $D[v]$ é um ponteiro para um vetor D^v com $d(v)$ posições e indexado pelos vizinhos de v em T . Considere dois vértices $v \in V$ e $u \in N(v)$. Se a aresta uv está no corte máximo na árvore T^v , então $D^v[u]$ é o peso dessa aresta, i.e. $c'(uv)$. Caso contrário, a posição $D^v[u]$ contém um marcador indicando que a aresta uv não está no corte máximo e, portanto, o corte de valor máximo em T^v contém o corte máximo na sub-árvore T^u . Assim, acessamos o vetor D^u apontado por $D[u]$ e aplicamos o mesmo procedimento para cada elemento de D^u , a menos de $D^u[v]$, a fim de enumerar as arestas do corte máximo em T^u .

Seja T' a árvore T enraizada em um vértice r . Podemos descrever o valor máximo do corte encontrado em T' pelo nosso algoritmo por meio de uma função recursiva $M : V \rightarrow \mathbb{Q}$ da seguinte forma:

$$M(u) = \begin{cases} 0, & \text{se } u \text{ é folha de } T', \\ \sum_{v \in F(u)} \max\{c'(uv); M(v)\}, & \text{caso contrário,} \end{cases}$$

onde $F(u)$ é o conjunto dos vértices filhos de u em T' . Note que $M(u)$ é o valor do maior corte na sub-árvore T'_u tal que quaisquer duas arestas distintas do corte não pertencem a um mesmo caminho entre u e uma folha de T'_u . Ademais, observe que a computação da função M pode ser feita em tempo $\mathcal{O}(n)$.

Definimos duas funções $f_1, f_2 : V \rightarrow V \cup \{\emptyset\}$ tais que:

$$f_1(u) = \begin{cases} \inf \left(\arg \max_{v \in V(T'_u) \setminus \{u\}} x_{v,c} \right), & \text{se } |F(u)| \geq 1, \\ \emptyset, & \text{caso contrário,} \end{cases}$$

$$f_2(u) = \begin{cases} \inf \left(\arg \max_{v \in V(T'_u) \setminus (\{u\} \cup V(T'_{u'}))} x_{v,c} \right), & \text{se } |F(u)| \geq 2, \\ \emptyset, & \text{caso contrário,} \end{cases}$$

onde u' é o filho de u em T' tal que $f_1(u) \in V(T'_{u'})$. Para cada $v \in V$, $f_1(v)$ e $f_2(v)$ podem ser determinados no momento em que calculamos o valor $M(v)$. Seja $g_1 : V \rightarrow V$ uma função tal que $g_1(v) = \inf \left(\arg \max_{z \in V \setminus \{v\}} x_{z,c} \right)$. Seja ainda $g_2 : V \rightarrow V \cup \{\emptyset\}$ uma função da seguinte forma:

$$g_2(v) = \begin{cases} \inf \left(\arg \max_{z \in V \setminus (\{v\} \cup V(T^v_v))} x_{z,c} \right), & \text{se } |N(v)| \geq 2, \\ \emptyset, & \text{caso contrário,} \end{cases}$$

onde v' é o filho de v em T^v tal que $g_1(v) \in V(T^v_{v'})$. Observe que $g_1(r) = f_1(r)$ e $g_2(r) = f_2(r)$.

Considere um filho u de r em T' e denote por T^u a árvore T enraizada em u . Seja $\hat{M} : V \rightarrow \mathbb{Q}$ a função dos cortes máximos na árvore T^u . É fácil ver que $\hat{M}(v) = M(v)$ para todo $v \in V \setminus \{r, u\}$. Assim, temos que $\hat{M}(r) = M(r) - \max\{c'(ru); M(u)\}$. Além disso, temos que $\hat{M}(u) = M(u) + \max\{x_{r^*,c} - x_{r,c}; \hat{M}(r)\}$, onde r^* é um vértice em $V(T^u_r) \setminus \{r\}$ tal que $x_{r^*,c} = \max\{x_{v,c} \mid v \in V(T^u_r) \setminus \{r\}\}$. Note que $r^* = g_1(r)$, se $g_1(r) \notin V(T'_u)$, e $r^* = g_2(r)$, caso contrário. Além disso, $g_1(u)$ e $g_2(u)$ são distintos e podem ser determinados em tempo constante a partir do conjunto $\{f_1(u), f_2(u), g_1(r), g_2(r), r\}$. Portanto, a função \hat{M} pode ser computada em tempo constante a partir de M .

A partir de r , podemos efetuar um percurso em largura na árvore T' e aplicar o procedimento anterior para cada vértice em V . Logo, dada uma cor $c \in \mathcal{C}$ fixa, o cálculo do valor máximo do lado esquerdo da desigualdade 4.8 é realizado em tempo $\mathcal{O}(n)$. Portanto, podemos usar a estrutura de dados descrita anteriormente para armazenar os cortes ótimos

e separar as desigualdades de convexidade-estrela em tempo $\mathcal{O}(kn)$. \square

5.2 Desigualdades de convexidade-árvore

Primeiramente, apresentamos uma adaptação direta do algoritmo de separação descrito por Lima [Lim11]. Em seguida, mostramos outro algoritmo de separação para as desigualdades de convexidade-caminho mais eficiente que esse. Finalmente, apresentamos um algoritmo de separação das desigualdades de convexidade-árvore cujo tempo de execução é $\mathcal{O}(kn^2)$ e que utiliza o algoritmo de separação para caminhos como sub-rotina.

Teorema 11 ([Lim11]). *As desigualdades de convexidade-caminho 4.9 podem ser separadas em $\mathcal{O}(kn^3)$.*

Prova. Considere uma cor $c \in \mathcal{C}$ e um vértice $w \in V$. Mostraremos como encontrar um conjunto $\{u_1, v_1, u_2, v_2, \dots, u_t, v_t, u_{t+1} = w\} \subseteq V(P_{uw})$ que maximiza o lado esquerdo da desigualdade 4.9, onde $u \in V$, $t \in \mathbb{Z}_{>}$ e $P_{uw} = z_1 z_2 \dots z_p$. Para cada $j \in \{1, 2, \dots, p\}$, a função recursiva $M : V(P_{uw}) \rightarrow \mathbb{Q}$ é definida como:

$$M(z_j) = \begin{cases} x_{z_p, c}, & \text{se } j = p, \\ -\infty, & \text{se } j = p - 1, \\ x_{z_j, c} + \max_{i=j+2, \dots, p} \left\{ M(z_i) - \min_{\ell=j+1, \dots, i-1} x_{z_\ell, c} \right\}, & \text{caso contrário.} \end{cases}$$

Para cada $z \in V(P_{uw})$, é fácil ver que $M(z) = \sum_{i=1}^{t'} x_{\dot{u}_i, c} - \sum_{i=1}^{t'} x_{\dot{v}_i, c} + x_{w, c}$, onde $\{z = \dot{u}_1, \dot{v}_1, \dot{u}_2, \dot{v}_2, \dots, \dot{u}_{t'}, \dot{v}_{t'}, \dot{u}_{t'+1} = w\} \subseteq V(P_{zw})$ e $t' \in \mathbb{Z}_{>}$. Por indução no comprimento dos sub-caminhos de P_{uw} que contêm w , podemos mostrar que, para cada $z \in V(P_{uw})$, $M(z)$ é o valor máximo alcançado no lado esquerdo da desigualdade 4.9 por qualquer desigualdade de convexidade-caminho gerada a partir da cor c e dos vértices z, w fixos.

Note que o cálculo da função M pode ser feito em tempo $\mathcal{O}(p^2)$, onde p é o comprimento do caminho P_{uw} . Dado um vértice $u' \notin V(P_{uw})$, se $V(P_{u'w}) \cap V(P_{uw}) \neq \emptyset$, então podemos reusar os valores de M computados para os vértices em P_{uw} e estender a função M para os vértices no caminho $P_{u'w}$ computando o valor da função apenas para vértices em

$V(P_{u'w}) \setminus V(P_{uw})$. Portanto, a função M pode ser computada em tempo $\mathcal{O}(n^2)$ para uma cor e um vértice fixos. Dessa forma, podemos separar as desigualdades 4.9 em tempo $\mathcal{O}(kn^3)$. \square

Teorema 12. *As desigualdades de convexidade-caminho 4.9 podem ser separadas em $\mathcal{O}(kn)$.*

Prova. Considere uma árvore T' enraizada em um vértice $r \in V$ e uma cor $c \in \mathcal{C}$. Para cada $v \in V$, seja $F(v)$ o conjunto dos vértices que são filhos de v em T' . Sejam $M^+ : V \rightarrow \mathbb{Q}$ e $M^- : V \rightarrow \mathbb{Q}$ duas funções recursivas definidas da seguinte maneira:

$$M^+(u) = \begin{cases} x_{u,c}, & \text{se } u \text{ é folha de } T', \\ \max \left\{ \max_{v \in F(u)} M^+(v); x_{u,c} + \max_{v \in F(u)} M^-(v) \right\}, & \text{caso contrário,} \end{cases} \quad (5.1)$$

$$M^-(u) = \begin{cases} 0, & \text{se } u \text{ é folha de } T', \\ \max \left\{ \max_{v \in F(u)} M^-(v); -x_{u,c} + \max_{v \in F(u)} M^+(v) \right\}, & \text{caso contrário.} \end{cases} \quad (5.2)$$

Intuitivamente, $M^+(u)$ é o maior valor atingido pelo lado esquerdo da desigualdade 4.9 obtida a partir de um caminho entre uma folha de T'_u e o vértice u . O significado do valor $M^-(u)$ é similar ao anterior exceto pelo fato de que a soma tem um número par de termos e o vértice de menor profundidade nesse caminho (cujo coeficiente é não-nulo) é associado a um coeficiente negativo. Sejam $f_1^+, f_2^+ : V \rightarrow V \cup \{\emptyset\}$ funções definidas da seguinte maneira:

$$f_1^+(u) = \begin{cases} \inf \left(\arg \max_{v \in F(u)} (\max\{M^+(v); x_{u,c} + M^-(v)\}) \right), & \text{se } |F(u)| \geq 1, \\ \emptyset, & \text{caso contrário,} \end{cases}$$

$$f_2^+(u) = \begin{cases} \inf \left(\arg \max_{v \in F(u) \setminus \{f_1^+(u)\}} (\max\{M^+(v); x_{u,c} + M^-(v)\}) \right), & \text{se } |F(u)| \geq 2, \\ \emptyset, & \text{caso contrário.} \end{cases}$$

De maneira análoga, definimos as funções f_1^- e f_2^- .

Por indução na profundidade dos vértices em T' , é fácil mostrar que as funções M^+ e M^- correspondem aos valores máximos para cada vértice em V na árvore T' . Dessa forma,

o maior valor atingido pelo lado esquerdo da desigualdade 4.9 obtido na árvore T' é

$$\max \left\{ M^+(r) + \max_{v \in F(r) \setminus \{f_1^+(r)\}} M^-(v); M^-(r) + \max_{v \in F(r) \setminus \{f_1^-(r)\}} M^+(v) \right\},$$

Esse é o maior valor atingido pelo lado esquerdo da desigualdade 4.9 que pode ser obtido a partir de qualquer caminho em T que contém r . Percorrendo a árvore T' a partir dos vértices de maior profundidade e em direção à raiz, é possível computar as funções M^+ e M^- em tempo $\mathcal{O}(n)$. Note que as funções f_1^+ , f_2^+ , f_1^- e f_2^- podem ser determinadas em tempo constante a partir da computação de M^+ e M^- .

Para cada $v \in V$, M_v^+ e M_v^- correspondem, respectivamente, às funções 5.1 e 5.2 referentes a árvore T^v . Definimos $g_1^+ : V \rightarrow V$ e $g_2^+ : V \rightarrow V \cup \{\emptyset\}$ da seguinte maneira:

$$g_1^+(v) = \inf \left(\arg \max_{z \in N(v)} (\max\{M_v^+(z); x_{v,c} + M_v^-(z)\}) \right),$$

$$g_2^+(v) = \begin{cases} \inf \left(\arg \max_{z \in N(v) \setminus \{g_1^+(v)\}} (\max\{M_v^+(z); x_{v,c} + M_v^-(z)\}) \right), & \text{se } |N(v)| \geq 2, \\ \emptyset, & \text{caso contrário.} \end{cases}$$

As funções g_1^- e g_2^- são definidas de forma análoga. Note que $g_1^+(r) = f_1^+(r)$ e $g_2^+(r) = f_2^+(r)$. Considere um filho u de r em T' , M_u^+ e M_u^- as funções associadas à árvore T^u , onde T^u é a árvore T enraizada no vértice u . Claramente, temos que $M_u^+(v) = M^+(v)$ e $M_u^-(v) = M^-(v)$ para todo $v \in V \setminus \{u, r\}$. Se $g_1^+(r) \neq u$, então $M_u^+(r) = M^+(r)$. Caso contrário, $M_u^+(r)$ é o maior valor que pode ser obtido entre os filhos de r em T^u , ou seja, $M_u^+(r) = \max\{\max_{v \in N(r) \setminus \{u\}} M^+(v); x_{r,c} + \max_{v \in N(r) \setminus \{u\}} M^-(v)\}$. Note que esse valor pode ser facilmente determinado a partir do valor $g_2^+(r)$. Uma análise similar pode ser feita com relação ao cálculo do valor $M_u^-(r)$. Logo, temos que $M_u^+(u) = \max\{M^+(u); M_u^+(r); x_{u,c} + M_u^-(r)\}$ e $M_u^-(u) = \max\{M^-(u); M_u^-(r); -x_{u,c} + M_u^+(r)\}$. Dessa forma, as funções M_u^+ e M_u^- podem ser computadas em tempo constante usando as funções M^+ e M^- . Além disso, os valores $g_1^+(u)$ e $g_2^+(u)$ são distintos e pertencem ao conjunto $\{f_1^+(u), f_2^+(u), r\}$. Logo, $g_1^+(u)$ e $g_2^+(u)$ são determinados em tempo constante a partir dos elementos desse conjunto. Uma análise similar pode ser feita com relação a $g_1^-(u)$ e $g_2^-(u)$.

A partir de r , podemos efetuar um percurso em largura na árvore T' e aplicar o procedimento anterior para cada vértice em V . Logo, dada uma cor $c \in \mathcal{C}$ fixa, podemos calcular o valor máximo do lado esquerdo da desigualdade 4.9 em $\mathcal{O}(n)$. Portanto, as desigualdades de convexidade-caminho podem ser separadas em tempo $\mathcal{O}(kn)$. \square

Teorema 13. *As desigualdades de convexidade-árvore 4.10 podem ser separadas em $\mathcal{O}(kn)$.*

Prova. Considere uma árvore T' enraizada em um vértice $r \in V$ e uma cor $c \in \mathcal{C}$. Para cada $u \in V$, seja $F(u)$, de forma análoga à definição no teorema anterior, o conjunto dos vértices que são filhos de u em T' . Seja $W \subseteq V$ o conjunto formado pela união dos vértices que possuem mais de um filho em T' , das folhas e da raiz de T' . Seja ainda W' a união do conjunto W e dos vértices em T' cujo pai pertence a W , i.e. $W' := W \cup \bigcup_{u \in W} F(u)$.

A seguir, definiremos formalmente uma função recursiva M sobre o conjunto W' . Intuitivamente, $M(u)$ é o valor máximo que o lado esquerdo da desigualdade 4.10 pode atingir em qualquer sub-árvore de T'_u . Para cada $u \in W' \setminus W$, $\tilde{M}(P_{u\hat{w}})$ é o valor máximo do lado esquerdo da desigualdade 4.9 obtido no caminho $P_{u\hat{w}}$, onde w é o vértice em W de menor profundidade na sub-árvore T'_u e \hat{w} é o vértice obtido pela contração da subárvore T'_w em T' (veja Figura 5.1). Após a contração, considere $x_{\hat{w},c} := M(w)$ para o cálculo de $\tilde{M}(P_{u\hat{w}})$. Para cada $u \in W$, seja $F_{>}(u) = \{v \in F(u) \mid M(v) > x_{u,c}\}$. Formalmente, a função $M : W' \rightarrow \mathbb{Q}$ é definida como:

$$M(u) = \begin{cases} \tilde{M}(P_{u\hat{w}}), & \text{se } u \in W' \setminus W, \\ (1 - |F_{>}(u)|)x_{u,c} + \sum_{v \in F_{>}(u)} M(v), & \text{caso contrário.} \end{cases}$$

Afirmção 13.1. *Para cada $u \in W'$, $M(u)$ é máximo.*

Prova. Por indução na profundidade dos vértices em T' , mostraremos que o valor $M(u)$ é máximo para cada $u \in W'$. Se u é folha de T' , então $M(u) = x_{u,c}$ e o resultado segue trivialmente. Seja $u \in W'$ tal que u não é folha de T' e suponha que o valor $M(v)$ é máximo para todo vértice $v \in V$ com profundidade maior de que a de u em T' . Se $u \in W' \setminus W$, então, por hipótese indutiva, $M(w)$ é o valor máximo que o lado esquerdo da desigualdade 4.10 pode

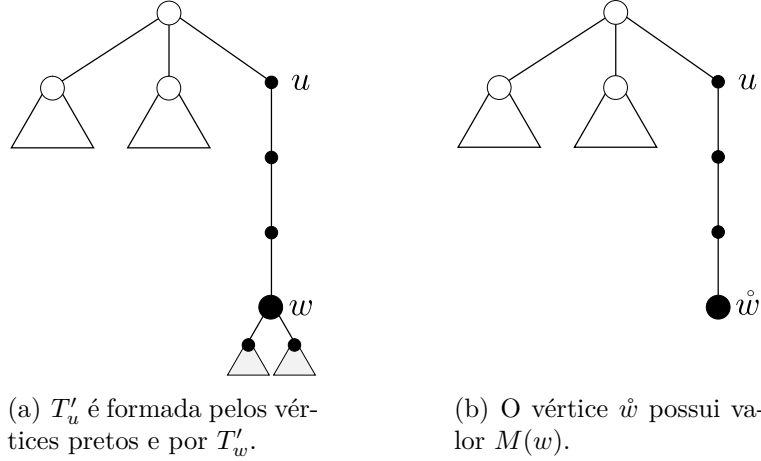


Figura 5.1: Contração da sub-árvore T'_w no vértice \hat{w} .

atingir em qualquer subárvore de T'_w , onde w é o vértice em W de menor profundidade na sub-árvore T'_u . Fazendo $M(w)$ ser o peso do vértice contraído \hat{w} , pelo Teorema 12, podemos concluir que o valor $\tilde{M}(P_{u\hat{w}})$ é máximo. Logo, $M(u)$ também é máximo.

Antes de analisar o último caso, considere a seguinte definição. Para cada $v \in W$, seja $\text{coef}_v : V(T'_v) \rightarrow \mathbb{Z}$ uma função que corresponde aos coeficientes da desigualdade 4.10 associados aos vértices da sub-árvore T'_v de tal forma que $M(v) = \sum_{z \in V(T'_v)} \text{coef}_v(z)x_{z,c}$. Se $u \in W$, então denote por $M^*(u)$ o maior valor que pode ser alcançado pelo lado esquerdo da desigualdade 4.10 e por $\text{coef}_u^* : V(T'_u) \rightarrow \mathbb{Z}$ uma atribuição de coeficientes da desigualdade 4.10 que corresponde a $M^*(u)$, i.e. $M^*(u) = \sum_{v \in V(T'_u)} \text{coef}_u^*(v)x_{v,c}$. Além disso, tome coef_u^* uma função maximal com respeito ao número de coeficientes nulos e suponha, por absurdo, que $M^*(u) > M(u)$. Se $\text{coef}_u^*(u) > \text{coef}_u(u)$, então existe $v \in F_{>}(u)$ tal que $\text{coef}_u^*(z) = 0 \forall z \in V(T'_v)$. Dessa forma, seja $\text{coef}'_u : V(T'_u) \rightarrow \mathbb{Z}$ uma função dos coeficientes definida da seguinte maneira:

$$\text{coef}'_u(z) = \begin{cases} \text{coef}_u^*(z) - 1, & \text{se } z = u, \\ \text{coef}_u(z), & \text{se } z \in V(T'_v), \\ \text{coef}_u^*(z), & \text{caso contrário.} \end{cases}$$

Uma vez que $M(v) > x_{u,c}$, temos que $\sum_{z \in V(T'_u)} \text{coef}'_u(z)x_{z,c} > M^*(u)$, uma contradição. Se $\text{coef}_u^*(u) < \text{coef}_u(u)$, então existe $v \in F(u) \setminus F_{>}(u)$ tal que $\text{coef}_u^*(z) \neq 0$ para algum $z \in V(T'_v)$, caso contrário poderíamos redefinir $\text{coef}_u^*(u) := \text{coef}_u(u)$ e obter $M^*(u) \leq M(u)$.

Note que $\text{coef}_u(z) = 0 \forall z \in V(T'_v)$. Assim, definimos coef'_u da seguinte forma:

$$\text{coef}'_u(z) = \begin{cases} \text{coef}_u^*(z) + 1, & \text{se } z = u, \\ 0, & \text{se } z \in V(T'_v), \\ \text{coef}_u^*(z), & \text{caso contrário.} \end{cases}$$

Por hipótese indutiva, temos que $\sum_{z \in V(T'_v)} \text{coef}_u^*(z)x_{z,c} \leq M(v)$. Além disso, sabemos que $M(v) \leq x_{u,c}$. Logo, podemos assumir que $\sum_{z \in V(T'_v)} \text{coef}_u^*(z)x_{z,c} = x_{u,c}$, caso contrário chegaríamos a uma contradição de maneira similar ao caso anterior. Portanto, $\sum_{z \in V(T'_u)} \text{coef}'_u(z)x_{z,c} = M^*(u)$ e coef'_u possui mais coeficientes nulos que coef_u^* , contrariando a maximalidade desse. Por hipótese indutiva, temos que $M(v)$ é máximo para todo $v \in F(u)$. Logo, se $\text{coef}_u^*(u) = \text{coef}_u(u)$, então $M^*(u) \leq M(u)$, uma contradição. Portanto, o valor $M(u)$ é máximo. \square

Para uma cor fixa, observe que a computação da função M associada à árvore T' pode ser feita em tempo $\mathcal{O}(n)$ usando o algoritmo para caminhos descrito no Teorema 12 como subrotina. Considere um filho u de r em T' , a partir da função M podemos computar a função M_u associada à árvore T^u , onde T^u é a árvore T enraizada no vértice u . Claramente, temos que $M_u(v) = M(v)$ para todo $v \in V \setminus \{u, r\}$. Se $M(u) \leq x_{r,c}$, então $M_u(r) = M(r)$. Senão, fazemos $M_u(r) = M(r) - M(u) + x_{r,c}$. Dado o valor $M_u(r)$, definimos $M_u(u)$ da seguinte maneira. Se $M_u(r) > x_{u,c}$, então $M_u(u) = M(u) + M_u(r) - x_{u,c}$. Caso contrário, fazemos $M_u(u) = M(u)$. Dessa maneira, a função M_u pode ser computada em tempo constante a partir de M .

A partir de r , podemos efetuar um percurso em largura na árvore T' e aplicar o procedimento anterior para cada vértice em V . Para uma cor $c \in \mathcal{C}$ fixa, podemos calcular o valor máximo do lado esquerdo da desigualdade 4.10 em $\mathcal{O}(n)$. Portanto, as desigualdades de convexidade-árvore podem ser separadas em tempo $\mathcal{O}(kn)$. \square

Capítulo 6

Experimentos computacionais

Neste capítulo, descrevemos um algoritmo de *branch-and-cut* que implementamos para o problema RCA e apresentamos os seus resultados computacionais. Na Seção 6.1, mostramos como foram obtidas as instâncias utilizadas nos experimentos. Na Seção 6.2, apresentamos os detalhes da implementação do algoritmo. Por fim, os resultados computacionais são mostrados na Seção 6.3.

6.1 Instâncias obtidas de árvores filogenéticas reais

Uma *árvore filogenética* é uma árvore que representa o curso de evolução de um certo conjunto de espécies. As folhas dessa árvore representam espécies existentes. Os vértices internos correspondem a espécies hipoteticamente extintas. Uma *característica* é um atributo compartilhado por todas as espécies do conjunto em questão. Cada espécie pode apresentar um *estado* diferente para uma determinada característica. Mais formalmente, se V é o conjunto de espécies, então uma característica é uma função $C : V \rightarrow \mathcal{C}$, onde \mathcal{C} é o conjunto de todos os possíveis estados dessa característica. Uma suposição natural em biologia é que os estados de uma característica evoluem sem transições *reversas* ou *convergentes*. Em uma transição reversa, uma espécie apresenta o mesmo estado de algum ancestral antigo ainda que o seu pai não apresente esse estado. Uma transição convergente ocorre quando duas espécies apresentam o mesmo estado de uma dada característica e o seu ancestral comum com maior profundidade apresenta um estado diferente deles. Note que, sem esses dois tipos de

transições, os estados de uma característica definem uma coloração total convexa da árvore considerada. Em aplicações biológicas reais, as instâncias provavelmente contêm informações erradas ou ausência de informações sobre o estado em uma espécie; portanto, a computação da distância de recoloração pode conduzir a um procedimento de detecção e correção de erros. Essa aplicação deu origem do problema de recoloração convexa de árvores, proposto por Moran e Snir [MS05].

As árvores filogenéticas utilizadas nos nossos experimentos foram obtidas no repositório <http://treebase.org/treebase-web/>. Para interpretar os dados biológicos e gerar uma coloração que corresponda a uma característica real, são necessários conhecimentos específicos nessa área. Portanto, optamos por gerar uma coloração inicial de forma aleatória em vez de usar coloração baseada em uma característica real. A seguir, apresentamos um procedimento para geração de uma coloração inicial de uma árvore.

1. Definimos dois parâmetros p_c e p_n tais que $0 \leq p_c, p_n \leq 1$;
2. Atribuimos a cor 1 à raiz;
3. Seja v um vértice já colorido. Para cada filho u de v , atribuimos ao vértice u a mesma cor de v com probabilidade $1 - p_c$ ou atribuimos a próxima cor que ainda não foi usada (com probabilidade p_c);
4. Para cada filho u de v , aplicamos o Passo 3 recursivamente para colorir os filhos de u ;
5. Após o Passo 4, temos uma coloração total convexa da árvore. Embora não tenhamos nenhum controle sobre o tamanho exato de \mathcal{C} , intuitivamente, $|\mathcal{C}|$ cresce à medida que aumentamos o valor de p_c . Neste passo, introduzimos uma “perturbação” na coloração da seguinte maneira: para cada vértice v , decidimos com probabilidade $1 - p_n$ se v mantém ou não a sua cor. Se decidimos mudar a cor de v , então escolhemos com probabilidade uniforme uma cor do conjunto \mathcal{C} e associamos essa cor ao vértice v .
6. Seja $C : V \rightarrow \mathcal{C}$ a coloração obtida após o Passo 5. Para cada $v \in V$ e cada $c \in \mathcal{C}$, definimos $w_{v,c} = 1$, se $C(v) = c$, e $w_{v,c} = 0$, caso contrário.

6.2 Detalhes da implementação

Utilizamos *Java 1.6* como linguagem de programação e *CPLEX 12.5* como pacote para otimização. Todos os recursos de *presolve* e geração automática de cortes foram desativados. Os outros parâmetros do *CPLEX* foram configurados com os seus respectivos valores padrão. Na prática, observamos que o desempenho do nosso método é bem melhor quando iniciamos o procedimento com uma relaxação da formulação inicial. Nesse sentido, usamos a seguinte relaxação para iniciar o nosso algoritmo de *branch-and-cut*:

$$\begin{aligned}
 (\mathcal{REL}) \quad & \max \sum_{v \in V} \sum_{c \in \mathcal{C}} w_{v,c} x_{v,c} \\
 & \text{s.a.} \sum_{c \in \mathcal{C}} x_{v,c} \leq 1 \quad \forall v \in V, \\
 & x_{v,c} \geq 0 \quad \forall v \in V, \forall c \in \mathcal{C}.
 \end{aligned}$$

As restrições de convexidade são tratadas de maneira dinâmica com a adição de cortes gerados pelas desigualdades de convexidade-estrela e convexidade-árvore que rejeitam as soluções fracionárias e as soluções inteiras que satisfazem as restrições do modelo \mathcal{REL} , mas que não representam colorações convexas. Tentamos adicionar apenas os cortes das desigualdades de convexidade-estrela, mas a qualidade da solução dada pela relaxação e o tempo de execução foram consideravelmente piores do que quando adicionamos esses cortes juntamente com os cortes gerados pelas desigualdades de convexidade-árvore. O mesmo comportamento foi verificado quando adicionamos somente os cortes das desigualdades de convexidade-árvore. Dessa forma, o melhor desempenho foi alcançado quando adicionamos os cortes de ambas as classes de desigualdades. Além disso, observamos um ganho de desempenho quando adicionamos ao modelo os cortes mais violados das desigualdades de convexidade-estrela e convexidade-árvore para cada cor ao invés de adicionar apenas o corte mais violado dentre todas as cores. Por fim, notamos que o *CPLEX* começa a fazer ramificações de forma prematura quando ele possui o controle do procedimento de *branch-and-cut* desde o início. Portanto, aplicamos o método de plano-de-cortes para obter uma melhor relaxação antes de passar para o *CPLEX* o controle do procedimento de *branch-and-cut*.

Uma outra questão importante, que pode afetar dramaticamente o desempenho, é o crité-

rio de parada para a geração de cortes. Nesse sentido, desejamos evitar que o algoritmo itere muitas vezes sem obter uma mudança significativa no valor da função objetivo. Após vários experimentos, optamos por parar o procedimento de plano-de-cortes quando o decréscimo do valor da solução é no máximo de 0.01 por 10 iterações consecutivas.

Note que as árvores filogenéticas possuem uma raiz especificada na entrada. No caso geral, podemos escolher arbitrariamente um vértice da árvore para ser a raiz. Assim, definimos $p : V \rightarrow V$ uma função tal que $p(v)$ é v , se v é raiz; e $p(v)$ é o pai do vértice v na árvore, caso contrário. Implementamos uma heurística que gera uma solução inteira a partir do procedimento de arredondamento de uma solução fracionária x da forma descrita a seguir. Denote por $C^H(v)$ a cor atribuída pela heurística ao pai de um vértice $v \in V$. Inicialmente, sejam $V' := \{r\}$ e $\mathcal{C}' := \mathcal{C}$, onde r é a raiz da árvore. Removemos de V' o vértice r e computamos a cor c tal que $c \in \arg \max_{d \in \mathcal{C}'} x_{r,d}$. Fixamos o valor de $x_{r,c}$ em 1, adicionamos todos os filhos de r ao conjunto V' e fazemos $\mathcal{C}' := \mathcal{C} \setminus \{c\}$. Enquanto $|V'| > 0$, removemos de V' o vértice v tal que $(v, c) \in \arg \max_{z \in V', d \in \mathcal{C}' \cup \{C^H(p(z))\}} x_{z,d}$. Em seguida, fixamos o valor de $x_{v,c}$ em 1 e adicionamos todos os filhos de v em V' . Se $c \neq C^H(p(v))$, então removemos a cor c de \mathcal{C}' . Aplicamos essa heurística em cada nó da árvore de *branch-and-bound*.

6.3 Resultados computacionais

Os experimentos computacionais cujos resultados são relatados nesta seção foram feitos em uma máquina com 65GB de memória RAM, executando em uma única *thread* de um processador com 1.6GHz. Estipulamos um tempo limite de 10 minutos para o algoritmo de plano-de-cortes resolver a relaxação \mathcal{REL} e um tempo adicional de 30 minutos para o *CPLEX* aplicar o *branch-and-cut* e resolver o problema com as variáveis inteiras. Para comparação, implementamos o algoritmo de programação dinâmica proposto por Moran e Snir em [MS08] para o problema RCA. Dado que esse algoritmo teve um péssimo desempenho mesmo para as menores instâncias, nós não relatamos os seus resultados computacionais.

Nas Tabelas 6.1 e 6.2, apresentamos os resultados dos experimentos computacionais obtidos com o nosso algoritmo de *branch-and-cut* no conjunto de instâncias descrito na Seção 6.1. Na coluna “TB-ID”, mostramos o identificador da instância que pode ser usado para en-

contrar a árvore filogenética correspondente em <http://treebase.org/treebase-web/>. O grau médio das árvores nas nossas instâncias é entre 3 e 4 e o grau máximo delas está entre 3 e 140. Observamos que a maior parte das árvores tem grau máximo entre 3 e 25. Sejam LP^* o valor de uma solução ótima da relaxação obtida logo após a aplicação do algoritmo de plano-de-cortes, IP^* o valor de uma solução ótima para o modelo \mathcal{T} e HR o valor da solução inteira encontrada pela heurística de arredondamento descrita na Seção 6.2 a partir da solução fracionária correspondente ao valor LP^* . Na coluna “Gap/LP”, mostramos o *gap* entre os valores LP^* e IP^* . Esse valor é dado por $\frac{(LP^* - IP^*)}{IP^*} \times 100$. Na coluna “Gap/HR”, apresentamos o *gap* entre HR e IP^* dado por $\frac{(IP^* - HR)}{IP^*} \times 100$. Na coluna “Gap/FN”, mostramos o *gap* final, ou seja, o *gap* entre o melhor limitante inferior e o melhor limitante superior obtido ao final do algoritmo. Os tempos (em segundos) gastos para computar os valores LP^* e IP^* são mostrados nas colunas “Tempo/LP” e “Tempo/IP”, respectivamente. A sigla *TLE* é acrônimo de *Tempo Limite Excedido*. As colunas “Cortes no LP” e “Cortes no IP” contêm o número total de cortes de convexidade-árvore e convexidade-estrela que foram adicionados para calcular os valores LP^* e IP^* , respectivamente. Dado que em todas as instâncias a nossa heurística executou em menos de um segundo e o número de nós na árvore de *branch-and-bound* criada pelo *CPLEX* foi muito pequeno, nós não relatamos esses valores.

As Tabelas 6.1 e 6.2 mostram que o nosso algoritmo de *branch-and-cut* resolveu otimalmente uma grande parte das instâncias. Em quase todas as instâncias o *gap* final foi bem pequeno. Observamos que, em geral, quanto maior forem $|\mathcal{C}|$ e p_n , mais difícil é a instância. O número total de cortes de convexidade-árvore e convexidade-estrela adicionados foram praticamente o mesmo. Isso indica que ambas as desigualdades são igualmente úteis para eliminar soluções fracionárias. Além disso, os pequenos *gaps* entre o valor da solução da relaxação após o plano-de-cortes e o valor da solução ótima mostram que as nossas facetas definem bons cortes.

Tabela 6.1: *Resultados computacionais do branch-and-cut – Parte 1.*

TB-ID	V	p_n	p_c	C	Tempo		Gap			Cortes no LP		Cortes no IP	
					LP	IP	HR	LP	FN	Árv.	Est.	Árv.	Est.
Tr60729	213	25%	0.5%	3	0	0	0%	0.12%	0%	77	71	0	0
Tr60729	213	25%	5.0%	14	0	0	0%	0%	0%	30	30	0	0
Tr60729	213	25%	50.0%	103	2	4	0%	0%	0%	102	102	0	0
Tr60729	213	50%	0.5%	4	0	0	0%	0%	0%	72	72	0	0
Tr60729	213	50%	5.0%	14	1	0	0%	0%	0%	114	118	0	0
Tr60729	213	50%	50.0%	98	2	4	0%	0%	0%	192	198	0	0
Tr60729	213	75%	0.5%	3	1	0	0%	0%	0%	90	90	0	0
Tr60729	213	75%	5.0%	12	5	0	0%	0.33%	0%	509	466	0	0
Tr60729	213	75%	50.0%	110	17	48	2.86%	0.51%	0%	952	945	12	15
Tr60079	301	25%	0.5%	6	0	0	0%	0%	0%	43	43	0	0
Tr60079	301	25%	5.0%	20	1	0	0%	0%	0%	58	58	0	0
Tr60079	301	25%	50.0%	148	3	22	0%	0%	0%	106	106	0	0
Tr60079	301	50%	0.5%	3	0	0	0%	0%	0%	52	52	0	0
Tr60079	301	50%	5.0%	20	2	0	0%	0%	0%	185	181	0	0
Tr60079	301	50%	50.0%	153	17	13	0%	0.04%	0%	484	497	0	0
Tr60079	301	75%	0.5%	7	20	0	0%	0%	0%	469	457	0	0
Tr60079	301	75%	5.0%	21	36	0	0%	0.42%	0%	1011	878	0	0
Tr60079	301	75%	50.0%	165	19	154	0.62%	0.18%	0%	576	608	14	18
Tr6287	404	25%	0.5%	6	1	0	0%	0.15%	0%	36	36	0	0
Tr6287	404	25%	5.0%	29	1	1	0%	0%	0%	54	54	0	0
Tr6287	404	25%	50.0%	197	13	47	0%	0%	0%	224	225	0	0
Tr6287	404	50%	0.5%	4	2	0	0%	0%	0%	115	114	0	0
Tr6287	404	50%	5.0%	22	6	1	0%	0%	0%	284	263	0	0
Tr6287	404	50%	50.0%	206	29	32	0%	0.14%	0%	619	631	0	0
Tr6287	404	75%	0.5%	5	59	0	0%	0%	0%	578	565	0	0
Tr6287	404	75%	5.0%	22	84	5	2.48%	0.04%	0%	1241	1044	7	3
Tr6287	404	75%	50.0%	207	60	TLE	-	-	3%	1574	1621	107	96
Tr4755	567	25%	0.5%	4	1	0	0%	0%	0%	43	41	0	0
Tr4755	567	25%	5.0%	29	7	5	0%	0%	0%	78	77	1	1
Tr4755	567	25%	50.0%	277	70	TLE	-	-	0.45%	169	169	11	11
Tr4755	567	50%	0.5%	5	1	0	0%	0%	0%	54	54	0	0
Tr4755	567	50%	5.0%	28	11	3	0%	0.09%	0%	204	195	0	0
Tr4755	567	50%	50.0%	273	124	TLE	-	-	0.75%	456	446	13	13
Tr4755	567	75%	0.5%	5	16	0	0%	0%	0%	310	299	0	0
Tr4755	567	75%	5.0%	34	153	42	11.83%	0.21%	0%	1886	1556	39	35
Tr4755	567	75%	50.0%	278	193	TLE	-	-	0.88%	757	754	21	21
Tr2400	636	25%	0.5%	6	1	0	0%	0%	0%	87	86	0	0
Tr2400	636	25%	5.0%	35	2	4	0%	0%	0%	123	124	0	0
Tr2400	636	25%	50.0%	333	19	323	0%	0%	0%	217	221	0	0
Tr2400	636	50%	0.5%	7	3	0	0%	0%	0%	140	136	0	0
Tr2400	636	50%	5.0%	28	17	56	0.3%	0.11%	0%	720	703	22	22
Tr2400	636	50%	50.0%	317	105	411	0.53%	0.02%	0%	1190	1163	0	0
Tr2400	636	75%	0.5%	5	149	0	0%	0%	0%	665	659	0	0
Tr2400	636	75%	5.0%	28	92	2	0%	0.09%	0%	1620	1539	0	0
Tr2400	636	75%	50.0%	309	431	1088	0.33%	0.33%	0%	3818	3951	22	22

Tabela 6.2: Resultados computacionais do branch-and-cut – Parte 2.

TB-ID	V	p_n	p_c	C	Tempo		Gap			Cortes no LP		Cortes no IP	
					LP	IP	HR	LP	FN	Árv.	Est.	Árv.	Est.
Tr4756	710	25%	0.5%	10	5	0	0%	0.09%	0%	295	294	0	0
Tr4756	710	25%	5.0%	35	4	7	0%	0%	0%	98	98	0	0
Tr4756	710	25%	50.0%	361	17	518	0%	0%	0%	199	198	0	0
Tr4756	710	50%	0.5%	4	1	0	0%	0%	0%	88	88	0	0
Tr4756	710	50%	5.0%	41	15	11	0%	0%	0%	579	568	0	0
Tr4756	710	50%	50.0%	338	338	1663	0.93%	0.01%	0%	844	852	7	7
Tr4756	710	75%	0.5%	6	167	20	0.35%	0.03%	0%	856	827	42	35
Tr4756	710	75%	5.0%	36	210	284	1.44%	0.54%	0%	1625	1529	76	79
Tr4756	710	75%	50.0%	361	338	TLE	-	-	2.17%	2746	2769	26	23
Tr6038	813	25%	0.5%	9	3	0	0%	0.03%	0%	173	173	0	0
Tr6038	813	25%	5.0%	42	7	10	0%	0%	0%	114	114	0	0
Tr6038	813	25%	50.0%	430	38	2661	0.16%	0%	0%	239	238	0	0
Tr6038	813	50%	0.5%	7	7	0	0%	0%	0%	219	215	0	0
Tr6038	813	50%	5.0%	49	21	20	0%	0%	0%	456	454	0	0
Tr6038	813	50%	50.0%	393	243	TLE	-	-	0.29%	1294	1284	0	0
Tr6038	813	75%	0.5%	9	573	8	0.38%	0.19%	0%	1288	1246	4	4
Tr6038	813	75%	5.0%	45	TLE	1461	0%	0.78%	0%	2442	2293	383	303
Tr6038	813	75%	50.0%	425	TLE	TLE	-	-	4.46%	4000	4088	62	60
Tr53777	981	25%	0.5%	13	4	2	0%	0%	0%	170	170	1	1
Tr53777	981	25%	5.0%	65	8	50	0%	0%	0%	133	133	0	0
Tr53777	981	25%	50.0%	469	43	TLE	-	-	0%	289	289	0	0
Tr53777	981	50%	0.5%	9	36	1	0%	0.14%	0%	436	434	0	0
Tr53777	981	50%	5.0%	45	32	17	0%	0%	0%	651	661	0	0
Tr53777	981	50%	50.0%	476	229	TLE	-	-	0.57%	1153	1160	0	0
Tr53777	981	75%	0.5%	6	TLE	38	0.5%	0.06%	0%	849	844	19	18
Tr53777	981	75%	5.0%	50	589	TLE	-	-	0.89%	2459	2336	324	257
Tr53777	981	75%	50.0%	492	TLE	TLE	-	-	5.09%	3341	3376	0	0
Tr25470	1441	25%	0.5%	7	15	1	0%	0.08%	0%	240	240	0	0
Tr25470	1441	25%	5.0%	75	14	89	0%	0%	0%	239	239	0	0
Tr25470	1441	25%	50.0%	699	130	TLE	-	-	0%	501	504	0	0
Tr25470	1441	50%	0.5%	10	65	72	0%	0.15%	0%	606	596	105	101
Tr25470	1441	50%	5.0%	80	180	263	0.14%	0.01%	0%	1978	1977	5	5
Tr25470	1441	50%	50.0%	681	TLE	TLE	-	-	1.64%	2473	2482	0	0
Tr25470	1441	75%	0.5%	9	TLE	TLE	-	-	1.09%	1085	1084	753	746
Tr25470	1441	75%	5.0%	65	TLE	TLE	-	-	26.46%	3191	3167	853	820
Tr25470	1441	75%	50.0%	750	TLE	TLE	-	-	5.92%	2655	2668	0	0
Tr9414	1577	25%	0.5%	13	8	3	0%	0.03%	0%	239	238	0	0
Tr9414	1577	25%	5.0%	74	24	109	0%	0%	0%	244	244	1	1
Tr9414	1577	25%	50.0%	822	228	TLE	-	-	0.5%	587	584	0	0
Tr9414	1577	50%	0.5%	13	135	3	0%	0.07%	0%	831	819	0	0
Tr9414	1577	50%	5.0%	89	146	101	0%	0%	0%	1312	1309	0	0
Tr9414	1577	50%	50.0%	822	TLE	TLE	-	-	2.45%	2349	2381	0	0
Tr9414	1577	75%	0.5%	11	TLE	TLE	-	-	1.12%	1030	1027	594	588
Tr9414	1577	75%	5.0%	70	TLE	TLE	-	-	52.15%	2445	2428	690	668
Tr9414	1577	75%	50.0%	792	TLE	TLE	-	-	4.7%	2912	2934	0	0

Capítulo 7

Conclusão

Desde o seu surgimento no contexto de árvores filogenéticas [MS05], o problema de recoloração convexa tem sido bastante estudado, principalmente no caso restrito a árvores. Resultados acerca da complexidade computacional desse problema em algumas classes de grafos e vários algoritmos de aproximação foram obtidos nos últimos anos [BYFR08, CFR⁺07, KT08, KK09, LW11, MS05, MS07]. Em [Lim11], foi proposta a primeira abordagem poliédrica para o problema de recoloração convexa. Nesse trabalho, foram mostradas classes de desigualdades que induzem facetas de um poliedro associado ao problema RC restrito a caminhos.

Neste trabalho, provamos resultados sobre a complexidade do problema de recoloração convexa. Provamos que uma versão parametrizada desse problema é $\mathcal{W}[2]$ -difícil a partir de uma redução do problema 3-SAT parametrizado pelo número de variáveis verdadeiras. Além disso, provamos que não existe algoritmo de aproximação para o problema RC-BIN com fator de aproximação menor que n , a menos de $\mathcal{P} = \mathcal{NP}$. Esse resultado mostra que é improvável que exista algum algoritmo para o problema RC-BIN com fator de aproximação menor do que o fator de um algoritmo que simplesmente atribui uma mesma cor a todos os vértices.

Na parte poliédrica, apresentamos formulações lineares 0/1 para o problema de recoloração convexa tanto no caso geral quanto no caso restrito a árvores. Além disso, exibimos várias classes de desigualdades que induzem facetas do politopo associado. Mostramos ainda que é possível separar todas essas classes de desigualdades não-triviais em tempo $\mathcal{O}(nk)$.

Por fim, usamos as facetas obtidas no estudo do poliedro associado ao problema RC em árvores para desenvolver um algoritmo de *branch-and-cut*. Esse algoritmo obteve excelentes resultados executando instâncias baseadas em árvores filogenéticas reais.

Parte dos resultados apresentados nesta dissertação foram publicados em [CLMW13]. Um artigo completo contendo os resultados apresentados nos Capítulos 4, 5 e 6 foi submetido para publicação em um periódico [CFL⁺13].

Referências Bibliográficas

- [BBF99] Vineet Bafna, Piotr Berman e Toshihiro Fujito. A 2-approximation algorithm for the undirected feedback vertex set problem. *SIAM Journal on Discrete Mathematics*, 12:289–297, 1999. [20](#)
- [BFL⁺11] Hans L. Bodlaender, Michael R. Fellows, Michael A. Langston, Mark A. Ragan, Frances A. Rosamond e Mark Weyer. Quadratic kernelization for convex recoloring of trees. *Algorithmica*, 61(2):362–388, 2011. [5](#)
- [BM08] J. Adrian Bondy e U. S. R. Murty. *Graph Theory*, volume 244 of *Graduate Texts in Mathematics*. Springer-Verlag, 3 edição, 2008. [7](#)
- [BY00] R. Bar-Yehuda. One for the price of two: A unified approach for approximating covering problems. *Algorithmica*, 27:131–144, 2000. [21](#)
- [BYE85] R. Bar-Yehuda e S. Even. A local-ratio theorem for approximating the weighted vertex cover problem. *Annals of Discrete Mathematics*, 25:27–46, 1985. [21](#)
- [BYFR08] Reuven Bar-Yehuda, Ido Feldman e Dror Rawitz. Improved approximation algorithm for convex recoloring of trees. *Theory of Computing Systems*, 43(1):3–18, 2008. [4](#), [13](#), [57](#)
- [CFL⁺13] Manoel Campêlo, Alexandre Freire, Karla R. Lima, Phablo F.S. Moura e Yoshiko Wakabayashi. The convex recoloring problem: polyhedra, facets, and computational experiments. 2013. Submetido para publicação. [58](#)
- [CFR⁺07] B. Chor, M. Fellows, M.A. Ragan, I. Razgon, F. Rosamond e S. Snir. Connected coloring completion for general graphs: Algorithms and complexity. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 4598 LNCS:75–85, 2007. [5](#), [13](#), [14](#), [57](#)
- [CHSW13] Manoel B. Campêlo, Cristiana G. Huiban, Rudini M. Sampaio e Yoshiko Wakabayashi. On the complexity of solving or approximating convex recoloring problems. Em *Proceedings of the 19th International Conference on Computing and Combinatorics*, COCOON'13, 2013. [4](#)

- [CLMW13] Manoel Campêlo, Karla R. Lima, Phablo F.S. Moura e Yoshiko Wakabayashi. Polyhedral studies on the convex recoloring problem. *Electronic Notes in Discrete Mathematics*, 44(0):233–238, 2013. Proceedings of LAGOS’13 – VII Latin-American Algorithms, Graphs and Optimization Symposium. 58
- [dCCD⁺01] Marcelo Henriques de Carvalho, Márcia Rosana Cerioli, Ricardo Dahab, Paulo Feofiloff, Cristina Gomes Fernandes, Carlos Eduardo Ferreira, Katia Silva Guimarães, Flávio Keidi Miyazawa, José Coelho de Pina Jr., José Augusto R. Soares e Yoshiko Wakabayashi. *Uma introdução sucinta a algoritmos de aproximação*. 1 edição, may 2001. 7
- [Die10] Reinhard Diestel. *Graph Theory*, volume 173 of *Graduate Texts in Mathematics*. Springer-Verlag, 4 edição, 2010. 7
- [DMS11] Stefan Dantchev, Barnaby Martin e Stefan Szeider. Parameterized proof complexity. *Computational Complexity*, 20(1):51–85, 2011. 16
- [FW96] Carlos Eduardo Ferreira e Yoshiko Wakabayashi. *Combinatória Poliédrica e Planos-de-Corte Faciais*. X Escola de Computação-UNICAMP, 1 edição, 1996. 7
- [KK09] Iyad A. Kanj e Dieter Kratsch. Convex recoloring revisited: Complexity and exact algorithms. Em *Proceedings of the 15th Annual International Conference on Computing and Combinatorics*, COCOON ’09, páginas 388–397, Berlin, Heidelberg, 2009. Springer-Verlag. 4, 57
- [KT08] Frank Kammer e Torsten Tholey. The complexity of minimum convex coloring. *Lecture Notes in Computer Science*, 5369:16–27, 2008. ISAAC 2008. 5, 57
- [KT12] Frank Kammer e Torsten Tholey. The complexity of minimum convex coloring. *Discrete Applied Mathematics*, 160:810–833, 2012. 18
- [Lim11] Karla Roberta Lima. *Recoloração convexa de caminhos*. Tese de Doutorado, Instituto de Matemática e Estatística, Universidade de São Paulo, 2011. 43, 57
- [LW11] Karla Roberta Lima e Yoshiko Wakabayashi. Convex recoloring of paths. *Electronic Notes in Discrete Mathematics*, 37(0):165–170, 2011. 4, 57
- [MS05] Shlomo Moran e Sagi Snir. Convex recolorings of strings and trees: Definitions, hardness results, and algorithms. Em *Proceedings WADS 2005: 9th International Workshop on Algorithms and Data Structures*, páginas 218–232. Springer-Verlag, 2005. 3, 50, 57

- [MS07] Shlomo Moran e Sagi Snir. Efficient approximation of convex recolorings. *Journal of Computer and System Sciences*, 73(7):1078–1089, 2007. [4](#), [5](#), [13](#), [20](#), [57](#)
- [MS08] Shlomo Moran e Sagi Snir. Convex recolorings of strings and trees: Definitions, hardness results and algorithms. *Journal of Computer and System Sciences*, 74(5):850–869, 2008. [4](#), [52](#)
- [NW99] George L. Nemhauser e Laurence A. Wolsey. *Integer and Combinatorial Optimization*. Wiley-Interscience Series in Discrete Mathematics and Optimization. Wiley-Interscience, 1999. [7](#)
- [PHN08] Oriana Ponta, Falk Hüffner e Rolf Niedermeier. Speeding up dynamic programming for some np-hard graph recoloring problems. Em *Proceedings of the 5th international conference on Theory and applications of models of computation, TAMC'08*, páginas 490–501, Berlin, Heidelberg, 2008. Springer-Verlag. [5](#)
- [Raz07] Igor Razgon. A $2^{O(k)}$ $\text{poly}(n)$ algorithm for the parameterized convex recoloring problem. *Information Processing Letters*, 104(2):53–58, 2007. [4](#)
- [Vaz01] Vijay V. Vazirani. *Approximation algorithms*. Springer-Verlag New York, Inc., New York, NY, USA, 2001. [7](#)