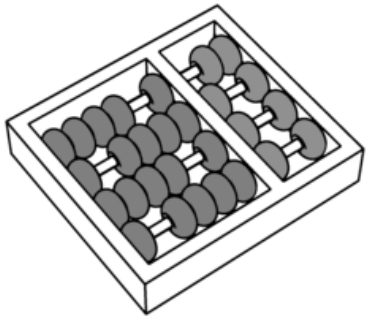
Universidade Estadual de Campinas

Instituto de Computação



Proposta de Dissertação de Mestrado

Recoloração Convexa de Grafos

Candidata: Ana Paula dos Santos Dantas

Orientador: Prof. Dr. Zanoni Dias

Coorientador: Prof. Dr. Cid Carvalho de Souza

Resumo

Dados um grafo $G = (V, E)$, um conjunto de cores $\mathcal{C}$ e uma coloração $C : V \to \mathcal{C}$ que colore os vértices de G, dizemos que a coloração C é convexa se cada classe de cor induz um subgrafo conexo. Quando uma coloração não é convexa, é importante saber qual o menor número de recolorações necessárias para alcançar essa propriedade. Uma recoloração convexa mínima é uma coloração obtida a partir de uma coloração não convexa com o menor número de vértices com cores trocadas. Esse problema surgiu a partir do estudo de árvores filogenéticas, em que a convexidade indica que não houveram convergências ou reversões durante a evolução de uma característica. Além da aplicação na biologia, o problema também pode ser aplicado a redes de computadores com poucas modificações. Nesta proposta, estudaremos o problema em árvores e grafos gerais. Para isso, desenvolveremos algoritmos com a meta-heurística GRASP e modelos matemáticos para as versões estudadas do problema. Por fim, serão realizados experimentos com os algortimos e modelos desenvolvidos neste trabalho e os presentes na literatura, a fim de comparar desempenho e qualidade da solução.

1 Introdução

Árvores filogenéticas são estruturas usadas para o estudo da evolução de um conjunto de organismos. Os vértices folhas representam as espécies atuais e os vértices internos antepassados hipotéticos dessas espécies. Os organismos presentes em uma árvore filogenética possuem características em comum, mas que podem se apresentar em diferentes estados [19].

Na versão mais simples do problema, cada árvore representa uma única característica e cada característica pode manifestar apenas um dos possíveis estados em cada espécie. Neste caso, podemos agrupar as espécies em classes que apresentam o mesmo estado da característica estudada. Ao atribuir uma cor para cada classe, cada vértice da árvore terá uma única cor, que representa o estado da característica [12].

Uma propriedade importante dessas classes é a convexidade. Uma classe ser convexa significa que não houve convergência ou reversão durante a evolução dessa característica. A convexidade é representada em uma árvore através da propriedade de que cada classe de cor deve induzir uma subárvore. Entretanto, algumas árvores

filogenéticas não possuem essa propriedade. Isso pode ser consequência de erros tanto na construção da árvore quanto na classificação das características das espécies [12].

Examinando o caso em que o erro pode ter ocorrido na observação da característica e a organização da árvore é confiável, Moran e Snir [19] introduziram o termo ***distância de recoloração***. Essa distância é dada pelo número de vértices que precisam ser recoloridos para que todas as classes sejam convexas. Assim, o Problema de Recoloração Convexa busca encontrar o menor número de alterações de cores de vértices que fazem com que a árvore se torne convexa.

Apesar de ter sua motivação em árvores, o problema pode ser generalizado para tratar grafos gerais, de modo que cada classe de cor deve induzir um grafo conexo. Variações do problema são comuns, tanto na classe do grafo, quanto na forma como a coloração é aplicada. As abordagens para resolver o problema variam desde soluções exatas [1, 9, 10, 16, 19, 22] a soluções aproximadas [2, 3, 14, 16].

Neste trabalho, abordaremos o problema de recoloração convexa em dois tipos de grafos: árvores e grafos gerais. No contexto de grafos gerais, também abordaremos o problema de Recoloração Convexa Restrita e Recoloração Convexa em Blocos, que possuem motivações em redes de computadores. Trataremos dos problemas com modelos matemáticos e algoritmos baseados na meta-heurística GRASP.

Este trabalho está organizado da seguinte maneira: na Seção 2 listamos os conceitos básicos utilizados; na Seção 3 discutimos as variações do problema, bem como suas respectivas complexidades; nas Seções 4 e 5 apresentamos, respectivamente, os objetivos e a metodologia que será empregada na execução; na Seção 6 apresentamos e discutimos os resultados preliminares; por último, o plano de trabalho na Seção 7.

2 Fundamentação Teórica

Para definirmos uma recoloração convexa, precisamos da definição de alguns conceitos. Um ***grafo*** G é definido como um par ordenado (V, E), sendo V um conjunto de vértices e E um conjunto de pares não ordenados de vértices, chamados de arestas, tal que $E \subseteq \binom{V}{2}$. Dizemos que dois vértices u e v são vizinhos se existe uma aresta $e = uv$.

Um ***caminho*** entre dois vértices v_0 e v_k é uma sequência de vértices e arestas

$\{v_0, e_1, v_1, \ldots, v_{k-1}, e_k, v_k\}$, tal que $e_i = v_{i-1}v_i$ e não existem vértices repetidos, ou seja, se $i \neq j$, então $v_i \neq v_j$ para todo $i, j \in \{0, 1, \ldots, k\}$. Dizemos que um grafo G é ***conexo*** se e somente se existe um caminho entre quaisquer dois vértices de G. Uma ***árvore*** é um grafo conexo com exatamente $|V| - 1$ arestas [5].

Seja $S \subseteq V$ um subconjunto de vértices de um grafo G, chamamos de subgrafo ***induzido*** por S, ou $G[S]$, o grafo cujo conjunto de vértices é o próprio conjunto S e o conjunto de arestas de S é igual a $\{e = uv |\ e \in E$, e $u, v \in S\}$, ou seja, são as arestas do grafo original que possuem os dois extremos em S.

Uma ***coloração***[1] de um grafo consiste na associação de uma cor a um vértice, ou seja, uma função que recebe como entrada um vértice e retorna uma cor associada a ele. Usamos números para indicar cores, de modo que um conjunto de k cores é definido da seguinte forma: $\mathcal{C} = \{1, 2, \ldots, k\}$. Também usamos $\emptyset$ para representar a ausência de cor em um vértice. Uma ***classe de cor*** é o conjunto de vértices que possuem a mesma cor.

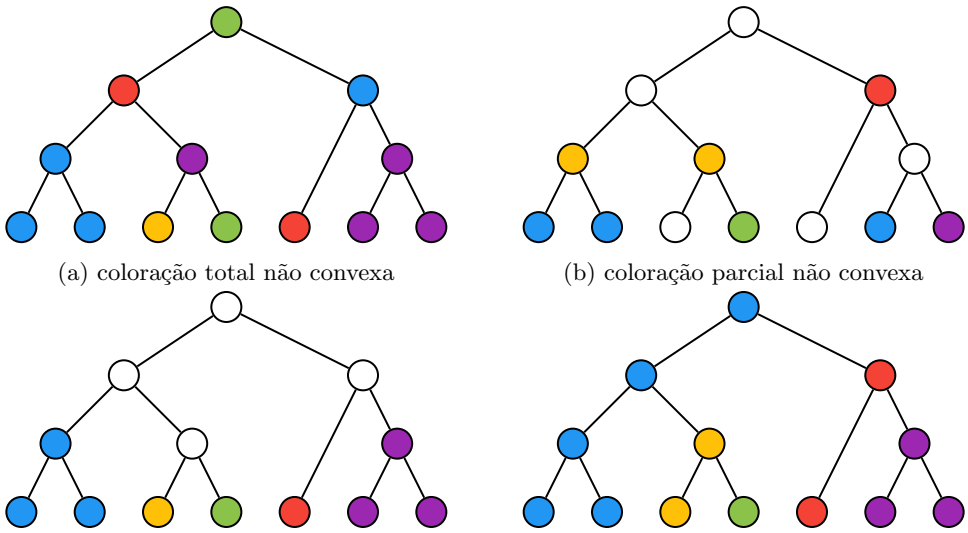
Se uma coloração $C : V \to \mathcal{C}$ define uma cor para todos os vértices do grafo, então chamamos essa coloração de ***coloração total***[1]. Chamamos de ***coloração parcial*** uma coloração que admite que um vértice não tenha cor associada a ele, definida por $C : V \to \mathcal{C} \cup \{\emptyset\}$. Chamamos de ***coloração convexa*** uma coloração na qual cada classe de cor induz um subgrafo conexo. Uma ***coloração boa*** é uma coloração que é parcial e convexa [8].

Uma coloração total pode ser obtida a partir de uma coloração boa em tempo linear de modo a manter a convexidade. Para tal, basta encontrar um caminho entre um vértice x sem cor, $C(x) = \emptyset$, e um vértice colorido y tal que $C(y) = c,\ c \in \mathcal{C}$, definir a cor dos vértices visitados como c, e repetir o processo enquanto houver vértice sem cor [8].

A Figura 1 mostra os tipos de coloração descritos, onde um vértice branco representa um vértice sem cor. Em particular, na Figura 1(c) temos uma coloração parcial, onde todas as classes de cor induzem subgrafos conexos, logo a coloração é boa. A Figura 1(d) apresenta uma coloração total obtida a partir da coloração anterior usando o algoritmo apresentado por Campêlo *et al.* [8].

Uma ***recoloração*** C' de um grafo colorido (G, C) é uma coloração no mesmo

[1] Termo usado aqui com significado diferente daquele na coloração clássica.



(a) coloração total não convexa (b) coloração parcial não convexa

(c) coloração boa (d) coloração total convexa

Figura 1: Exemplos dos tipo de colorações apresentados com variações na totalidade e convexidade.

conjunto de vértices. Quando criamos uma recoloração, queremos alcançar uma certa propriedade no grafo. No nosso caso, a convexidade. Podemos medir a ***distância de recoloração*** usando uma função $w : V \to \mathbb{Q}_{\geq 0}$, que indica quanto custa mudar a cor de um determinado vértice. Na versão não ponderada do problema, o custo de mudança de cor é unitário.

Dados uma grafo colorido (G, C) e uma recoloração C', definimos o conjunto de vértices $R_C(C')$ como o conjunto dos vértices que tiveram sua cor alterada na recoloração, ou seja, $R_C(C') = \{v \in V \mid C(v) \neq \emptyset \text{ e } C(v) \neq C'(v)\}$. O conjunto $R_C(C')$ e a função de custo w são usados para calcular o custo total de uma solução.

3 Recoloração Convexa

Nesta seção apresentaremos o problema de recoloração convexa e suas variações de forma mais detalhada. Além de resultados sobre a complexidade e as abordagens encontradas na literatura para resolver os problemas.

(a) coloração não-convexa (b) coloração convexa

Figura 2: Recoloração Convexa de Caminhos. A Figura 2(a) apresenta um exemplo de instância para o problema CPR. Na Figura 2(b) os dois vértices destacados indicam os vértices que foram recoloridos. A Figura 2(b) apresenta uma recoloração mínima, que transforma a anterior em uma coloração convexa.

3.1 Caminhos

A versão do problema que usa um caminho como o grafo de entrada é conhecida como Recoloração Convexa de Caminhos (Definição 3.1). Um caminho é ainda uma árvore que possui exatamente duas folhas, e pode ser chamado de *string*. Moran e Snir [19] provaram que a Recoloração Convexa de Caminhos é NP-Difícil. A Figura 2 apresenta um exemplo de uma recoloração convexa de um caminho.

Definição 3.1. Recoloração Convexa de Caminhos (Convex Path Recoloring - CPR)

Entrada: um caminho P, um conjunto de cores $\mathcal{C}$, uma coloração parcial C e uma função de custo w;

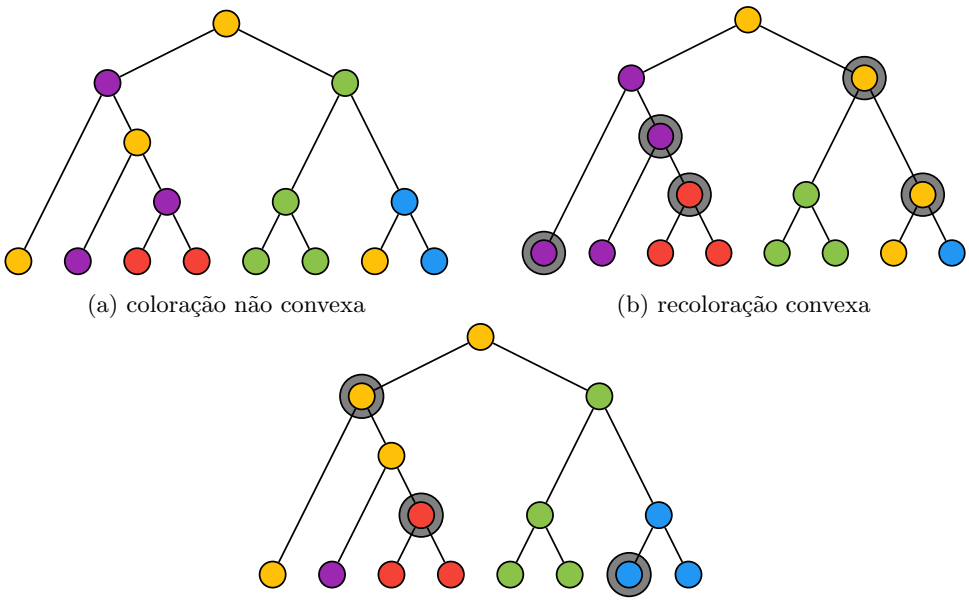
Objetivo: encontrar uma coloração boa C' com custo mínimo, $\min(\sum_{v \in R_C(C')} w(v))$;

Saída: coloração boa C'.

Como um caminho também é uma árvore, os algoritmos exatos para o problema de recoloração de árvores também podem ser utilizados para recoloração de caminhos. Entretanto o problema de Recoloração Convexa de Caminhos possui uma 2-aproximação [18], além de um modelo de programação linear específico apresentado por Lima e Wakabayashi [16]. Lima [17] desenvolveu estudos sobre as facetas do poliedro e combinações de *branch-and-cut* com programação dinâmica que obtiveram bons resultados. Ao adicionar a restrição de que cada classe de cor possua no máximo dois vértices, versão conhecida como 2-CPR, o melhor fator de aproximação conhecido passa a ser 5/4 [3].

3.2 Árvores

A versão do problema com árvores (Definição 3.2) é a mais estudada na literatura, devido à sua aplicação em árvores filogenéticas. A Figura 3 mostra



(a) coloração não convexa

(b) recoloração convexa

(c) recoloração convexa mínima

Figura 3: Recoloração Convexa em Árvores. A Figura 3(a) apresenta uma instância do problema CTR. As figuras 3(b) e 3(c) apresentam recolorações convexas, entretanto apenas esta última é uma recoloração mínima.

um exemplo de instância dessa versão do problema. As figuras seguintes são duas recolorações convexas da instância inicial. A recoloração na Figura 3(b) troca as cores de 5 vértices, enquanto a da Figura 3(c) troca apenas 3, e esse é o menor número possível de trocas. Logo, a distância de recoloração é 3.

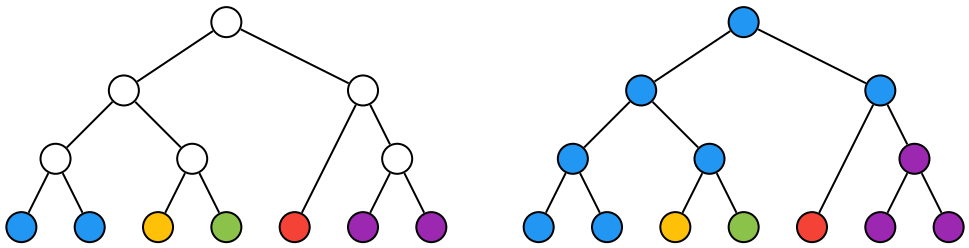
Definição 3.2. Recoloração Convexa de Árvores (Convex Tree Recoloring - CTR)

Entrada: uma árvore T, uma coloração parcial C, um conjunto de cores $\mathcal{C}$ e uma função de custo w;

Objetivo: encontrar uma coloração boa C' com custo mínimo, $\min(\sum_{v \in R_C(C')} w(v))$;

Saída: coloração boa C'.

Moran e Snir [19] provaram que o problema é NP-Difícil. Kanj e Kratsch [15] e Moran e Snir [18] criaram algoritmos exatos com tempo exponencial para encontrar a distância de recoloração. A melhor aproximação conhecida na literatura é uma $(2+\epsilon)$-aproximação, de Bar-Yehuda, Feldman e Rawitz [2]. Esse algoritmo é dividido em duas partes, a primeira com complexidade de tempo de $\mathcal{O}(n^2)$ e a segunda $\mathcal{O}(n^2 + nk^2 2^k)$, onde n é o número de vértices e k é um parâmetro de acurácia definido por $k = \lceil \frac{2}{\epsilon} \rceil + 1$.



(a) coloração parcial não convexa (b) coloração total convexa

Figura 4: Recoloração Convexa de Árvores com Apenas Folhas Coloridas. A Figura 4(a) traz uma coloração não convexa, mas que pode ser estendida para coloração convexa com custo zero, como mostrado na Figura 4(b).

Chopra *et al.* [10] desenvolveram um modelo de programação linear inteira com bons resultados computacionais. Esse modelo tem número linear de restrições e de variáveis. Em uma continuação do trabalho anterior, Chopra *et al.* [9] trataram o problema com um modelo de geração de colunas, o qual também tem um bom desempenho computacional. Esse novo modelo foi capaz de resolver instâncias maiores que o modelo anterior não consguia.

Uma modificação natural para o problema surge a partir da forma como as árvores filogenéticas são construídas. Como os pesquisadores conhecem apenas as espécies que estão representadas nas folhas dessas árvores, podemos desconsiderar as suposições feitas sobre os antepassados removendo a coloração desses nós. A Definição 3.3 enuncia essa variação do problema.

Definição 3.3. Recoloração Convexa de Árvores com Apenas Folhas Coloridas (Convex Leaf-Only Recoloring - CLR)

Entrada: uma árvore $T = (V, E)$, um conjunto de cores $\mathcal{C}$, uma coloração parcial $C : V \to \mathcal{C} \cup \{\emptyset\}$, tal que $C(v) \neq \emptyset$, se e somente se v é uma folha e uma função de custo w;

Objetivo: encontrar uma coloração boa C' com custo mínimo, $\min(\sum_{v \in R_C(C')} w(v))$;

Saída: coloração boa C'.

Bachoore e Bodlaender [1] apresentam um algoritmo linear que testa se a coloração parcial das folhas pode ser estendida para uma coloração total com custo 0, ou seja, sem a necessidade de trocar as cores dos vértices. A Figura 4 exemplifica o caso identificado pelo algoritmo.

Além disso, Bachoore e Bodlaender [1] apresentaram duas estratégias de preprocessamento de uma instância sem prejudicar a solução ótima. A primeira estratégia aplicada é a de simplificação, que reduz a árvore para uma árvore menor e equivalente. A segunda é a de divisão e conquista, que divide a árvore em duas ou mais árvores menores. Por último, também apresentam um algoritmo exato usando técnicas de *branching*, o qual é usado para provar que o problema é tratável em parâmetro-fixo.

Essa versão do problema sem limitações no número de cores também é NP-Difícil [19]. Entretanto, Kanj e Kratsch [15] provaram que se o tamanho máximo de cada classe de cor é menor ou igual a 3, o problema pode ser resolvido em tempo polinomial reduzindo-o ao problema de conjuntos independentes em grafos cordais. Essa restrição no problema é conhecida como 3-CLR.

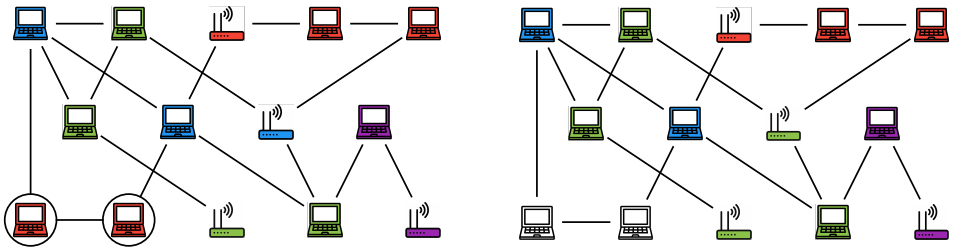
3.3 Grafos Gerais

Quando o problema não especifica o tipo do grafo de entrada, ele é chamado apenas de Recoloração Convexa (Definição 3.4) e também é NP-Difícil [20]. Moura [21] provou que mesmo quando a instância possui apenas duas cores, $|\mathcal{C}| = 2$, não existe uma $2^{poly(n)}$-aproximação para o problema, sendo n o número de vértices e $poly(n)$ um polinômio em n.

Definição 3.4. Recoloração Convexa (Convex Recoloring - CR)
Entrada: um grafo qualquer G, um conjunto de cores $\mathcal{C}$, uma coloração parcial C e uma função de custo w;
Objetivo: encontrar uma coloração boa C' com custo mínimo, $\min(\sum_{v \in R_C(C')} w(v))$;
Saída: coloração boa C'.

Campêlo *et al.* [6] introduziram um modelo de programação linear para grafos gerais, juntamente com estudos sobre as facetas do poliedro. No entanto não reportaram resultados de experimentos computacionais. Moura [22] desenvolveu um modelo de programação linear para grafos gerais, combinado com a técnica de geração de colunas.

Um resultado interessante que Campêlo *et al.* [7] provaram é que existem classes de grafos - cografos e grafos com poucos P4's - nas quais o problema pode ser resolvido em tempo linear.



(a) coloração total não convexa (b) coloração convexa

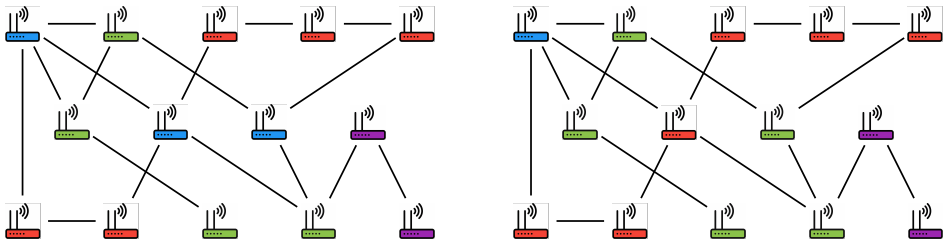
Figura 5: Recoloração Convexa Restrita. Os vértices em vermelho que estão circulados não podem ser conectados com o demais clientes em vermelho, pois eles estão separados por clientes de outras cores (azul e verde). Logo em uma recoloração dessa instância, os vértices circulados seriam descoloridos.

3.4 Recoloração em Redes de Computadores

Kammer e Tholey [14] aplicaram o problema de recoloração a redes de computadores, supondo que uma rede de computadores seja modelada como um grafo, cujos vértices são roteadores e clientes e as arestas representam a comunicação entre os dispositivos. Nesse contexto, o problema possui algumas diferenças provenientes do domínio da aplicação que resultam na criação de duas variações do problema chamadas de Recoloração Convexa Restrita e Recoloração Convexa em Blocos.

Na Recoloração Convexa Restrita (Definição 3.5) supomos que a cor de um cliente indica um serviço que este possui e que os roteadores possuem todos os serviços. Deste modo, os vértices clientes não podem ser usados para estabelecer conexões entre clientes de outras cores, pois não possuem outros serviços. Isso é interpretado no problema como as seguintes restrições: apenas roteadores podem ser recoloridos e se um cliente não pode ser conectado com os outros clientes da mesma cor, ele pode ser apenas descolorido.

Na Figura 5(a) apenas o conjunto de vértices da cor roxo induzem um subgrafo conexo. Existem dois clientes vermelhos que estão marcados, esses vértices não possuem comunicação com os outros vértices vermelhos através de um roteador, apenas por clientes de outras cores (azul e verde). Logo, não é possível conectar os vértices vermelhos. Assim a única possibilidade para atingir a convexidade é descolorir os vértices marcados. Já para os vértices verdes, existe uma conexão entre os dois grupos por meio de um roteador. A Figura 5(b) mostra uma solução para a instância.



(a) coloração total não convexa (b) coloração total convexa

Figura 6: Recoloração Convexa em Blocos. A Figura 6(a) apresenta um modelo de instância para o problema, onde todos os vértices são interpretados como roteadores. A Figura 6(b) apresenta uma recoloração convexa em blocos com custo um, pois apenas vértices da classe de cor azul mudaram de cor.

Definição 3.5. Recoloração Convexa Restrita (Minimum Restricted Convex Recoloring Problem - MRRP)

Entrada: um grafo $G = (V_r \cup V_c, E)$, onde V_r é o conjunto de vértices roteadores e V_c é conjunto dos vértices clientes, um conjunto de cores $\mathcal{C}$, uma coloração parcial C e uma função de custo w;

Objetivo: encontrar uma coloração boa C' com custo mínimo, $\min(\sum_{v \in R_C(C')} w(v))$, tal que para todo vértice cliente $v \in V_c$, $C'(v) = C(v)$ ou $C'(v) = \emptyset$;

Saída: coloração boa C'.

Já o problema de Recoloração Convexa em Blocos (Definição 3.6) modifica a função de custos e considera apenas os vértices roteadores. Nesse caso um custo é atribuído a cada cor c e deve ser pago se, na recoloração, um vértice que tinha a cor c foi recolorido. A Figura 6 contém um exemplo de instância desse problema.

Definição 3.6. Recoloração Convexa em Blocos (Minimum Block Recoloring Problem - MBRP)

Entrada: um grafo qualquer G, um conjunto de cores $\mathcal{C}$, uma coloração parcial C e uma função de custo $w : \mathcal{C} \to \mathbb{Q}$;

Objetivo: encontrar uma coloração boa C' com custo mínimo, $\min(\sum_{c \in B(C')} w'(c))$, com $B(C')$ sendo o conjunto de cores que possuem um vértice recolorido em C', $B(C') = \{c \in C | \ \exists\ v \in V$ tal que $C(v) = c$ e $C'(v) = c'$, com $c \neq c'\}$;

Saída: coloração boa C'.

A Tabela 1 apresenta a complexidade dos dois problemas considerando como

entrada grafos com largura arbórea limitada [4]. Na tabela, k representa o número de cores, n o número de vértices e b o número de vértices por classe de cor. A tabela mostra que existem versões dos problemas que podem ser resolvidas em tempo polinomial.

	$k \in O(log\ n)$	$k \in O(n)$	
		$b = 2, b = 3$	$b \geq 4$
MRRP	P	P	NP-Difícil
MBRP	P	P	P

Tabela 1: Complexidades dos problemas Recoloração Convexa Restrita (MRRP) e Recoloração Convexa em Blocos (MBRP) para grafos com largura arbórea limitada.

Além das novas versões do problema, Kammer e Tholey [14] também criaram uma (2+ϵ)-aproximação para o problema aplicado a grafos com largura arbórea limitada. Esse algoritmo é baseado no algoritmo de Bar-Yehuda, Feldman e Rawitz [2], e o parâmetro ϵ tem o mesmo significado. Também é provado que se o número de vértices por classe de cor for no máximo três, as duas versões, MRRP e MBRP, podem ser resolvidas em tempo polinomial, resultado da segunda coluna apresentado na Tabela 1.

4 Objetivos

Neste trabalho pretendemos estudar a recoloração convexa em árvores e em grafos gerais. Na parte do trabalho com árvores, faremos uma análise do desempenho dos métodos de solução para árvores, aplicando-os à árvores com apenas folhas coloridas. Também criaremos um algoritmo GRASP para comparação de desempenho.

Já para a parte do trabalho com grafos gerais, desenvolveremos um modelo matemático e um algoritmo GRASP para a resolução do problema. Avaliaremos ambas abordagem comparando-as entre si e com aquelas encontradas na literatura. Ainda em grafos gerais, pretendemos adaptar o modelo matemático e a meta-heurística para os problemas de Recoloração Convexa Restrita e Recoloração Convexa em Blocos.

Criaremos também um gerador de instâncias, que possa ser usado para *benchmarking* do problema em grafos gerais nas três formas citadas acima.

5 Metodologia

Esta seção detalha quais os passos que tomaremos para atingir os objetivos especificados na Seção 4. Para ambos os tipos de grafos, a abordagem é dividida em três etapas: modelagem matemática, meta-heurística e análise dos resultados. Cada etapa é detalhada a seguir.

5.1 Modelagem Matemática

Problemas de otimização combinatória podem ser modelados como problemas de programação linear inteira. Um programa linear inteiro é um programa linear com a restrição de que as variáveis de decisão são inteiras. Neste trabalho utilizaremos modelos de programação interia para obter soluções exatas para os problemas estudados.

Para a modelagem matemática do problema de Recoloração Convexa em Árvores usaremos o modelo de programação linear de Chopra *et al.* [9] apresentado a seguir. O modelo possui duas variáveis de decisão x_{ut} e y_{et}, ambas binárias. A variável x_{ut} recebe o valor 1 quando o vértice u for colorido com a cor t. Analogamente, a variável y_{et} recebe o valor 1 quando a aresta e for colorida com a cor t. Os conjuntos $\mathcal{C}, V, E$ são conjuntos de cores, vértices e arestas, respectivamente. A constante k indica o número de cores na coloração inicial e as contantes $w(t, v)$ recebem o valor $w(v)$ se a cor do vértice v é t, ou seja, $C(v) = t$, e zero caso contrário.

$$\max \sum_{t=1}^{k} \sum_{u \in V} w(u,t) x_{ut} \tag{1}$$

$$\text{s.a} \quad \sum_{t=1}^{k} x_{ut} \leq 1 \qquad \forall u \in V \tag{2}$$

$$\sum_{u \in V} x_{ut} - \sum_{e \in E} y_{et} \leq 1 \qquad \forall t \in \mathcal{C} \tag{3}$$

$$-x_{ut} + y_{uvt} \leq 0 \qquad \forall uv \in E,\ \forall t \in \mathcal{C} \tag{4}$$

$$-x_{vt} + y_{uvt} \leq 0 \qquad \forall uv \in E,\ \forall t \in \mathcal{C} \tag{5}$$

$$x_{ut} \in \{0,1\} \qquad \forall u \in V,\ \forall t \in \mathcal{C} \tag{6}$$

$$y_{et} \in \{0,1\} \qquad \forall e \in E,\ \forall t \in \mathcal{C} \tag{7}$$

A função objetivo, Equação (1), desse modelo maximiza o número de vértices que mantém a cor original, que é equivalente a minimizar o número de vértices recoloridos. As Desigualdades (2) restringem que cada vértice só pode receber no máximo uma única cor e as Desigualdades (3) limitam o número de arestas ao número de vértices com a cor t menos um. Essas equações remetem à propriedade de árvores que diz que o número de arestas é igual ao número de vértices menos um. Entretanto essas restrições não garantem conexidade. Esse problema é resolvido pelas Desigualdades (4) e (5) que restringem que uma aresta seja colorida com a cor t apenas se os dois vértices das pontas estiverem coloridos com a t.

O modelo obteve bons resultados nos experimentos de Chopra *et al.* [9] usando árvores com colorações totais como instâncias. Neste trabalho investigaremos o seu desempenho com o problema CLR e depois faremos comparações com a meta-heurística.

Para o problema de recoloração convexa em grafos gerais desenvolveremos um novo modelo. Esse novo modelo será comparado com os outros mencionados na Seção 3.3 e, posteriormente, será adaptado para os problemas MRRP e MBRP. O desempenho também será comparado aos das meta-heurísticas.

5.2 Meta-Heurística GRASP

A meta-heurística GRASP (*Greedy Randomized Adaptive Search Procedure*) é um procedimento iterativo, que guarda a melhor solução geral como resultado. Cada iteração é dividida em duas partes: construção e busca local [11]. O Algoritmo 1 apresenta um esboço do GRASP simples.

Algorithm 1 GRASP - *Greedy Randomized Adaptive Search Procedure*

```
1: função GRASP(max-iterações, semente)
2:     LerInstancia();
3:     melhor-solução ← ∅;
4:     para k ← 1 até max-iterações faça
5:         solução ← ConstruirSoluçãoGulosaAleatorizada(semente);
6:         solução ← BuscaLocal(solução);
7:         AtualizarSolução(solução, melhor-solução);
8:     fim para
9:     devolve melhor-solução
10: fim função
```

O GRASP possui apenas dois parâmetros de entrada, o número máximo de

iterações e a semente usada para a geração de números pseudo aleatórios. Para construir uma solução, a sub-rotina ConstruirSoluçãoGulosaAleatorizada cria uma lista das melhores escolhas naquele ponto e seleciona aleatoriamente uma delas para adicionar na solução. A criação da lista e adição do novo elemento são passos repetidos até que uma solução esteja completa.

As três característica do GRASP são representadas nessa sub-rotina. A adaptatividade está presente durante a construção da lista, que é influenciada pelas escolhas anteriores. A característica gulosa está na criação da lista e a aleatoriedade está em qual escolha será usada na solução.

A sub-rotina BuscaLocal explora a vizinhança da solução encontrada pela sub-rotina anterior. Já a sub-rotina AtualizarSolução apenas guarda qual a melhor solução encontrada até o momento.

Escolhemos esta meta-heurística pela sua simplicidade e pela robustez de variações presentes na literatura [23]. Implementaremos versões do GRASP para o problema de recoloração convexa aplicado a árvores e grafos gerais.

5.3 Análise dos Resultados e Validação

Os resultados obtidos em cada abordagem, modelagem matemática e meta-heurística, serão comparados entre si. Quando possível, também serão comparados com soluções encontradas na literatura. Usaremos técnicas estatísticas para analisar os dados obtidos através dos experimentos.

Inicialmente, os pontos avaliados serão tempo de execução e qualidade da solução encontrada. Para as medidas de tempo, usaremos as médias dos tempos de execução de cada instância. Já para a qualidade da solução, quantificaremos a diferença entre os valores das soluções encontradas.

Toda a implementação será documentada e publicada juntamente com as instâncias e o gerador que serão usados nos experimentos deste trabalho.

6 Resultados Preliminares

Nesta fase do projeto, realizamos experimentos com o modelo de programação linear inteira de Chopra *et al.* [9] descrito na Seção 5.1. Implementamos o modelo

utilizando a linguagem de modelagem OPL®, que é parte do pacote CPLEX Studio da IBM® [13]. Após a implementação, os testes foram realizados com as instâncias originais usadas por Chopra *et al.* [9].

As árvores usadas nas instâncias são árvores filogenéticas obtidas da base de dados TreeBase[2] e as colorações foram criadas usando a metodologia desenvolvida por Campêlo *et al.* [8]. Para criar a coloração inicial de uma árvore, são definidos dois parâmetros p_c e p_n, com $0 \leq p_c,\ p_n \leq 1$, onde $1 - p_c$ indica a probabilidade de um vértice manter a cor do vértice pai e $1 - p_n$ a probabilidade de um vértice manter sua cor na segunda fase da criação da instância.

O procedimento começa pela raiz da árvore, determinada na construção da árvore, que recebe a cor 1. Para cada vértice filho, é escolhido aleatoriamente, com probabilidade $1 - p_c$, se este recebe a mesma cor que o pai ou a próxima cor que ainda não foi usada. Esse passo é aplicado recursivamente em todos vértices. Dessa forma, é criada uma coloração convexa e número de cores usadas é aleatório, influenciado pelo parâmetro p_c.

O próximo passo é introduzir ruído. Para cada vértice é feita uma escolha aleatória, com probabilidade $1 - p_n$, se o vértice mantém a sua cor ou recebe uma cor aleatória dentre as que já foram usadas. Ao final desse procedimento, temos uma coloração não convexa.

A partir das instâncias cedidas por Chopra *et al.* [9], criamos instâncias coloridas apenas nas folhas. Para tal, apenas removemos as cores dos vértices internos. As instâncias foram divididas nos conjuntos *Original* e *LeafOnly* que representam as instâncias originais para o problema CTR e as instâncias coloridas apenas nas folhas para o problema CLR, respectivamente.

Executamos o modelo com os dois conjuntos de instâncias em uma máquina com CPU Intel(R) Core(TM) i7-4790K com 8 *cores* de 4.00GHz, 16GB de memória RAM e sistema operacional ArchLinux, versão 4.2.5-1-ARCH. Como *solver* do modelo, usamos o IBM CPLEX Studio, versão 12.8, com as funções de *presolve* e paralelismo desativadas, pois Chopra *et al.* [9] não usaram essas funções. Foram executadas 126 instâncias criadas a partir de 14 árvores filogenéticas, com $n \in \{301, 404, 567, 636, 710, 813, 981, 1441, 1838, 2025, 2387, 2409, 2632\}$ variando os

[2]Disponível em http://treebase.org/treebase-web

$n \setminus p_c$		0.5%	5%	50%	$n \setminus p_c$		0.5%	5%	50%
213	k	0.00%	0.00%	-21.81%	**981**	k	0.00%	-1.04%	-20.48%
	t	-9.82%	33.82%	-42.49%		t	6.42%	46.10%	-67.96%
301	k	0.00%	0.00%	-24.10%	**1441**	k	0.00%	0.00%	-21.65%
	t	11.59%	47.60%	-57.73%		t	12.74%	42.03%	-85.33%
404	k	0.00%	0.00%	-22.90%	**1838**	k	0.00%	-0.59%	-23.08%
	t	-6.46%	29.14%	-62.40%		t	32.62%	47.26%	-73.38%
567	k	0.00%	0.00%	-11.62%	**2025**	k	0.00%	-0.37%	-14.15%
	t	-20.55%	12.49%	-35.99%		t	24.47%	45.60%	-57.06%
636	k	0.00%	0.00%	-22.09%	**2387**	k	0.00%	0.00%	-23.70%
	t	-6.49%	41.65%	-73.34%		t	53.00%	34.31%	-78.30%
710	k	0.00%	0.00%	-18.84%	**2409**	k	0.00%	0.00%	-22.34%
	t	4.67%	27.26%	-71.32%		t	28.78%	35.49%	-72.27%
813	k	0.00%	-0.81%	-24.68%	**2632**	k	0.00%	-1.04%	-22.76%
	t	12.86%	36.95%	-73.89%		t	48.83%	52.29%	-71.81%

Tabela 2: Média da diferença percentual do número de cores iniciais (linha k) e do tempo de execução (linha t) entre as instâncias para o problema CLR e CTR, para cada instância com p_c fixo e p_n variado.

parâmetros $p_n \in \{25\%, 50\%, 75\%\}$ e $p_c \in \{0.5\%, 5\%, 50\%\}$.

Na Tabela 2, a coluna n indica o número de vértices da instância e as três colunas seguintes indicam o valor do parâmetro p_c. Para cada valor de n temos as linhas k e t, que indicam a média da diferença percentual do número de cores iniciais usadas nas instâncias e o tempo de execução das instâncias *Original* e *LeafOnly*. Para cada valor do parâmetro p_c foi usada a média dos valores obtidos na execução das três colorações obtidas com o valor de p_c fixo, que influencia no número de cores utilizado na coloração convexa inicial, variando p_n, que influencia na distância de recoloração. O tempo limite de execução foi de duas horas. As instâncias com os cinco maiores números de vértices ($n > 2000$) e $(p_n, p_c) = (50\%, 75\%)$ não obtiveram solução ótima.

Quando o valor da linha k é igual a 0.00%, isso significa que as instâncias dos dois problemas, CTR e CLR, utilizaram o mesmo número de cores; quando o valor é negativo, isso significa que a instância para o problema CLR utilizou menos cores que instância para o problema CTR. Analogamente, quando o valor da linha t é negativo, indica que a instância para o problema CLR foi resolvida mais rapidamente.

Observando as colunas cujos valores de $p_c \in \{0.5\%, 5\%\}$, percebemos que a tendência é que as instâncias para o problema CLR levem mais tempo que as instâncias do CTR para serem resolvidas quando o número de cores é o mesmo. Analisando as colunas cujo valor de $p_c = 50\%$, percebemos que os tempos de execução são menores para

as instâncias do problema CLR (todos os valores das linhas t são negativos). Contudo, o número de cores usados, de fato, na coloração inicial também é menor. A redução no tempo de execução pode ser explicada por esse fato.

Com base nestes resultados iniciais, concluímos que as instâncias do problema CLR são mais difíceis de resolver, do que as do problema CTR. Entretanto, estes testes preliminares foram realizados com apenas uma instância para cada combinação de valores de p_c e p_n, o que compromete a relevância estatística desses resultados. Por este motivo, planejamos realizar experimentos com mais instâncias por par (p_n, p_c) para validar nossas conclusões.

7 Plano de Trabalho

	2017					2018											2019						
	A	S	O	N	D	J	F	M	A	M	J	J	A	S	O	N	D	J	F	M	A	M	J	J
1	*	*	*	*	*			*	*	*	*	*												
2		*	*	*	*	*	*			*			*			*			*			*		
3						*	*	*																
4									*															
5													*	*	*	*	*							
6							*	*																
7									*	*	*													
8												*	*	*										
9															*	*	*							
10																		*	*	*				
11											*	*				*	*			*	*			
12												*	*				*	*			*	*	*	
13																							*	
14																								*

Tabela 3: Cronograma de atividades.

A Tabela 3 apresenta o cronograma das atividades a serem executadas no decorrer do trabalho, que são listadas a seguir:

1. Obtenção de créditos obrigatórios em disciplinas do programa de mestrado;
2. Revisão bibliográfica;
3. Escrita da proposta de mestrado;
4. Exame de Qualificação de Mestrado;
5. Participação no Programa de Estágio Docente (PED);

6. Implementação de modelos para Recoloração Convexa em Árvores;
7. Criação do Algoritmo GRASP para o problema de Recoloração Convexa em Árvores;
8. Modelagem do problema de Recoloração Convexa;
9. Criação do Algoritmo GRASP para o problema de Recoloração Convexa em grafos gerais;
10. Adaptação do modelo e do GRASP aos problemas de Recoloração Convexa Restrita e Recoloração Convexa em Blocos;
11. Análise dos resultados;
12. Escrita da dissertação;
13. Revisão da dissertação;
14. Defesa da dissertação.

O tempo alocado e a ordem de execução das atividades podem ser alterados no decorrer do desenvolvimento da pesquisa, uma vez que alguns resultados podem ser mais promissores que os outros, fazendo com que alguma(s) atividade(s) sejam realocada(s).

Referências

[1] E. H. Bachoore and H. L. Bodlaender. Convex Recoloring of Leaf-Colored Trees. Technical Report UU-CS-2006-010, Department of Information and Computing Sciences, Utrecht University, 2006.

[2] R. Bar-Yehuda, I. Feldman, and D. Rawitz. Improved Approximation Algorithm for Convex Recoloring of Trees. ***Theory of Computing Systems***, 43(1):3–18, 2008.

[3] R. Bar-Yehuda, G. Kutiel, and D. Rawitz. 1.5-Approximation Algorithm for the 2-Convex Recoloring Problem. In ***Proceedings of the 26th International Workshop on Combinatorial Algorithms (IWOCA'2015)***, Theoretical

Computer Science and General Issues, pages 299–311, Cham, Switzerland, 2015. Springer International Publishing.

[4] H. L. Bodlaender. ***Classes of graphs with bounded tree-width***, volume 86. Unknown Publisher, 1986.

[5] J. A. Bondy and U. S. R. Murty. ***Graph Theory***, volume 244 of ***Graduate Texts in Mathematics***. Springer-Verlag London, 1st edition, 2011.

[6] M. B. Campêlo, K. R. Lima, P. F. S. Moura, and Y. Wakabayashi. Polyhedral Studies on the Convex Recoloring Problem. ***Electronic Notes in Discrete Mathematics***, 44:233–238, 2013.

[7] M. B. Campêlo, C. G. Huiban, R. M. Sampaio, and Y. Wakabayashi. Hardness and Inapproximability of Convex Recoloring Problems. ***Theoretical Computer Science***, 533:15–25, 2014.

[8] M. B. Campêlo, A. S. Freire, K. R. Lima, P. F. S. Moura, and Y. Wakabayashi. The Convex Recoloring Problem: Polyhedra, Facets and Computational Experiments. ***Mathematical Programming***, 156(1-2):303–330, 2016.

[9] S. Chopra, E. Erdem, E. Kim, and S. Shim. Column Generation Approach to the Convex Recoloring Problem on a Tree. In ***Proceedings of the 16th Modeling and Optimization: Theory and Applications Conference (MOPTA'2016)***, volume 213 of , pages 39–53, Cham, Switzerland, 2016. Springer International Publishing.

[10] S. Chopra, B. Filipecki, K. Lee, M. Ryu, S. Shim, and M. V. Vyve. An Extended Formulation of the Convex Recoloring Problem on a Tree. ***Mathematical Programming***, 165(2):529–548, 2017.

[11] T. A. Feo and M. G. C. Resende. Greedy Randomized Adaptive Search Procedures. ***Journal of Global Optimization***, 6(2):109–133, 1995.

[12] Z. Frenkel, Y. Kiat, I. Izhaki, and S. Snir. Convex Recoloring as an Evolutionary Marker. ***Molecular Phylogenetics and Evolution***, 107:209–220, 2017.

[13] ***IBM ILOG CPLEX Optimization Studio - OPL Language Reference Manual***. IBM.

[14] F. Kammer and T. Tholey. The Complexity of Minimum Convex Coloring. ***Discrete Applied Mathematics***, 160(6):810–833, 2012.

[15] I. A. Kanj and D. Kratsch. Convex Recoloring Revisited: Complexity and Exact Algorithms. In ***Proceedings of the 15th International Computing and Combinatorics Conference (COCOON'2009)***, Lecture Notes in Computer Science, pages 388–397, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[16] K. R. Lima and Y. Wakabayashi. Convex Recoloring of Paths. ***Discrete Applied Mathematics***, 164:450–459, 2014.

[17] K. R. P. S. Lima. ***Recoloração Convexa de Caminhos***. PhD thesis, Instituto de Matemática e Estatística, Universidade de São Paulo, 2011.

[18] S. Moran and S. Snir. Efficient Approximation of Convex Recolorings. ***Journal of Computer and System Sciences***, 73(7):1078–1089, 2007.

[19] S. Moran and S. Snir. Convex Recolorings of Strings and Trees: Definitions, Hardness Results and Algorithms. ***Journal of Computer and System Sciences***, 74(5):850–869, 2008.

[20] S. Moran, S. Snir, and W.-K. Sung. Partial Convex Recolorings of Trees and Galled Networks: Tight Upper and Lower Bounds. ***ACM Transactions on Algorithms***, 7(4):42, 2011.

[21] P. F. S. Moura. Recoloração Convexa de Grafos: Algoritmos e Poliedros. Master's thesis, Instituto de Matemática e Estatística, Universidade de São Paulo, 2013.

[22] P. F. S. Moura. ***Graph Colorings and Digraph Subdivisions***. PhD thesis, Instituto de Matemática e Estatística, Universidade de São Paulo, 2017.

[23] M. G. Resende and C. C. Ribeiro. ***Greedy Randomized Adaptive Search Procedures: Advances, Hybridizations, and Applications***, chapter 10, pages 283–319. International Series in Operations Research & Management Science. Springer US, Boston, MA, 2nd edition, 2010.