



## **SOBRE MODELOS E ALGORITMOS PARA COLORAÇÃO EM GRAFOS COM RESTRIÇÕES DE DISTÂNCIA**

**Bruno Dias, Rosiane de Freitas**

Instituto de Computação - Universidade Federal do Amazonas  
Manaus/AM - Brazil  
{bruno.dias, rosiane}@icomp.ufam.edu.br

**Nelson Maculan, Jayme Szwarcfiter**

PESC/COPPE - Universidade Federal do Rio de Janeiro  
Rio de Janeiro/RJ - Brazil  
{maculan, jayme}@cos.ufrj.br

**Philippe Michelon**

Centre d'Enseignement et de Recherche en Informatique, Université d'Avignon et des Pays de  
Vaucluse  
Avignon - France  
philippe.michelon@univ-avignon.fr

### **RESUMO**

Colorações com distâncias consistem em variações do problema de coloração de vértices em grafos com restrições adicionais de adjacência envolvendo distâncias, as quais são representadas por números naturais. Neste trabalho, são abordados algumas destas variações, onde é apresentada uma síntese dos resultados obtidos para as mesmas, incluindo propriedades envolvendo factibilidade e complexidade dos problemas, além de modelos de programação inteira por restrições e estratégias algorítmicas baseadas em métodos aplicados em outros problemas de geometria de distâncias relacionados.

**PALAVRAS CHAVE.** Colorações em grafos. Geometria de distâncias. Programação matemática.

**Tópicos:** Otimização combinatória. Teoria dos grafos.

### **ABSTRACT**

Distance colorings consist of variations of the classic vertex coloring problem, where additional adjacency constraints involving distances, which are represented by natural numbers, are inserted. In this work, we study some of these variations and present a summary of results obtained for them, including factibility and complexity properties, integer and constraint programming models and algorithmic strategies based on methods applied to other related distance geometry problems.

**KEYWORDS.** Graph coloring. Distance geometry. Mathematical programming.

**Paper topics:** Combinatorial optimization. Graph theory.



## 1. Introdução

Seja  $G = (V, E)$  um grafo, onde  $V$  é seu conjunto de vértices e  $E$  é o de arestas. No problema clássico de coloração de vértices em grafos, deve-se encontrar um mapeamento  $x : V \rightarrow [1 .. k]$  tal que  $\forall (i, j) \in A, x(i) \neq x(j)$ , onde o valor  $x(i)$  representa a cor atribuída ao vértice  $i$  e  $k$  deve ser o menor valor possível (onde pode-se definir  $k = \max_{i \in V} x(i)$ ). Diz-se então que  $x$  é uma  $k$ -coloração. A obtenção do menor valor possível para  $k$  pode ser vista como uma minimização da utilização de recursos (cores) em suas diversas aplicações, como alocação de canais em redes sem fio, escalonamento de tarefas, alocação de registradores em compiladores, dentre outras.

Existem diversas variações de colorações em grafos na literatura, sendo a T-coloração uma de bastante importância, originalmente proposta para aplicações de alocação de canais em redes sem fio. Nela, tem-se o conceito de conjuntos proibidos (onde, para cada aresta, existe um conjunto  $T_{i,j}$ ), de forma que uma coloração válida passa a ser um mapeamento  $x : V \rightarrow [1, k]$ , onde  $\forall (i, j) \in A, |x(i) - x(j)| \notin T_{i,j}$  [Hale, 1980]. Destacam-se dois casos específicos da T-coloração:

1. Para cada aresta  $(i, j) \in E$ , tem-se  $T_{i,j} = \mathbb{Z}_{\geq 0} \setminus \{d_{i,j}\}$ , onde  $\forall (i, j) \in E, d_{i,j} \in \mathbb{N}$ .
2. Para cada aresta  $(i, j) \in E$ , tem-se  $T_{i,j} = \{0, 1, \dots, d_{i,j} - 1\}$ , onde  $\forall (i, j) \in E, d_{i,j} \in \mathbb{N}$ .

Ambos os cenários acima consistem em colorações com restrições de distâncias, onde o valor  $d_{i,j}$  representa a distância a ser respeitada entre as cores atribuídas aos vértices  $i$  e  $j$ , sendo que cada caso possui um diferente tipo de restrição envolvendo tal distância na aresta. Tais casos específicos são o objeto de estudo deste trabalho.

O restante do artigo é organizado como se segue. Na Seção 2 são dadas as definições formais de ambos os casos específicos e mostrada a visão de geometria de distâncias para modelagem dos mesmos, além de propriedades envolvendo a complexidade computacional e factibilidade dos problemas. A Seção 3 mostra modelos de programação por restrições e inteira para solução dos problemas. Na Seção 4, são enunciados algoritmos para as colorações com distâncias, baseando-se em métodos para outros problemas em geometria de distâncias. Por fim, a Seção 5 dá as considerações finais do trabalho, indicando trabalhos futuros e em andamento.

## 2. Colorações com distâncias

Nesta seção, serão definidos independentemente os dois casos específicos da T-coloração citados na Seção 1. O primeiro deles consiste em conjuntos proibidos no formato  $T_{i,j} = \mathbb{Z}_{\geq 0} \setminus \{d_{i,j}\}$ , onde  $d_{i,j}$  é um número natural. Tal restrição exige que a diferença absoluta entre as cores dos vértices  $i$  e  $j$  seja exatamente o valor  $d_{i,j}$ , o que equivale à restrição  $|x(i) - x(j)| = d_{i,j}$ . Assim, tem-se o seguinte problema de coloração com distâncias.

### **Definição 1. Problema de Coloração Mínima em Geometria de Distâncias Iguais (MinEQ-CDGP):**

*Dado um grafo simples ponderado não direcionado  $G = (V, E)$  e uma função de distâncias  $d : E \rightarrow \mathbb{N}$ , encontrar uma projeção  $x : V \rightarrow \mathbb{N}$  tal que  $\forall (i, j) \in E, |x(i) - x(j)| = d_{i,j}$  e a extensão da projeção, definida por  $S = \max_{i \in V} x(i)$ , seja a menor possível.*

Tal problema é modelado utilizando geometria de distâncias envolvendo o posicionamento dos vértices do grafo de entrada em pontos de cores na reta numérica ( $\mathbb{R}^1$ ), mas considerando apenas números naturais (ou seja,  $x : V \rightarrow \mathbb{N}$ ). Cada aresta do grafo define uma 0-esfera (que consiste em um segmento de reta) e uma solução para a coloração com distâncias consiste em interseções de tais 0-esferas, como pode ser visto na Figura 1.

O segundo caso específico da T-coloração ocorre com conjuntos proibidos  $T_{i,j}$  seguindo números consecutivos a partir de 0 até um número natural  $d_{i,j}$ , ou seja,  $T_{i,j} = \{0, 1, \dots, d_{i,j} - 1\}$ . Este cenário requer que a diferença absoluta entre as cores dos vértices  $i$  e  $j$  seja qualquer valor maior ou igual a  $d_{i,j}$ , o que equivale a  $|x(i) - x(j)| \geq d_{i,j}$ . Define-se então o seguinte problema:

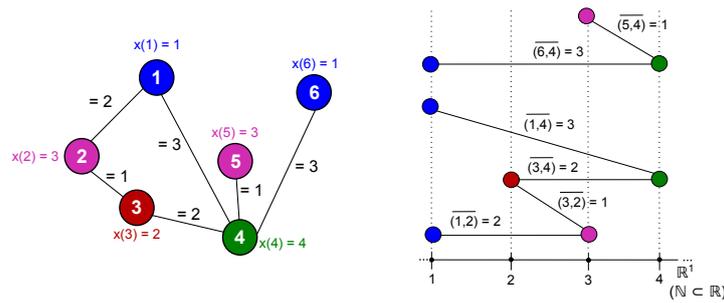


Figura 1: Exemplo de instância do MinEQ-CDGP e sua solução vista como intersecção de 0-esferas.

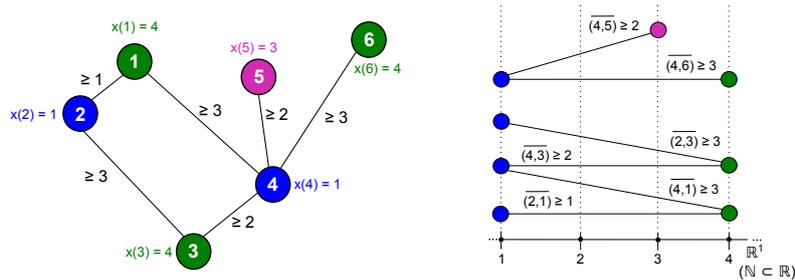


Figura 2: Exemplo de instância do MinGEQ-CDGP e sua solução vista como intersecção de 0-esferas.

**Definição 2. Problema de Coloração Mínima em Geometria de Distâncias Maiores ou Iguais (MinGEQ-CDGP):** Dado um grafo simples ponderado não direcionado  $G = (V, E)$  e uma função de distâncias  $d : E \rightarrow \mathbb{N}$ , encontrar uma projeção  $x : V \rightarrow \mathbb{N}$  tal que  $\forall (i, j) \in E, |x(i) - x(j)| \geq d_{i,j}$  e a extensão da projeção seja a menor possível.

O MinGEQ-CDGP é equivalente ao problema de coloração com largura de banda (*Bandwidth Coloring Problem - BCP*) [Lai e Lü, 2013], que por sua vez é equivalente ao problema de alocação de canais com minimização de extensão (*Minimum Span Channel Assignment Problem - MS-CAP*) [Koster, 1999; Audhya et al., 2011]. Um exemplo de instância resolvida do MinGEQ-CDGP e sua representação como intersecção de 0-esferas é mostrada na Figura 2.

### Casos especiais

Para os problemas previamente definidos, pode-se identificar alguns casos especiais. O primeiro é para o MinEQ-CDGP, onde pode-se definir um problema no qual todas as distâncias impostas nas arestas são iguais, como enunciado abaixo.

**Definição 3. Problema de Coloração Mínima em Geometria de Distâncias Iguais Uniformes (MinEQ-CDGP-Unif):** Dado um grafo simples ponderado não direcionado  $G = (V, E)$  e um número natural  $\varphi$ , encontrar uma projeção  $x : V \rightarrow \mathbb{N}$  tal que  $\forall (i, j) \in E, |x(i) - x(j)| = \varphi$  e a extensão da projeção, definida por  $S = \max_{i \in V} x(i)$ , seja a menor possível.

Nesse modelo, o grafo de entrada é definido pelos conjuntos de vértices e arestas, além do valor  $\varphi$ , que substitui o conjunto de distâncias para cada aresta. Um caso similar existe para o MinGEQ-CDGP, como definido abaixo.

**Definição 4. Problema de Coloração Mínima em Geometria de Distâncias Maiores ou Iguais Uniformes (MinGEQ-CDGP-Unif):** Dado um grafo simples ponderado não direcionado  $G = (V, E)$  e um número natural  $\varphi$ , encontrar uma projeção  $x : V \rightarrow \mathbb{N}$  tal que  $\forall (i, j) \in E, |x(i) - x(j)| \geq \varphi$  e a extensão da projeção, definida por  $S = \max_{i \in V} x(i)$ , seja a menor possível.

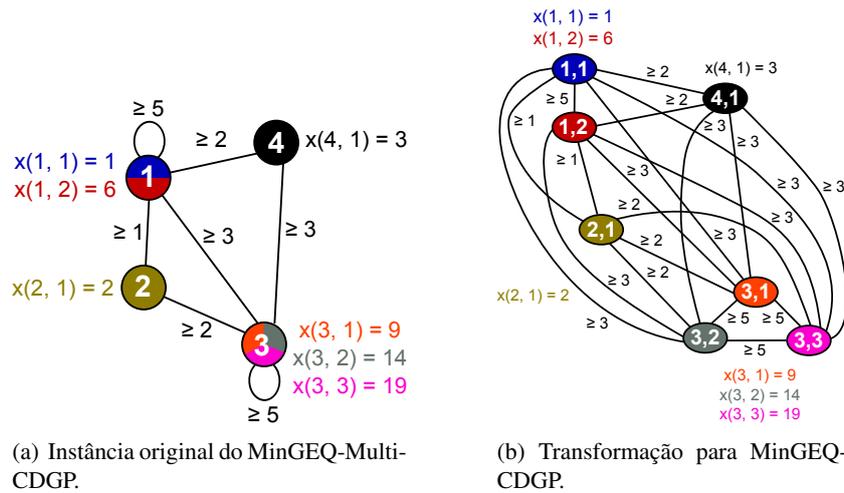


Figura 3: Exemplo de instância do MinGEQ-Multi-CDGP e sua transformação para MinGEQ-CDGP.

Uma solução para o problema de coloração clássica induzida pelo grafo  $G$  pode ser transformada em solução para o MinGEQ-CDGP-Unif definido por  $G$  e  $\varphi$  definindo-se  $x(i) = (\varphi \times (c(i) - 1)) + 1$ , onde  $c(i)$  é a cor atribuída ao vértice  $i$  na solução do problema de coloração clássica. Devido a isso, o MinGEQ-CDGP-Unif é equivalente à coloração clássica.

Seguindo outra direção, pode-se generalizar o MinGEQ-CDGP para grafos com demandas de multicoloração, isto é, onde cada vértice pode exigir mais de uma cor, sendo todas distintas. Nas colorações com distâncias, pode haver uma restrição de distância também entre as cores de um mesmo vértice. Tal generalização é definida abaixo.

**Definição 5. Problema de Multicoloração Mínima em Geometria de Distâncias Maiores ou Iguais - MinGEQ-Multi-CDGP:** Dado um grafo simples ponderado não direcionado  $G = (V, E)$ , uma função de distâncias  $d : (E \cup \{(i, i) : i \in V\}) \rightarrow \mathbb{N}$  e uma função de demandas  $q : V \rightarrow \mathbb{N}$ , encontrar uma projeção  $x : V \rightarrow \mathcal{P}(\mathbb{N})$  tal que, sendo  $x(i, k)$  a  $k$ -ésima cor atribuída a  $i$ , tenha-se  $\forall (i, j) \in E, k \in [1 .. q_i], m \in [1 .. q_j]$  que  $|x(i, k) - x(j, m)| \geq d_{i,j}$ , e  $\forall i \in V, k, m \in [1 .. q_i]$  que  $|x(i, k) - x(i, l)| \geq d_{i,i}$ , onde a extensão da projeção, definida por  $S = \max_{i \in V, 1 \leq k \leq q_i} x(i)$ , seja a menor possível.

Como ocorre com todos os problemas de multicoloração, os problemas MinGEQ-CDGP e MinGEQ-Multi-CDGP são equivalentes. Toda instância deste último pode ser transformada em uma do primeiro replicando-se cada vértice  $i$  em uma clique de  $q_i$  subvértices. Cada aresta de uma destas cliques tem uma distância igual ao valor  $d_{i,i}$  da instância original do MinGEQ-Multi-CDGP, e cada subvértice da clique é adjacente aos subvértices das cliques dos outros vértices adjacentes a  $i$  no grafo original, como exemplificado na Figura 3. Desse modo, qualquer algoritmo para o MinGEQ-CDGP pode ser usado para resolver o MinGEQ-Multi-CDGP, porém, em alguns casos, a explosão combinatória pode se tornar intratável sem o uso de técnicas que explorem a multicoloração.

Ressalta-se que não foi definido um modelo aplicando multicoloração ao MinEQ-CDGP. Isso ocorre pois tal transformação de cada vértice  $i$  em cliques de  $q_i$  subvértices esbarra em uma propriedade do MinEQ-CDGP que indica a infactibilidade de algumas classes de grafos, como será visto na subseção a seguir.

### Factibilidade e complexidade

Algumas propriedades sobre a factibilidade e complexidade computacional podem ser enunciadas para algumas classes de grafos em relação às colorações abordadas neste artigo. Nota-se que toda instância dos problemas MinGEQ-CDGP (e suas variações mostradas) sempre é factível



(ou seja, sempre admite ao menos uma solução válida). Isso ocorre devido ao fato de que, em termos da T-coloração, como o conjunto proibido destes problemas é finito (como visto anteriormente, para cada aresta  $(i, j) \in E$ , tal conjunto é definido como  $T_{i,j} = \{0, 1, \dots, d_{i,j} - 1\}$ ), o complemento  $\overline{T_{i,j}}$  do mesmo (que seria o conjunto de valores permitidos para a diferença absoluta entre as cores dos vértices da aresta  $(i, j)$ , ou seja,  $|x(i) - x(j)| \in \overline{T_{i,j}}$ ) é infinito, o que garante que sempre haverá como colorir os vértices, mesmo que seja necessário que alguns usem cores de valores altos.

No entanto, esse não é o caso para o MinEQ-CDGP e suas variações, já que os conjuntos proibidos são definidos como  $T_{i,j} = \mathbb{Z}_{\geq 0} \setminus \{d_{i,j}\}$ , ou seja, são infinitos, o que torna seus complementos finitos, restringindo as formas de que os vértices podem ser coloridos sem violar a distância. Desse modo, nem todas as instâncias do MinEQ-CDGP serão factíveis, como será visto a seguir em relação a algumas classes de grafos. O primeiro teorema diz respeito ao caso especial MinEQ-CDGP-Unif, como descrito abaixo.

**Teorema 1.** *Um grafo  $G$  admite ao menos uma solução factível para o MinEQ-CDGP-Unif se, e somente se,  $G$  for bipartido.*

Como resultado complementar do Teorema 1, tem-se que grafos contendo ciclos ímpares nunca admitirão solução factível para o MinEQ-CDGP-Unif. Devido a isso, tem-se que todo grafo completo  $K_n$  com  $n \geq 3$  não admite solução factível para este problema. Por conta disso, não se define um modelo combinando o mesmo com multicoloração, já que se ao menos um vértice  $i$  tiver uma demanda de cores  $q_i \geq 3$ , automaticamente a instância não terá solução, uma vez que, aplicando-se a transformação para um problema de demandas unitárias vista na subseção anterior, a clique formada pelos subvértices de  $i$  formará uma sub-instância do MinEQ-CDGP-Unif contendo ciclo ímpar, que é infactível.

Os próximos dois teoremas são relacionados ao caso mais geral, com distâncias arbitrárias. Assim, o segundo teorema abordado nesta subseção é dado abaixo.

**Teorema 2.** *Seja  $G = (V, E)$  uma árvore e  $d : E \rightarrow \mathbb{N}$  uma função de distâncias. Então, a instância do MinEQ-CDGP definida por  $G$  e  $d$  sempre admite ao menos uma solução factível.*

O terceiro teorema é baseado na prova da NP-completude do problema de *1-Embedding*, que é equivalente a encontrar uma solução factível do MinEQ-CDGP, que foi descrita por Saxe [1979] por meio de uma redução do problema de particionamento de conjuntos. Tal teorema é dado abaixo.

**Teorema 3.** *Seja  $G = (V, E)$  um ciclo e  $d : E \rightarrow \mathbb{N}$  uma função de distâncias. Então, a instância do MinEQ-CDGP definida por  $G$  e  $d$  sempre admite ao menos uma solução factível se o conjunto  $D$  de todas as distâncias  $\left( D = \bigcup_{(i,j) \in E} \{d_{i,j}\} \right)$  puder ser particionado em dois subconjuntos disjuntos  $D_1$  e  $D_2$  tais que  $\sum D_1 = \sum D_2$*

As provas detalhadas dos teoremas mostrados nesta seção podem ser vistas no artigo de de Freitas et al. [2016]. Um sumário de todos os modelos de colorações com distâncias abordados é dado na Tabela.

### 3. Modelos de programação inteira e por restrições

Os modelos de geometria de distâncias para problemas de coloração em grafos podem ser mapeados diretamente para uma formulação de programação por restrições (*constraint programming* - CP), já que tais modelos envolvem a projeção do grafo em uma dimensão, ou seja, uma rotulação dos vértices do grafo usando números naturais que indicam sua posição na linha, de tal modo que as distâncias dos segmentos definidos pelos pesos das arestas sejam respeitadas. Neste aspecto, a técnica de programação por restrições se mostra capaz de tratar diretamente estas características. Modelos tradicionais de programação linear inteira serão abordados, de forma que uma comparação entre as modelagens será feita para os problemas mostrados.



Tabela 1: Sumário dos modelos de colorações com distâncias.

Problema	Tipo de restrição	Tipo de distância
MinEQ-CDGP	$\forall (i, j) \in E \quad  x(i) - x(j)  = d_{i,j}$	$\forall (i, j) \in E, d_{i,j} \in \mathbb{N}$
MinGEQ-CDGP	$\forall (i, j) \in E \quad  x(i) - x(j)  \geq d_{i,j}$	$\forall (i, j) \in E, d_{i,j} \in \mathbb{N}$
MinEQ-CDGP-Unif	$\forall (i, j) \in E \quad  x(i) - x(j)  = d_{i,j}$	$\forall (i, j) \in E, d_{i,j} = \varphi$ ( $\varphi \in \mathbb{N}$ )
MinGEQ-CDGP-Unif	$\forall (i, j) \in E \quad  x(i) - x(j)  \geq d_{i,j}$	$\forall (i, j) \in E, d_{i,j} = \varphi$ ( $\varphi \in \mathbb{N}$ )
MinGEQ-Multi-CDGP	$\forall (i, j) \in (E \cup \{(i, i) : i \in V\}),$ $k \in [1 .. q_i], m \in [1 .. q_j]$ $ x(i, k) - x(j, m)  \geq d_{i,j}$	$\forall (i, j) \in E, d_{i,j} \in \mathbb{N}$ $\forall i \in V, d_{i,i} \in \mathbb{N}$

Para o restante desta seção, será assumido que  $\boxtimes$  é um operador relacional, de tal modo que, para dois vértices  $i$  e  $j$ , tem-se a restrição  $|x(i) - x(j)| \boxtimes d_{i,j}$ , onde  $\boxtimes$  é  $=$  para o MinEQ-CDGP e  $\geq$  para o MinGEQ-CDGP.

A primeira formulação utiliza em programação por restrições e é baseada diretamente nas definições mostradas na Seção 2. Seja  $x_i$  uma variável inteira consistindo na cor atribuída ao vértice  $i$ . Então, o modelo, denotado por CDGP-CP, é definido por:

$$\text{Minimizar} \quad \max_{i \in V} x(i) \quad (1)$$

$$\text{Sujeito a to} \quad |x(i) - x(j)| \boxtimes d_{i,j} \quad (\forall (i, j) \in E) \quad (2)$$

$$x(i) \in \mathbb{Z}^* \quad (\forall i \in V) \quad (3)$$

A função objetivo (1) consiste na minimização da maior cor usada dentre todos os vértices. O conjunto de restrições (2) envolve as distâncias entre vértices. Para cada variável  $x_i$ , o domínio inicial  $D(x_i)$  consiste em todos os inteiros entre 1 e um dado limite superior  $U$ , ou seja,  $D(x_i) = [1 .. U]$ . Ressalta-se que todos os domínios iniciais de todas as variáveis são iguais. Este modelo possui  $O(|V|)$  variáveis (uma por vértice) e  $O(|E|)$  restrições (uma por aresta).

Para o MinGEQ-CDGP-Unif, ou seja, onde todas as distâncias são iguais e as restrições são do tipo  $\geq$ , (MinEQ-CDGP-Unif e MinGEQ-CDGP-Unif), pode-se usar uma restrição global específica. Tal modelo, que denotado por MinGEQ-CDGP-Unif-CP, é definido por:

$$\text{Minimizar} \quad \max_{i \in V} x(i) \quad (4)$$

$$\text{Sujeito a} \quad \text{allMinDistance}(\{x(j) : (i, j) \in E\}, \beta) \quad (\forall i \in V) \quad (5)$$

$$x(i) \in \mathbb{Z}^* \quad (\forall i \in V) \quad (6)$$

A restrição global *allMinDistance* usa, como parâmetros, um conjunto de variáveis e uma constante  $w$  e faz com que, para todos os pares de variáveis  $y$  e  $z$  no conjunto, a condição  $|y - z| \geq w$  seja respeitada, o que é exigido pelo MinGEQ-CDGP. Esta formulação tem  $O(|V|)$  variáveis e  $O(|V|)$  restrições. Como, para a maioria dos grafos conectados, tem-se que  $|V| \leq |E|$ , a formulação MinGEQ-CDGP-Unif-CP tem menos restrições e pode tirar proveito de propagadores especializados na busca.

Para o MinGEQ-Multi-CDGP, que usa multicoloração, uma formulação de programação por restrições pode ser construída com base em características de ambos os modelos anteriores. Como abordado na seção 2, um problema com multicoloração pode ser convertido em uma versão



com demandas unitárias transformando-se cada vértice  $i$  em uma clique de  $q_i$  vértices, cada um adjacente aos outros que eram originalmente adjacentes a  $i$ . Com base nisto, tem-se que cada vértice consiste em uma pequena sub-instância do MinGEQ-CDGP-Unif, onde o grafo maior (ou seja, considerando as restrições impostas nas arestas originais entre vértices diferentes) é uma instância do MinGEQ-CDGP se as demandas de multicoloração forem ignoradas. Assim, tem-se o seguinte modelo de programação por restrições, denotado por MinGEQ-Multi-CDGP-CP:

$$\text{Minimizar} \quad \max_{\substack{i \in V \\ 1 \leq k \leq q_i}} x(i, k) \quad (7)$$

$$\text{Sujeito a} \quad |x(i, k) - x(j, m)| \geq d_{i,j} \quad (\forall (i, j) \in E, 1 \leq k \leq q_i, 1 \leq m \leq q_j) \quad (8)$$

$$\text{allMinDistance}(\{x(i, k) : 1 \leq k \leq q_i\}, d_{i,i}) \quad (\forall i \in V) \quad (9)$$

$$x(i, k) \in \mathbb{Z}^* \quad (\forall i \in V, 1 \leq k \leq q_i) \quad (10)$$

Na formulação acima, restrições (8) garantem que as cores atribuídas a vértices diferentes respeitem as distâncias das arestas. As restrições (8) fazem com que as cores distintas atribuídas a um mesmo vértice  $i$  respeitem a distância mínima  $d_{i,i}$  entre elas (por meio da restrição global *allMinDistance*, já que a clique do vértice define uma pequena sub-instância do MinGEQ-CDGP-Unif). Esta formulação possui  $O(|V| \times q_{max})$  variáveis e  $O(|E| \times q_{max})$  restrições, onde  $q_{max} = \max_{i \in V} q_i$ .

A segunda abordagem utilizada nesta seção envolve programação linear inteira. Um modelo para as colorações com distâncias, denotado por CDGP-IP, é baseado nas formulações definidas em Koster [1999], que foram modificadas para se tornarem mais compactas e acomodarem os dois tipos de restrições ( $=$  e  $\geq$ ). Dois conjuntos de variáveis são usadas:

$$\bullet \quad x_{ic} = \begin{cases} 1 & \text{se a cor } c \text{ for atribuída ao vértice } i; \\ 0 & \text{caso contrário.} \end{cases}$$

$$\bullet \quad z_{max} = \text{maior cor usada na solução..}$$

Com estas variáveis, define-se o modelo CDGP-IP como se segue:

$$\text{Minimizar} \quad z_{max} \quad (11)$$

$$\text{Sujeito a} \quad \sum_{c=1}^U x_{ic} = 1 \quad (\forall i \in V) \quad (12)$$

$$x_{ic} + x_{je} \leq 1 \quad (\forall (i, j) \in E; 1 \leq c, e \leq U : \neg(|c - e| \boxtimes d_{i,j})) \quad (13)$$

$$z_{max} \geq cx_{ic} \quad (\forall i \in V; 1 \leq c \leq U) \quad (14)$$

$$x_{ic} \in \{0, 1\} \quad (\forall i \in V; 1 \leq c \leq U) \quad (15)$$

$$(16)$$

Na formulação CDGP-IP, a função objetivo é (11) o valor da variável  $z_{max}$ , que deverá ser minimizado. O conjunto de restrições (12) faz com que todos os vértices recebam uma cor. Restrições (13) garantem que os vértices só possam receber cores que não violem a restrição de distância do problema considerado (MinEQ-CDGP ou MinGEQ-CDGP). As restrições (14) garantem que a variável  $z_{max}$  tenha o valor da maior cor usada. Por fim, o conjunto (15) define que as variáveis  $x$  serão binárias. De maneira similar aos domínios das variáveis nos modelos de



programação por restrições, as variáveis do conjunto  $x$  do CDGP-IP são indexadas por vértice e cor, o que torna necessário definir um limite superior  $U$  para o valor da maior cor a ser usada, de forma que o conjunto de cores seja  $[1 .. U]$ . Este modelo de programação linear inteira tem  $O(U \times |V|)$  variáveis (uma para cada par de cor e vértice) e  $O(U \times (|E| + |V|))$  restrições, de forma que o CDGP-IP tem dimensão pseudopolinomial.

Para o MinGEQ-Multi-CDGP, a formulação de programação linear inteira também pode ser modificada para tratar das demandas de multicoloração. O novo modelo, denotado por MinGEQ-Multi-CDGP-IP, deriva de duas modificações feitas na formulação CDGP-IP. A primeira consiste em alterar o lado direito das restrições (12) de 1 para  $q_i$ , o que fará com que, para cada vértice  $i$ , exatamente  $q_i$  variáveis  $x_{ic}$  tenham valor 1, o que significa que o vértice terá  $q_i$  cores distintas. A segunda modificação consiste em expandir o conjunto de restrições (13) de modo a adicionar novos elementos que garantirão que cada vértice  $i$  não use cores distintas que violem a distância  $d_{i,i}$ , ou seja, além de uma restrição para cada  $(i, j) \in E; 1 \leq c, e \leq U$  tal que  $\neg(|c - e| \geq d_{i,j})$  e também uma para cada  $i \in V; 1 \leq c, e \leq U$  tal que  $\neg(|c - e| \geq d_{i,i})$ . Esta modificação equivale a adicionar uma aresta  $(i, i)$  no grafo de entrada, cujo peso associado será igual a  $d_{i,i}$ , de tal modo que aplicar a formulação CDGP-IP ao grafo resultante desta modificação automaticamente adicionaria as novas restrições. As duas modificações citadas não tem impacto no tamanho assintótico da formulação. O modelo resultante é, então, definido por:

$$\text{Minimizar } z_{max} \tag{17}$$

$$\text{Sujeito a } \sum_{c=1}^U x_{ci} = q_i \quad (\forall i \in V) \tag{18}$$

$$x_{ic} + x_{je} \leq 1 \quad (\forall (i, j) \in E; 1 \leq c, e \leq U : \neg(|c - e| \geq d_{i,j})) \tag{19}$$

$$x_{ic} + x_{ie} \leq 1 \quad (\forall i \in V; 1 \leq c, e \leq U | \neg(|c - e| \geq d_{i,i})) \tag{20}$$

$$z_{max} \geq cx_{ic} \quad (\forall i \in V; 1 \leq c \leq U) \tag{21}$$

$$x_{ci} \in \{0, 1\} \quad (\forall i \in V; 1 \leq c \leq U) \tag{22}$$

$$\tag{23}$$

A Tabela 2 apresenta um sumário das diferentes formulações apresentadas nesta seção.

Tabela 2: Sumário dos modelos de programação inteira e por restrições para as colorações com distâncias.

Formulação	Tipo	Problema(s)	# Var.	# Restr.
CDGP-CP	Programação por Restrições	MinEQ-CDGP e MinGEQ-CDGP	$O( V )$	$O( E )$
CDGP-Unif-CP		MinEQ-CDGP-Unif e MinGEQ-CDGP-Unif	$O( V )$	$O( V )$
MinGEQ-Multi-CDGP-CP		MinGEQ-CDGP-Multi	$O( V  \times q_{max})$	$O( E  \times q_{max})$
CDGP-IP	Prog. Inteira	MinEQ-CDGP e MinGEQ-CDGP	$O(U \times  V )$	$O(U \times ( E  +  V ))$
MinGEQ-Multi-CDGP-IP	Prog. Inteira	MinGEQ-CDGP-Multi	$O(U \times  V )$	$O(U \times ( E  +  V ))$

### Limites superiores para os conjuntos de cores

Todas as formulações mostradas anteriormente precisam de um conjunto finito de cores previamente definido, consistindo em um intervalo de inteiros  $[1 .. U]$ , onde  $U$  é um limite superior para a maior cor usada. Um valor trivial para  $U$  pode ser calculado pela soma de todas as distâncias impostas nas arestas mais um, ou seja,  $U = \sum_{(i,j) \in E} d_{i,j} + 1$ . No entanto, este limite é muito fraco, pois ao efetuar-se a soma de todas as distâncias, está sendo ignorado o fato de que as cores podem



---

**Algoritmo 1** Heurística gulosa para geração de soluções iniciais para MinGEQ-CDGP e MinGEQ-CDGP-Multi.

---

**Require:** grafo  $G = (V, E)$ , distâncias  $d : E \cup \{(i, i) : i \in V\} \rightarrow \mathbb{N}$ , demandas de cores  $q : V \rightarrow \mathbb{N}$ .

```
1: função SOLUCAOINICIALGULOSA( $G = (V, E), d, q$ )
2:    $V' \leftarrow V$ 
3:   para cada  $i \in V$  faça
4:      $numCores[i] \leftarrow 0$ 
5:      $coresAtrib[i] \leftarrow \emptyset$ 
6:   enquanto  $V' \neq \emptyset$  faça
7:      $i \leftarrow \arg \max_{v \in V'} q_v$ 
8:      $corCandidata \leftarrow (numCores[i] \times d_{i,i}) + 1$ 
9:      $violada \leftarrow$  verdadeiro
10:    enquanto  $violada =$  verdadeiro faça
11:       $violada \leftarrow$  falso
12:      para cada  $j \in V - (V' \cup \{i\})$  faça
13:        para cada  $k \in coresAtrib[j]$  faça
14:          se  $|k - corCandidata| < d_{i,j}$  então
15:             $violada \leftarrow$  verdadeiro
16:             $corCandidata \leftarrow corCandidata + 1$ 
17:          interrompa
18:        se  $violada =$  verdadeiro então
19:          interrompa
20:         $coresAtrib[i] \leftarrow coresAtrib[i] \cup \{candColor\}$ 
21:         $V' \leftarrow V' - \{i\}$ 
22:    retorne  $coresAtrib$ 
```

---

ser reutilizadas por vértices que não sejam adjacentes uns aos outros. Devido a isso, tal limite superior é muito distante do valor ótimo da maior cor usada, o que também faz com que o conjunto de cores tenha uma cardinalidade muito alta, levando a um grande impacto no desempenho computacional dos modelos, em especial o de programação inteira, já que tanto o número de variáveis quanto o de restrições são proporcionais ao limite superior, o que pode induzir a cenários de falta de memória.

Para o MinGEQ-CDGP e MinGEQ-CDGP-Multi, um valor melhor para  $U$  pode ser obtido por um pré-processamento do grafo de entrada, onde aplica-se uma heurística que não necessite de um limite superior explícito. O valor da maior cor usada na solução resultante é, então, usado como valor para  $U$ , que será usado para a montagem dos modelos de programação inteira e por restrições. Uma heurística gulosa pode ser usada para isso, onde os vértices são processados de acordo com suas demandas de cores - vértices com maiores demandas são coloridos primeiro. A cor de um vértice  $i$  é determinada considerando-se, primeiramente, a cor  $(numCores[i] \times d_{i,i}) + 1$  (onde  $numCores[i]$  é o número de cores que já foram atribuídas a  $i$ ) e, em seguida, verificando-se se a mesma viola alguma restrição de distância em relação aos vértices vizinhos. Se ocorrer alguma violação, a cor candidata é incrementada em 1 e o processo de verificação é repetido até que uma cor factível seja encontrada. Tal cor é, então atribuída a  $i$  (e  $numCores[i]$  é incrementado em 1) e, caso sua demanda ainda não tenha sido completamente atendida, uma cor adicional é calculada e atribuída ao vértice. Todo este procedimento é repetido até que o grafo seja totalmente colorido. O pseudocódigo desta heurística é dado no Algoritmo 1.

Resultados obtidos em experimentos computacionais com os modelos de programação



inteira e por restrições são apresentados no trabalho de Dias et al. [2016].

#### 4. *Branch-prune-and-bound*

Um algoritmo *branch-and-prune* (BP) foi proposto por Lavor et al. [2012] para o Problema de Geometria de Distâncias Moleculares Discretizáveis (*Discretizable Molecular Distance Geometry Problem - DMDGP*), que é um caso especial do DGP. No DMDGP, os vértices do grafo são ordenados de tal modo que cada quatro vértices consecutivos induzem cliques ( $\forall i \in \{4, \dots, n\} \forall j, k \in \{i-3, \dots, i\} (\{j, k\} \in E)$ ). Além disso, há uma desigualdade triangular estrita nas distâncias entre vértices consecutivos ( $\forall i \in \{2, \dots, n-1\} d_{i-1, i+1} < d_{i-1, i} + d_{i, i+1}$ ). A posição de um vértice  $i$  é determinada usando, como base, as posições dos três vértices anteriores, de modo que a intersecção de três esferas de raios  $d_{i-3, i}$ ,  $d_{i-2, i}$  e  $d_{i-1, i}$  fornecem duas possíveis posições para  $i$ . Porém, a distância entre  $i$  e algum vértice anterior (na ordenação) aos outros três que serviram de base pode ter sido violada, o que exige testes de factibilidade para garantir que a solução sendo construída é válida. Em Lavor et al. [2012], é usado o teste de Factibilidade Direta de Distância (*Direct Distance Feasibility - DDF*), onde a condição  $\forall (i, j) \in E \ ||x(i) - x(j)| - d_{ij}| < \epsilon$  deve ser certificada (sendo  $\epsilon$  uma dada tolerância).

Para os problemas de colorações com distâncias, pode-se adaptar tais conceitos de modo a obter um método de resolução dos mesmos. Uma das primeiras observações a serem feitas é que, nas colorações com distâncias, não há suposições iniciais a serem respeitadas, portanto, não há uma ordenação explícita de vértices a ser considerada. Deste modo, tal ordenação é construída seguindo um processo de enumeração implícita. Para obter não só uma solução factível (como no BP), mas sim a ótima, conceitos do método *branch-and-bound* (BB) também podem ser usados, de modo que soluções parciais (ou seja, ordenações de vértices já coloridos) que não possam levar a uma melhor solução completa sejam descartadas. Denota-se o método resultante da combinação de BP e BB de *branch-prune-and-bound* (BPB).

O funcionamento básico do BPB será descrito a seguir. Ressalta-se que o mesmo será mostrado para problemas em grafos com demandas unitárias, uma vez que, como mostrado na Seção 2, existe equivalência entre tais problemas e os que envolvem multicoloração.

O primeiro passo do BPB envolve selecionar um vértice  $i$  que ainda não tenha sido colorido. O mesmo recebe a cor 1, que é a menor disponível (já que as cores são números naturais). Então, um vizinho  $j$  de  $i$  que não tenha sido colorido é selecionado, que será considerado como vértice atual (e  $i$  o anterior). Aplica-se um algoritmo de seleção de cor em  $j$  e o processo é repetido recursivamente nos vizinhos de  $j$  que não tenham sido coloridos. Quando um vizinho sem cor do vértice atual não puder ser determinado, então o próximo a ser colorido será um vértice qualquer do grafo, sem a exigência de ser vizinho do atual, mas a cor do mesmo será determinada de acordo com o primeiro vizinho que já foi colorido na ordenação sendo construída, caso haja algum. O pseudocódigo deste procedimento geral é mostrado no Algoritmo 2.

Existem duas possibilidades para a seleção de cores de um vértice, ou seja, definições para a função SELECAOCOR(), que retorna um conjunto de cores possíveis para o vértice. A primeira, denotada por BPB-Prev, é baseada no algoritmo BP original Lavor et al. [2012]. Quando um vértice  $j$  tem de ser colorido, o anterior ( $i$ ), que deverá ser seu vizinho, ou, na ausência de um vértice previamente colorido, o que foi marcado como predecessor, é usado para determinar a cor, de acordo com a seguinte estratégia: se não houver nenhum vizinho colorido, a única cor válida é 1, caso contrário, a função SELECAOCOR() retorna um conjunto cuja cardinalidade é, no máximo, 2, cujos elementos são:

1.  $x(i) + d_{i,j}$ .
2.  $x(i) - d_{i,j}$  (retornado apenas se  $x(i) > d_{i,j}$ ).

Esta estratégia de seleção de cor é aplicada em tempo  $O(1)$ , já que são necessárias apenas duas operações aritméticas para sua execução.



---

**Algoritmo 2** Procedimento geral *branch-prune-and-bound* (BPB).

---

**Require:** grafo  $G = (V, E)$ , distâncias  $d : E \cup \{(i, i) : i \in V\} \rightarrow \mathbb{N}$ , vértice anterior  $i$ , vértice atual  $j$  a ser colorido, coloração parcial  $x$ , melhor coloração completa atual  $x_{melhor}$ , limite superior  $ls$ , vetor  $predec$  of predecessores de cada vértice (inicializados com -1 na primeira chamada) e profundidade da árvore de enumeração  $prof$ .

```
1: função BRANCH-PRUNE-AND-BOUND( $G = (V, E)$ ,  $d$ ,  $i$ ,  $j$ ,  $x$ ,  $x_{best}$ ,  $ub$ ,  $pred$ ,  $dpt$ )
2:   para cada vizinho  $k$  of  $j$  faça
3:     se  $predec[k] = -1$  então
4:        $predec[k] \leftarrow i$       ▷ Coloca o vértice atual como predecessor de seus vizinhos
5:   se  $i = -1$  então
6:      $i \leftarrow predec[j]$       ▷ Se não veio de um vizinho, usa informação de predecessor
7:    $coresDisponiveis \leftarrow SELECAOCOR(G, d, i, j, x, ls)$ 
8:   enquanto  $coresDisponiveis \neq \emptyset$  faça
9:      $cor \leftarrow$  elemento de  $coresDisponiveis$ 
10:     $coresDisponiveis \leftarrow coresDisponiveis \setminus \{cor\}$ 
11:     $x(j) \leftarrow cor$ 
12:    se  $\max_{v \in V \mid v \text{ foi colorido}} x(v) \geq ub$  então
13:      Remove cor de  $i$ 
14:      continue      ▷ Descarta esta solução parcial por limite (bounding)
15:    se TESTEFACTIBILIDADE( $G, d, x, i$ ) = falso então
16:      Remove cor de  $i$ 
17:      retorne      ▷ Violação de distância, descarta solução parcial (pruning)
18:    se  $prof = |V|$  então      ▷ Se verdadeiro, todos os vértices foram coloridos
19:      se  $\max_{v \in V} x(v) < \max_{v \in V} x_{melhor}(v)$  então
20:         $x_{melhor} \leftarrow x$ 
21:         $ls \leftarrow \max_{v \in V} x(v)$ 
22:    senão
23:       $temVizinho \leftarrow$  falso
24:      para cada vizinho  $k$  of  $j$  faça
25:        se  $k$  não foi colorido então
26:           $temVizinho \leftarrow$  verdadeiro
27:          BRANCH-PRUNE-AND-BOUND( $G, d, j, k, x, x_{melhor}, ls, prof + 1$ )
28:        se  $temVizinho =$  falso então
29:          para cada vértice  $k$  of  $G$  tal que  $predec[k] \neq -1$  faça      ▷ Apenas vértices que
tem algum vizinho já colorido
30:            se  $k$  não foi colorido então
31:              BRANCH-PRUNE-AND-BOUND( $G, d, -1, k, x, x_{melhor}, ls, prof + 1$ )
32:            Remove cor de  $i$ 
33:    retorne  $x_{melhor}$ 
```

---



O segundo critério utiliza informações de todos os vizinhos coloridos para determinar a cor do vértice atual  $j$ . Isto é feito solucionando-se um sistema de equações (para o MinEQ-CDGP) ou inequações (para o MinGEQ-CDGP), que são derivadas das restrições de distâncias. Assim, sendo  $j$  o vértice a ser colorido, sua cor  $x(j)$  deve ser solução do sistema, onde há uma expressão para cada vizinho  $i$  de  $j$ , a qual é definida como:

$$|x(j) - x(i)| \boxtimes d_{i,j}$$

Onde  $\boxtimes$  é  $=$  para o MinEQ-CDGP ou  $\geq$  para o MinGEQ-CDGP. A cor que será atribuída a  $j$ , então, é o menor número natural que satisfaz todas as expressões. Este procedimento sempre retorna um conjunto de cardinalidade 1, ou seja, apenas uma cor (já que apenas o menor índice é retornado), porém, requer mais operações por vértice. Esta estratégia é realizada em tempo  $O(ls)$ , sendo  $ls$  um limite superior para a maior cor usada, já que, para resolver o sistema, deve-se rastrear todos os valores solução no intervalo  $[1 .. ls]$  e selecionar o menor valor.

Resultados computacionais para os algoritmos *branch-prune-and-bound*, bem como uma discussão sobre o comportamento do método em relação à obtenção da solução ótima nos diferentes modelos de colorações com distâncias, são apresentados por de Freitas et al. [2016].

### 5. Considerações finais

Neste artigo, foi realizado um estudo dos resultados para problemas de colorações em grafos com restrições de distâncias, incluindo propriedades de factibilidade e complexidade para algumas classes de grafos, modelos de programação inteira e por restrições, além de algoritmos *branch-prune-and-bound*, baseados no método *branch-and-prune* utilizado em problemas de geometria de distâncias moleculares discretizáveis.

Alguns trabalhos em andamento relacionados ao tema incluem o desenvolvimento e uso de outras formulações de programação inteira, a aplicação de planos de corte aos mesmos e a identificação de outras características de factibilidade para outras classes de grafos. Outros tipos de restrições de distâncias também estão sendo explorados, consistindo em outros casos especiais da T-coloração.

### Referências

- Audhya, G., Sinha, K., Ghosh, S., e Sinha, B. (2011). A survey on the channel assignment problem in wireless networks. *Wireless Communications and Mobile Computing*, 11:583–609.
- de Freitas, R., Dias, B., Maculan, N., e Szwarcfiter, J. (2016). Distance geometry approach for special graph coloring problems. *ArXiv e-prints*. URL <https://arxiv.org/abs/1606.04978>.
- Dias, B., de Freitas, R., Maculan, N., e Michelon, P. (2016). Solving the bandwidth coloring problem applying constraint and integer programming techniques. *Optimization Online (e-print)*. URL [http://www.optimization-online.org/DB\\_HTML/2016/06/5514.html](http://www.optimization-online.org/DB_HTML/2016/06/5514.html).
- Hale, W. (1980). Frequency assignment: Theory and applications. *Proceedings of the IEEE*, 25: 1497–1514.
- Koster, A. (1999). *Frequency assignment: models and algorithms*. PhD thesis, Universiteit Maastricht, the Netherlands.
- Lai, X. e Lü, Z. (2013). Multistart Iterated Tabu Search for Bandwidth Coloring Problem. *Computers & Operations Research*, 40:1401–1409.
- Lavor, C., Liberti, L., Maculan, N., e Mucherino, A. (2012). The discretizable molecular distance geometry problem. *European Journal of Operational Research*, 52(1):115–146.
- Saxe, J. B. (1979). Embeddability of weighted graphs in  $k$ -space is strongly NP-hard. In *Proceedings of 17th Allerton Conference in Communications, Control and Computing*, p. 480–489.