Sobre Cortes em Grafos Aresta-Coloridos

Luerbio Faria, Rubens Sucupira
IME/UERJ
UERJ, Rio de Janeiro, RJ, Brasil
luerbio@cos.ufrj.br, rasucupira@gmail.com

Sulamita Klein
IM/UFRJ e PESC/COPPE
UFRJ, Rio de Janeiro, RJ, Brasil
sula@cos.ufrj.br

Ignasi Sau
CNRS, LIRMM
Montpellier, França
ignasi.sau@gmail.com

Uéverton Souza
Instituto de Computação
UFF, Niterói, RJ, Brasil
ueverton@ic.uff.br

RESUMO

O objetivo desse artigo é estudar cortes de arestas em grafos que apresentem uma coloração de arestas previamente definida. Mais especificamente, a partir de um grafo aresta-colorido estamos interessados em determinar cortes que utilizem o menor número de cores (MIN-COLORED-CUT). Embora esse problema esteja intimamente relacionado com MIN-CUT, existe uma diferença inerente entre eles, pois em MIN-COLORED-CUT estamos interessados no número de cores presentes no corte, independentemente do número de arestas no corte. Enquanto MIN-CUT é polinomial, provamos que MIN-COLORED-CUT é NP-completo e difícil de aproximar em diversos cenários. Além disso, alguns casos polinomiais e tratáveis por parâmetro fixo também são apresentados.

PALAVRAS CHAVE. Corte mínimo. Grafos aresta-coloridos. Complexidade de algoritmos.

ABSTRACT

Our goal in this paper is to study the edge cuts on simple connected graphs where the edges are colored, in order to determine cuts using the minimum number of colors (MIN-COLORED-CUT). Although this problem is closely related to MIN-CUT, there is a diference between them, because in MIN-COLORED-CUT we are interested in the number of colors in the cut, dispite of the number of edges in it. While MIN-CUT is polynomial, we prove that MIN-COLORED-CUT is NP-complete and has hard approximation in various scenarios. Furthermore, some polynomial and fixed parameter tractable cases are presented.

KEYWORDS. Minimum cut. Edge colored graphs. Algorithms complexity.

1. Introdução

Seja $G = (V, E)$ um grafo conexo simples, e sejam S e T dois subconjuntos não vazios disjuntos de $V(G)$. Usaremos a notação $[S, T]$ para representar o subconjunto de $E(G)$ formado pelas arestas que possuem um extremo em S e o outro em T. Um *corte de arestas* de G é um subconjunto $[S, V \setminus S] = \partial S$ de $E(G)$ para algum subconjunto não vazio $S \subsetneq V(G)$. Dados s e t dois vértices distintos de $V(G)$, um (s,t)-*corte* de G é um corte de arestas $[S, T]$ de G tal que $s \in S$ e $t \in T$.

Seja $G^c = (V, E, c)$ um grafo conexo simples $G = (V, E)$ com uma coloração de arestas $c : E(G) \to \{1, 2, \cdots, p\}$ não necessariamente própria. Aqui os rótulos numéricos representam as cores atribuídas às arestas de G através da coloração c. A *capacidade* $C(\partial S)$ de um corte de arestas ∂S é o número de cores presentes nesse corte. Denotamos por $c(\partial S) = \{c(e) | e \in \partial S\}$. Observe que $C(\partial S) = |c(\partial S)|$.

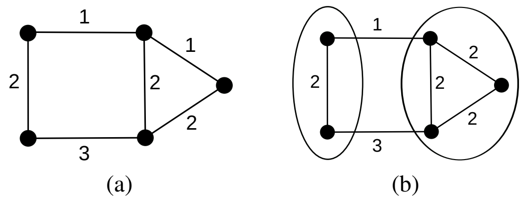


Figura 1: (a) Um grafo aresta-colorido G e (b) Um corte de G com capacidade 2.

A partir de agora consideramos $G^c = (V, E, c)$ um grafo conexo simples munido de uma coloração de arestas c. Dado um grafo G^c, nosso objetivo é determinar cortes de arestas com a menor capacidade. Esse é um problema de otimização que chamamos de MIN-COLORED-CUT. A seguir apresentamos a forma de decisão desse problema.

MIN-COLORED-CUT
INSTÂNCIA: Um grafo aresta-colorido G^c e um inteiro $k > 0$.
PERGUNTA: Existe um subconjunto não vazio $S \subsetneq V(G)$ tal que $C(\partial S)$ é no máximo k ?

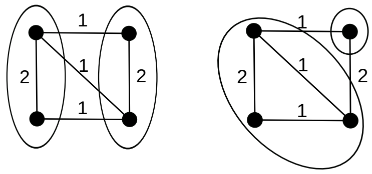
Associado ao problema MIN-COLORED-CUT, temos o problema de decisão MIN-COLORED-(s,t)-CUT, obtido quando fixamos dois vértices $s \in S$ e $t \in T$.

MIN-COLORED-(s,t)-CUT
INSTÂNCIA: Um grafo aresta-colorido G^c com vértices $s, t \in V$ fixados e um inteiro $k > 0$.
PERGUNTA: Existe um corte de arestas $[S, T]$ com $s \in S$, $t \in T$ e capacidade no máximo k ?

Os problemas MIN-COLORED-CUT e MIN-COLORED-(s,t)-CUT foram inicialmente introduzidos por Coudert et al. [2007]; Rizzi [2000], usando uma terminologia diferente da nossa.

Note que se a função de coloração $c : E(G) \to \{1, 2, \cdots, p\}$ é injetiva, isto é, se todas as arestas possuem cores distintas, então os problemas MIN-COLORED-CUT e MIN-COLORED-(s,t)-CUT correspondem exatamente a MIN-CUT e MIN-(s,t)-CUT, respectivamente, sendo portanto solucionáveis em tempo polinomial pelo algoritmo clássico do Fluxo Máximo: Ford e Fulkerson [1956]; Edmonds e Karp [1972].

Não podemos resolver MIN-COLORED-CUT através do algoritmo anterior porque nesse caso o tamanho do corte de arestas geralmente não importa: podemos ter um corte de arestas com muitas arestas coloridas com poucas cores e, por outro lado podemos encontrar um corte de arestas pequeno que apresenta muitas das cores da coloração de arestas desse grafo. A Figura 2 ilustra dois exemplos dessas situações num mesmo grafo.



(a) (b)

Figura 2: (a) Um corte colorido mínimo com muitas cores e poucas arestas e (b) um corte colorido máximo de um grafo com muitas (todas as) cores e poucas arestas.

Neste artigo estudamos a complexidade computacional desses problemas, com ênfase especial em suas complexidades aproximativa e parametrizada para várias escolhas de parâmetros. Para uma introdução ao assunto veja Flum e Grohe [2006]; Niedermeier [2006]; Downey e Fellows [2013a]; Cygan et al. [2015]. Utilizamos a notação padrão de Teoria dos Grafos: Diestel [2010]. Ao longo do artigo, denotamos por n o número de vértices do grafo de entrada do problema considerado.

2. Preliminares

Nesta seção apresentamos alguns conceitos relacionados a algoritmos aproximativos e algoritmos parametrizados necessários para este trabalho.

Definição. Downey e Fellows [2013b]*(S-redução)* Sejam A e B dois problemas de otimização, $c_A : A \to \mathbb{N}$, $c_B : B \to \mathbb{N}$ suas respectivas funções de custo, e $OPT_A(x_A)$ e $OPT_B(x_B)$ as soluções ótimas dos problemas A e B considerando as instâncias x_A e x_B. Uma *S-redução* é um par de algoritmos de tempo polinomial f e g, tais que:

- f leva instâncias x_A de A em instâncias $x_B = f(x_A)$ de B,
- Se y' é uma solução de x_B, então $g(y')$ é uma solução de x_A
- $|OPT_B(f(x))| = |OPT_A(x)|$
- $c_A(g(y')) = c_B(y')$.

Uma *classe de complexidade de aproximação* é um conjunto de problemas de otimização que permitem algoritmos de aproximação com uma dada relação de aproximação.

Proposição 2.1. *Crescenzi [1997]. Sejam A e B dois problemas de otimização, se A se S-reduz a B e B pertence a uma classe de complexidade de aproximação $\mathcal{C}$ então $A \in \mathcal{C}$.*

A seguir apresentamos alguns conceitos da Complexidade Parametrizada.

Definição. Downey e Fellows [2013a]*(Classe FPT)* Dado um problema A, diremos que A é *tratável a parâmetro fixo*, i.e., *pertence à classe FPT* com relação a um parâmetro k (escolhido em A), se existe um algoritmo que resolve A em tempo $f(k) \cdot n^{O(1)}$ para alguma função computável f.

Definição. Downey e Fellows [2013a]*(FPT-redução)* Dados dois problemas parametrizados (A, k) e (B, k'), uma *FPT-redução* é uma transformação que cumpre as seguintes condições:

- leva uma instância (x, k) de A em uma instância (x', k') de B
- $(x, k) \in A$ se e somente se $(x', k') \in B$
- (x', k') é computável em tempo FPT

- $k' \leq g(k)$ para alguma função computável g

Além da classe FPT, Downey e Fellows definiram classes de problemas parametrizados de acordo com seu nível de intratabilidade parametrizada. Essas classes são organizadas em uma W-hierarquia ($FPT \subseteq W[1] \subseteq W[2] \subseteq \ldots \subseteq W[P]$), e são baseadas na complexidade dos circuitos necessários para se verificar a validade de uma solução, ou, alternativamente, na profundidade natural dos circuitos lógicos para o problema. É conjecturado que cada uma dessas inclusões de classes são próprias, e se $P = NP$ então $FPT = W[P]$. Veja Santos e Souza [2015].

Proposição 2.2 (Downey e Fellows [2013a]). *Sejam (A, k) e (B, k') dois problemas parametrizados, se A se FPT-reduz a B e B pertence a uma classe da W-hierarquia $\mathcal{C}$ então $A \in \mathcal{C}$.*

É conhecido na literatura especializada que todo problema em FPT admite um núcleo (kernel), isto é, pode-se aplicar um conjunto de regras de redução que eliminam partes da instância original do problema, tornando-o mais simples. Em teoria da complexidade parametrizada, frequentemente é possível reduzir em tempo polinomial a instância original do problema em uma instância cujo tamanho resultante seja não maior que o tamanho da instância original. Essa instância resultante é conhecida como núcleo. Além disso, o tamanho do núcleo é sempre limitado por uma função do parâmetro. Sendo assim, cabe perguntar se tais problemas em FPT admitem um núcleo de tamanho limitado por uma função polinomial com relação ao parâmetro escolhido. Todavia núcleos polinomiais nem sempre ocorrem[Downey e Fellows [2013a]].

A seguir apresentamos um framework para demonstrar a inviabilidade de núcleos polinomiais. Para estabelecê-lo precisaremos da definição de *OU-composição* e do teorema dado a seguir.

Teorema 2.3. *[Bodlaender et al., 2008]. Se o problema Π admite um algoritmo de OU-composição, então Π não admite núcleo polinomial a menos que* $\mathrm{NP} \subseteq \mathrm{coNP/poly}$.

Um *algoritmo de OU-composição* para um problema parametrizado $\Pi \subseteq \Sigma^* \times \mathbb{N}$ é um algoritmo que:

- recebe como entrada uma sequência de instâncias $(G_1, k_1), \cdots, (G_t, k_t)$ de Π e
- produz, em tempo polinomial em $\Sigma|G_i|$, uma instância (G, k) tal que
- $k \leq f(\max |k_i|)$ com f polinomial e
- $(G, k) \in \Pi$ se e somente se existe $1 \leq i \leq t$ tal que $(G_i, k_i) \in \Pi$.

3. Resultados de NP-completude

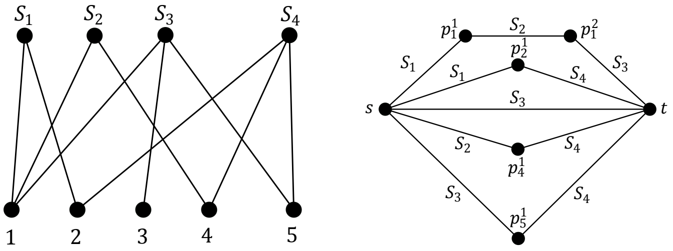
Richard Karp apresentou 21 problemas combinatórios em seu artigo de 1972 e mostrou que todos eles são NP-completos. Entre esses problemas figura SET COVER. A seguir apresentamos o problema SET COVER em sua forma de decisão.

SET COVER
INSTÂNCIA: Um conjunto $C = \{1, \cdots, m\}$, uma família $S = \{S_1, \cdots, S_p\}$ de subconjuntos de C tal que $C \subset \bigcup_{i=1}^{p} S_i$ e um inteiro $k > 0$.
PERGUNTA: Existe uma subfamília $S_{i_1}, \cdots, S_{i_k}$ com k conjuntos de S tal que $C \subset \bigcup_{j=1}^{k} S_{i_j}$?

Se $C \subset \bigcup_{j=1}^{k} S_{i_j}$ dizemos que a subfamília $S_{i_1}, \cdots, S_{i_k}$ *cobre* C.

Apresentamos a seguir uma redução de instâncias de SET COVER em instâncias de MIN-COLORED-(s,t)-CUT, mostrando que esse último problema é NP-completo.

Vamos identificar os elementos de C e os subconjuntos de S com vértices de um grafo bipartido $G' = (V', E')$ de modo que $V' = C \cup S$, C e S são conjuntos independentes e existe uma aresta unindo um elemento i de C a um conjunto S_j de S se $i \in S_j$. A Figura 3(a) ilustra um exemplo do grafo G'.



(a) (b)

Figura 3: (a) Grafo bipartido G' usado na redução de SET COVER e (b) o (s,t)-grafo G^c.

A partir de G' construimos o grafo $G^c = (V, E, c)$ do seguinte modo:

- Crie dois vértices s e t.
- Para cada $i \in C$ com $N_{G'}(i) = \{S_i^1, S_i^2, \cdots, S_i^{d(i)}\}$ crie $P_i = (s, p_i^1, p_i^2, \cdots, p_i^{d(i)-1}, t)$ um caminho unindo s a t de forma que o número de arestas em P_i seja igual ao grau de i em G', e cada aresta em P_i tenha uma cor associada a um conjunto S_j vizinho de i em G', isto é, $c(s, p_i^1) = S_i^1, c(p_i^j, p_i^{j+1}) = S_i^j$ para $1 \leq j \leq d(i) - 2$ e $c(p_i^{d(i)-1}, t) = S_i^{d(i)}$.

A Figura 3 ilustra a construção de G^c a partir do grafo G'.

Teorema 3.1. *O grafo G^c obtido anteriormente tem um corte de arestas $[S,T]$ tal que $s \in S$, $t \in T$ com capacidade no máximo k se e somente se existem k conjuntos de S que cobrem C.*

Demonstração. Suponha que o (s,t)-grafo G^c possui um (s,t) corte de arestas $[S,T]$ com capacidade k (k cores) em que $s \in S$ e $t \in T$. Seja $\{S_1, S_2, S_3 \ldots, S_k\}$ o conjunto das cores do corte $[S,T]$. Como G tem m caminhos disjuntos ligando s a t, qualquer (s,t)-corte contém pelo menos uma aresta de cada (s,t)-caminho. Logo, por definição de $[S,T]$, cada um dos m (s,t)-caminhos possui pelo menos uma aresta colorida com uma das k cores do corte. Assim, para cada $i \in \{1, 2, 3, \ldots, m\}$ existe $S_j, j \in \{1, 2, 3 \ldots, k\}$ tal que $i \in S_j$. Portanto, $C \subset \bigcup_{j=1}^{k} S_j$.

Por outro lado, suponha que existem k conjuntos $S_{i_1}, S_{i_2}, \cdots, S_{i_k}$ de S cobrindo C, i.e., $C \subset \bigcup_{j=1}^{k} S_{i_j}$. Por construção, existe um representante de cada uma das cores $S_{i_1}, S_{i_2}, \cdots, S_{i_k}$ em cada um dos (s,t)-caminhos de G'. Dessa forma, o (s,t)-grafo G^c tem um corte de arestas $[S,T]$ com $s \in S$ e $t \in T$ com capacidade no máximo k. $\square$

Dado um grafo $G = (V, E)$, um subconjunto $C \subset V$ é dito uma *cobertura de vértices* de G se toda aresta de G possui pelo menos um extremo em C. A seguir enunciamos o problema de decisão VERTEX COVER.

VERTEX COVER
INSTÂNCIA: Um grafo $G = (V, E)$ e um inteiro $k > 0$.
PERGUNTA: Existe uma cobertura de vértices $S \subseteq V$ com tamanho no máximo k ?

Observe que o problema VERTEX COVER é um caso particular de SET COVER quando os conjuntos S_j da família S representam os vértices em $V(G)$ e os elementos $i \in C$ representam as arestas em $E(G)$. Portanto a construção da instância para o Teorema 3.3 será análoga à realizada na Seção 3.

Lema 3.2. *Garey et al. [1974].* VERTEX COVER *permanece* NP*-completo mesmo restrito a grafos cúbicos.*

Teorema 3.3. MIN-COLORED-(s, t)-CUT *é* NP*-completo mesmo se toda cor aparece no máximo três vezes e cada (s, t)-caminho tem tamanho dois.*

Demonstração. Dado um grafo $G = (V, E)$ subcúbico com n vértices e m arestas, construa um (s, t)-grafo G' com m (s, t)-caminhos $P_i = (s, p_i, t)$, de modo que se P_i corresponde à aresta $(u, v) \in G$, então $c(s, p_i) = u$ e $c(p_i, t) = v$. A Figura 4 ilustra essa construção. Temos um (s, t)-grafo G' no qual os extremos das arestas de G representam as cores das arestas em G'.

Dado um (s, t)-corte de G', cada um dos m (s, t)-caminhos tem uma aresta no corte, caso contrário t poderia ser alcançado a partir de s através de um caminho que não tivesse arestas no corte. Como as arestas de G' estão coloridas pelos vértices de G, e os caminhos de G' representam as arestas de G, então para cada aresta de G ao menos um de seus extremos possui sua cor no corte, sendo assim as cores presentes no corte induzem uma cobertura de vértices de G.

Reciprocamente, se existe uma cobertura de vértices $B \subset V$ de G, então todas as arestas de G têm pelo menos um extremo em B. Portanto, todos os (s, t)-caminhos de G' têm pelo menos uma aresta colorida por um vértice de B, assim podemos escolher um subconjunto dessas arestas para formar um (s, t)-corte de G'.

Sem perda de generalidade, podemos assumir que o grau máximo de cada vértice de G é três, implicando que em G' cada cor ocorrerá no máximo três vezes. $\square$

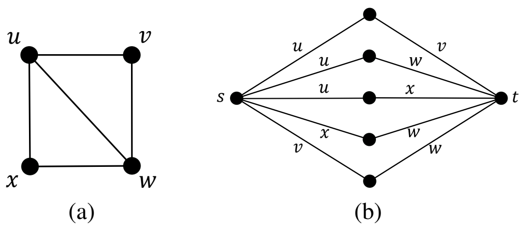


Figura 4: (a) O grafo G e (b) seu (s, t)-grafo G'.

A seguir provamos que MIN-COLORED-(s, t)-CUT é NP-completo mesmo se toda cor aparece no máximo duas vezes e cada (s, t)-caminho tem comprimento no máximo três. Usaremos uma redução de VERTEX COVER em que todo vértice possui grau três e toda aresta incide a no máximo um vértice de grau três.

Lema 3.4. VERTEX COVER *permanece NP-completo mesmo em grafos com grau máximo três, em que toda aresta incide a no máximo um vértice de grau três.*

Demonstração. Seja G um grafo cúbico. Construiremos a partir de G um grafo G' de grau máximo três, em que toda aresta incide a no máximo um vértice de grau três, de modo que G possua uma cobertura de tamanho k se e somente se G' possuir uma cobertura de tamanho $k + |E(G)|$. A construção é dada a seguir partindo de um grafo G' vazio. Tal construção é ilustrada na Figura 5

- Faça $V(G') = V(G) \cup \{v_e^1, v_e^2 : e \in E(G)\}$;
- Para cada aresta $e = (u, w) \in E(G)$ acrescente em G' o caminho (u, v_e^1, v_e^2, w).

Seja B uma cobertura por vértices de G, tal que $|B| = k$. Podemos construir uma cobertura B' de G' primeiramente atribuindo todo vértice em B também a B'. Como B é uma cobertura de G, então para todo caminho (u, v_e^1, v_e^2, w) de G' $(u, w \in V(G))$, temos que u ou w ou ambos pertencem a B', de acordo com a atribuição que foi feita. Desta forma, se $u \in B'$ adicionamos v_e^2 em B', senão adicionamos v_e^1 em B'. Sendo assim temos que B' cobre todas as arestas de G' e possui tamanho $k + |E(G)|$.

Por outro lado, seja B' uma cobertura mínima de G' de tamanho $k + |E(G)|$. Todo caminho (u, v_e^1, v_e^2, w) de G' possui ao menos dois vértices em B', sendo pelo menos um deles em $\{v_e^1, v_e^2\}$. Como B' é uma cobertura mínima, B' possui exatamente um vértice em $\{v_e^1, v_e^2\}$ para todo $e \in E(G)$. Sendo assim, podemos definir uma cobertura B para G como $B = V(G) \cap B'$. Logo B é uma cobertura de G de cardinalidade k, pois B' possui exatamente um vértice em $\{v_e^1, v_e^2\}$ e consequentemente ao menos um vértice em $\{u, w\}$ para todo $e = (u, w) \in E(G)$. □

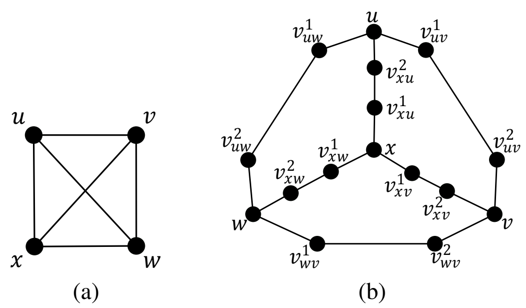


Figura 5: (a) Um grafo cúbico G e (b) seu grafo associado G'.

Teorema 3.5. MIN-COLORED-(s,t)-CUT *é* NP*-completo mesmo se toda cor aparece no máximo duas vezes e cada* (s,t)*-caminho tem comprimento no máximo três.*

Demonstração. Dado um grafo $G = (V, E)$ em que o grau dos vértices é no máximo três e toda aresta incide a no máximo um vértice de grau três, construímos um grafo G' como no Teorema 3.3. Como cada vértice tem grau no máximo três, teremos um (s,t)-grafo G' no qual para cada cor há no máximo três (s,t)-caminhos com arestas coloridas por essa cor. Usaremos o grafo G' para construir um novo grafo G'' do seguinte modo:

- Escolha uma cor c_i que ocorre três vezes em G' e selecione os três (s,t)-caminhos de G' que possuem arestas coloridas com a cor c_i. Substitua os vértices intermediários dos caminhos selecionados por novas arestas, cada uma colorida com uma nova cor, distintas das cores originais de G', e distintas entre si. Contraia as arestas coloridas com a mesma cor e que são adjacentes em uma única aresta (isto faz com que essa cor apareça no máximo duas vezes em G'');

- Repita o procedimento anterior para as demais cores que também ocorrem três vezes no grafo corrente;

Como toda aresta incide a no máximo um vértice de grau três, o procedimento acima descrito é aplicado no máximo uma vez para cada (s,t)-caminhos de G'. Sendo assim, G'' é um (s,t)-grafo no qual cada (s,t)-caminho tem comprimento no máximo três e cada cor aparece no máximo duas vezes, como ilustra a Figura 6.

Dado um (s,t)-corte de G'', cada um dos (s,t)-caminhos tem uma aresta no corte. Se o corte só envolve arestas coloridas com as cores antigas, isso significa que para cada (s,t)-caminho de G'' (aresta de G) está sendo tomado um vértice em cada aresta de G: temos uma cobertura de vértices de G. Caso o corte envolva arestas coloridas com as cores novas, é possível trocar cada cor nova por uma antiga, já que a remoção de qualquer aresta do (s,t)-caminho correspondente desconecta s de t ao longo desse caminho. A recíproca é idêntica à do Teorema 3.3. □

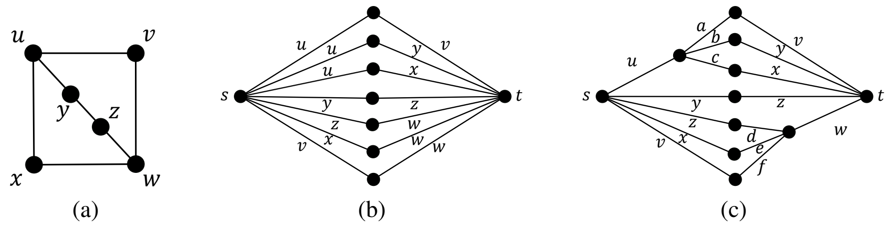


Figura 6: (a) O grafo G, (b) seu (s,t)-grafo G' e (c) G''

Na demonstração acima, embora as cores ocorram no máximo duas vezes, algumas cores aparecem em três (s,t)-caminhos distintos.

Embora esteja em aberto a complexidade de MIN-COLORED-CUT para grafos não direcionados, sabemos mostrar que a versão para grafos direcionados é NP-completo.

Teorema 3.6. MIN-COLORED-CUT *é* NP*-completo para digrafos.*

Demonstração. Em um grafo direcionado a capacidade de um corte contabiliza somente arestas que saem da parte S e chegam na parte T. Sendo assim, podemos modificar a redução de VERTEX COVER para MIN-COLORED-(s,t)-CUT primeiramente direcionando as arestas conforme uma busca em largura a partir de s. Em seguida, para forçar s e t a não pertencerem a uma mesma parte da partição, basta para cada vértice intermediário v, adicioná-lo a um subgrafo direcionado completo $\overrightarrow{K_v}$ de tamanho $2m+1$, sendo cada aresta de $\overrightarrow{K_v}$ colorida com uma nova cor. Em seguida adicionamos uma aresta partindo de t com uma nova cor para todo novo vértice de $\overrightarrow{K_v}$, assim como uma aresta de cada novo vértice de $\overrightarrow{K_v}$ para s. Desse modo, todo corte de tamanho menor que m terá $s \in S$, $t \in T$, e todo subgrafo completo criado anteriormente deverá estar em uma única parte, mantendo o comportamento da construção original. O restante segue como no Teorema 3.4. □

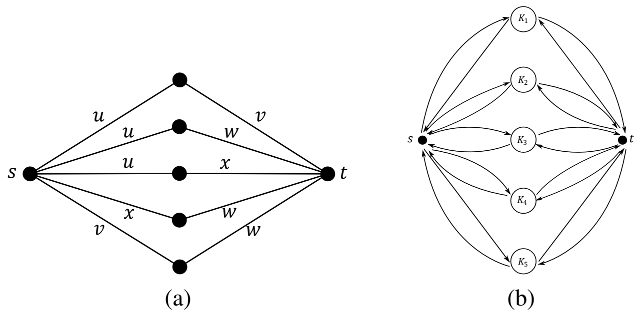


Figura 7: (a) Grafo G' usado na redução de VERTEX COVER e (b) o digrafo G''.

4. Aproximação

Lema 4.1. *Dinur e Steurer [2014]. Para todo $\epsilon > 0$ é* NP*-difícil aproximar* SET COVER *por um fator de $(1-\epsilon)\ln m$, sendo m o tamanho do universo (número de elementos).*

Nos corolários a seguir, representaremos, por abuso de linguagem, o número de cores da coloração $c : E(G) \to \{1, 2, \cdots, p\}$ por c.

Corolário 4.2. MIN-COLORED-(s,t)-CUT *restrito a grafos bipartidos planares não pode ser aproximado por um fator de $(1-\epsilon)\ln(c)$ para qualquer constante $\epsilon > 0$ a menos que* P = NP.

Demonstração. Tome o grafo G^c construído na Seção 3 e para cada um dos m (s,t)-caminhos $P_i = (s, p_i^1, p_i^2, \cdots, p_i^{d(i)-1}, t)$ execute o seguinte procedimento. Se $d(i)$ é par, então mantenha P_i e suas cores inalteradas. Se $d(i)$ é impar, então substitua P_i por $P_i^* = (s, p_i^1, p_i^2, \cdots, p_i^{d(i)-1}, p_i^{d(i)}, t)$, tal que $c(p_i^{d(i)-1} p_i^{d(i)}) = c(p_i^{d(i)} t)$ e as demais arestas de P_i^* têm as suas cores inalteradas. Desse modo obteremos um (s,t)-grafo H^c planar que possui apenas ciclos pares, ou seja, H^c é um grafo bipartido planar com arestas coloridas. Neste ponto, é fácil ver que H^c tem um (s,t)-corte $[S,T]$ com capacidade k se e somente se G^c também possui tal corte. Logo, pelo Teorema 3.1 existe um subconjunto de S de tamanho k que cobre C se e somente se H^c tem um (s,t)-corte $[S,T]$ com capacidade k.

Como podemos observar, a construção descrita acima é tanto uma S-redução quanto uma FPT-redução quando ambos os problemas são parametrizados pelo tamanho da solução buscada. Notem que o número de cores do grafo G^c é igual ao número de elementos da instância de SET COVER. Logo pelo Lema 4.1 e pela Proposição 2.1 temos o requerido. □

Embora o resultado de inaproximação para o problema MIN-COLORED-(s,t)-CUT resulte diretamente da S-redução de SET COVER para o problema em grafos bipartidos planares, o mesmo não se aplica tão diretamente para grafos completos.

Corolário 4.3. MIN-COLORED-(s,t)-CUT *em grafos completos não pode ser aproximado por um fator de* $\frac{(1-\epsilon)}{2}\ln(c)$ *para qualquer constante* $\epsilon > 0$ *a menos que* $\mathrm{P} = \mathrm{NP}$.

Demonstração. Considere a redução apresentada no Corolário 5.3. Pelo Corolário temos que G'' possui uma solução de custo $k+1$ se e somente se a instância de SET COVER associada possui uma solução de custo k. Sejam *Opt* e *Opt'* soluções ótimas para MIN-COLORED-(s,t)-CUT e para SET COVER, respectivamente. Pela construção apresentada no Corolário 5.3 sabemos que $Opt = Opt' + 1$. Sendo assim, a partir de uma solução de custo $(1-\epsilon)\ln(c)(Opt-1)$ para MIN-COLORED-(s,t)-CUT obtém-se uma solução de custo $(1-\epsilon)\ln(c)(Opt')$ para set cover, que sabemos ser NP-difícil de ser obtido. Sendo x o fator tal que $x.Opt = (1-\epsilon)\ln(c)(Opt-1)$ temos:

$$x.Opt = (1-\epsilon)\ln(c)Opt - (1-\epsilon)\ln(c) \Rightarrow x = (1-\epsilon)\ln(c) - \frac{(1-\epsilon)\ln(c)}{Opt}$$

Como $Opt \geq 2$ então $x \geq (1-\epsilon)\ln(c) - \frac{(1-\epsilon)}{2}\ln(c) = \frac{(1-\epsilon)}{2}\ln(c)$. □

5. Complexidade Parametrizada

A seguir apresentamos alguns resultados referentes à complexidade parametrizada dos problemas. Começamos com resultados que serão utilizados na demonstração do Corolário 5.2.

Lema 5.1. *Downey e Fellows [2013b]* SET COVER *é W[2]-completo quando parametrizado pelo tamanho da solução buscada.*

Corolário 5.2. MIN-COLORED-(s,t)-CUT *restrito a grafos bipartidos planares é* W[2]*-difícil quando parametrizado pelo tamanho da solução.*

Demonstração. A mesma do Corolário 4.2, utilizando-se o Lema 5.1 e a Proposição 2.2. □

Podemos modificar a redução apresentada na Seção 3 também para provar que MIN-COLORED-(s,t)- CUT é W[2]-difícil em grafos completos.

Corolário 5.3. MIN-COLORED-(s,t)-CUT *é* W[2]*-difícil em grafos completos, quando parametrizado pelo tamanho da solução buscada.*

Demonstração. Tome o grafo G^c construído na Seção 3 e crie arestas ligando todos os vértices dois a dois não adjacentes, de modo a obter um grafo completo G''. Atribua a todas as novas arestas uma nova cor diferente das cores já existentes em G. A partir dessa construção teremos o seguinte resultado: G'' tem um corte de arestas $[S, T]$ tal que $s \in S$, $t \in T$ e capacidade $k + 1$ se e somente se existem k conjuntos de S que cobrem C. Note que a aresta $\{s, t\}$ criada tem cor igual à cor adicional, e portanto essa nova cor está sempre presente em qualquer $[S, T]$ corte. □

Teorema 5.4. MIN-COLORED-CUT *e* MIN-COLORED-(s, t)-CUT *são* FPT *quando parametrizados pelo número de cores da coloração dada.*

Demonstração. Dadas as cores $1, \cdots, c$ da coloração original do grafo G, testamos exaustivamente a remoção das arestas coloridas com as cores da coloração dada até esgotarmos todas as possibilidades, o que no pior caso, empregaria todas as c cores. Esse processo tem um custo de $2^c \cdot O(m)$, o que justifica o teorema. □

Observe que MIN-COLORED-(s, t)-CUT pode ser resolvido em tempo polinomial quando $c = O(\log n)$, e pode ser resolvido em tempo pseudo-polinomial quando G tem um (s, t)-corte não colorido de tamanho $O(\log n)$.

Como temos um algoritmo FPT para ambos os problemas quando parametrizados pelo número de cores, é interessante verificar se esses problemas possuem núcleo (kernel) polinomial. Sendo assim o próximo resultado complementa o Teorema 5.4 acima.

Lema 5.5. MIN-COLORED-(s, t)-CUT *não admite núcleo polinomial quando parametrizado pelo número de cores, a menos que* $\mathrm{NP} \subseteq \mathrm{coNP/poly}$.

Demonstração. Usando o Teorema [Bodlaender et al., 2008], basta mostrar que o problema $\Pi =$ MIN-COLORED-(s, t)-CUT admite um algoritmo de OU-composição. Para isso, consideremos uma sequência de instâncias $(G_1, k_1), \cdots, (G_t, k_t)$ desse problema, nas quais $G_1, \cdots, G_t$ são grafos conexos dois a dois disjuntos tais que em cada G_i existem dois vértices distintos s_i e t_i fixados e k_i é um inteiro positivo para $1 \leq i \leq t$. Vamos construir G colapsando os vértices t_i com s_{i+1} para cada $1 \leq i \leq t - 1$ e fazendo $s = s_1$ e $t = t_t$. Em cada grafo G_i existe uma coloração com as cores $1, \cdots, c_i$ para $1 \leq i \leq t$. Tome $c = \max_{1 \leq i \leq t} c_i$. Então G tem uma coloração com c cores. Observe que G é construido em tempo polinomial em $\Sigma|G_i|$ (colapsando os vértices especiais) e $k \leq \max_{1 \leq i \leq t} c_i$. Suponhamos, sem perda de generalidade, que $k_i = k$ para todo $1 \leq i \leq t$. Existe um (s, t)-corte em G com no máximo k cores se e somente se existe algum $1 \leq i \leq t$ tal que G_i admite um (s_i, t_i)-corte com no máximo k cores. De fato, se algum G_i admite um corte com no máximo k cores, este também é um corte de G com no máximo k cores. Reciprocamente, se nenhum grafo G_i admitisse um corte com no máximo k cores, então todos os G_i admitiriam cortes com pelo menos $k + 1$ cores e consequentemente G admitiria apenas cortes com pelo menos $k + 1$ cores. Desse modo o problema admite um algoritmo de OU-composição e o resultado segue. □

A seguir, algumas definições necessárias são apresentadas. O *span* de uma cor i é o número de componentes conexas induzidas pelo conjunto de arestas do grafo coloridas com a cor i.

Dado o grafo $G^c = (V, E, c)$, se cada cor tem *span* $= 1$, vamos construir o hipergrafo associado no qual o conjunto de vértices que compõem a componente conexa da cor i formam uma hiperaresta. De acordo com os resultados de Rizzi [2000], MIN-COLORED-CUT pode ser resolvido em tempo polinomial nesse caso, já que esse problema é equivalente a determinar o corte de hiperarestas mínimo em hipergrafos.

Teorema 5.6. *Dado um grafo aresta-colorido* (G, c)*, denotemos por* c_2 *o número de cores com span pelo menos dois. O problema* MIN-COLORED-CUT *pode ser resolvido em tempo* $2^{c_2} \cdot n^{O(1)}$.

Demonstração. Aplicaremos indução no número de cores com span pelo menos 2. O caso $c_2 = 0$ pode ser resolvido em tempo polinomial utilizando-se o algoritmo para MIN-CUT em hipergrafos: Rizzi [2000]. Suponha que existe um algoritmo recursivo que resolve o problema quando $0 \leq c_2 \leq t$. Se $c_2 = t + 1$, para cada cor i com span pelo menos 2, consideremos os dois grafos seguintes: G_1^i (no qual todas as arestas com a cor i são removidas), e G_2^i (no qual cada componente com a cor i recebe uma nova cor diferente das já existentes). Note que $OPT(G^c) = \min\{OPT(G_1^i) + 1, OPT(G_2^i)\}$ e como ambos os grafos decrementam o parâmetro c_2 em uma unidade, podemos aplicar o algoritmo recursivo para cada um deles.

□

Lema 5.7. *Rizzi [2000]* MIN-COLORED-CUT *é solucionável em tempo randomizado polinomial, quando o span máximo da instância G^c é delimitado por uma constante p.*

Dado um inteiro $p > 0$, denotemos por $c_p(G^c)$ o número de cores com span maior que p.

Teorema 5.8. *O problema* MIN-COLORED-CUT *pode ser resolvido por um algoritmo randomizado em tempo $2^{c_p(G^c)} \cdot n^{O(p)}$.*

Demonstração. Basta aplicar uma abordagem semelhante a apresentada na prova do Theorem 5.6 tomando a cada passo uma cor com span maior que p, até obter um grafo com span máximo p (caso base da nossa recursão) que será solucionado diretamente pelo algoritmo randomizado polinomial de Blin et al. [2014]. □

6. Resultados Polinomiais

Teorema 6.1. MIN-COLORED-(s,t)-CUT *pode ser solucionado em tempo polinomial quando toda cor aparece em no máximo dois (s,t)-caminhos de G, sendo G um grafo de entrada com (s,t)-caminhos de comprimentos quaisquer.*

Demonstração. Primeiramente observe que encontrar, caso exista, um caminho entre s e t passando por uma determinada aresta e pode ser executado em tempo polinomial através de um algoritmo de fluxo máximo. Além disso, um segundo e um terceiro caminhos entre s e t passando por e, caso existam, devem não conter ao menos uma aresta de cada um dos caminhos anteriores que passam por e. Sendo assim tais caminhos, caso existam, também podem ser obtidos em tempo polinomial. Note também que se toda cor aparece em no máximo dois (s,t)-caminhos, então toda aresta ocorre em no máximo dois (s,t)-caminhos. Logo, podemos reconhecer em tempo polinomial se toda cor aparece em no máximo dois (s,t)-caminhos de G, assim como enumerar todos os (s,t)-caminhos de G em caso positivo.

Portanto, caso toda cor de G apareça em no máximo dois (s,t)-caminhos, podemos enumerar todos os (s,t)-caminhos de G em tempo polinomial. Em seguida, em tempo polinomial, podemos construir uma instância F de MONOTONE WEIGHTED SAT, na qual cada clásula representa um (s,t)-caminho e toda cor representa uma variável, de tal forma tal que F possua uma atribuição satisfatível de peso k se e somente se G possui um corte de capacidade k. Neste ponto, para resolver MIN-COLORED-(s,t)-CUT basta executar em F o algoritmo polinomial para MONOTONE WEIGHTED SAT quando cada variável ocorre no máximo duas vezes (veja Porschen e Speckenmeyer [2007]). □

Corolário 6.2. MIN-COLORED-(s,t)-CUT *pode ser resolvido em tempo polinomial quando cada cor ocorre no máximo duas vezes e cada (s,t)-caminho tem comprimento dois.*

Lema 6.3. MIN-COLORED-CUT *pode ser resolvido em tempo polinomial quando G tem um (s,t)-corte não colorido de tamanho constante, e pode ser resolvido em tempo $O(c^5 \cdot m)$ em grafos planares.*

Referências

Blin, G., Bonizzoni, P., Dondi, R., Rizzi, R., e Sikora, F. (2014). Complexity insights of the minimum duplication problem. *Theoretical Computer Science*, 530:66–79. URL http://dx.doi.org/10.1016/j.tcs.2014.02.025.

Bodlaender, H. L., Downey, R. G., Fellows, M. R., e Hermelin, D. (2008). On Problems without Polynomial Kernels (Extended Abstract). In *Proc. of the 35th International Colloquium on Automata, Languages and Programming (ICALP)*, p. 563–574.

Coudert, D., Datta, P., Perennes, S., Rivano, H., e Voge, M. (2007). Shared risk resource group complexity and approximability issues. *Parallel Process. Lett.*, 17(2):169–184. URL http://dx.doi.org/10.1142/S0129626407002958.

Crescenzi, P. (1997). A short guide to approximation preserving reductions. In *Computational Complexity, 1997. Proceedings., Twelfth Annual IEEE Conference on (Formerly: Structure in Complexity Theory Conference)*, p. 262–273. IEEE.

Cygan, M., Fomin, F. V., Kowalik, L., Lokshtanov, D., Marx, D., Pilipczuk, M., Pilipczuk, M., e Saurabh, S. (2015). *Parameterized Algorithms*. Springer. ISBN 978-3-319-21274-6. URL http://dx.doi.org/10.1007/978-3-319-21275-3.

Diestel, R. (2010). *Graph Theory*, volume 173. Springer-Verlag, 4th edition.

Dinur, I. e Steurer, D. (2014). Analytical approach to parallel repetition. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, STOC '14, p. 624–633, New York, NY, USA. ACM. ISBN 978-1-4503-2710-7. URL http://doi.acm.org/10.1145/2591796.2591884.

Downey, R. G. e Fellows, M. R. (2013a). *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer. ISBN 978-1-4471-5558-4, 978-1-4471-5559-1.

Downey, R. G. e Fellows, M. R. (2013b). *Fundamentals of parameterized complexity*, volume 4. Springer.

Edmonds, J. e Karp, R. M. (1972). Theoretical improvements in algorithmic efficiency for network flow problems. *Journal of the ACM (JACM)*, 19(2):248–264.

Flum, J. e Grohe, M. (2006). *Parameterized Complexity Theory*. Texts in Theoretical Computer Science. Springer.

Ford, L. R. e Fulkerson, D. R. (1956). Maximal flow through a network. *Canadian journal of Mathematics*, 8(3):399–404.

Garey, M. R., Johnson, D. S., e Stockmeyer, L. (1974). Some simplified np-complete problems. In *Proceedings of the sixth annual ACM symposium on Theory of computing*, p. 47–63. ACM.

Niedermeier, R. (2006). *Invitation to Fixed-Parameter Algorithms*, volume 31. Oxford University Press.

Porschen, S. e Speckenmeyer, E. (2007). Algorithms for variable-weighted 2-sat and dual problems. In *Theory and Applications of Satisfiability Testing–SAT 2007*, p. 173–186. Springer.

Rizzi, R. (2000). NOTE - on minimizing symmetric set functions. *Combinatorica*, 20(3):445–450. URL http://dx.doi.org/10.1007/s004930070017.

Santos, V., Fernandes e Souza, U., Santos (2015). Uma introdução à complexidade parametrizada. In de Computação (SBC), S. B., editor, *Anais da 34ª Jornada de Atualização em Informática JAI 2015*, p. 232–273. XXXV Congresso da Sociedade Brasileira de Computação.