Coloração equilibrada de grafos n-Star-Clique*

Matheus R. Guedes[1], **Vinicius F. dos Santos**[1]

[1] Departamento de Ciência da Computação
Universidade Federal de Minas Gerais (UFMG) – Belo Horizonte, MG – Brazil

matheusrguedes@gmail.com, viniciussantos@dcc.ufmg.br

Abstract. *In this paper we investigate the equitable coloring problem on unipolar graphs, a superclass of split graphs. In particular, we present an algorithm based on the maximum flow which solves our problem in polynomial time, generalizing the previously known result for split graphs.*

Resumo. *Nesse artigo investigamos o problema de coloração equilibrada para grafos unipolares, uma superclasse de grafos split. Em particular, apresentamos um algoritmo baseado em fluxo máximo que soluciona esse problema em tempo polinomial, generalizando o resultado previamente conhecido para grafos split.*

1. Introdução

Problemas de coloração estão entre os mais estudados em teoria de grafos, devido à sua grande importância na resolução de muitos problemas, principalmente dentre os que requerem alocação de tarefas ou recursos. Um exemplo de aplicação é a alocação de registradores feitas pelos compiladores. O problema de coloração equilibrada é um tipo especial de problema de coloração no qual queremos uma alocação mais uniforme de recursos, uma restrição natural caso queiramos, por exemplo, distribuir tarefas entre funcionários de modo que dois funcionários façam aproximadamente a mesma quantidade de tarefas.

Os problema de coloração e coloração equilibrada são conhecidos por serem NP-completo [Karp 1972, de C. M. Gomes et al. 2018], portanto para grafos gerais não se conhece um algoritmo que, em tempo polinomial, resolva esses problemas. No entanto, para algumas classes de grafos específicas, como é o caso de árvores e grafos split, tais problemas admitem algoritmos polinomiais [Bodlaender and Fomin 2005, Chen et al. 1996, Grötschel et al. 1984].

Grafos split são aqueles cujo conjunto de vértices pode ser particionados em uma clique e um conjunto independente. Em particular, para grafos split, um algoritmo para o problema de coloração equilibrada foi apresentado por Chen, Ko e Lih [Chen et al. 1996]. O algoritmo se baseia na construção de um grafo bipartido e na determinação de um emparelhamento neste grafo, de forma que as arestas do emparelhamento definem as cores a serem atribuídas aos vértices do conjunto independente.

Neste trabalho mostramos que o problema de coloração equilibrada pode ser resolvido em tempo polinomial para grafos unipolares. Nosso algoritmo generaliza o algoritmo de Chen, Ko e Lih, utilizando fluxo em grafos para colorir grafos unipolares, que podem ser vistos como uma generalização de grafos split.

*Após a submissão deste trabalho, os autores tomaram conhecimento de que grafos n-Star-Clique já foram estudados previamente com o nome de grafos Unipolares. Em consideração ao trabalho dos revisores, os autores decidiram manter o título do artigo inalterado, mas utilizar o nome Unipolar ao longo do texto.

2. Preliminares

Definição 1. *Uma coloração de um grafo $G = (V, E)$ consiste em uma partição $S_1, \ldots, S_k$ de $V(G)$ em k conjuntos (cores), satisfazendo $\forall u, v \in S_i \Rightarrow uv \notin E(G)$. O problema de coloração é um problema de decisão que consiste em, dado um grafo G e um inteiro k, determinar se existe uma coloração de G que utilize até k cores.*

Definição 2. *Uma coloração de um grafo $G = (V, E)$ é dita equilibrada se $||S_i| - |S_j|| \leq 1, \forall i, j$. O problema de decisão correspondente é dado um grafo G e um inteiro k, determinar se existe uma coloração equilibrada de G que utilize até k cores.*

Definição 3. *Vamos chamar de unipolar a classe de todos os grafos G que aceitem uma partição de $V(G)$ em conjuntos $C, C_1, \ldots, C_n$ de modo que C seja uma clique e $\forall u \in C_i, \forall v \in C_j$, $uv \in E(G)$ se e somente se $i = j$.*

Teorema 1 ([McDiarmid and Yolov 2015])**.** *Seja G um grafo com n vértices. É possível decidir se G é um grafo unipolar em tempo polinomial.*

De fato, o algoritmo de McDiarmid and Yolov não só reconhece um grafo unipolar em tempo polinomial, como também determina uma partição de $V(G)$ em cliques $C, C_1, \ldots, C_n$ válida, em tempo $O(n^2)$ quando o grafo for unipolar.

3. Coloração equilibrada em tempo polinomial

Vamos apresentar um algoritmo que recebe um grafo G e um inteiro k e que resolve o problema de decisão da coloração equilibrada. Assumiremos que a partição $C, C_1, \ldots, C_n$ de G é dada e que $|C| \leq k$ pois, do contrário, uma coloração de G com k cores seria impossível. Seja $l = \left\lfloor \frac{|V|}{k} \right\rfloor$. Note que cada conjunto da partição terá l ou $l + 1$ vértices.

Utilizamos um algoritmo de fluxo máximo como sub-rotina em nosso algoritmo. Por questões de espaço assumimos que o leitor já se encontra familiarizado com o problema de fluxo máximo, para o qual existe um algoritmo de tempo polinomial (discutido em livros textos de algoritmos, como [Cormen et al. 2009]).

1. Crie um conjunto $A = \{a_1, \ldots, a_k\}$ de k vértices auxiliares e conecte o vértice fonte s a cada $a_i \in A$ por uma aresta de capacidade l.
2. Para cada vértice $v_i \in C$, $1 \leq i \leq |C|$ adicione uma aresta $a_i v_i$ de capacidade 1.
3. Crie um conjunto de vértices auxiliares $W = \{w_{ij}, 1 \leq i \leq k, 1 \leq j \leq n\}$
4. Para cada vértice $a_i \in A$ adicione uma aresta $a_i w_{ij}$ de capacidade 1 para cada $1 \leq j \leq n$.
5. Para cada $w_{ij} \in W$ adicione uma aresta de capacidade 1 entre cada w_{ij} e cada $c_k \in C_j$ se $c_k \notin N(v_i)$.
6. Conecte cada vértice $v \in C \cup C_1 \cup \ldots \cup C_n$ ao vértice terminal t por uma aresta de capacidade 1.
7. Execute um algoritmo de fluxo máximo de s para t.
8. Se fluxo $\neq k \cdot l$ retorne impossível.
9. No grafo residual incremente em 1 a capacidade de cada aresta incidente a s.
10. Execute um algoritmo de fluxo máximo de s para t.
11. Se fluxo $= |V(G)|$ então temos uma coloração equilibrada com k cores, caso contrário retorne impossível.

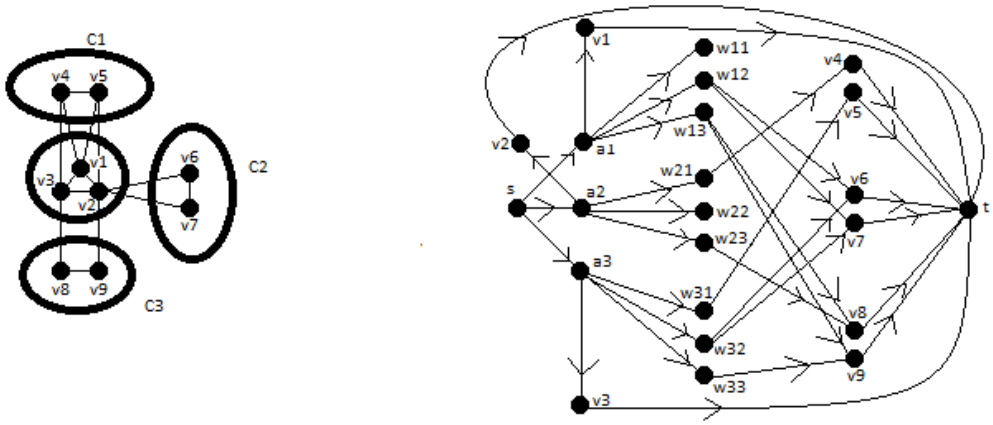

Figura 1. Um grafo unipolar e a rede montada pelo algoritmo.

Lema 1. *Se fluxo* $= |V(G)|$ *após o passo 11 então* G *admite coloração com* k *cores.*

Demonstração. Cada vértice auxiliar de A representa uma cor. Cada unidade de fluxo que sai de um vértice a_i passa por algum vértice do grafo original antes de chegar ao sumidouro t. Atribua a tais vértices a i-ésima cor. Para certificar que não escolhemos vértices adjacentes para uma mesma cor observe que cada $v_i \in C$ possui no máximo um correspondente em A que é o a_i, em outras palavras fixamos a cor de cada vértice $v_i \in C$ com cores distintas. Para que o restante dos vértices da cor a_i não sejam vizinhos de v_i criamos vértices auxiliares w_{ij} cuja função é limitar o algoritmo a escolher no máximo 1 vértice de cada C_j e que não seja vizinho de v_i quando $i \leq |C|$. Como o grau de entrada de w_{ij} é 1 garantimos que não pegaremos dois vértices distintos de C_j para a cor i. Durante a construção não colocamos arestas entre w_{ij} e $N(v_i)$ assim não existe caminho que começa em a_i e chega em um vizinho de v_i. Para verificar que atribuímos a cada vértice exatamente uma cor basta observar que qualquer caminho de s para t deve passar por algum vértice do grafo original e que como as arestas de tais vértices para t tem peso 1, só podemos escolhê-lo uma única vez.

Nos resta demonstrar que tal coloração é equilibrada. Como fixamos o grau de entrada de cada a_i como sendo l garantimos inicialmente que cada cor será atribuída a no máximo l vértices. No passo 8, ao fazer a verificação, o fluxo será $l \cdot k$ e portanto temos uma coloração parcial de $l \cdot k$ vértices do grafo original em que cada cor possui l vértices. Ao aumentar em 1 a capacidade das arestas e executar o fluxo é garantido que nenhuma cor terá menos que l elementos, pois nenhum caminho aumentante volta à origem s. □

Lema 2. *Se* G *admite coloração com* k *cores então o algoritmo tera fluxo* $= |V(G)|$ *após o passo 11.*

Demonstração. Dada uma coloração equilibrada $S_1, \ldots, S_k$ de G mostraremos que o fluxo de s para t será pelo menos $|V(G)|$ unidades. Criamos o grafo feito pelo algoritmo e vamos determinar o fluxo que passa por cada aresta. Para cada vértice auxiliar a_i fornecemos $|S_i|$ unidades de fluxo. O vértice a_i fornecerá uma unidade de fluxo para cada vértice de S_i. Com cada vértice do grafo original recebendo uma unidade de fluxo basta perceber que nenhuma unidade de fluxo passa por dois vértices distintos do grafo original e portanto ele chega inalterado a t e temos fluxo de pelo menos $|V(G)|$ unidades.

Como o conjunto das arestas que ligam os vértices do grafo original a t são arestas de corte, concluímos que o fluxo tem tamanho exatamente $|V(G)|$. □

Teorema 2. *O problema de coloração equilibrada pode ser resolvido em tempo polinomial para grafos unipolares.*

Demonstração. Utilizando o algoritmo anterior que funciona devido aos Lemas 1 e 2, obtemos um algoritmo polinomial para o problema de coloração equilibrada de grafos unipolares. □

Para manter a concisão do algoritmo, a forma como ele é apresentado não define, explicitamente, uma coloração válida. Entretanto, tal coloração pode ser determinada facilmente a partir da rede residual construída pelo algoritmo de fluxo executado no passo 10. Para isso basta lembrar que um vértice v pertence a cor i se somente se o vértice a_i fornece alguma unidade de fluxo para o vértice v.

4. Considerações finais

Neste trabalho mostramos que o problema de coloração equilibrada em grafos unipolares pode ser resolvido em tempo polinomial por meio de um algoritmo de fluxo em um grafo construído a partir da instância G, k.

Um problema natural seria tentar generalizar o resultado para superclasses dos grafos unipolares. Note que, se contrairmos todas as cliques da partição, obtemos uma estrela. Seria possível generalizar o resultado para grafos que pudessem ser particionados em cliques cuja contração fornecesse uma árvore qualquer?

Referências

Bodlaender, H. L. and Fomin, F. V. (2005). Equitable colorings of bounded treewidth graphs. *Theoretical Computer Science*, 349(1):22 – 30. Graph Colorings.

Chen, B.-L., Ko, M.-T., and Lih, K.-W. (1996). Equitable and m-bounded coloring of split graphs. In Deza, M., Euler, R., and Manoussakis, I., editors, *Combinatorics and Computer Science*, pages 1–5, Berlin, Heidelberg. Springer Berlin Heidelberg.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2009). *Introduction to Algorithms, Third Edition*. The MIT Press, 3rd edition.

de C. M. Gomes, G., Lima, C. V. G. C., and dos Santos, V. F. (2018). Parameterized complexity of equitable coloring. *Discrete Mathematics and Theoretical Computer Science*. Artigo aceito para publicação.

Grötschel, M., Lovász, L., and Schrijver, A. (1984). Polynomial algorithms for perfect graphs. In Berge, C. and Chvátal, V., editors, *Topics on Perfect Graphs*, volume 88 of *North-Holland Mathematics Studies*, pages 325 – 356. North-Holland.

Karp, R. M. (1972). *Reducibility among Combinatorial Problems*, pages 85–103. Springer US, Boston, MA.

McDiarmid, C. and Yolov, N. (2015). Recognition of unipolar and generalised split graphs. *Algorithms*, 8(1):46–59.