

Seleção de Representantes para Cobertura de Componentes Conexas em Grafos

Anderson Lemos¹, Vinicius F. dos Santos¹, Olga Goussevskaia¹

¹Departamento de Ciência da Computação
Universidade Federal de Minas Gerais (UFMG)
Belo Horizonte – MG – Brasil

{andersonlemons, viniussantos, olga}@dcc.ufmg.br

Abstract. We introduce a new covering problem, called *Component Cover by Vertices (CCV)*. In this problem, we are given a graph G and a partition A, B of $V(G)$, and our goal is to find the smallest subset B' of B such that, for every connected component C of $G[A]$, there is a vertex $v \in C$ with some neighbor in B' . We study the complexity of this problem and give positive and negative results. We show that the CCV problem is NP-Complete in several classes of graphs. We also show that the problem is hard to approximate and parameterize. On the other hand, we present polynomial algorithms for other classes of graphs. Finally, we show FPT parameterizations and approximation algorithms for certain classes of graphs.

Resumo. Introduzimos um novo problema de cobertura chamado *Cobertura de Componentes por Vértices (CCV)*. Neste problema são dados um grafo G e uma partição A e B de $V(G)$, e nosso objetivo é encontrar o menor subconjunto B' de B tal que, para cada componente conexa C de $G[A]$, há um vértice $v \in C$ com algum vizinho em B' . Estudamos a complexidade deste problema e damos resultados positivos e negativos. Mostramos que o problema CCV é NP-Completo para várias classes de grafos. Mostramos também que o problema é difícil de aproximar e de parametrizar. Por outro lado, apresentamos algoritmos polinomiais para outras classes de grafos. Por fim, mostramos parametrizações FPT e algoritmos aproximativos para determinadas classes de grafos.

1. Introdução

Neste trabalho introduzimos um novo problema de cobertura chamado *Cobertura de Componentes por Vértices (CCV)*, que pode ser definido como a seguir: dado um grafo G e uma partição A, B de $V(G)$, nosso objetivo é encontrar o menor subconjunto $B' \subseteq B$ tal que cada componente conexa do subgrafo induzido por A tenha pelo menos um vizinho em B' . Assumimos que B é sempre uma solução válida para o problema, pois, caso contrário, o problema não admitiria solução.

Um exemplo de aplicação deste problema foi estudado em [Ferreira Neto et al. 2017], com ênfase nas propriedades de conectividade em redes sem fio obstruídas, aspectos geométricos e distribuições aleatórias de nós. O cenário estudado consiste em um conjunto de dispositivos de comunicação e um conjunto de pontos de acesso a um *backbone* que fornece conectividade à internet. Os dispositivos têm um raio de transmissão e destinam-se a formar uma topologia conexa. Em cenários

onde o raio de transmissão não tem alcance suficiente para conectar todos os dispositivos uns aos outros, ou seja, a rede de dispositivos é dividida em várias componentes conexas, os pontos de acesso se tornam úteis. Dois dispositivos podem se comunicar se houver um caminho entre eles, isto é, se ambos pertencerem a mesma componente conexa, ou se houver um caminho entre cada dispositivo e algum ponto de acesso do *backbone*. O objetivo é, então, encontrar o menor subconjunto de pontos de acesso de modo que todos os dispositivos estejam conectados a pelo menos um ponto de acesso. Mapeando este problema para o CCV, o conjunto A seria composto pelos dispositivos, e o conjunto B seria composto pelos pontos de acesso. Pares de vértices de A e B seriam conectados por arestas se os dispositivos ou pontos de acesso correspondentes estivessem dentro do alcance de comunicação um do outro. Queremos então encontrar o menor subconjunto B' de pontos de acesso, para que todos os dispositivos em A estejam conectados a pelo menos um ponto de acesso em B' por meio de um caminho.

Classes de grafos podem ser definidas como um conjunto de grafos que satisfazem uma propriedade específica dessa classe. Por exemplo, a classe de grafos cúbicos é composta de grafos cujos todos os vértices têm grau exatamente igual a 3. Muitos problemas NP-Completo para grafos gerais possuem algoritmos polinomiais para certas classes de grafos. Outros, no entanto, são difíceis, mesmo para grafos muito específicos. Neste trabalho mostramos que o problema CCV é NP-Completo em várias classes de grafos: cordais, bipartidos com grau máximo 3, grafos de disco unitário planares e grades. Além disso, mostramos que o problema é difícil se parametrizado pelo tamanho da solução e apresentamos um resultado de inaproximabilidade. No entanto, mostramos algoritmos polinomiais para algumas classes de grafos: árvores, grafos bloco, cografos, cactos e grafos *outerplanares*. Mostramos que o problema é FPT se parametrizado por largura arbórea (*treewidth*) ou por tamanho da solução e grau máximo. Por fim, mostramos dois algoritmos aproximativos para grafos gerais e um algoritmo aproximativo de fator constante para grafos de disco unitário.

2. NP-Completo

Foram encontrados vários resultados negativos e positivos. Muitos deles, utilizam o fato de que a remoção das arestas de B e a contração das arestas de A não alteram a solução do problema. O grafo resultante dessas operações é um grafo bipartido em que as partes da bipartição são exatamente os conjuntos A e B , ou seja, uma instância de *Red-Blue Dominating Set* (RBDS). Nesse problema, nos é dado um grafo bipartido, onde as partes da bipartição são denominadas como vértices azuis e vértices vermelhos. Queremos então, cobrir todos os vértices vermelhos com um conjunto mínimo de vértices azuis. Também utilizamos o fato de que CCV é equivalente a *Set Cover* (SC), pois podemos reduzir um problema ao outro. No problema SC, nos é dado um conjunto de elementos X e uma família \mathcal{Y} de subconjuntos de X , e queremos cobrir X com um mínimo subconjunto de \mathcal{Y} , isto é, queremos encontrar um subconjunto $\mathcal{Y}' \subseteq \mathcal{Y}$, tal que, $\forall x \in X$ existe um $Y \in \mathcal{Y}'$ tal que $x \in Y$.

A ideia de redução de SC para CCV é a seguinte: se X é o conjunto de elementos que se quer cobrir no SC, podemos pensar em uma redução em que elementos de X são vértices de A e os subconjuntos de X que podem ser usados para cobrir são vértices de B . Além disso, há arestas de um vértice de B para um de A se o elemento representado pelo vértice de A pertencia ao subconjunto representado pelo vértice de B . Assim, se

uma solução cobre todos os elementos de X , se selecionarmos os vértices de B que representam os conjuntos dessa solução, ele será uma solução para CCV. É fácil ver que a transformação de uma solução de CCV para uma solução de SC também é válida, utilizando a mesma estratégia. Também é fácil ver que podemos transformar uma instância de CCV para SC contraindo as arestas de A , removendo as arestas entre os vértices de B e fazendo a mesma coisa da redução anterior: vértices de A se transformam em elementos, vértices de B se transformam em subconjuntos, que têm os elementos que representam seus vizinhos que são vértices de A .

O problema CCV é NP-Completo para diversas classes de grafos. Uma delas é classe dos grafos bipartidos com grau máximo 3. Para mostrar isso, repare que a redução de SC para CCV é uma prova de NP-Completude para o problema CCV, e o grafo resultante já é bipartido. Repare que o grau de um vértice $a \in A$ que representa um elemento $x \in X$ é dado pelo número de conjuntos a qual x pertence, e o grau de um vértice $b \in B$ que representa um conjunto $Y \in \mathcal{Y}$ é dado pelo número elementos que Y possui. Assim, se fizemos uma redução a partir do problema *3-Set Cover*, onde os conjuntos tem tamanho no máximo 3, e que é NP-Completo [Duh and Fürer 1997], teremos um grafo onde o grau dos vértices de B é no máximo 3. No entanto, se um vértice de $a \in A$ tem grau maior que três, podemos primeiramente criar dois vértices auxiliares a_2 e a_3 que estarão conectados entre si e a_2 estará conectado a a . Depois, desconectamos dois dos vizinhos de a que pertencem a B e os conectamos a a_3 . Assim, a_2 terá grau 2, a_3 terá grau 3 e a terá diminuído seu grau em uma unidade. Repare que a , a_2 e a_3 ainda fazem parte da mesma componente conexa, e todos os vértices de B que poderiam cobrir tal componente, continuam podendo. Podemos repetir o processo até que nenhum vértice de A tenha grau maior que três, e assim teremos um grafo onde todos os vértices tem grau no máximo 3. Logo, CCV é NP-Completo para grafos bipartidos com grau máximo 3.

O problema CCV também é NP-Completo para grafos *split* e, conseqüentemente, para grafos cordais. Para mostrar isso, basta notar que adicionar arestas entre vértices de B não altera a solução. Se obtermos o grafo gerado a partir da redução de SC para CCV e adicionarmos todas as arestas possíveis entre os vértices de B , o conjunto B será uma clique, enquanto A é um conjunto independente, ou seja, um grafo *split*. Logo, CCV é NP-Completo para grafos *split*. Como todo grafo *split* é cordal, CCV é NP-Completo para grafos cordais.

Um grafo de disco unitário (do inglês, *Unit Disk Graph*), ou simplesmente UDG, é um grafo em que os vértices podem ser representados por discos (círculos) em um plano, tal que todos os discos tem o mesmo raio e há arestas entre dois vértices se há interseção entre os discos que representam tais vértices (assume-se que discos tangentes têm interseção) [Brandstadt et al. 1999, Bondy et al. 1976]. Um grafo é dito planar se é possível desenhar o mesmo em um plano de tal forma que nenhum par de arestas se cruze [Bondy et al. 1976]. O problema CCV é NP-Completo para UDGs planares. Essa redução é feita a partir do problema *Vexter Cover* (VC). Nesse problema, nos é dado um grafo G e queremos cobrir todas as arestas de G com um conjunto mínimo de vértices, ou seja, queremos encontrar um subconjunto $V' \subseteq V(G)$ tal que V' é mínimo e para cada $uv \in E(G)$ temos que $\{u, v\} \cap V' \neq \emptyset$. Esse problema é NP-Completo mesmo que o grafo de entrada seja planar com grau máximo 3 [Bar-Yehuda and Even 1985]. Um grafo planar com grau máximo igual a 4 pode ser desenhado em um plano usando uma área de

tamanho $O(n)$ tal que todos os vértices estão em coordenadas inteiras do plano e todas as arestas podem ser desenhadas sobre as linhas do plano [Clark et al. 1990]. Desenhamos o grafo original no plano dessa forma. Para criar o novo grafo da redução, copiamos os vértices do grafo original para B e os desenhamos em um outro plano, sendo que, como esses vértices devem ser discos, seus centros serão exatamente nas coordenadas em que estão situados no plano original e terão raio igual a $\frac{1}{4}$. Após isso, substituiremos as arestas do grafo original por componentes conexas formadas por vértices intermediários que são desenhados seguindo o caminho que a aresta segue no plano original e como esses vértices são discos, eles terão raio igual a $\frac{1}{4}$ e serão colocados sempre tangentes aos vértices vizinhos. Esses vértices estarão no conjunto A , assim, formam uma única componente conexa. É fácil ver que cada componente conexa criada representa uma aresta do grafo original e que os problemas são equivalentes. Também vale frisar que o grafo gerado é um UDG planar, logo, CCV é NP-Completo para UDGs planares.

O problema CCV também é NP-Completo para grades. Na redução acima, o grafo gerado não só é um UDG planar, como também é uma grade parcial. Para reduzir o problema a uma grade, basta adicionar os vértices inexistentes e colocá-los em B (como não foram inseridas arestas, novas soluções não foram criadas) e depois inserir duas novas colunas entre cada par de colunas (o mesmo vale para as linhas). Os vértices não vizinhos de nenhum vértice do grafo original, farão parte de B . Os vértices entre cada par de vértices do grafo original serão escolhidos de acordo com o que existia no grafo original (depende de qual conjunto eles pertenciam e se havia aresta entre eles). Assim, podemos inserir vértices fazendo uma correspondência um para um de componentes dos grafos e não criaremos soluções melhores que as originais. Esse resultado é interessante pois essa classe de grafos não é fechada para as operações de remoção e contração de arestas, o que implica que o resultado não pode ser diretamente aplicado ao problema RBDS.

3. Algoritmos polinomiais exatos

Por outro lado, foram encontrados algoritmos para outras classes de grafos: árvores, blocos, cografos, cactos e *outerplanares*. Em todos eles, podemos contrair as arestas entre os vértices de A e remover as arestas entre os vértices de B . Nas árvores, teremos sempre algum vértice folha. Se ele pertencer a A , então o único vértice adjacente tem que estar na solução, pois é o único que o cobre. Se pertencer a B , e não existir outro vértice de B na árvore, ele é a solução. Caso contrário, podemos descartá-lo pois ele cobriria apenas uma componente que pode ser coberta por outro vértice de B . Para blocos, após as contrações e remoções de arestas, o grafo resultante é uma árvore, e podemos aplicar o mesmo algoritmo. Para cografos, se o grafo não for conexo, podemos aplicar o algoritmo para cada componente separadamente. Se o grafo for conexo, a última operação feita para construí-lo foi uma junção, ou seja, ou algum vértice de A está conectado a todos os outros e forma uma única componente que pode ser coberta por um único vértice de B , ou um vértice de B está conectado a todos os vértices de A e pode ser usado na solução.

Para cactos, se aplicarmos as regras que aplicamos para árvores, teremos um grafo apenas com ciclos de tamanho par com no máximo um vértice em comum. Chamaremos os vértices que pertencem a mais de um ciclo de vértices de corte. Como o grafo continua sendo um cacto sempre há um ciclo com um único vértice de corte. Podemos escolher, desse ciclo, os vértices de B alternadamente, que cobriremos todos vértices de A do ciclo. Se o vértice de corte pertencer a A , não precisamos cobri-lo com os vértices desse ciclo,

pois outro vértice poderá cobri-lo e cobrir outras componentes de A . Se o vértice de corte pertencer a B , é sempre melhor escolher os vértices da solução desse ciclo iniciando do vértice de corte e colocando-o na solução, pois assim, poderemos cobrir mais vértices que os que estão nesse ciclo. Por fim, para grafos *outerplanares*, se utilizarmos as regras utilizadas para cactos, o grafo restante será *outerplanar* com ciclos biconectados. Como o grafo é *outerplanar*, haverá sempre um ciclo que tem vértices em comum com apenas outro ciclo. Nesse caso, cobriremos todos os vértices de A desse ciclo que não são vértices de corte. Assim como anteriormente, devemos escolher os vértices de B desse ciclo alternadamente. Se o tamanho do ciclo for um múltiplo de 4, é melhor escolher o vértice de corte que está em B para a solução e cobrir o ciclo todo, se não, não o escolhemos e cobrimos apenas a parte do ciclo sem os vértices de corte, e isso nos dá a solução ótima.

4. Parametrização e Aproximação

Sobre algoritmos de complexidade parametrizada, o problema CCV é $W[2]$ -Completo se parametrizado pelo tamanho da solução. Isso pode ser demonstrado fazendo uma redução parametrizada a partir do RBDS, que por sua vez é $W[2]$ -Completo [Garnero et al. 2017]. Por outro lado, como RBDS é FPT para grafos planares se parametrizado pelo tamanho da solução [Garnero et al. 2017], fazendo uma redução contrária, de CCV para RBDS, demonstramos que CCV é FPT para grafos planares se parametrizado pelo tamanho da solução. O mesmo acontece se parametrizarmos CCV pela largura arbórea (*treewidth*): como RBDS é FPT se parametrizado pela largura arbórea [Alber et al. 2000], reduzindo CCV à RBDS, demonstramos que CCV é FPT se parametrizado por largura arbórea. Por fim, mostramos que CCV é FPT se parametrizado por dois parâmetros combinados: tamanho da solução e grau do grafo. Para isso, encontramos um *kernel* com $O(2^{kd})$ vértices (sendo k o tamanho da solução e d o grau do grafo). A ideia dessa demonstração é que pelo grau do grafo ser limitado a d , vértices de B cobrem no máximo kd componentes de A e vértices de B que cobrem as mesmas componentes podem ser excluídos (deixando apenas um deles), e se houver apenas kd componentes de A , existem menos que 2^{kd} conjuntos diferentes dessas componentes.

A respeito de algoritmos aproximativos, o *Set Cover* (SC) é inaproximável a um fator melhor que $O(\log n)$, sendo n o número de elementos a serem cobertos [Feige 1998]. Se reduzirmos SC a CCV, teremos que o número c de componentes que se quer cobrir é o número de elementos do SC. Logo, CCV é inaproximável a um fator de aproximação melhor que $O(\log c)$. Por outro lado, se reduzirmos CCV a SC e utilizamos o algoritmo guloso ou o algoritmo de camadas, ambos mostrados em [Vazirani 2013], teremos algoritmos para CCV com fatores de aproximação $O(\log c)$ e Δ_A , respectivamente (onde Δ_A é o máximo número de vértices de B que pode cobrir uma única componente de A). Por fim, construímos um algoritmo aproximativo para grafos de disco unitário. No trabalho de [Wu et al. 2006] é mostrado que, em um UDG, a vizinhança de um vértice contém um conjunto independente de tamanho no máximo 5. Logo, após contrair as arestas de A de um UDG, cada vértice de B tem no máximo 5 vizinhos de A . Ao reduzir CCV em UDGs para SC, temos que o tamanho dos conjuntos é no máximo 5, ou seja, o problema *5-Set Cover*. A partir disso, podemos utilizar o algoritmo desenvolvido por [Levin 2008], que tem um fator de aproximação igual a $H_d - \frac{196}{390}$ para o problema *d-Set Cover* — sendo H_d o d -ésimo número harmônico, isto é, $H_d = \sum_{i=1}^d \frac{1}{i}$. Como nesse caso $d = 5$, teremos um fator de aproximação $\approx 1,7807 < 1,79$.

5. Conclusão

Pode-se concluir que o problema CCV é bastante complexo, pois é NP-Completo para grafos muito específicos, como UDGs planares e grades. Também é difícil construir algoritmos eficientes para o problema geral, dadas as provas de dificuldade de obtenção de algoritmos aproximados ou parametrizados. Por outro lado, apresentamos algoritmos polinomiais para diversas classes de grafos e um algoritmo de aproximação de fator constante para UDGs, o que pode indicar que algoritmos eficientes podem ser encontrados para outras classes de grafos. Observamos que os resultados positivos podem ser aplicados automaticamente ao *Red-Blue Dominating Set* e esperamos que isso possa motivar novas pesquisas sobre os dois problemas. Atualmente estamos desenvolvendo uma versão do trabalho para submissão em forma de artigo completo em um periódico.

Referências

- Alber, J., Bodlaender, H. L., Fernau, H., and Niedermeier, R. (2000). Fixed parameter algorithms for planar dominating set and related problems. In *Scandinavian Workshop on Algorithm Theory*, pages 97–110. Springer.
- Bar-Yehuda, R. and Even, S. (1985). A local-ratio theorem for approximating the weighted vertex cover problem. *North-Holland Mathematics Studies*, 109:27–45.
- Bondy, J. A., Murty, U. S. R., et al. (1976). *Graph theory with applications*, volume 290. Citeseer.
- Brandstadt, A., Spinrad, J. P., et al. (1999). *Graph classes: a survey*, volume 3. Siam.
- Clark, B. N., Colbourn, C. J., and Johnson, D. S. (1990). Unit disk graphs. *Discrete mathematics*, 86(1-3):165–177.
- Duh, R.-c. and Fürer, M. (1997). Approximation of k-set cover by semi-local optimization. In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, pages 256–264. ACM.
- Feige, U. (1998). A threshold of $\ln n$ for approximating set cover. *Journal of the ACM (JACM)*, 45(4):634–652.
- Ferreira Neto, M., Goussevskaia, O., and dos Santos, V. F. (2017). Connectivity with backbone structures in obstructed wireless networks. *Computer Networks*, 127:266 – 281.
- Garnero, V., Sau, I., and Thilikos, D. M. (2017). A linear kernel for planar red–blue dominating set. *Discrete Applied Mathematics*, 217:536–547.
- Levin, A. (2008). Approximating the unweighted k-set cover problem: greedy meets local search. *SIAM Journal on Discrete Mathematics*, 23(1):251–264.
- Vazirani, V. V. (2013). *Approximation algorithms*. Springer Science & Business Media.
- Wu, W., Du, H., Jia, X., Li, Y., and Huang, S. C.-H. (2006). Minimum connected dominating sets and maximal independent sets in unit disk graphs. *Theoretical Computer Science*, 352(1-3):1–7.