Lights Out em grafos: apagando luzes da menor maneira

Fernanda Couto[1], **Ighor Bruno de Brito**[1], **Willian de Assis Sento Sé**[1]

[1] Universidade Federal Rural do Rio de Janeiro (UFRRJ)

fernandavdc@ufrrj.br, ighor.bruno.brito@gmail.com, will.asse@outlook.com

Abstract. *Lights Out is an eletronic puzzle whose goal is to find a set of buttons that, when pressed, turns off all the lights on the board. In this work, we consider the generalization of the game whose input is a graph and we focus on its minimize version, whose goal is to determine the minimum number of vertices that must be activated in order to turn all graph lights off. We deal with Caterpillar graphs, Flower Snarks and Goldberg Snarks. To help us on obtaining the results, a software that simulates the game for any input data has been implemented as well as a brute-force algorithm that solves the minimize version of Lights Out game.*

Resumo. *Lights Out é um quebra-cabeça eletrônico cujo objetivo é encontrar um conjunto de botões que, quando pressionados, apaguem todas as luzes do tabuleiro. Neste trabalho, consideramos a generalização cuja entrada é um grafo e focamos na versão de minimização, onde o objetivo é determinar o número mínimo de vértices que devem ser acionados para apagar todo o grafo. Trabalhamos com grafos Caterpillar, Snarks Flor e Snarks Goldberg. Para auxiliar na obtenção dos resultados, implementamos um software que simula o jogo para qualquer grafo de entrada e também um algoritmo força-bruta que soluciona a versão de minimização.*

1. Introdução

Lights Out é um jogo de quebra-cabeça que consiste, originalmente, em uma matriz de botões (grid) 5×5 onde cada posição possui apenas 2 estados: *aceso*, denotado por 1, ou *apagado*, denotado por 0. Ao pressionar um botão, seu estado e o estado dos botões adjacentes são alternados. O jogo tem início com uma matriz com estados aleatórios, chamada de *configuração inicial*, e o objetivo do jogador é desligar todos os elementos, chamada de *configuração final*. Existem muitas variantes para este jogo que vão desde uma mudança no formato do quebra-cabeças ou de possíveis configurações iniciais, até diferentes objetivos a serem alcançados. Podemos, por exemplo, ao invés de usar uma entrada em forma de grid, aplicar as mesmas regras a grafos de outras classes [Berman et al. 2019]. Além disso, pode-se desejar minimizar o conjunto de vértices que, quando acionados, conduzem a uma configuração final. Sabe-se que todo grafo possui ao menos uma solução para o *Lights Out* quando iniciado com todos os vértices apagados. Além disso, dada uma solução, a configuração final é alcançada independentemente da ordem em que os vértices forem pressionados [Fleischer and Yu 2013]. Entretanto, determinar o conjunto mínimo de vértices que, ao serem pressionados, conduzem à configuração final é um problema NP-difícil [Sutner 1988]. Por isso, alguns trabalhos da literatura restringem o estudo desta versão de minimização a algumas classes de grafos, como: *Caterpillars,* *Caminhos, Ciclos e Grafos Bipartidos Completos*. Neste trabalho, realizamos um estudo deste problema de minimização em duas classes de grafos, a saber: Caterpillars, onde obtivemos alguns avanços em relação ao estado da arte, e em duas famílias de Snarks, os *Snarks Flor* e os *Snarks Goldberg*.

1.1. Definições Gerais

Seja $G = (V, E)$ um grafo, onde $n = |V|$ e $m = |E|$. Cada vértice $v \in V$ possui um estado $e_v \in \{0, 1\}$, onde 0 e 1 indicam que o vértice está, respectivamente, *apagado* ou *aceso*. O ato de pressionar um vértice v é chamado de *movimento* e alterna o estado de todos os vértices $u \in N_G[v]$ de 0 para 1 ou vice-versa. Iniciando com todos os vértices apagados, $C_{inicial} = \{e_v = 0 \mid v \in V\}$, desejamos um conjunto de movimentos $X \subseteq V$, chamado de *solução*, que, quando executados, acendam todos os vértices do grafo e tenha tamanho mínimo. Seja $S(G)$ o conjunto de todas as possíveis soluções para um grafo G inicialmente apagado e $S^*(G)$ uma solução mínima.

Proposição 1. *Seja G um grafo com n vértices. Então* $\lceil \frac{n}{\Delta+1} \rceil \leq |S^*(G)|$.

Note que para grafos completos, grafos ciclo C_{3n}, grafo de Petersen (Fig. 1 (b)), o limite descrito na Proposição 1 é justo. Haveria outras classes onde o mesmo ocorre? Seriam Snarks grafos para os quais este limite inferior é justo?

Um *Snark* é um grafo conexo, cúbico, classe 2 e que não possui pontes. Um grafo é classe 2 quando o menor número de cores para colorir sua arestas é $\Delta + 1$, onde Δ é o maior grau do grafo. Uma ponte é uma aresta que ao ser removida aumenta o número de componentes conexas do grafo.

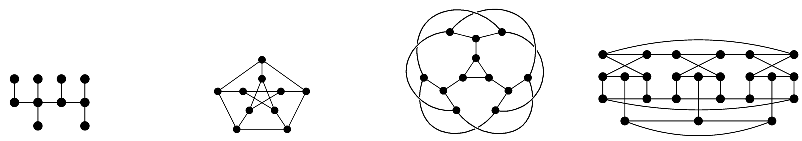

Figura 1. (a) Caterpilar (b) Petersen (c) Snark Flor F_3 (d) Snark Goldberg G_3

Focaremos em duas famílias de Snarks chamadas Snarks Flor e Goldberg [Isaacs 1975]. Um *Snark Flor* (Fig. 1 (c)) é denotado por F_i, onde i é ímpar e $i \geq 1$, e tem sua construção definida por i blocos de 4 vértices. Cada bloco B_i (Fig. 2) (a) consiste em três vértices u_i, x_i e y_i interligados por um vértice central v_i. Cada bloco é conectado pelas arestas de conexão. Neste trabalho, mostramos que o limite inferior apresentado na Proposição 1 é justo para Snarks Flor e exibimos uma solução mínima. Um *Snark Goldberg* é denotado por G_i, onde i é ímpar e $i \geq 3$. Eles são formados através da ligação entre i blocos pré-definidos de 8 vértices. O bloco B_i do grafo Goldberg é o grafo que tem como conjunto de vértices é $V(B_i) = \{a_i, b_i, c_i, d_i, e_i, f_i, g_i, h_i\}$ e seu conjunto de arestas é $E(B_i) = \{a_ib_i, a_ie_i, b_ic_i, c_id_i, c_if_i, d_ih_i, d_ie_i, e_ig_i, f_ig_i\}$ como indicado na Figura 2 (b). O grafo final é obtido pelas seguintes arestas de ligação $E_{k,j} = \{b_ka_j, g_kf_j, h_kh_j\}$ entre os respectivos blocos B_{i-1} e B_i. Mais informações sobre a construção dos Snarks Flor e Goldberg em [Isaacs 1975]. Snarks Goldberg são exemplos de Snarks para o qual o limite inferior dado pela Proposição 1 não é justo.

Além do estudo em Snarks, obtivemos avanços com relação ao estado da arte do problema da minimização para Grafos Caterpillar. *Grafos Caterpillar*, (Fig. 1 (a)) são árvores onde cada vértice ou pertence a um caminho P, denominado *espinha*, ou está à distância 1 de P, chamado *folha*.

Figura 2. (a) Bloco do Snark Flor B_i (b) Bloco do Snark Goldberg G_i

2. Lights Out em Caterpillars

De acordo com nosso conhecimento da literatura, a versão de minimização do jogo Lights Out quando restrito à classe de grafos Caterpillar já foi solucionada em alguns casos de acordo com o Teorema 1.

Teorema 1. *[Fleischer and Yu 2013] Seja G um grafo caterpillar com e vértices na espinha cada um com f folhas. Então:*

$$|S^*(G)| = \begin{cases} \lceil \frac{e}{3} \rceil + f \lfloor \frac{2e}{3} \rfloor, & f \textit{ par} \\ ef, & f \textit{ ímpar e } e \textit{ par} \\ \frac{1}{2}(e-1)(f+1)+1, & e, f \textit{ ímpares} \end{cases}$$

Em nosso estudo da versão de minimização para grafos caterpillar, consideramos o caso onde a quantidade de folhas em cada vértice da espinha alterna entre ímpares e pares. Como a quantidade folhas é variável, os resultados a seguir exibem o conjunto S^*, e não sua cardinalidade. Neste caso, o conjunto S^* é obtido pela seleção dos vértices correspondentes a sequências descritas da seguinte forma: EF, onde E significa "acionar o vértice da espinha correspondente" e F significa "acionar todas as folhas do vértice da espinha correspondente". Dada uma sequência, ela deve ser aplicada da esquerda para direita ou vice-versa no caterpillar, sucessivas vezes, até acendermos todo o grafo. Denotaremos por $(EF)_l$ se a sequência EF é aplicada da esquerda para a direita, e $(EF)_r$, caso contrário.

Teorema 2. *Seja G um grafo caterpillar com e vértices na espinha, onde a quantidade de folhas em cada vértice de e alterna entre números ímpares e pares, nesta ordem. Então $S^*(G)$ é obtido por sucessivas aplicações da sequência:*

$$\begin{cases} (FFEF)_l, & \textit{se } e \bmod 4 \equiv 0 \quad (Fig.3) \\ (EFFF)_l, & \textit{se } e \bmod 4 \equiv 2 \\ (EFFF)_l \textit{ ou } (EFFF)_r, & \textit{caso contrário} \end{cases}$$

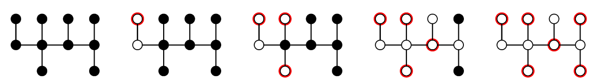

Figura 3. Aplicação da sequência $FFEF$ para obter o menor conjunto solução do Lights Out para um caterpillar com $e = 4$ e paridade de folhas alternadas, iniciando em ímpar. Os vértices em vermelho são os selecionados.

Teorema 3. *Seja G um grafo caterpillar com e vértices na espinha, onde a quantidade de folhas em cada vértice de e alterna entre números pares e ímpares, nesta ordem. Então $S^*(G)$ é obtido por sucessivas aplicações da sequência:*

$$\begin{cases} (FEFF)_l, & \textit{se } e \bmod 4 \equiv 0 \\ (EF)_l, & \textit{se } e \bmod 4 \equiv 1 \\ (EFFF)_r, & \textit{se } e \bmod 4 \equiv 2 \\ (FEFF)_l \textit{ ou } (EF)_l, & \textit{caso contrário} \end{cases}$$

O Lema 4 é o argumento base para garantir a minimalidade das soluções obtidas através dos Teoremas 2 e 3.

Lema 4. *Sejam G um caterpillar, v um vértice da espinha de G e $F(v)$ o conjunto de todas as folhas de v. Em qualquer solução $S_i \in S(G)$ ou $v \in S_i$ ou $F(v) \subseteq S_i$.*

3. Lights Out em famílias de Snarks

Como visto na Seção 1, Snarks Flor F_i, i é ímpar, são construídos por blocos B_i (Fig. 2 (a)). Cada um desses blocos possui um vértice central v_i que, ao ser pressionado, acende todos os vértices do bloco e não alterna o estado de vértices de outros blocos do grafo. Além disso, como um Snark Flor possui $4i$ vértices, temos que o limite inferior apresentado na Proposição 1 é justo para a solução apresentada no Teorema 5, o que garante sua minimalidade.

Teorema 5. *Seja F_i um Snark Flor, construído por blocos B_i com vértices centrais denotados por v_i. Então, $S^*(F_i) = \{v_i\}, \forall v_i \in V(F_i)$ e $|S^*(F_i)| = i$.*

Agora, vamos considerar a família de Snarks Goldberg. Uma solução pode ser obtida para um Snark Goldberg G_i acionando vértices que acendem cada bloco do grafo. Cada bloco pode ser aceso em 3 movimentos (acionando os vértices c_i, e_i, h_i de cada bloco B_i). Esta solução foi obtida através de um software que desenvolvemos, descrito na Seção 4. Note que a Proposição 1 fornece um valor inferior ao obtido nesta solução. Uma prova teórica acerca da minimalidade desta solução está em fase de elaboração.

Teorema 6. *Seja G_i um Snark Goldberg composto por i blocos B_i de vértices $a_i, b_i, c_i, d_i, e_i, f_i, g_i, h_i$ como na Fig. 2 (b). Então $S(G_i) = \{c_i, e_i, h_i\}$ é uma solução para G_i.*

Através do software, observamos que nos Snarks testados, dada uma solução S_i, a solução complementar $\overline{S_i}$ também era solução. A Proposição 2 generaliza este resultado.

Proposição 2. *Seja G um grafo cujos vértices têm grau ímpar. Então, para cada solução $S_i \in S(G)$, temos que $\overline{S_i} \in S(G)$, $\overline{S_i} = V \setminus S_i$.*

4. Software

Para auxiliar nos estudos, foi desenvolvido um software que permite criar grafos e analisar o problema manualmente ou usando um algorítimo força-bruta. A técnica exaustiva consiste na contagem de 1 até 2^n. Em cada passo, percorremos todos os bits; pressionamos cada vértice cujo bit corresponde a 1; e verificamos se o grafo foi solucionado. A complexidade desde algoritmo é $O(2^n)$. Entretanto, para grafos caterpillar, é possível reduzir consideravelmente a quantidade de iterações do algoritmo força-bruta, tornando sua complexidade $O(2^e)$, onde e é o tamanho da espinha.

Referências

Berman, A., Borer, F., and Hungerbühler, N. (2019). Lights out on graphs. *arXiv:1903.06942*.

Fleischer, R. and Yu, J. (2013). A survey of the game lights out! *LNCS*, 8066.

Isaacs, R. (1975). Infinite families of nontrivial trivalent graphs which are not tait colorable. *The American Mathematical Monthly*.

Sutner, K. (1988). Additive automata on graphs. complex systems 2. *Stevens Institute of Technology*.