Problema de Coloração em Grafos

Jhonattan C. B. Cabral[1]

[1]Departamento de Matemática e Informática Aplicada (DIMAp)
Universidade Federal do Rio Grande do Norte (UFRN)

jhonattan.yoru@gmail.com

Resumo. *É possível encontrar na literatura alguns estudos modelando determinados problemas utilizando o conceito de coloração em grafos, além da modelagem, os estudos propõem algoritmos que tem como finalidade resolver o problema em um tempo mínimo. O presente trabalho tem como objetivo detalhar o problema de coloração, realizar a implementação do algoritmo heurístico DSATUR e em seguida verificar os resultados obtidos após a sua execução.*

1. Introdução

Na literatura pode-se encontrar uma vasta gama de trabalhos que objetivam modelar e solucionar problemas utilizando conhecimentos extraídos a partir da Teoria dos Grafos, algumas dessas modelagens necessitam impor rótulos sujeitos a restrições e normalmente a restrição aplicada é de rotular os vértices de um grafo tal que não haja dois vértices adjacentes que compartilhem o mesmo rótulo. Esta forma de moldar o problema está relacionado ao conceito de coloração em grafos.

Colorir da melhor forma os vértices de um grafo é um problema presente na classe NP-Difícil e é um dos mais conhecidos na Teoria dos Grafos. Basicamente, colorir um grafo G é atribuir cores aos seus vértices de forma que vértices adjacentes recebam cores distintas (GOLDBARG, 2012). Em estudos já realizados, podemos encontrar alguns algoritmos exatos, heurísticos e metaforísticos que tentam encontrar uma solução para o problema em um tempo plausível.

Diante do exposto, este artigo tem como objetivo detalhar de forma clara o problema de encontrar uma coloração miníma para um determinado grafo, destacar algumas aplicações reais, realizar uma breve análise acerca dos principais algoritmos heurísticos existentes e efetuar o estudo e a implementação de uma heurística a fim de solucionar tal problema. Por fim, o trabalho se concentra em expor os resultados obtidos após a execução do algoritmo e efetua uma breve análise com base nas soluções ótimas de cada caso de teste.

2. Descrição do problema

Sejam $G(V, E)$ um grafo e $C = \{c_i, 1 \leq i \leq n, n \in \mathbb{R}\}$ um conjunto de cores. Uma coloração de vértices de G é uma atribuição de cores de C para os vértices, de maneira que aos nós adjacentes são atribuídas cores diferentes. Seguindo o contexto, pode-se definir ainda que uma $k-$coloração é uma coloração que consiste de k cores diferentes.

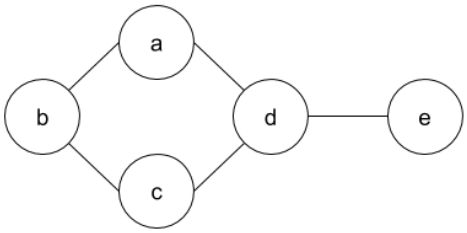

Figura 1. Grafo G

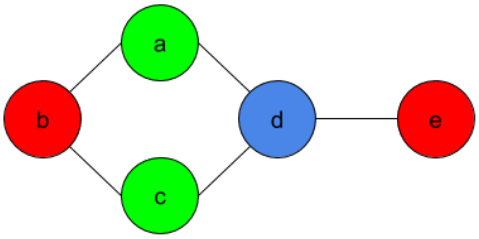

Figura 2. 3-coloração de G

Além de determinar as cores dos vértices do grafo, é possível ainda tentar encontrar o número mínimo de cores necessárias para coloração própria de um grafo G, neste caso, estaremos em busca do número cromático do grafo ou $\chi(G)$. Algumas informações extras podem ser concluídas a partir da extração do número cromático, por exemplo, durante a procura do $\chi(G)$, cada cor atribuída aos vértices de G forma um conjunto independente de vértices (GOLDBARG, 2012).

Na figura 3 é possível observar o número cromático do grafo G, o mesmo grafo das figuras anteriores, note que na figura 2 o grafo foi colorido utilizando 3 cores, porém o seu número cromático é 2.

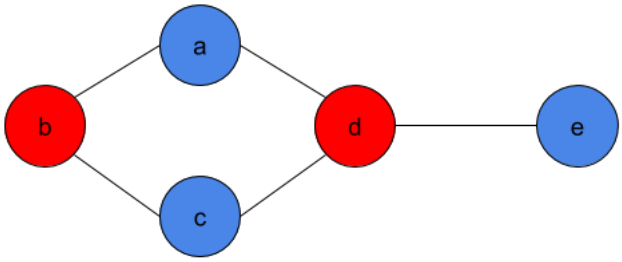

Figura 3. $\chi(G) = 2$

2.1. Aplicações

2.2. Agendamento de avaliações na universidade

Trata-se de uma aplicação simples que consiste em definir a melhor forma de agendar avaliações de uma instituição de ensino de modo que duas disciplinas com estudantes em comum não tenham seus exames agendados para o mesmo horário. Ou seja, pode-se determinar utilizando coloração em grafos qual o número mínimo de horários necessários para agendar as avaliações.

2.2.1. Coloração de mapas

Colorir mapas é uma aplicação clássica que consiste em determinar o menor número de cores necessário para pintar os mapas de forma que duas regiões adjacentes não recebam a mesma cor. Um estudo realizado com foco nesta aplicação deu origem ao teorema das

quatro cores. Basicamente, o teorema diz que quatro cores são suficientes para colorir a superfície de qualquer mapa tal que regiões adjacentes recebam cores distintas, este teorema foi provado em 1976 pelos matemáticos Kenneth e Wolfgang Haken.

2.2.2. Descarte de rejeitos químicos

Considerando a importância da etapa de gerenciamento de resíduos em um laboratório químico, deve-se levar em conta, por questões de segurança, que alguns tipos de resíduo não podem ser descartados juntos. Com isso, para evitar a possibilidade de reações violentas e prejuízos ao meio ambiente, deve-se encontrar um número mínimo de recipientes que o laboratório precisa para realizar o descarte apropriado dos resíduos.

2.3. Algoritmo DSATUR

Com o objetivo de solucionar o problema de coloração e tentar obter uma coloração mínima para um determinado grafo G, foi implementado o algoritmo DSATUR. O algoritmo foi proposto por Daniel Brélaz em 1979 como uma heurística gulosa e é utilizado o conceito de grau de saturação de um vértice, que nada mais é do que a quantidade de cores distintas atribuídas aos vizinhos já coloridos de um vértice qualquer de G.

Apesar de ser uma heurística, o DSATUR pode ser considerado exato se for executado para grafos bipartidos (LIMA, 2017). O algoritmo possui uma complexidade de (n^2) e abaixo podemos conferir o pseudo código que o descreve, a etapa principal do algoritmo consiste em colorir os vértices considerando o seu grau de saturação.

Algoritmo 1: DSATUR

Entrada: Grafo G
Saída: Coloração de G

```
1  v ⟵ vértice com maior grau em G
2  Colore o vértice v com a cor inicial
3  Calcula o grau de saturação para todos os vértices adjacentes a v
4  repita
5  |  s ⟵ vértice com o maior grau de saturação em G
6  |  se Existe mais de um vértice com o maior grau de saturação então
7  |  |  s ⟵ vértice que possui o maior grau em G
8  |  fim
9  |  Colore s com a menor cor disponível
10 |  Calcula o grau de saturação para todos os vértices adjacentes a s
11 até Existir vértices não coloridos;
12 retorna Coloração de G
```

3. Resultados

Para o experimento foi utilizado um computador com o processador AMD A8-4500m, 8GB de ram e 64 bit.

A base de dados é composta por 30 grafos, cada um possui uma quantidade variada de vértices e arestas. Todos os dados foram extraídos a partir da internet e cada arquivo

contem o número de vértices, número de arestas e a sua coloração ótima. A seguir pode-se analisar o resultado obtido para cada caso testado, as informações estão separadas em: nome do arquivo (*file*), quantidade de vértices (N), quantidade de arestas (M), coloração ótima χ, coloração extraída a partir do algoritmo implementado (*DSATUR*) e o tempo de execução do algoritmo (T) em segundos.

file	N	M	χ	*DSATUR*	T
test_0	6	9	3	3	0,00060
test_1	11	20	4	4	0,00168
test_2	23	71	5	5	0,01821
test_3	10	15	3	3	0,00134
test_4	25	160	5	5	0,08488
test_5	36	290	7	9	0,26477
test_6	47	236	6	6	0,17036
test_7	49	476	7	11	0,70498
test_8	64	728	10	12	1608435,00000
test_9	74	301	11	11	0,27909
test_10	80	254	10	10	0,19957
test_11	81	1056	10	13	3418677,00000
test_12	87	406	11	11	0,53224
test_13	95	755	7	7	1794554,00000
test_14	96	1368	12	14	5891476,00000
test_15	128	387	8	8	0,45208
test_16	138	493	11	11	0,71715
test_17	169	3328	13	17	38356467,00000
test_18	197	3925	49	49	57016777,00000
test_19	191	2360	8	8	18902274,00000
test_20	450	5714	5	10	123008613,0000
test_21	450	5734	5	9	132313018,00000
test_22	450	9803	5	10	431703100,00000
test_23	450	9757	5	12	441306006,00000
test_24	450	8168	15	17	287392791,00000
test_25	450	8169	15	16	299018507,00000
test_26	450	8260	25	25	308186661,00000
test_27	450	8263	25	25	309862322,00000
test_28	451	8691	30	30	341131562,00000
test_29	561	3258	13	13	38629323,00000

No gráfico abaixo podemos avaliar de uma melhor forma a relação do resultado obtido a partir da execução do algoritmo e a coloração ótima de cada caso testado.

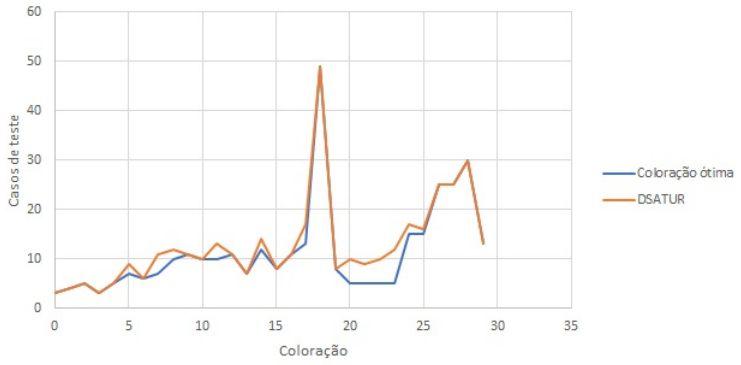

Figura 4. Gráfico da relação entre a coloração ótima e o resultado do DSATUR

4. Conclusão

Após o estudo realizado pode-se concluir que o algoritmo DSATUR, se bem implementado, pode atingir resultados aceitáveis, uma vez que durante a análise do resultado foi possível comparar os dados obtidos através da execução algoritmo valores ótimos, ou seja, o número cromático de cada grafo.

Quanto ao tempo de execução, pode-se afirmar que os valores obtidos podem ser minimizados implementando a heurística utilizando outra linguagem de programação, pois o algoritmo foi implementado utilizando a linguagem Python e por ser uma linguagem interpretada, a execução dos algoritmos tendem a ser mais lenta.

5. Referência

GOLDBARG, Marco; GOLDBARG, Elizabeth. **Grafos : conceitos, algoritmos e aplicações**. Rio de Janeiro: Elsevier, 2012.

LIMA, A. M.**Algoritmos exatos para o problema da coloração de grafos.** 2017. Disponível em:(encurtador.com.br/hlqr7). Acesso em 13-03-2019.

GROSS, J. L;YELLEN, J. **Graph theory and its applications.** 2005.