Atividade Prática – Análise de Algoritmos

Aluno: Guilherme Vieira de Figueiredo
Matricula: 11421BCC027
Data: 07/06/2021

O algoritmo escolhido para ser implementado, foi o algoritmo de coloracao de vértices, para sua implementaçao representamos os grafos atráves de listas de adjacência, que são percorridas pelo algoritmo, fazendo-se uma checagem das cores já utilizadas pelos vizinhos, e em seguida atribuindo a menor cor disponivel para o vértice, a relação entre as cores e os vértices é implementada atráves de uma lista, onde o indice da lista equivale ao número do vértice e o seu valor a cor do mesmo. Após isso, chegamos ao resultado verificando o maior número (cor) na lista de relação de cores.

Para a implementação do algoritmo, foi escolhida a linguagem Python, devido a familiaridade com a linguagem, a simplicidade, e ao fato de ser possível o uso de diversas bibliotecas como, pyplot, numpy e etc.

Os grafos utilizados no algoritmo, foram extraidos do benchmark disponível em : https://mat.tepper.cmu.edu/COLOR/instances.html, o algoritmo implementado, lê os dados dos arquivos referentes a cada grafos, e coloca suas informações dentro da lista de adjacências, assim criando o grafo, os grafos utilizados foram:

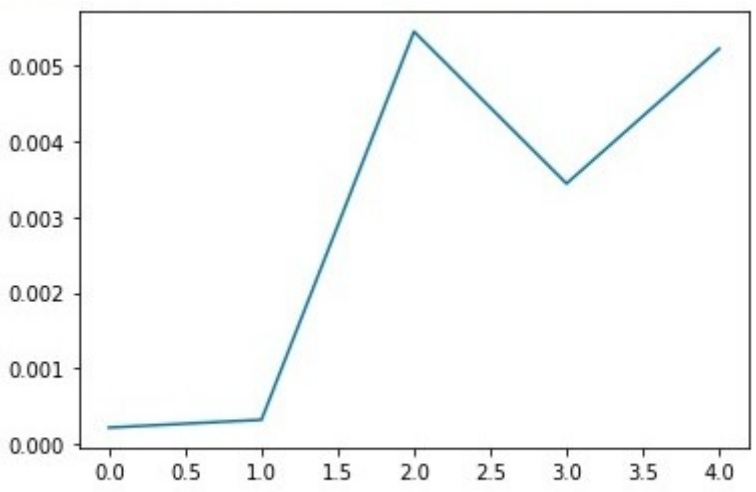
Myciel3: grafo com 11 vértices, 20 arestas, e melhor coloração possível de 4 cores
Myciel4: grafo com 23 vértices, 71 arestas, e melhor coloração possível de 5 cores
Myciel6: grafo com 95 vértices, 755 arestas, e melhor coloração possível de 7 cores
Myciel7: grafo com 191 vértices, 2360 arestas, e melhor coloração possível de 8 cores
Homer : grafo com 561 vértices, 1629 arestas, e melhor coloração possível de 13 cores.

Devido ao elevado número de vértices dos grafos apresentados, não será possível mostrar as matrizes de adjacência dos mesmos.

Após a execução do algoritmo, foram obtidos os seguintes resultados:

```
Numero de cores encontradas para o Grafo x:  4
Melhor solução da literatura: 4
Tempo de execução:  0.00021958351135253906

Numero de cores encontradas para o Grafo x:  5
Melhor solução da literatura: 5
Tempo de execução:  0.0003223419189453125

Numero de cores encontradas para o Grafo x:  7
Melhor solução da literatura: 7
Tempo de execução:  0.005446195602416992

Numero de cores encontradas para o Grafo x:  8
Melhor solução da literatura: 15
Tempo de execução:  0.0034399032592773438

Numero de cores encontradas para o Grafo x:  15
Melhor solução da literatura: 13
Tempo de execução:  0.005218505859375
```

Como podemos observar, o algoritmo guloso para coloroção de vértices obteve resultados excelentes, encontrando o melhor valor possível na maioria dos grafos testados, e com um tempo de execução extremamente baixo, podemos concluir dessa forma que o paradigma guloso tem resultados bons para o problema de coloração de vértices. Uma pequena observação é que o paradigma adotado não produziu um algoritmo **ótimo,** uma vez que não encontra o melhor resultado para todas as execuções do algoritmo, pelo fato de que o problema não possui **sub-estrutura ótima**, apesar disso é um algoritmo de **aproximação** excelente para o problema.



Com relação ao tempo de execução, vemos que obteve um tempo excelente, que se manteve 'estavel' até mesmo para um grande número de vértices, o que jamais seria possível com um algoritmo exato para o problema de coloração de vértices, uma vez que a complexidade de tempo do algoritmo implementado é $O(V^2)$ enquanto que o melhor algoritmo ótimo conhecido para o problema possui complexidade de tempo de O(2.445^v).

Observação: tempo do gráfico está em segundos.