

Lista 7 - MAC0328

Matheus de Mello Santos Oliveira - 8642821

1 Implementação

Queremos encontrar uma k coloração em um grafo. Para isso realizamos um backtrack que realiza uma busca completa em todas as possíveis colorações e verifica se alguma delas é válida, isto é, para alguma dessas colorações todos as arestas possuem vértice de início e fim de cores diferentes. Uma redução no espaço de busca está implícita na ideia de um vértice não deve ser colorido da mesma cor de algum de seus vizinhos. Podemos limitar superiormente essa função com $O(k^n)$, mas podemos perceber que o fato do backtrack não permitir colorações inválidas, varias dessas opções não serão testadas, o que torna esse upper bound bastante folgado.

2 Experimentos

Realizamos testes para $V = \{5, 10, 35\}$ e $E = \{V, 2V, 5V\}$ e $k = \{2, 3, 5, 8\}$. Todos os testes foram repetidos 10 vezes. O número de vezes em que k cores foram suficientes para pintar o grafo. Linhas

são números de vértices e colunas são número de arestas. Para $k = 2$

	V	2V	5V
5	8	0	0
10	3	0	0
35	0	0	0

Para $k = 3$		V	2V	5V
	5	10	0	0
	10	10	10	0
	35	10	5	0

Para $k = 5$		V	2V	5V
	5	10	10	10
	10	10	10	0
	35	10	10	9

Para $k = 8$		V	2V	5V
	5	10	10	10
	10	10	10	0
	35	10	10	10

Importante notar que em geral a heurística gulosa retorna um resultado similar, porém com uma fração do custo computacional.

3 Uso do programa

O programa recebe como argumento da linha de comando o número de vértices o número de arestas e a quantidade de cores que desejamos usar para colorir o grafo.

Para compilar:

```
$ make
```

Para executar o código:

```
$ ./ep7 [V] [A] [k]
```

Onde V é o número de vértices e A é o número de arestas e k é o número de cores disponíveis.