



Coloração de grafos



Problema da coloração de vértices aplicando Estratégia Evolutiva (ES) e Busca Tabu

CSI557 – Computação Evolucionária


Aluna: Nathany Aparecida Salles 13.2.8382

Prof: Fernando Bernardes de Oliveira



Problema da Coloração de grafos

- ▶ Origem em 1852;
- ▶ Frederick Guthrie sugeriu que seria necessário no máximo quatro cores para colorir as regiões de qualquer mapa, evitando que duas regiões adjacentes tivessem a mesma cor;

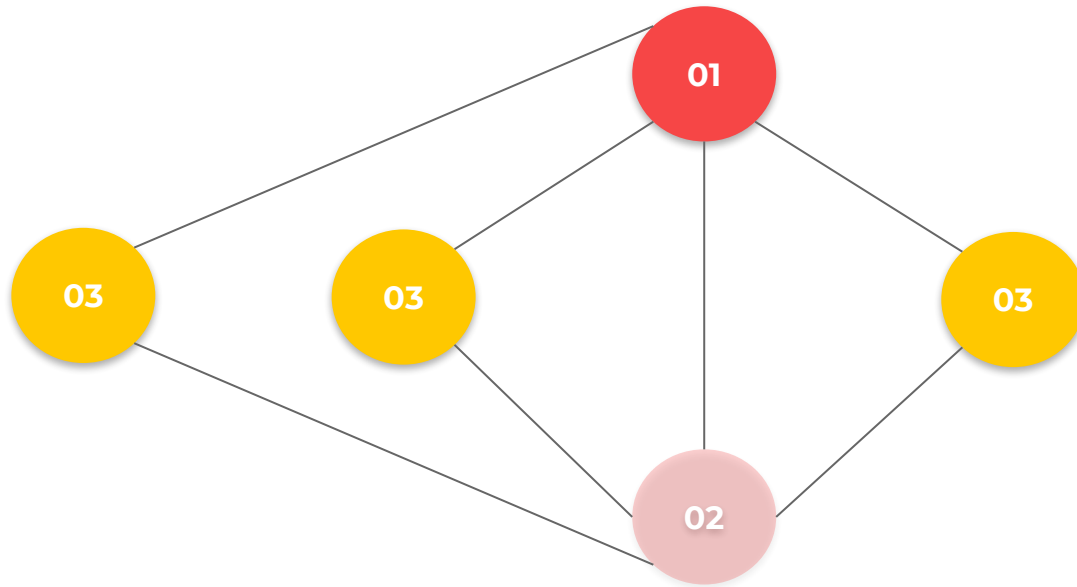
- 
- ▶ Teorema das Quatro Cores: qualquer um grafo planar é 4-colorível;
 - ▶ Surge um outro problema: estabelecer um valor mínimo de cores para colorir um grafo qualquer;
 - ▶ Problema da Coloração de Grafos;
 - ▶ Considerado um problema NP-completo



“

Dado um grafo, colorir seus vértices de modo que vértices adjacentes tenham cores diferentes e o número de cores aplicadas seja o mínimo possível

Exemplo





Aplicações

- ▶ Scheduling;
- ▶ Timetabling;
- ▶ Otimização de alocação de registradores;
- ▶ Frequency assignment;

Busca Tabu





“

É um procedimento adaptativo auxiliar para a busca local dentro de um espaço de busca.



Ideia principal

A partir de uma solução inicial, a busca se move para a melhor solução na vizinhança, não realizando movimentos que já foram realizados por estarem armazenados na lista tabu. A lista permanece na memória durante um determinado tempo ou até um número de iterações sem melhora.

procedimento BT

1. Seja s_0 solução inicial;
2. $s^* \leftarrow s$; {Melhor solução obtida até então}
3. Iter $\leftarrow 0$; {Contador do número de iterações}
4. MelhorIter $\leftarrow 0$; {Iteração mais recente que forneceu s^* }
5. Seja BTmax o número máximo de iterações sem melhora em s^* ;
6. T $\leftarrow \emptyset$; {Lista Tabu}
7. Inicialize a função de aspiração A;
8. enquanto (Iter - MelhorIter \leq BTmax) faça
9. Iter \leftarrow Iter + 1;
- ($m \notin T$) 10. Seja $s' \leftarrow s \oplus m$ o melhor elemento de $V \subseteq N(s)$ tal que o movimento m não seja tabu
 ou s' atenda a condição de aspiração ($f(s') < A(f(s))$);
11. Atualize a Lista Tabu T;
12. $s \leftarrow s'$;
13. se $f(s) < f(s^*)$ então
14. $s^* \leftarrow s$;
15. MelhorIter \leftarrow Iter ;
16. fim-se;
17. Atualize a função de aspiração A;
18. fim-enquanto;
19. Retorne s^* ;
- fim BT**;

Estratégia Evolutiva





Modelagem do problema

Indivíduos:

O array de cromossomos foi definido como um array de cores, considerando 1 cor para cada vértice do grafo analisado.

População:

Conjunto de indivíduos.

Problema:

Atribuir o custo de 100 para cada cor usada no grafo.



Classes adicionais

Grafo.java

Reune os principais atributos de um grafo: vértices, arestas e lista de adjacência

Leitor.java

Faz a leitura de um arquivo txt e retorna um objeto grafo

Aresta.java

Possui os atributos origem e destino, essenciais para criação da lista de adjacência

```
public ArrayList<Integer> encontraVizinho(ArrayList<Integer> s){
    ArrayList<Integer> s1;
    Random rand = new Random();
    ArrayList<Integer> cores = (ArrayList<Integer>) problema.listaCores(s);
    //sotear um vértice
    int vertice = rand.nextInt(grafo.vertices.size());
    //sortear uma cor
    int cor = rand.nextInt(cores.size());
    int c = cores.get(cor);
    s1 = (ArrayList<Integer>) s.clone();
    int cont = 0;
    // confiro a lista de adjacência para saber se tem conflito
    for (int j = 0; j < grafo.listaAdj.get(vertice).size(); j++) {
        int adjacente = grafo.listaAdj.get(vertice).get(j).getDestino();
        if(s.get(adjacente) == c){
            cont++;
            break;
        }
    }
    //se não tiver adj com a mesma cor, então a cor pode ser add ao vértice
    if (cont == 0) {
        s1.set(vertice, c);
    }
    return s1;
}
```

Parâmetros

Integer mu = 50; // Tamanho da populacao

Integer lambda = 80; // numero de descendentes

Integer geracoes = 300; // criterio de parada

Double pMutacao = 0.5; // mutacao - aplicacao ao descendente -
variacao/perturbacao

Comparações

	Busca Tabu	ES
Melhor resultado	97 cores	78 cores
Pior resultado	97 cores	84 cores
Média	97 cores	81.36 cores
Desvio padrão	0	1.49674
Melhor tempo	832ms	438ms
Pior tempo	986ms	528ms

Unpaired *t* test results

P value and statistical significance:

The two-tailed P value is less than 0.0001

By conventional criteria, this difference is considered to be extremely statistically significant.

Confidence interval:

The mean of Group One minus Group Two equals -15.63

95% confidence interval of this difference: From -16.18 to -15.09

Intermediate values used in calculations:

$t = 57.2092$

$df = 58$

standard error of difference = 0.273

Learn more:

GraphPad's web site includes portions of the manual for GraphPad Prism that can help you learn statistics. First, review the meaning of [P values](#) and [confidence intervals](#). Then learn how to interpret results from an [unpaired](#) or [paired](#) *t* test. These links include GraphPad's popular *analysis checklists*.

Review your data:

Group	Group One	Group Two
Mean	81.37	97.00
SD	1.50	0.00
SEM	0.27	0.00
N	30	30

Obrigada 🍷👍

Dúvidas?