MOHAMED BOUDIAF UNIVERSITY OF M'SILA

DEPARTMENT OF COMPUTER SCIENCE

Advanced Graph Theory

Chapter 3: Coloring problem in graphs

Dr. SAID KADRI

Senior lecturer

Department of Computer Science, Faculty of Mathematics and Computer Science,

Mohamed Boudiaf University of M'sila

E-mail:kadri.said28@gmail.com

Website:https://kadrisaid28.wixsite.com/sgadri

2017 - 2018

Coloring problem in graphs

Problem Statement

Let G(X, U) be an undirected graph.

There are three types of coloring in G.

1. **Vertices' Coloring.**
2. **Edges' Coloring.**
3. **Faces' Coloring.**

Definition

- **Coloring the vertices/edges/faces of a graph G consists of assigning a color to each of the vertices (edges/faces) so that no two adjacent vertices (edges/faces) have the same color.**
- **A graph is said to be *p*-chromatic if its vertices admit a coloring in p colors.**
- **We call chromatic number $\gamma(G)$ (chromatic index q(G) for edge coloring) the minimum number of different colors needed to perform a vertex coloring (edges/faces) of G.**

Vertices' Coloring

G(X, U) an undirected graph

- **Let I be a subset of vertices $I \subset X$, I is said stable set if it only includes non-adjacent vertices two by two (any two vertices of I are not adjacent).**
- **Vertices with the same color in a vertex coloring form a stable set (non-adjacent).**
- **So, a vertices' coloring of G is just a partition of G into stable sets.**
- **We define $\alpha(G)$ the stability number which is the maximum cardinality of a stable set. So if $\gamma(G)$ is the chromatic number, and since every set of vertices of the same color has a cardinality less than or equal to $\alpha(G)$ so we have:**

$\alpha(G).\gamma(G) \geq N(G)$

With:

$N(G) = |X|$ is the number of vertices of the graph (the order)

We can deduce:

  - **$\gamma(G) \geq [N(G)/\alpha(G)]$**

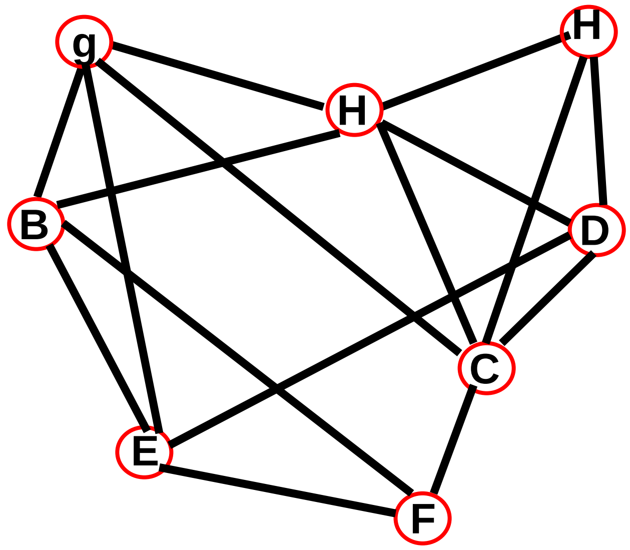
- **Note that the determination of the chromatic number $\gamma(G)$, as well as obtaining a minimal coloring of the vertices of G is a complex problem and requires the use of heuristic coloring algorithms.**

- **We should also note that γ(G) for a graph with *n* vertices and m edges is defined under the following conditions:**

  - ✓ **$\gamma(G) \leq (n - d_{min})$, with: d_{min} the minimum degree of the vertices in G.**
  - ✓ **$\gamma(G) \geq$ the cardinal of the largest clique in G**
  - ✓ **$\gamma(G) \geq n^2/(n^2 - 2m)$**
  - ✓ **$\gamma(G) \leq (n+1-\alpha(G))$ with $\alpha(G)$ the stability number of G.**
  - ✓ **$\gamma(G) \leq d_{max}+1$ with d_{max} the maximum degree of vertices in G.**

OBS: The above properties are demonstrated in the literature.

Coloring algorithm (Welsh and Powell)



1. Put the vertices of the graph G in decreasing order of degree x1, x2, ….., xn with $d(x_i) \geq d(x_j)$ for $1 \leq J \leq n$

Summit	Degree
A	5
C	5
B	4
D	4
E	4
G	4
F	3
H	3

2. Assign color c_1 to x_1 and the next vertex in the list that is not adjacent to x_1 and not already colored, and so on with vertices in the list that are not adjacent to vertices already colored with color c_1.

3. Assign color c_2 to the first vertex not yet colored, as well as to subsequent vertices that are not adjacent to vertices already colored with color c_2

4. Continue the process of Coloring using New colors c_3, c_4, …, c_k until we finish coloring all the vertices in the list.

Summit	Degree	Color
A	5	C1
C	5	C2
B	4	C2
D	4	C3
E	4	C1
G	4	C3
F	3	C3
H	3	C4

The coloring result of the previous example is as follows:

- The color c_1 is assigned to the vertices {A, E}
- The color c_2 is assigned to the vertices {C, B}
- The color c_3 is assigned to the vertices {D, F, G}
- The color c_4 is assigned to the vertex {H}

We say that G is 4-colorable

OBS : a bipartite graph is a 2-colorable graph

Welsh and Powell algorithm(a variant based on the use of the adjacency matrix).

Step 0: Initialization

- ✓ M: Adjacency matrix of the graph whose vertices are arranged in descending order of degree.
- ✓ K=1 (index of the used color)

Step 1:

- ✓ N=M (N: Row index; M: Column index)

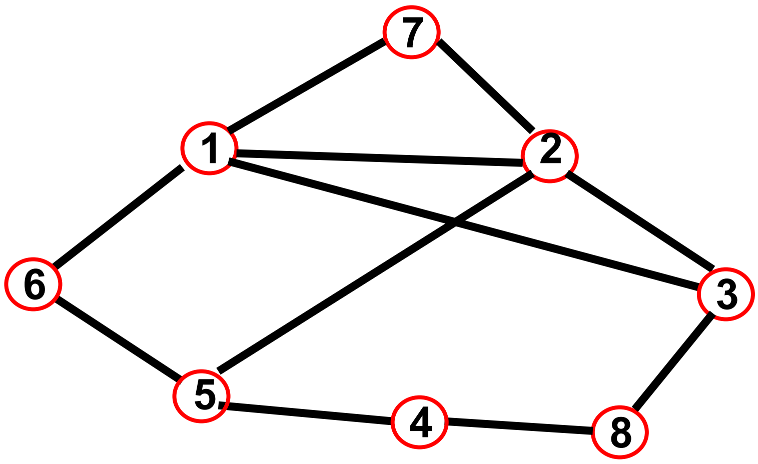
Step 2:

- ✓ Find the set N of rows not yet colored having a zero in the columns of color c_k
- ✓ Color with the color c_k the first row not yet colored in the set N, as well as the corresponding column (no two adjacent vertices must have the same color c_k otherwise we skip it)
- ✓ If N≠ϕ go to step 2, otherwise go to step 3

Step 3:

- ✓ If all lines are colored then STOP (and we have a k-coloring)
- ✓ k=k+1 (change color)
- ✓ Go to step 1.

Example:



Step 0:

1. **We draw the adjacency matrix of G**

	1	2	3	4	5	6	7	8
1	0	1	1	0	0	1	1	0
2	1	0	1	0	1	0	1	0
3	1	1	0	0	0	0	0	1
4	0	0	0	0	1	0	0	1
5	0	1	0	1	0	1	0	0
6	1	0	0	0	1	0	0	0
7	1	1	0	0	0	0	0	0
8	0	0	1	1	0	0	0	0

2. **K=1 choose the color c_1**

Step 1:

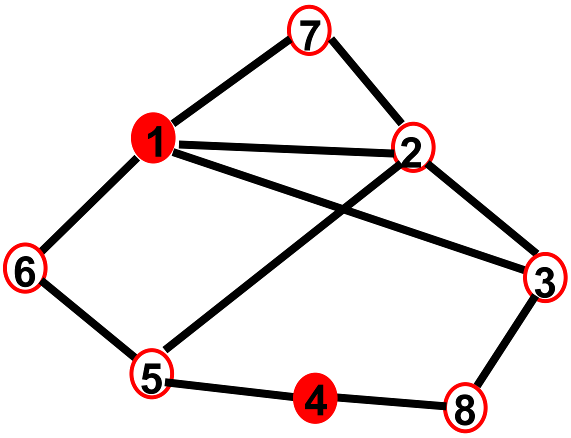
N=M=1 (Row 1, Column 1)

Step 2:

- ✓ **Find the set N_k (k is the color index) of rows not yet colored having a zero in the columns of color c_1**

  $N_k=\{N_1, N_4, N_5, N_8\}$

- ✓ Color with color c_1 the first row not yet colored in the set N_k, as well as the corresponding column (no two adjacent vertices must have the same color c_k otherwise we skip it)



Ck,

	1	2	3	4	5	6	7	8
1	0	1	1	0	0	1	1	0
2	1	0	1	0	1	0	1	0
3	1	1	0	0	0	0	0	1
4	0	0	0	0	1	0	0	1
5	0	1	0	1	0	1	0	0
6	1	0	0	0	1	0	0	0
7	1	1	0	0	0	0	0	0
8	0	0	1	1	0	0	0	0

We color the rows 1, 4 as well as the corresponding columns (but not 5 and 8 because they are adjacent to vertex 4 already colored)

- ✓ If N≠ϕ go to step 2, otherwise go to step 3

Go to step 3

Step 3:

- ✓ If all lines are colored then STOP (and we have a k-coloring)
- ✓ k=k+1 (change color k=2)

Go to step 1.

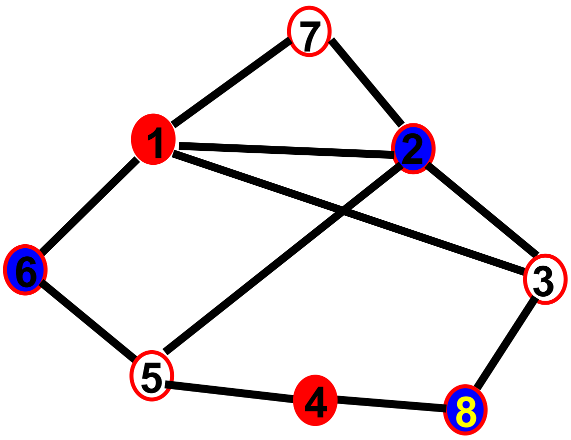
N=M=2 (Row 2, Column 2)

Step 2:

- ✓ **Find the set N_k of lines not yet colored having a zero in the columns of color c_2**

  $N_k=\{N_2, N_6, N_8\}$

- ✓ **Color with the color c_2 the first row not yet colored in the set N_k, as well as the corresponding column (no two adjacent vertices must have the same color ck otherwise we skip it)**



Ck,

	1	2	3	4	5	6	7	8
1	0	1	1	0	0	1	1	0
2	1	0	1	0	1	0	1	0
3	1	1	0	0	0	0	0	1
4	0	0	0	0	1	0	0	1
5	0	1	0	1	0	1	0	0
6	1	0	0	0	1	0	0	0
7	1	1	0	0	0	0	0	0
8	0	0	1	1	0	0	0	0

We color the lines 2, 6, 8 as well as the corresponding columns.

- ✓ **If N≠ϕ go to step 2, otherwise go to step 3**

Go to step 3

Step 3:

- ✓ **If all lines are colored then STOP (and we have a k-coloring)**
- ✓ **k=k+1 (change color k=3)**

Go to step 1.

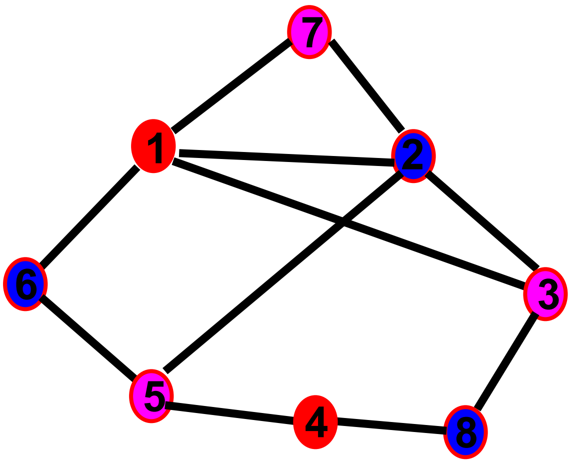
N=M=3 (Row 3, Column 3)

Step 2:

- ✓ **Find the set N_k of lines not yet colored having a zero in the columns of color c_3**

  $N_k=\{N_3, N_5, N_7\}$

- ✓ **Color with the color c_2 the first row not yet colored in the set N_k, as well as the corresponding column (no two adjacent vertices must have the same color c_k otherwise we skip it)**



Ck,

	1	2	3	4	5	6	7	8
1	0	1	1	0	0	1	1	0
2	1	0	1	0	1	0	1	0
3	1	1	0	0	0	0	0	1
4	0	0	0	0	1	0	0	1
5	0	1	0	1	0	1	0	0
6	1	0	0	0	1	0	0	0
7	1	1	0	0	0	0	0	0
8	0	0	1	1	0	0	0	0

We color the lines 3, 5, 7 as well as the corresponding columns.

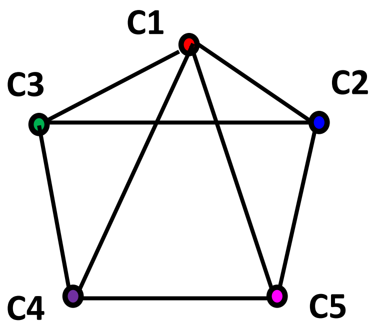
✓ If N≠ϕ go to step 2, otherwise go to step 3

Go to step 3

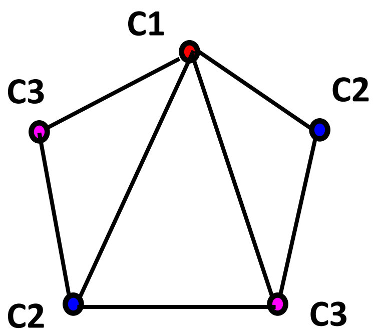
Step 3:

✓ If all lines are colored then STOP (and we have a k-coloring k=3)

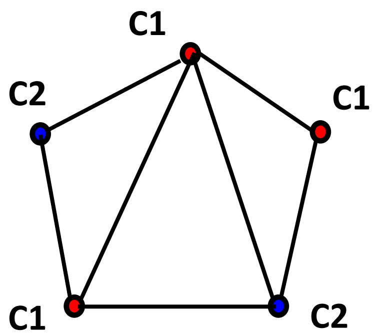
Coloring examples



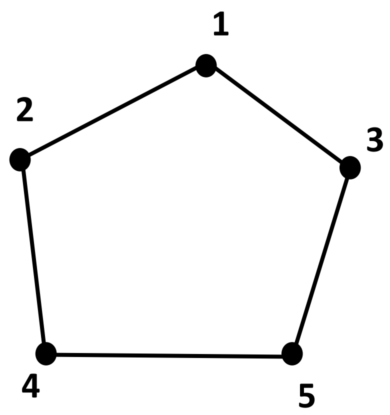
5 colors



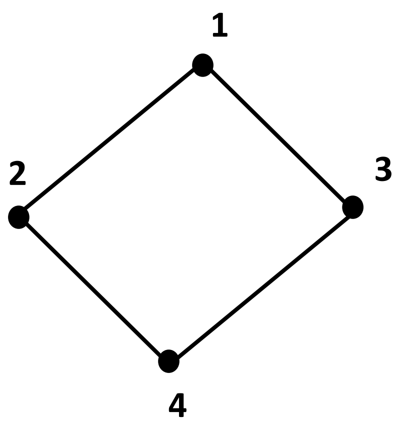
3 colors (sufficient)
γ(G) = 3



Incorrect coloring



Nb. vertices = 5 (odd)
⇨ γ(G) = 3
(3 needed colors)



Nb. vertices = 4 (even)
⇨ γ(G) = 2
(2 needed colors)

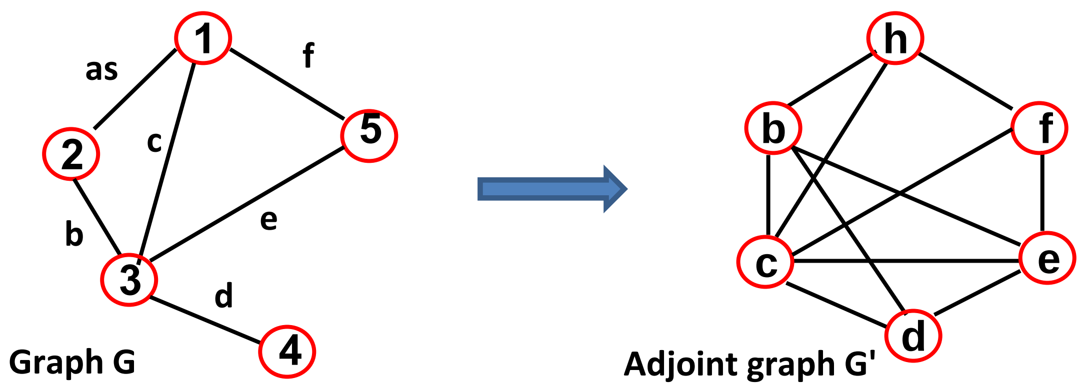
Note: Any planar graph is colorable with 04 colors.

Edges' Coloring

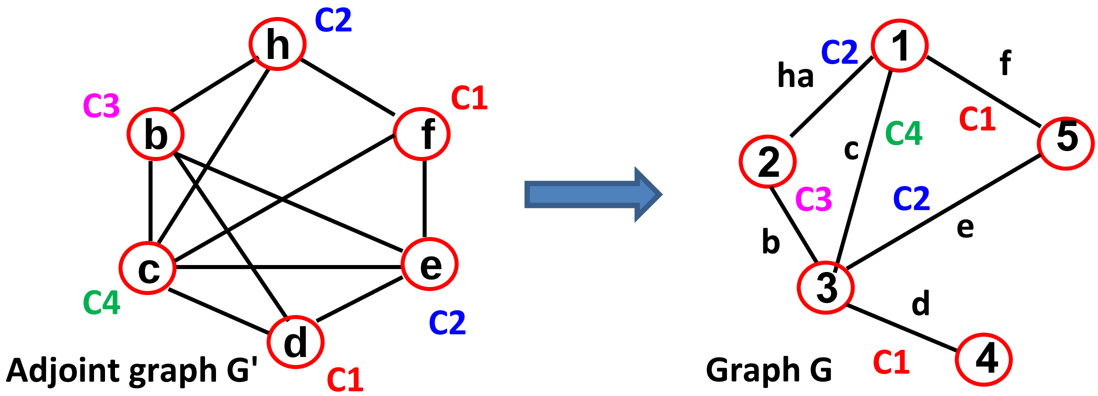
- Similar to vertices' coloring, edges' coloring of a graph involves assigning a color to any edge of the graph so that no two adjacent edges have the same color.
- We define the chromatic index of the graph G (similar to the chromatic number for vertices' coloring) as being the smallest k for which there exists a coloring of the edges, it is noted $\chi(G)$.
- Edges' Coloring of a graph can be reduced to vertices' coloring by working not on the graph itself, but on the adjoint graph G' which is defined as follows:

1. To each edge of G(X, U), we associate a vertex of G'(X', U').
2. Two vertices of G' are connected by an edge, if the two corresponding edges of G are adjacent.

Example :



Subsequently, we apply the WELSH POWELL algorithm on the graph G' to color its vertices. Once this coloring is done, we will color the edges of G with the same colors of the corresponding vertices.

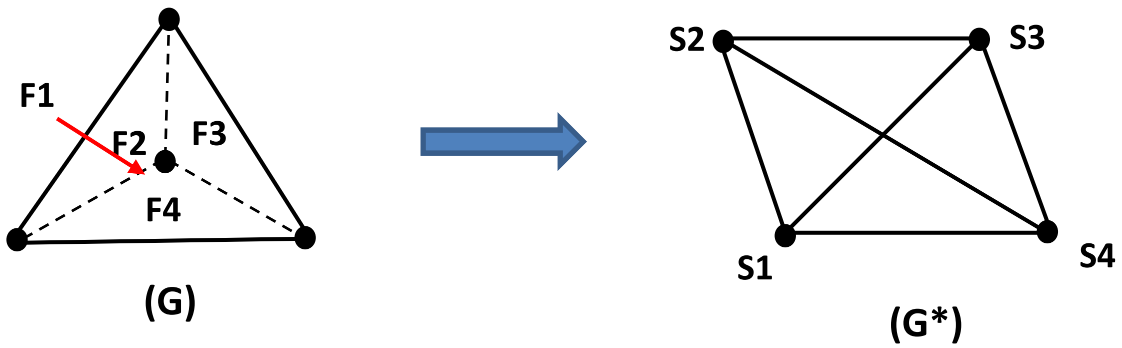


Coloring faces

Definition of dual of a graph

- **Let G(X, U) be a planar multigraph, the dual G* of G is the graph defined as follows:**
- **The faces of G are represented by vertices in G***
- **For each edge of G belonging to the boundary of two faces F1, F2, there corresponds an edge in G* connecting the vertices S1, S2 corresponding to the faces F1, F2.**
- **The Dual of a planar graph G is also a planar graph.**

  (G*)* = G



And so, the problem of coloring the faces of a planar graph G comes down to coloring the vertices of its dual graph G* and vice versa.