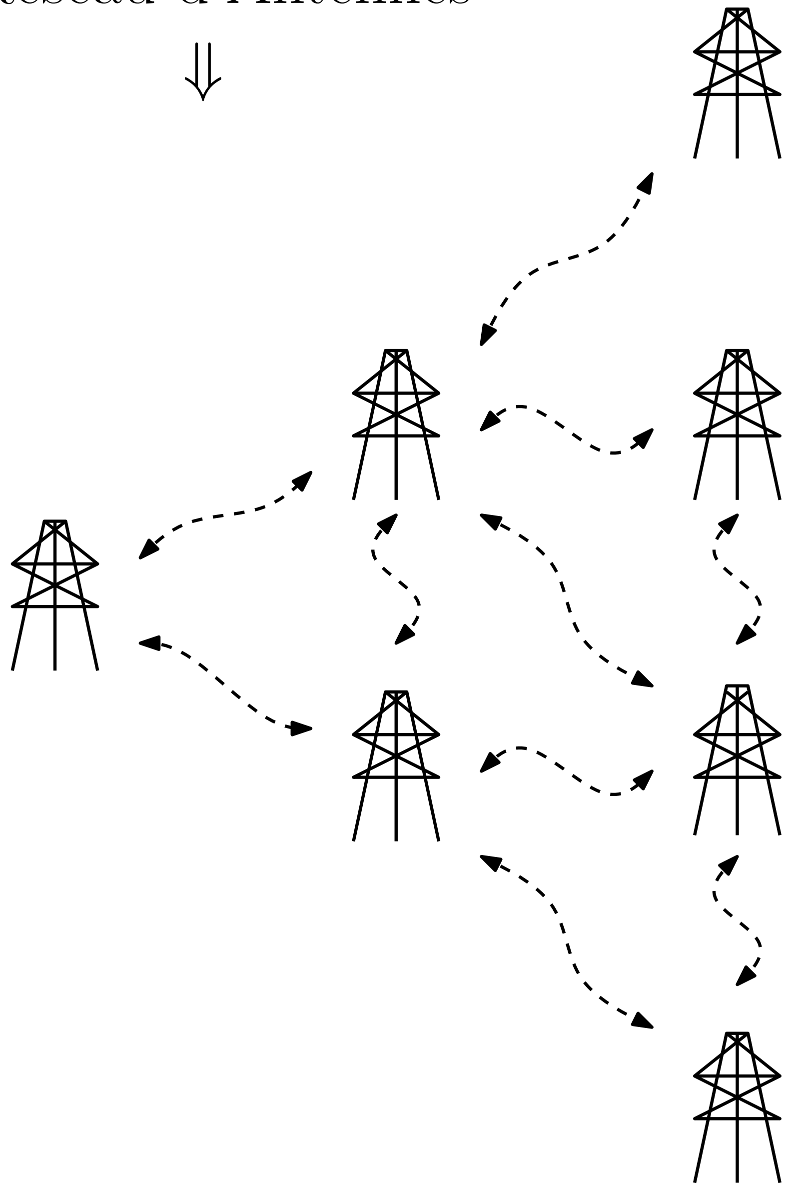
Allocation de Fréquences par Coloration de Graphes

Sommaire

Réseau d'Antennes

$\Downarrow$



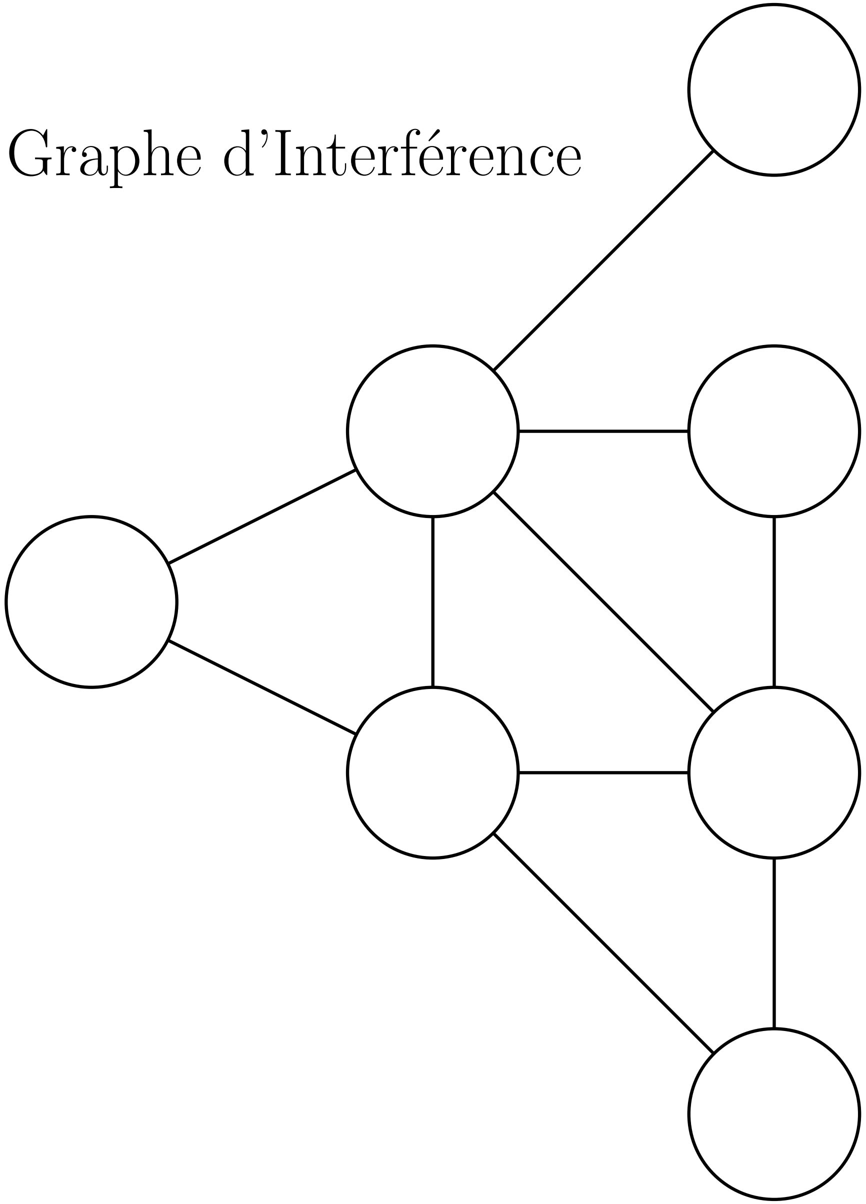
1.1 Modélisation

Antennes $\Rightarrow$

Interférences $\Rightarrow$

Fréquences $\Rightarrow$

Graphe d'Interférence

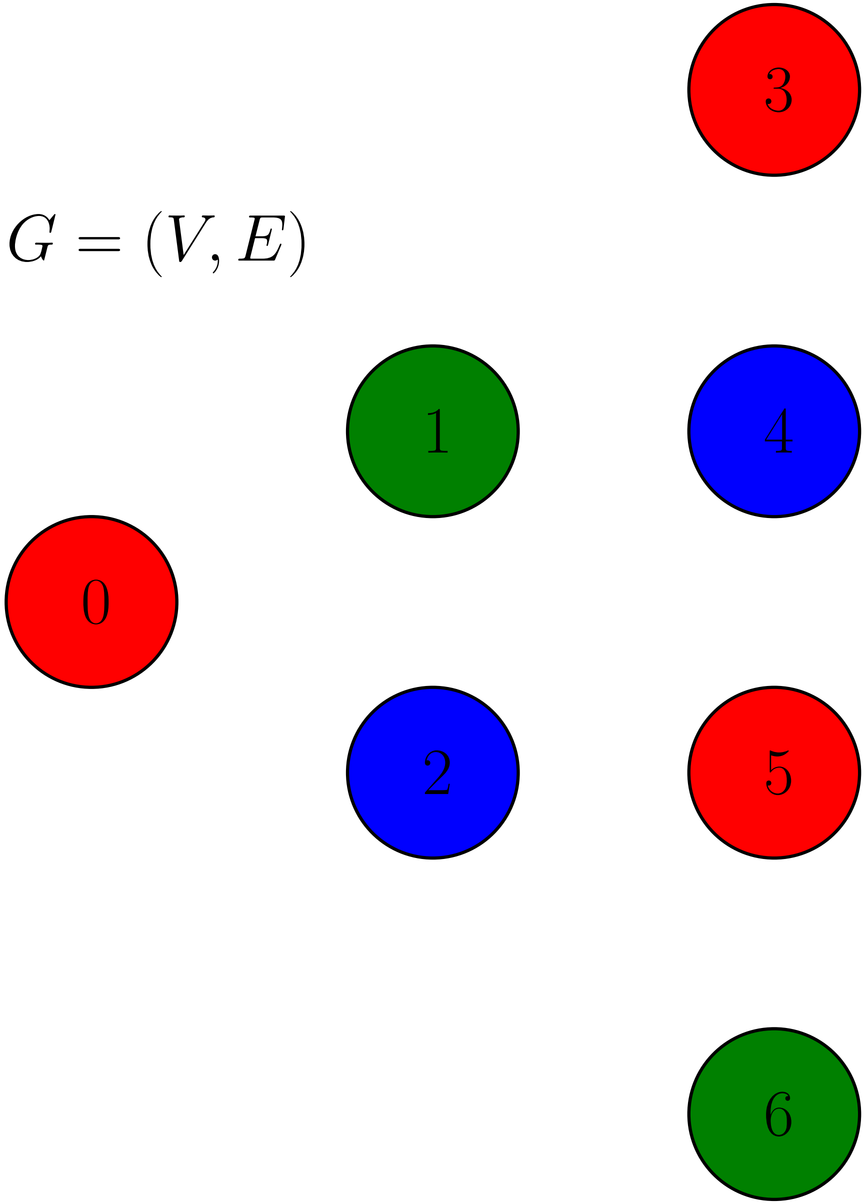


Sommets

Arêtes

Couleurs

$G = (V, E)$



$V = \{0, 1, 2, 3, 4, 5, 6\}$

$E = \{(0, 1), (0, 2), (1, 0), \ldots\}$

Entiers $i \in \mathbb{N}$

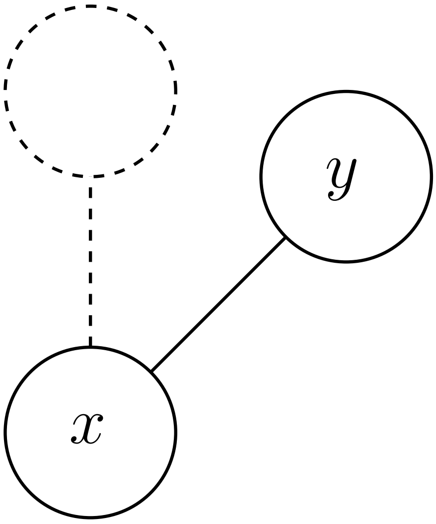
Coloration $\sigma : V \longrightarrow \mathbb{N}$

$\sigma(0) = \sigma(3) = \sigma(5) = 0$

$\sigma(1) = \sigma(4) = 1$

$\sigma(2) = \sigma(6) = 2$

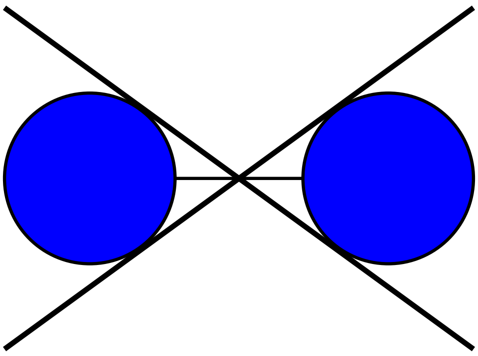
1.2 Vocabulaire



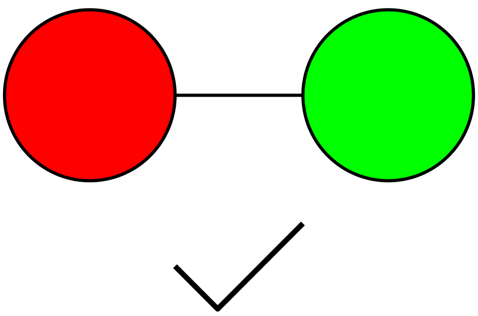
y voisin de x
x, y adjacents $\rightsquigarrow (x, y) \in E$

Ensemble des voisins de x :
$\mathcal{N}(x) = \{y \in V, (x, y) \in E\}$

Degré de x : $d(x) = |\mathcal{N}(x)|$



Coloration propre $\sigma : V \rightarrow \mathbb{N}$
$(x, y) \in E \Rightarrow \sigma(x) \neq \sigma(y)$



$\chi(G) = \min\{|\sigma(V)|, \sigma \text{ coloration propre}\}$

Trouver une $\chi(G)$-coloration dans le cas général :
Problème NP-Complet

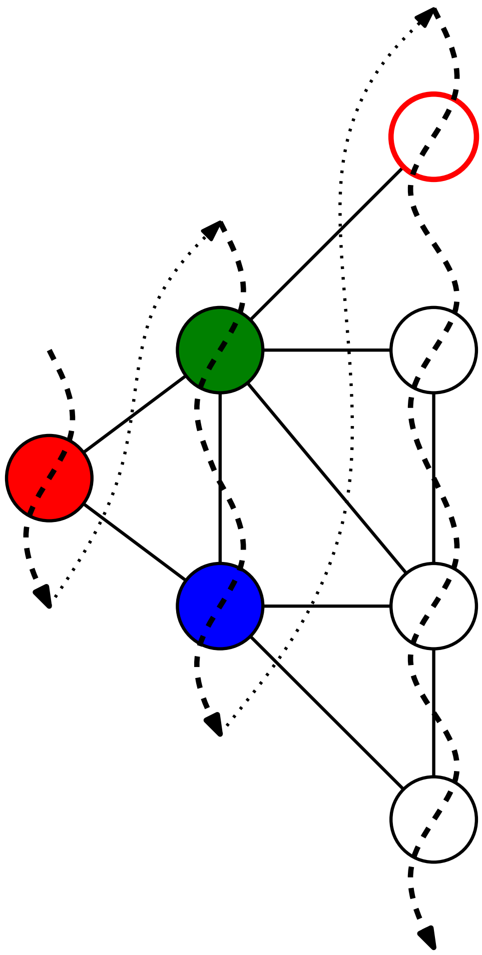
2.1 Coloration Séquentielle

Entrée - Un graphe G, un ordre de visite $\sigma : [\![1, n]\!] \to V$

Sortie - Une coloration propre c

- Colorer les sommets les uns après les autres, selon l'ordre de visite σ,
- Sommet $x \leftarrow$ affecter la plus petite couleur non utilisée dans $\mathcal{N}(x)$

Complexité : $O(n + m)$



Notations :
$n = |V|$, nombre de sommets
$m = |E|$, nombre d'arêtes

0 1 2

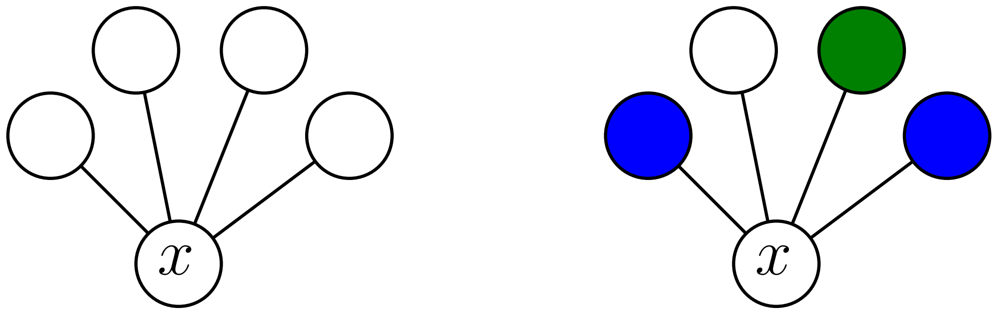
2.2 Deux Heuristiques différentes

Heuristique	Principe	Complexité	Idée
Welsh-Powell	Colorer séquentiellement par ordre de degré décroissant	$O(n+m)$ ou $O(n\ln(n)+m)$	Les sommets ayant le plus de voisins sont plus difficiles à colorer
DSATUR	Colorer les sommets par ordre de degré de saturation décroissant	$O(n^2)$ ou $O(n^2+nm)$	On colore d'abord les sommets qui ont le plus de voisins de couleurs différentes

2.3 L'heuristique DSATUR

$DSAT(x)$ = degré de saturation de x

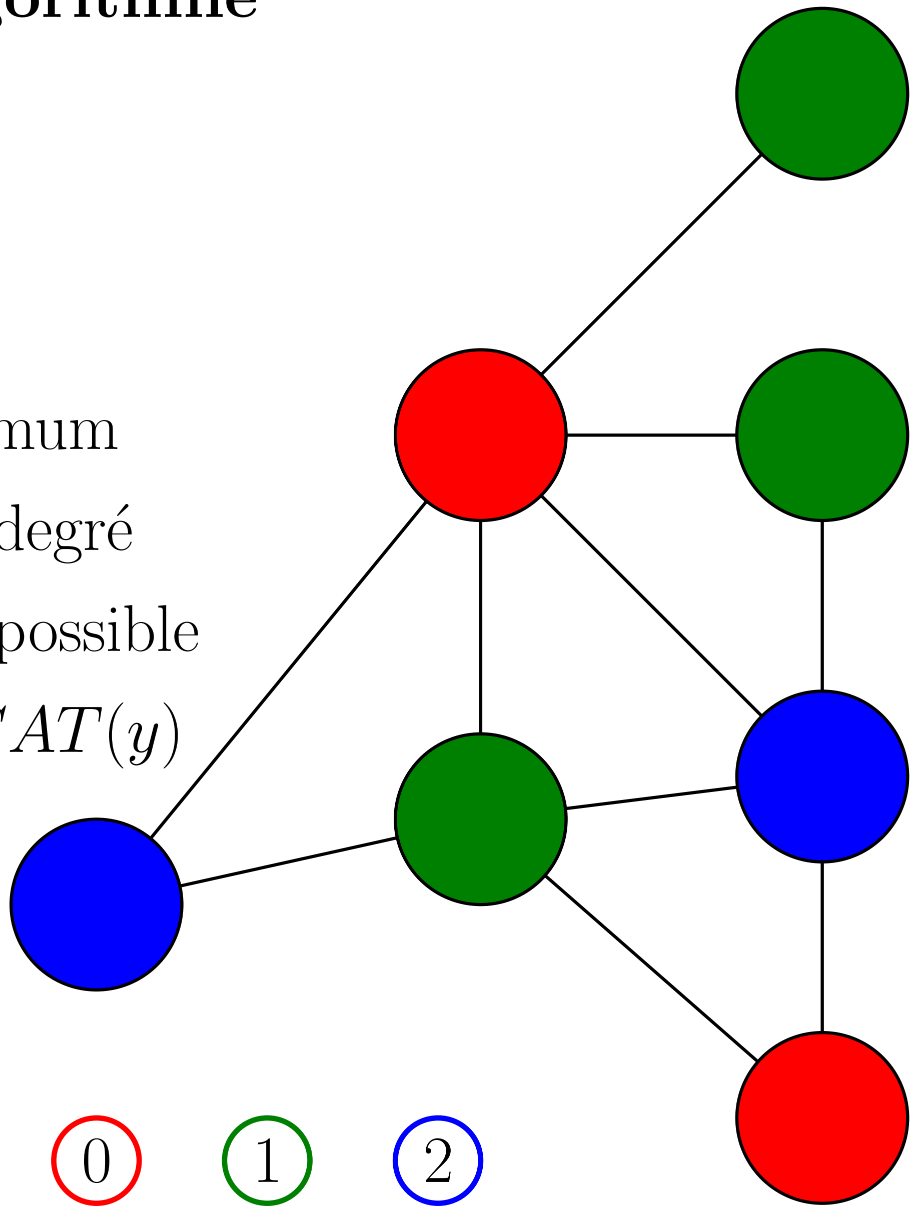
- Si aucun voisin n'est coloré, $DSAT(x) = d(x)$
- Sinon, $DSAT(x) =$ nombre de couleurs différentes utilisées parmi les sommets de $\mathcal{N}(x)$



$DSAT(x) = 4$ $\qquad$ $DSAT(x) = 2$

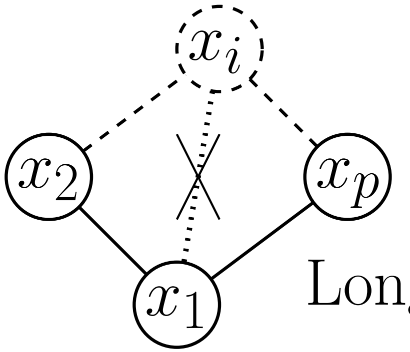
2.4 L'Algorithme

- Choisir un sommet x de DSAT maximum
- Si conflit, prendre celui de plus haut degré
- Colorer x avec la plus petite couleur possible
- Pour y dans $\mathcal{N}(x)$, mettre à jour $DSAT(y)$



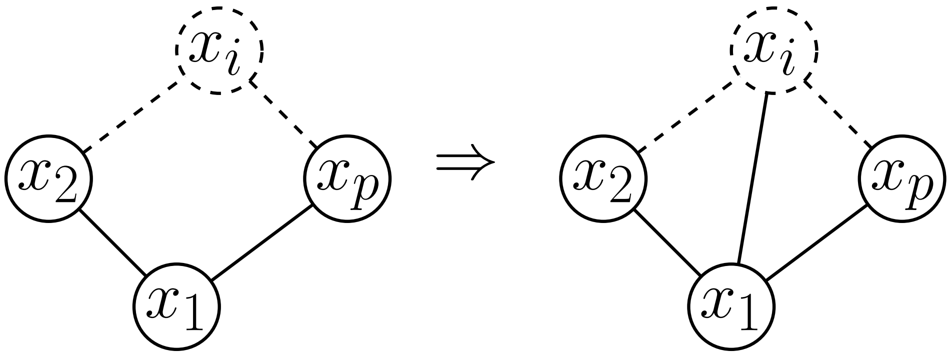
3.1 Définitions

Trou : Cycle sans corde



Longeur p

Graphe Triangulé



$(p \geq 4)$

Sommet simplicial :

Tous ses voisins sont adjacents 2 à 2

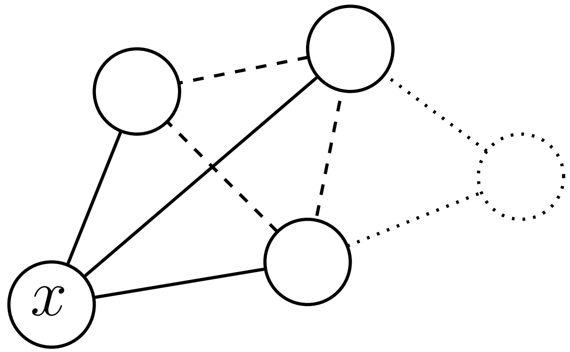
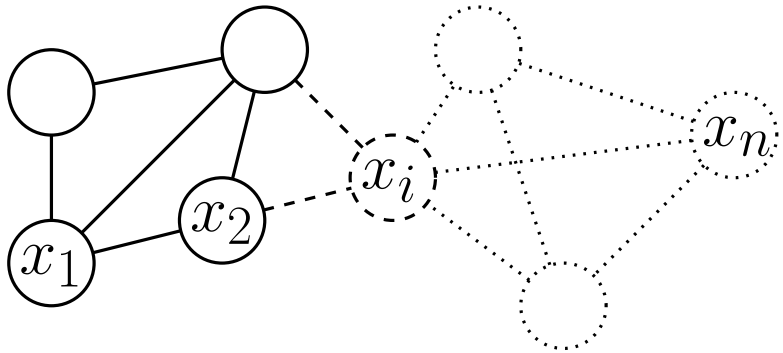


Schéma d'élimination simplicial :

$x_1, x_2, ..., x_n$

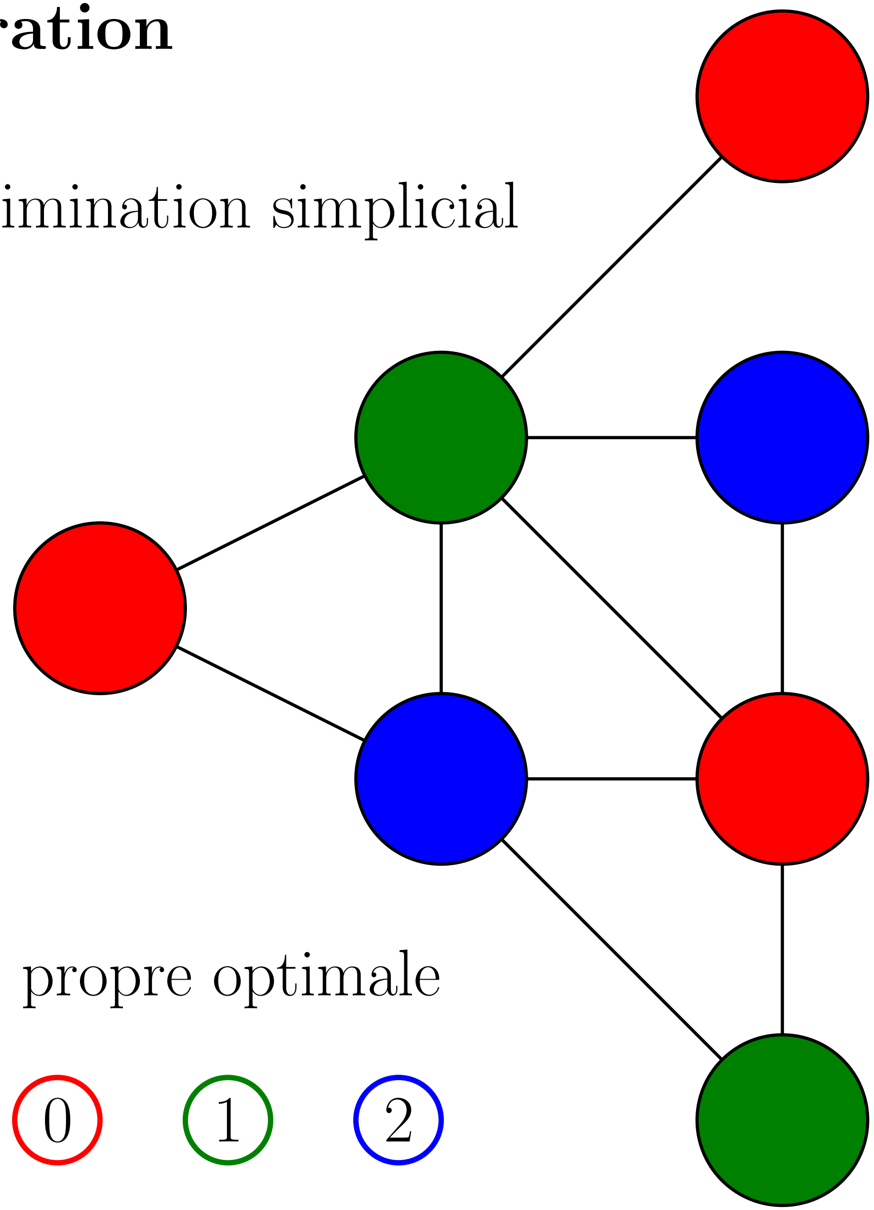


3.2 Coloration

G triangulé $\Leftrightarrow$ G possède un schéma d'élimination simplicial

G triangulé
$$\Downarrow$$
Ordre LexBFS = Schéma d'élimination simplicial

+ Coloration séquentielle $\Rightarrow$ Coloration propre optimale



3.3 LexBFS

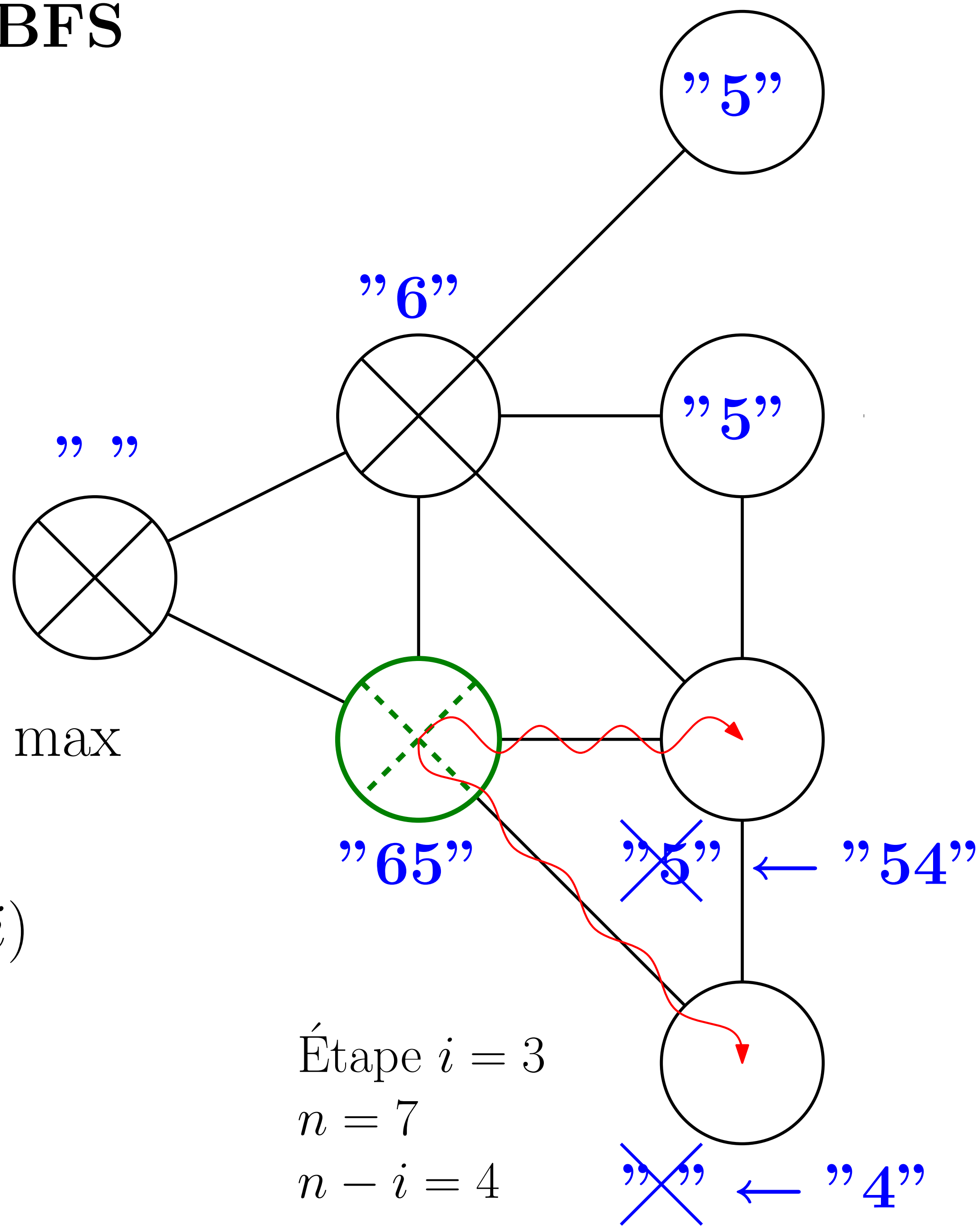
Ordre Lexicographique $\preceq$

- Initialisation : $\forall x, L(x) = $ " "

<u>Étape $i \in [\![1, n]\!]$:</u>

- Prendre un sommet x d'étiquette $L(x)$ max
- $x \leftarrow$ Affecter numéro i
- $\forall y \in \mathcal{N}(x)$, faire $L(y) \leftarrow L(y) \cdot (n - i)$

Complexité $O(n + m)$



Conclusion

- Les temps d'exécution varient selon l'implémentation
- L'efficacité dépend aussi des heuristiques
- Applications multiples

Les problèmes de DIMACS

• Allocation de registe :

→ Allouer des registres du processeur aux variables d'un code

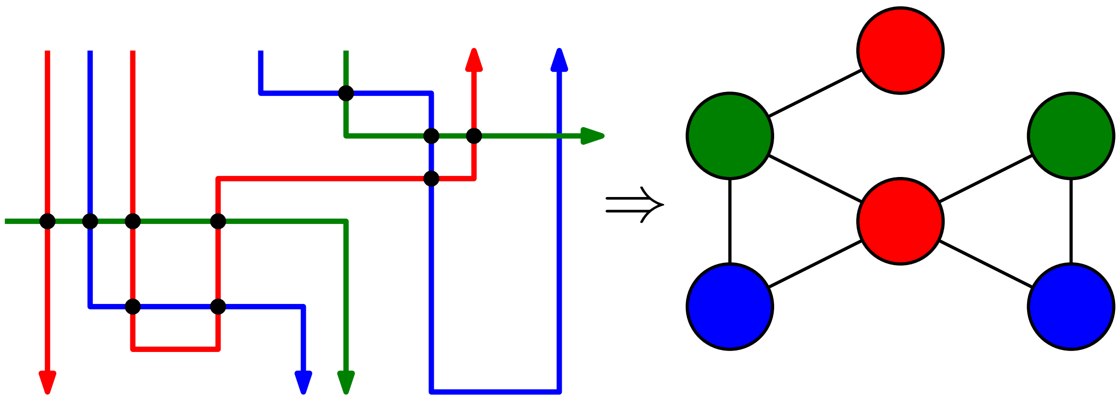
• Planification :

→ Donner des horaires à des cours (classe + prof)

• Graphes de Mycielski M_k :

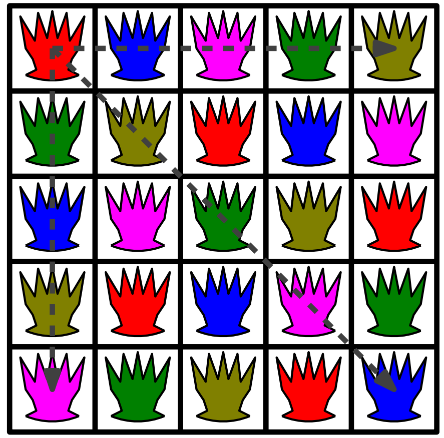
$\rightarrow \nexists$ triangle ; $\chi(M_k) = k$

• Réseaux optiques :



• Graphes de reines :

→ Placer correctement n fois n reines sur un échiquier de taille $n \times n$

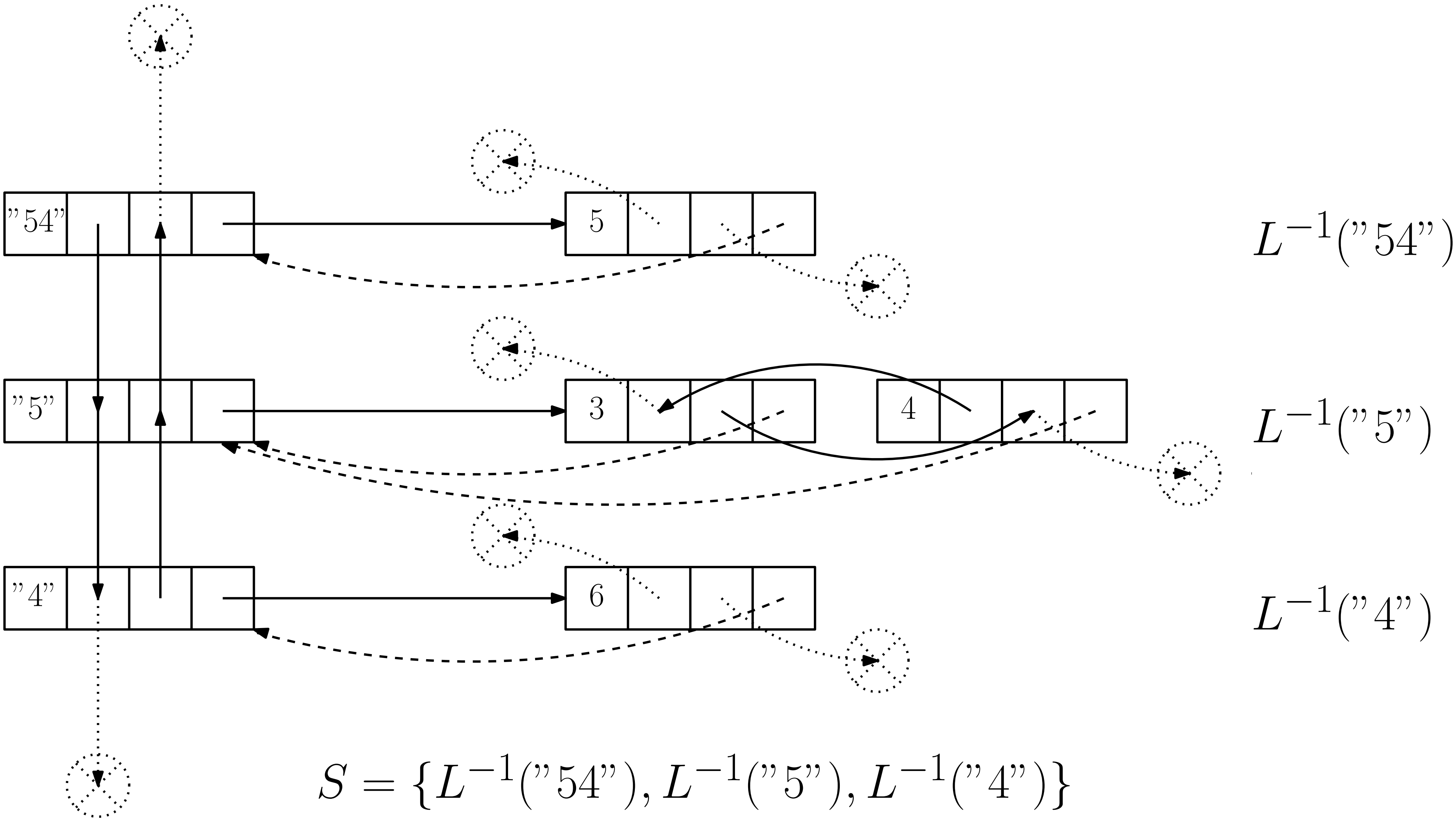


• Carrés Latins :

1	2	$\cdots$	$\cdots$	n
2	$\vdots$			1
$\vdots$	$\vdots$			$\vdots$
$\vdots$	n			$\vdots$
n	1	$\cdots$	$\cdots$	$n-1$

n

Affinage de partition



$$S = \{L^{-1}("54"), L^{-1}("5"), L^{-1}("4")\}$$

$S =$ ensemble des sommets restant à parcourir

Reconnnaissance de Graphes Triangulés

$$PN(x) = \{y \in \mathcal{N}, \sigma(y) < \sigma(x)\}$$

$$\sigma \text{ simplicial} \Leftrightarrow \forall x, PN(x) \text{ est une clique}$$

$p(x) = \sigma^{-1}(\max \sigma\langle PN(x)\rangle)$ s'il existe, sommet de $PN(x)$ visité en dernier

Alors, si $PN(\sigma^{-1}(i))$ est une clique pour $0 \leq i \leq k-1$ et $x = \sigma^{-1}(k)$:

$$PN(x) \text{ est un clique} \Leftrightarrow PN(x) \setminus p(x) \subseteq PN(p(x))$$