COLLÈGE
DE FRANCE
1530

Aléa et algorithmes distribués

Claire Mathieu

cnrs

Deux problèmes, trois algorithmes distribués

Stable maximal
Popularité des pages web (Pagerank)

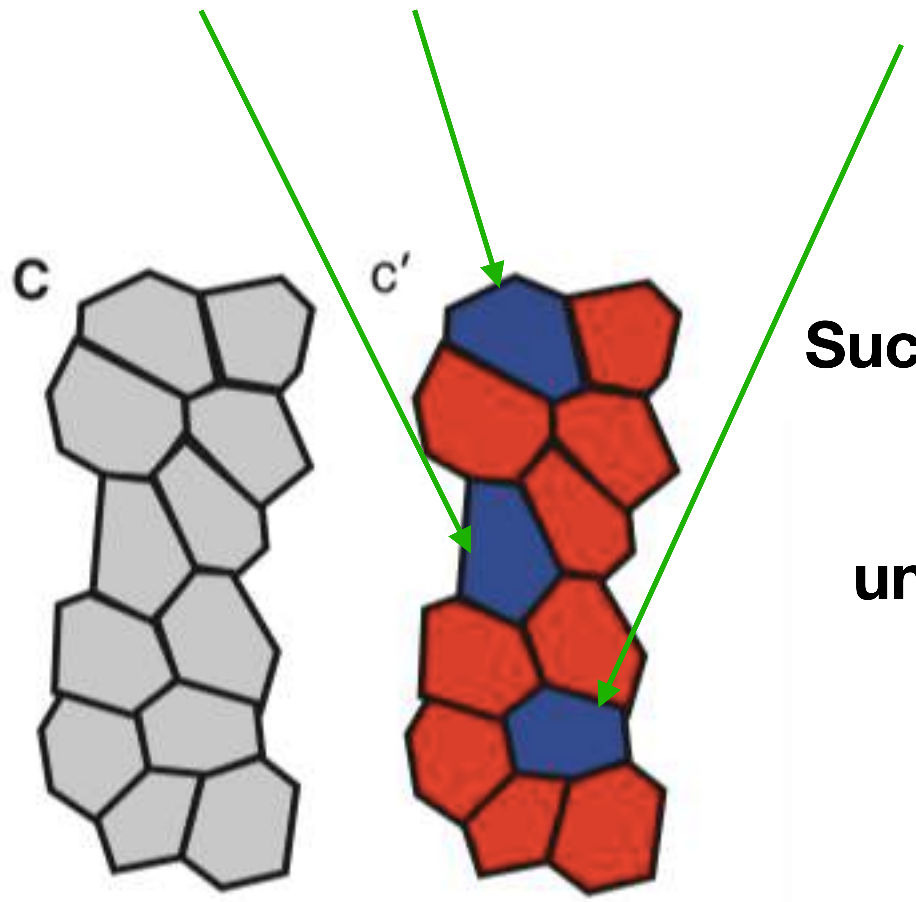
Stable maximal

Stable maximal

mouche drosophile

développement du système nerveux :
sélection des cellules “sensory organ precursor”

inhibition des cellules voisines par expression de la protéine Delta

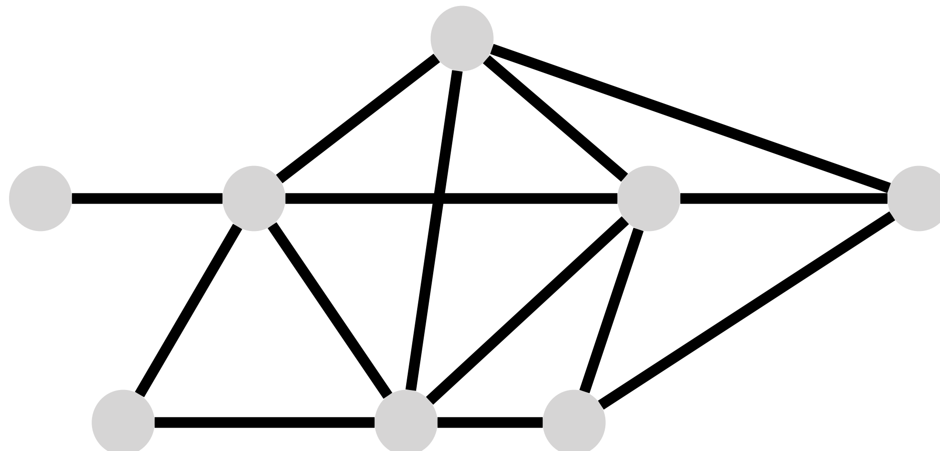


Succès : chaque cellule est, soit sélectionnée, soit inhibée par une voisine sélectionnée

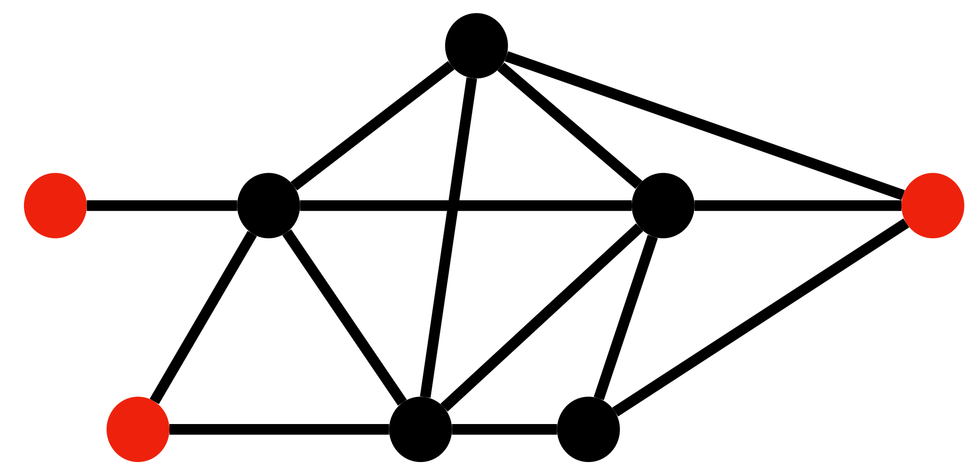
Anarchie : pas de contrôle central pour faire la sélection
pas de connaissance de la structure globale par les cellules
pas de communication autre qu’aux cellules voisines

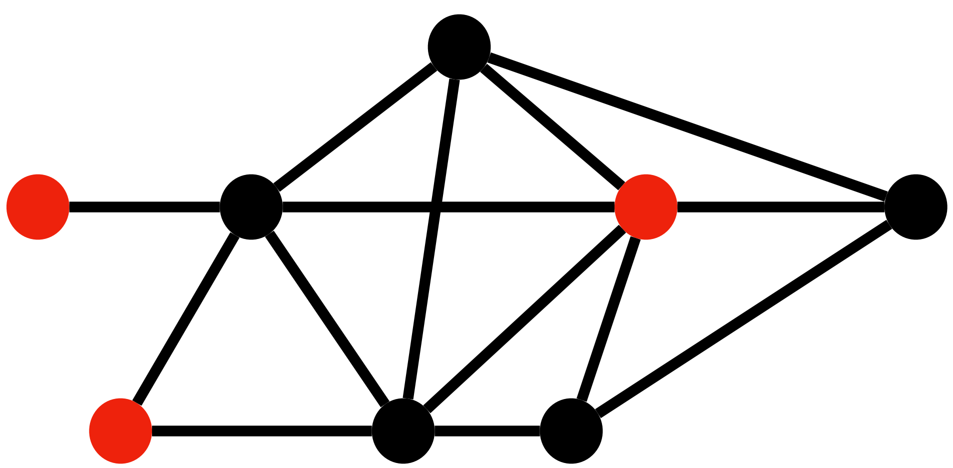
Si toutes les cellules exécutent le protocole alors : succès

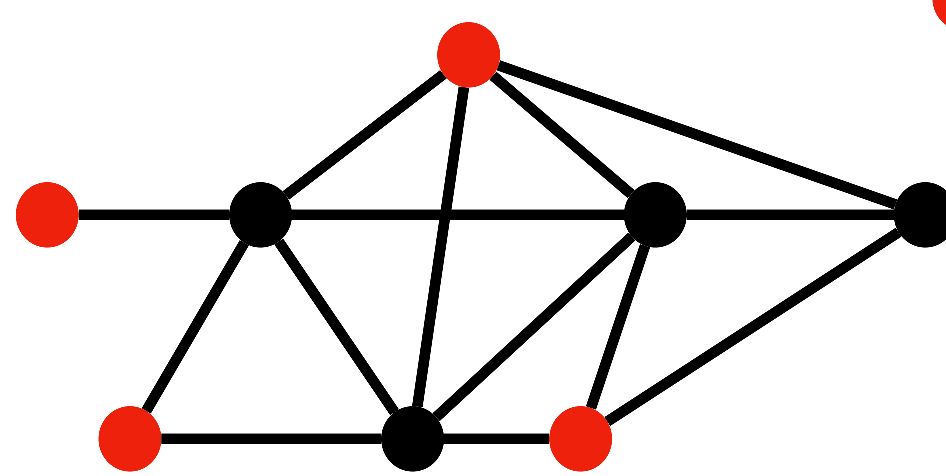
Le problème du stable maximal

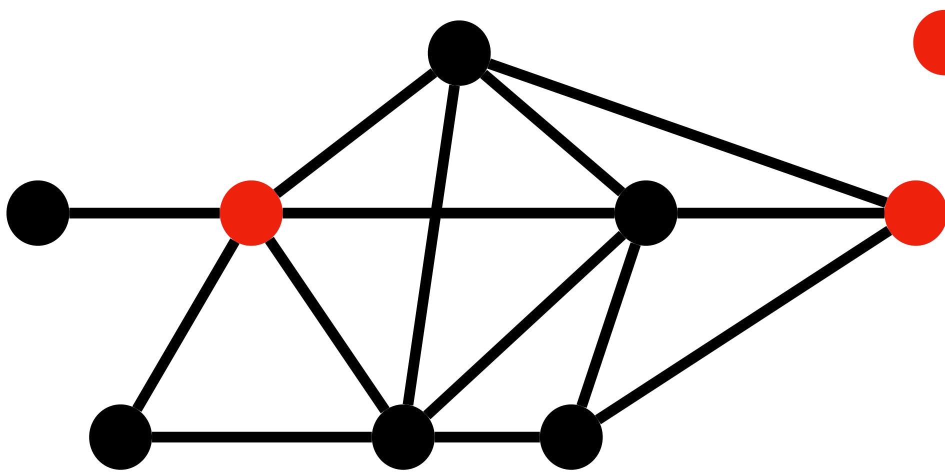


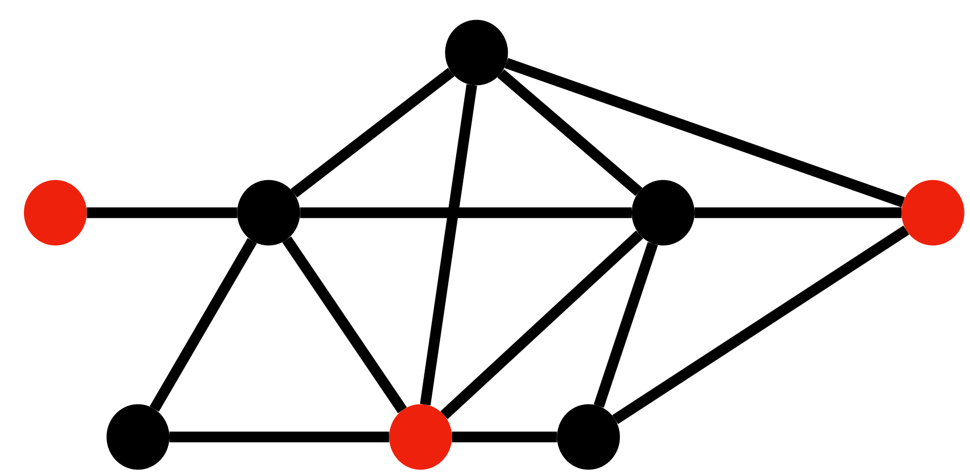
Choisir des “décideurs locaux”
- **pas voisins les uns des autres**
- **tout sommet est décideur ou voisin d’un décideur**





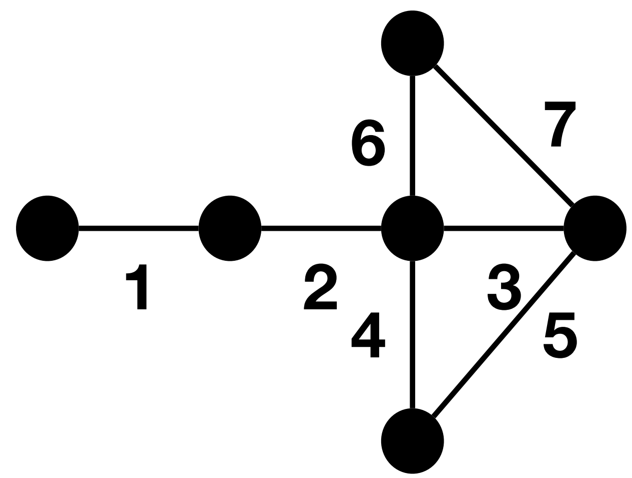


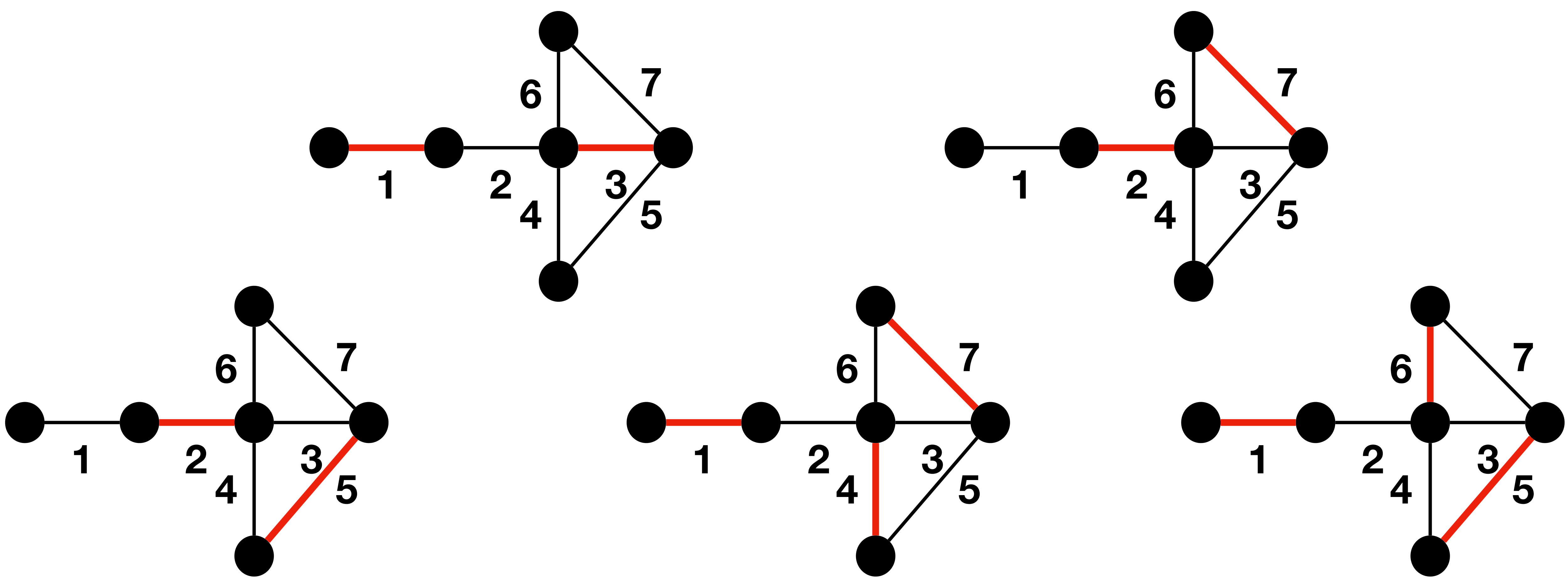


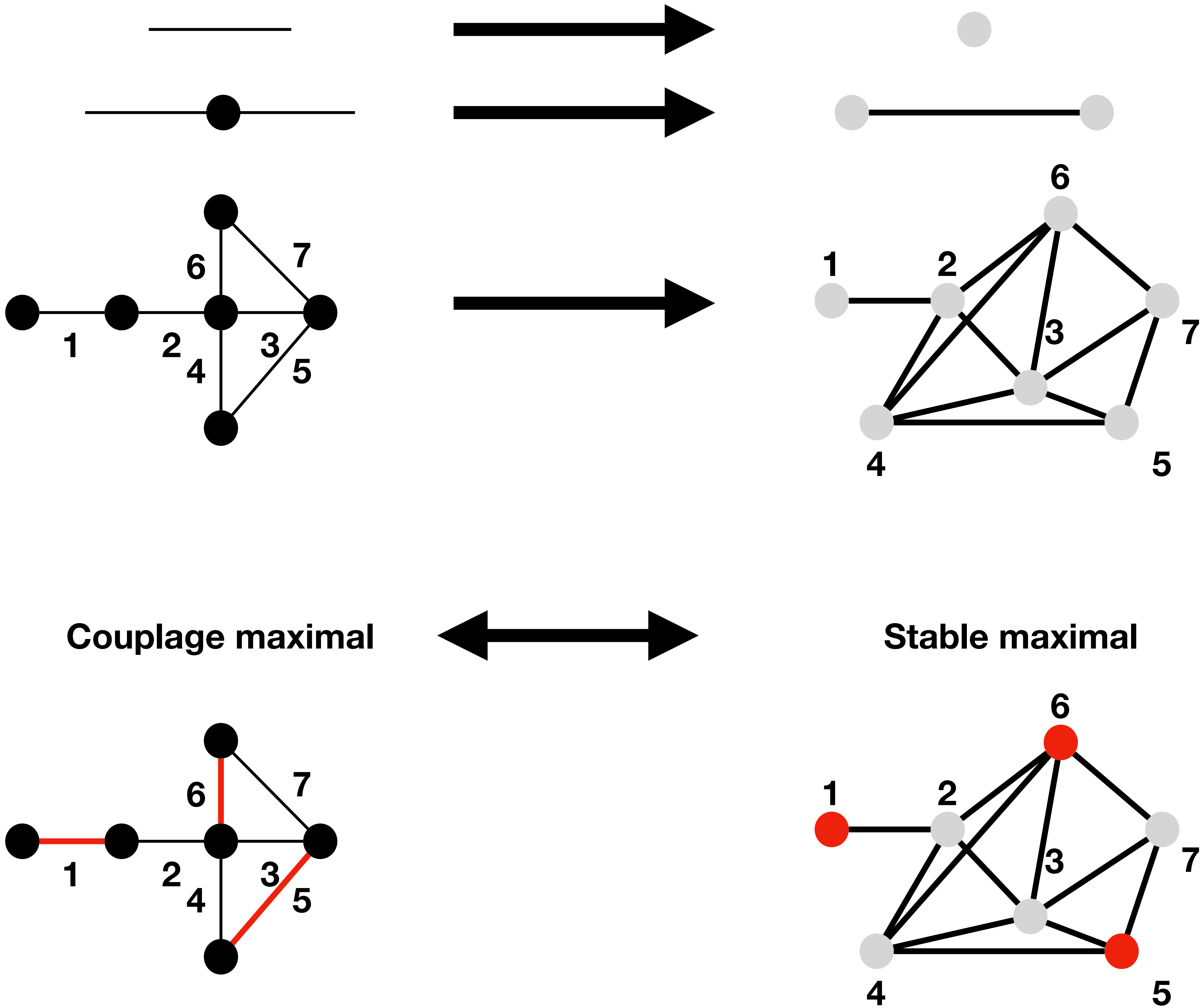


Que peut-on faire avec?

Couplage maximal



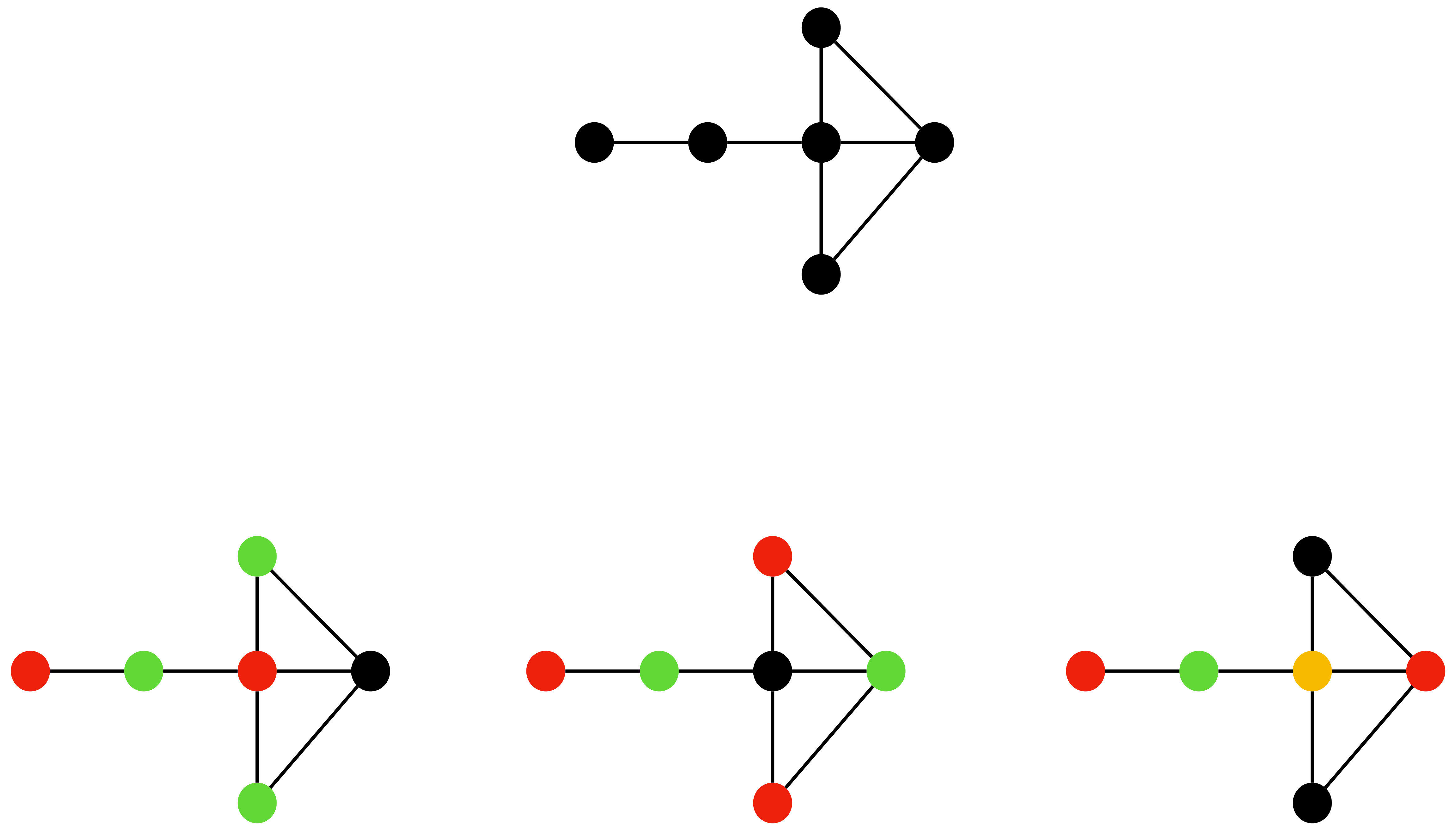




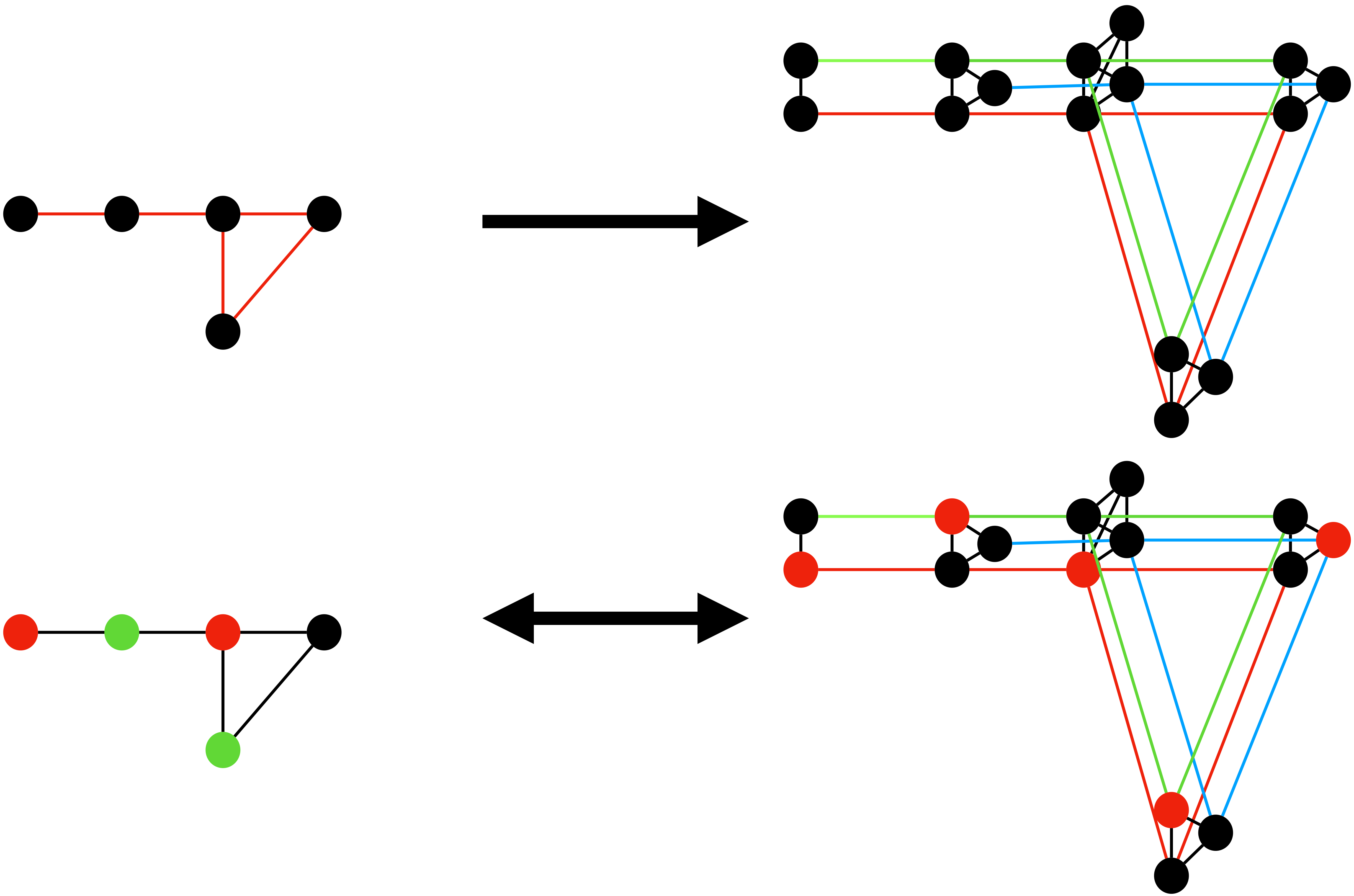
6
7
1
2
3
4
5
Couplage maximal
Stable maximal

Que peut-on faire avec?

Coloriage des sommets d'un graphe

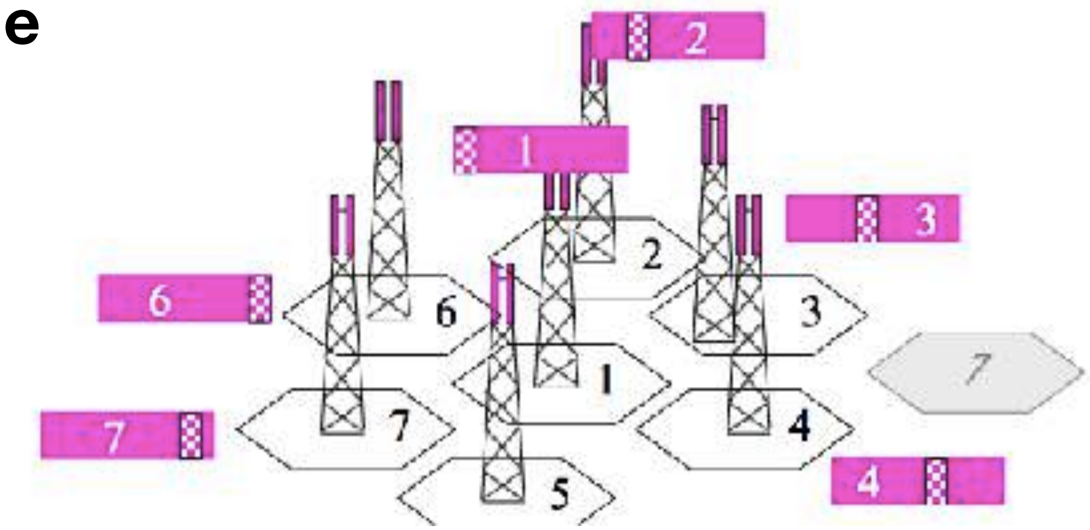


Que peut-on faire avec?

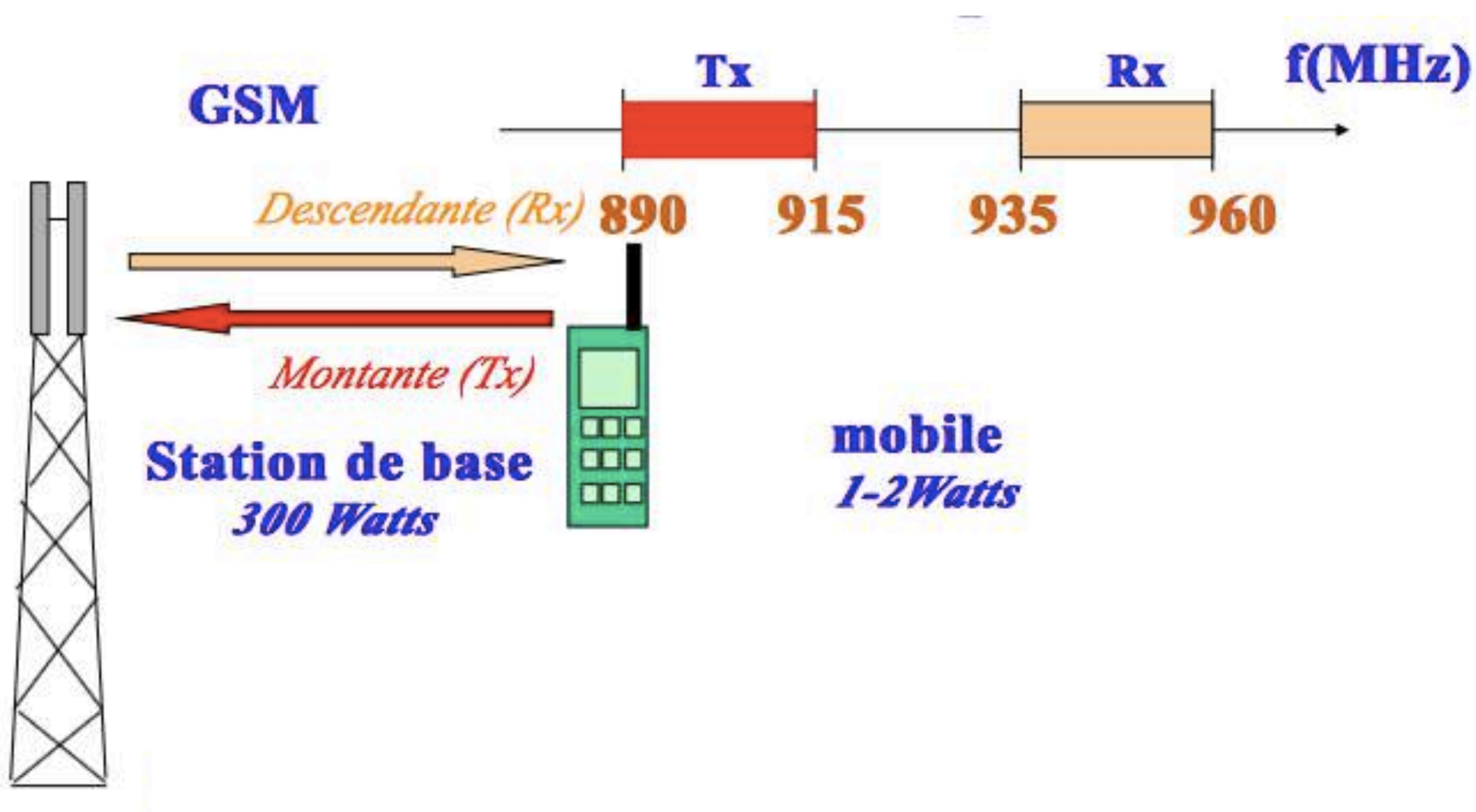


Pourquoi colorier les sommets d'un graphe ?

Allocation de fréquences pour téléphonie mobile



Réseau cellulaire GSM
cellules géographiques (hexagonales)
tour de communication dans la cellule
bandes de fréquences différentes
pour éviter interférences
entre cellules voisines

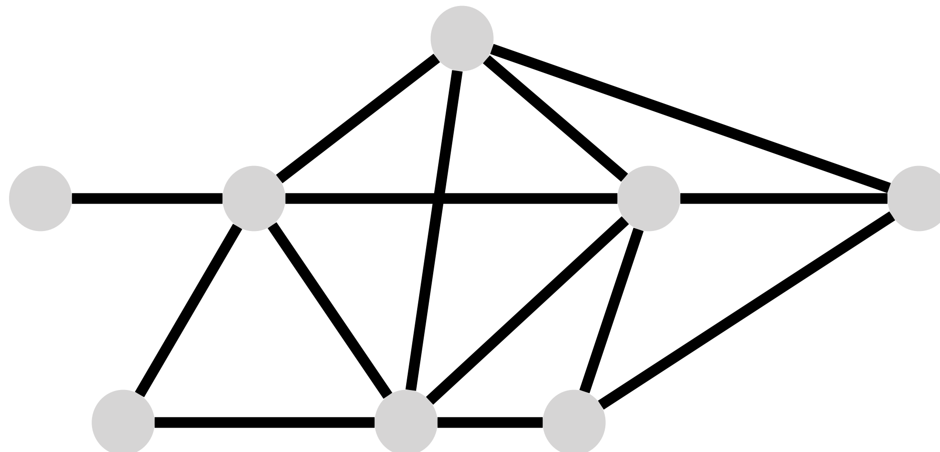


Répartition des fréquences

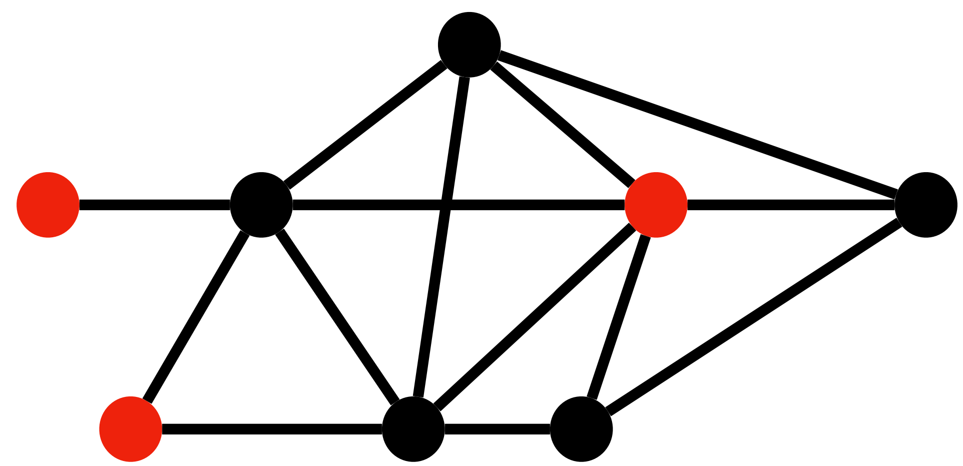
Au coeur de l’informatique :
le principe des réductions

Stable maximal
Couplage maximal
Coloriage de graphe
Allocation de fréquences

Le problème du stable maximal

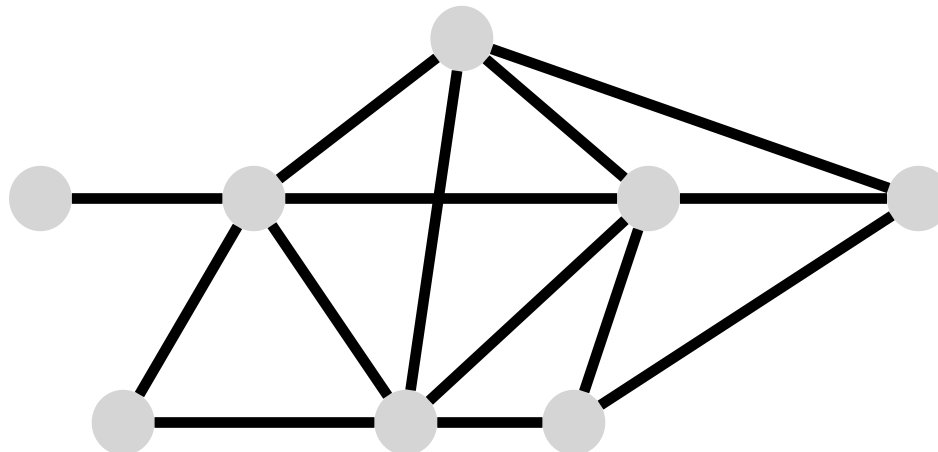


Choisir des “décideurs locaux”
- **pas voisins les uns des autres**
- **tout sommet est décideur ou voisin d’un décideur**



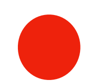
Choix des décideurs par décisions locales

Le problème du stable maximal

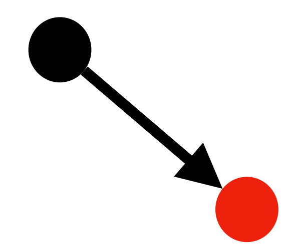


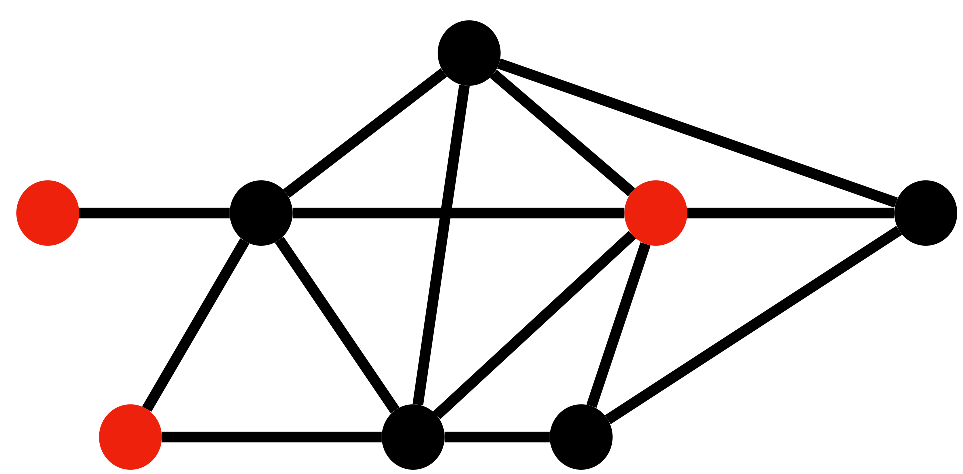
Choisir des “décideurs locaux”
- pas voisins les uns des autres
- tout sommet est décideur ou voisin d’un décideur

- Initialement : incertitude, pas de décideur connu
- À la fin: chaque noeud,

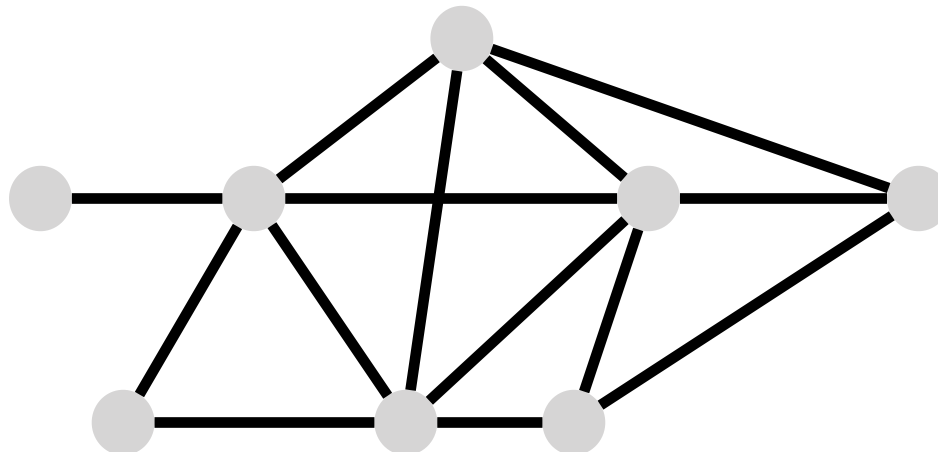
ou bien sait qu’il est décideur, 

ou bien sait qu’il ne l’est pas et connaît un voisin décideur



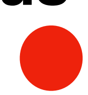


Le problème du stable maximal

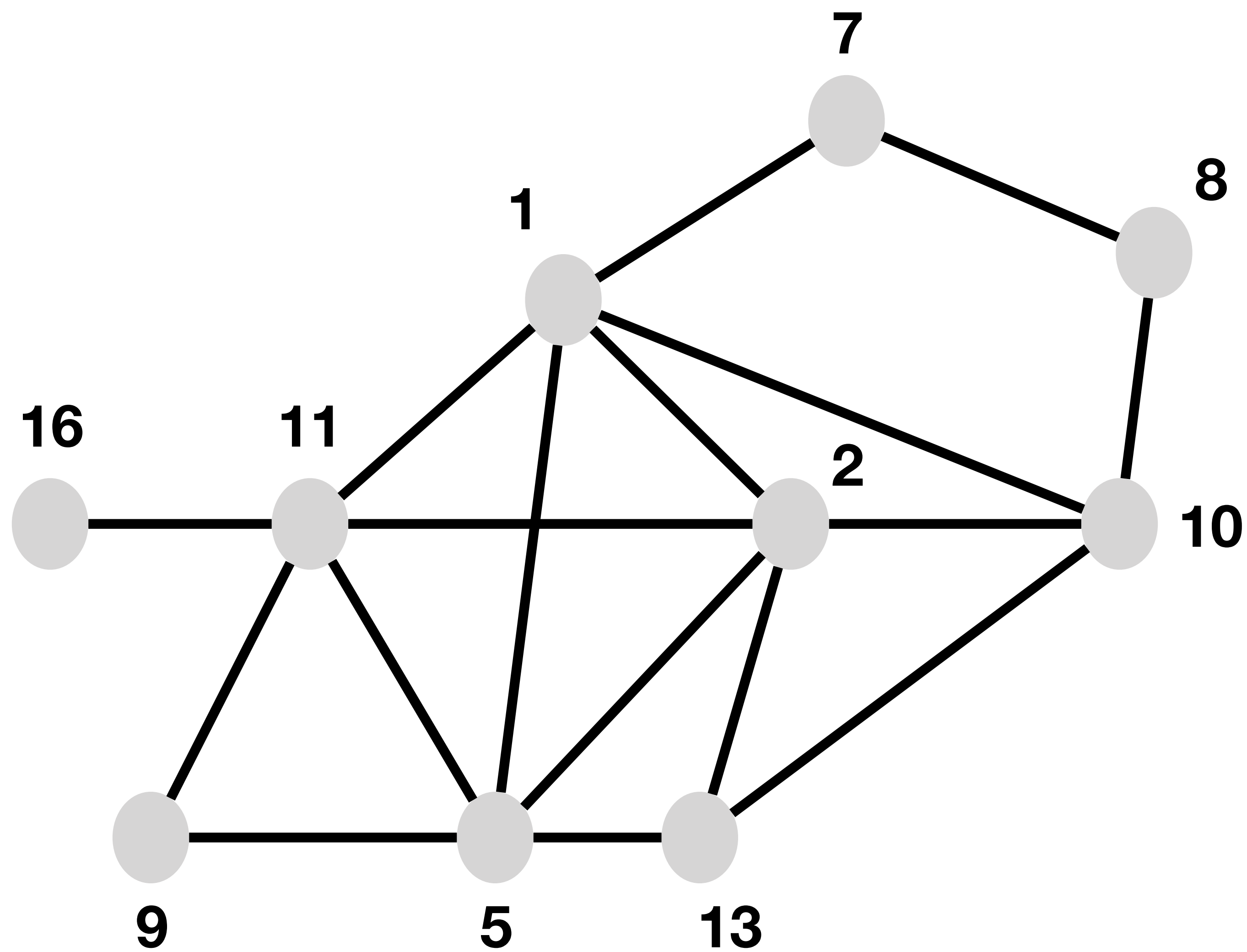


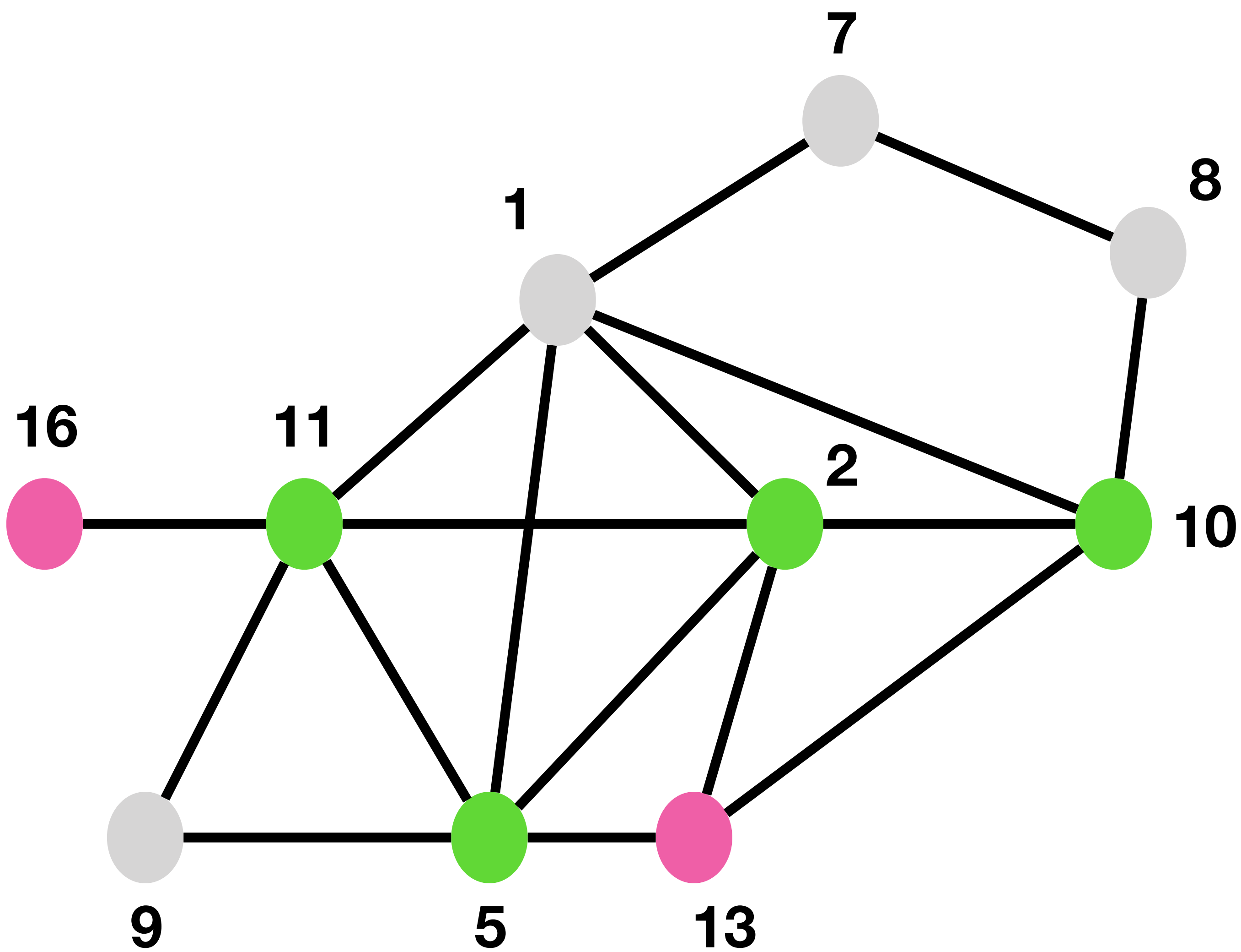
Choisir des “décideurs locaux”
- pas voisins les uns des autres
- tout sommet est décideur
ou voisin d’un décideur

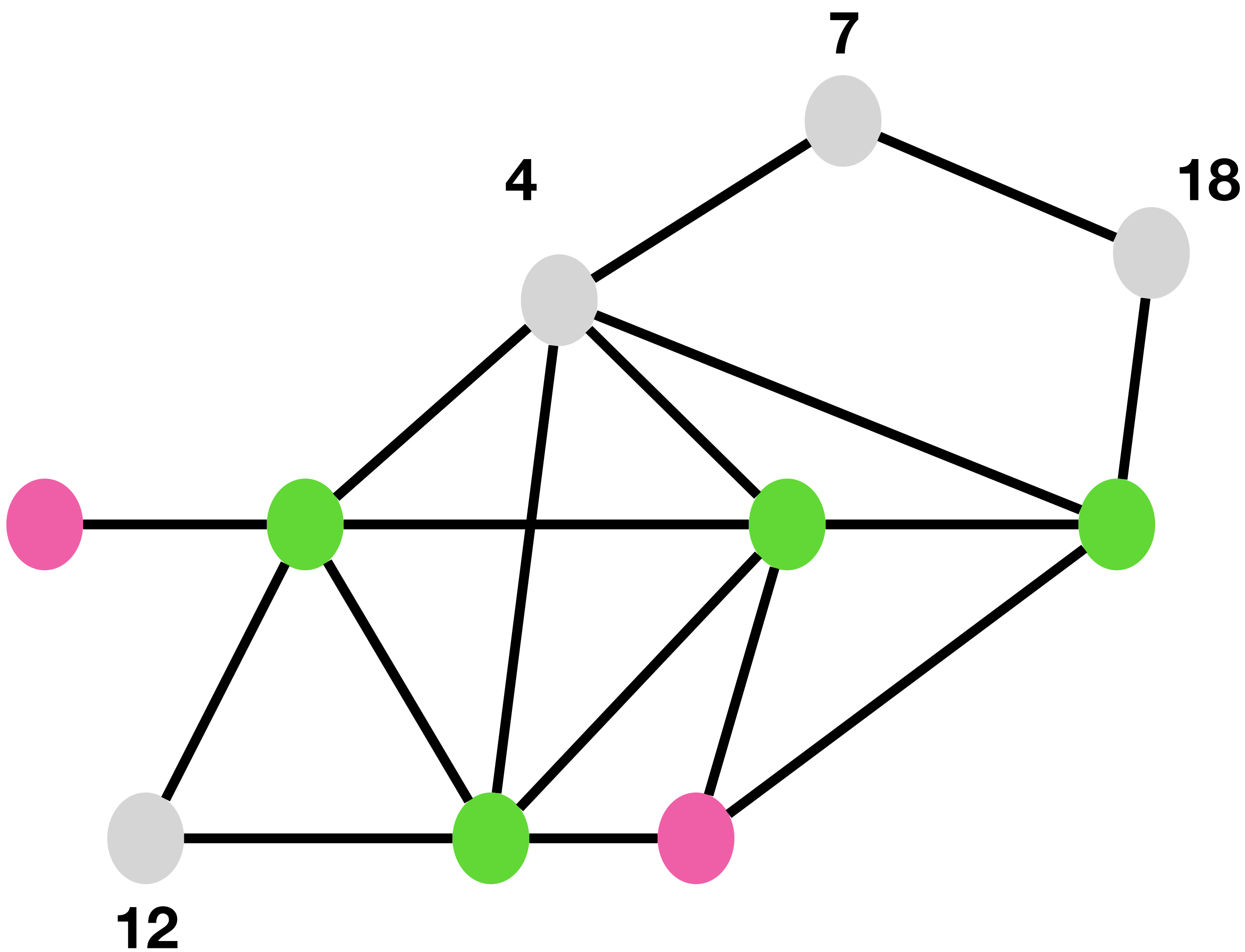
- Initialement : incertitude, pas de décideur connu, tout le monde est actif

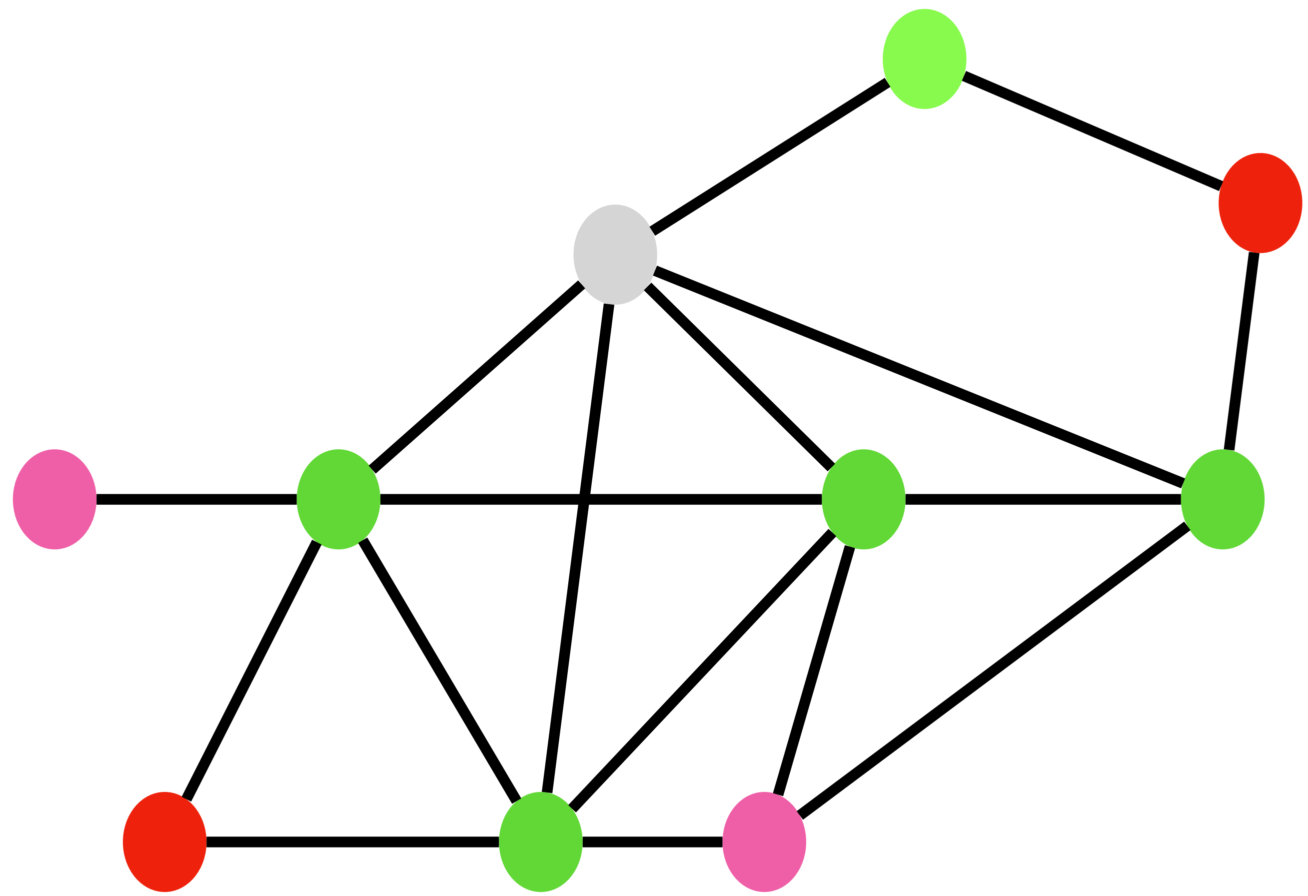
au milieu, répéter :
chaque noeud encore actif
choisit une valeur
l’envoie à ses voisins
et reçoit les valeurs de ses voisins
si c’est lui qui a la plus grande valeur,
il devient décideur 
et en prévient ses voisins 
qui deviennent inactifs

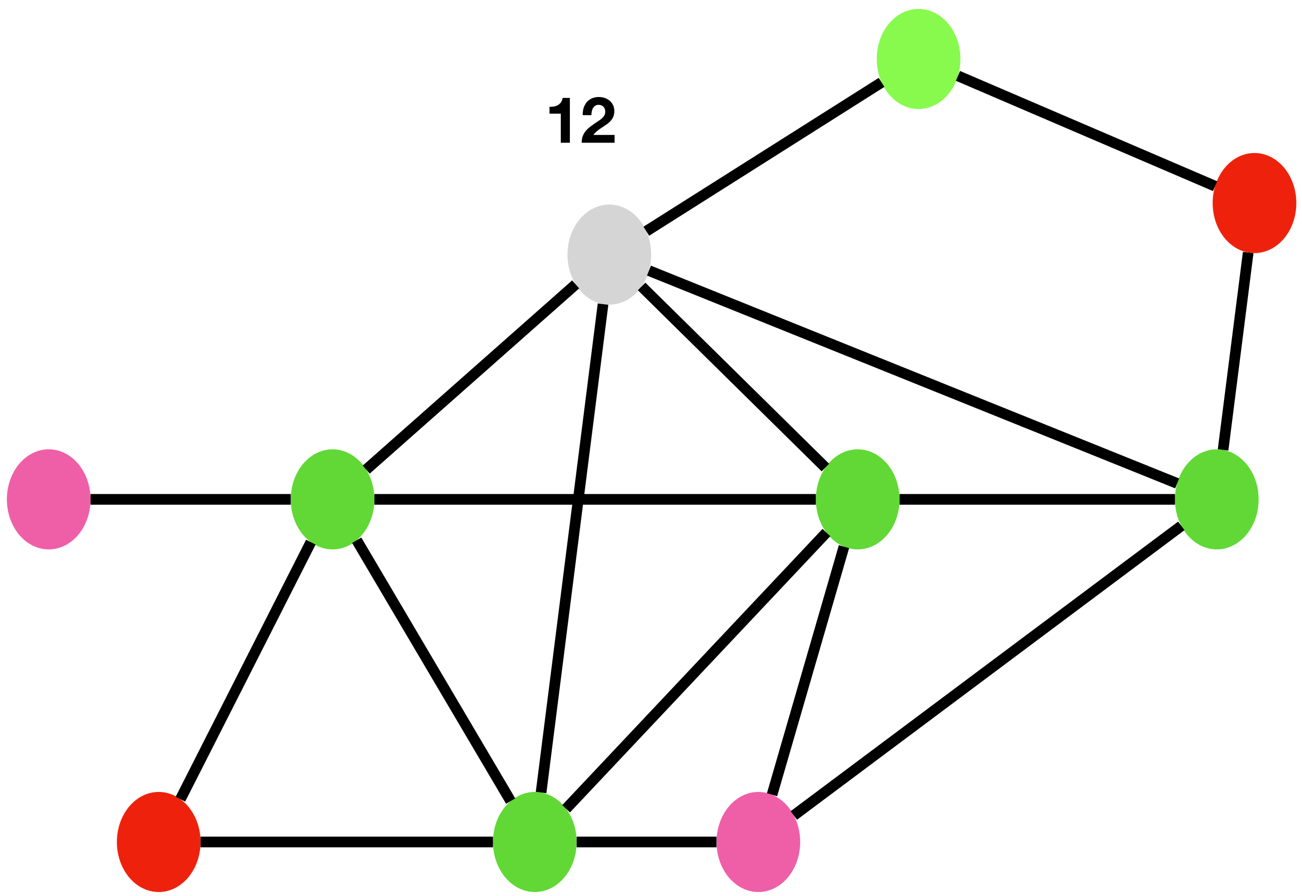
- À la fin: plus personne n’est actif, et chaque noeud,
ou bien sait qu’il est décideur,
ou bien sait qu’il ne l’est pas et connaît un voisin décideur

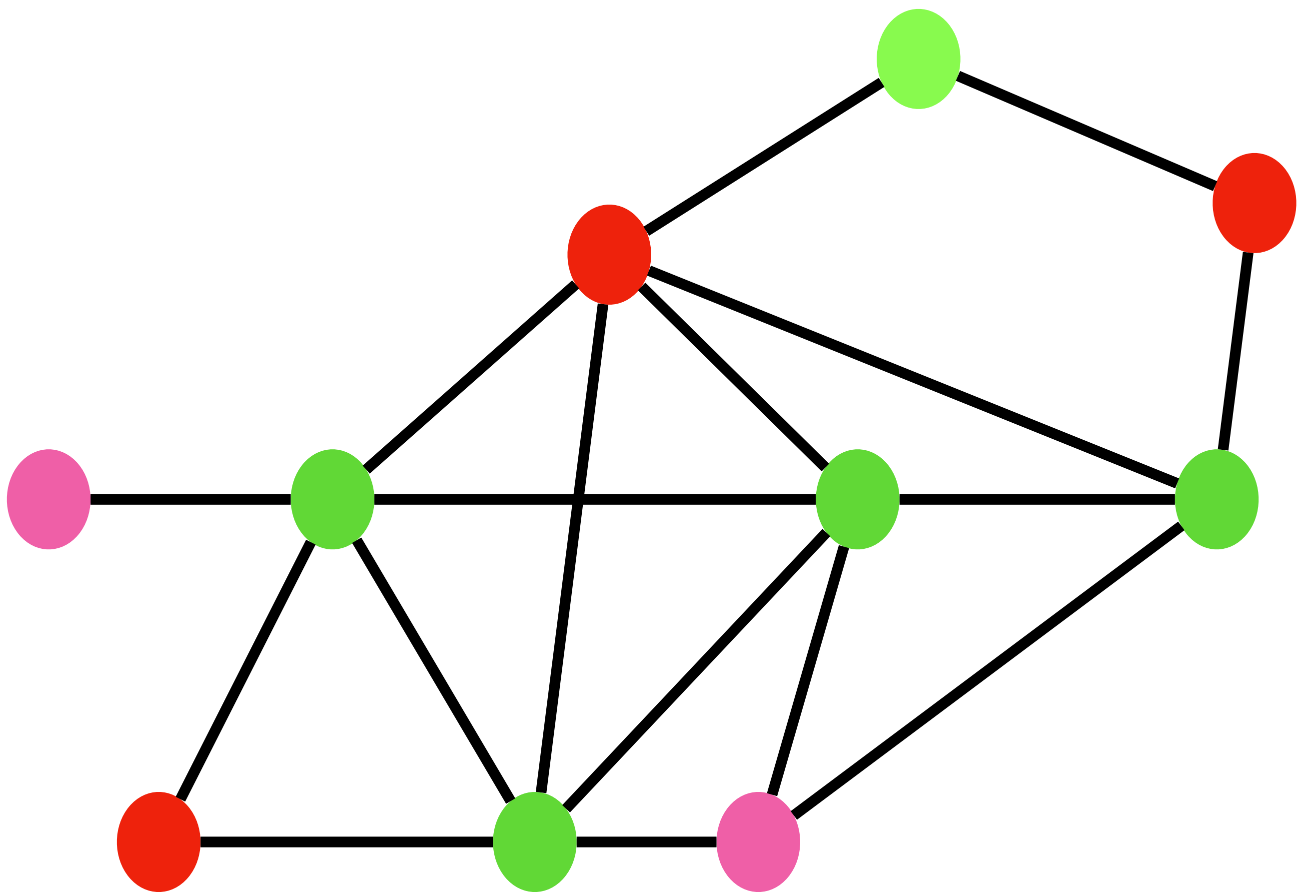

7
8
1
16
11
2
10
9
5
13


7
8
1
16
11
2
10
9
5
13


7
4
18
12

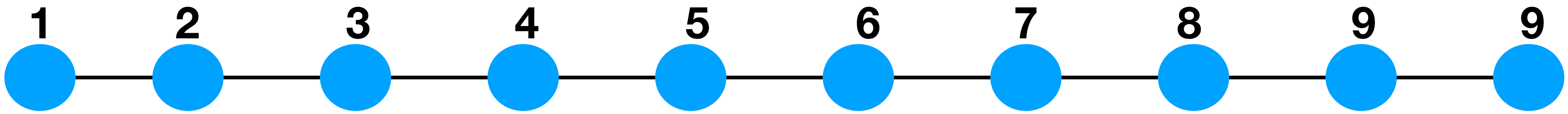



12



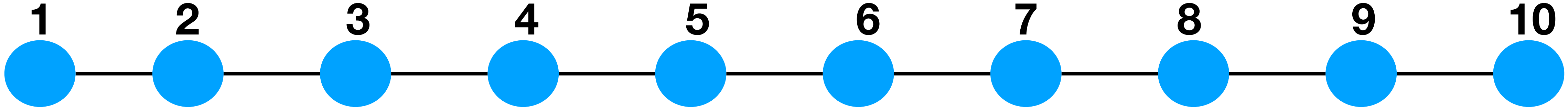
On obtient un stable maximal

Au bout de combien de temps?



Aucun progrès !
Le temps peut être infini !

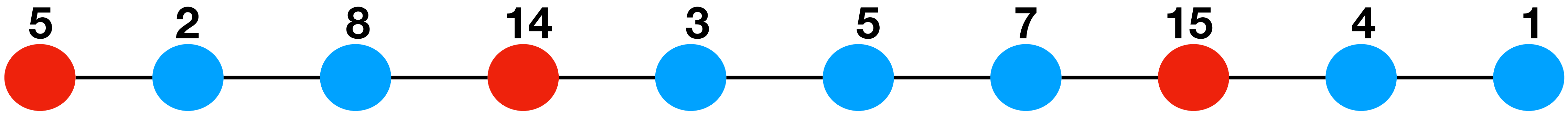
Au bout de combien de temps?



Tout petit progrès !
Le temps peut être très long !

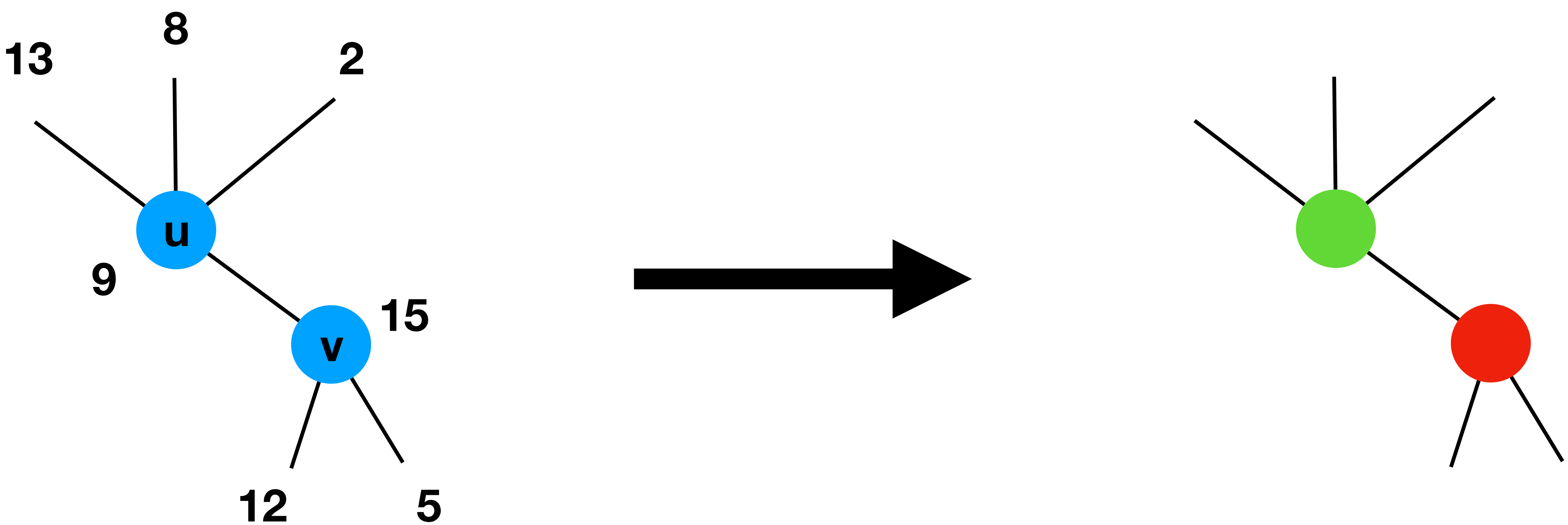
Au bout de combien de temps?

Le rôle de l'aléa :
chaque sommet choisit sa valeur au hasard
dans un ensemble assez grand
pour que les valeurs soient (probablement) toutes distinctes



Progrès rapide !

Au bout de combien de temps?

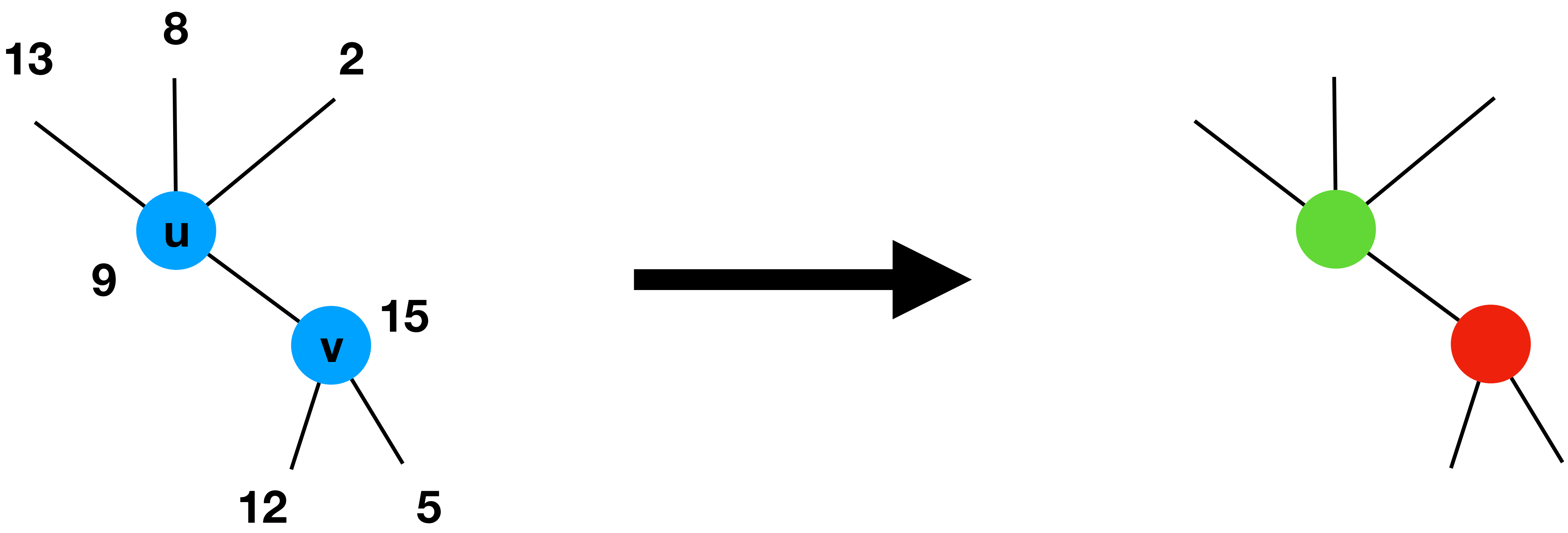


Probabilité que la valeur de v soit plus grande que celle de u et que celles de tous leurs voisins:

$$\frac{1}{d(u)+d(v)}$$

$$\Pr[u \text{ dominé par } v] \geq \frac{1}{d(u)+d(v)}$$

Au bout de combien de temps?



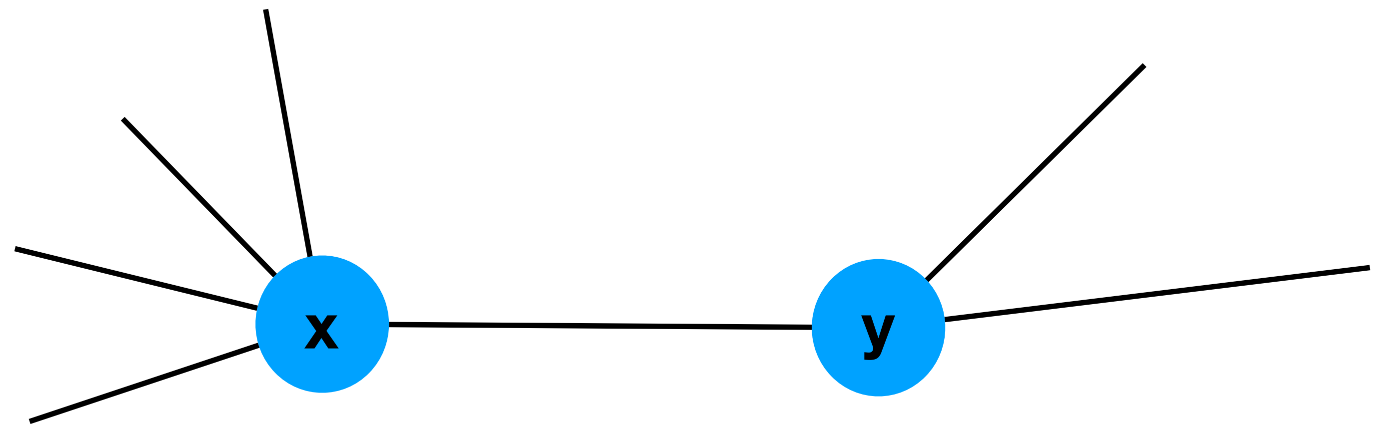
$$\Pr[u \text{ dominé par un voisin}] \geq \sum_{v \sim u} \frac{1}{d(u)+d(v)}$$

$$\Pr[\text{arête } uv \text{ devient inactive}] \geq (1/2) \sum_{w \sim u} \frac{1}{d(u)+d(w)} + \sum_{w \sim v} \frac{1}{d(v)+d(w)}$$

Nombre moyen d'arêtes devenant inactives:

$$(1/2) \sum_{uv \in E} \sum_{w \sim u} \frac{1}{d(u)+d(w)} + \sum_{w \sim v} \frac{1}{d(v)+d(w)}$$

Au bout de combien de temps?



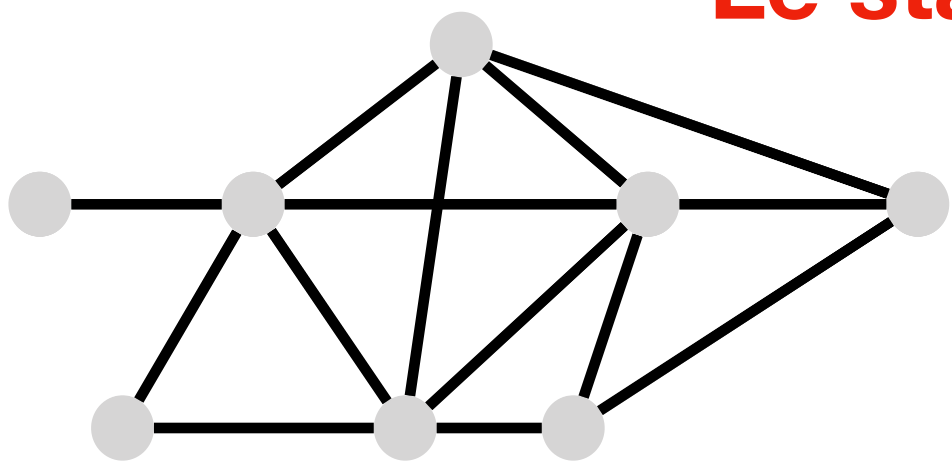
$$(1/2)\sum_{uv\in E}\sum_{w\sim u}\frac{1}{d(u)+d(w)}+\sum_{w\sim v}\frac{1}{d(v)+d(w)}$$

Question : Combien de fois a-t-on le terme 1/(d(x)+d(y))?
Réponse : d(x)+d(y)

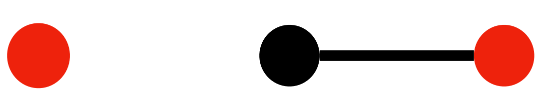
Nombre moyen d'arêtes devenant inactives: |E|/2
Initialement : |E| est au plus quadratique en n
Quand |E|=1 : encore une étape et on a fini

Combien de temps: O(log(n))

Le stable maximal en distribué



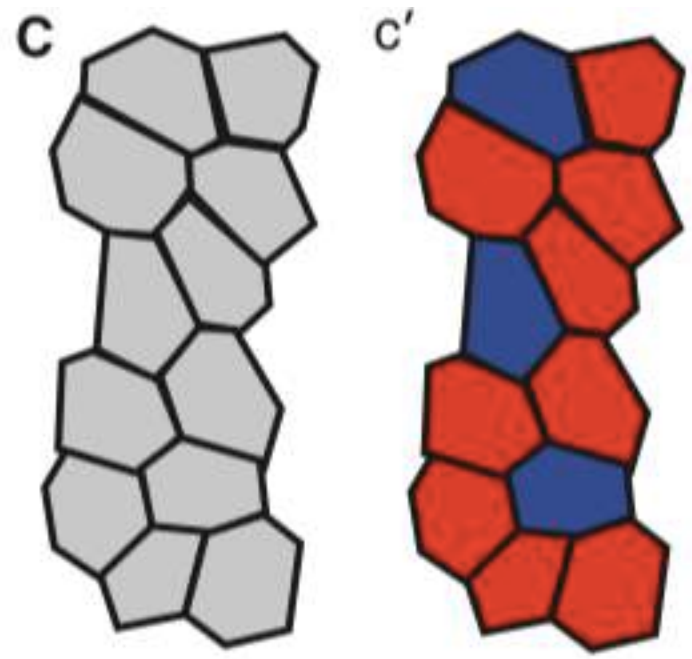
Problème : Choisir des "décideurs locaux"
- pas voisins les uns des autres
- tout sommet est décideur
ou voisin d'un décideur

Algorithme:
Initialement : pas de décideur connu,
tout le monde est actif
Répéter :
chaque noeud encore actif
choisit une valeur
l'envoie à ses voisins
et reçoit les valeurs de ses voisins
si c'est lui qui a la plus grande valeur,
il devient décideur
et en prévient ses voisins
qui deviennent inactifs
À la fin : plus personne n'est actif ;
chaque noeud,
ou bien est décideur, 
ou bien connaît un voisin décideur

O(log n) étapes
en moyenne et
avec grande probabilité
communication
avec les voisins uniquement

Est-ce ainsi que font les cellules des mouches?

inhibition des cellules voisines par expression de la protéine Delta



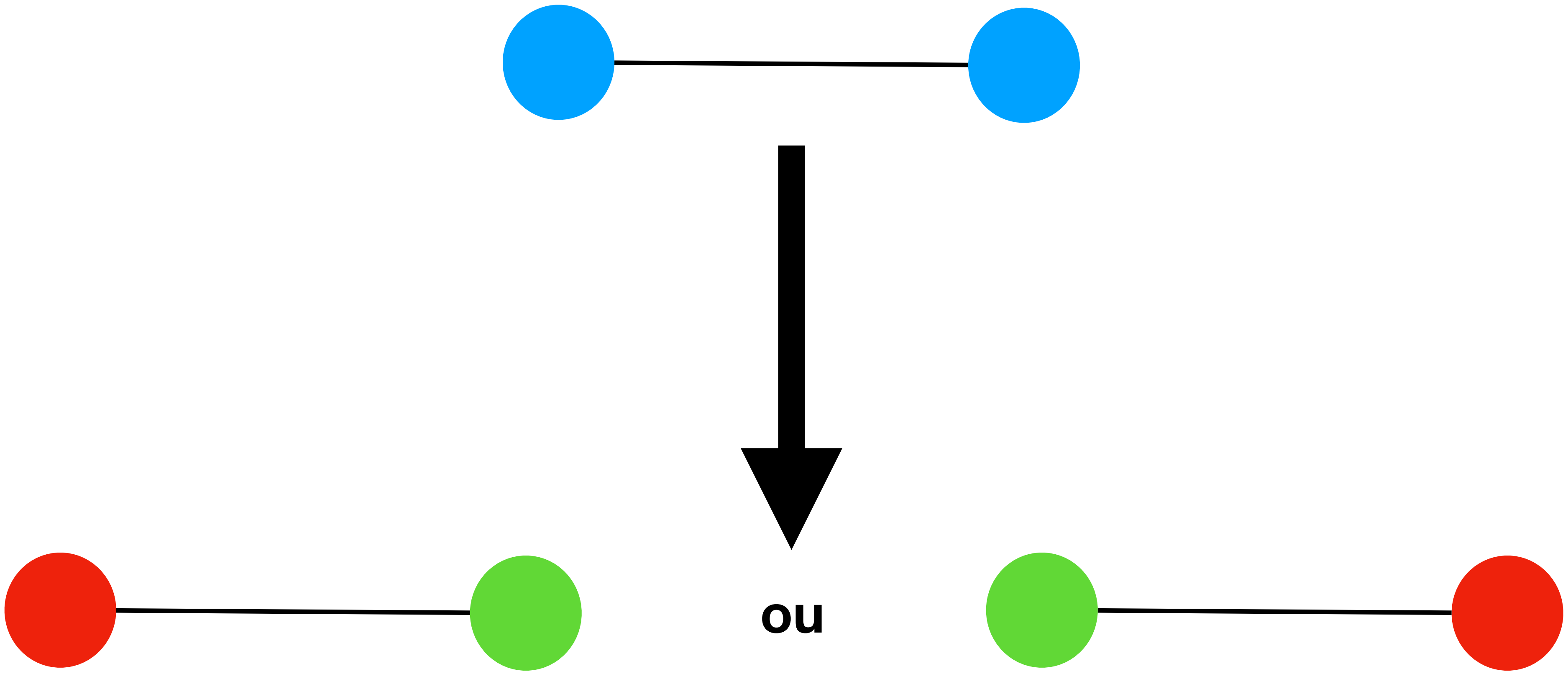
Succès : chaque cellule est, soit sélectionnée, soit inhibée par une voisine sélectionnée

Elles n'envoient à leurs voisins que des bits de communication, 0 ou 1
Une cellule devient décideur local si
elle envoie un 1 à ses voisines
et que celles-ci n'envoient pas de 1
Une cellule encore active devient de plus en plus active
envoie des 1 de plus en plus fréquemment
phase i (répétée log(n) fois) : Probabilité d'envoyer un 1: $2^i/N$

$O(\log^2 n)$ **étapes, 1 bit par étape**

(Intuition: si tous les degrés sont égaux…)

L'aléa est-il indispensable?

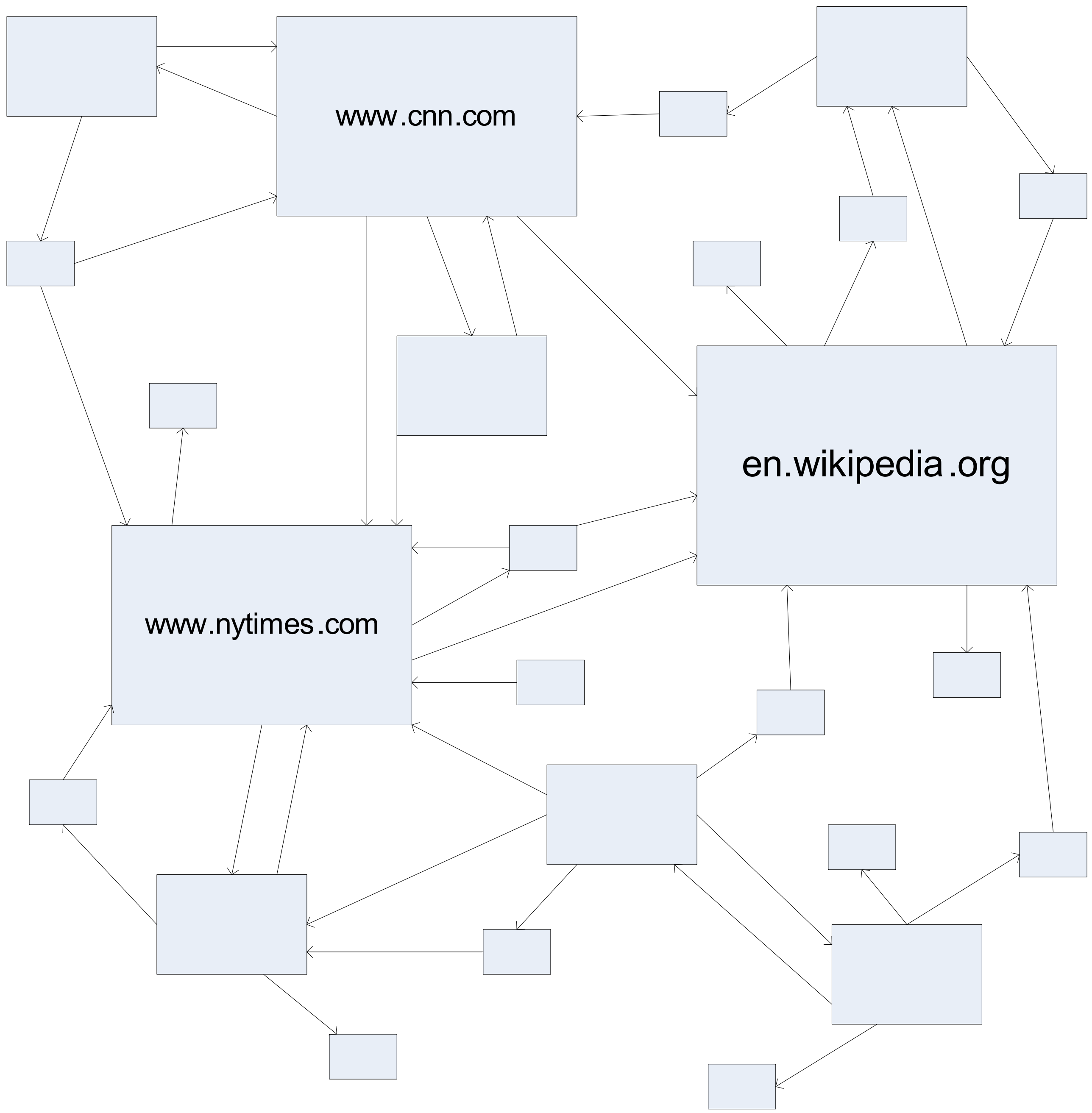


Sinon comment choisir ?
Blocage

Comment décider qui passe la porte en premier ?

Popularité des pages web

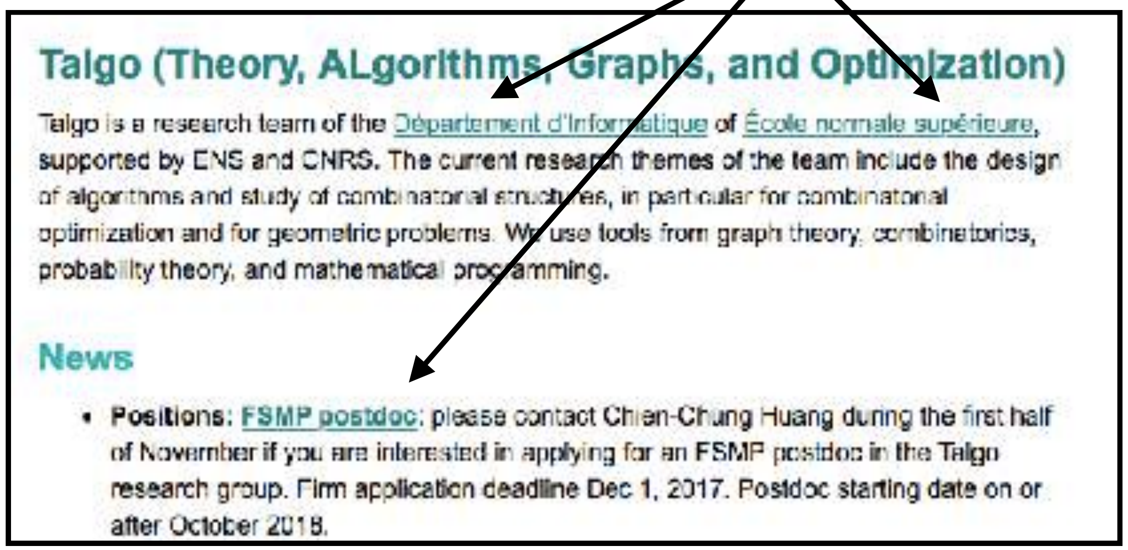
Popularités des pages web



L'aléa pour modéliser : classement des pages web

moteur de recherche : dans quel ordre montrer les résultats ?

page web avec des liens

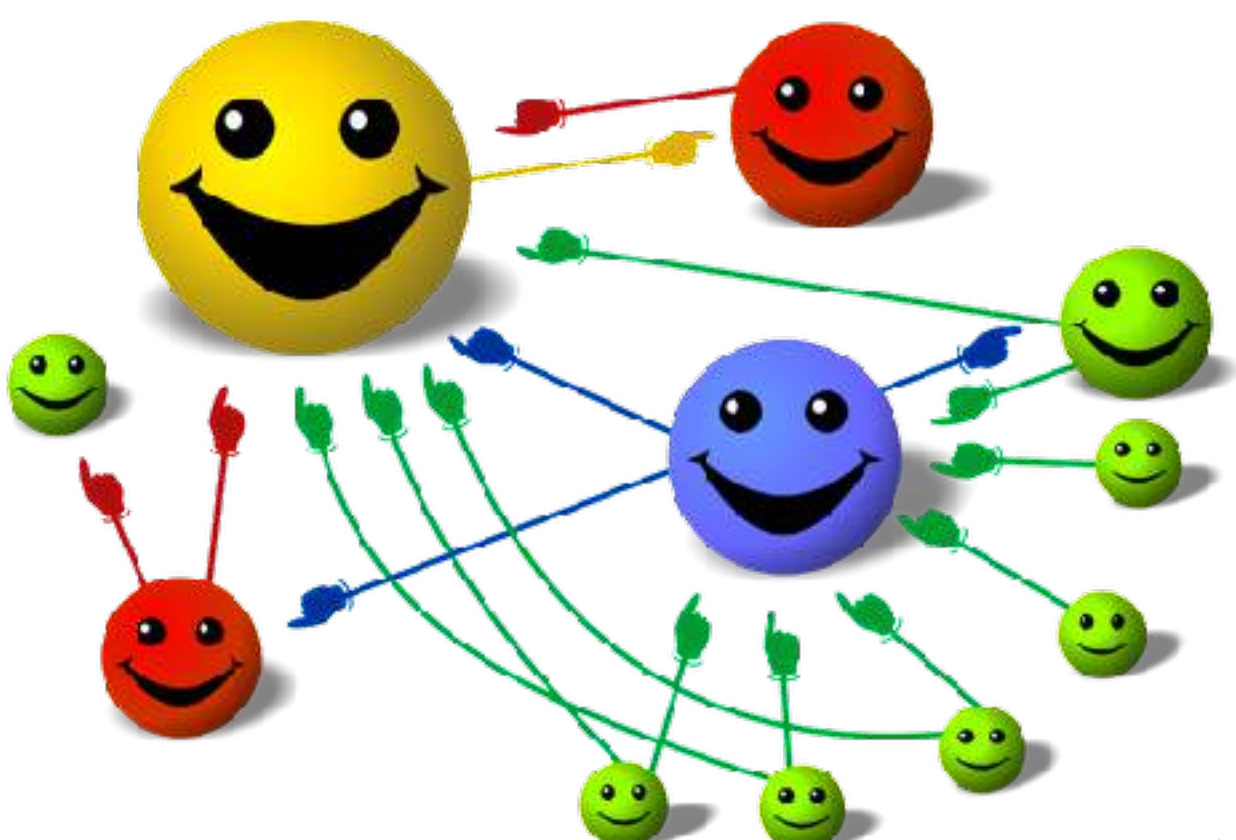


Talgo (Theory, ALgorithms, Graphs, and Optimization)

Talgo is a research team of the Département d'Informatique of École normale supérieure, supported by ENS and CNRS. The current research themes of the team include the design of algorithms and study of combinatorial structures, in particular for combinatorial optimization and for geometric problems. We use tools from graph theory, combinatorics, probability theory, and mathematical programming.

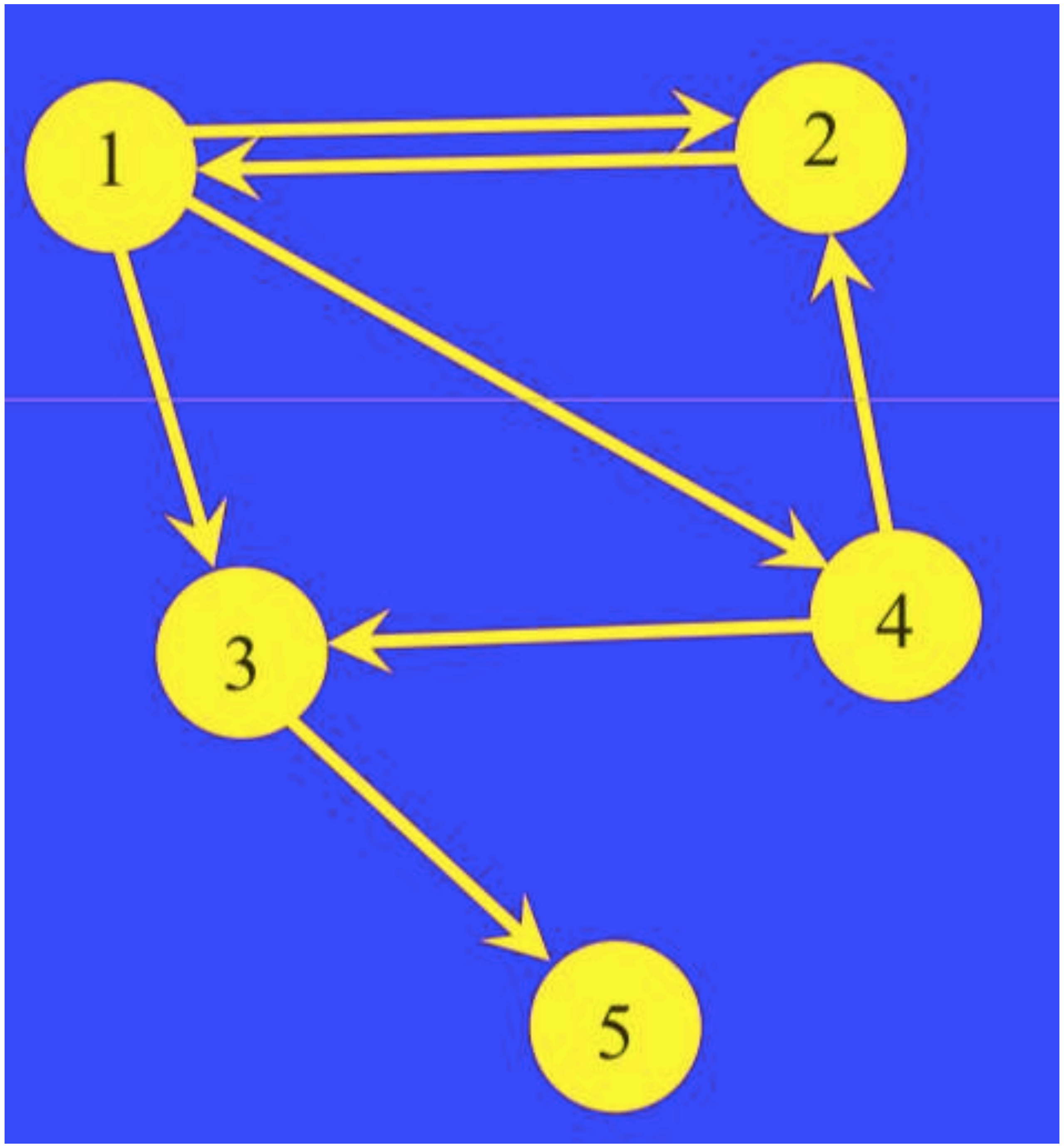
News

- Positions: FSMP postdoc: please contact Chien-Chung Huang during the first half of November if you are interested in applying for an FSMP postdoc in the Talgo research group. Firm application deadline Dec 1, 2017. Postdoc starting date on or after October 2018.



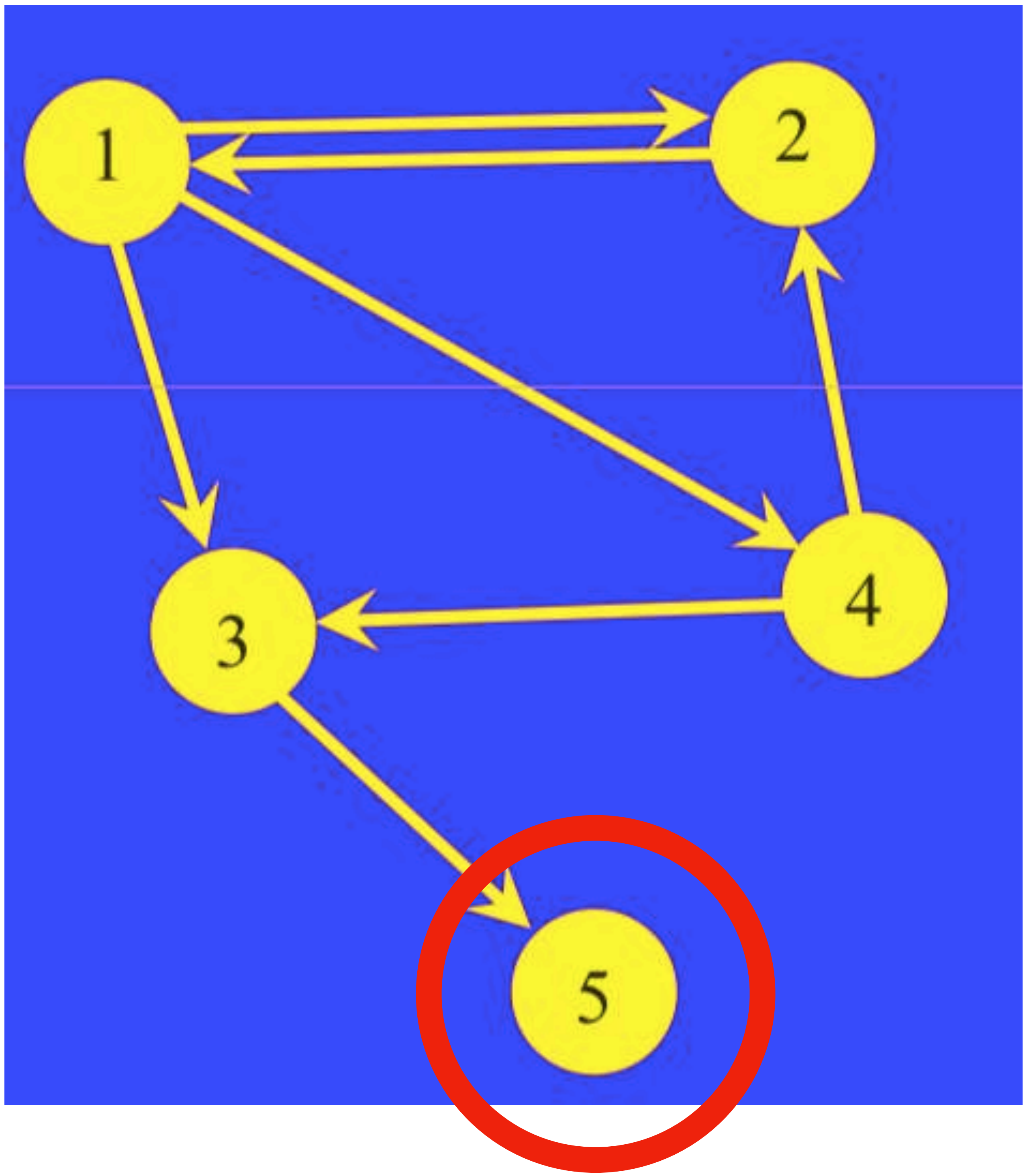
définition : une page est populaire si beaucoup d'autres pages ont un lien vers elle, surtout si ces pages sont elles-mêmes populaires…

l'explorateur simplifié du web : partant d'une page au hasard, il clique sur un lien au hasard, puis sur un lien au hasard de la nouvelle page, etc.



L’utilisateur fait une
marche aléatoire
sur le graphe du Web

Le temps passé
par l’utilisateur
sur une page
est une mesure
de sa popularité

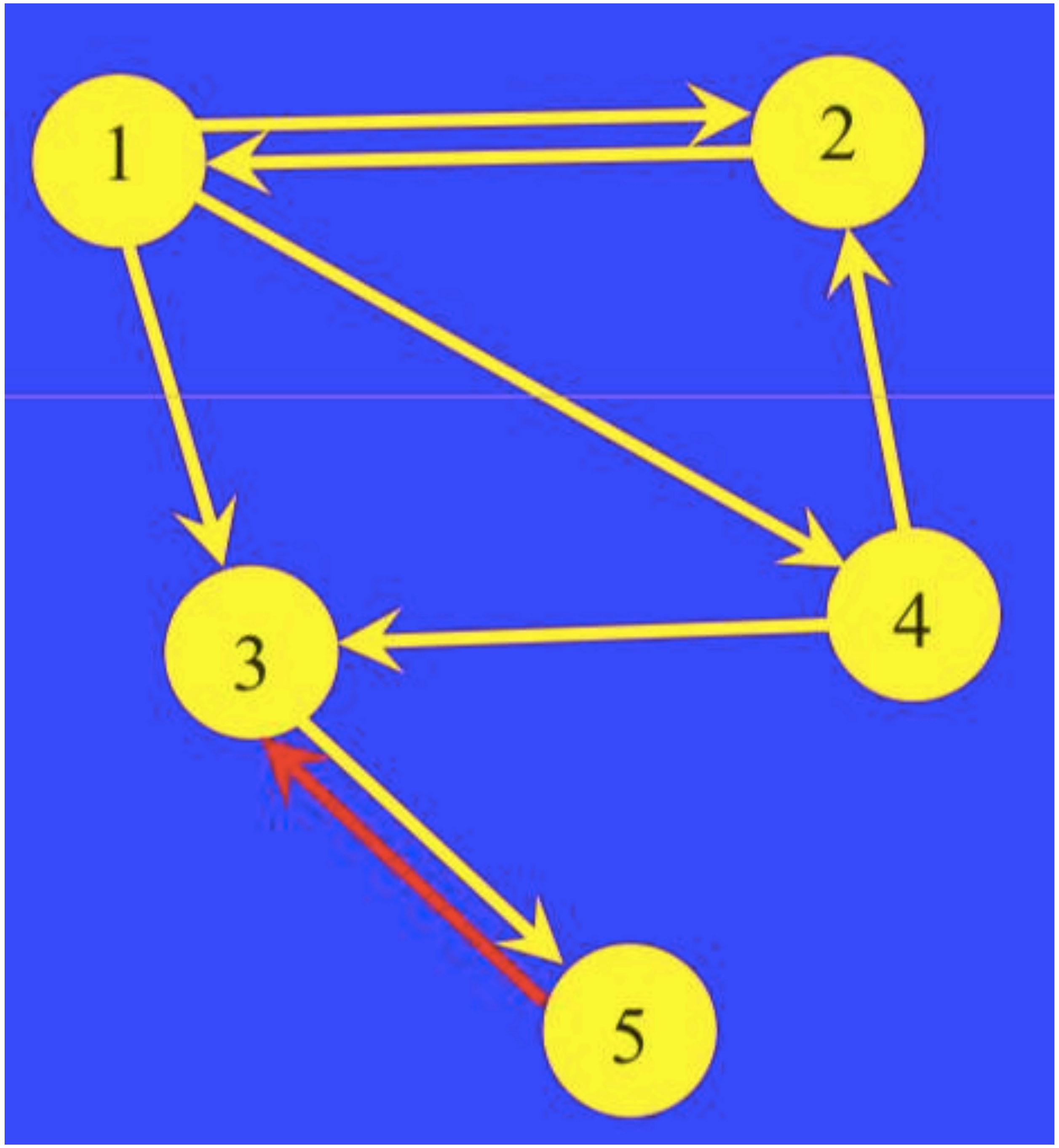


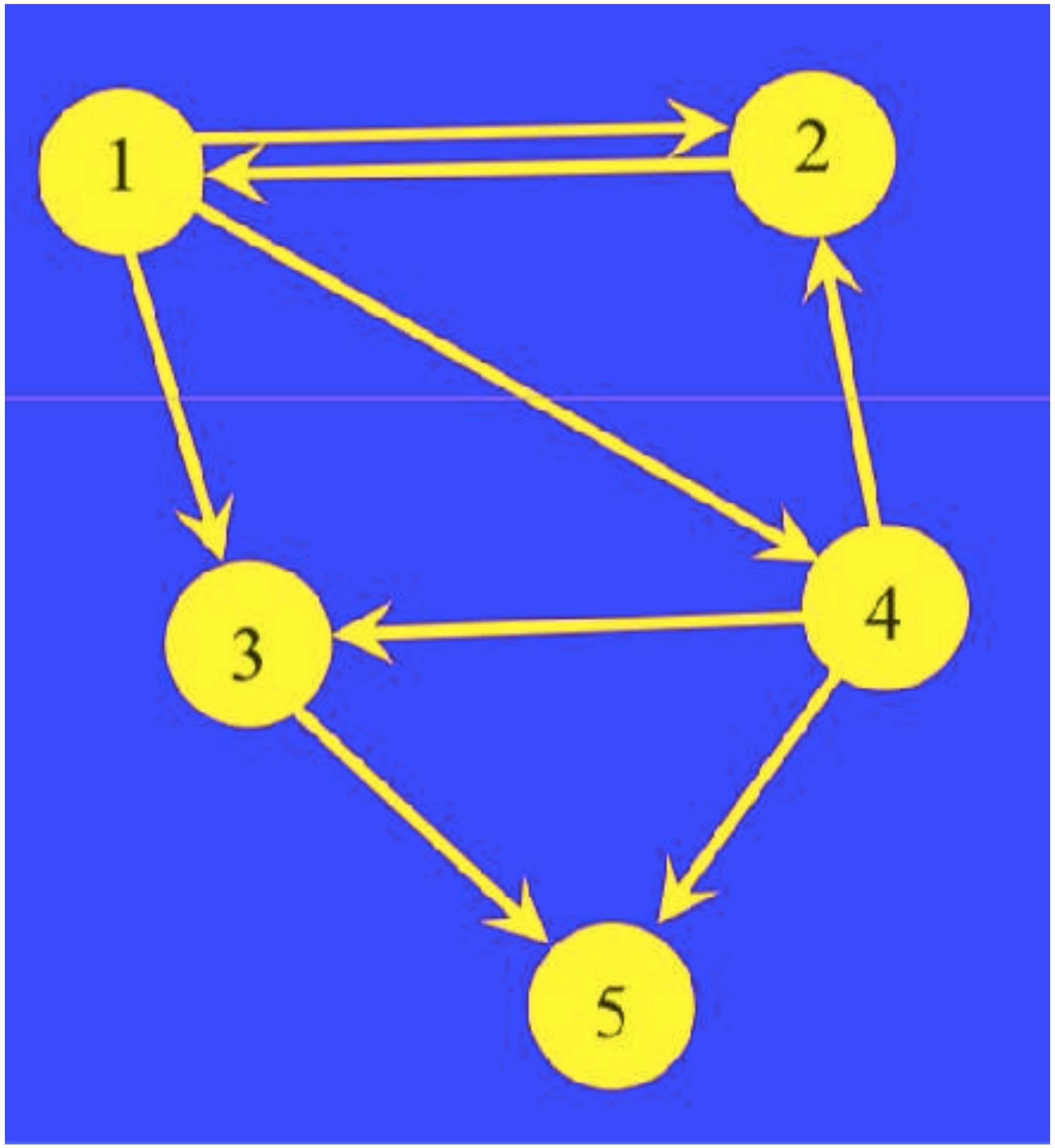
Cul-de-sac !

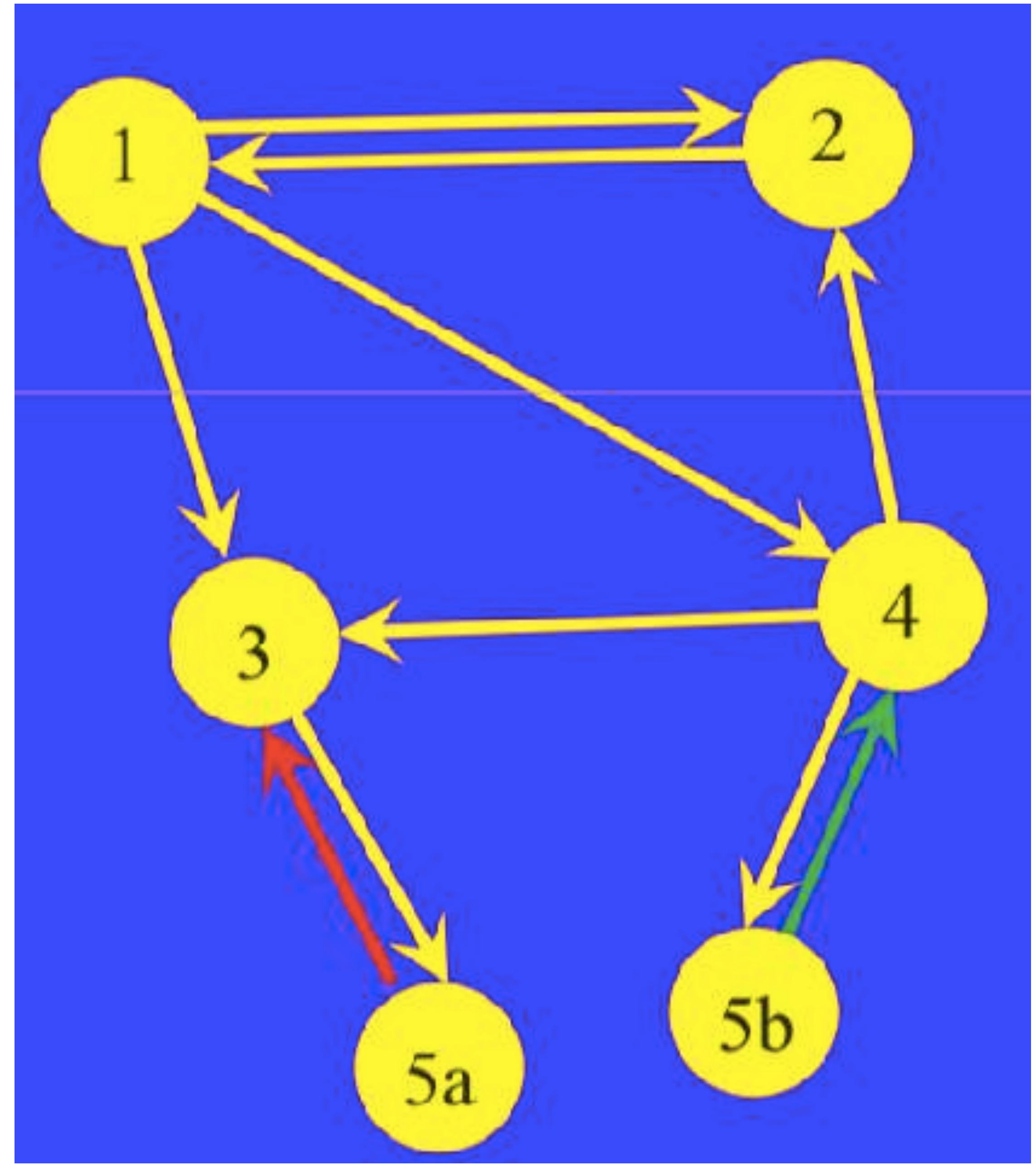
Il faut faire demi-tour

Cela arrive-t-il ?

Document en pdf sans liens

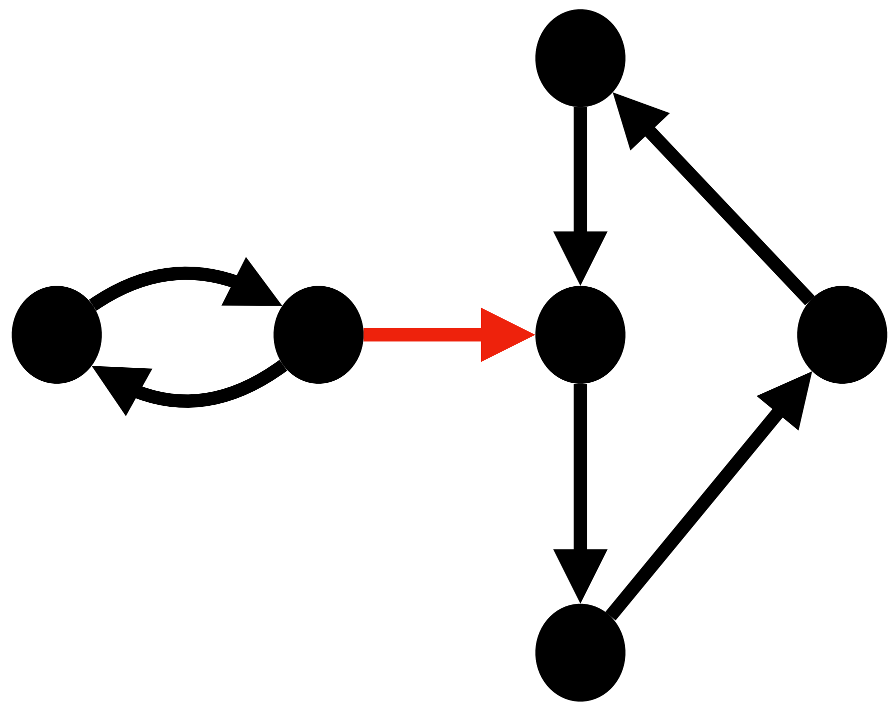

1
2
3
4
5





Double cul-de-sac !

Double demi-tour

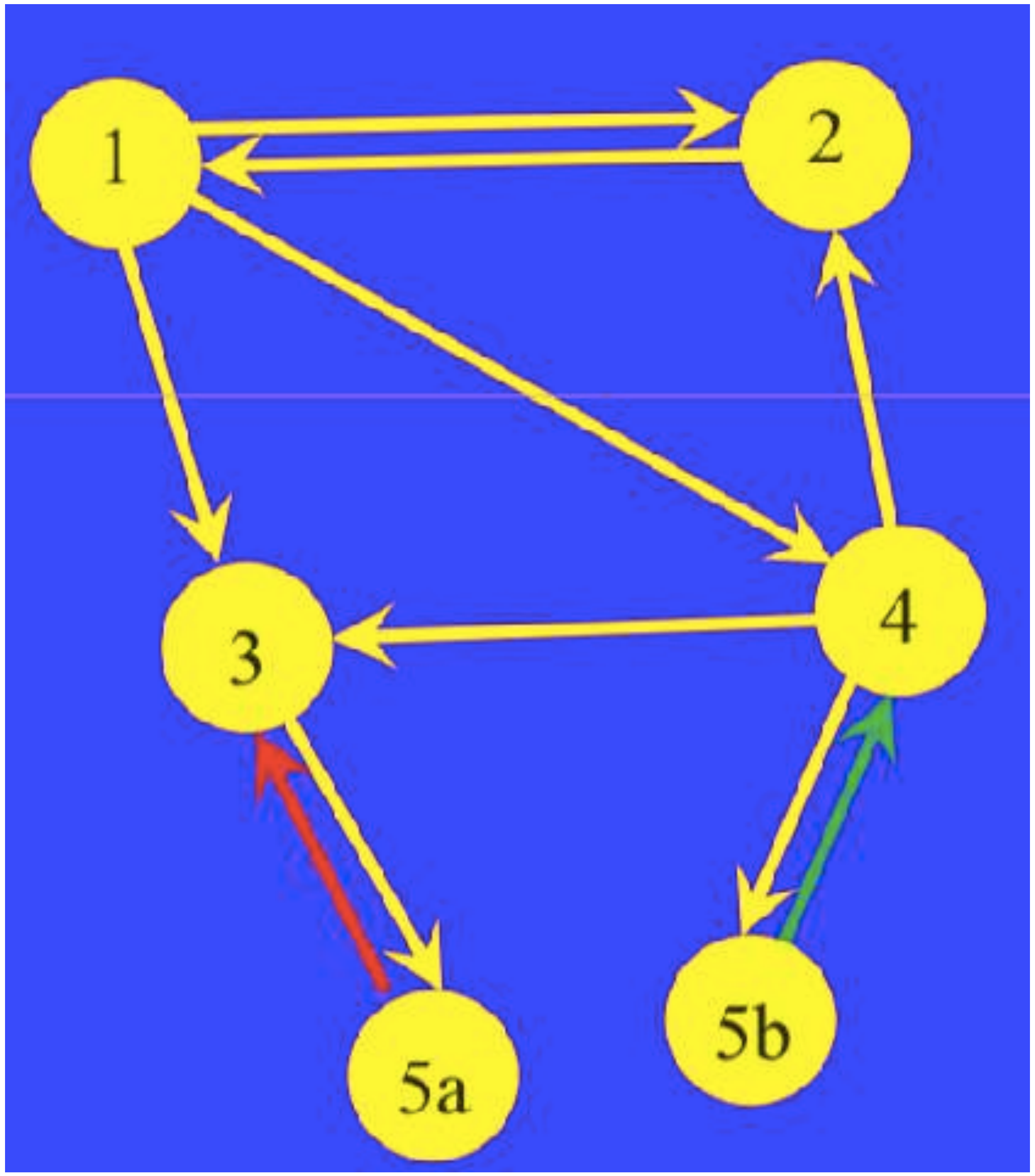


Sens unique sans retours possibles !

“ l’explorateur du web : partant d’une page au hasard, il clique sur un lien au hasard, puis sur un lien au hasard de la nouvelle page, etc., et de temps à autre redémarre d’une page prise au hasard…”

Avec probabilité 85%
suivre un lien au hasard
Avec probabilité 15%
faire un saut à une page au hasard

Popularité = fréquence de visite d’une page dans l’état stationnaire



85%

0	1/3	1/3	1/3	0	0
1	0	0	0	0	0
0	0	0	0	1	0
0	1/3	1/3	0	0	1/3
0	0	1	0	0	0
0	0	0	1	0	0

+

15%

1/6	1/6	1/6	1/6	1/6	1/6
1/6	1/6	1/6	1/6	1/6	1/6
1/6	1/6	1/6	1/6	1/6	1/6
1/6	1/6	1/6	1/6	1/6	1/6
1/6	1/6	1/6	1/6	1/6	1/6
1/6	1/6	1/6	1/6	1/6	1/6

état stationnaire : vecteur propre pour valeur propre maximum (1)

Algorithme pour trouver la distribution stationnaire :

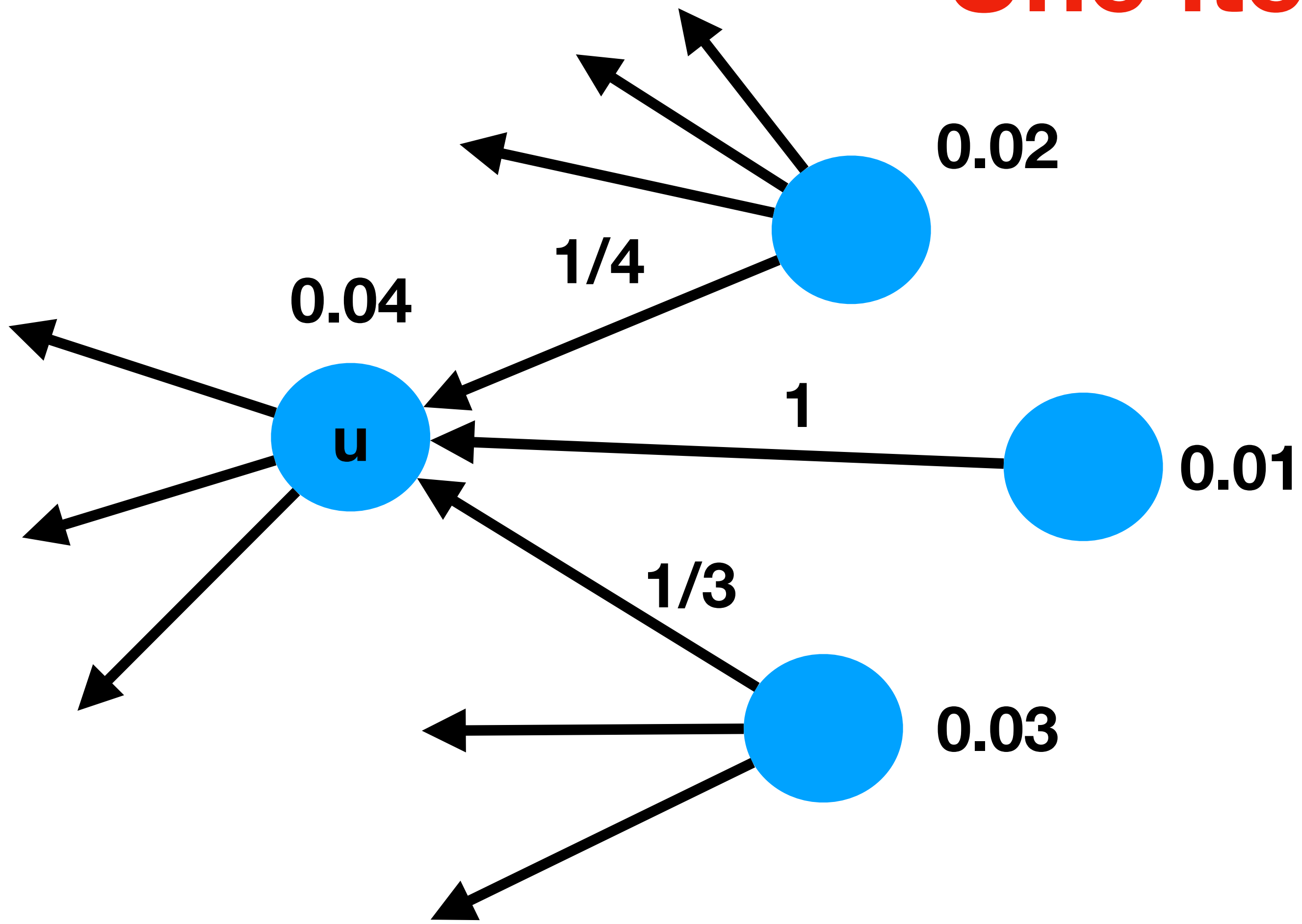
x0 : distribution arbitraire
Appliquer la matrice suffisamment de fois
x1,x2,x3,…,xt
pour que xt soit proche de l'état stationnaire

(Convergence rapide grâce au saut aléatoire)

Taille du graphe du web : dizaines de milliards de pages
Nombre d'itérations : 50 à 100
Centralisé : 1 semaine de travail

Comment aller plus vite ?

Une itération



Nouvelle popularité de u :

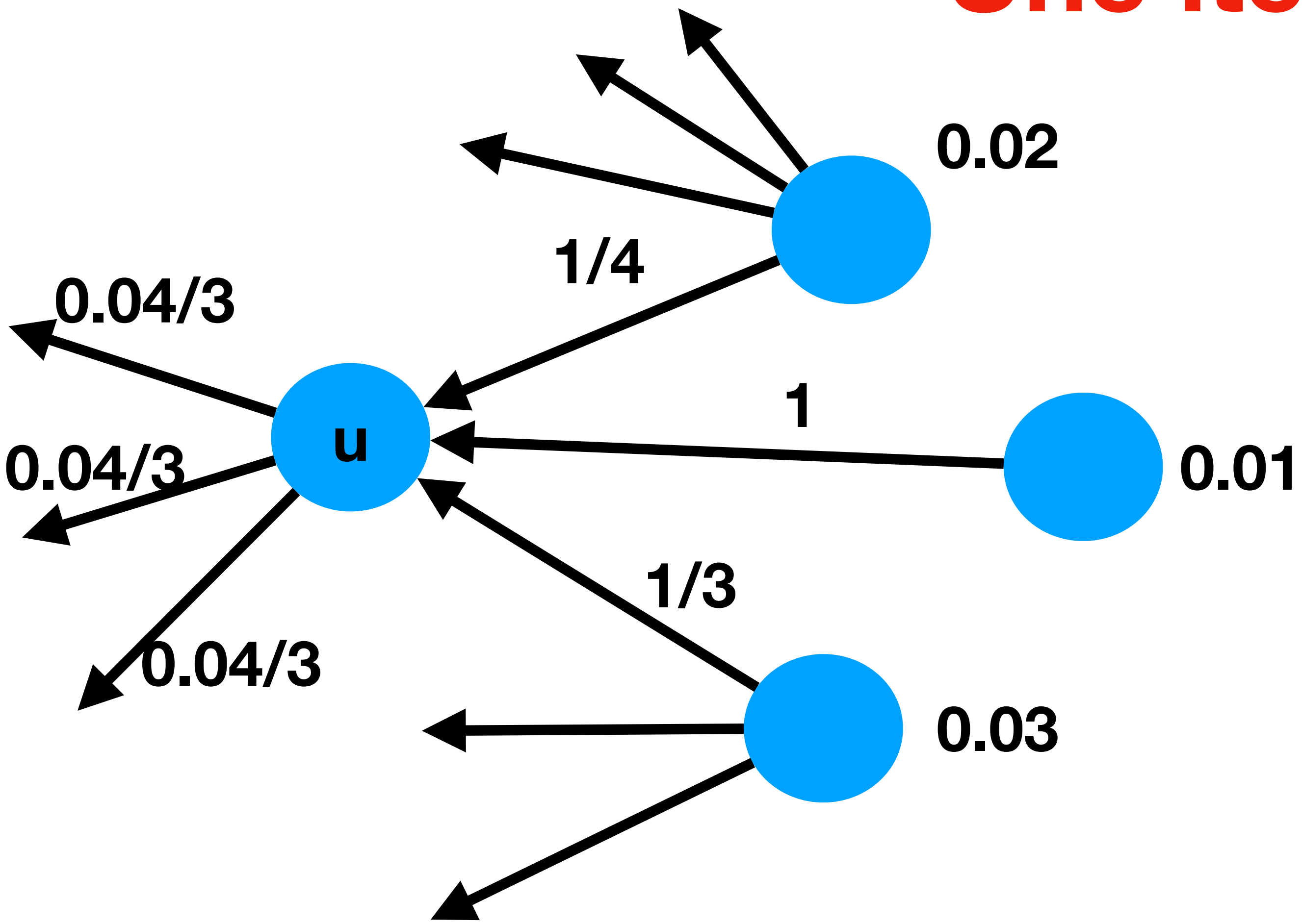
$$0.15*(1/N)$$
$$+$$
$$0.85*(0.02*(1/4)+0.01*1+0.03*(1/3))$$

C'est un calcul local
qui ne dépend que des voisins de u

Utiliser les prédécesseurs de u
pour mettre à jour sa popularité

On peut donc envisager de le faire de façon distribuée !

Une itération



Nouvelle popularité de u :

0.15*(1/N)
+
0.85*(0.02*(1/4)+0.01*1+0.03*(1/3))

Distribuer la popularité de u
aux successeurs de u

La dynamique est un peu différente mais
la distribution stationnaire est la même

En distribué : des agents dispersés sur le Web travaillent
indépendamment les uns des autres
de façon décentralisée
K agents : calcul plus rapide d'un facteur K

Conclusion

L'aléa en algorithmique distribuée permet de

- **sortir de situations de blocage autrement impossibles**
- **accélérer les calculs**