université PARIS DIDEROT PARIS 7

Algorithmes distribués

Pierre Fraigniaud

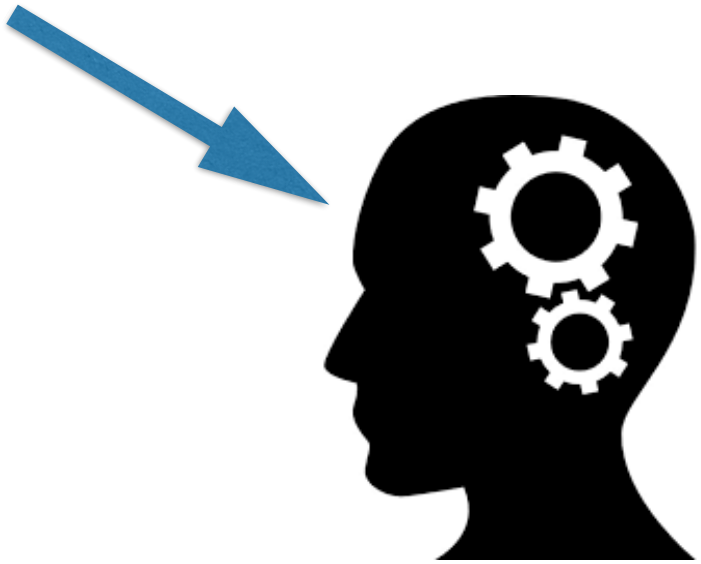
INSTITUT DE RECHERCHE EN INFORMATIQUE FONDAMENTALE

Collège de France, 28 novembre 2017

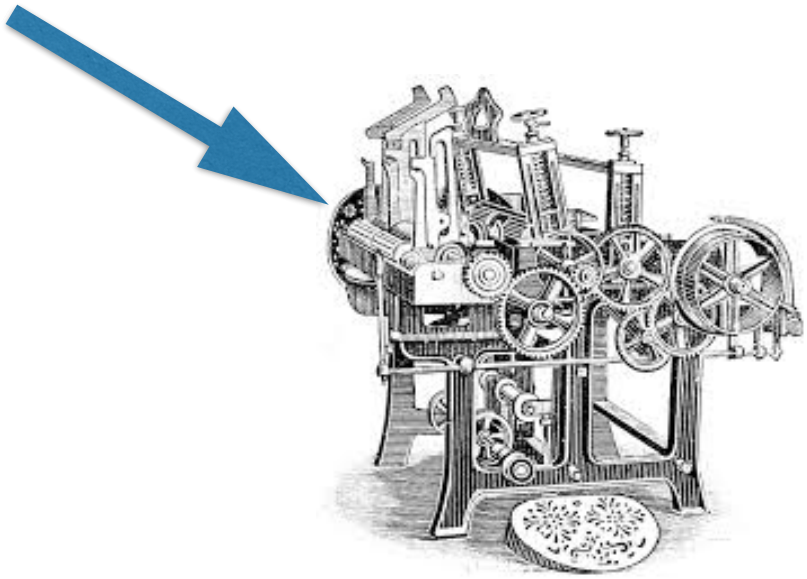
Algorithmes / programmes

Procédures « mécaniques »
pour résoudre un problème donné

algorithme



programme



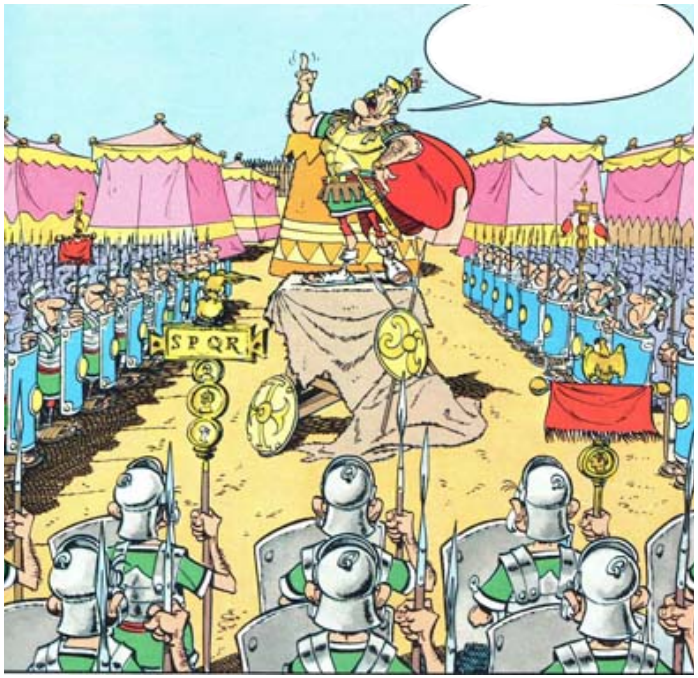
Algorithmes distribués

Ensemble d'entités de calcul concourant
à la résolution d'un problème

Parallèle vs. Distribué

Parallèle vs. Distribué

Calcul parallèle

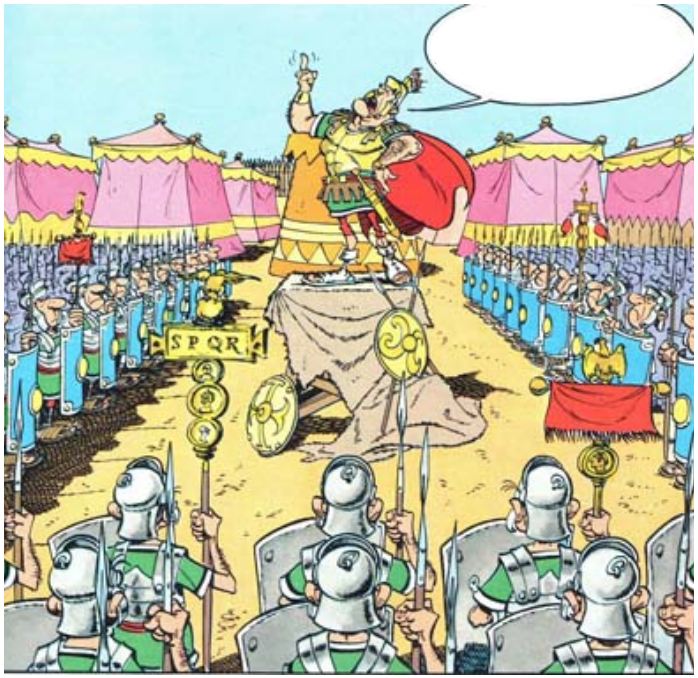


Performances de calcul

> pétaFLOPS (10^{15} op./s)

Parallèle vs. Distribué

Calcul parallèle



Calcul distribué

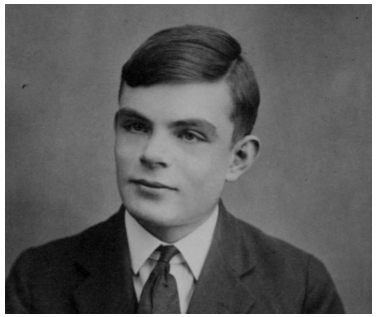
Performances de calcul

> pétaFLOPS (10^{15} op./s)

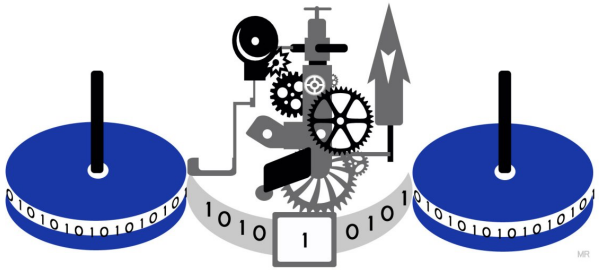
Gestion de l'incertitude

temporelle et/ou spatiale

Du séquentiel au distribué

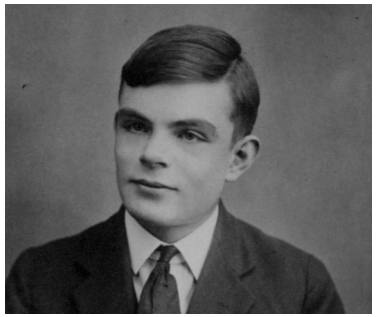


Alan Turing

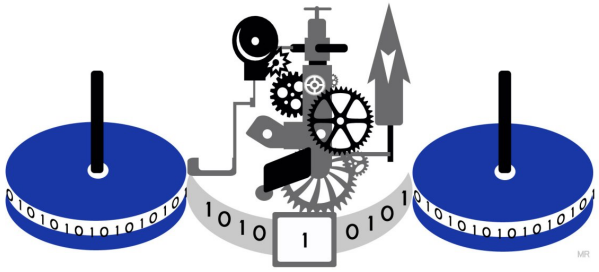


Alonzo Church

Du séquentiel au distribué

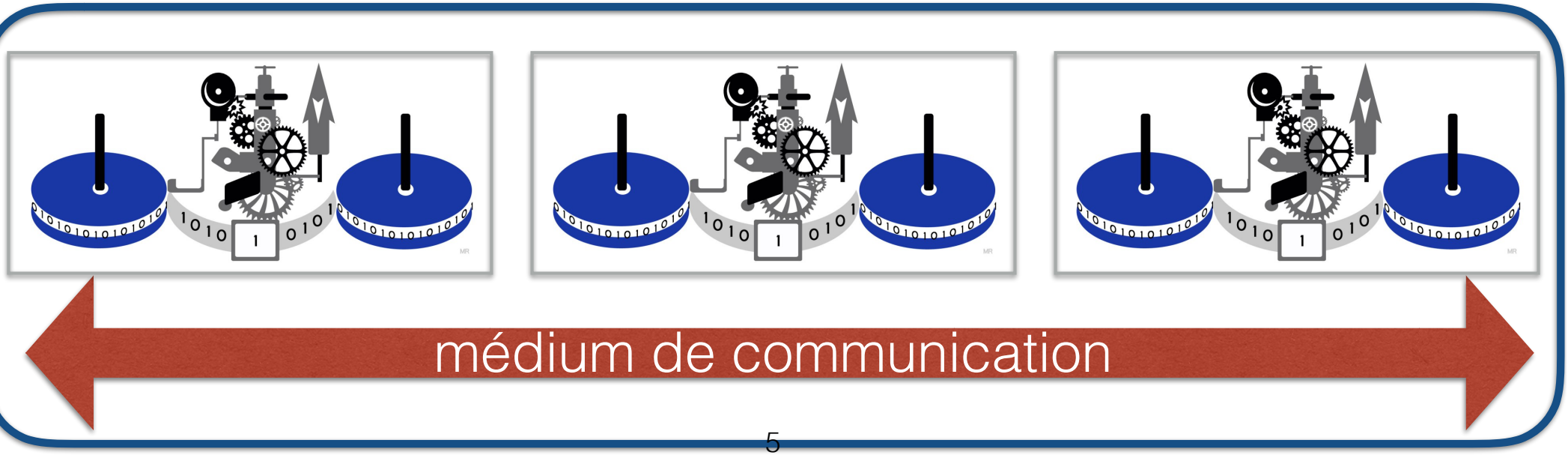


Alan Turing



Alonzo Church

Modèle « classique » pour le calcul distribué



Limites du calcul séquentiel : Indécidabilité 

- Mortalité des matrices :

Exemple : $A = \begin{pmatrix} 0 & 1 & -1 \\ 0 & 1 & -1 \\ 1 & 0 & 1 \end{pmatrix}$ $B = \begin{pmatrix} -1 & 0 & 1 \\ -1 & -1 & 0 \\ -1 & 1 & 1 \end{pmatrix}$

$$AB^2A^3B^4A^2BAB^2A = 0$$

<u>Données :</u> six matrices 3×3
<u>Question :</u> peut-on multiplier ces matrices entre elles pour obtenir la matrice nulle ?

Limites du calcul séquentiel : Indécidabilité 

- Mortalité des matrices :

Exemple : $A = \begin{pmatrix} 0 & 1 & -1 \\ 0 & 1 & -1 \\ 1 & 0 & 1 \end{pmatrix}$ $B = \begin{pmatrix} -1 & 0 & 1 \\ -1 & -1 & 0 \\ -1 & 1 & 1 \end{pmatrix}$

$$AB^2A^3B^4A^2BAB^2A = 0$$

voie sans algorithme

<u>Données :</u> six matrices 3×3

<u>Question :</u> peut-on multiplier ces matrices entre elles pour obtenir la matrice nulle ?

Limites du calcul séquentiel : Indécidabilité 

- Mortalité des matrices :

Exemple : $A = \begin{pmatrix} 0 & 1 & -1 \\ 0 & 1 & -1 \\ 1 & 0 & 1 \end{pmatrix}$ $B = \begin{pmatrix} -1 & 0 & 1 \\ -1 & -1 & 0 \\ -1 & 1 & 1 \end{pmatrix}$

$AB^2A^3B^4A^2BAB^2A = 0$

voie sans algorithme

<u>Données :</u> six matrices 3×3
<u>Question :</u> peut-on multiplier ces matrices entre elles pour obtenir la matrice nulle ?

- Vérification de programmes

Limites du calcul distribué : Indécidabilité  + incertitude 

Incertitudes :

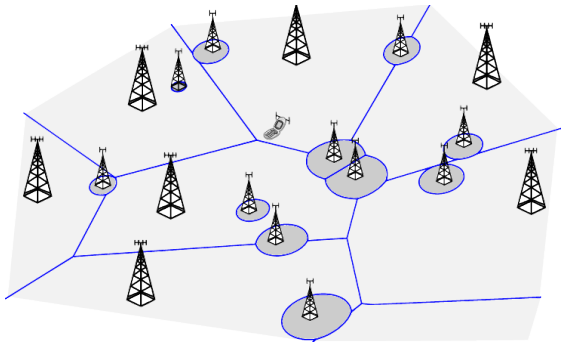
- spatiales (réseau de communication)
- temporelles (horloges asynchrones, charge, etc.)
- pannes (transitoires, crash, malicieuses, etc.)
- comportements égoïstes (théorie des jeux)
- …

Plusieurs MT sont moins puissantes qu'une seule !

Briser la symétrie

- Election d'un leader
- Consensus
- Coloration
- etc.

Applications :



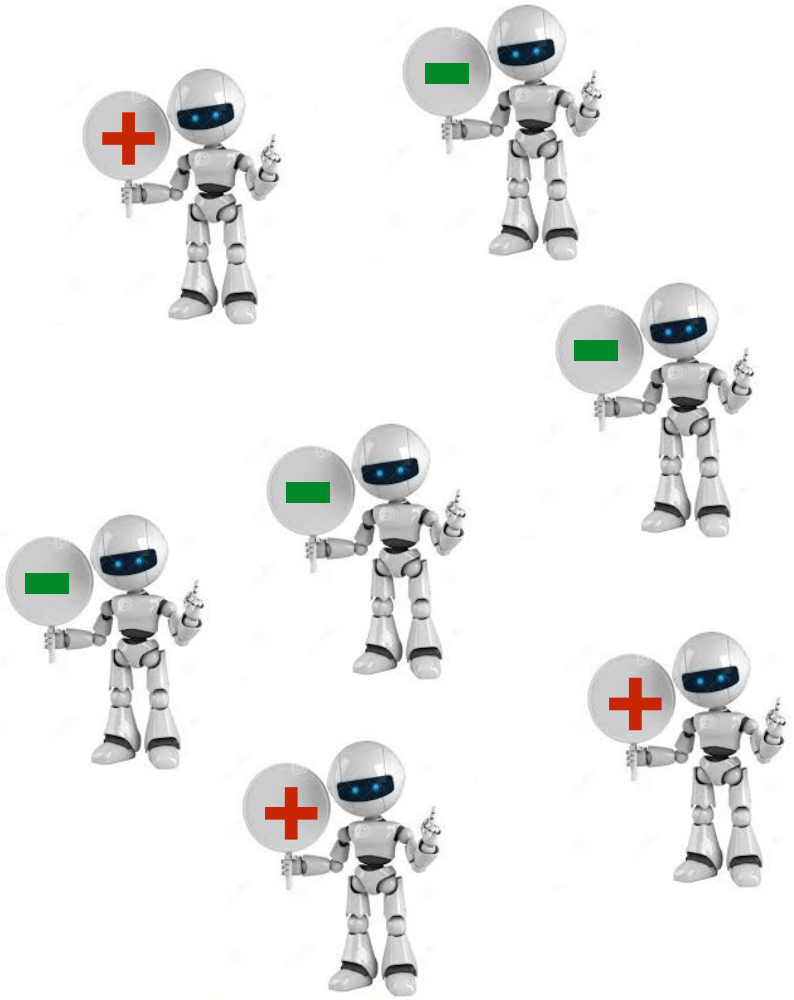
Affectation de fréquences

Consistence des bases de données

Incertitudes temporelles

Gestion de l’asynchronisme

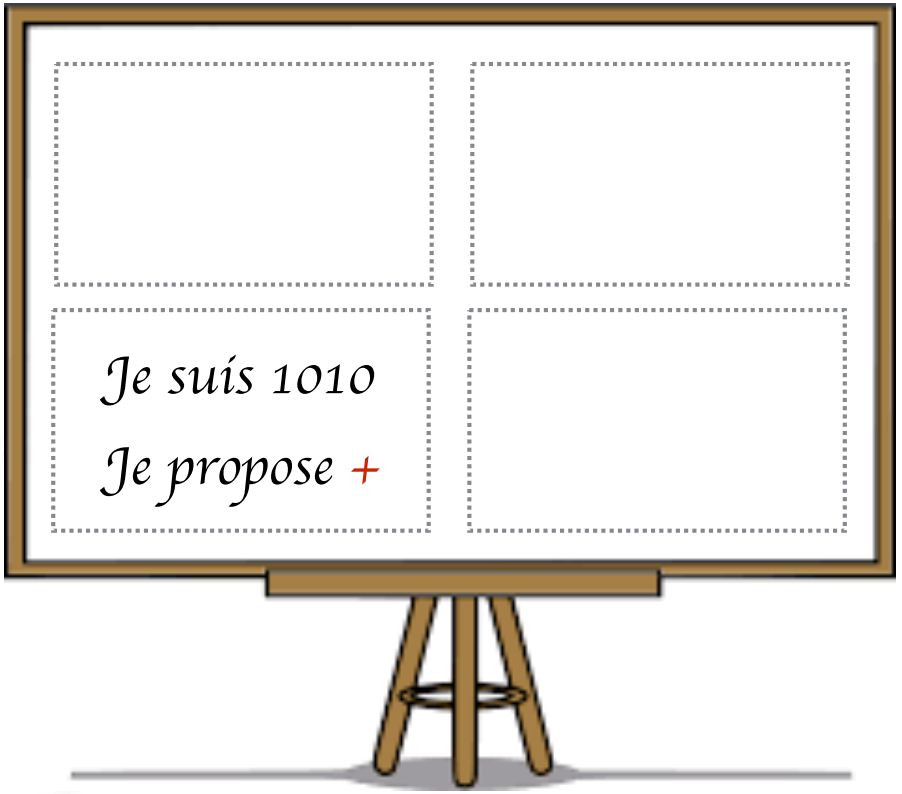
Consensus

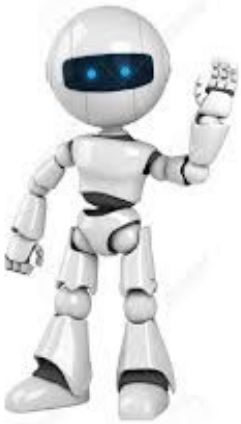


- **Terminaison :** tout robot decide une valeur proposée + ou — ;
- **Accord :** toutes les valeurs décidées sont identiques.
- **Validité :** si tous les robots proposent la même valeur, alors ils doivent décider cette valeur ;

Modèle

Mémoire partagée





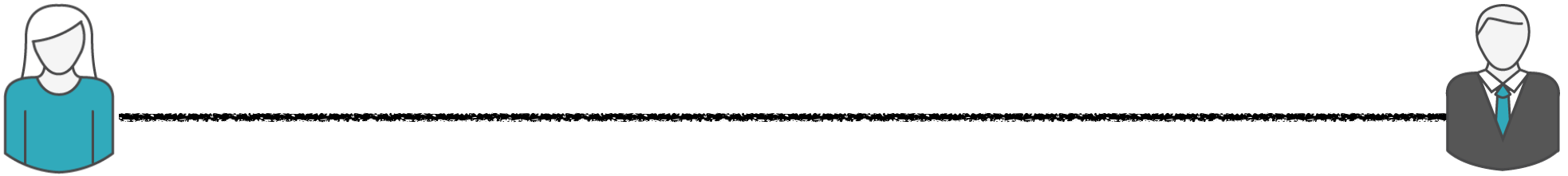
Impossibilité du consensus

M. Fischer, N. Lynch, M. Paterson (1985)

Théorème Le consensus binaire (+/–) est impossible dans un modèle asynchrone, même en présence d'au plus un crash.

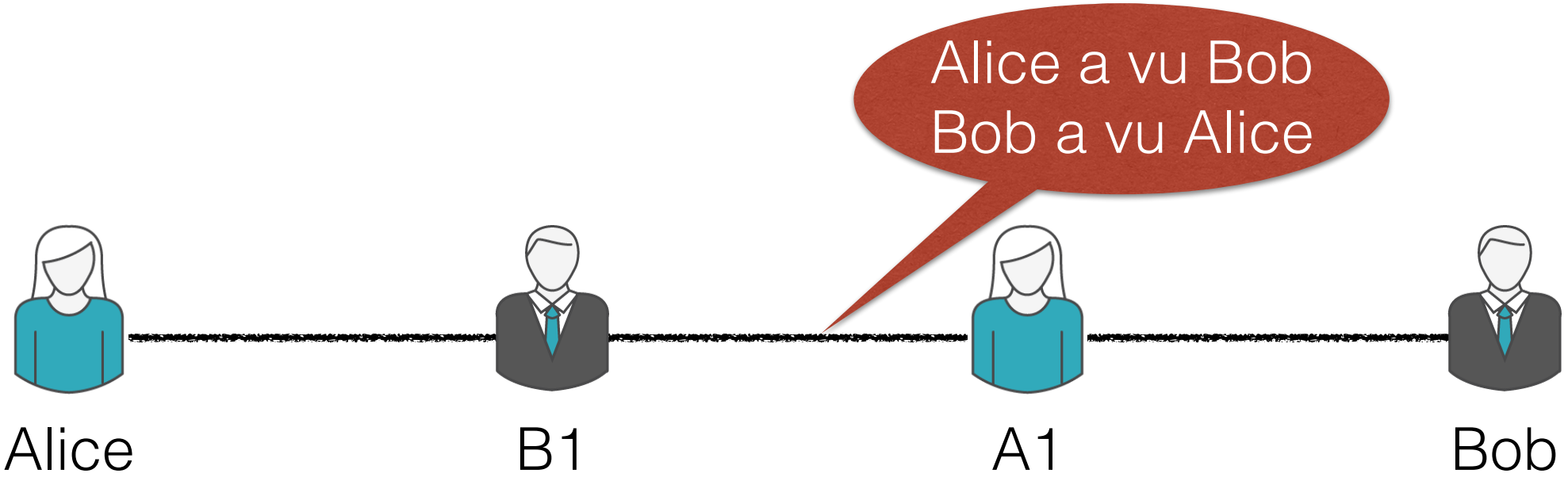
Dijkstra Prize 2001

Modélisation topologique du calcul distribué asynchrone

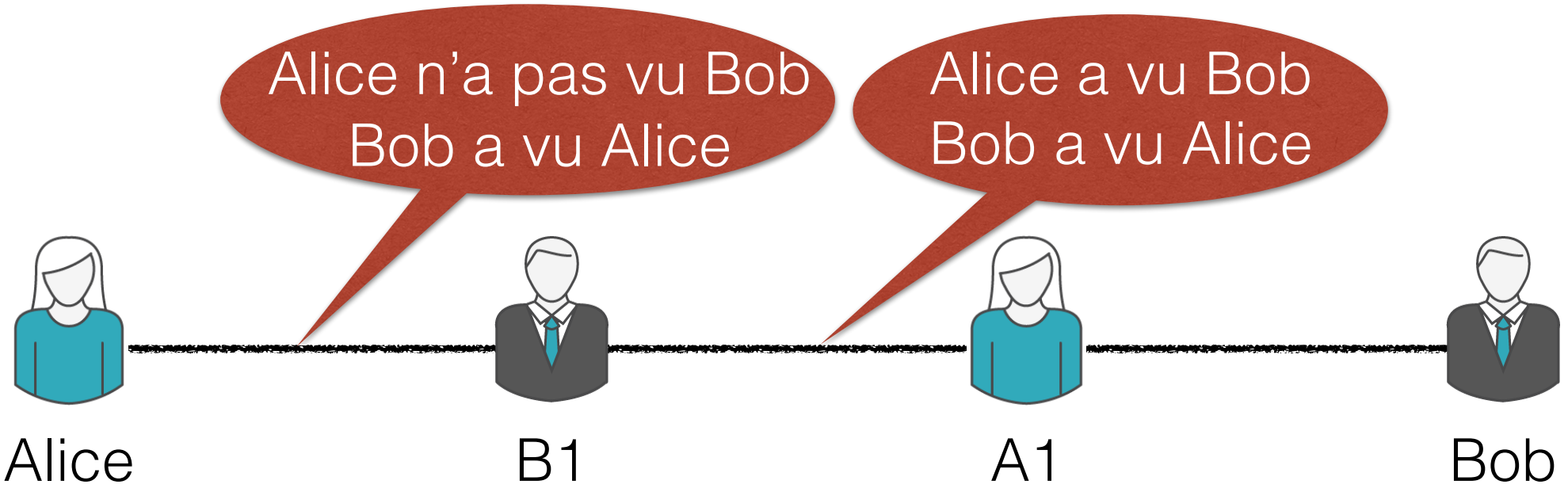


Modélisation topologique du calcul distribué asynchrone

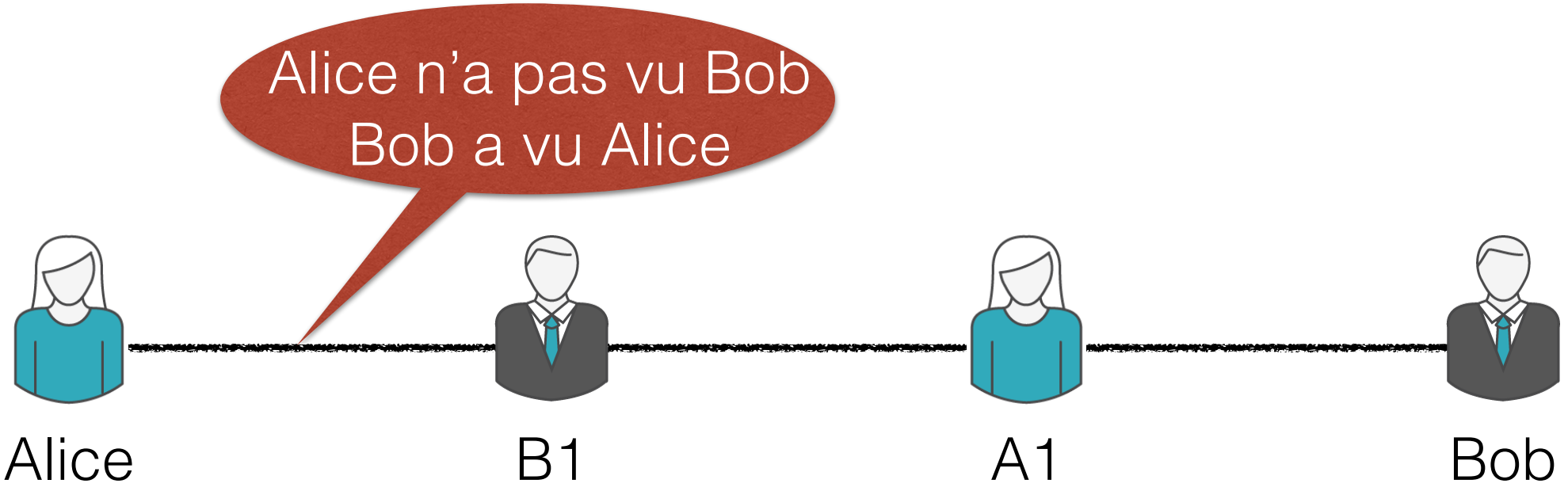
Modélisation topologique du calcul distribué asynchrone



Modélisation topologique du calcul distribué asynchrone

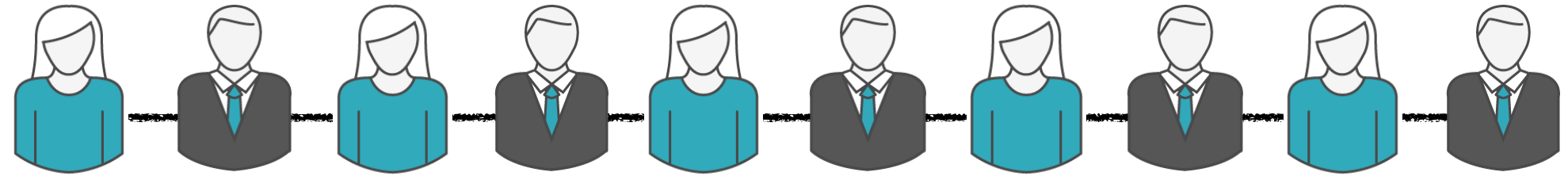


Modélisation topologique du calcul distribué asynchrone

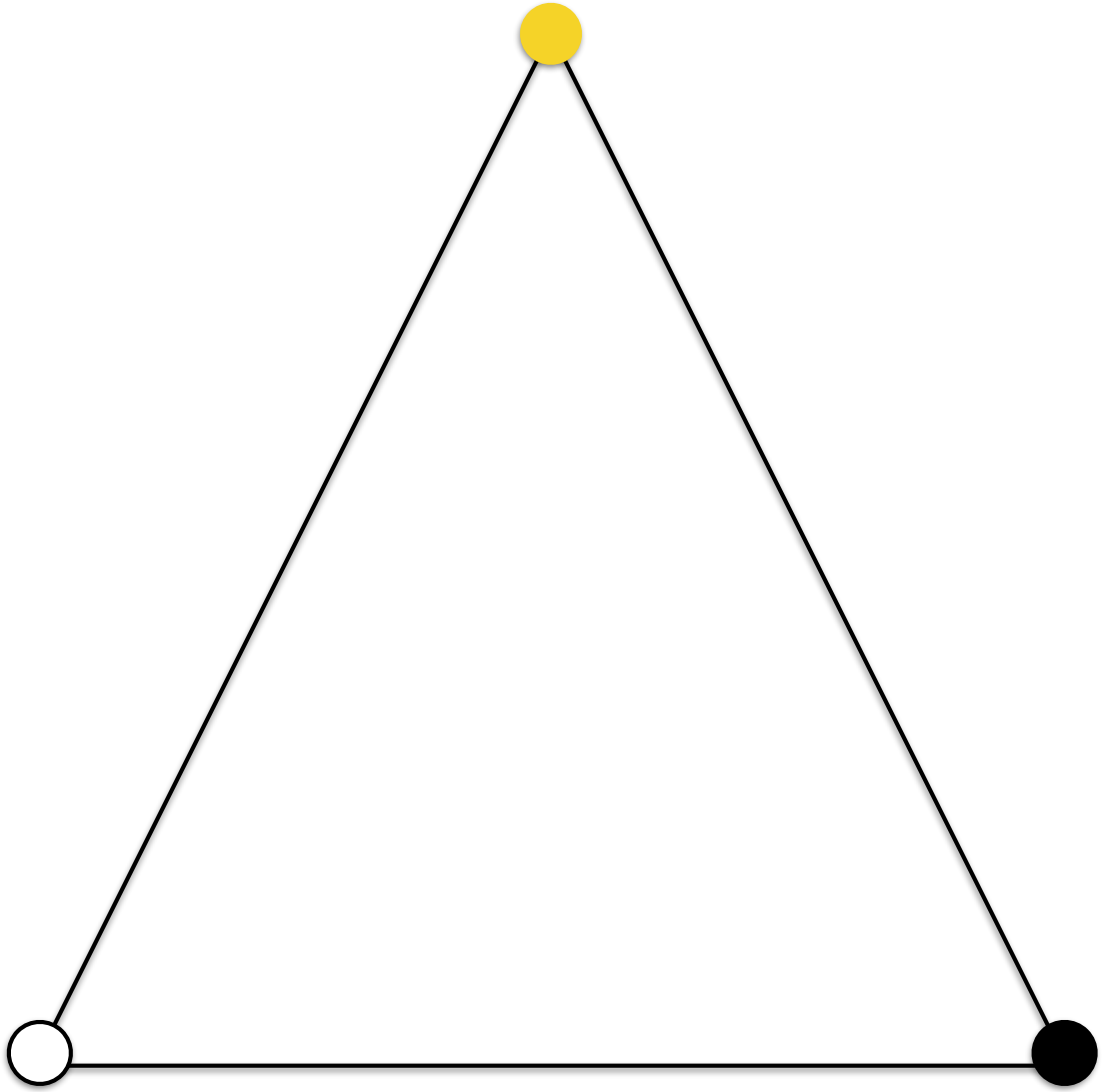


Modélisation topologique du calcul distribué asynchrone

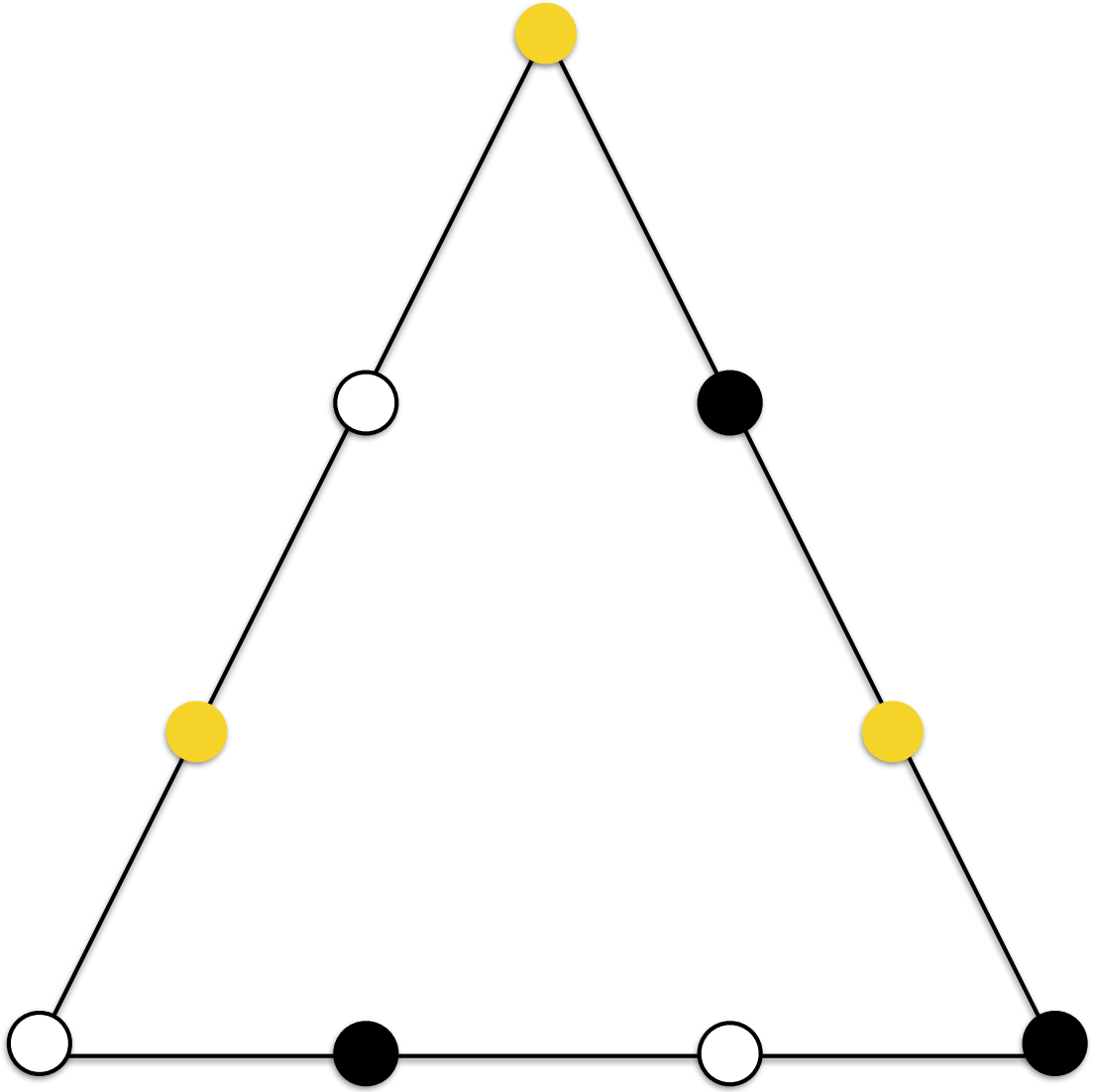
Modélisation topologique du calcul distribué asynchrone



Exemple : 3 processus

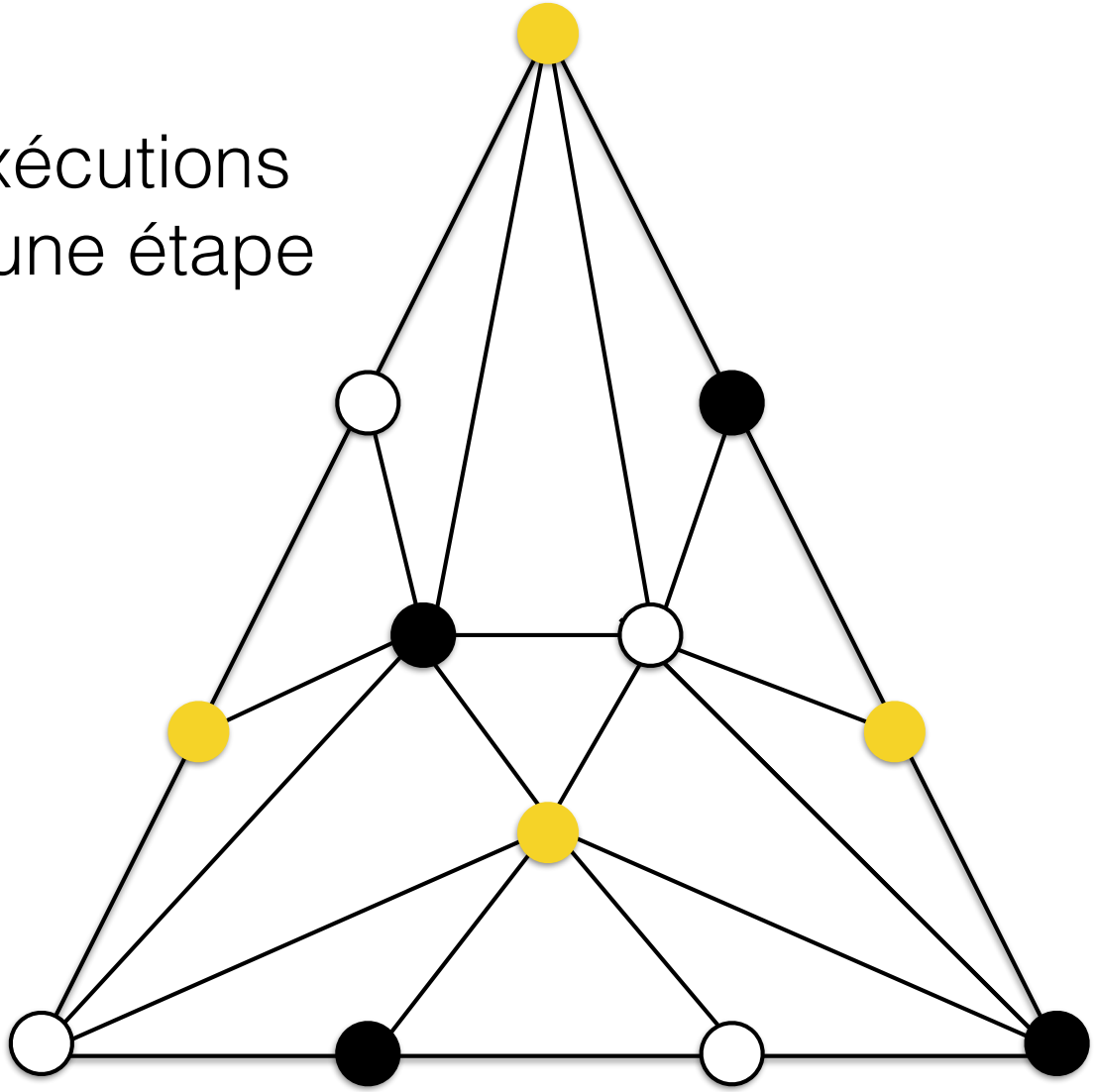


Exemple : 3 processus



Exemple : 3 processus

Toutes les exécutions possibles d'une étape



Résolution distribuée

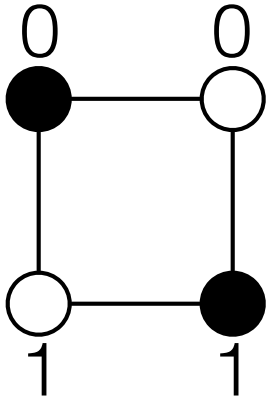
M. Herlihy and N. Shavit (1999)

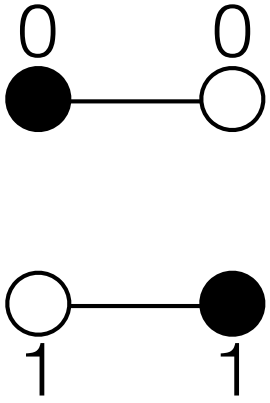
Théorème Un problème peut être résolu dans un modèle asynchrone avec crashs si et seulement si il existe une application simpliciale d'une subdivision du complexe d'entrée vers le complexe de sortie respectant les spécifications du problème.

Gödel Prize 2004

Application au consensus

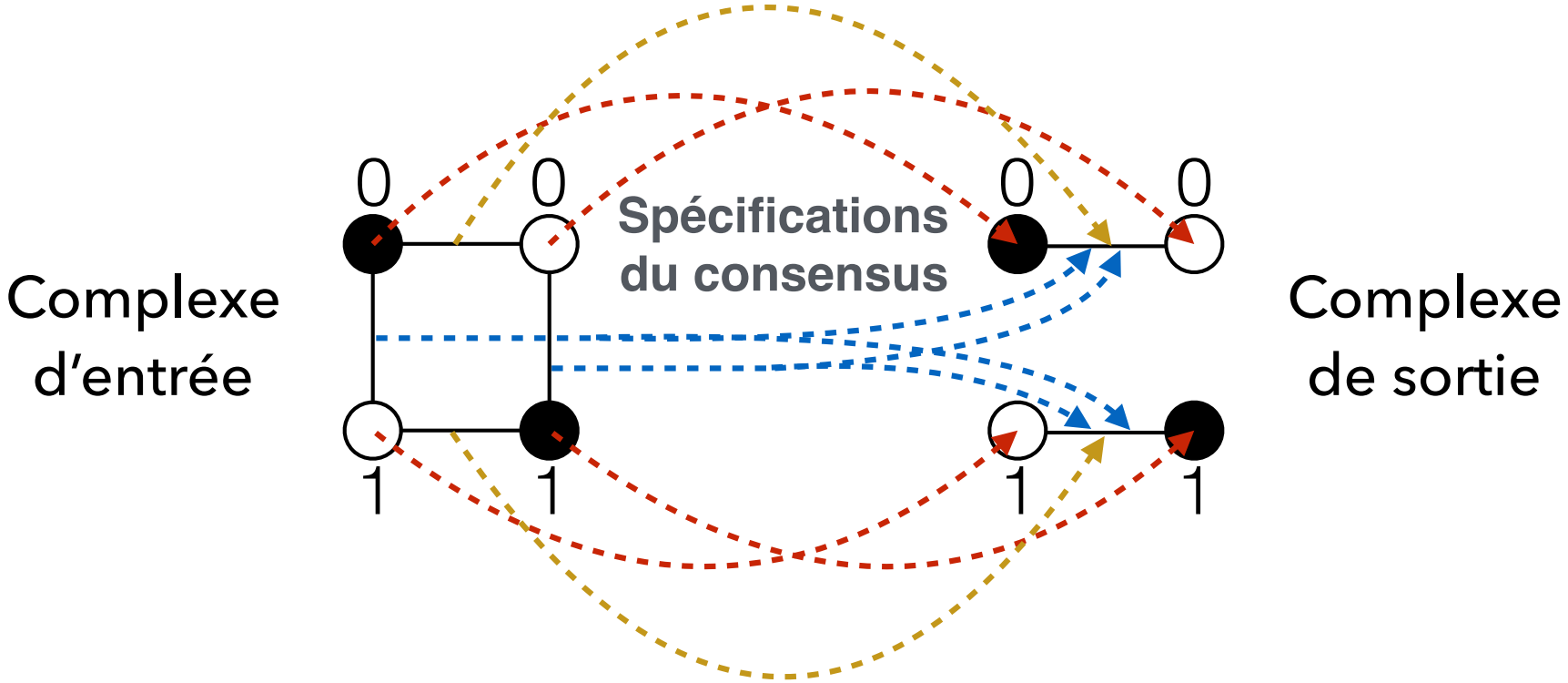
Complexe d'entrée



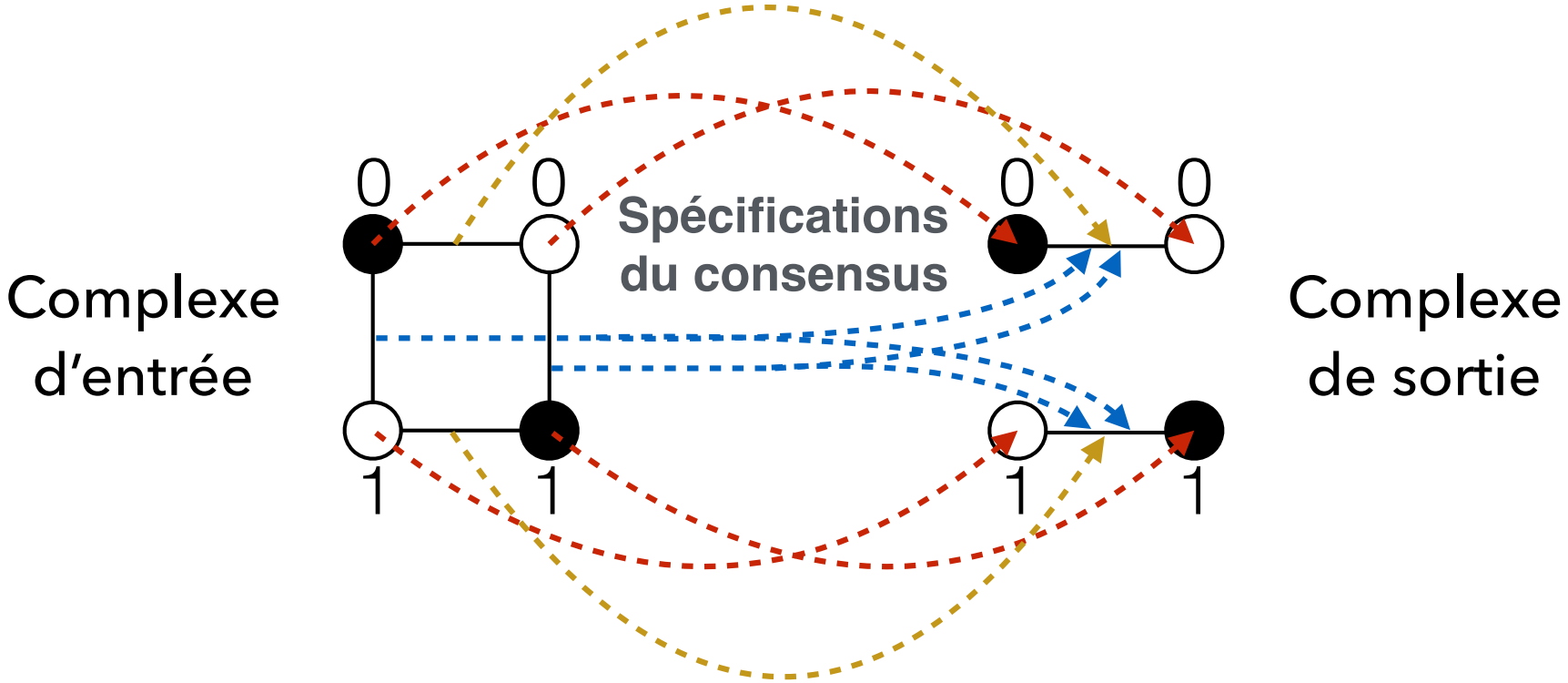


Complexe de sortie

Application au consensus



Application au consensus



Pas d'application simpliciale d'aucune subdivision du complexe d'entrée vers le complexe de sortie respectant les spécifications du consensus.

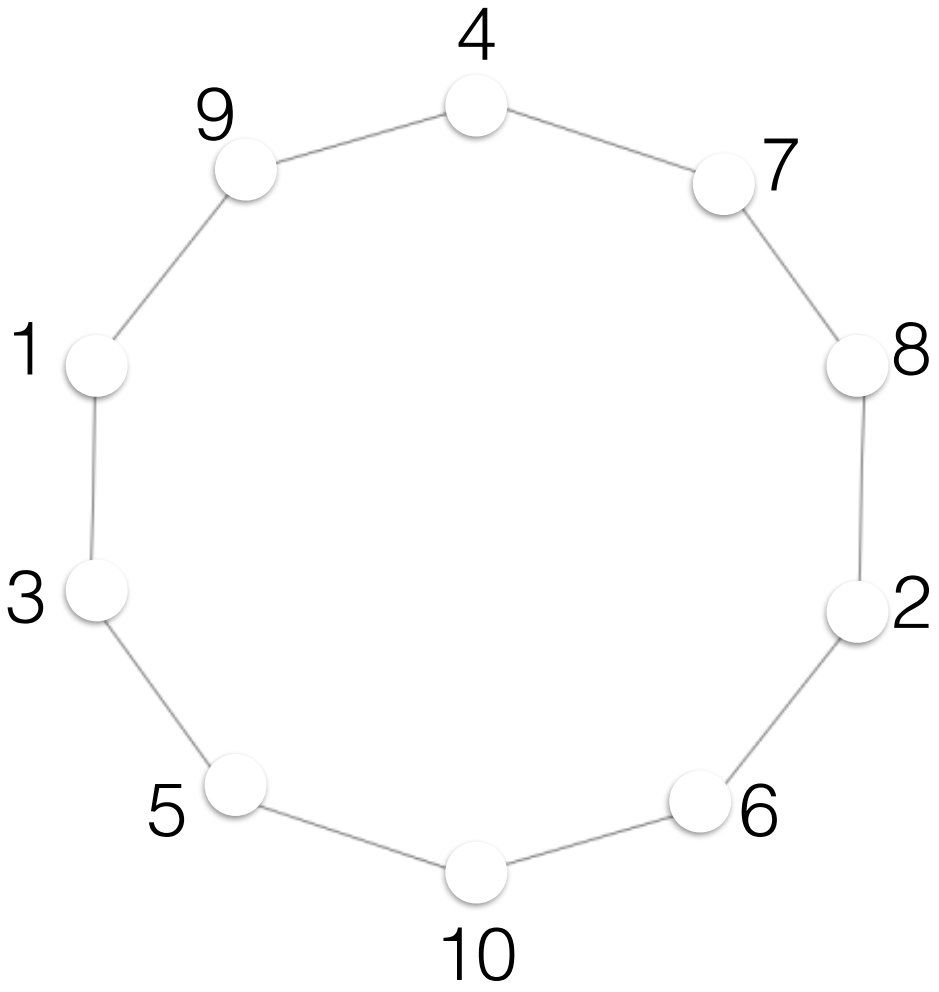
En pratique

- Consensus est la brique de base de nombreux mécanismes utilisés dans les systèmes distribués
- Leslie Lamport
  - Prix Turing 2014
  - Algorithme *Paxos* (1989)

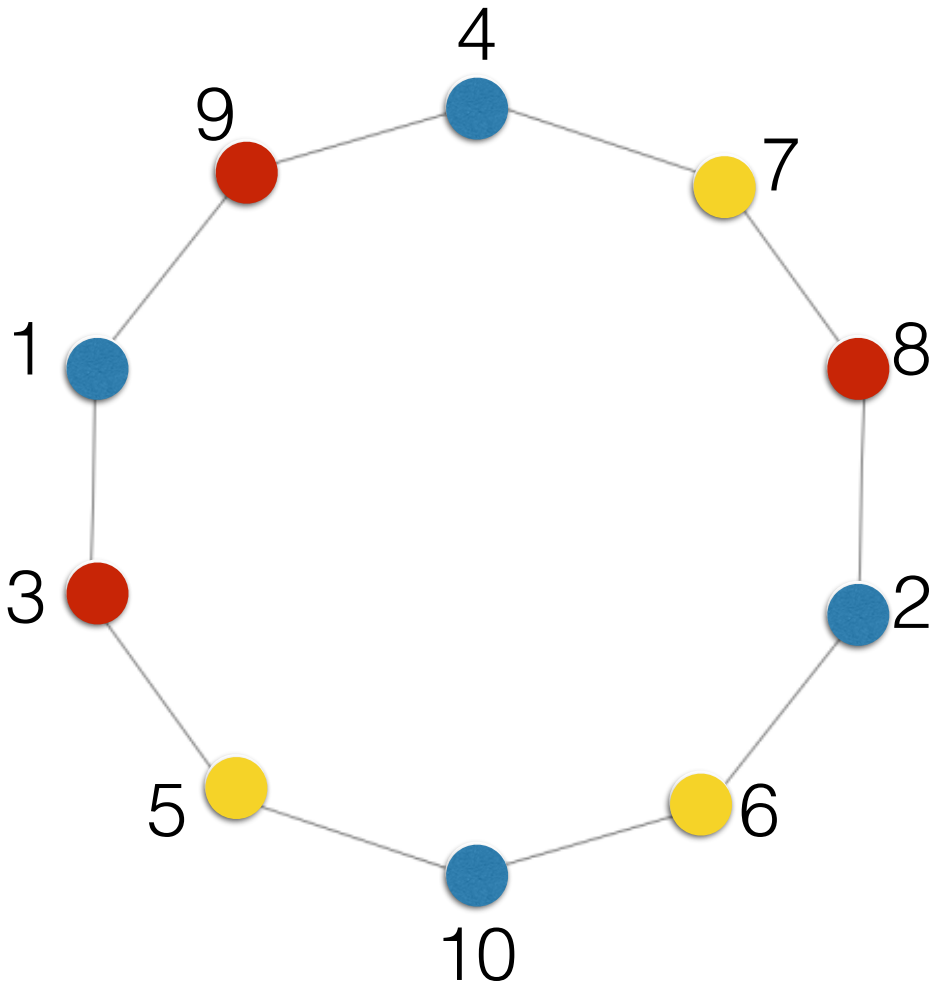
Incertitudes spatiales

Calcul local

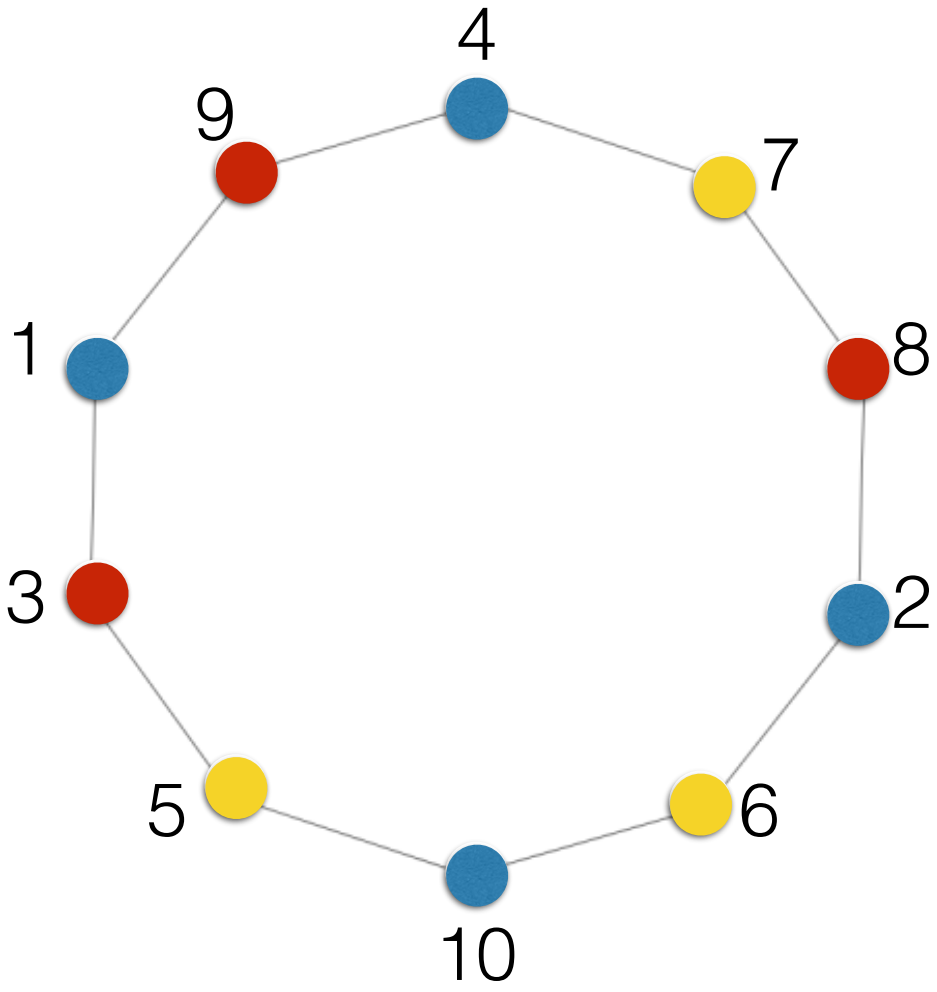
3-coloration de l’anneau

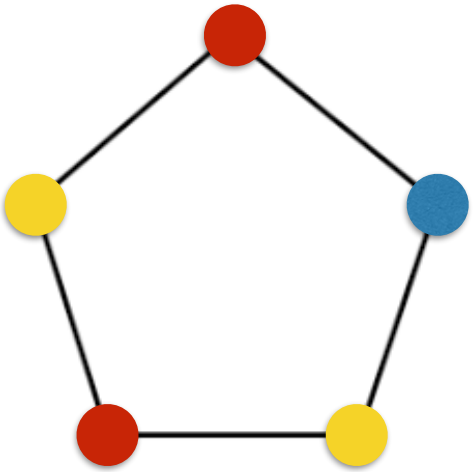


3-coloration de l’anneau

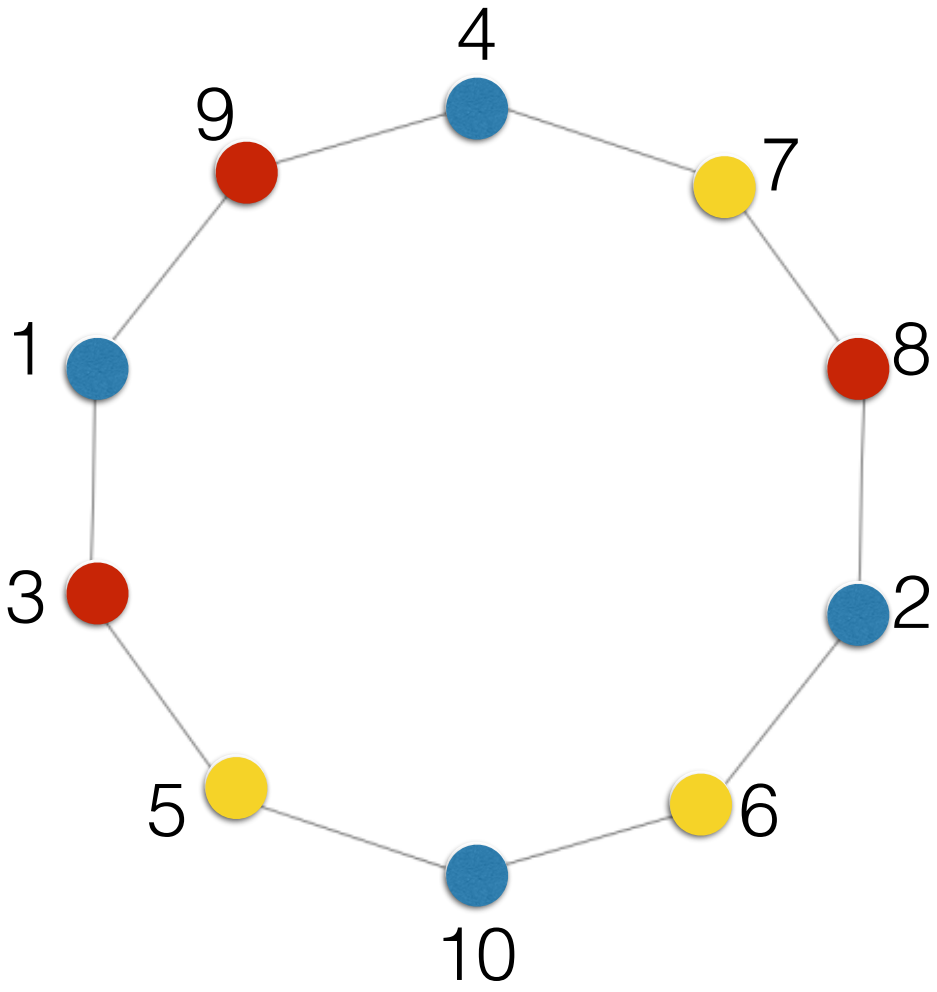


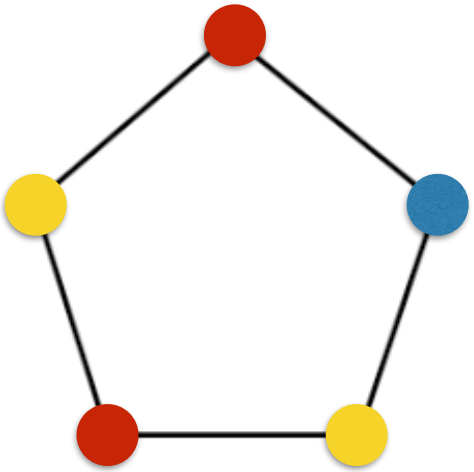
3-coloration de l’anneau





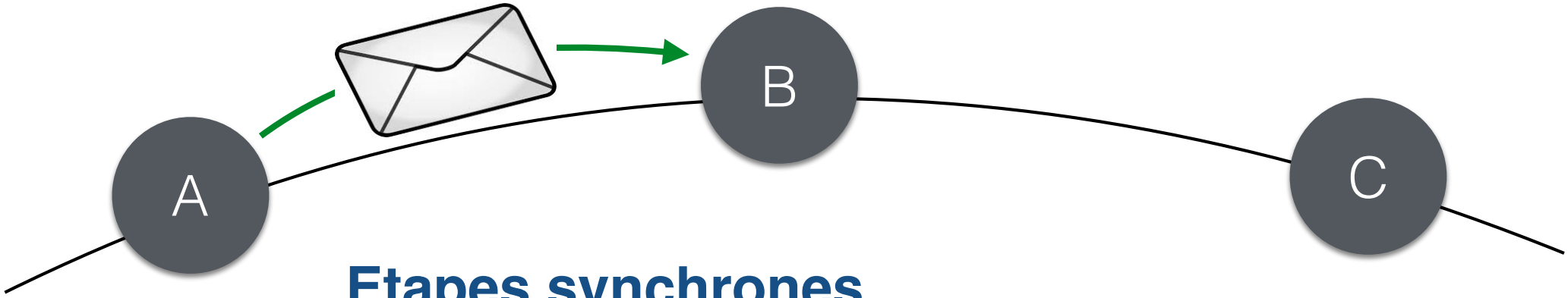
3-coloration de l'anneau





Modèle

Passage de messages



Etapes synchrones

- **Envois** de messages aux voisins ;
- **Réceptions** des messages
- **Calculs** individuels

nombre d'étapes = mesure de localité

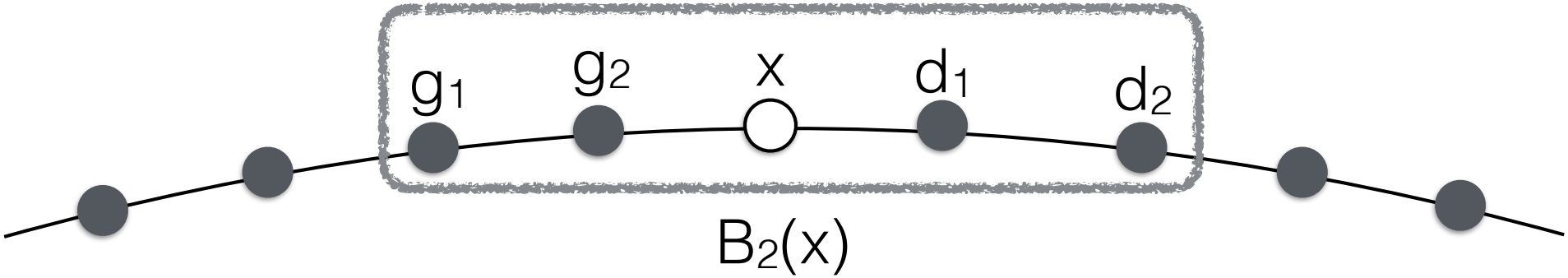
Impossibilité de la coloration locale

N. Linial (1992)

Théorème La 3-coloration de l'anneau de n sommets nécessite au moins log*n étapes.

Dijkstra Prize 2013

Preuve

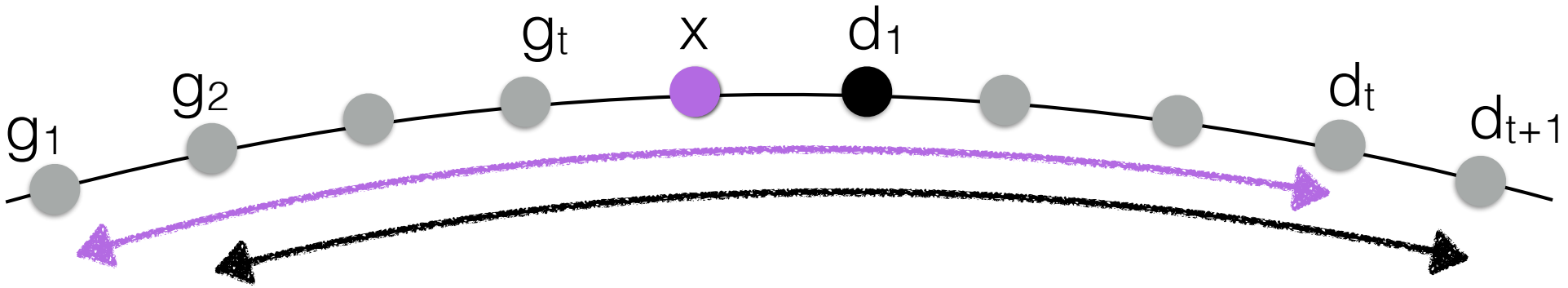


algorithme en t étapes ⟷ chaque sommet x décide en fonction de $B_t(x)$ où $B_t(x)=(g_1,g_2,\ldots,g_t,x,d_1,\ldots,d_{t-1},d_t)$

Graphe des configurations $G_{t,n}$

sommets = $\{ (g_1,\ldots,g_t,x,d_1,\ldots,d_t) \in \{1,\ldots,n\}^{2t+1} \}$

arêtes = $\{ (g_1,\ldots,g_t,x,d_1,\ldots,d_t) \quad (g_2,\ldots,g_t,x,d_1,\ldots,d_t,d_{t+1}) \}$



1. Algorithme en t étapes pour $C_n \Rightarrow \chi(G_{t,n}) \leq 3$
2. Lemme : $t<\log^* n \Rightarrow \chi(G_{t,n}) > 3$

❑

Utiliser l'aléa

- **P** = {$\mathcal{P}$ décidable déterministiquement en temps poly($|I|$)}
- **BPP** = {$\mathcal{P}$ décidable aléatoirement en temps poly($|I|$)}

BPP $\stackrel{?}{=}$ **P**

Utiliser l'aléa

- **P** = {$\mathcal{P}$ décidable déterministiquement en temps poly($|I|$)}
- **BPP** = {$\mathcal{P}$ décidable aléatoirement en temps poly($|I|$)}

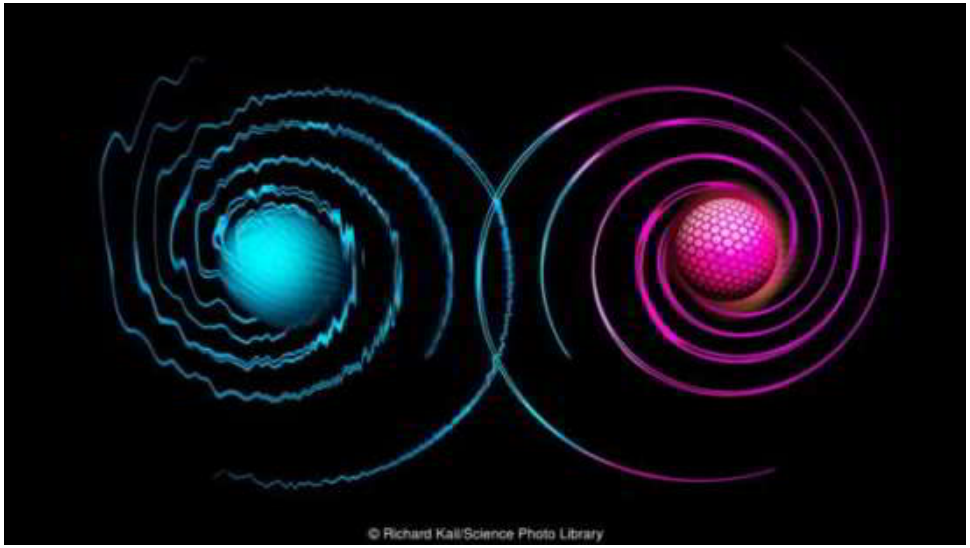
BPP $\stackrel{?}{=}$ P

- L'aléa aide pour briser la symétrie
- Algorithme de coloration aléatoire « essais-erreurs » en O(log n) étapes

est-ce que l'aléa aide pour la coloration distribuée?

Calcul distribué non classique

Calcul quantique



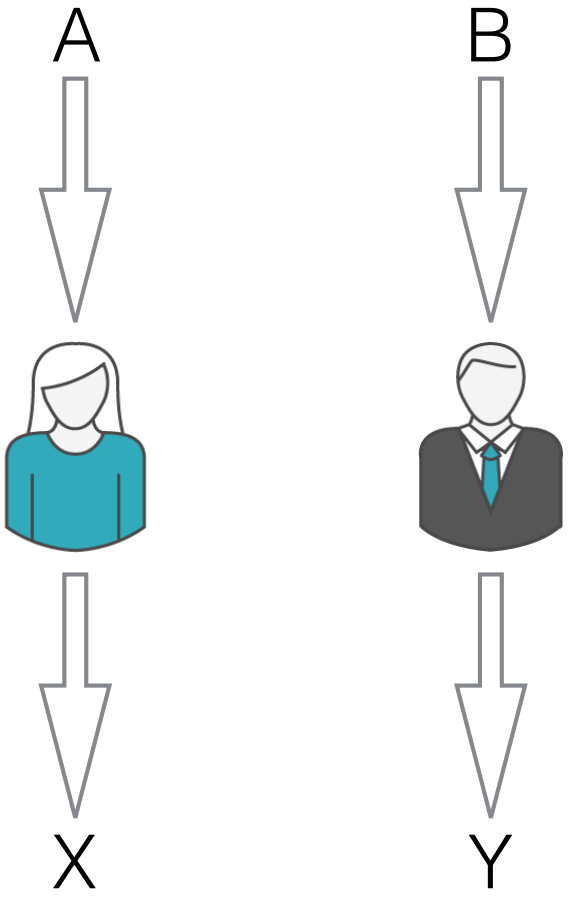
Le jeu XOR

ET

$\wedge$	VRAI	FAUX
VRAI	VRAI	FAUX
FAUX	FAUX	FAUX

OU exclusif

$\oplus$	VRAI	FAUX
VRAI	FAUX	VRAI
FAUX	VRAI	FAUX



$$X \oplus Y = A \wedge B$$

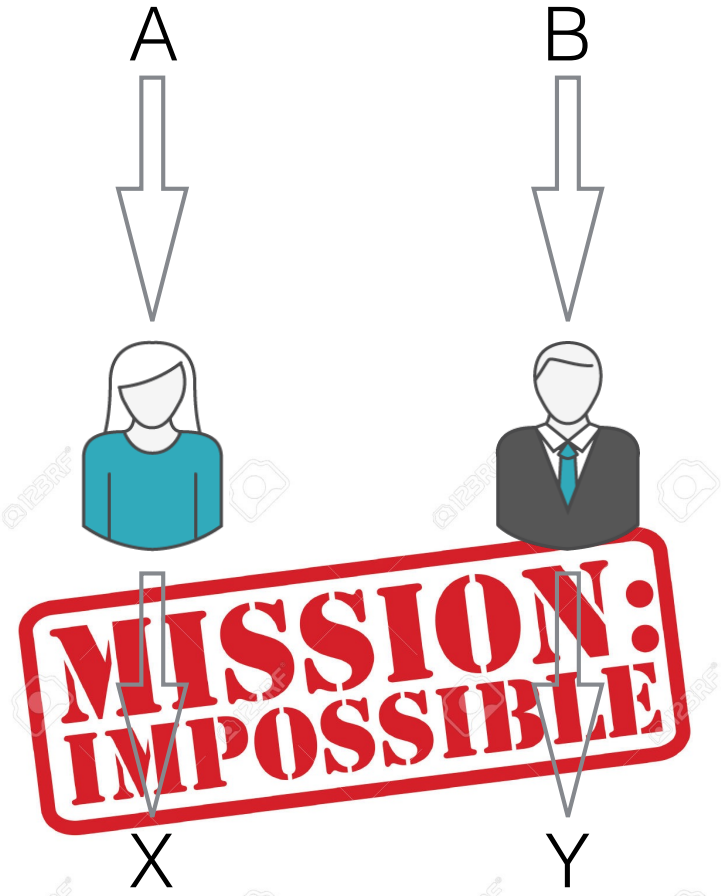
Le jeu XOR

ET

$\wedge$	VRAI	FAUX
VRAI	VRAI	FAUX
FAUX	FAUX	FAUX

OU exclusif

$\oplus$	VRAI	FAUX
VRAI	FAUX	VRAI
FAUX	VRAI	FAUX



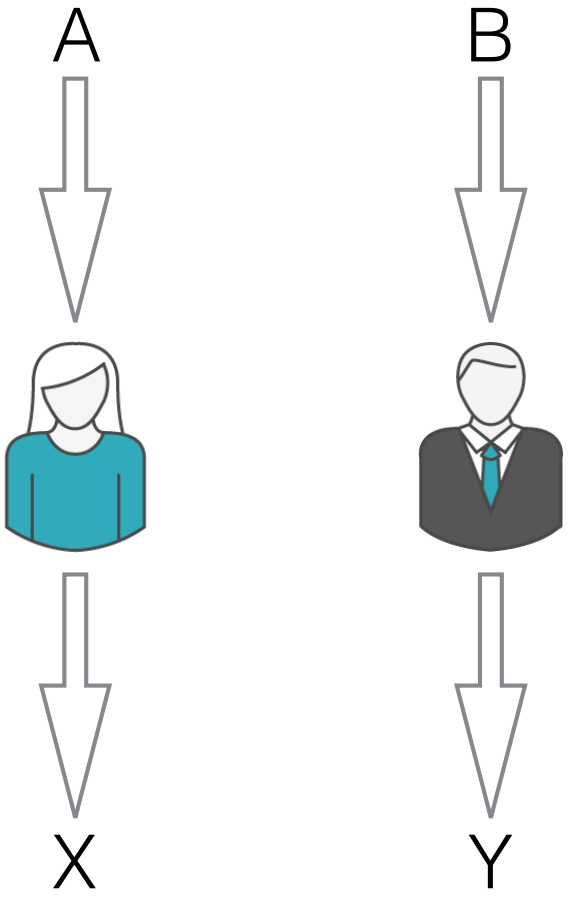
$$X \oplus Y = A \wedge B$$

Partage de ressources

$\wedge$	VRAI	FAUX
VRAI	VRAI	FAUX
FAUX	FAUX	FAUX

$\oplus$	VRAI	FAUX
VRAI	FAUX	VRAI
FAUX	VRAI	FAUX

$$X \oplus Y = A \wedge B$$

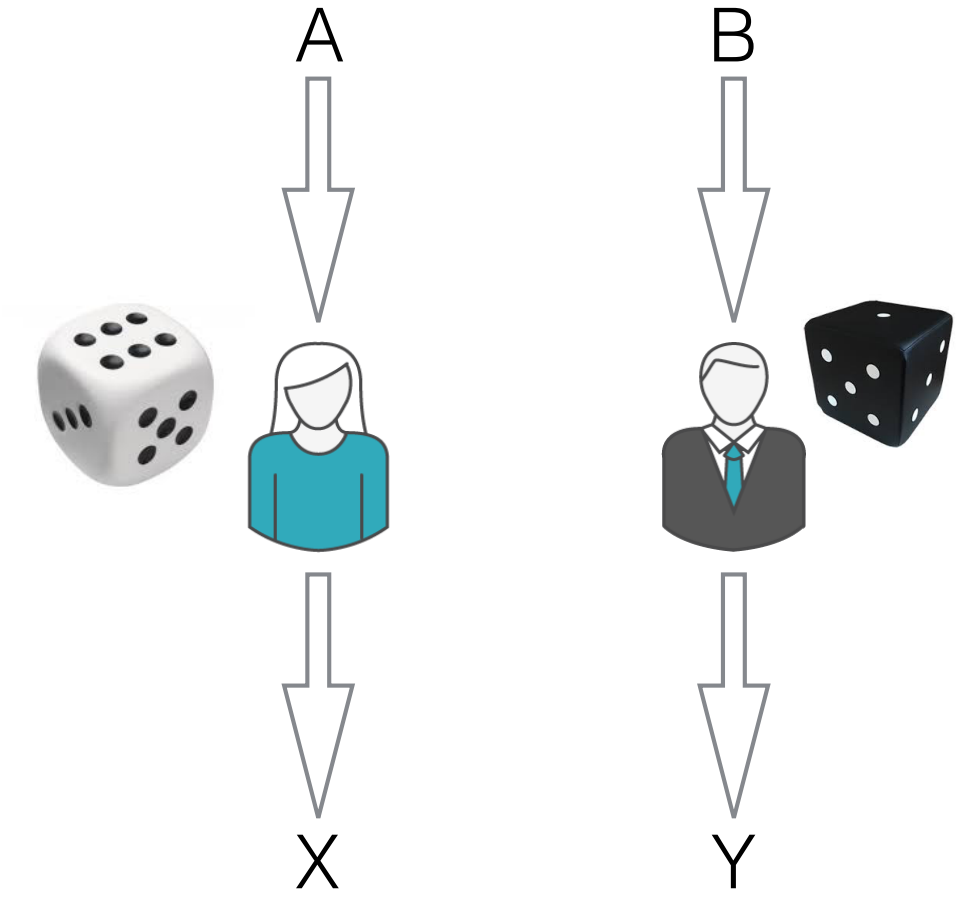


Partage de ressources

$\wedge$	VRAI	FAUX
VRAI	VRAI	FAUX
FAUX	FAUX	FAUX

$\oplus$	VRAI	FAUX
VRAI	FAUX	VRAI
FAUX	VRAI	FAUX

$$X \oplus Y = A \wedge B$$



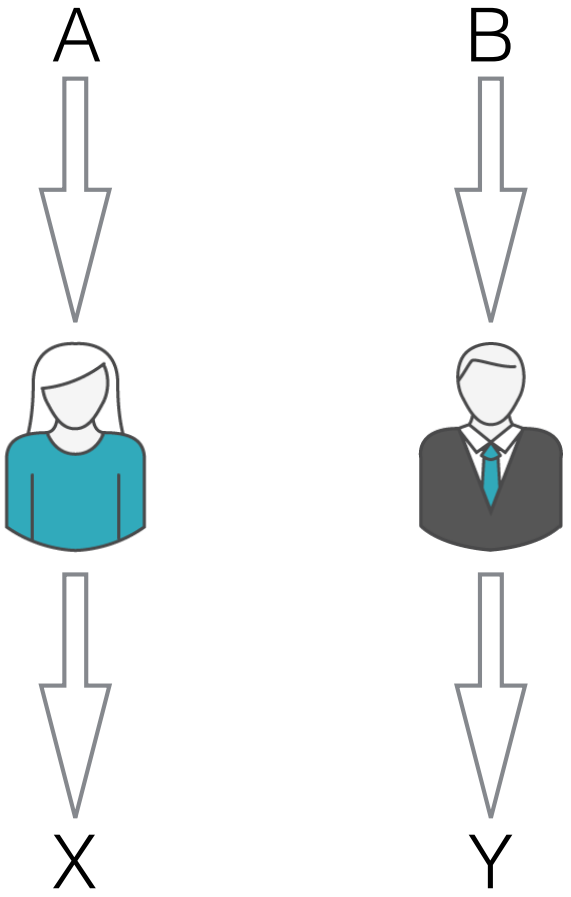
Aléa non partagé : Prob[succès] = ½

Partage de ressources

$\wedge$	VRAI	FAUX
VRAI	VRAI	FAUX
FAUX	FAUX	FAUX

$\oplus$	VRAI	FAUX
VRAI	FAUX	VRAI
FAUX	VRAI	FAUX

$$X \oplus Y = A \wedge B$$

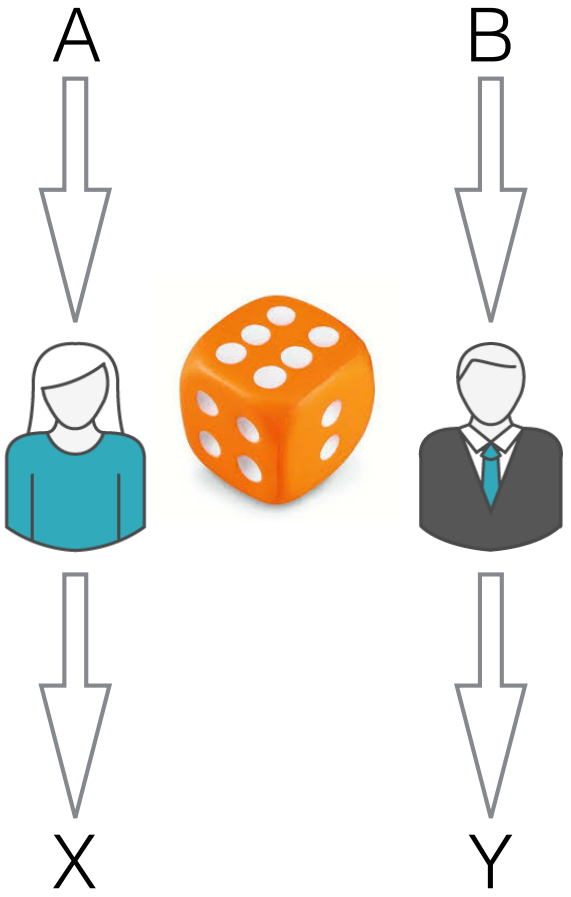


Partage de ressources

$\wedge$	VRAI	FAUX
VRAI	VRAI	FAUX
FAUX	FAUX	FAUX

$\oplus$	VRAI	FAUX
VRAI	FAUX	VRAI
FAUX	VRAI	FAUX

$$X \oplus Y = A \wedge B$$



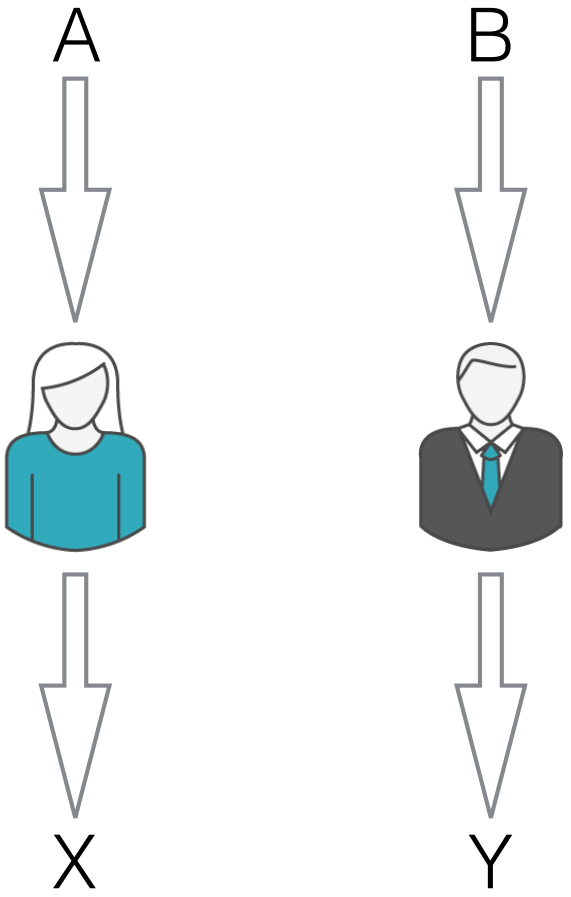
Aléa partagé : Prob[succès] = ¾

Partage de ressources

$\wedge$	VRAI	FAUX
VRAI	VRAI	FAUX
FAUX	FAUX	FAUX

$\oplus$	VRAI	FAUX
VRAI	FAUX	VRAI
FAUX	VRAI	FAUX

$$X \oplus Y = A \wedge B$$

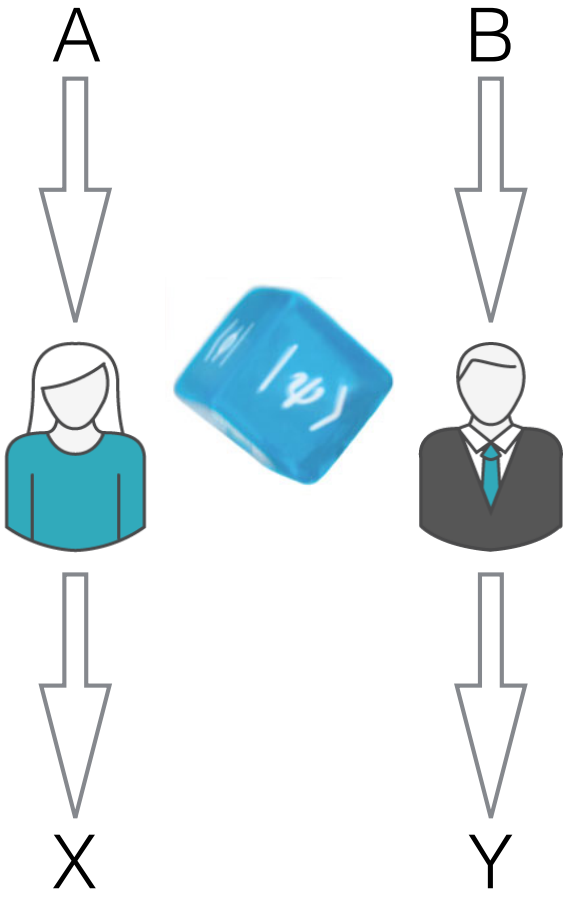


Partage de ressources

$\wedge$	VRAI	FAUX
VRAI	VRAI	FAUX
FAUX	FAUX	FAUX

$\oplus$	VRAI	FAUX
VRAI	FAUX	VRAI
FAUX	VRAI	FAUX

$$X \oplus Y = A \wedge B$$



Aléa quantique :

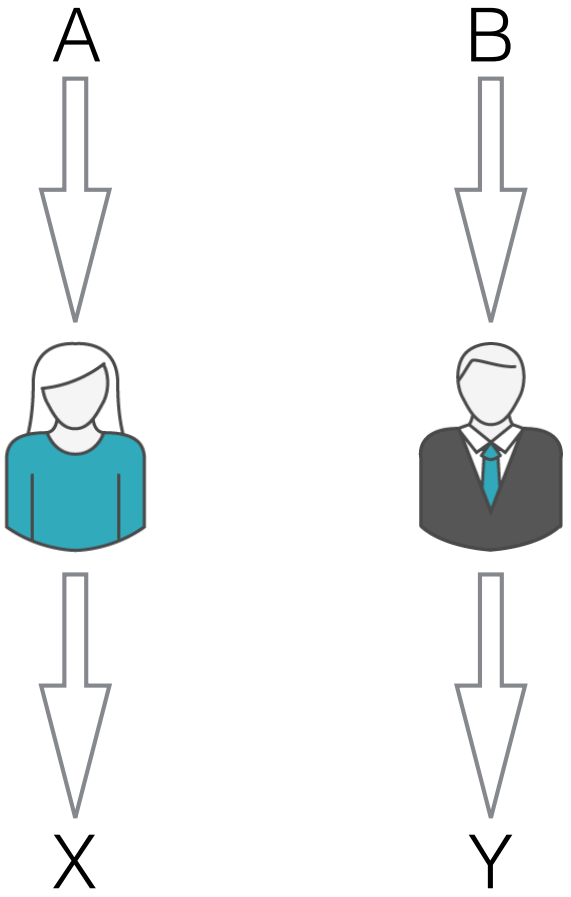
Prob[succès] = $\cos^2(\pi/8) = 0{,}85\ldots$

Partage de ressources

$\wedge$	VRAI	FAUX
VRAI	VRAI	FAUX
FAUX	FAUX	FAUX

$\oplus$	VRAI	FAUX
VRAI	FAUX	VRAI
FAUX	VRAI	FAUX

$$X \oplus Y = A \wedge B$$

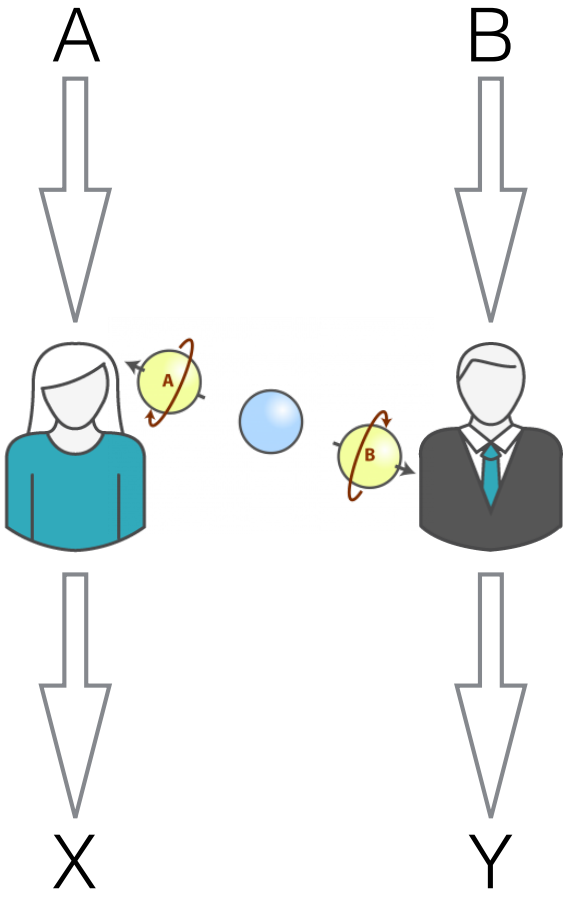


Partage de ressources

$\wedge$	VRAI	FAUX
VRAI	VRAI	FAUX
FAUX	FAUX	FAUX

$\oplus$	VRAI	FAUX
VRAI	FAUX	VRAI
FAUX	VRAI	FAUX

$$X \oplus Y = A \wedge B$$



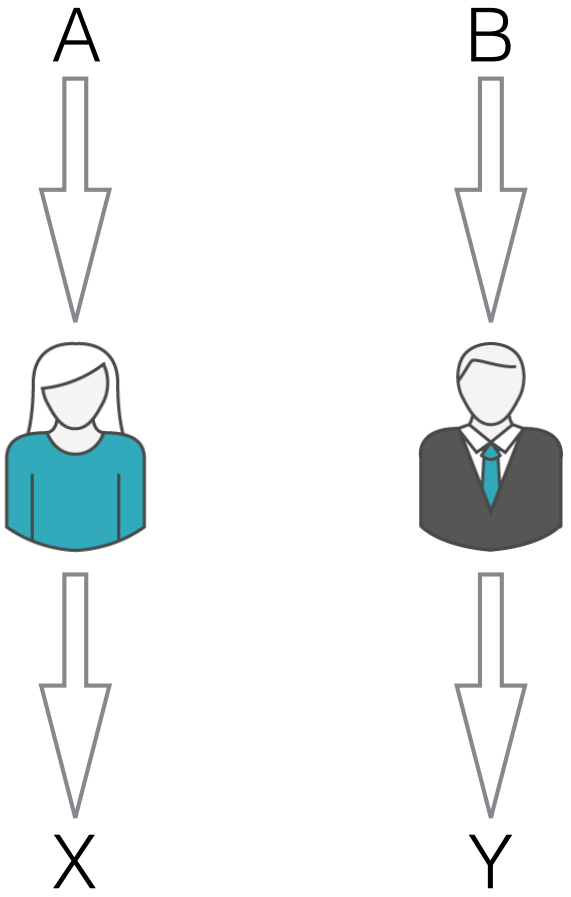
Aléa non local : Prob[succès] = 1

Partage de ressources

$\wedge$	VRAI	FAUX
VRAI	VRAI	FAUX
FAUX	FAUX	FAUX

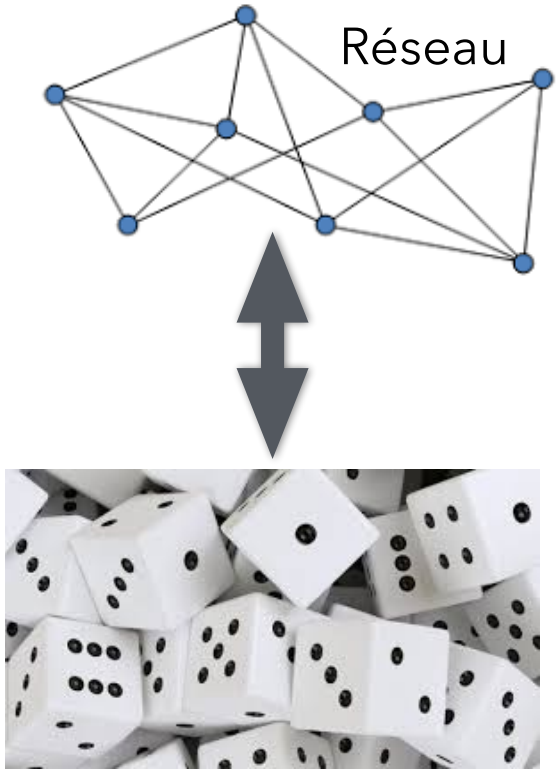
$\oplus$	VRAI	FAUX
VRAI	FAUX	VRAI
FAUX	VRAI	FAUX

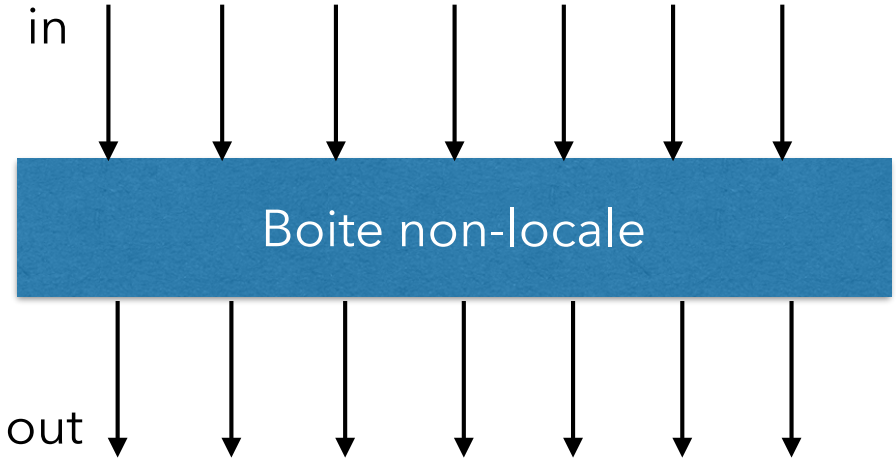
$$X \oplus Y = A \wedge B$$



Question ouverte

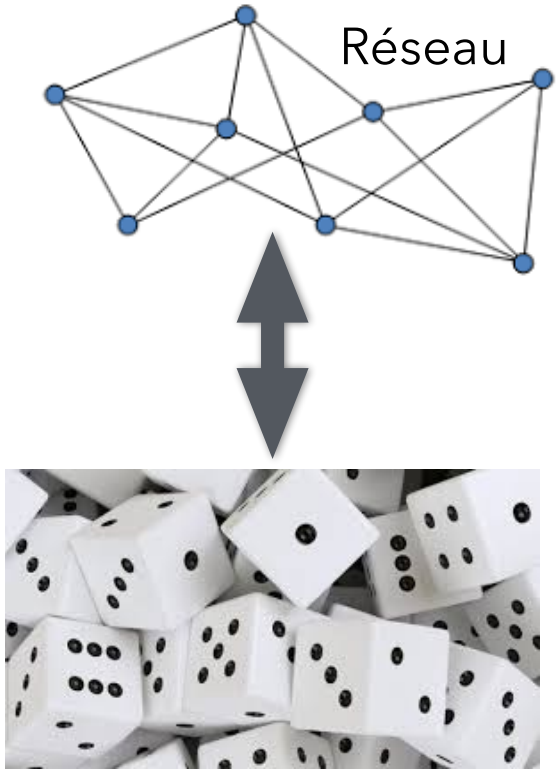
Conception d'un algorithme distribué quantique efficace pour la coloration.

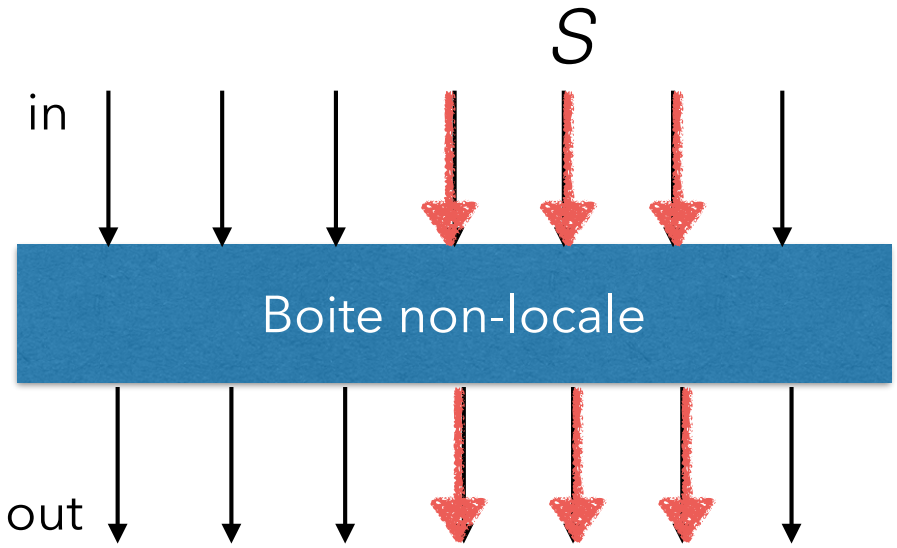




Question ouverte

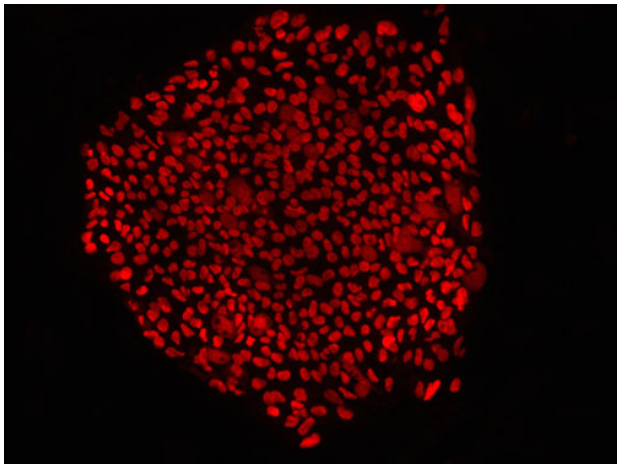
Conception d'un algorithme distribué quantique efficace pour la coloration.



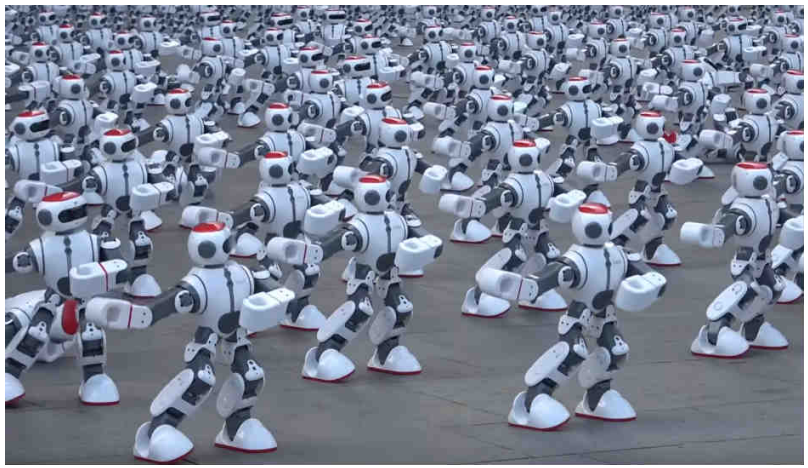


distribution de out(S)
ne dépend que de in(S)

Quantité d'autres modèles



- Réseaux dynamiques
- Mobilité (agents logiciels, robots,…)
- Affaiblissement de la puissance de calcul individuelle
- Médium communication (radio, faible bande passante,…)
- Etc.



Conclusion

Conclusion

algorithmes distribués

- gérer l'incertitude

Conclusion

algorithmes distribués

- gérer l'incertitude

outils

- algorithmique
- théorie des graphes
- combinatoire
- topologie algébrique
- complexité de la communication

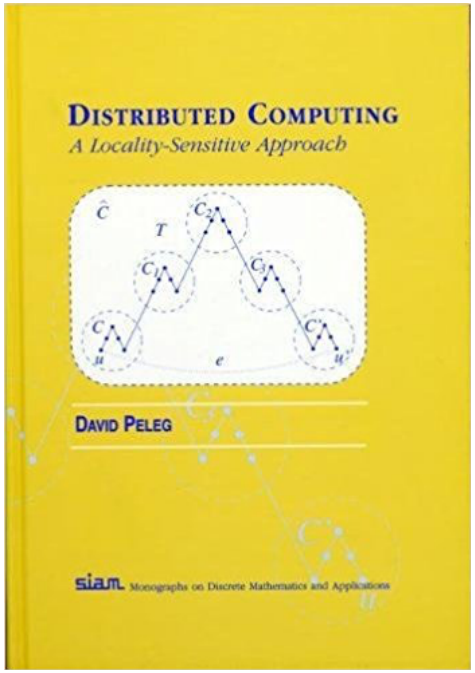
Conclusion

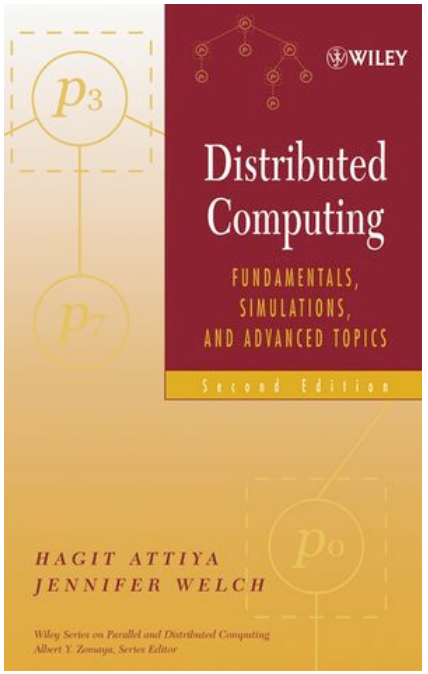
algorithmes distribués

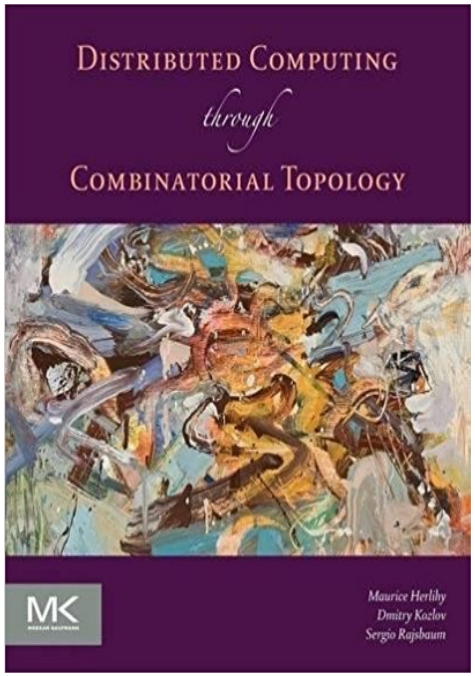
- gérer l'incertitude

outils

- algorithmique
- théorie des graphes
- combinatoire
- topologie algébrique
- complexité de la communication







Conclusion

algorithmes distribués

- gérer l'incertitude

Merci !

outils

- algorithmique
- théorie des graphes
- combinatoire
- topologie algébrique
- complexité de la communication