# Algorithmique répartie

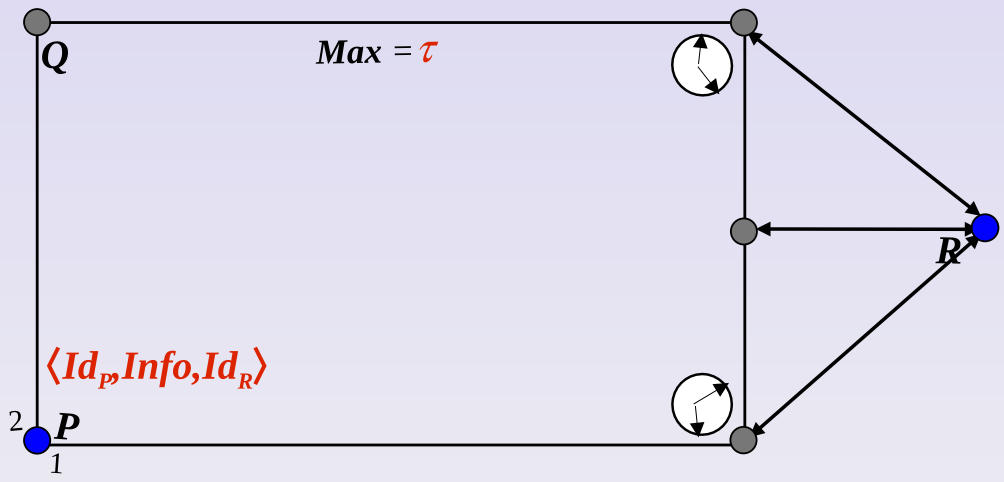
C. Lavault

LIPN (UMR 7030), université Paris 13

Aléa 2009

## Partie I – Algorithmique répartie classique

## Système réparti $\mathcal{S}$ : processus-communication

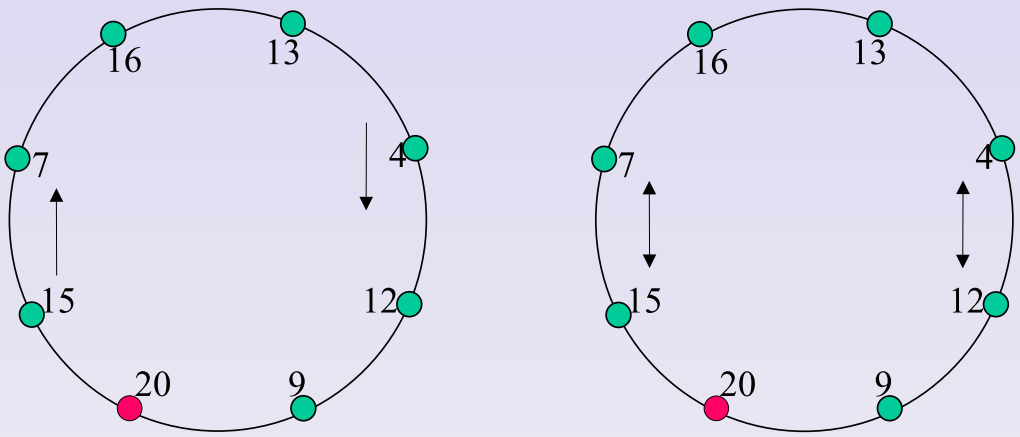



## Modèle de système réparti $\mathcal{S}$

Modèle de $\mathcal{S}$ : $G = (\mathcal{P}, \mathcal{L})$ $(|\mathcal{P}| = n, |\mathcal{L}| = m)$
graphe connexe, simple et symétrique.

- Processus $\rightsquigarrow$ tout est local : état, mémoire, horloge, identité, tampons, variables, calculs, etc.
- Communication par messages
  – point-à-point/multipoint (voisins/diffusions)
  – synchrone/asynchrone
  synchronisation virtuelle : algorithme $\mathcal{A}$ *Event-Message Driven*
  asynchrone à délais bornés (par $\tau$) : algorithme $\mathcal{A}$ partiellement synchrone
- $\mathcal{A}$ - $\mathcal{S}$ avec/sans information globale statique/dynamique avec identités/anonyme
- Terminaison de $\mathcal{A}$ ? explicite/implicite
- Fautes ? Pannes franches, fautes byzantines (processus, lignes), défaillances transitoires/systémiques $\rightsquigarrow$ autostabilisation
- Mesures de complexité de $\mathcal{A}$ ? en messages/bits, en mémoire, en temps/phases-rondes

## *Algorithmes d'élection sur un réseau en anneau*




Élection sur un anneau unidirectionnel [Chang,Roberts-79]
et sa variante bidirectionnelle [CL-90]

## *Analyse de l'algorithme de Chang-Roberts*

- Complexité en messages au pire :

$$M_{max}(n) = \sum_{1\le j\le n} j + n = \frac{n^2}{2} + \frac{3n}{2}$$

- Complexité en messages $\langle j\rangle$ en moyenne $(j = 1,\ldots, n)$

$$p_{k,j} = \mathbb{P}\,(\langle j\rangle \text{ parcourt } k \text{ liens}) \quad (1 \le k \le j \le n-1)$$

$$= \frac{\binom{j-1}{k-1}}{\binom{n-1}{k-1}} \times \frac{n-j}{n-k}$$

$$\mathbb{E}(M_n) = n + \sum_{1\le j\le n-1} \sum_{1\le k\le j} k\,p_{kj} + n = nH_n + n$$

$$\text{FGP de } M_n : P(z) = \frac{n}{i\binom{n}{i}} \sum_{1\le k\le n-1} \binom{n}{i-k} (z-1)^k.$$

## *Analyse de l'algorithme de Chang-Roberts (fin)*

$$\mathbb{E}(M_n) = n\ln n + (\gamma+1)n + 1/2 + \mathcal{O}\left(n^{-1}\right) = \Theta(n\log n).$$
$$\mathrm{var}(M_n) = (2 - H_n^{(2)})n^2 - nH_n = (2-\pi^2/6)n^2 - n\ln n - 1 + \mathcal{O}\left(n^{-1}\right).$$

Attention. Pour $i \neq \ell$ entiers de $[n-1]$, les v.a. $M_i$ et $M_\ell$ ne sont même *pas deux à deux indépendantes*. Mais, les v.a. $M_j$ sont deux à deux non corrélées $(1 \leq j \leq n-1)$. Donc,

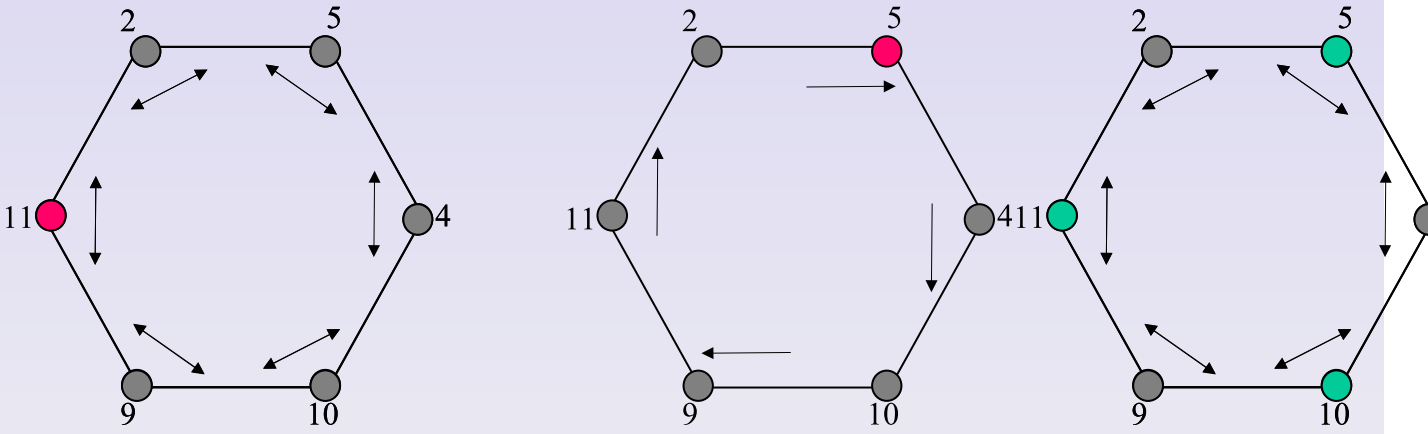
$$\mathrm{var}(M_n) = \sum_{1\leq j\leq n-1} \mathrm{var}(M_j) \ldots$$

- Complexité en phases virtuelles (en « temps ») :

$$T_{max}(n) = (n-1) + n + n = 3n - 1.$$

□

## *Algorithmes d'élection sur un anneau (bis)*




Deux élections : sur un anneau bidirectionnel [Franklin-82] et unidirectionnel [Peterson-82], [Dolev,Klawe,Rodeh-82]

## *Analyse de l'algorithme de Franklin*

Au pire/en moyenne : $2n$ messages par phase, sauf terminaison : $n$.

- Complexité en phases virtuelles et en messages au pire : $M_{max}(n) = 2n \times$ # maximal de phases, i.e. lorsque le nombre de pics est maximal $= \lfloor n/2 \rfloor$.

$$\#\text{ maximal de phases} = \lfloor \lg n \rfloor + 1$$
$$M_{max}(n) = 2n\lfloor \lg n \rfloor + n = \Theta(n \log n).$$

- Complexité en messages en moyenne ($n$ candidats initiaux)

Moyenne et variance du nombre de pics d'une permutation $\sigma \in \mathbb{S}_n$ [Carlitz-74], [Flajolet,Sedgewick-09] :

$$\mathbb{M}(n) = (n-2)/3 \quad (n \geq 2) \quad \text{et} \quad \mathbb{V}(n) = 2(n+1)/45 \quad (n \geq 4).$$

## *Analyse de l'algorithme de Franklin (suite)*

Distribution des pics sur l'anneau : $P_c(n,k) = \mathbb{P}(n \text{ candidats}, k \text{ pics})$.

Distribution des $k$ pics d'une permutation $\sigma \in \mathbb{S}_n$ : $P(n,k)$.

Par une récurrence simple, $P_c(n,k) = P(n-1,k-1)$, et donc,

$$\mathbb{M}_c(n) = n/3 \quad (n \geq 3) \quad \text{et} \quad \mathbb{V}_c(n) = 2n/45 \quad (n \geq 5).$$

## *Analyse de l'algorithme de Franklin (suite)*

Problème ! Les pics restant (candidats) comparent leurs *Ids* de phase en phase, jusqu'à l'élection. En conditionnant sur le nombre $\xi$ de pics restant en fin de phase 1, les $\xi!$ ordres distincts de la phase 2 n'ont pas la même probabilité.

Pour $n = 8$ et $\xi = 4$ en fin de phase 1, la probabilité de garder 2 pics en phase 2 est $10/34$, tandis qu'elle est $1/3$ dans le cas uniforme...

Nombre moyen de pics survivant après 2 phase : $c_2 n + o(1)$ ($c_2 \approx 0,1096868681 \lesssim 1/9$) : Franklin n'est pas asymptotiquement equivalent à sa variante par permutations « en ligne »... ouvert ! [Janson,CL,Louchard-09]

## *Analyse de l'algorithme de Franklin (suite)*

Simplification ?

Les pics survivant tirent de nouvelles *Ids* à chaque phase : leur distribution reste inchangée de la phase 1 à la dernière.

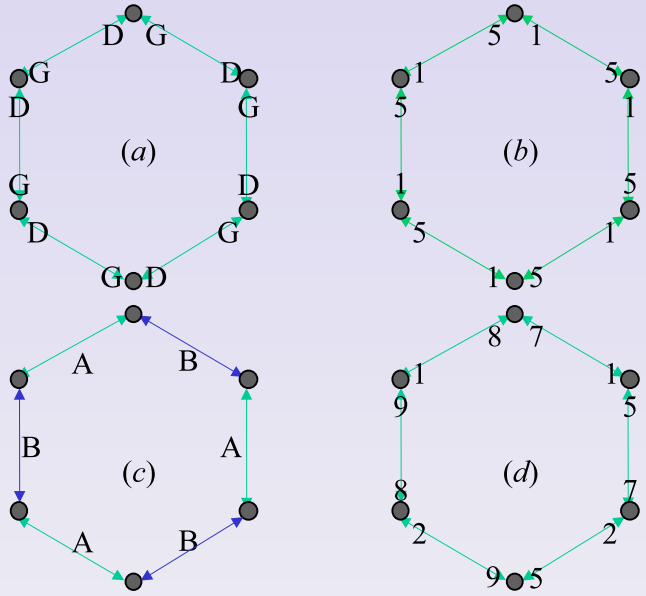
$$\#\text{ moyen de phases} = \log_3(n) + \phi(n) + o(1)$$

(où $\phi(n)$ périodique et $|\phi(n)| < 1$) :

$$\mathbb{E}(M_n) = 2n\log_3(n) + \mathcal{O}(n) = \Theta(n \log n).$$

□

## *Les sens de la direction (SD) sur l'anneau*




Les SD sur l'anneau : (a) gauche/droite, (b) par cordes,
(c) dimensionnel, (d) par voisinage

## *Élection sur les anneaux – bref récapitulatif*

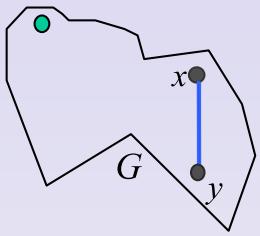
| **messages** | **CR** | **Franklin** | **Peterson** | **DKR** | Przytycka-93 |
|---|---|---|---|---|---|
| au pire | $\mathcal{O}(n^2)$ | $2n \lg n$ | $n \log_\phi(n)$ | $1.356\, n \lg n$ | $1.271\, n \lg n$ |
| moyenne | $nH_n$ | $2n \log_3(n)$ | – | – | – |
| | $0.69\, n \lg n$ | $3.169\, n \lg n$ | – | – | – |

Algorithmes d'élection optimaux, anonymes, asynchrones.
Taille des messages $= \mathcal{O}(\log(Id_{max}) + \log n)$.

### Théorème

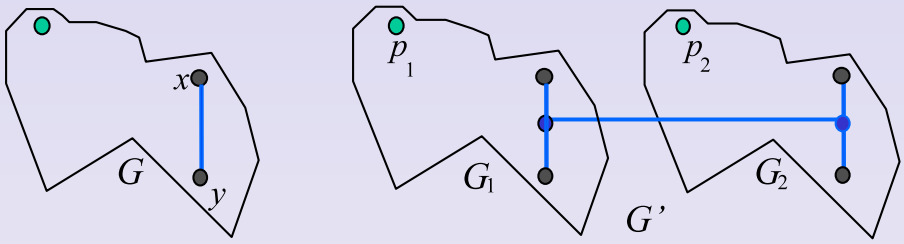
*La borne inférieure asymptotique de complexité en messages (en moyenne ou au pire) des algorithmes d'élection par comparaisons d'identités sur un anneau est en* $\Omega(n \log n)$ *[Pachl,Korach,Rotem-84].*
*Celle des algorithmes d'élection sur un anneau unidirectionnel (avec SD) est* $(1/4 - \epsilon)nH_n$ *en moyenne et* $\frac{1}{2}n \lg n$ *au pire [Bodlaender-91]* □

# *Borne inférieure de (E) sur un réseau quelconque*



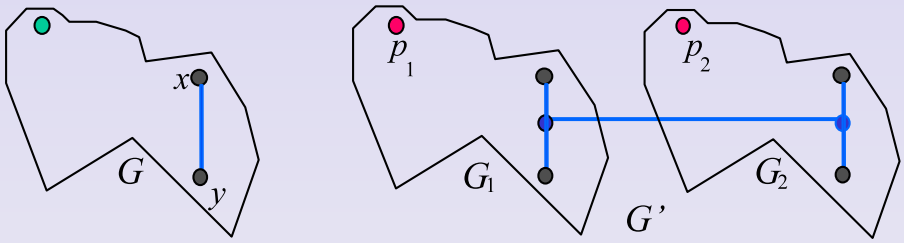

Supposons que le # messages de $(E)$ soit $< m$. Alors, pour un algorithme d'élection $\mathcal{A}$ (par comparaisons), aucun message n'utilise une arête $xy$ de $G$ dans au moins une exécution $\mathcal{E}$ de $\mathcal{A}$

# *Borne inférieure de (E) sur un réseau quelconque (suite)*




Le graphe inital $G$ est dupliqué en $G' = G_1 \cup G_2$, reliés par une arête d'extrémités sur $xy$. Les identités ont le même ordre relatif que dans $G$ ; $\mathcal{E}$ est identique sur $G_1$ et $G_2$ dans $\mathcal{A}$

## Borne inférieure de (E) sur un réseau quelconque (suite)




Conclusion : deux élus par $\mathcal{A}$, $p_1$ et $p_2$ : pas d'élection !.

Dans toute exécution de $\mathcal{A}$, un message passe par chaque arête de $G$.

□

## Bornes inférieures, de l'élection à la diffusion

- Une élection $(E)$ permet de construire un protocole de contrôle :
  $(AR)$ un arbre de recouvrement ou $(ARM)$ de poids total minimal. $(E)$, $(AR)$ et $(ARM)$ sont des problèmes de complexité équivalente.
- Dans une diffusion $(D)$ (*Broadcast*, inondation) un sommet-source unique envoie un message à tous les autres sommets du réseau.

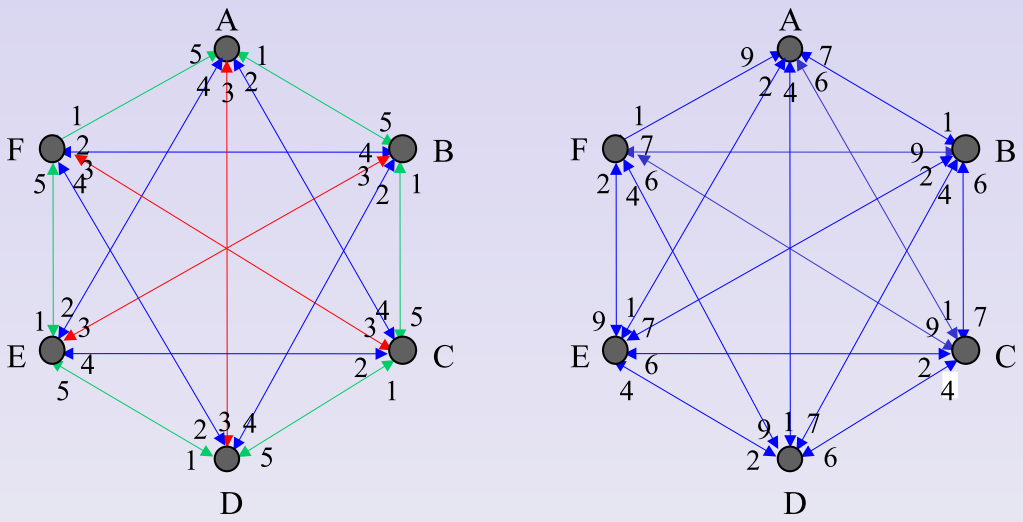
### Théorème

*– La borne inférieure de complexité en messages de (E), (AR) et (ARM) sur un réseau quelconque est en $\Omega(m + n \log n)$. Celle de leur complexité en phases (en « temps ») est en $\Omega(D)$, où $D$ diamètre du graphe.*

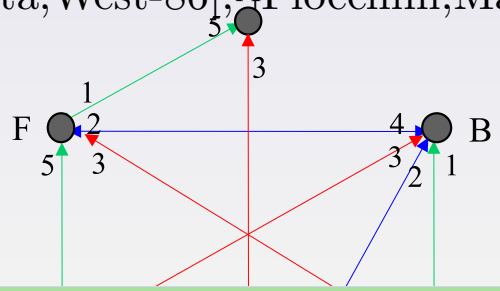
*– Les bornes inférieures de complexité en messages (resp. en temps) de (D) sont en $\Omega(m)$ et en $\Omega(R)$, où $R$ rayon du graphe.*

□

## *Les sens de la direction (SD) sur les réseaux*




SD par cordes et par voisinage sur un réseau complet
[Loui,Matsushita,West-86],[Flocchini,Mans,Santoro-98]




## *Étiquetage d'un réseau et SD construit sur un groupe*

Groupe commutatif $G$. Deux sommets voisins $p$ et $q$, $\mathcal{E}_p(q)$ est le nom de l'arête $pq$ en $p$.
Idée SD : faire coïncider la topologie du réseau et la structure de $G$.

### Définition

*L'étiquetage d'arêtes $\mathcal{E}$ est un SD (construit sur $G$) si ces étiquettes sont des éléments de $G$ et qu'il existe une injection $\sigma : \mathcal{P} \to G$, telle que pour tout couple $(p, q)$ de voisins, $\sigma_q = \sigma_p + \mathcal{E}_p(q)$.*

### Remarques

$\mathcal{E}_p(q) + \mathcal{E}_q(p) \equiv 0 \pmod{n}$.
Désormais, $n = ord(G)$ : $\sigma$ est bijective (simplification). Un réseau de taille $n$ et un groupe $G$ d'ordre $n$ permettent de construire un SD : on étiquette les sommets arbitrairement par $G$, puis, pour chaque arête $pq$, on prend $\mathcal{E}_p(q) = \sigma_q - \sigma_p \ldots$

## *Influence d'un SD sur la complexité en messages*

| **Réseau** | $(D)$ et $(PP)$ | $(E)$, $(AR)$ et $(M)$ |
|---|---|---|
| quelconque sans SD | $\Omega(m)$ | $\Omega(m + n \log n)$ |
| quelconque avec SD | $2n - 2 = \Theta(n)$ | $3n \lg n + \mathcal{O}(n)$ |
| complet sans SD | $\Omega\left(n^2\right)$ | $\Omega\left(n^2\right)$ |
| complet avec SD | $\Theta(n)$ | $\Theta(n)$ |
| $d$-Hypercube sans SD | $\Omega(nd)$ | $\Omega(n \log n)$ |
| $d$-Hypercube avec SD | $\Theta(n)$ | $\Theta(n)$ |

Réduction de la complexité en messages (SD quelconque).
($(PP)$ : parcours en profondeur, $(M)$ : calcul de Max/Min.)

Tout SD sur $G = \mathbb{Z}_n$ est un SD par cordes : pour un anneau ou une clique : $G = \mathbb{Z}_n$, pour un tore : $G = (\mathbb{Z}_n)^2$, pour un $d$-hypercube : $G = (\mathbb{Z}_2)^d$.

## *Borne inférieure en messages de l'orientation (SD)*

### Théorème

*Un algorithme de construction d'un SD nécessite au moins $m - n/2$ messages pour toute exécution sur un réseau quelconque.* □

### Exemples

Sans aucune information structurelle($D$, $\Delta$, topologie, etc.)... sauf $n$

L'orientation (SD) d'une $n$-clique nécessite $\Omega\left(n^2\right)$ messages.
L'orientation (SD) d'un $d$-hypercube nécessite $\Omega(nd)$ messages.
L'orientation (SD) du tore $n \times n$ nécessite $\Omega\left(n^2\right)$ messages.

## *Un algorithme de coloration synchrone*

Problème COLORATION. Soit un graphe non orienté $G = (X, U)$, attribuer une couleur $c_u$ à chaque sommet $u$, t.q. si $vw \in U$, $c_v \neq c_w$ en utilisant un minimum de couleurs : $\chi(G)$ (indice chromatique de $G$).

Algorithme 1. Initialement, les sommets sont colorés par des couleurs dans $[C]$. Chaque sommet $u$ exécute (en parallèle)

1. pour $x$ de $\Delta + 2$ à $C$ faire
2. **si** $c_u = x$ **alors**
3. $u$ tire la plus petite couleur disponible (celle d'aucun voisin) dans $\{1, \ldots, \Delta + 1\}$
4. $u$ informe tous ses voisins de son choix

Complexité en temps : $C + \Delta + 1$, soit $\mathcal{O}(n)$ si $C = \mathcal{O}(n)$.
L'algorithme utilise $\Delta + 1$ couleurs.
Complexité en messages : $\mathcal{O}(\Delta)$ (si $\Delta$ constant, exemple de coût en messages négligeable devant le coût en temps...). □

## *Un bon algorithme de coloration d'arbres*

Rappel : pour un arbre $A$, $\chi(A) \leq 2$.

Algorithme 2. (*Descente*) En concurrence, pour tout sommet de $G$, recolorer avec la couleur de son père. Pour la racine de l'arbre (exception), tirer un nouvelle couleur au hasard.

Remarque. L'algorithme 2 préserve la coloration(cohérente) et tous les enfants d'un même père sont monochromes.

Algorithme 3. (*Coloration six-à-trois*) En tout sommet de $G$

1. **pour** $x$ de 4 à 6 **faire**
2. exécuter l'algorithme 2 (*Descente*)
3. un sommet $u$ de couleur $x$ choisit une nouvelle couleur selon la procédure 3 de l'algorithme 1

Complexité en temps : L'algorithme 3 colore un arbre avec trois couleurs en temps $\mathcal{O}(\log^* n)$. □

## *Complexité d'algorithmes de $(\Delta+1)$-coloration*

| Références | Complexité | Borne inférieure |
|---|---|---|
| Linial-87 | $\mathcal{O}\left(\Delta^2\right)+\log^* n$ | $\log^* n-\mathcal{O}(1)$<br>*($f(\Delta)$−coloration $\forall f$)* |
| Szegedy, Vishwanathan-93 | $\mathcal{O}\left(\Delta^2\right)+\frac{1}{2}\log^* n$ | $\mathcal{O}(\Delta\log\Delta)$<br>*(localt. itér.)* |
| Peleg-00 | $\mathcal{O}(\Delta\log n)$<br>$\mathcal{O}(\log^* n)$ *($\Delta$ borné)* | $\frac{1}{2}(\log^* n-1)$<br>*($n$-anneau, 3 couleurs)* |
| Kuhn, Wattenhoffer-06 | $\mathcal{O}(\Delta\log\Delta)+\log^* n$<br>$\mathcal{O}(\Delta\log\log n)$<br>*(probabiliste)* | –<br>–<br>– |
| Barenboim, Elkin-08 | $\mathcal{O}(\Delta)+\frac{1}{2}\log^* n$ | – |

## *Briser la symétrie – Théorèmes d'impossibilité*

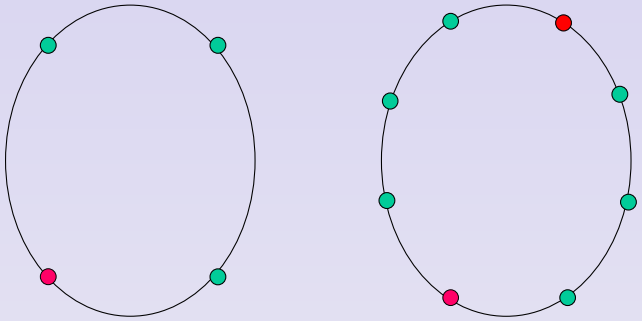
**Théorème [Angluin-80], [Itai,Rodeh-81]**

*– Il n'existe aucun algorithme d'élection déterministe dans un réseau anonyme, de taille $n$ connue ou non des processus.*

*– Il n'existe aucun algorithme probabiliste uniforme ($n$ inconnu) d'élection sur un anneau anonyme de taille unique qui termine en une unique exécution.*

*– Il n'existe aucun algorithme déterministe uniforme avec terminaison distribuée explicite (ou « par processus »), qui calcule une fonction quelconque non constante.*

## *Impossibilité d'élection sur un anneau avec $\mathcal{A}$ uniforme*



$\mathcal{A}$ ne peut terminer l'élection à la fois pour $n = 4$ et $n = 8$.

Pour une exécution $\mathcal{E}_1$(avec l'aléa), $\mathcal{A}$ élit $P$ sur un $n$-anneau. Soit une exécution $\mathcal{E}_2$de $\mathcal{A}$ sur un $2n$-anneau, telle que les événements de $P_i$ et $P_{n-i}$ dans $\mathcal{E}_1$ sont ceux de $Q_i$ dans $\mathcal{E}_2$ ($i = 0, 1, \ldots, n-1$), dans le même ordre relatif. $\mathcal{E}_2$ est licite et $P_i$ et $P_{n-i}$ suivent les mêmes étapes : dans $\mathcal{E}_2$, ils sont tous deux élus. □

## *Comment briser la symétrie ? Deux conditions*

Conséquences. Briser la symétrie ou/et obtenir un consensus dans un réseau anonyme suppose deux conditions.

1. La connaissance globale de la taille $n$ du réseau, exacte ou approchée (de bornes $N < n < kN$ connues ou bien $n < N$, par exemple).
2. L'existence d'un algorithme probabiliste de brisure de symétrie : élection, coloration, exclusion mutuelle ou/et de consensus.