Maintenance et simulation de graphes aléatoires dynamiques

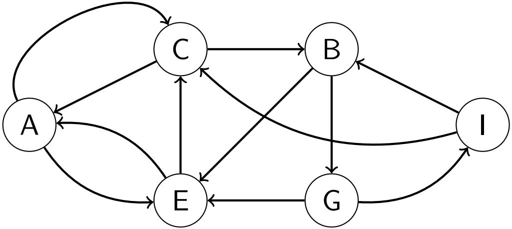
Romaric Duvignau

Soutenance de thèse

16 octobre 2015

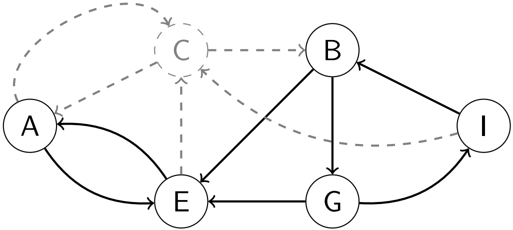
sous la direction de Philippe Duchon

Introduction : contexte et motivations 1/3



Contexte scientifique

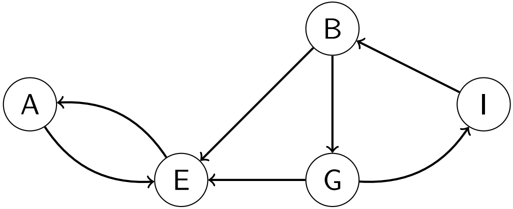
- Réseau de type « pair à pair » :
  - **logique** : liens virtuels
  - **décentralisé** : structure du réseau maintenue par des algorithmes locaux
  - **dynamique** : départs et arrivées de nœuds
- Structure dépendant de son historique de mises à jour



Contexte scientifique

- Réseau de type « pair à pair » :
  - **logique** : liens virtuels
  - **décentralisé** : structure du réseau maintenue par des algorithmes locaux
  - **dynamique** : départs et arrivées de nœuds
- Structure dépendant de son historique de mises à jour

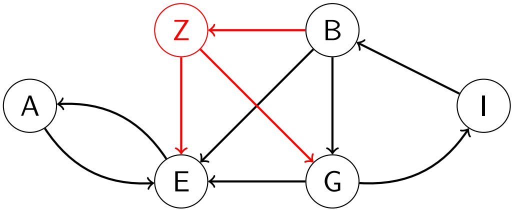
Introduction : contexte et motivations 1/3



Contexte scientifique

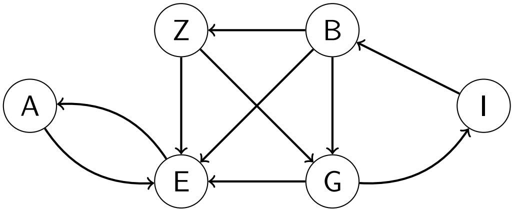
- Réseau de type « pair à pair » :
  - **logique** : liens virtuels
  - **décentralisé** : structure du réseau maintenue par des algorithmes locaux
  - **dynamique** : départs et arrivées de nœuds
- Structure dépendant de son historique de mises à jour

Introduction : contexte et motivations 1/3



Contexte scientifique

- Réseau de type « pair à pair » :
  - **logique** : liens virtuels
  - **décentralisé** : structure du réseau maintenue par des algorithmes locaux
  - **dynamique** : départs et arrivées de nœuds
- Structure dépendant de son historique de mises à jour



Contexte scientifique

- Réseau de type « pair à pair » :
  - **logique** : liens virtuels
  - **décentralisé** : structure du réseau maintenue par des algorithmes locaux
  - **dynamique** : départs et arrivées de nœuds
- Structure dépendant de son historique de mises à jour

Introduction : contexte et motivations 2/2

Évolution des réseaux P2P

- **Dans la littérature**, certaines *bonnes propriétés* du réseau sont maintenues, au prix de :
  - difficulté de conception des algorithmes de mise à jour
  - dynamicité modélisée par un processus probabiliste (Poisson) : **non réaliste** [Pouwelse *et al*, 2005]
  - analyse difficile sans ces hypothèses probabilistes

Bonnes propriétés

- Degré et diamètre faible, forte connectivité, etc

Évolution des réseaux P2P

- **Dans la littérature**, certaines *bonnes propriétés* du réseau sont maintenues, au prix de :
  - difficulté de conception des algo. de màj
  - dynamicité modélisée par un processus probabiliste (Poisson) : **non réaliste** [Pouwelse *et al*, 2005]
  - analyse difficile sans ces hypothèses probabilistes

Bonnes propriétés

- Degré et diamètre faible, forte connectivité, etc

Introduction : contexte et motivations 2/2

Évolution des réseaux P2P

- **Dans la littérature**, certaines *bonnes propriétés* du réseau sont maintenues, au prix de :
  - difficulté de conception des algo. de màj
  - dynamicité modélisée par un processus probabiliste (Poisson) : **non réaliste** [Pouwelse *et al*, 2005]
  - analyse difficile sans ces hypothèses probabilistes

- **Notre solution** : *préservation d'aléa*

Bonnes propriétés

- Degré et diamètre faible, forte connectivité, etc

Introduction : contexte et motivations 2/2

Évolution des réseaux P2P

- **Dans la littérature**, certaines *bonnes propriétés* du réseau sont maintenues, au prix de :
  - difficulté de conception des algo. de màj
  - dynamicité modélisée par un processus probabiliste (Poisson) : **non réaliste** [Pouwelse *et al*, 2005]
  - analyse difficile sans ces hypothèses probabilistes

- **Notre solution** : *préservation d'aléa*, répond à ces problématiques :
  - propriétés conservées
  - analyse simplifiée
  - n'est pas affectée par une séquence **arbitraire** de màj (potentiellement contrôlée par un adversaire)
  - pas de phénomènes de dérives

Bonnes propriétés

- Degré et diamètre faible, forte connectivité, etc

Introduction : contexte et motivations 3/3

Notre solution : *préservation d'aléa* [Knuth, 1977]

- Structures randomisées (Graphes aléatoires)
- **Distribution indépendante de la séquence de mises à jour**

Introduction : contexte et motivations 3/3

Notre solution : *préservation d'aléa* [Knuth, 1977]

- Structures randomisées (Graphes aléatoires)
- **Distribution indépendante de la séquence de mises à jour**

Dans la continuité de structures de données

- skip-list [Pugh, 1990]
- Treap [Aragon et Seidel, 1996]
- ABR randomisé [Martínez et Roura, 1998]
- Arbre k-d et quadtree randomisé [Duch *et al*, 1998-2005]

Introduction : contexte et motivations 3/3

Notre solution : *préservation d'aléa* [Knuth, 1977]

- Structures randomisées (Graphes aléatoires)
- **Distribution indépendante de la séquence de mises à jour**

Dans la continuité de structures de données

- skip-list [Pugh, 1990]
- Treap [Aragon et Seidel, 1996]
- ABR randomisé [Martínez et Roura, 1998]
- Arbre k-d et quadtree randomisé [Duch *et al*, 1998-2005]

Extension selon deux aspects : **Graphes** et **Distribué**.

Introduction : contexte et motivations 3/3

Notre solution : *préservation d'aléa* [Knuth, 1977]

- Structures randomisées (Graphes aléatoires)
- **Distribution indépendante de la séquence de mises à jour**

Dans la continuité de structures de données

- skip-list [Pugh, 1990]
- Treap [Aragon et Seidel, 1996]
- ABR randomisé [Martínez et Roura, 1998]
- Arbre k-d et quadtree randomisé [Duch *et al*, 1998-2005]

Extension selon deux aspects : **Graphes** et **Distribué**.

Notre objectif

- Décrire des algorithmes décentralisés qui maintiennent **exactement** une distribution souhaitée de graphes aléatoires

Plan

1. Maintenance de graphes aléatoires :
    - **Définition du modèle**
    - Exemple 1 : maintenance du modèle par paires
    - Exemple 2 : maintenance des graphes k-sortants
2. Sans connaître la taille du graphe :
    - Simulation en aveugle de lois binomiales
    - Maintenance sans accès à la taille

Problème de la *préservation* de loi

- Afin de préserver une famille de distributions de probabilité μ donnée, telle que μ_V est la distribution visée pour l'ensemble de sommets V, nous devons trouver :

  1. Un algorithme d'**insertion** de sommet $\mathcal{I}$:
     tel que pour tout $u \notin V$, si G est distribué selon μ_V alors
     $$\mathcal{I}(G, u) \sim \mu_{V+u}$$
  2. Un algorithme de **suppression** de sommet $\mathcal{D}$:
     tel que pour tout $u \in V$, si G est distribué selon μ_V alors
     $$\mathcal{D}(G, u) \sim \mu_{V-u}$$

Conservation d'aléa dans les graphes

Problème de la *préservation* de loi

- Afin de préserver une famille de distributions de probabilité μ donnée, telle que μ_V est la distribution visée pour l'ensemble de sommets V, nous devons trouver :

1. Un algorithme d'**insertion** de sommet $\mathcal{I}$:
   tel que pour tout $u \notin V$, si G est distribué selon μ_V alors
   $$\mathcal{I}(G, u) \sim \mu_{V+u}$$
2. Un algorithme de **suppression** de sommet $\mathcal{D}$:
   tel que pour tout $u \in V$, si G est distribué selon μ_V alors
   $$\mathcal{D}(G, u) \sim \mu_{V-u}$$

Attention, G est aléatoire mais u est arbitraire.

Maintenance de graphes aléatoires : notre modèle

Algorithmes de mise à jour

- Chaque nœud connaît son voisinage
- Chaque nœud peut appeler une primitive globale effectuant un tirage aléatoire uniforme sur l'ensemble des nœuds du graphe :

$$RandomVertex()$$

- Modèle **optimiste** de *premier contact*
- Deux sous-modèles :
  - Chaque nœud connaît la taille du graphe
  - La taille du graphe est inconnue

Maintenance de graphes aléatoires : notre modèle

Algorithmes de mise à jour

- Chaque nœud connaît son voisinage
- Chaque nœud peut appeler une primitive globale effectuant un tirage aléatoire uniforme sur l'ensemble des nœuds du graphe :

$$RandomVertex()$$

- Modèle **optimiste** de *premier contact*
- Deux sous-modèles :
  - Chaque nœud connaît la taille du graphe
  - La taille du graphe est inconnue

Coûts (asymptotiques)

- Espérance de la complexité temps (*sous-linéaire* souhaitée)
- Espérance du nombre d'appels à $RandomVertex()$

Exemple : maintenance des graphes d'Erdős–Rényi $G(n, p)$

En connaissant la taille du graphe au moment de la mise à jour

- Arrivée de u : ajout de u et de Binomiale(n, p) arêtes incidentes avec des sommets obtenus via `RandomVertex`
- Départ de u : suppression du sommet u et de ses arêtes incidentes

Exemple : maintenance des graphes d'Erdős–Rényi $G(n, p)$

En connaissant la taille du graphe au moment de la mise à jour

- Arrivée de u : ajout de u et de Binomiale(n, p) arêtes incidentes avec des sommets obtenus via `RandomVertex`
- Départ de u : suppression du sommet u et de ses arêtes incidentes

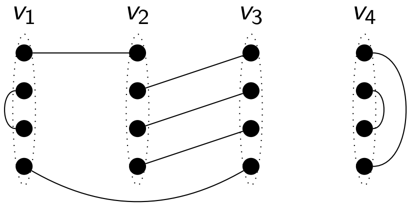
Coût (p constant)

- Arrivée de u : en moyenne, $\Theta(n)$ pas de calculs et appels à `RandomVertex`
- Départ de u : en moyenne, $\Theta(n)$ pas de calculs et aucun appels à `RandomVertex`

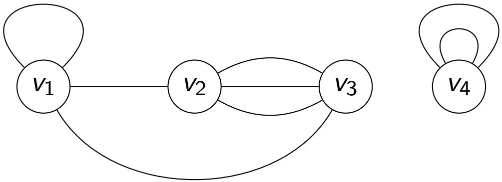
Plan

1. Maintenance de graphes aléatoires :
   - Définition du modèle
   - **Exemple 1 : maintenance du modèle par paires**
   - Exemple 2 : maintenance des graphes k-sortants
2. Sans connaître la taille du graphe :
   - Simulation en aveugle de lois binomiales
   - Maintenance sans accès à la taille

Maintenance du modèle par paires 1/2



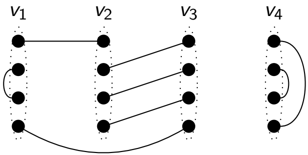
(a) Couplage parfait M



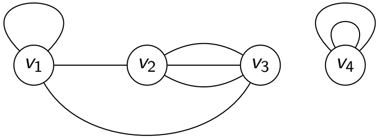
(b) Multigraphe $P(M)$ de degré $k = 4$

Maintenance du modèle par paires 2/2

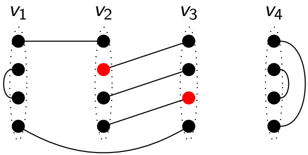
(a) Couplage parfait M



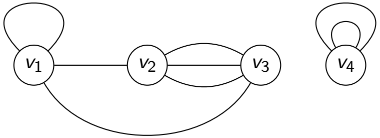
(b) Multigraphe $P(M)$ de degré $k = 4$



(a) Couplage parfait M



(b) Multigraphe $P(M)$ de degré $k = 4$

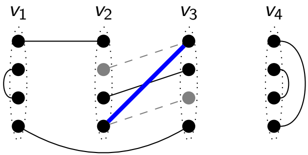


Multigraphe de paires de degré k pair

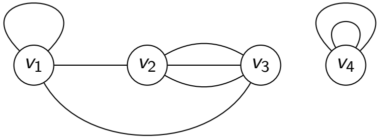
- Suppression : $k/2$ suppressions sur les couplages parfaits (ordre uniforme)

Maintenance du modèle par paires 2/2

(a) Couplage parfait M



(b) Multigraphe $P(M)$ de degré $k = 4$

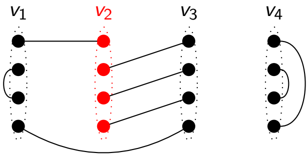


Multigraphe de paires de degré k pair

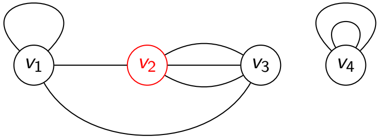
- Suppression : $k/2$ suppressions sur les couplages parfaits (ordre uniforme)

Maintenance du modèle par paires 2/2

(a) Couplage parfait M



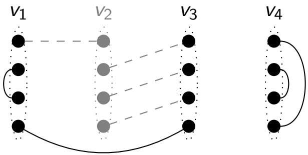
(b) Multigraphe $P(M)$ de degré $k = 4$



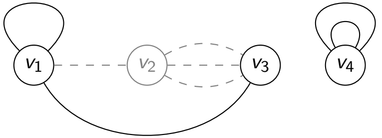
Multigraphe de paires de degré k pair

- Suppression : $k/2$ suppressions sur les couplages parfaits (ordre uniforme)

(a) Couplage parfait M



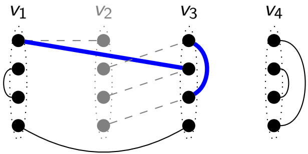
(b) Multigraphe $P(M)$ de degré $k = 4$



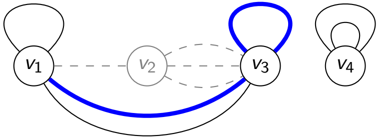
Multigraphe de paires de degré k pair

- Suppression : $k/2$ suppressions sur les couplages parfaits (ordre uniforme)

(a) Couplage parfait M



(b) Multigraphe $P(M)$ de degré $k = 4$

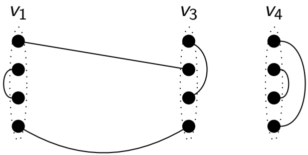


Multigraphe de paires de degré k pair

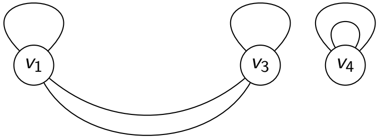
- Suppression : $k/2$ suppressions sur les couplages parfaits (ordre uniforme)

Maintenance du modèle par paires 2/2

(a) Couplage parfait M



(b) Multigraphe $P(M)$ de degré $k = 4$

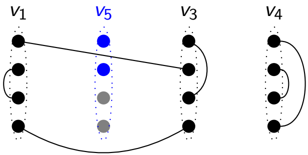


Multigraphe de paires de degré k pair

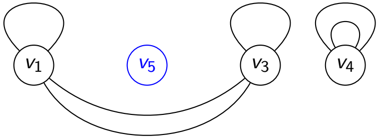
- Suppression : $k/2$ suppressions sur les couplages parfaits (ordre uniforme)

Maintenance du modèle par paires 2/2

(a) Couplage parfait M



(b) Multigraphe $P(M)$ de degré $k = 4$

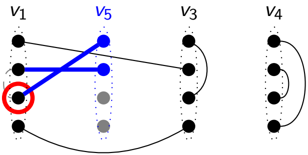


Multigraphe de paires de degré k pair

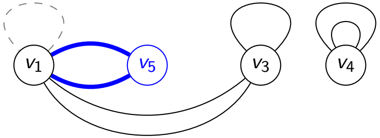
- Suppression : $k/2$ suppressions sur les couplages parfaits (ordre uniforme)
- Insertion : $k/2$ insertions sur les couplages parfaits (nécessite en moyenne $k/2 + \mathcal{O}(1/n)$ appels à $\mathtt{RV}$)

Maintenance du modèle par paires 2/2

(a) Couplage parfait M



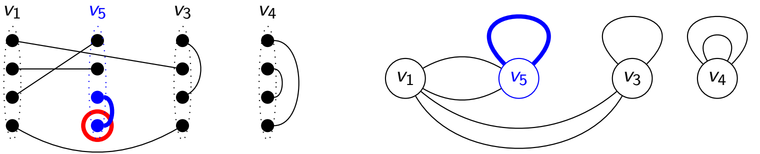
(b) Multigraphe $P(M)$ de degré $k = 4$



Multigraphe de paires de degré k pair

- Suppression : $k/2$ suppressions sur les couplages parfaits (ordre uniforme)
- Insertion : $k/2$ insertions sur les couplages parfaits (nécessite en moyenne $k/2 + \mathcal{O}(1/n)$ appels à RV)

(a) Couplage parfait M (b) Multigraphe $P(M)$ de degré $k = 4$

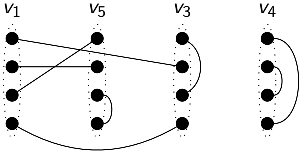


Multigraphe de paires de degré k pair

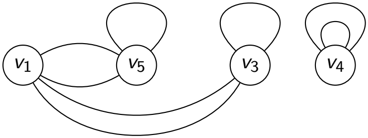
- Suppression : $k/2$ suppressions sur les couplages parfaits (ordre uniforme)
- Insertion : $k/2$ insertions sur les couplages parfaits (nécessite en moyenne $k/2 + \mathcal{O}(1/n)$ appels à $\texttt{RV}$)

Maintenance du modèle par paires 2/2

(a) Couplage parfait M



(b) Multigraphe $P(M)$ de degré $k = 4$



Multigraphe de paires de degré k pair

- Suppression : $k/2$ suppressions sur les couplages parfaits (ordre uniforme)
- Insertion : $k/2$ insertions sur les couplages parfaits (nécessite en moyenne $k/2 + \mathcal{O}(1/n)$ appels à $\mathtt{RV}$)

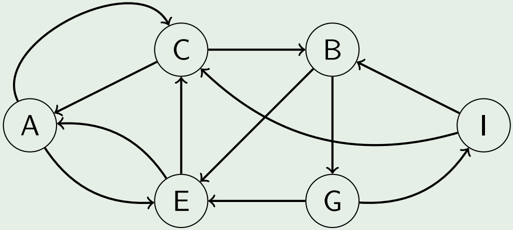
Ces algorithmes n'ont pas besoin de connaître la taille du graphe.

Plan

1. Maintenance de graphes aléatoires :
   - Définition du modèle
   - Exemple 1 : maintenance du modèle par paires
   - **Exemple 2 : maintenance des graphes k-sortants**
2. Sans connaître la taille du graphe :
   - Simulation en aveugle de lois binomiales
   - Maintenance sans accès à la taille

Maintenance des graphes *k*-sortants

Exemple de graphe 2-sortant

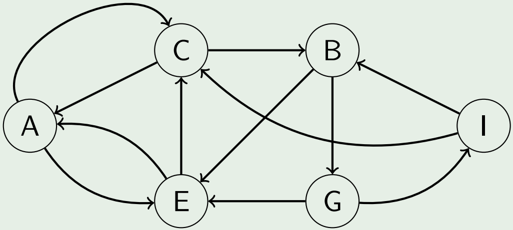


Graphes *k*-sortants uniformes

- Graphes dirigés sans boucles
- Chaque nœud possède k voisins sortants
- Loi uniforme associée à de bonnes propriétés de type réseau P2P (degré et diamètre faible, forte connectivité)

Maintenance des graphes k-sortants : suppression 1/2

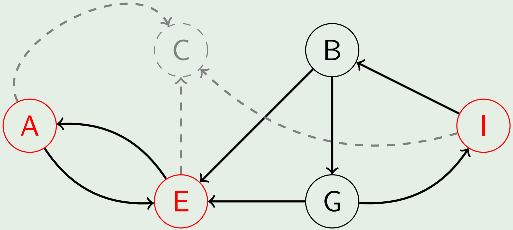
Suppression du sommet C



Le nœud C souhaite quitter le réseau.

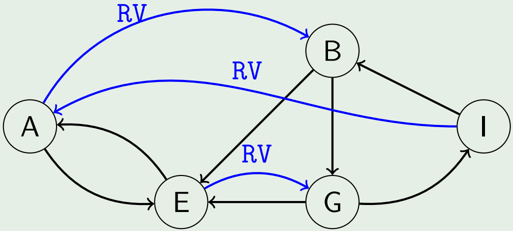
Maintenance des graphes k-sortants : suppression 1/2

Suppression du sommet C



Le nœud C quitte le réseau, et laisse 3 arêtes pendantes.

Suppression du sommet C



Les nœuds A, E et I trouvent un remplaçant à C en appelant `RandomVertex()`.

Au total, nous avons besoin de $k + \mathcal{O}(1/n)$ appels en moyenne.

Maintenance des graphes k-sortants : suppression 2/2

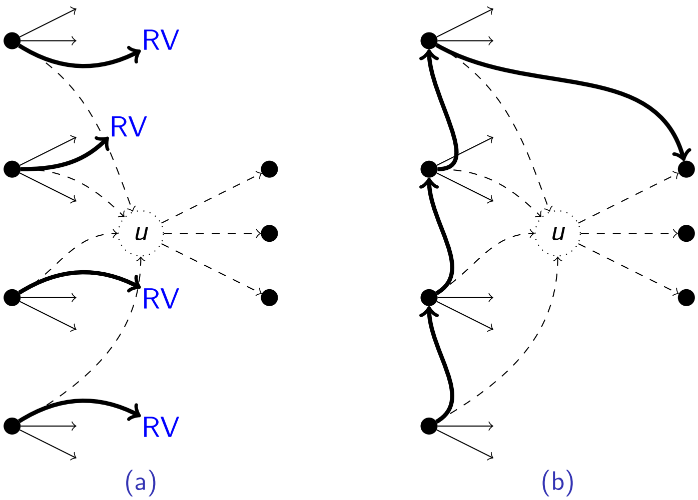
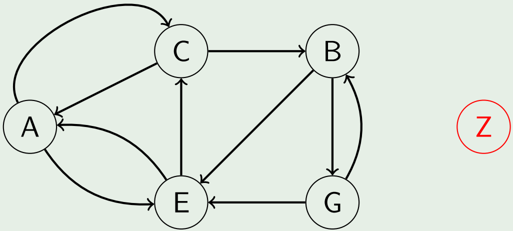


Figure : Exemples de suppression de u *typiques* des deux algorithmes.

Maintenance des graphes k-sortants : insertion 1/2

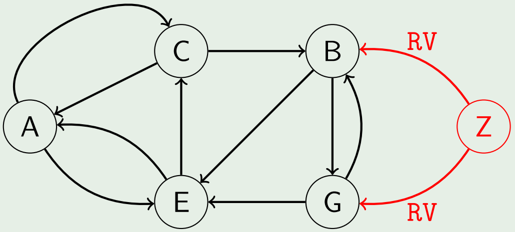
Insertion du sommet Z



Le nœud Z souhaite rejoindre le réseau.

Maintenance des graphes k-sortants : insertion 1/2

Insertion du sommet Z

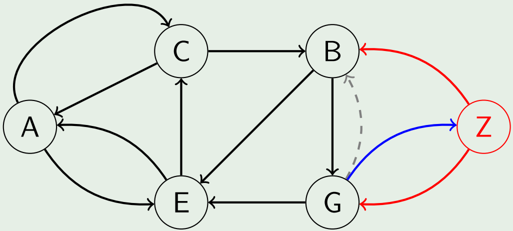


Le nœud Z choisit 2 sommets distincts comme successeurs, en utilisant `RandomVertex()`.

En moyenne, $k + \mathcal{O}(1/n)$ appels à la primitive seront effectués.

Maintenance des graphes k-sortants : insertion 1/2

Insertion du sommet Z



Le nœud Z choisit $X \sim$ Binomiale$(n, k/n)$ sommets distincts comme prédécesseurs, et *vole* une arête de chacun d'entre eux. Nous avons besoin de $k + \mathcal{O}(1/n)$ appels supplémentaires en moyenne pour choisir les prédécesseurs.

Maintenance des graphes k-sortants : insertion 2/2

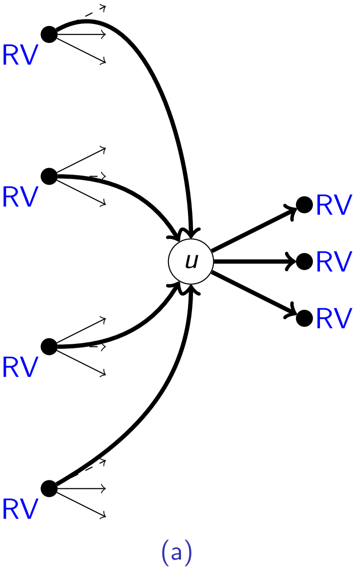


Figure : Exemples d'insertion de u *typiques* des deux algorithmes.

Maintenance des graphes k-sortants : bilan

Théorème (Duchon, D., 14)

Il existe des algorithmes de mise à jour pour les graphes k-sortants :

- *modifiant un nombre minimal d'arêtes,*
- *de complexité moyenne constante, et utilisant en moyenne :*
  - $k + \mathcal{O}(1/n)$ *appels à* $\mathtt{RV}$ *pour l'insertion ;*
  - $\mathcal{O}(1/n)$ *appels à* $\mathtt{RV}$ *pour la suppression.*
- **et ces bornes sont asymptotiquement optimales.**

Maintenance des graphes k-sortants : bilan

Théorème (Duchon, D., 14)

Il existe des algorithmes de mise à jour pour les graphes k-sortants :

- *modifiant un nombre minimal d'arêtes,*
- *de complexité moyenne constante, et utilisant en moyenne :*
  - $k + \mathcal{O}(1/n)$ *appels à* $\mathtt{RV}$ *pour l'insertion ;*
  - $\mathcal{O}(1/n)$ *appels à* $\mathtt{RV}$ *pour la suppression.*
- **et ces bornes sont asymptotiquement optimales.**

Tous les algorithmes d'insertion nécessitent de connaître la taille exacte du graphe au moment de la mise à jour afin de simuler la loi binomiale Binomiale(n, k/n).

Plan

1. Maintenance de graphes aléatoires :
   - Définition du modèle
   - Exemple 1 : maintenance du modèle par paires
   - Exemple 2 : maintenance des graphes k-sortants
2. Sans connaître la taille du graphe :
   - **Simulation en aveugle de lois binomiales**
   - Maintenance sans accès à la taille

Simulation en aveugle de lois binomiales 1/3

Algorithm 1 Algorithme-premier-doublon pour Bin$(n, 1/n)$

$S \leftarrow \emptyset$
loop
    $x \leftarrow$ RandomVertex()
    if $x \in S$ **then**
        return nbFixedPoint(randomPermutation(S))
    end if
    $S \leftarrow S + x$
end loop

Simulation en aveugle de lois binomiales 1/3

Algorithm 1 Algorithme-premier-doublon pour Bin$(n, 1/n)$

$S \leftarrow \emptyset$
loop
  $x \leftarrow$ RandomVertex()
  if $x \in S$ **then**
    return nbFixedPoint(randomPermutation(S))
  end if
  $S \leftarrow S + x$
end loop

L'algorithme ci-dessus utilise en moyenne $\Theta(\sqrt{n})$ tirages pour simuler le cas $k = 1$.

Simulation en aveugle de lois binomiales 1/3

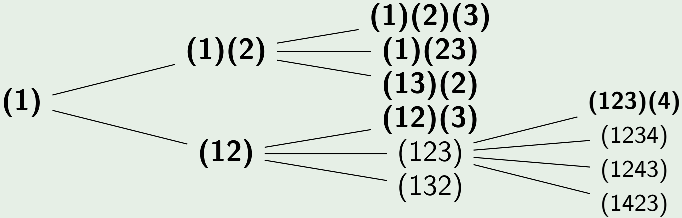
Algorithm 1 Algorithme-premier-doublon pour $\mathrm{Bin}(n, 1/n)$

$S \leftarrow \emptyset$
loop
  $x \leftarrow$ RandomVertex()
  if $x \in S$ **then**
    return nbFixedPoint(randomPermutation(S))
  end if
  $S \leftarrow S + x$
end loop

L'algorithme ci-dessus utilise en moyenne $\Theta(\sqrt{n})$ tirages pour simuler le cas $k = 1$.

Peut-on réaliser cette simulation en moins de $\Theta(\sqrt{n})$ tirages en moyenne ?

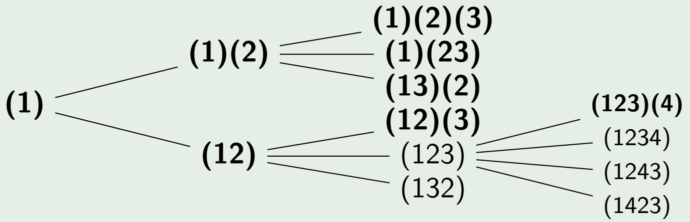
Exemple d'arbre de génération



- Arbre infini de permutations où les nœuds au niveau n possèdent $n+1$ enfants
- Un tel arbre fournit un "algorithme" de génération pour les permutations uniformes de taille n

Simulation en aveugle de lois binomiales 2/3

Exemple d'arbre de génération

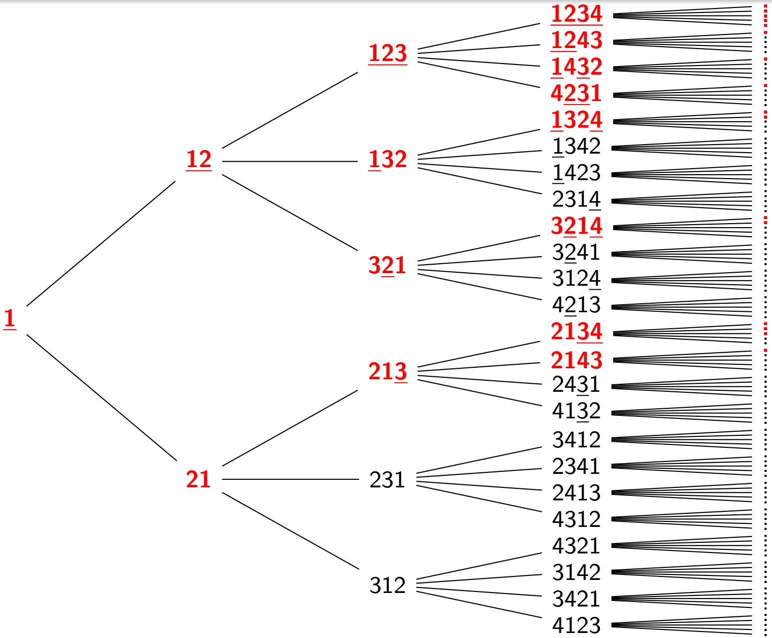


- Arbre infini de permutations où les nœuds au niveau n possèdent $n+1$ enfants
- Un tel arbre fournit un “algorithme” de génération pour les permutations uniformes de taille n

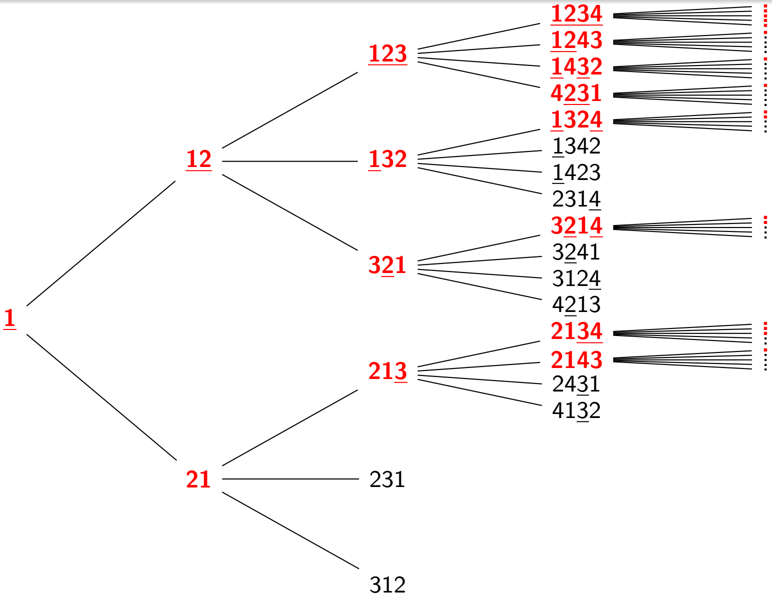
Propriété essentielle de notre arbre (Duchon, D. 14)

Toute descente infinie atteint avec probabilité 1 une permutation dont tous les descendants ont le même nombre de points fixes.

Simulation en aveugle de lois binomiales 3/3



Simulation en aveugle de lois binomiales 3/3



Simulation en aveugle de lois binomiales 3/3

Algorithme du premier doublon amélioré

- Grâce à notre arbre : il est possible de tirer le nombre de points fixes d'une permutation de taille $|V|$ en moins de 3,2 appels à `RV` en moyenne
- Effectue en parallèle des tirages sur V et une descente dans notre arbre de génération

Théorème (Duchon, D. 14, D. 15+)

Il est possible de simuler Binomiale$(n, k/n)$ sans connaître $n = |V|$ en $5{,}2k + \mathcal{O}(1/n)$ appels à `RV` *en moyenne.*

Plan

1. Maintenance de graphes aléatoires :
    - Définition du modèle
    - Exemple 1 : maintenance du modèle par paires
    - Exemple 2 : maintenance des graphes k-sortants
2. Sans connaître la taille du graphe :
    - Simulation en aveugle de lois binomiales
    - **Maintenance sans accès à la taille**

Maintenance sans accès à la taille

Maintenabilité des distributions abordées

- Maintenance dans le modèle par paires : n'utilise pas la taille
- Maintenance des graphes k-sortants : simulation de loi Binomiale$(n, k/n)$ sans utiliser n

Maintenance sans accès à la taille

Maintenabilité des distributions abordées

- Maintenance dans le modèle par paires : n'utilise pas la taille
- Maintenance des graphes k-sortants : simulation de loi Binomiale$(n, k/n)$ sans utiliser n

Théorème (D. 15+)

Il est impossible de maintenir la distribution des graphes d'Erdős–Rényi (p fixe) sans connaître la taille du graphe.

Maintenance sans accès à la taille

Maintenabilité des distributions abordées

- Maintenance dans le modèle par paires : n'utilise pas la taille
- Maintenance des graphes k-sortants : simulation de loi Binomiale$(n, k/n)$ sans utiliser n

Théorème (D. 15+)

Il est impossible de maintenir la distribution des graphes d'Erdős–Rényi (p fixe) sans connaître la taille du graphe.

Nous pouvons donc déduire que Binomiale(n, p) n'est pas simulable sans connaître la taille du graphe. Peut-on caractériser les lois simulables ?

Maintenance sans accès à la taille : simulabilité de lois

Simulabilité

- Étude de la simulabilité de lois dépendants d'un paramètre n, sans connaître n, mais en ayant accès à un générateur d'uniformes sur un ensemble de taille n à la place.

Maintenance sans accès à la taille : simulabilité de lois

Simulabilité

- Étude de la simulabilité de lois dépendants d'un paramètre n, sans connaître n, mais en ayant accès à un générateur d'uniformes sur un ensemble de taille n à la place.

Théorème (D. 15+)

Une famille $\rho = (\rho_n)_{n \geq a}$ de distributions de probabilité est simulable, si et seulement si,

- $\mathit{Supp}(\rho_n) \subset \mathit{Supp}(\rho_{n+1})$ *pour tout* $n \geq 1$
- *il existe une fonction calculable* $f : \mathit{Supp}(\rho) \to \mathbb{N}$, *telle que*

$$\rho_n(k) \geq 1/n^{f(k)}$$

pour tout $k \in \mathit{Supp}(\rho)$, *et pour tout* $n \geq 2$, *de sorte que* $k \in \mathit{Supp}(\rho_n)$.

Conclusion

Bilan

- Définition précise de conservation d'aléa dans les graphes de manière décentralisée
- Préservation de loi efficace sur plusieurs modèles intéressants
- Étude de l'influence de la connaissance de la taille

Conclusion

Bilan

- Définition précise de conservation d'aléa dans les graphes de manière décentralisée
- Préservation de loi efficace sur plusieurs modèles intéressants
- Étude de l'influence de la connaissance de la taille

Perspectives

- Algorithmes de mise à jour sous-linéaires pour le graphe des paires et un modèle par attachement préférentiel
- Graphes géométriques (ou d'autres lois non invariantes par renommage)
- Liens entre maintenabilité et simulabilité de loi (degrés, ...)

Merci

Merci pour votre attention.