Génération aléatoire

Nicolas Hanusse
Chargé de Recherche CNRS - LaBRI

Origine de la génération aléatoire

- Wilf (1960-70): simulation des trajectoires de particules à l'intérieur de réacteurs nucléaires;
- Wilf-Nijenhuis: génération *uniforme* ou *équiprobable*
- Problème:
  - Soit E un ensemble d'objets (combinatoire) et n un entier
  - E_n : éléments de E de taille n
  - Comment engendrer un ou plusieurs éléments de E_n de sorte que tous aient même probabilité d'apparaître.

Pourquoi générer des objets aléatoires

- Validation de modèles:
  - Comparaison données réelles et données aléatoires engendrées par des modèles
- Connaissance profonde des propriétés des structures aléatoires
  - Autres outils: visualisation de paramètres, mesure (théorique et statistique)
- Génération de jeux de données:
  - Pour tester et comparer des algorithmes
- Estimation de quantité: si le dénombrement de E_n est inconnu, la génération peut aider à estimer E_n.

Complexité – Temps et Mémoire

- On veut générer beaucoup d'objets de grande taille n
  - génération en temps $\theta(n^2)$ un peu élevé
  - génération en temps $\theta(n^3)$ inacceptable
- Deux modèles de complexité:
  - Arithmétique (nombres bornés):
    - opérations (+,*, …) en temps constant;
    - Codage d'un nombre en espace constant.
  - Logarithmique:
    - opérations (+,*, …) en temps $\theta(\log k)$;
    - Codage d'un nombre k en espace $\theta(\log k)$.

Différentes approches

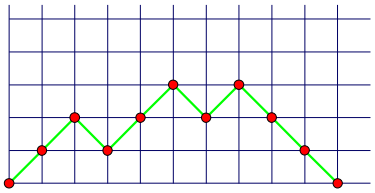
- Approche récursive;
- Méthodes à rejet;
- Chaîne de Markov;
- Outils complémentaires:
  - Rang inverse: codage et bijection

Rang Inverse

- Principe:
  - On connaît une bijection entre objets et un ensemble d'entiers;
  - On tire de manière aléatoire un entier (le rang) et on construit l'objet correspondant.
- Exemple:
  - Bijection arbre plan – mot de Dyck – entier

Arbres = Mots de Dyck

Parenthésages « bien formés »

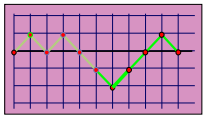


D_{2n}

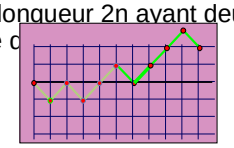
a a b a a b a b b b

Nombres de Catalan $C_n = \binom{2n}{n} - \binom{2n}{n-1}$

Mots de longueur 2n ayant autant de a que de b



- Mots de longueur 2n ayant deux a de plus que d



Le Rang des mots de Dyck

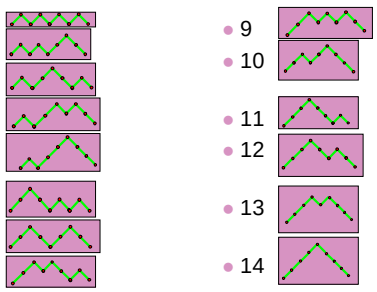
- Ordre Lexicographique
  - a b a b a b
  - a b a a b b
  - a a b b a b
  - a a b a b b
  - a a a b b b

Ordre lexicographique

- Soient deux mots de Dyck
  $v=v_1v_2\ldots v_{2n}$ et $w=w_1w_2\ldots w_{2n}$
- $V < W$ lorsqu 'il existe k tel que
  - $v_i=w_i$ pour $i=1,2,\ldots,k-1$
  - $v_k=b$
  - $w_k=a$
- a a b b a b a b < a a b b a a b b

Ordre Lexicographique

- Mots de Dyck de longueur 8
  - 1
  - 2
  - 3
  - 4
  - 5
  - 6
  - 7
  - 8
  - 9
  - 10
  - 11
  - 12
  - 13
  - 14



Calcul du rang

- Soit un mot de Dyck $v=v_1v_2\ldots v_{2n}$
- Pour i tel que $v_i = a$, on définit
  $P(i) = \{w \in D_{2n}, w=v_1v_2\ldots v_{i-1}\ b\ w_{i+1}\ldots w_{2n}\}$
- $\bigcup_i P(i) = \{w \in D_{2n}$ tels que $w < v\}$
- P(i) disjoints

$$\text{Rang}(v)=1+\sum_{i/v_i=a} |P(i)|$$

Rang(v)=1+$\sum$ #P(i)

v= a a a b b a b b

1 2 3 6

- P(1)={b …} ⇒ #P(1)=0
- P(2)={a b…} ⇒ #P(2)=5
- P(3)={a a b…} ⇒ #P(3)=5
- P(6)={a a a b b b…} ⇒ #P(6)=1

Rang(a a a b b a b b)=1+0+5+5+1=12

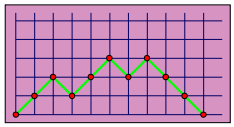
Proposition #P(i) = $\binom{2n-1}{f(i)} - \binom{2n-1}{f(i)-1}$

- où f(i) = $|v_i v_{i+1} \ldots v_{2n-1}|_a$
- On en déduit la procédure

```
ranginverse(int h){
   int f,i,P;
    f=n;
   for(int i=1;i<=2n;i++){
      P=binomial(2n-i,f)- binomial(2n-i,f-1)
       if(h>P) then
          v_i=a; f=f-1;h=h-P;
       else v_i=b;
   }
}
```

Génération aléatoire d 'un mot de Dyck de longueur 2n

- 1 —— h —— C_n
- Rang-inverse(h)
- $v=v_1 v_2 \ldots v_{2n}$



Complexité du rang-inverse

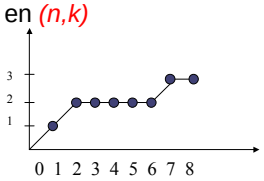
- Tirage d'un nombre aléatoire:
  - *O(1)* ou *O(log n)* en moyenne
- Décodage: Application du rang inverse
  - Linéaire
- Bilan: espace et temps linéaire.

L'approche récursive

- Principe:
  - On construit un élément de E_i en fonction d'un élément de E_{i-1}
- Exemple:
  - Comment construire un groupe aléatoire de *k* personnes parmi n personnes ?
  - $E_{n,k}$: ensemble de sous-ensembles de taille *k* d'un ensemble de taille *k*

Interprétation en terme de chemins

- Un élément de $E_{n,k}$ peut être vu comme un chemin $C_{n,k}$ dans le plan commençant au point *(0,0)* et finissant en *(n,k)*



$e=\{1,2,7\}$

$C_{8,3}$

Algorithme récursif

- Technique du pas à pas
  - Coordonnée courante = *(i,j)*
  - On fait un pas *s0=(+1,+1)* avec proba *p0=(k-j)/(n-i)*
  - On fait un pas *s1=(+1,0)* avec proba *p1=1-p0*
- Complexité linéaire *O(n)*
- Algorithme naïf (mauvais pour k et n grand):
  - On tire *k* fois un rang
  - Complexité *O(k log n)* pour n grand

Exemple d'un algorithme naïf de génération non uniforme

- A chaque pas:
  - Si i<k, on génère un pas s0 avec probabilité ½ sinon pas suivant est s1.
  - Exemple (n=4,k=2):
    - S0,s0,s1,s1 a probabilité ½*½*1*1=1/4
    - S0,s1,s0,s1 a probabilité ½*½*½*1=1/8
- En résumé:
  - Vérifier que tout élément peut être engendré et de manière équiprobable.

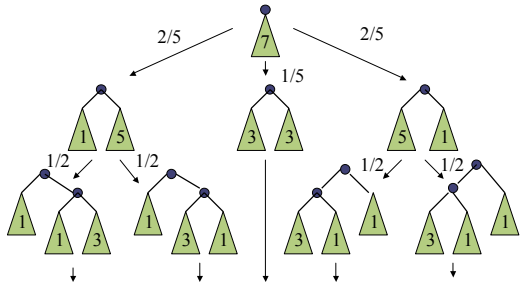
Structures décomposables - Spécifications

- On part d'une description non ambiguë
- Les probabilités se calculent en fonction des ensembles mis en jeu
- Objets primitifs:
  - Objet vide de taille 0, noté 1
  - Atomes de taille 1, noté Z
  - Opérateurs:
    - Union disjointe +
    - Produit (non commutatif) *
    - La séquence seq(A), l'ensemble set(A), le cycle cyc(A)

Exemples de spécifications de structures décomposables

- Sous-ensembles de cardinalité k:
  - A ← 1 + s0*A + s1*A
- Arbres binaires complets:
  - A ← 1 + A*A
- …

Arbres binaires complets

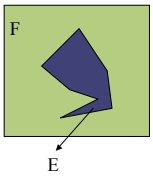


Bilan de le méthode décomposition

- Structures décomposables non ambiguë
  - Difficile à engendrer selon plusieurs paramètres
- Temps:
  - Calcul de E_i pour tout $i < n+1$: $O(n^2)$ pour 1 paramètre.
  - Puis temps génération en temps $O(n)$
- Mémoire:
  - tables des $E_i = O(n^2)$
  - Codage linéaire: chemin dans l'arbre de génération.

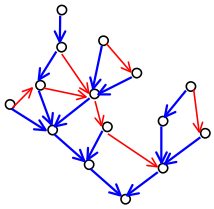
Méthode à rejet

- Principe:
  - Tirage uniforme de x dans F;
  - E inclus dans F;
  - Si $x \notin E$, on retire x.
  - Obtention pseudo-algorithme: complexité moyenne en O(|E|/|F|).
- Exemple:
  - graphe connexe étiqueté à n sommets;
  - Tirage dans $G_{n,1/2}$
  - Avec proba 1-o(1), G est connexe



Carte planaire extérieure

- *Planaire extérieure*: tous les sommets bordent la face extérieure;
- = un arbre + une arête supplémentaire par sommet menant au premier sommet de la branche suivante.



T = ((((((()))(()))))(((()))))

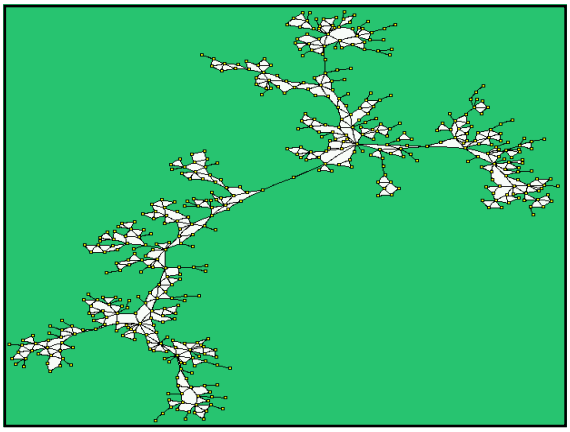
R = 000111011010010

Génération de carte planaire extérieure

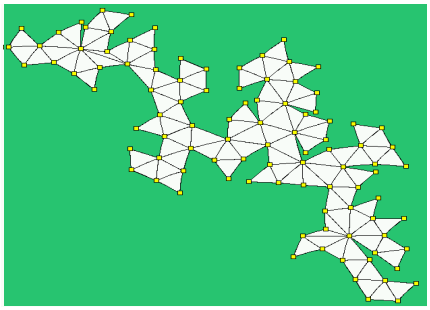
CartePlanaireExt(*n*)
1. Génération d'un arbre T de *n* sommets;
2. Chaque sommet de T (excepté les extrémités de la dernière branche) est coloré en rouge avec probabilité ½ et en blanc sinon;
3. Si tous les sommets de la dernière branche sont blanches, garder T sinon relancer CartePlanaireExt.

Théorème (Bonichon, Gavoille, Hanusse 2003): Une carte planaire extèrieure connexe de n sommets et m arêtes peut être généré de manière aléatoire uniforme en temps moyen O(n).

Pourquoi: Proba(dernière branche de longueur 2) > 1/4



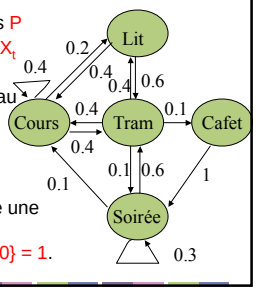
Carte planaire extérieure maximale aléatoire à 100 sommets



Chaînes de Markov et Marche aléatoire

Principe:

- états X et matrice de transitions P
- CM = séquence de v.a. X_0, X_1..X_t
- Prob(X_{i+1} = y | X_i = x) = P(x,y)
- On suit une arête (x,y) choisie au hasard avec probabilité P(x,y).



CM est *ergodique* si elle est:

- –*Irréductible* (connexité): il existe une chaîne de x à y pour tout x,y.
- –*Apériodique*: PGCD {t: P^t (x,y)>0} = 1.

Convergence CM ergodique vers une distribution stationnaire

- Théorème: Si CM est ergodique, elle converge vers une unique distribution stationnaire $\pi = \pi P$.
- La distribution stationnaire vérifie pour tout état initial x:
  - $\lim P^n(x,y) = \pi(y)$ *quand n tend vers l'infini.*
- Si la matrice de transition est symétrique, pour tout état y:
  - $\pi(y) = 1/|X|$ *(distribution uniforme)*
- Temps de mélange T: temps requis pour la distribution soit proche de la distribution stationnaire à epsilon prés.
  - $|Prob(X_T = x) - \pi(x)| < \varepsilon\, \pi(x)$

Application directe à la génération aléatoire

- Si la distribution stationnaire est uniforme
- Si le temps de mélange T est petit (*mélange rapide):*
  - $poly(n, \log(1/\varepsilon))$
- Difficulté: Borner le temps de mélange !!!
- Méthode simple mais complexe à étudier.

Exemple: Génération de graphes planaires

- X = ensemble des graphes planaires étiquetés à n sommets
- On met une transition d'un graphe x à un graphe y si y est obtenu en rajoutant ou enlevant une arête à x
- Algorithme (variante de Denise, Vasconcellos, Welsh 1996):
  - On tire un couple de sommets (i,j) dans x:
  - Avec proba ½, j'enlève (i,j) et sinon j'essaye de rajouter (i,j) via test de la préservation de la planarité.

Conjectures de Denise et al.

- Distribution stationnaire uniforme
  - Matrice P symétrique: si x et y sont voisins dans X, alors P(x,y)=P(y,x)=2/n(n-1)
- Conjectures (basés sur des expériences):
  - Presque tous les graphes planaires sont connexes;
  - Presque tous les graphes planaires ne sont pas 2-connexes;
  - Le nombre moyen d'arêtes est 2n (FAUX)
  - Hypothèse (FAUSSE): temps de mélange $T < 2n^2$

Bilan sur la technique « chaîne de Markov »

- Difficulté: borner le temps de mélange
  - Conductance, calcul de valeurs propres, …
- Obtention de tirages *presque uniforme* (et parfois uniforme)
- On peut *transformer* une chaîne dont la distribution stationnaire n'est pas uniforme en chaîne de distribution uniforme (Métropolis)