

ALGO2 – Chaînes de Markov

François Schwarzentruher

8 mai 2023

1 Motivation

1.1 Génération aléatoire

Choisir un crayon parmi 5 crayons, c'est facile. Là, on s'intéresse à générer des objets parmi un ensemble de taille exponentielle. Mais choisir un élément de $\{0, 1\}^n$ c'est facile. Là, en plus, les objets sont contraints.

Génération uniforme d'ensembles indépendants

entrée : G graphe non orienté

sortie : un ensemble indépendant dans G , tiré aléatoirement de manière uniforme dans l'ensemble des ensembles indépendants

Génération uniforme d'une q -coloration d'un graphe

entrée : G graphe non orienté

sortie : une q -coloration de G , tirée aléatoirement de manière uniforme dans l'ensemble des q -colorations

↪ Simulation d'une chaîne de Markov

1.2 Problème d'optimisation

Ensemble indépendant maximal

entrée : G graphe non orienté

sortie : un ensemble indépendant maximal dans G

Voyageur de commerce

entrée : G graphe complet pondéré

sortie : un tour minimal dans G

↪ Recuit simulé

1.3 Problème de comptage

Compter le nombre d'ensembles indépendants

entrée : G graphe non orienté

sortie : le nombre d'ensembles indépendants

Compter le nombre de q -coloration d'un graphe

entrée : G graphe non orienté

sortie : le nombre de q -colorations de G

↪ lire le chapitre 9 de [H⁺02]

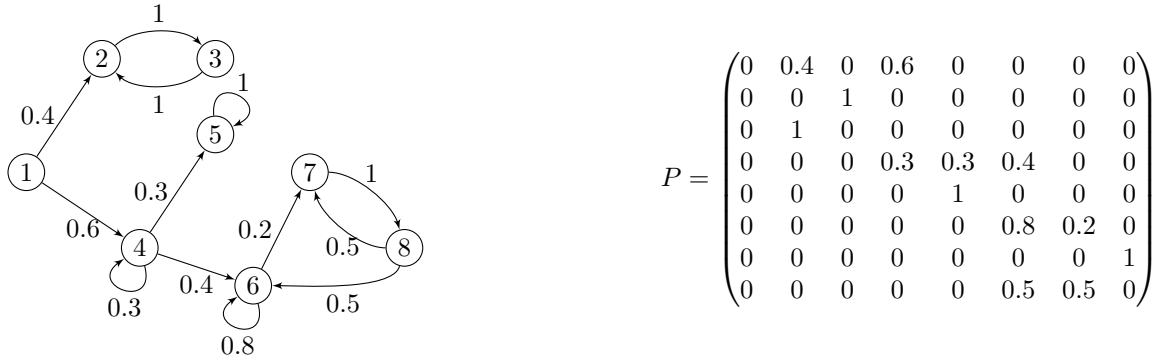
2 Chaîne de Markov

Définition 1 Une **chaîne de Markov** est une suite de variables aléatoires $\mathcal{M} = (X_t)_{t \in \mathbb{N}}$ à valeur dans \mathcal{S} telles que pour toute suite $s_0 \cdots s_t, s_{t+1} \in \mathcal{S}$, $\mathbb{P}(X_{t+1} = s_{t+1} \mid X_t = s_t \cdots X_0 = s_0) = \mathbb{P}(X_{t+1} = s_{t+1} \mid X_t = s_t)$.

Définition 2 On dit qu'une chaîne de Markov est **homogène**, si pour tout $(s, s') \in \mathcal{S}$ la valeur $\mathbb{P}(X_{t+1} = s' \mid X_t = s)$ est indépendante de t , c'est-à-dire pour tout $t, t' \in \mathbb{N}$, $\mathbb{P}(X_{t+1} = s' \mid X_t = s) = \mathbb{P}(X_{t'+1} = s' \mid X_{t'} = s)$.

Si \mathcal{S} est fini (voire dénombrable), une chaîne de Markov homogène \mathcal{M} se représente par la **distribution initiale** μ_0 (celle de X_0) et un **graphe** qui décrit l'évolution d'une étape. Les sommets du graphe sont les éléments de \mathcal{S} , et pour tout $(s, s') \in \mathcal{S}$, si $P_{s,s'} = \mathbb{P}(X_{t+1} = s' \mid X_t = s) > 0$ il y a une arête entre s et s' , pondérée par $P_{s,s'}$.

Exemple 3

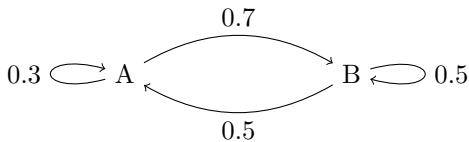


Dans la matrice P , la ligne pour l'état s donne les probabilités d'être dans un état s' quand on part de l'état s .

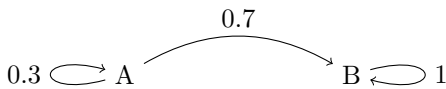
3 Propriétés

Définition 4 $\mathcal{M} = (X_k)_{k \in \mathbb{N}}$ est **irréductible** si pour toute paire d'états $s, t \in \mathcal{S}$, il y a un chemin de s de t , ou de façon équivalente s'il existe $k \in \mathbb{N}$ tel que $(P^k)_{s,t} > 0$.

Exemple 5 (chaîne irréductible)

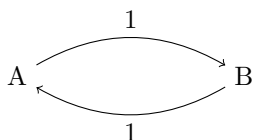


Exemple 6 (chaîne réductible)



Définition 7 \mathcal{M} est **apériodique** si pour tout état $s \in \mathcal{S}$, le pgcd de la longueur des cycles autour de s est 1 : $\text{pgcd}\{n \geq 1 \mid (P^n)_{s,s} > 0\} = 1$.

Exemple 8 (chaîne périodique)



4 Convergence

Définition 9 Soit $\mu \in \text{Dist}(\mathcal{S})$ une distribution.

- μ est **réversible** si pour tous états $s, s' \in \mathcal{S}$, $\mu(s) \times P_{s,s'} = \mu(s') \times P_{s',s}$;
- μ est **stationnaire** si $\mu P = \mu$.

Proposition 10 Si μ est réversible, alors μ est stationnaire.

Théorème 11 (Convergence des chaînes de Markov) Soit $\mathcal{M} = (\mathcal{S}, \mu_0, P)$ une chaîne de Markov irréductible apériodique, et μ une distribution stationnaire. Alors $\lim_{n \rightarrow \infty} |(\mu_0 P^n) - \mu| = 0$.

Corollaire 12 Si \mathcal{M} est **irréductible apériodique**, alors il existe une **unique distribution stationnaire**.

5 Application : génération uniforme d'objets

1. On définit une chaîne de Markov \mathcal{M} où chaque état est un objet possible (e.g. un ensemble indépendant)
2. On se balade dans \mathcal{M} .
3. On espère que X_{100000} est un objet aléatoire qui suit à peu près la distribution que l'on souhaite.

Soit $G = (V, E)$ un graphe non orienté.

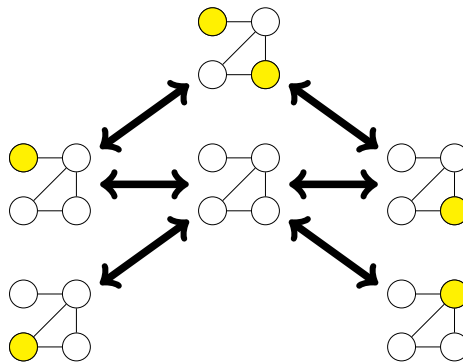
Définition de la chaîne de Markov pour générer un ensemble indépendant dans G . On pose \mathcal{S} comme la collection des sous-ensembles de V qui sont indépendants.

Posons $X_0 = \emptyset$: la distribution initiale de X_0 est donc concentrée sur l'ensemble vide, qui est un ensemble indépendant. La fonction de transition de cette chaîne de Markov est ensuite définie comme suit. Donnons un algorithme qui à un état $S \in \mathcal{S}$ (un ensemble indépendant donc), en associe un autre de manière probabiliste :

```

fonction etatSuivant(S)
  choisir  $v \in V$  au hasard
  tirer une pièce à pile ou face
  si face et  $S \cup \{v\}$  est indépendant
    | renvoyer  $S \cup \{v\}$ 
  sinon
    | renvoyer  $S \setminus \{v\}$  Remarque :  $v$  n'appartient pas nécessairement à  $S$ .
  
```

Exemple 13 Avec le graphe $G = (V, E)$ à 4 nœuds , on obtient la chaîne de Markov suivante à 6 états :



Proposition 14 Soit μ la distribution uniforme sur les ensembles indépendants : pour tout $S \in \mathcal{S}$, $\mu(S) = \frac{1}{|\mathcal{S}|}$. Alors μ est l'unique distribution stationnaire.

6 Application : résoudre un problème d'optimisation

1. On définit une chaîne de Markov \mathcal{M} où chaque état est une solution possible
2. On se balade dans \mathcal{M} .
3. On espère que X_{100000} est une solution optimale (ou quasi-optimale)

Dans cette section, on illustre l'approche sur le voyageur de commerce. On considère une instance du voyageur de commerce où les villes sont : $1, 2, \dots, m$.

On note \mathcal{S} l'ensemble des tours. Les tours sont notés s, s' , etc. Chaque tour est une permutation de $\{1, \dots, m\}$. On note $f(s)$ le poids d'un tour s .

Définition 15 Soit $T > 0$. La **distribution de Boltzmann** π_T est définie par :

$$\pi_T(s) \propto e^{-\frac{f(s)}{T}}.$$

Avec cette distribution, plus un tour est bon, plus il est probable. Le paramètre T est appelé **température**. Plus T est petit, plus cette distribution est fine ; plus T est grand plus elle est étalée.

Définition de la chaîne de Markov pour résoudre le voyageur de commerce. Les états sont tous les tours possibles. On dit que deux tours s et s' sont voisins s'il existe $i, j \in \{1, \dots, m\}$ avec $i < j$ et si on note $s = (s_1, s_2, \dots, s_{i-1}, s_i, s_{i+1}, \dots, s_{j-1}, s_j, s_{j+1}, \dots, s_m)$ alors

$$s' = (s_1, s_2, \dots, s_{i-1}, s_j, s_{j-1}, \dots, s_{i+1}, s_i, s_{j+1}, \dots, s_m)$$

$$\mathbb{P}(s, s') := \begin{cases} \frac{1}{d^\circ(s)} \min\left(\frac{\pi(s')d^\circ(s)}{\pi(s)d^\circ(s')}\right) & \text{si } s \text{ et } s' \text{ voisins} \\ 0 & \text{si } s \neq s' \text{ et } s \text{ et } s' \text{ ne sont pas voisins} \\ 1 - \sum_{v \text{ voisin de } s} \frac{1}{d^\circ(s)} \min\left(\frac{\pi(v)d^\circ(s)}{\pi(s)d^\circ(v)}\right) & \text{si } s = s' \end{cases}$$

Autrement dit quand on se balade, voici l'algorithme qui retourne l'état suivant :

```

fonction etatSuivant(s)
    choisir s' de manière uniforme parmi les successeurs de s
    avec probabilité  $\min\left(\frac{\pi(s')d^\circ(s)}{\pi(s)d^\circ(s')}\right)$ 
    | renvoyer s'
    sinon
    | renvoyer s
    
```

La chaîne de Markov que l'on vient de définir s'appelle **chaîne de Metropolis**.

Théorème 16 π_T est l'unique distribution stationnaire pour la chaîne de Markov ci-dessus.

Algorithme du recuit simulé

Théorème 17

$$\mathbb{P}(Y \text{ choisi selon } \pi_T \text{ soit tel que } f(Y) = \min_{s \in \mathcal{S}} f(s)) \xrightarrow{T \rightarrow 0} 1.$$

L'algorithme consiste à se balader dans la chaîne de Markov en faisant décroître la température doucement vers 0. La chaîne de Markov résultante n'est pas donc pas homogène.

7 Simulation d'une chaîne de Markov

Pour estimer la distribution stationnaire, autant simuler la chaîne de Markov. Soit $\mathcal{M} = (\mathcal{S}, \mu_0, P)$ une chaîne de Markov. Voyons comment simuler le comportement de \mathcal{M} .

Tirer un état initial selon μ_0 . Initialement, on tire une valeur aléatoire r_0 uniformément dans $[0, 1]$. La valeur de r détermine l'état initial de la chaîne de Markov dans cette simulation, selon un découpage de $[0, 1]$:

$$[0, 1] = \underbrace{[0, \mu_0(s_1)]}_{\text{démarrer en } s_1} \sqcup \underbrace{] \mu_0(s_1), \mu_0(s_1) + \mu_0(s_2)]}_{\text{démarrer en } s_2} \sqcup \dots \sqcup \underbrace{] \mu_0(s_1) + \mu_0(s_2) \dots + \mu_0(s_{n-1}), 1]}_{\text{démarrer en } s_n}$$

Prendre une transition. À partir d'un état s , on tire une valeur aléatoire r uniformément dans $[0, 1]$. La valeur de r obtenue permet de savoir dans quel état, selon un découpage de l'intervalle $[0, 1]$:

$$[0, 1] = \underbrace{[0, P_{s,s_1}]}_{\text{aller en } s_1} \sqcup \underbrace{]P_{s,s_1}, P_{s,s_1} + P_{s,s_2}]}_{\text{aller en } s_2} \sqcup \cdots \sqcup \underbrace{]P_{s,s_1} + P_{s,s_2} \cdots + P_{s,s_{n-1}}, 1]}_{\text{aller en } s_n}$$

Typiquement :

$$\varphi(s, u) = \begin{cases} s_1 & \text{si } u \in [0, P_{s,s_1}] \\ s_2 & \text{si } u \in]P_{s,s_1}, P_{s,s_1} + P_{s,s_2}] \\ \vdots & \\ s_n & \text{si } u \in]P_{s,s_1} + P_{s,s_2} \cdots + P_{s,s_{n-1}}, 1] \end{cases}$$

Et voici la fonction qui calcule l'état suivant :

```

fonction etatSuivant( $s$ )
|   choisir  $u \in [0, 1]$  uniformément
|   renvoyer  $\varphi(s, u)$ 

```

Remarque : le choix de l'ordre des intervalles fait ci-dessus est arbitraire. Aussi, on pourrait pour chaque état utiliser une union d'intervalles. La seule chose qui importe est que leur mesure totale soit la probabilité voulue.

Définition 18 Une **fonction de mise à jour** est une fonction $\varphi : \mathcal{S} \times [0, 1] \rightarrow \mathcal{S}$ telle que pour tous états $s, s' \in \mathcal{S}$, la mesure de Lebesgue de $\{u \in [0, 1] \mid \varphi(s, u) = s'\}$ vaut $P_{s,s'}$.

Pour la simulation, le choix de la **fonction de mise à jour** φ , c'est à dire des valeurs $\varphi(s, u)$ pour $s \in \mathcal{S}$ et $u \in [0, 1]$ n'a pas d'importance. Ce sera différent pour l'algorithme de Propp et Wilson décrit dans la section suivante.

Pour estimer la distribution stationnaire d'une chaîne de Markov, on peut donc procéder de la façon suivante. On choisit n un nombre d'étapes de simulation de la chaîne. On effectue N simulations, chacune de n étapes. Si pour $1 \leq i \leq |\mathcal{S}|$, N_i dénote le nombre de simulations dont le n -ième état est s_i , on définit $\nu_{n,N}(s_i) = \frac{N_i}{N}$. La distribution $\nu_{n,N}$ est une « bonne » approximation de la distribution stationnaire, au sens où $\lim_{n \rightarrow \infty} \lim_{N \rightarrow \infty} \nu_{n,N} = \mu$.

Limites Il y a des inconvénients à la simulation présentée ci-dessus. Tout d'abord, $\nu_{n,N}$ converge vers la distribution stationnaire μ , mais dans « la plupart des cas » μ n'est jamais atteinte. Par exemple, si $P = \begin{pmatrix} .75 & .25 \\ .25 & .75 \end{pmatrix}$ et $\mu_0 = [1, 0]$,

on a $\nu_n = \mu_0 P^n = [\frac{1}{2}(1 + 2^{-n}), \frac{1}{2}(1 - 2^{-n})]$.

Pour contrôler la variation totale entre ν_n et la distribution stationnaire, il faut déterminer quelle valeur de n choisir en fonction de l'erreur admissible ε . En pratique, il est difficile d'obtenir des bornes supérieures sur n qui soient suffisamment petites pour être utilisables.

8 Couplage depuis le passé : algorithme de Propp et Wilson

L'algorithme de Propp et Wilson [PW96], bien que basé sur la simulation, pallie deux inconvénients de la simulation présentée précédemment. Premièrement, il calcule **exactement** la distribution stationnaire, au sens où il produit une sortie dont la distribution est exactement la distribution stationnaire. Deuxièmement, on **sait quand arrêter** la simulation.

8.1 Principe

Le principe est de **simuler k exécutions** de la chaîne de Markov en parallèle, depuis chaque état s_1, \dots, s_k . On arrête la simulation quand les k chaînes ont toutes convergé vers un unique même état.

Plus précisément, on simule les k chaînes depuis le temps -1 jusqu'au temps 0 , puis depuis le temps -2 au temps 0 , puis du temps -4 au temps 0 , etc. On s'arrête dès lors qu'il y a convergence. Toutes ces simulations réutilisent les mêmes résultats des tirages uniformes utilisées dans la fonction de mise à jour.

entrée : \mathcal{M} une chaîne de Markov avec k états s_1, \dots, s_k , décrite avec une fonction de mise à jour
 sortie : un état selon la distribution stationnaire de \mathcal{M}

```

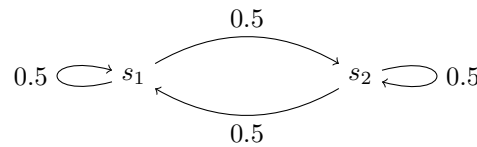
fonction Propp-Wilson( $\mathcal{M}$ )
  Soit  $(u_{-i})_{i \in \mathbb{N}^*}$ , une suite de nombres dans  $[0, 1]$  obtenus par tirages uniformes indépendants
  pour  $m := 0, 1, 2, \dots$  faire
    pour  $i := 1$  à  $k$  faire
       $s'_i :=$  état après la simulation de  $\mathcal{M}$  depuis  $s_i$  du temps  $-2^m$  jusqu'au temps 0
      en utilisant  $u_{-2^m}, \dots, u_{-1}$  dans la fonction de mise à jour
    si  $\{s'_i \mid 1 \leq i \leq k\}$  est un singleton  $\{s'\}$  alors
      renvoyer  $s'$ 
  
```

Remarque : il est crucial, lors de la j -ième simulation de réutiliser les nombres aléatoires déjà utilisés pour la $j - 1$ -ième, à partir du temps -2^{j-1} ! 🎓 fait en TD

8.2 Terminaison ?

Attention, l'algorithme de Propp-Wilson **ne termine pas forcément**. Le choix de la fonction de mise à jour est important. Voici un exemple où l'algorithme ne termine pas :

Exemple 19 Considérons la chaîne de Markov suivante :



avec la fonction de mise à jour : $\varphi(s_1, r) = \begin{cases} s_1 & \text{si } r < \frac{1}{2} \\ s_2 & \text{sinon} \end{cases}$ et $\varphi(s_2, r) = \begin{cases} s_2 & \text{si } r < \frac{1}{2} \\ s_1 & \text{sinon.} \end{cases}$

Voyez-vous pourquoi l'algorithme ne termine pas ? Voyez-vous comment modifier la fonction de mise à jour pour que l'algorithme termine ?

Théorème 20 Soit \mathcal{M} une description d'une chaîne de Markov :

$$\mathbb{P}(\text{algorithme de Propp-Wilson termine sur } \mathcal{M}) \in \{0, 1\}.$$

8.3 Correction

Sous l'hypothèse de terminaison avec probabilité 1, l'algorithme de Propp-Wilson est correct.

Théorème 21 Soit \mathcal{M} une chaîne de Markov irréductible aperiodique.

Si $\mathbb{P}(\text{algorithme termine sur } \mathcal{M}) = 1$ alors le résultat Y suit la distribution stationnaire.

8.4 Optimisation pour les chaînes monotones

Problème : l'algorithme de Propp-Wilson demande de simuler k chaînes de Markov. Parfois, k est très grand. Pour certaines chaînes de Markov dites **monotones**, on peut utiliser du **sandwiching**.

On considère une marche aléatoire sur $\llbracket 1, k \rrbracket$ définie par : $P(1, 1) = P(1, 2) = \frac{1}{2}$; $P(k, k) = P(k, k - 1) = \frac{1}{2}$; et pour $1 < i < k$, $P(i, i - 1) = P(i, i + 1) = \frac{1}{2}$.

Définissons la fonction de mise à jour naturelle par : « si r est petit, on descend dans la chaîne, sinon, on monte ». Alors, pour l'ordre naturel sur les états, pour toute valeur $r \in [0, 1]$, pour tout $i \leq j \in \llbracket 1, k \rrbracket$, $\varphi(s_i, r) \leq \varphi(s_j, r)$. En conséquence, les k chaînes de Markov partant des états s_2 à s_{k-1} restent toujours « entre » celles partant de s_1 et s_k . Donc, lorsque les chaînes associées à s_1 et s_k convergent vers un même état, c'est le cas également des chaînes intermédiaires $s_2 \cdots s_{k-1}$.

La figure 1 illustre une exécution de l'algorithme de Propp et Wilson sur cette marche aléatoire pour $k = 5$.

Notes bibliographiques

Je remercie Nathalie Bertrand qui avait enseigné ALGO2 il y a quelques années. Le matériel provient principalement de l'excellent livre [H⁺02] (excellent, car les chapitres sont courts et digestes). L'algorithme de Propp-Wilson provient de [PW96]. Les chaînes de Markov donnent aussi un algorithme d'approximation pour calculer le nombre de modèles d'une formule propositionnelle [WS05]. Pour en savoir plus, sur le comptage de modèles en logique propositionnelle, voir [CMV21].

Références

- [CMV21] Supratik Chakraborty, Kuldeep S. Meel, and Moshe Y. Vardi. Approximate model counting. In Armin Biere, Marijn Heule, Hans van Maaren, and Toby Walsh, editors, *Handbook of Satisfiability - Second Edition*, volume 336 of *Frontiers in Artificial Intelligence and Applications*, pages 1015–1045. IOS Press, 2021.
- [H⁺02] Olle Häggström et al. *Finite Markov chains and algorithmic applications*, volume 52. Cambridge University Press, 2002.
- [PW96] James Gary Propp and David Bruce Wilson. Exact sampling with coupled markov chains and applications to statistical mechanics. *Random Struct. Algorithms*, 9(1-2) :223–252, 1996.
- [WS05] Wei Wei and Bart Selman. A new approach to model counting. In Fahiem Bacchus and Toby Walsh, editors, *Theory and Applications of Satisfiability Testing, 8th International Conference, SAT 2005, St. Andrews, UK, June 19-23, 2005, Proceedings*, volume 3569 of *Lecture Notes in Computer Science*, pages 324–339. Springer, 2005.

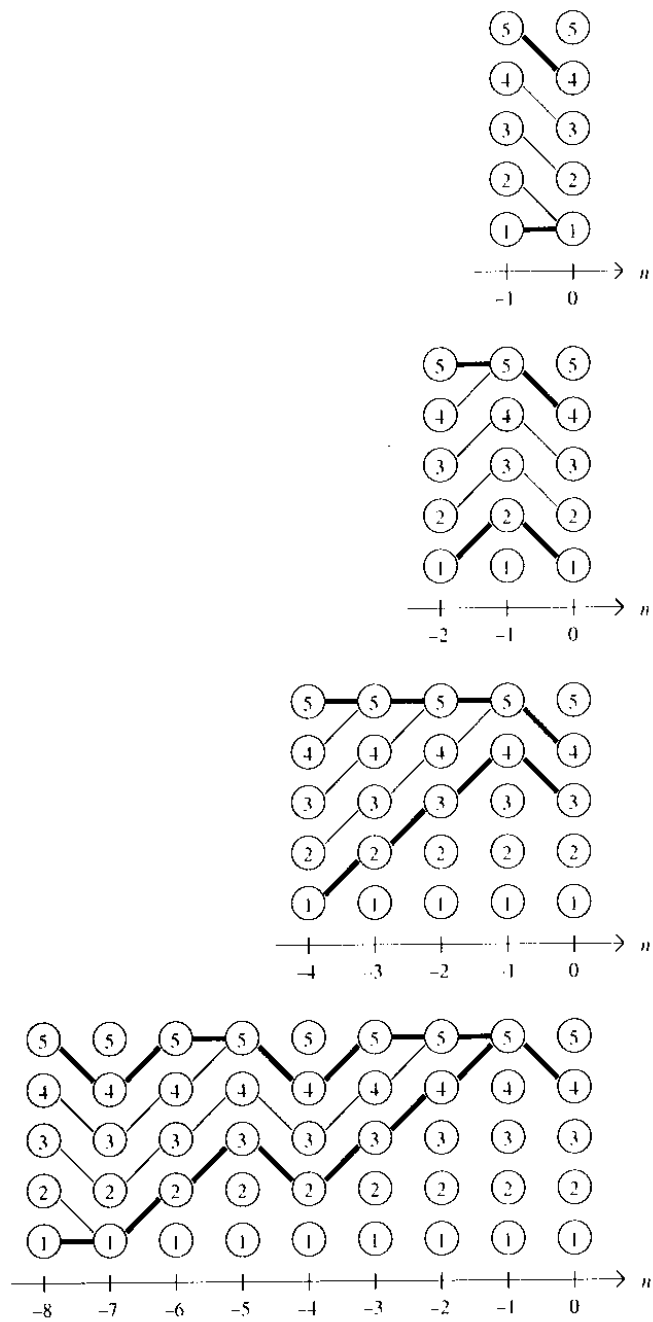


FIGURE 1 – Exemple d'exécution de l'algorithme de Propp et Wilson avec sandwich.