
Perfekte Simulation

1 Wiederholung

2 Perfekte Simulation

2.1 Kopplung von Markov-Ketten ('Coupling')

2.2 Forward Coupling (FC)

2.3 Propp-Wilson: Coupling-From-The-Past (CFTP)

3 Literatur

1 Wiederholung

Im Folgenden sei $(\Omega, \mathcal{F}, \mathcal{P})$ ein beliebiger Wahrscheinlichkeitsraum.

$X = \{X_t, t \in \mathbb{N}\}$ sei eine (homogene) Markov-Kette (MK) mit endlichem Zustandsraum $\mathcal{E} = \{0, 1, \dots, l\}$ und Übergangsmatrix $P = (p_{ij})_{ij}$, wobei $p_{ij} = \mathcal{P}(X_t = j | X_{t-1} = i)$ sei.

Definition 1

- Die MK X heißt *irreduzibel*, falls $\forall i, j \in \mathcal{E} \exists t$ mit $p_{ij}^{(t)} > 0$.
- Die MK X heißt *aperiodisch*, falls $\text{ggT}\{t | p_{ii}^{(t)} > 0\} = 1 \forall i \in \mathcal{E}$.

Definition 2

Die MK X heißt *ergodisch*, falls

- a) $\forall i, j \in \mathcal{E} \pi_j = \lim_{t \rightarrow \infty} p_{ij}^{(t)}$ existiert und
- b) $\pi_j > 0$ und von i unabhängig ist.

Theorem 1

Eine MK X ist ergodisch genau dann, wenn sie irreduzibel und aperiodisch ist.

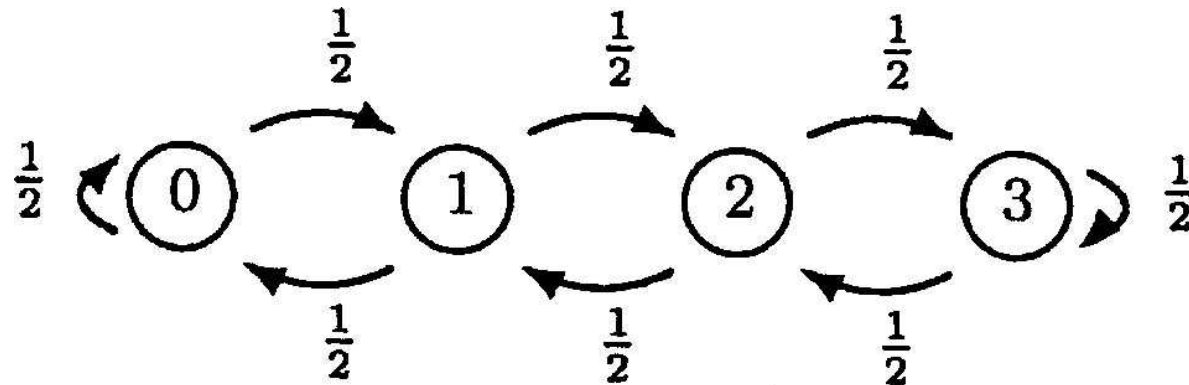
Bemerkung 1

- Bis jetzt haben wir Algorithmen betrachtet, um zu einer gegebenen Verteilung π ergodische MK zu konstruieren, deren Verteilung $a^{(t)}(\cdot) = \mathcal{P}(X_t = \cdot)$ gegen π konvergiert.
- Jetzt wollen wir diese (ergodischen) MK verwenden, um Stichproben zu gewinnen, die möglichst exakt der Gleichgewichtsverteilung π folgen.
- Dabei tritt das Problem der korrekten Bestimmung der sogenannten Burn-In-Zeit (Vorlaufzeit) auf, das heißt wir müssen die Güte der Approximation $\|a^{(t)} - \pi\|_{TV}$ abschätzen.
- Es gibt verschiedene Methoden, die Konvergenzgeschwindigkeit der MK abzuschätzen. (vgl. Vorlesung)

Beispiel 1

- Aufgabenstellung: Wir haben drei Kugeln, die in zwei Urnen verteilt sind. Mit Wahrscheinlichkeit $1/2$ nehmen wir eine Kugel aus der rechten und legen sie in die linke. Alternativ legen wir eine Kugel aus der linken in die rechte Urne. Falls die gewählte Urne leer ist, tun wir nichts. Was ist auf lange Zeit gesehen die durchschnittliche Anzahl M der Kugeln in der rechten Urne?
- Model: Random Walk mit MK $X = \{X_t\}$, wobei $\{X_t\}$ die Anzahl der Kugeln in der rechten Urne beschreibt.

Abbildung 1: Random Walk bei Bsp. 1



- Hier wäre $\pi^T P = \pi^T$ einfach zu lösen und man könnte damit dann $M = E_\pi(X)$ schnell bestimmen.
- Aber wir wollen M mittels Simulation schätzen.
- 1. Lösungsansatz: Simuliere X mit beliebigen Startwert X_0 n -mal und berechne $M \approx \frac{1}{n} \sum_{t=1}^n X_t$.

- Problem: Bei diesem Ansatz hängt der Wert für M stark vom gewählten Anfangswert X_0 ab.
- 2. Lösungsansatz: Simuliere X $(m + n)$ -mal, ignoriere die ersten m Schritte (Burn-In-Zeit) und berechne $M \approx \frac{1}{n} \sum_{t=m+1}^{t=m+n} X_t$.
- Problem: Bestimmung von geeignetem m (Burn-In-Zeit).
- Lösung: PERFEKTE SIMULATION
Diese Algorithmen liefern automatisch erst dann Stichprobenrealisierungen zurück, wenn die Gleichgewichtsverteilung bereits erreicht ist.

2 Perfekte Simulation

2.1 Kopplung von Markov-Ketten ('Coupling')

Seien X und X' zwei MK mit dem selben Zustandsraum \mathcal{E} und der selben Übergangsmatrix P . Definiere $T = \inf\{t \geq 0 \mid X_t = X'_t\}$ und setze

$$X''_t = \begin{cases} X_t, & \text{falls } t < T, \\ X'_t, & \text{falls } t \geq T \end{cases}$$

Dann ist (X', X'') eine Kopplung der MK X .

Definition 3

Eine *Kopplung* einer MK X ist ein Prozess $\{(X'_t, X''_t), t \in \mathbb{N}\}$, für den gilt:

1. $\{X'_t\}, \{X''_t\}$ genügen den Übergangsvorschriften der MK X , d.h.
 $\mathcal{P}(X'_t = j | X'_{t-1} = i) = p_{ij} = \mathcal{P}(X''_t = j | X''_{t-1} = i)$ und
2. falls $X'_t = X''_t$, dann gilt $X'_{t+1} = X''_{t+1}$.

Stochastische Rekursionsgleichung

Sei $\{X_t, t \in \mathbb{N}\}$ eine ergodische MK. Dann existiert eine Funktion $\phi : \mathcal{E} \times D \longrightarrow \mathcal{E}$ und eine Folge von iid. ZV (U_t) mit Werten in D , so dass $X_t = \phi(X_{t-1}, U_t)$. Wir bezeichnen ϕ als die *Update-Funktion* von $\{X_t\}$.

Beispiel 1: Update-Funktion des betrachteten Urnenmodells

Die ZV U_t sei gleichverteilt auf $\{0, 1\}$. Dann ist

$$\phi(i, U) = \begin{cases} \min(i + 1, 3), & \text{falls } U = 1, \\ \max(i - 1, 0), & \text{falls } U = 0. \end{cases}$$

2.2 Forward Coupling (FC)

Dieses Verfahren läuft von der 'Gegenwart' in die 'Zukunft'.

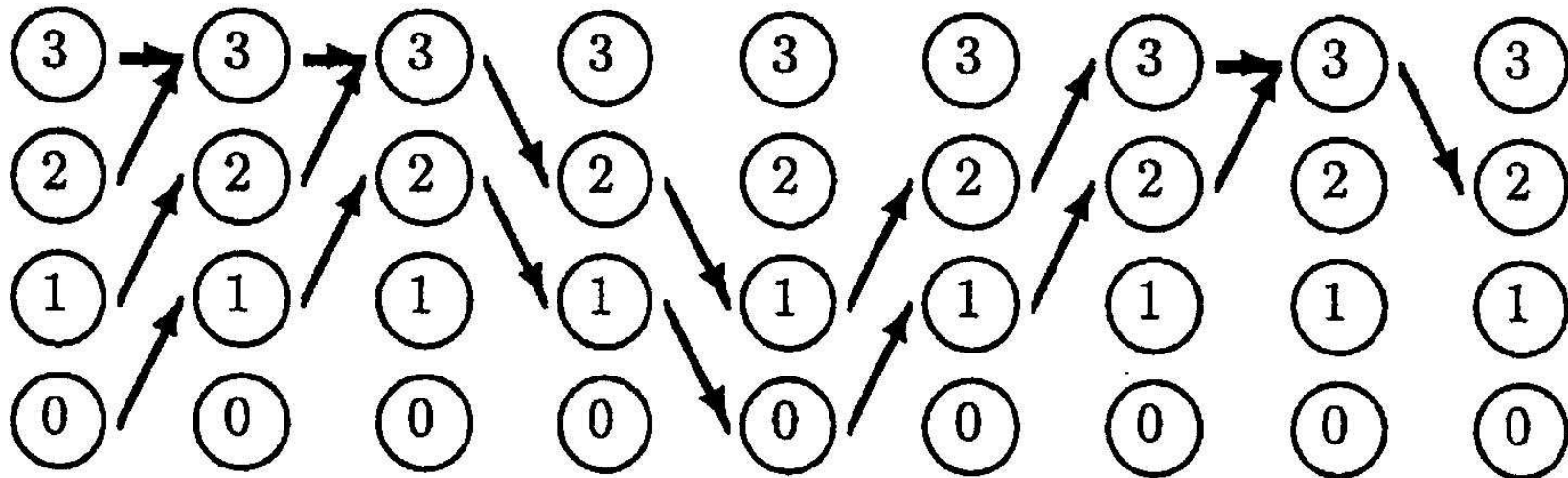
Algorithmus 1

- In jedem möglichen Zustand $i \in \mathcal{E}$ wird eine Kopie der MK $\{X_t, t \in \mathbb{N}\}$ gestartet.
- Zur Entwicklung dieser Ketten über die Update-Regel $X_t = \phi(X_{t-1}, U_t)$ werden dieselbe Update-Funktionen ϕ und (hier) dieselbe Folge von ZV (U_t) verwendet.

- Nehmen zwei Ketten in einem Schritt den gleichen Zustand an, so verschmelzen ihre Pfade aufgrund der Konstruktion.
- Sind alle $l + 1$ Kopien zu einem Zeitpunkt T_{FC} ('Kopplungszeitpunkt') miteinander verschmolzen, dann hat die spezielle Wahl des jeweiligen Anfangszustandes keinen Einfluss mehr auf das Ergebnis.

Beispiel 1: Schaubild des betrachteten Urnenmodells

Abbildung 2: Kopplungspfade des Random Walks, falls die Zufallsmünze die Werte 1, 1, 0, 0, 1, 1, 1, 0 liefert.



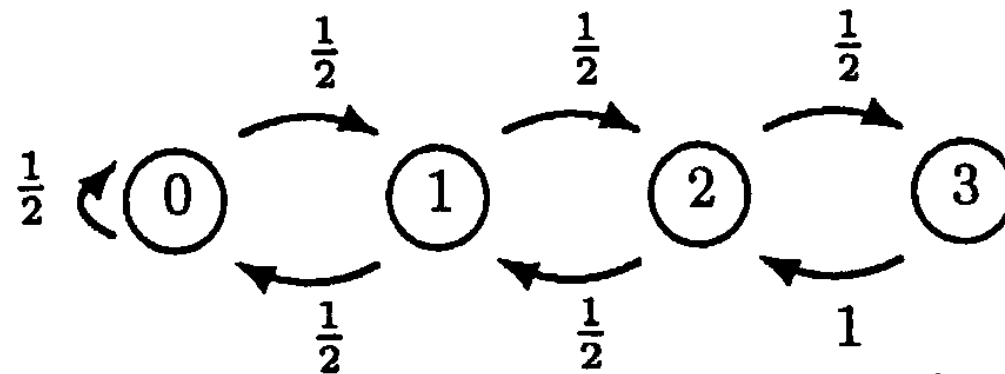
Bemerkung 2

- Wegen der Monotonie der Updatefunktion ϕ genügt es, die Pfade zu beobachten, die in Zustand 0 und 3 starten. Denn vollständige Verschmelzung aller Pfade tritt genau dann ein, falls diese beiden Pfade miteinander verschmelzen.
- In Modellen mit sehr großem Zustandsraum \mathcal{E} kann durch eine derartige Monotonieeigenschaft die Komplexität (und damit der Rechenaufwand und die Rechenzeit) deutlich reduziert werden.
- Ist dagegen die Updatefunktion ϕ monoton fallend, dann kann durch einen ähnlichen Trick die Komplexität gesenkt werden.

Beispiel 2

- Wir betrachten folgenden Random Walk:

Abbildung 3: Random Walk in Beispiel 2

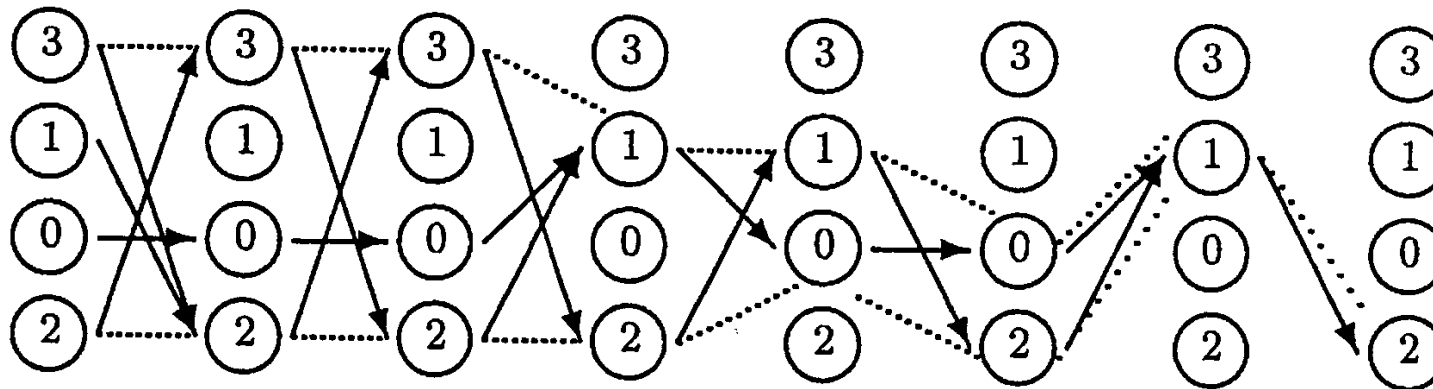


- Unsere Updatefunktion sieht nun wie folgt aus, ist aber leider nicht monoton:

$$\phi(i, U) = \begin{cases} i + 1, & \text{falls } U = 1 \text{ und } i \in \{1, 2\}, \\ i - 1, & \text{falls } U = 0 \text{ und } i \in \{1, 2\}, \\ 0, & \text{falls } U = 1 \text{ und } i = 0, \\ 1, & \text{falls } U = 0 \text{ und } i = 0, \\ 2, & \text{falls } i = 3. \end{cases}$$

- Ändern wir aber die natürliche Ordnungsrelation in $2 \prec 0 \prec 1 \prec 3$, dann ist unsere Updatefunktion ϕ monoton fallend: Falls unsere Münze z.B. '1' zurückgibt, wird der Zustand '3' (also der höchste) in den Zustand '2' (also den niedrigsten) übergeführt und umgekehrt.

Abbildung 4: Kopplungspfade des Random Walks, falls die Zufallsmünze die Werte 1, 1, 0, 0, 1, 1, 1, 0 liefert.



- Damit können wir wie im vorhergehenden Beispiel wiederum uns darauf beschränken, nur zwei Pfade X^{min} und X^{max} zu betrachten, um vollständige Verschmelzung aller Pfade zu erkennen.
- Wir entwickeln die Pfade wie folgt:

$$X_0^{min} = 2 \text{ und } X_0^{max} = 3$$

$$X_{t+1}^{min} = \phi(X_t^{max}, U_t) \text{ und } X_{t+1}^{max} = \phi(X_t^{min}, U_t)$$

- Die zwei Komponenten sind einzeln betrachtet keine MK, solange sie noch nicht miteinander verschmolzen sind. Sobald sie das aber sind, verhalten sie sich wie unser Random Walk.

Bemerkung 3

- In den beiden letzten Beispielen haben wir für die Update-Funktionen der unterschiedlichen Pfade die selbe Folge von ZV (U_t) benützt.
- Im Allgemeinen (bei nicht monotonen Update-Funktionen) dürfen wir das nicht, sondern müssen für unterschiedliche Pfade bis zu ihrer Verschmelzung unabhängige Folgen (U_t^i) betrachten.

Zurück zum Forward Coupling!

Bemerkung 4

- Es liegt nun nahe, anzunehmen, dass $X_{T_{FC}}$ der Gleichgewichtsverteilung folgt. Dies gilt jedoch im Allgemeinen nicht.
- Die Begründung liegt darin, dass die Realisierungen zu einem zufälligen Zeitpunkt ausgegeben werden und sich hierdurch eine verfälschte Stichprobe ergeben kann.

Beispiel 3

- Es sei $\mathcal{E} = \{0, 1\}$. Unsere Folge (U_t) sei gleichverteilt auf $\{0, 1\}$.
- Die Update-Regel lautet:

$$\phi(0, U) = \begin{cases} 0, & \text{falls } U = 0, \\ 1, & \text{falls } U = 1. \end{cases} \quad \phi(1, U) = 0.$$

- Für die Gleichgewichtsverteilung dieser MK ergibt sich $\pi = (\frac{2}{3}, \frac{1}{3})^T$.
- Aus den beiden Startzuständen wird nun je eine Kette gestartet und unter Verwendung der Update-Funktion ϕ entwickelt.
- Der einzige Zustand, in dem diese beiden Ketten verschmelzen können, ist der Zustand 0.

2.3 Propp-Wilson: Coupling-From-The-Past (CFTP)

- Anstatt die Simulation zum Zeitpunkt 0 zu starten, simuliert man von einem Zeitpunkt in der Vergangenheit bis zu $T = 0$.
- Die Stichprobe wird so zu dem deterministischen Zeitpunkt $T = 0$ erhoben und man kann zeigen, dass sie exakt die gewünschte Gleichgewichtsverteilung π besitzt.

Bezeichnungen

- Sei $\Phi = (\Phi(0), \Phi(1), \dots, \Phi(l))$ mit $\Phi(i) = \phi(i, U^i)$, wobei die ZV U^i iid. sind, bzw. im monotonen Fall mit $\Phi(i) = \phi(i, U)$.
- $(\Phi_t)_{t \in \mathbb{Z}}$ sei eine doppelt unendliche Folge von ZV mit $\Phi_t \stackrel{d}{=} \Phi \quad \forall t$.

- Die Abbildung $F_{s,t} : \mathcal{E} \rightarrow \mathcal{E}$ sei folgendermaßen definiert:

$$F_{t,t}(i) = id(i)$$

$$F_{s,t}(i) = \Phi_t \circ \dots \circ \Phi_{s+1}(i), \text{ falls } s < t.$$

- $X_t^s(i) = \Phi_t \circ \dots \circ \Phi_{s+1}(i)$ ist der Zustand der MK X zum Zeitpunkt t , wenn sie zur Zeit s im Zustand i war.
- Der Zeitpunkt T_{CFTP} bezeichne den Kopplungszeitpunkt des Coupling-from-the-Past-Verfahrens, wobei $T_{CFTP} = \inf\{t \geq 0 | X_0^{-t}(0) = \dots = X_0^{-t}(l)\}$.

Algorithmus 2: CFTP

1. Wähle einen Startzeitpunkt $-t_0, t_0 \in \mathbb{N}$, und setze $n = 0$.
2. Generiere $F_{-t_n, -t_{n-1}}$ und bestimme
$$F_{-t_n, 0} = F_{-t_{n-1}, 0} \circ F_{-t_n, -t_{n-1}} \text{ mit } t_{-1} = 0.$$
3. Falls $F_{-t_n, 0}$ eine konstante Abbildung ist, gib $F_{-t_n, 0}$ aus.
4. Andernfalls setze $n = n + 1$ und fahre in Schritt 2 fort mit $-t_n > -t_{n+1}$ für $t_{n+1} \in \mathbb{N}$.

Das Zusammentreffen aller Pfade bis zum Zeitpunkt 0 ist für jedes $t > T_{CFTP}$ gleichbedeutend mit dem Ereignis $\{F_{-t, 0} \text{ ist eine konstante Abbildung}\}$.

Bemerkung 5

- Wegen Schritt 2 nennt man CFTP auch 'Backward Coupling':
Während der Simulation greift man immer weiter in die Vergangenheit des Prozesses zurück.
- Um die Anzahl der notwendigen Auswertungen $m \cdot t_0 + m \cdot t_1 + \dots + m \cdot t_{n_{CFTP}}$ (mit $m = 2$ im monotonen Fall und $m = l + 1$ sonst und $t_{n_{CFTP}} \geq T_{CFTP}$) zu minimieren, wählt man oft die Folge der Startzeitpunkte (t_n) wie folgt: $t_{n+1} = 2t_n$.

.

Theorem 2

Der CFTP-Algorithmus terminiert mit Wahrscheinlichkeit 1 und der zurückgegebene Wert ist gemäß der Gleichgewichtsverteilung π der MK X verteilt.

Beweisidee

Der Beweis ist nachzulesen zum Beispiel bei Thönnies. Dabei zeigt man die beiden Aussagen:

1. $\mathcal{P}(T_{CFTP} < \infty) = 1$.
2. Für den zurückgegebenen Wert $Y = X_0^{-T_{CFTP}}$ des Algorithmus gilt:

$$\mathcal{P}(Y = i) = \pi(i) \quad \forall i \in \mathcal{E}.$$

.

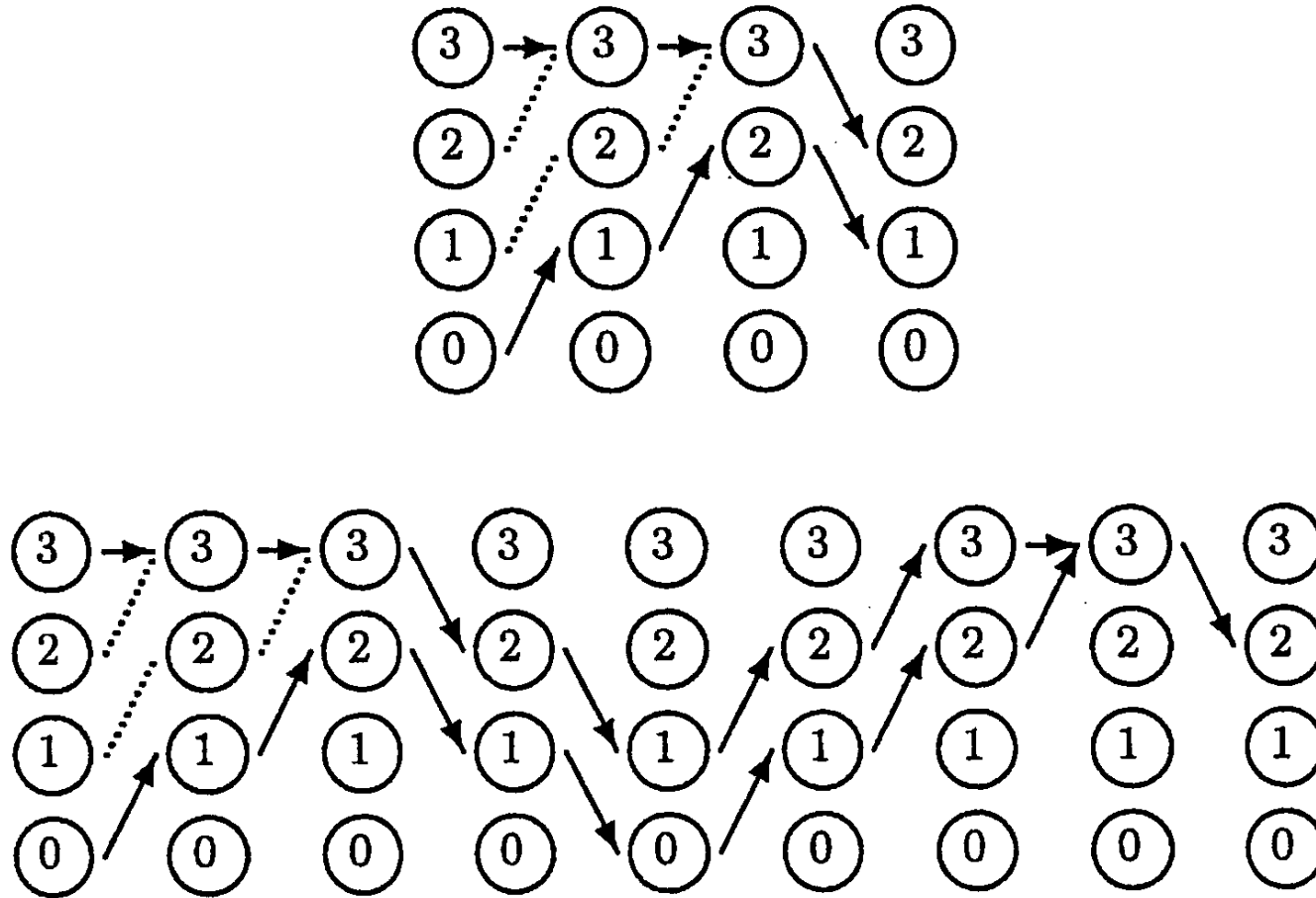
Heuristische Erklärung für das Funktionieren dieses Algorithmus

- Angenommen man könnte die MK X in $t = -\infty$ starten, dann wäre sie bis zum Zeitpunkt $-T_{CFTP}$ bereits unendlich lange gelaufen und folgte aufgrund ihrer Ergodizität der Gleichgewichtsverteilung π .
- Sei i^* die Realisierung zum Kopplungszeitpunkt. Weil π eine Gleichgewichtsverteilung der MK X ist und die Update-Regel Φ diese erhält, ist auch $F_{-T_{CFTP},0}(i^*)$ eine Realisierung gemäß π .
- Eine Simulation ist natürlich unrealistisch, aber der tatsächliche Wert i^* ist irrelevant für die Realisierung zum Zeitpunkt 0, da $F_{-T_{CFTP},0}$ eine konstante Abbildung ist.

Noch einmal: Beispiel 1

- Zur Zeit $-t, t \in \mathbb{N}$ werfen wir unabhängig von den vorhergehenden Würfeln eine Münze U_t .
- Wir starten in allen möglichen Zuständen $\{0,1,2,3\}$ und entwickeln sie bis zur Zeit 0 entsprechend den geworfenen Zahlen $U_{-t}, U_{-t+1}, \dots, U_{-1}$. (Beachte: Reihenfolge der Münzen)
- Falls alle Pfade zur Zeit 0 verschmolzen sind, haben wir eine Stichprobe der Gleichgewichtsverteilung. Andernfalls gehen wir in der Zeit einen Schritt zurück.
- Dabei verwenden wir die Ergebnisse der bereits geworfenen Münzen wieder.

Abbildung 5: Kopplungspfade bei CFTP, falls die Zufallsmünze die Werte 1, 1, 0, 0, 1, 1, 1, 0 liefert.



Bemerkung 6: Probleme dieses Algorithmus

1. Bei großen Zustandsräumen und nicht-monotonen Update-Regeln wird viel Speicher benötigt, da die in den vorangegangenen Schritten generierten Zufallszahlen gespeichert werden müssen.

Möglicher Ausweg: Read-once Algorithmus von D. Wilson

2. Wir wissen lediglich, dass der Propp-Wilson-Algorithmus in endlicher Zeit terminiert. Die Laufzeit ist jedoch zufällig.

3. Ein ungeduldiger Anwender, der langandauernde Simulationsabläufe abbricht, kann eine Verzerrung ('Bias') der Stichprobe erzeugen.

Möglicher Ausweg: Fills 'unterbrechbarer' Algorithmus

Beispiel 4: Bias durch ungeduldigen Benutzer

- Sei X eine MK mit $\mathcal{E} = \{0, 1, 2\}$ und $P = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 0 & 1 \\ \frac{1}{2} & 0 & \frac{1}{2} \end{pmatrix}$.
- Dann ist $\pi = (\frac{2}{5}, \frac{1}{5}, \frac{2}{5})^T$.
- Wir können folgende Update-Regel benutzen:

$$\forall i \in \{0, 2\} \quad \phi(i, U) = \begin{cases} 0, & \text{falls } U = 0, \\ \min\{i + 1, 2\}, & \text{falls } U = 1, \end{cases}$$

$$\phi(1, U) = 2,$$

wobei U wieder gleichverteilt auf $\{0, 1\}$ ist.

- Nehmen wir an, dass der Benutzer die Ausführung des CFTP-Algorithmus stoppt, falls dieser I Iterationen ohne Ergebnis durchlaufen hat.
- Die Folge der Startzeitpunkte sei gemäß der Verdoppelungsstrategie $t_{n+1} = 2t_n$ gewählt; d.h. nach der I -ten Iteration sind wir $N = 2^{I-1}$ Schritte in die Vergangenheit gegangen.
- Nach der ersten Iteration kann noch kein Verschmelzen der Pfade eintreten. Mögliche Zustände sind $\{0, 2\}$ und $\{1, 2\}$.
- Bis das Verschmelzen der Pfade eintritt, alterniert die Menge ihrer Zustände zwischen $\{0, 2\}$ und $\{1, 2\}$.
- In jedem der beiden Fälle ist die Wahrscheinlichkeit des Verschmelzens der beiden Pfade im nächsten Schritt $1/2$.

- Insgesamt erhält man also als Wahrscheinlichkeitsdichte der generierten Stichprobe Y unter der Bedingung A_I

$A_I :=$ Die Pfade verschmelzen innerhalb von I Iterationen

$$\mathcal{P}(Y = i | A_I) = \frac{\mathcal{P}(A_I, Y=i)}{\mathcal{P}(A_I)}$$

$$\mathcal{P}(Y = i | A_I) = \frac{\sum_{k=2}^N \frac{1}{2}^{k-1} \left[\frac{1}{2} p_{0,i}^{(N-k)} + \frac{1}{2} p_{2,i}^{(N-k)} \right]}{1 - \frac{1}{2}^{N-1}}.$$

- Für $I = 2$ erhält man:

$$\mathcal{P}(Y = 0 | A_2) = \frac{1}{2} = \mathcal{P}(Y = 2 | A_2) .$$

- Für $I \geq 3$ erhält man:

$$\mathcal{P}(Y = 0 | A_I) = \frac{2}{5} \left[\frac{1 - 2^{-N}}{1 - 2^{-N+1}} \right],$$

$$\mathcal{P}(Y = 1 | A_I) = \frac{1}{5} \left[\frac{1 - 2^{-N}}{1 - 2^{-N+1}} \right],$$

$$\mathcal{P}(Y = 2 | A_I) = \frac{2}{5} \left[\frac{1 - 7 \times 2^{-N-1}}{1 - 2^{-N+1}} \right].$$

3 Literatur

- J. Fill, An Interruptible Algorithm for Perfect Sampling via Markov Chains
- Morten Fismen, Exact simulation using Markov Chains
- O. Häggström, Finite Markov Chains and algorithmic applications
- C. Lantujoul, Geostatistical simulation: models and algorithms
- J. Propp, D. Wilson, Coupling from the Past: a User's Guide
- E.Thönnnes, A primer on perfect simulation
- E.Thönnnes, Perfect Simulation of some Point Processes for the Impatient User