Perfect sampling of non-monotone Markov chains

Ana Bušić

LIAFA, University Paris 7, France

joint work with Bruno Gaujal (INRIA) and Jean-Marc Vincent (University Joseph Fourier)

Eindhoven, January 2009

Discrete Event System

System description: $(\mathcal{X}, \pi^0, \mathcal{E}, p, \phi)$

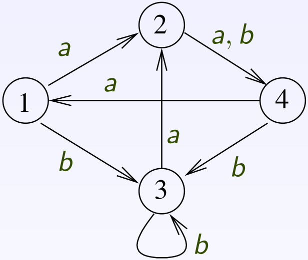
- Finite state space $\mathcal{X}$.
  Without loss of generality, $\mathcal{X} = \{1, \dots, N\}$.
- Probability measure π^0 on $\mathcal{X}$:
  $\pi^0_x \geq 0,\ x \in \mathcal{X}$ is the probability that the system is in state x at time 0.
- Finite set of events $\mathcal{E}$.
- Probability measure p on $\mathcal{E}$:
  $p_e > 0,\ e \in \mathcal{E}$ is the probability of event e.
- Transition function $\phi : \mathcal{X} \times \mathcal{E} \to \mathcal{X}$.

Discrete Event System (II)

Evolution of the system (over n steps):

1. Choose initial state X_0 with probability measure π^0.
2. For $i = 1$ to n do:
   - Choose an event $e_i \in \mathcal{E}$ with probability measure p
   - $X_i := \phi(X_{i-1}, e_i)$

Example



Let $p_a = 1/3, p_b = 2/3$, and $\pi^0 = (1/4, 1/4, 1/4, 1/4)$.

A possible trajectory of the system is
$1 - 3 - 3 - 2 - 4 - 1 - 3 - 3 - \cdots$ starting from state 1 and for sequence of events $bbababb\ldots$.

Remarks

Random sequence $\{X_n\}_{n \in \mathbb{N}}$ is a discrete time Markov chain (DTMC) with transition probability matrix:

$$P_{i,j} \stackrel{\text{def}}{=} \mathbb{P}(X_n = j | X_{n-1} = i) = \sum_{e \in \mathcal{E}} p_e \mathbf{1}_{\phi(i,e)=j}.$$

Conversely, every DTMC can be represented in a form $(\mathcal{X}, \pi^0, \mathcal{E}, p, \phi)$. For a chain with N states, we can construct an event representation with at most N^2 events.

Sampling the Steady-state

Assumption: $\{X_n\}_{n \in \mathbb{N}}$ is irreducible and aperiodic.

Problem: How to sample its stationary distribution π?

Sampling the Steady-state

Assumption: $\{X_n\}_{n \in \mathbb{N}}$ is irreducible and aperiodic.

Problem: How to sample its stationary distribution π?

Naive idea: solve the linear system $\pi = \pi P$ to find π, then use discrete probability measure sampling.
Complexity of computing π: $O(N^3)$ (where $N = |\mathcal{X}|$).

Sampling the Steady-state

Assumption: $\{X_n\}_{n\in\mathbb{N}}$ is irreducible and aperiodic.

Problem: How to sample its stationary distribution π?

Naive idea: solve the linear system $\pi = \pi P$ to find π, then use discrete probability measure sampling.
Complexity of computing π: $O(N^3)$ (where $N = |\mathcal{X}|$).

How to avoid computing π?

Monte-Carlo Simulation

Algorithm:

- Sample X_0 from π^0.
- For $i = 1$ to n:
  - Sample e_i from p.
  - $X_i = \phi(X_{i-1}, e_i)$.

Output: a sample from the probability measure $\pi^0 P^n$.

Complexity: $O(\mathcal{C}(\phi)n)$.

Remark: sampling from discrete probability measure can be done in $O(1)$ using alias method [Walker, 74]. GOTO

Monte-Carlo Simulation

Algorithm:

- Sample X_0 from π^0.
- For $i = 1$ to n:
  - Sample e_i from p.
  - $X_i = \phi(X_{i-1}, e_i)$.

Output: a sample from the probability measure $\pi^0 P^n$.

Complexity: $O(\mathcal{C}(\phi) n)$.

Remark: sampling from discrete probability measure can be done in $O(1)$ using alias method [Walker, 74]. GOTO

Inconvenient: approximation.
Error estimation is difficult: depends on the second eigenvalue of P which is hard to compute [Brémaud, Glynn, Whitt, Hordijk].

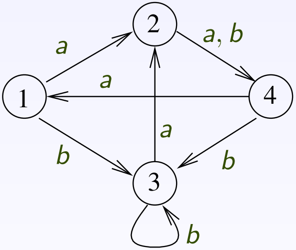
Perfect Simulation

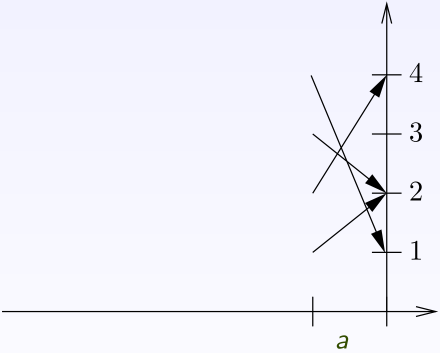
Goal:

- unbiased samples of π without computing it (nor P).
- finite stopping time.

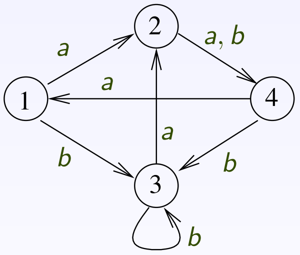
Propp and Wilson (1996) proposed an efficient perfect simulation algorithm based on a backward coupling scheme.

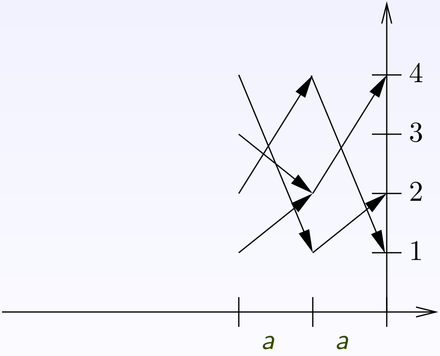
Backward coupling



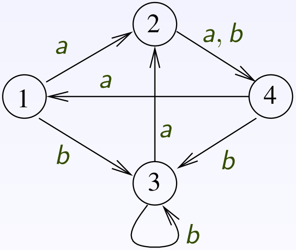


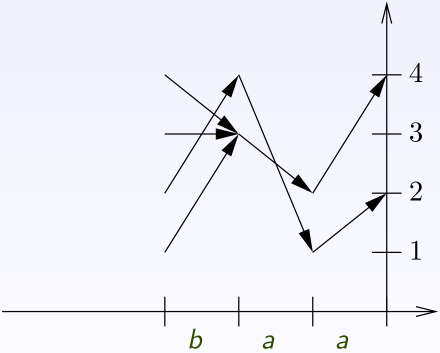
Backward coupling



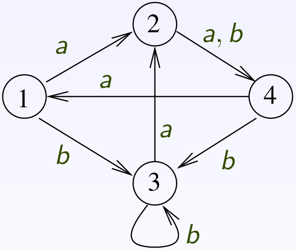


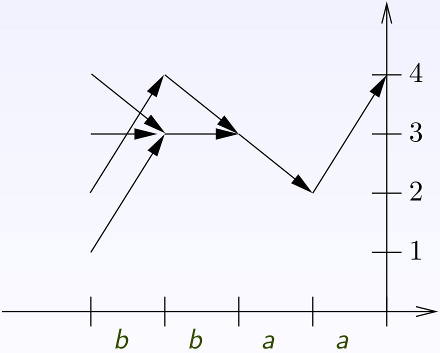
Backward coupling



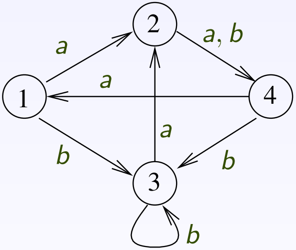


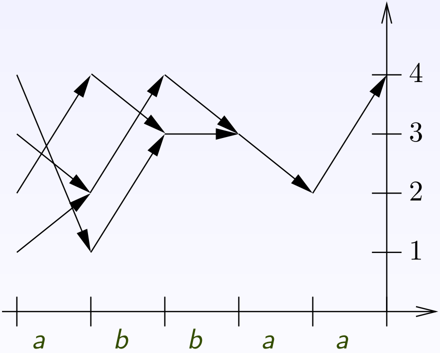
Backward coupling





Backward coupling





Backward coupling (II)

$\Phi^n(x, e_{1\to n}) \stackrel{\text{def}}{=} \Phi(\ldots\Phi(\Phi(x, e_1), e_2), \ldots, e_n)$.
For $A \subset \mathcal{X}$, $\Phi^n(A, e_{1\to n}) \stackrel{\text{def}}{=} \{\Phi^n(x, e_{1\to n}), x \in A\}$.

Theorem ([Propp and Wilson (1996)])

There exists $\ell \in \mathbb{N}$ such that

$$\lim_{n\to\infty} \left|\Phi^n(\mathcal{X}, e_{-n+1\to 0})\right| = \ell \textit{ almost surely.}$$

The system couples if $\ell = 1$. In that case, the value of $\Phi^n(\mathcal{X}, e_{-n+1\to 0})$ is steady state distributed.

Coupling time: $\tau^b \stackrel{\text{def}}{=} \min\{n \in \mathbb{N} \ : \ |\Phi^n(\mathcal{X}, e_{-n+1\to 0})| = 1\}$.

Backward coupling (II)

$\Phi^n(x, e_{1\to n}) \stackrel{\text{def}}{=} \Phi(\ldots\Phi(\Phi(x, e_1), e_2), \ldots, e_n)$.

For $A \subset \mathcal{X}$, $\Phi^n(A, e_{1\to n}) \stackrel{\text{def}}{=} \{\Phi^n(x, e_{1\to n}), x \in A\}$.

Theorem ([Propp and Wilson (1996)])

There exists $\ell \in \mathbb{N}$ such that

$$\lim_{n\to\infty} \left|\Phi^n(\mathcal{X}, e_{-n+1\to 0})\right| = \ell \textit{ almost surely.}$$

The system couples if $\ell = 1$. In that case, the value of $\Phi^n(\mathcal{X}, e_{-n+1\to 0})$ is steady state distributed.

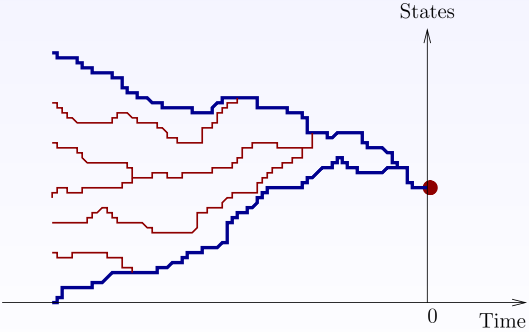
Coupling time: $\tau^b \stackrel{\text{def}}{=} \min\{n \in \mathbb{N} \ : \ |\Phi^n(\mathcal{X}, e_{-n+1\to 0})| = 1\}$.

Inconvenient: Complexity $O(\tau^b \mathcal{C}(\phi) N)$.

Monotone systems

Assumption: state space is partially ordered ($\prec$) and transition function is monotone:

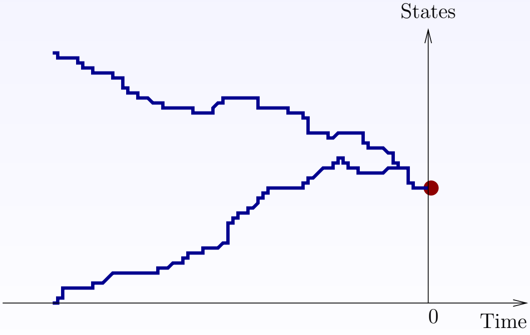
$$x \prec y \Rightarrow \forall e \in \mathcal{E}, \phi(x,e) \prec \phi(y,e).$$



Monotone systems

Assumption: state space is partially ordered ($\prec$) and transition function is monotone:

$$x \prec y \Rightarrow \forall e \in \mathcal{E}, \phi(x, e) \prec \phi(y, e).$$



Algorithm

Algorithm 1: Monotone perfect simulation algorithm.

```
begin
    n = 1;
    repeat
        for i = n − 1 downto ⌊n/2⌋ do
            Sample e_{−i} from p
        E := MIN ∪ MAX;
        for i = n − 1 downto 0 do
            E := φ(E, e_{−i});
        n := 2n;
    until |E| = 1 ;
    return x ∈ E;
end
```

Complexity: $O(\tau^b \mathcal{C}(\phi)|MIN \cup MAX|)$.

Non-monotone case

Assumption: $(\mathcal{X}, \prec)$ is a lattice. Let $T \stackrel{\text{def}}{=} \sup \mathcal{X}$ and $B \stackrel{\text{def}}{=} \inf \mathcal{X}$.

New transition function $\Gamma : \mathcal{X} \times \mathcal{X} \times \mathcal{E} \to \mathcal{X} \times \mathcal{X}$

$$\Gamma_1(m, M, e) \stackrel{\text{def}}{=} \inf_{m \prec x \prec M} \phi(x, e)$$

$$\Gamma_2(m, M, e) \stackrel{\text{def}}{=} \sup_{m \prec x \prec M} \phi(x, e).$$

Theorem

If $\Gamma^n(B, T, e_{-n+1 \to 0})$ hits the diagonal $\mathcal{D}$ (i.e. states of the form (x, x)) in finite time: $\tau^e \stackrel{\text{def}}{=} \min \left\{ n : \Gamma^n(B, T, e_{-n+1 \to 0}) \in \mathcal{D} \right\}$, then $\Gamma^{\tau_e}(B, T, e_{-\tau_e+1 \to 0})$ has the steady state distribution π.

Non-monotone case

Assumption: $(\mathcal{X}, \prec)$ is a lattice. Let $T \stackrel{\text{def}}{=} \sup \mathcal{X}$ and $B \stackrel{\text{def}}{=} \inf \mathcal{X}$.

New transition function $\Gamma : \mathcal{X} \times \mathcal{X} \times \mathcal{E} \to \mathcal{X} \times \mathcal{X}$

$$\Gamma_1(m, M, e) \stackrel{\text{def}}{=} \inf_{m \prec x \prec M} \phi(x, e)$$

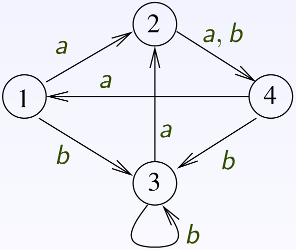
$$\Gamma_2(m, M, e) \stackrel{\text{def}}{=} \sup_{m \prec x \prec M} \phi(x, e).$$

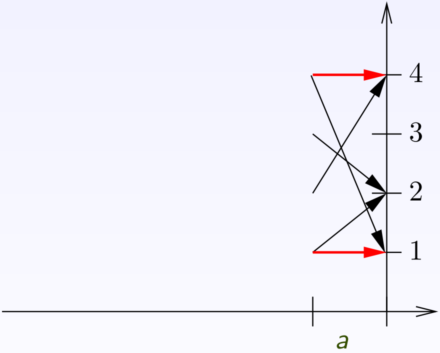
Theorem

If $\Gamma^n(B, T, e_{-n+1\to 0})$ hits the diagonal $\mathcal{D}$ (i.e. states of the form (x, x)) in finite time: $\tau^e \stackrel{\text{def}}{=} \min \left\{ n : \Gamma^n(B, T, e_{-n+1\to 0}) \in \mathcal{D} \right\}$, then $\Gamma^{\tau_e}(B, T, e_{-\tau_e+1\to 0})$ has the steady state distribution π.

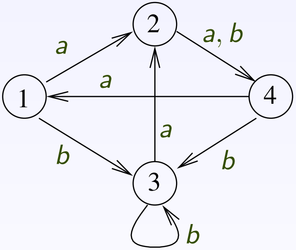
Proof: If $(m_0, M_0) \stackrel{\text{def}}{=} \Gamma^n(B, T, e_{-n+1\to 0})$, then the set $\phi^n(\mathcal{X}, e_{-n+1\to 0})$ is included in $\{x : m_0 \prec x \prec M_0\}$. If the latter is reduced to one point, so is the set $\phi^n(\mathcal{X}, e_{-n+1\to 0})$.

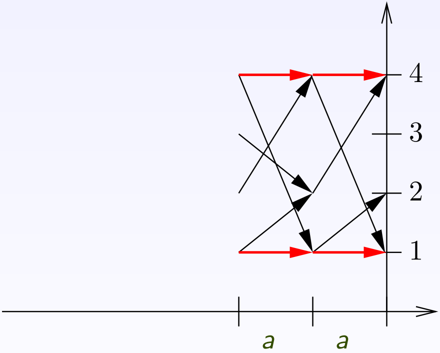
Example



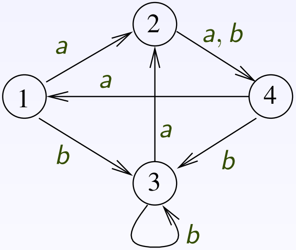


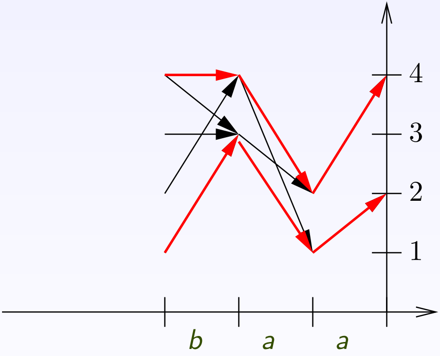
Example



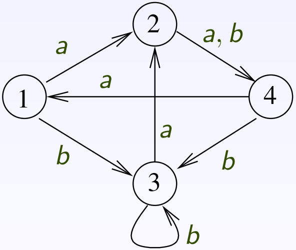


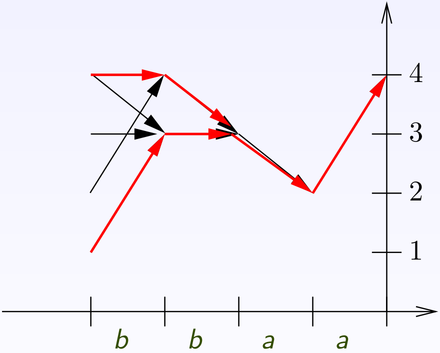
Example



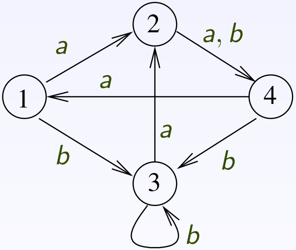


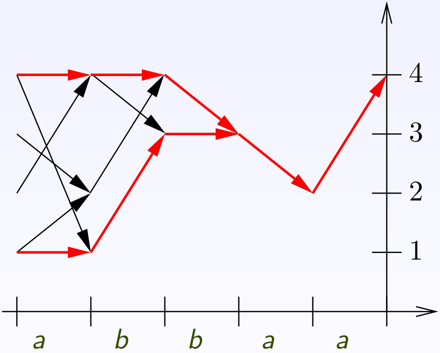
Example





Example





Envelope perfect simulation

Algorithm 2: EPSA: Envelope Perfect Simulation Algorithm.

Data: - Φ , $\{e_{-n}\}_{n\in\mathbb{N}}$
- Γ the pre-computed envelope function

Result: A state $x^* \in \mathcal{X}$ generated according to the stationary distribution of the system

```
begin
    n = 1; M := T; m := B;
    repeat
        for i = n − 1 downto 0 do
            (m, M) := Γ(m, M, e_{-i});
        n := 2n;
    until M = m ;
    x* := M;
    return x*;
end
```

Complexity: $O(\mathcal{C}(\Gamma)\tau^e)$ (to compare with $O(\mathcal{C}(\phi)N\tau^b)$).

Comments

- The definition of the envelopes is based on the constructive definition Φ of the Markov chain. For a new event representation Φ' of the Markov chain envelopes are modified accordingly.
- If the function $\Phi(., e)$ is non-decreasing for all event e, then for any $m \leq M$, $\Gamma_1(m, M, e) = \Phi(m, e)$ and $\Gamma_2(m, M, e) = \Phi(M, e)$, so that Algorithm EPSA coincides with the classical monotone perfect simulation algorithm for monotone Markov chains.

Problems

- The envelopes may not couple even if the trajectories do.
- When the envelopes couple, the coupling time of envelopes can be much longer.
- The complexity of envelope computation might be too high.
  Complexity of EPSA: $O(\mathcal{C}(\Gamma) \cdot \tau^{e})$.
  $\mathcal{C}(\Gamma)$ should not be in $\Omega(N)$.

  Everything works the same if Γ_1 (resp. Γ_2) is replaced by a lower (resp. upper) bound on the infimum (res. supremum).

Queuing networks

In most classical queuing networks events are piece-wise space homogeneous (i.e. $\phi(x,e) = x + v_R$ for x in region R) and we often have: $\mathcal{C}(\Gamma) \sim \mathcal{C}(\phi)$.
Difference between PSA and EPSA in $N\tau^b$ and τ^e.

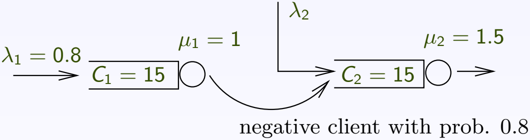


Figure: A network with negative customers.

Queuing networks (II)

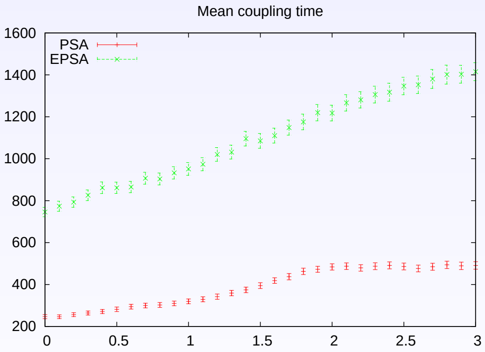


Figure: Mean coupling times of PSA and EPSA algorithms as a function of λ_2.

Queues with batches

Customers arrive and depart in batches. A new batch is accepted if there is enough space in the buffer. Otherwise the whole batch is rejected.

- Batch events are non-monotone.
  (Example. Batch arrival of size 2 in a queue of capacity C: $C-2 < C-1$ but $C > C-1$.)
- Proposition: EPSA couples if and only if batches of size 1 can happen with positive probability in each queue.
- The envelopes can be computed in constant time:

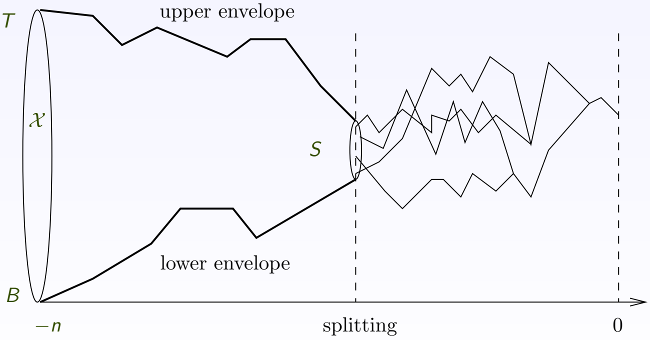
$$\Gamma_1(m, M) = \begin{cases} m+k, & M \leq C-k \\ (m+k) \wedge ((C+k-1) \vee m), & M > C-k. \end{cases}$$

$$\Gamma_2(m, M) = \begin{cases} (M+k) \wedge C, & m \leq C-k \\ M, & m > C-k \end{cases}$$

Beyond envelopes

When the coupling time for envelopes is too long (or if they do not couple):

- bounds
- splitting



Example

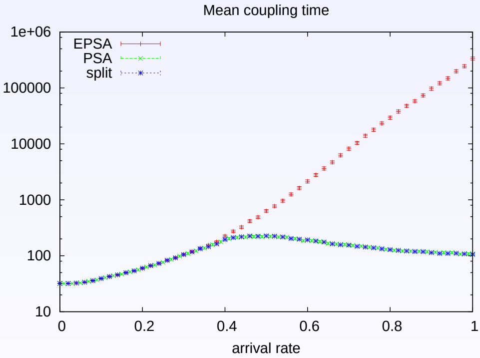


Figure: Mean coupling times for PSA, EPSA and EPSA with splitting for a $(+2, +3, -1)$ queue.

Example (II)

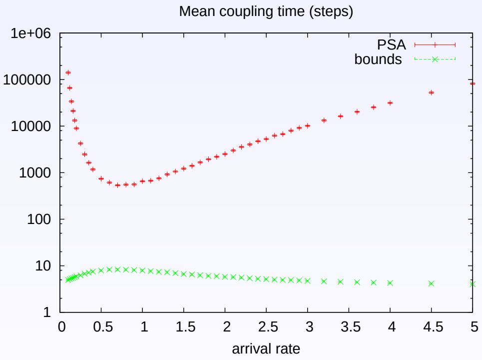
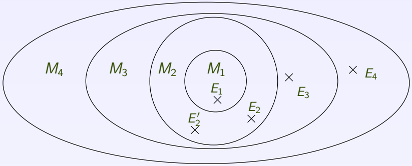


Figure: Exact values vs. bounds for a $(+4, -3)$ queue.

Classes



Classes:

- M_1 - monotone MC
- M_2 - non-monotone MC, where envelope perfect simulation can be used efficiently
- M_3 - envelopes do couple but take a much larger time
- M_4 - envelopes do not couple (bounds, splitting)

Examples:

- E_1 - a network of finite queues with monotone routing.
- E_2 - a network as E_1 with negative customers
  E_2' - a network as E_1 with fork and join nodes
- E_3 - a network with individual customers and batches
- E_4 - a network of queues with only batches larger than two.

Conclusions

A new technique to sample non-monotone discrete event systems.

Questions:

- Formally identify the classes of DES for which sup and inf are efficiently computable.
  First results: networks with index based routing with negative customers or batches, fork and join nodes.
- Study coupling time of EPSA and compare it with the coupling time of PSA.
  First results (numerical): for networks with negative customers or fork and join nodes EPSA coupling time is similar to that of PSA.
- What to do in practice?
  EPSA with splitting allows sampling of non-monotone systems very fast.

Open source software: Ψ

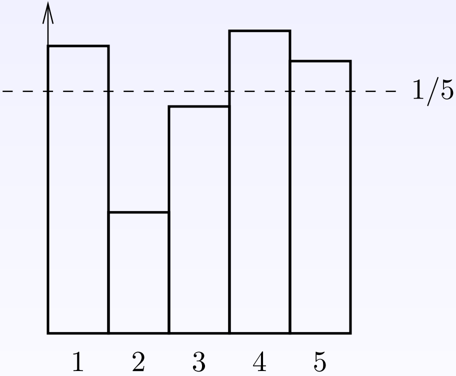
More info: `http://psi.gforge.inria.fr/website/Psi2_Unix_Website/Introduction.html`

Available: classical (Ψ) and monotone (Ψ^2) perfect sampler

Under development: EPSA sampler

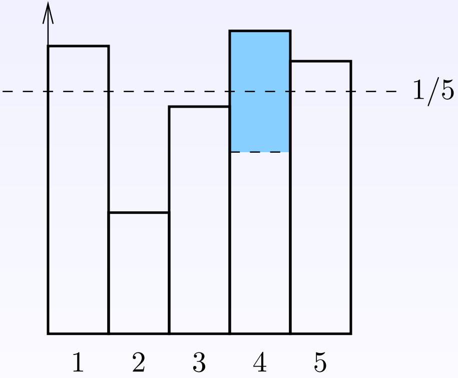
Download: `http://gforge.inria.fr/projects/psi`

Alias sampling method for discrete distributions (Walker 1977)



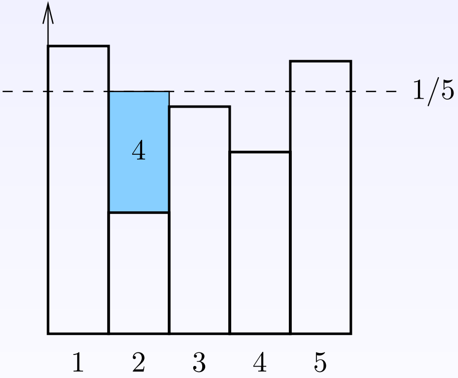
Start with a histogram of μ.

Alias sampling method for discrete distributions (Walker 1977)



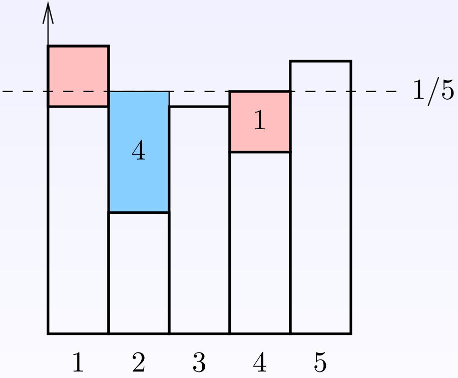
Fill the lowest column up to $1/N$ from the highest column.

Alias sampling method for discrete distributions (Walker 1977)



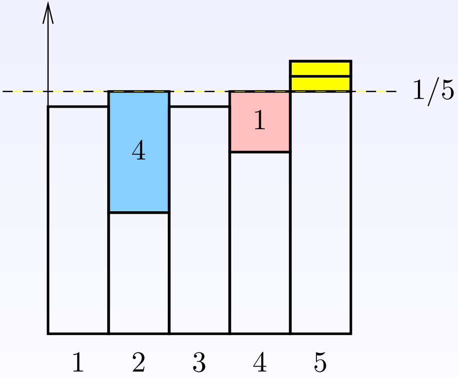
Fill the lowest column up to $1/N$ from the highest column.

Alias sampling method for discrete distributions (Walker 1977)



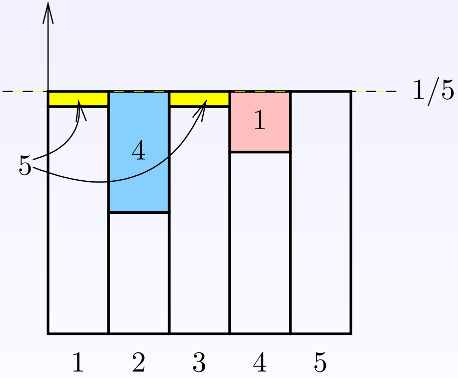
Continue until all the columns reach $1/N$. Above the threshold $S[i]$, column i corresponds to the value of $V[i]$.

Alias sampling method for discrete distributions (Walker 1977)



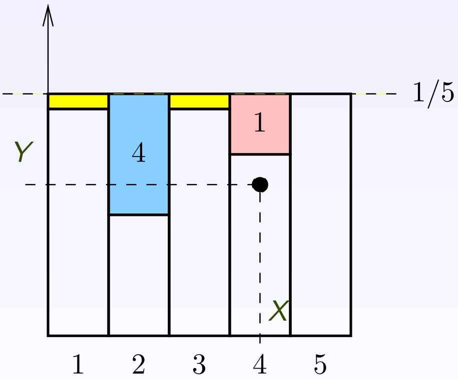
Continue until all the columns reach $1/N$. Above the threshold $S[i]$, column i corresponds to the value of $V[i]$.

Alias sampling method for discrete distributions (Walker 1977)



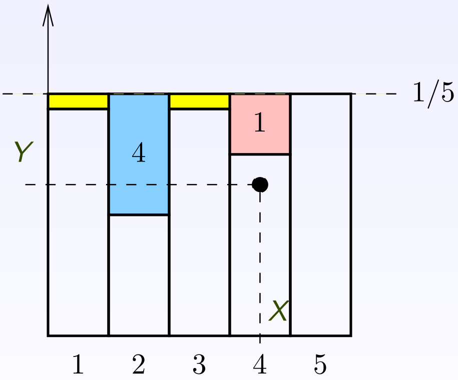
Continue until all the columns reach $1/N$. Above the threshold $S[i]$, column i corresponds to the value of $V[i]$.

Alias sampling method for discrete distributions (Walker 1977)



X and Y two uniform random variables: X in $\{0, 1, \ldots, N\}$, Y in $[0, 1/N]$. By construction of S and V, r.v. $X\mathbf{1}_{Y<S[X]} + V[X]\mathbf{1}_{Y\geq S[X]}$ has distribution μ.

Alias sampling method for discrete distributions (Walker 1977)



Once S and V are built, sampling in constant time.