Aci SurePath

Perfect simulation of finite capacity queueing networks

J-M. Vincent and B. Tanzi

Decore-Imag and Apache-Inria Projects

ID-IMAG Laboratory

Universities of Grenoble

http://www-id.imag.fr

CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE

INP Grenoble

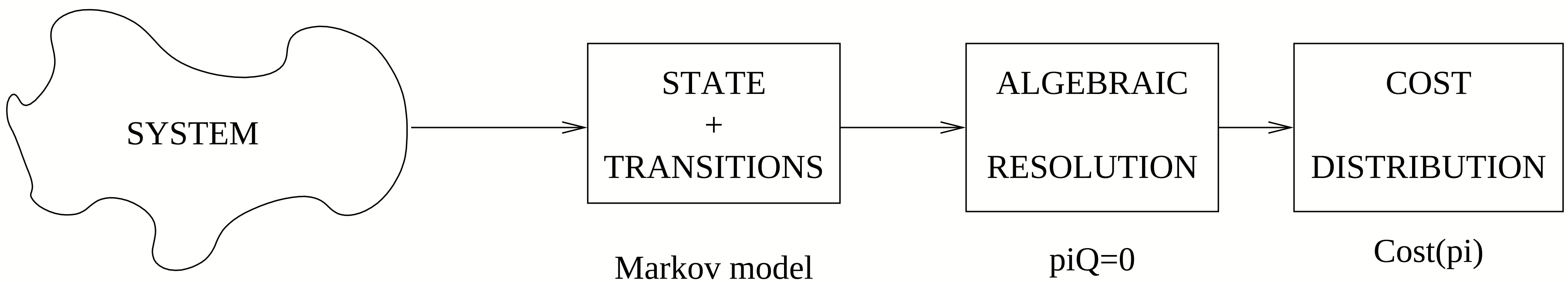
INRIA INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

GRENOBLE 1 UNIVERSITÉ JOSEPH FOURIER SCIENCES,TECHNOLOGIE,MÉDECINE

Outline

1. Motivations, simulation of Markov chains and availability
2. Markovian queueing networks
3. Perfect simulation
4. Events and monotonicity
5. PSI2 architecture
6. Examples and demo
7. Conclusion and future works

Modeling discrete event systems

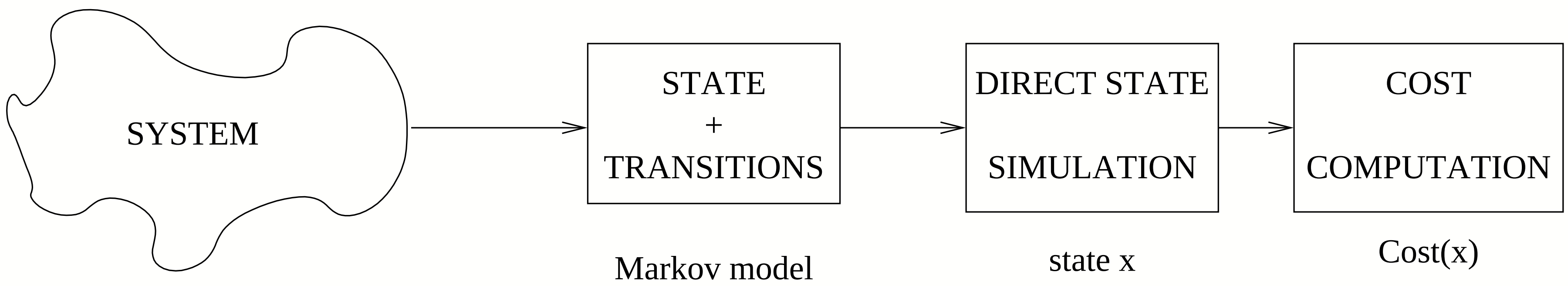


Difficulties:

- complex structure synchronizations
- rare event probability estimation
- analytical/numerical method
- approximation/bounding techniques

$\Rightarrow$ reduction of the state space

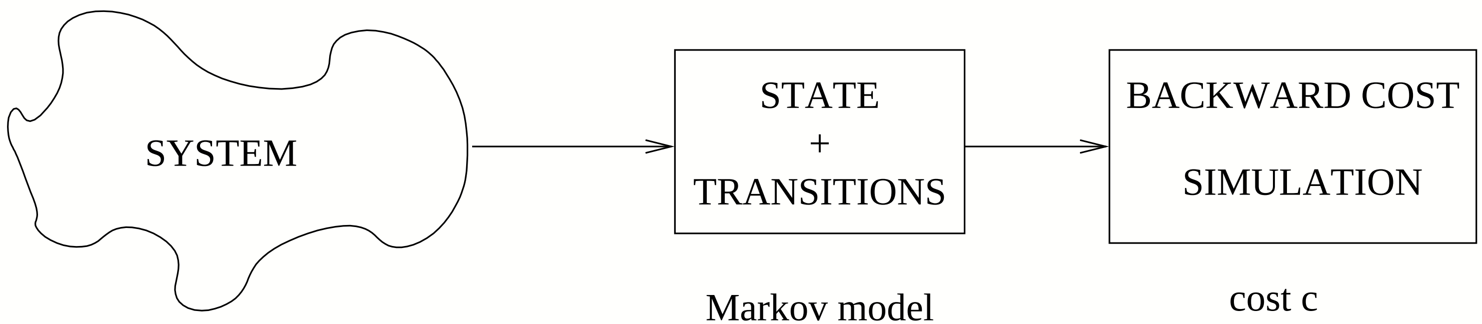
Modeling discrete event systems



Difficulties:

- stopping criteria : burn in time
- simulation biases $||\pi_n - \pi_\infty||$
- estimation biases : confidence intervals $\mathcal{O}(\frac{1}{\sqrt{n}})$

Modeling discrete event systems



Properties:

- Exact stopping criteria

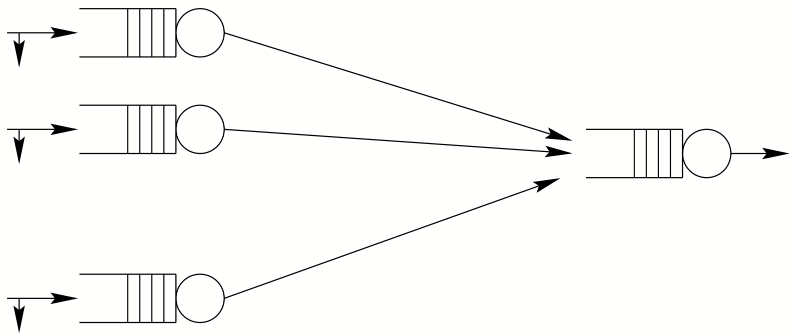
$\Rightarrow$ no simulation bias

Constraints:

- N parallel trajectories

Queueing systems models

Model of resource contention



Properties:

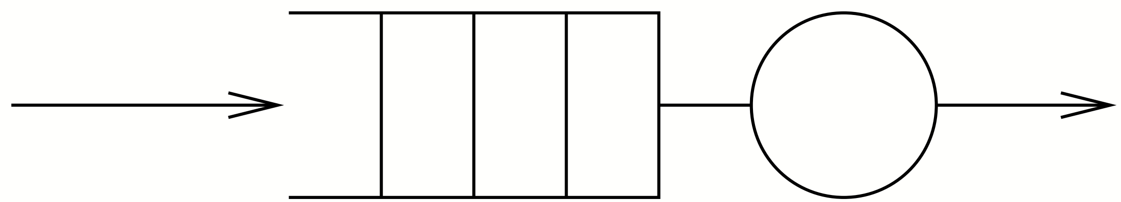
- Finite capacity queues
- Routing policies
- Blocking schemes

$\Rightarrow$ estimation of losses and saturation

$\Rightarrow$ rare events, availability

Basic queue

Basic model of resource contention :
time (server) and memory (capacity)



Markovian queues :

- arrival Poisson process λ : +1 transition
- exponential service time μ : -1 transition
- Birth and Death process

$\Rightarrow$ monotonous process

Basic queue : transition function

State space = $\{0, 1, \cdots, C\}$
Two types of events :

$$\Phi(x, arrival) = \min(x + 1, C)$$

$$\Phi(x, departure) = \max(x - 1, 0)$$

Monotonicity according to events :

$$x \leqslant y \quad \Rightarrow \quad \Phi(x, event) \leqslant \Phi(y, event)$$

Remark : link with the st-monotonicity

Network of queues : transition functio

Queueing network

- K queues, capacity C_i for queue i
- State space

$\mathcal{X} = \{0, \cdots, C_1\} \times \cdots \times \{0, \cdots, C_K\}$

Events : $e_1, \cdots, e_m$

$$\Phi(x, event) = next\ \ state$$

Iterated system of functions

$\mathcal{X}$ state space (size n); $\mathcal{U}$ set of external input values
Transition function Φ

$$\begin{array}{rccc} \Phi : & \mathcal{X} \times \mathcal{U} & \longrightarrow & \mathcal{X} \\ & (x,u) & \longmapsto & \Phi(x,u) \end{array}$$

If $\{U_n\}_{n\in\mathbb{Z}}$ is IID then

$$X_0 = x_0, \quad X_{n+1} = \Phi(X_n, U_{n+1})$$

is a Markov chain (stochastic recursive sequence).

$$\{\Phi(.,u)\}_{u\in[0,1[}$$

is an iterated system of function.
Reciprocally, given a transition matrix Q it is possible to build a family of function $\Phi(.,u)$ such that the associated process is a markov chain with transition matrix Q.
$\Longrightarrow$ Simulation kernel

Backward coupling simulation

Idea :

Propp & Wilson(1996)

- reverse time
- run N parallel trajectories
- wait for coupling.

$$\mathcal{Z}_n = \Phi(\Phi(\cdots(\Phi(\mathcal{X}, U_{-n+1}), \cdots), U_{-1}), U_0).$$

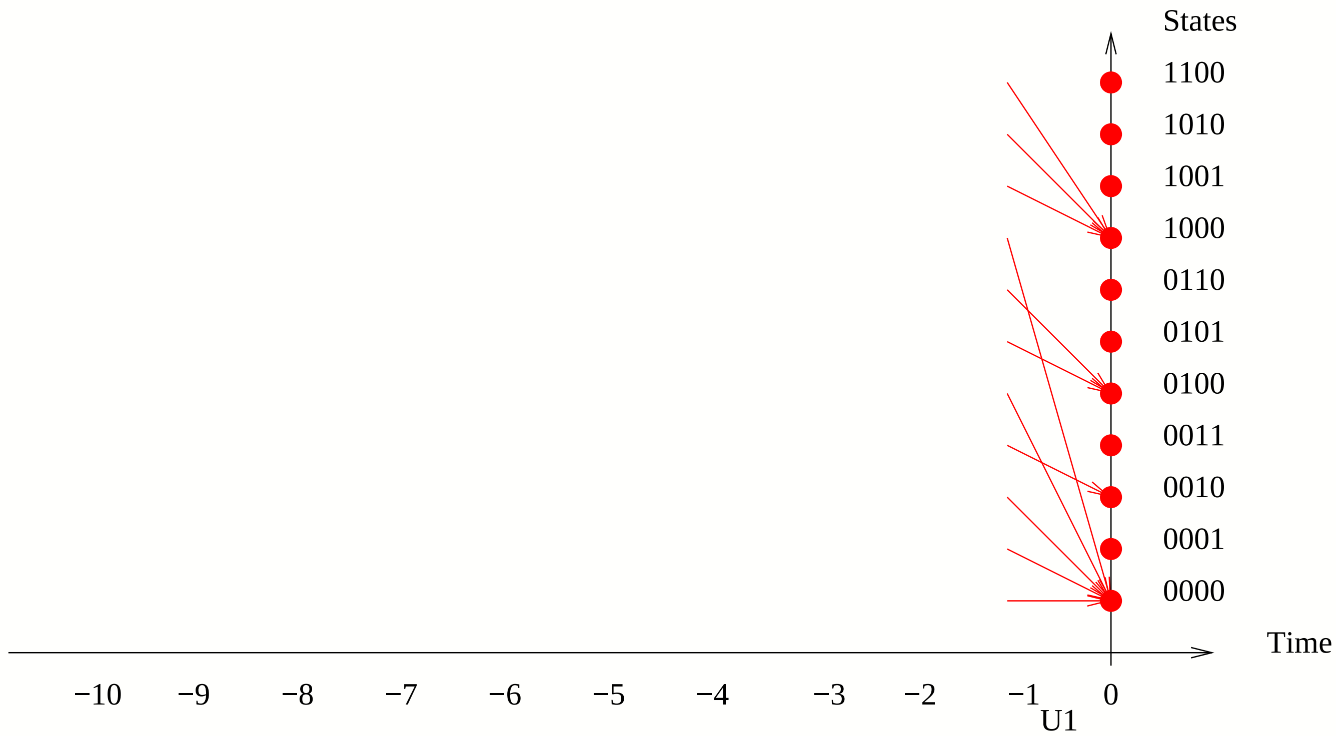
potential set of reachable states at step n

for all $x \in \mathcal{X}$ **do**
  $y(x) \leftarrow x$
end for
repeat
  u $\leftarrow$ Random;
  for all $x \in \mathcal{X}$ **do**
    $y(x) \leftarrow y(\Phi(x, u))$;
  end for
until All $y(x)$ are equal
return $y(x)$

Backward simulation example

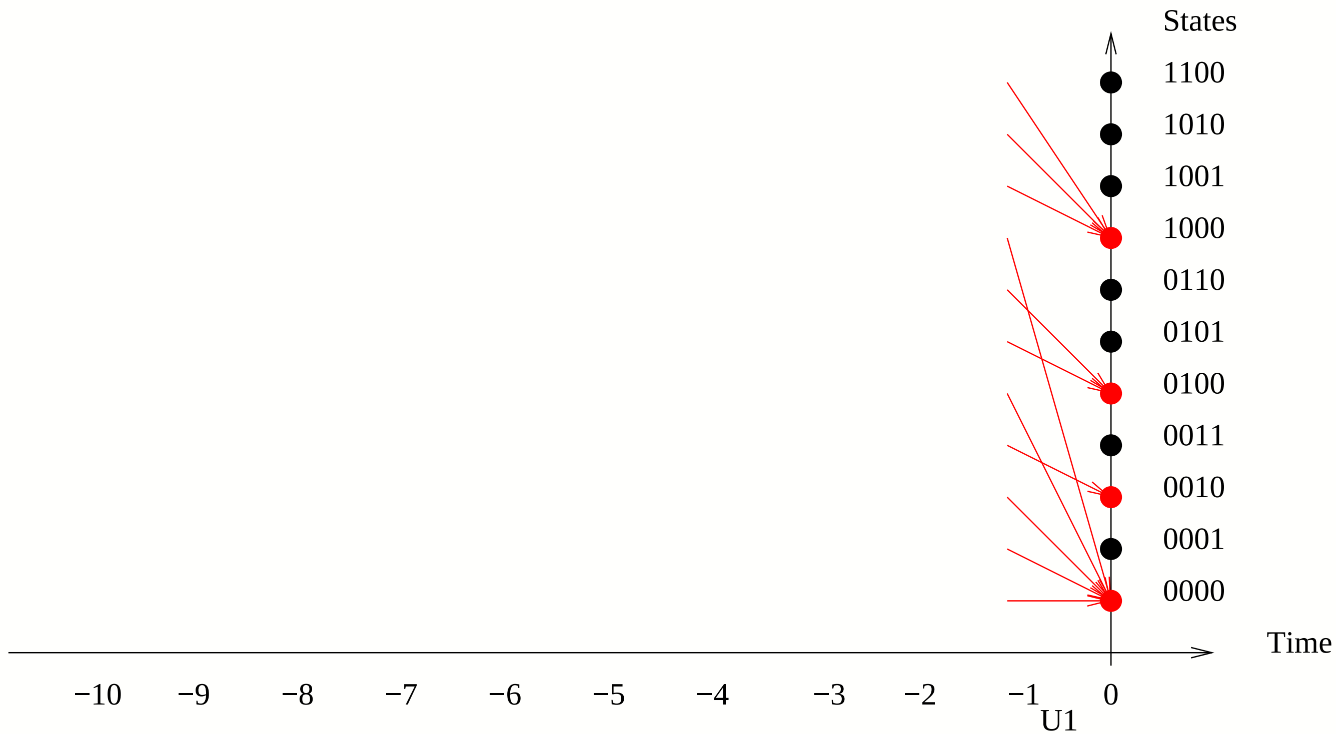
$\mathcal{Z}_0 = \mathcal{X}$

Backward simulation example



$\mathcal{Z}_0 = \mathcal{X}$

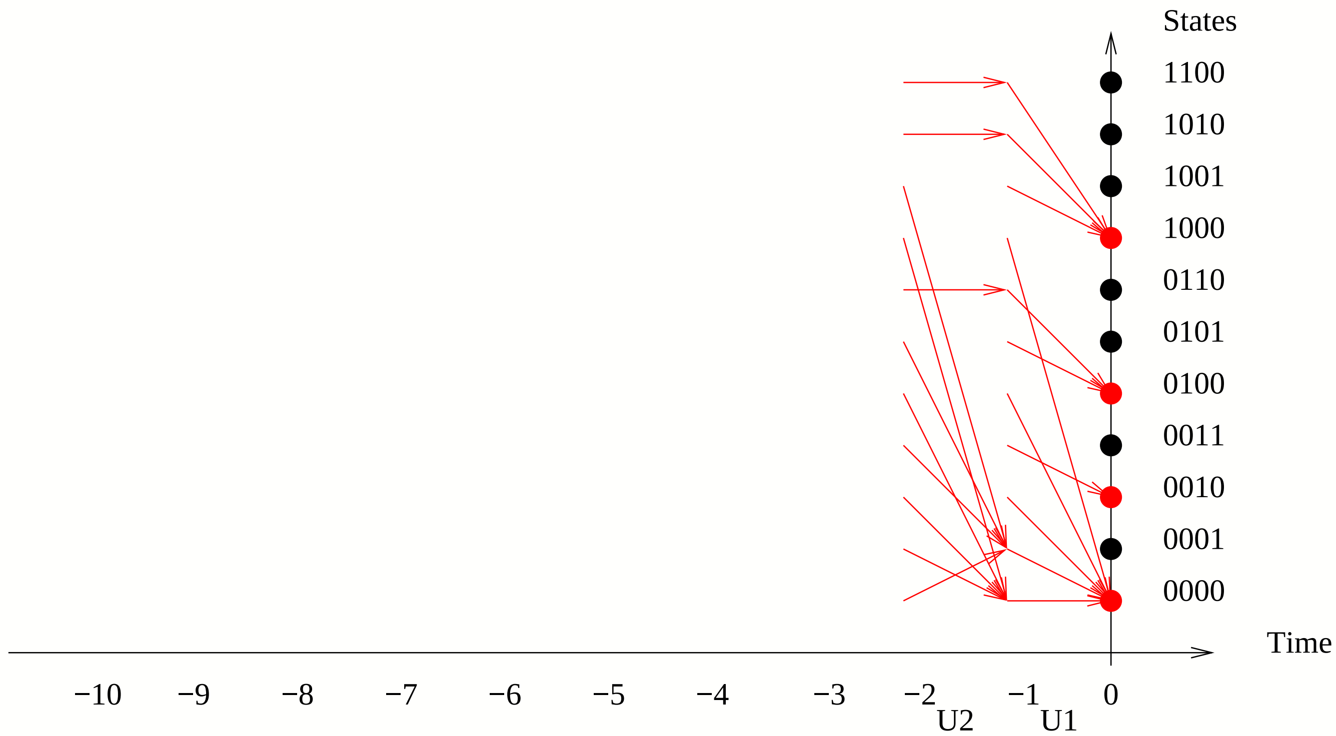
Backward simulation example



$\mathcal{Z}_0 = \mathcal{X}$

$\mathcal{Z}_1 = \{0000, 0010, 0100, 1000\}$

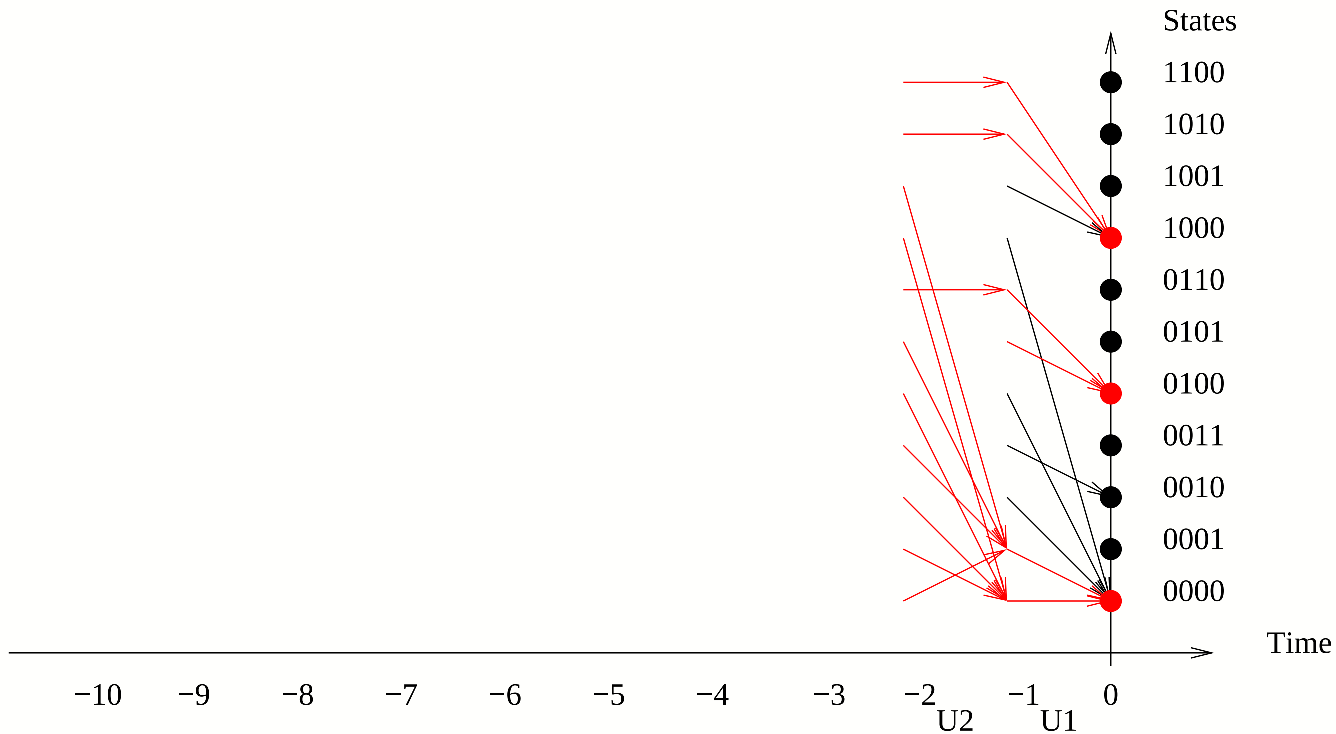
Backward simulation example



$\mathcal{Z}_0 = \mathcal{X}$

$\mathcal{Z}_1 = \{0000, 0010, 0100, 1000\}$

Backward simulation example

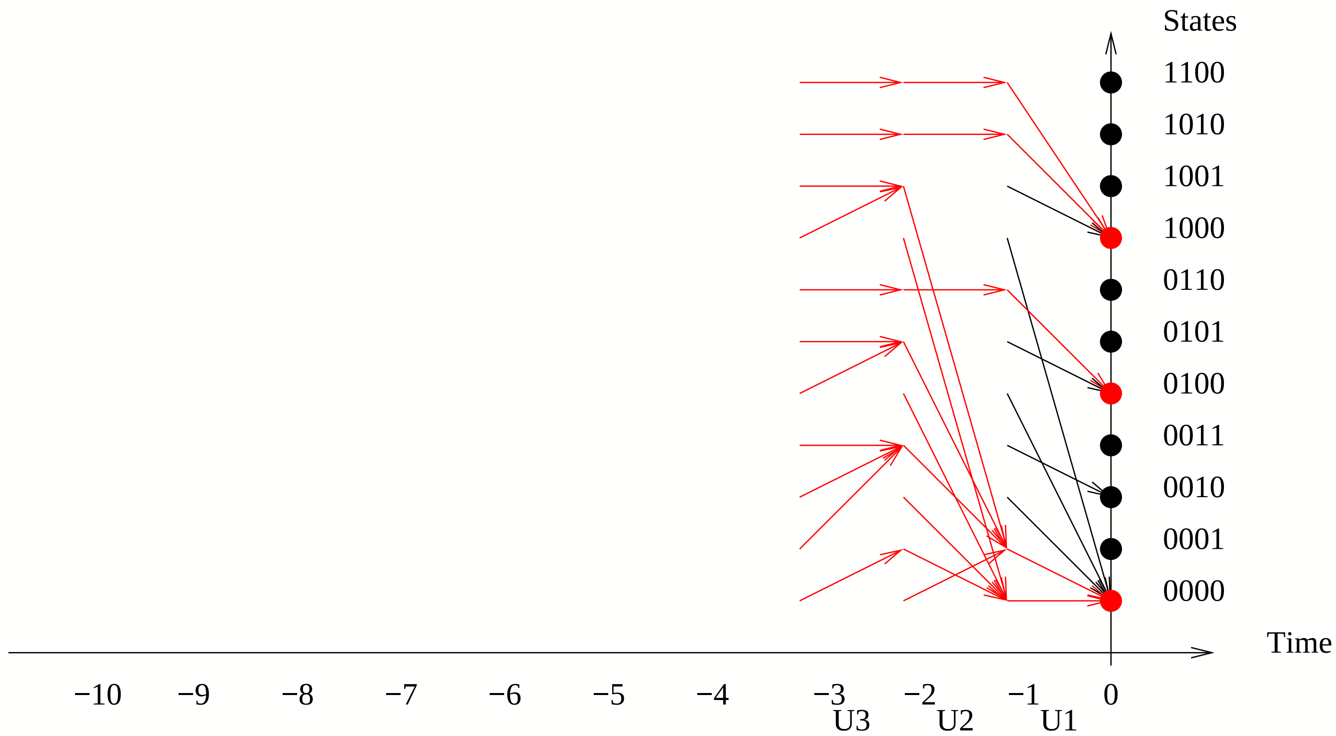


$$\mathcal{Z}_0 = \mathcal{X}$$

$$\mathcal{Z}_1 = \{0000, 0010, 0100, 1000\}$$

$$\mathcal{Z}_2 = \{0000, 0100, 1000\}$$

Backward simulation example

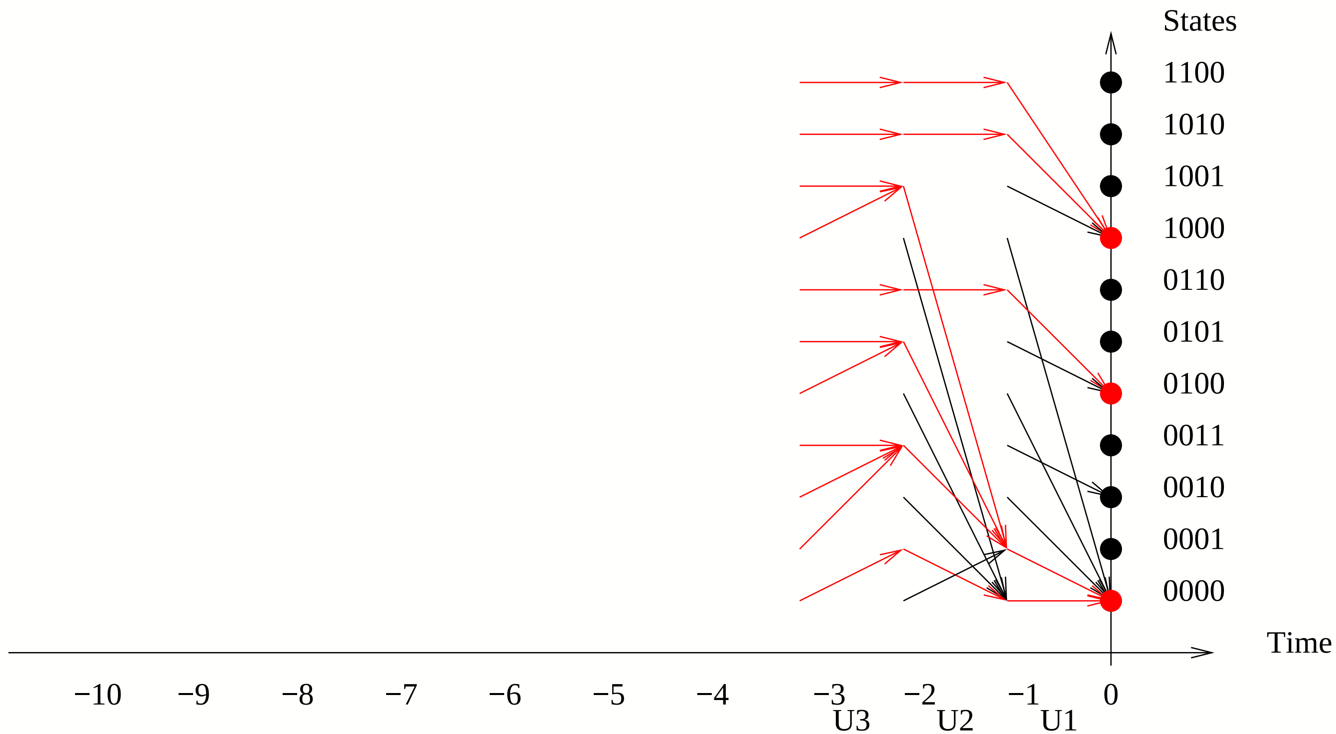


$$\mathcal{Z}_0 = \mathcal{X}$$

$$\mathcal{Z}_1 = \{0000, 0010, 0100, 1000\}$$

$$\mathcal{Z}_2 = \{0000, 0100, 1000\}$$

Backward simulation example



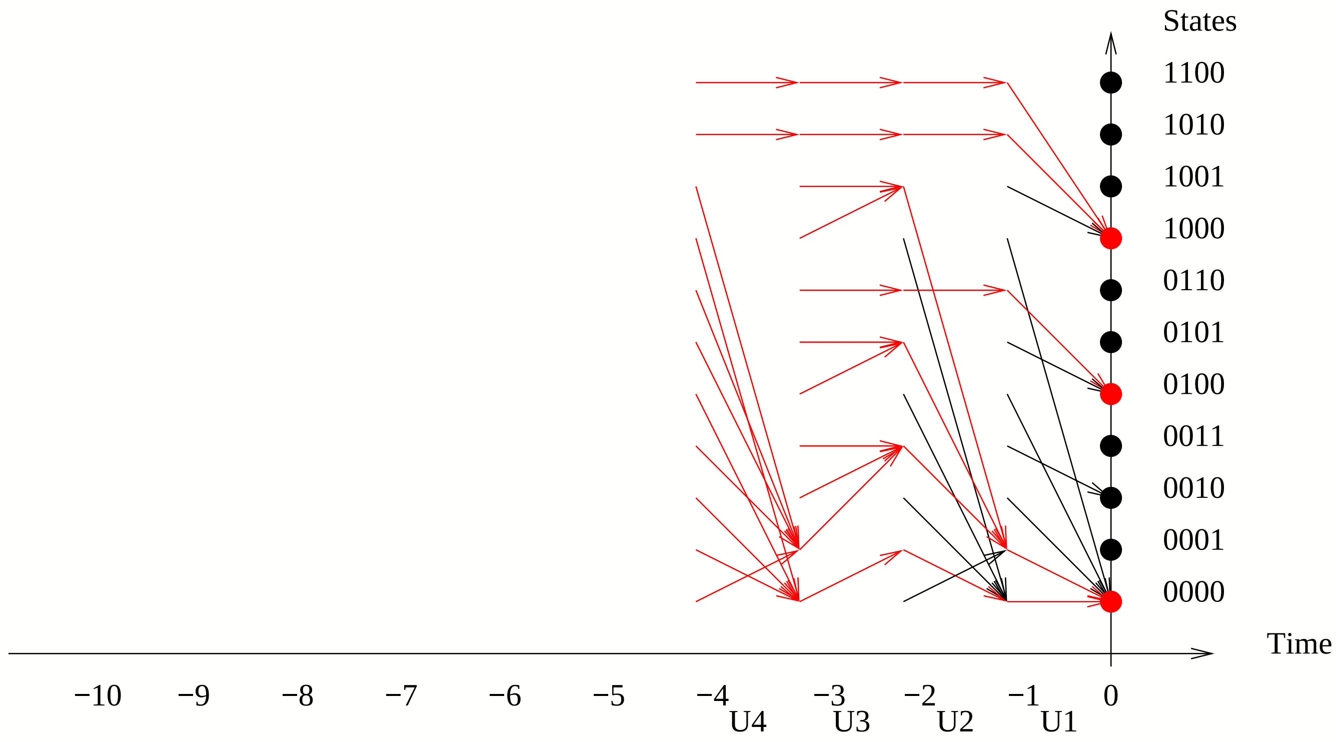
$\mathcal{Z}_0 = \mathcal{X}$

$\mathcal{Z}_1 = \{0000, 0010, 0100, 1000\}$

$\mathcal{Z}_2 = \{0000, 0100, 1000\}$

$\mathcal{Z}_3 = \{0000, 0100, 1000\}$

Backward simulation example



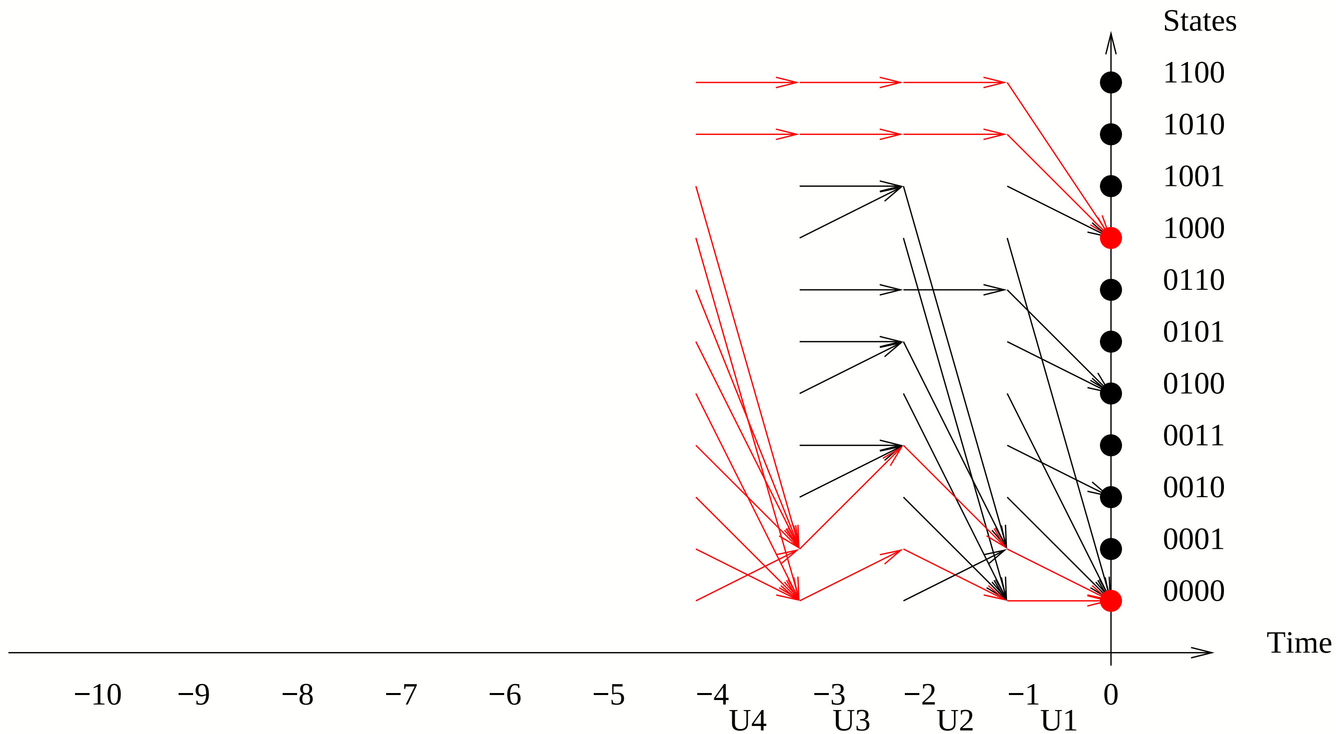
$\mathcal{Z}_0 = \mathcal{X}$

$\mathcal{Z}_1 = \{0000, 0010, 0100, 1000\}$

$\mathcal{Z}_2 = \{0000, 0100, 1000\}$

$\mathcal{Z}_3 = \{0000, 0100, 1000\}$

Backward simulation example



$$\mathcal{Z}_0 = \mathcal{X}$$

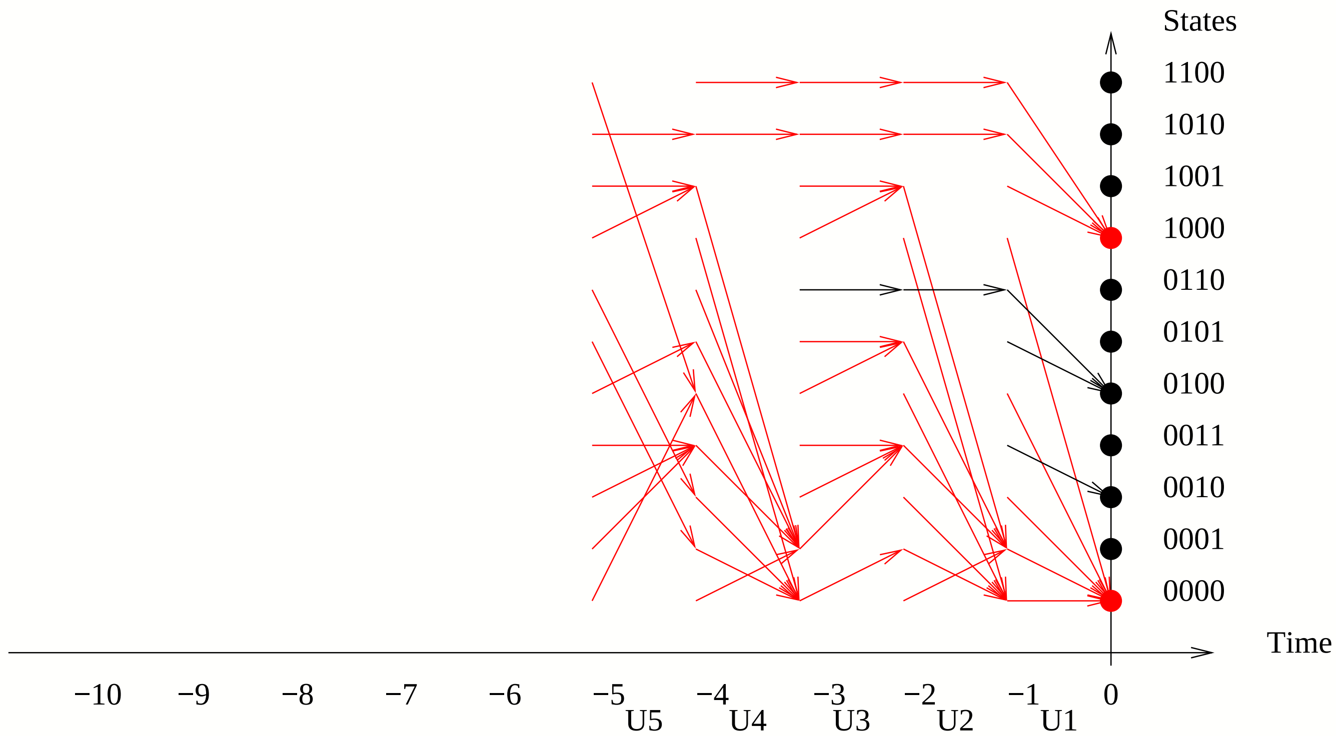
$$\mathcal{Z}_1 = \{0000, 0010, 0100, 1000\}$$

$$\mathcal{Z}_2 = \{0000, 0100, 1000\}$$

$$\mathcal{Z}_3 = \{0000, 0100, 1000\}$$

$$\mathcal{Z}_4 = \{0000, 1000\}$$

Backward simulation example

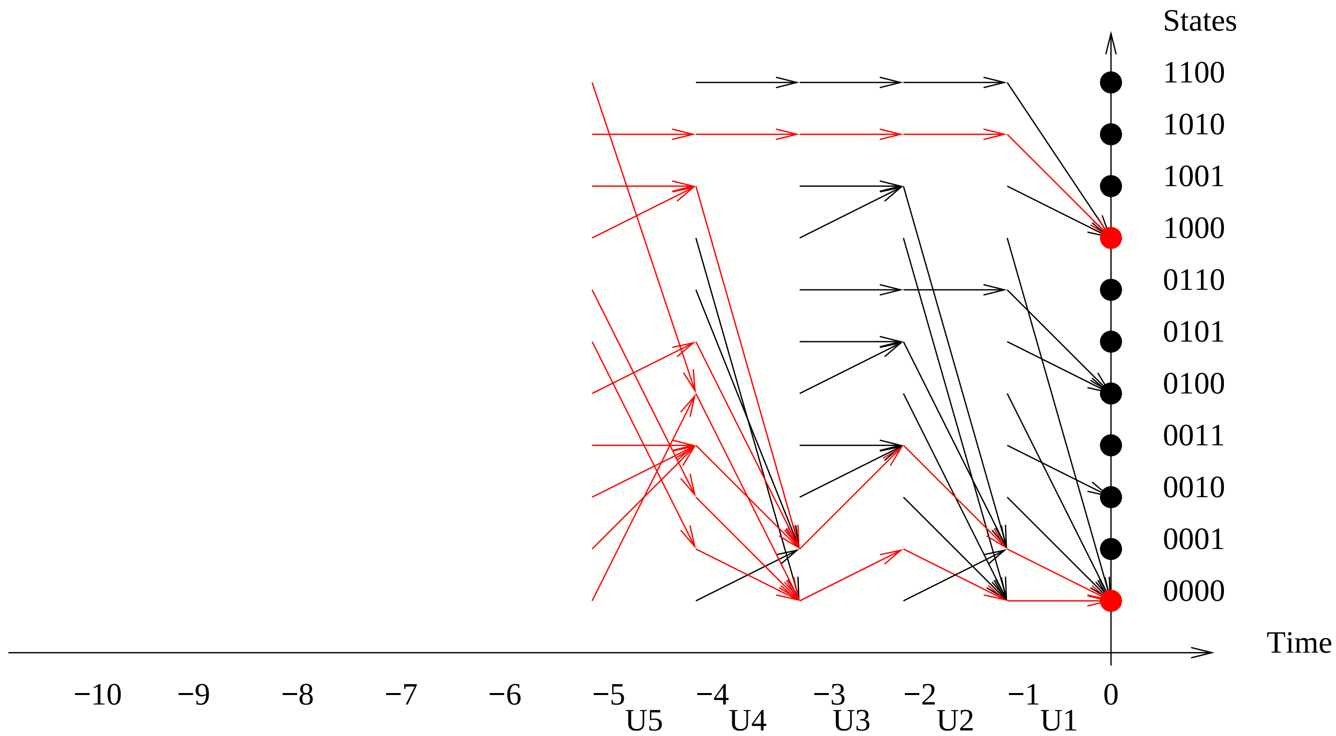


$$\mathcal{Z}_0 = \mathcal{X} \qquad \mathcal{Z}_3 = \{0000, 0100, 1000\}$$

$$\mathcal{Z}_1 = \{0000, 0010, 0100, 1000\} \qquad \mathcal{Z}_4 = \{0000, 1000\}$$

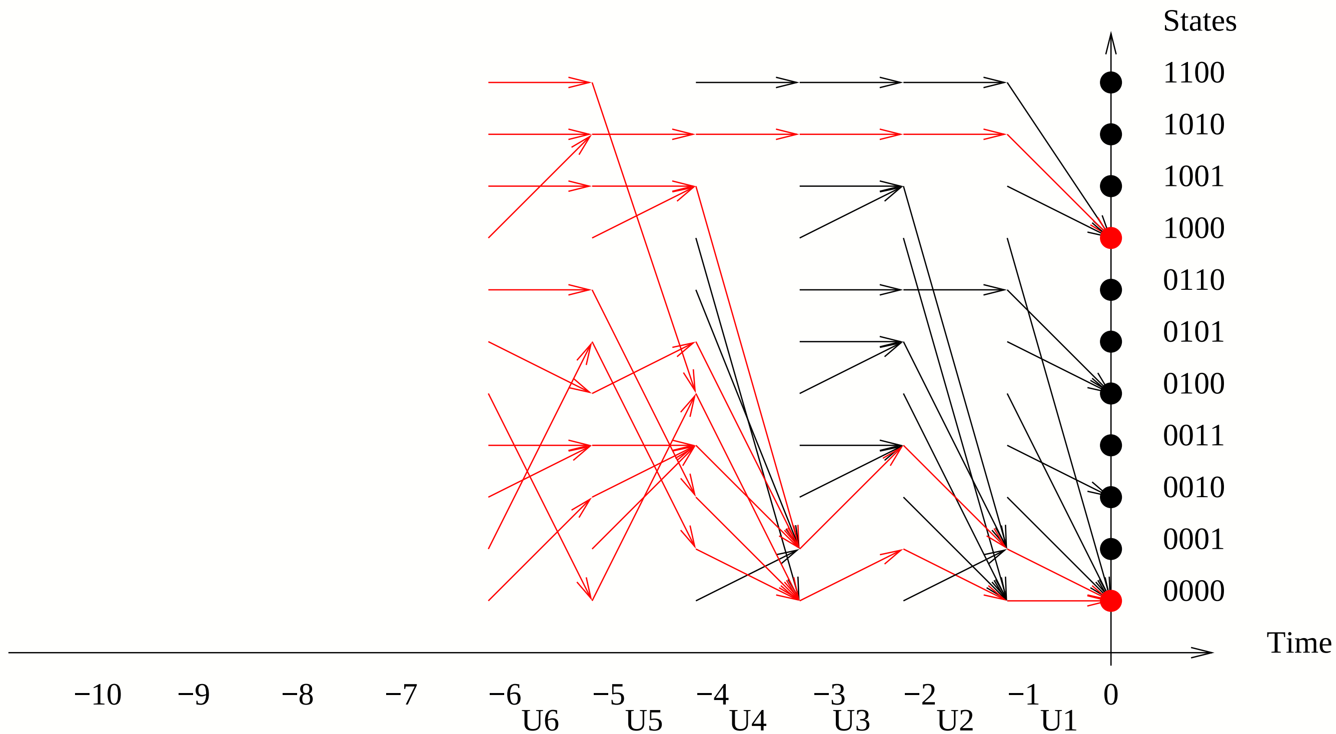
$$\mathcal{Z}_2 = \{0000, 0100, 1000\}$$

Backward simulation example



$$\mathcal{Z}_0 = \mathcal{X}$$
$$\mathcal{Z}_1 = \{0000, 0010, 0100, 1000\}$$
$$\mathcal{Z}_2 = \{0000, 0100, 1000\}$$
$$\mathcal{Z}_3 = \{0000, 0100, 1000\}$$
$$\mathcal{Z}_4 = \{0000, 1000\}$$
$$\mathcal{Z}_5 = \{0000, 1000\}$$

Backward simulation example



$\mathcal{Z}_0 = \mathcal{X}$

$\mathcal{Z}_1 = \{0000, 0010, 0100, 1000\}$

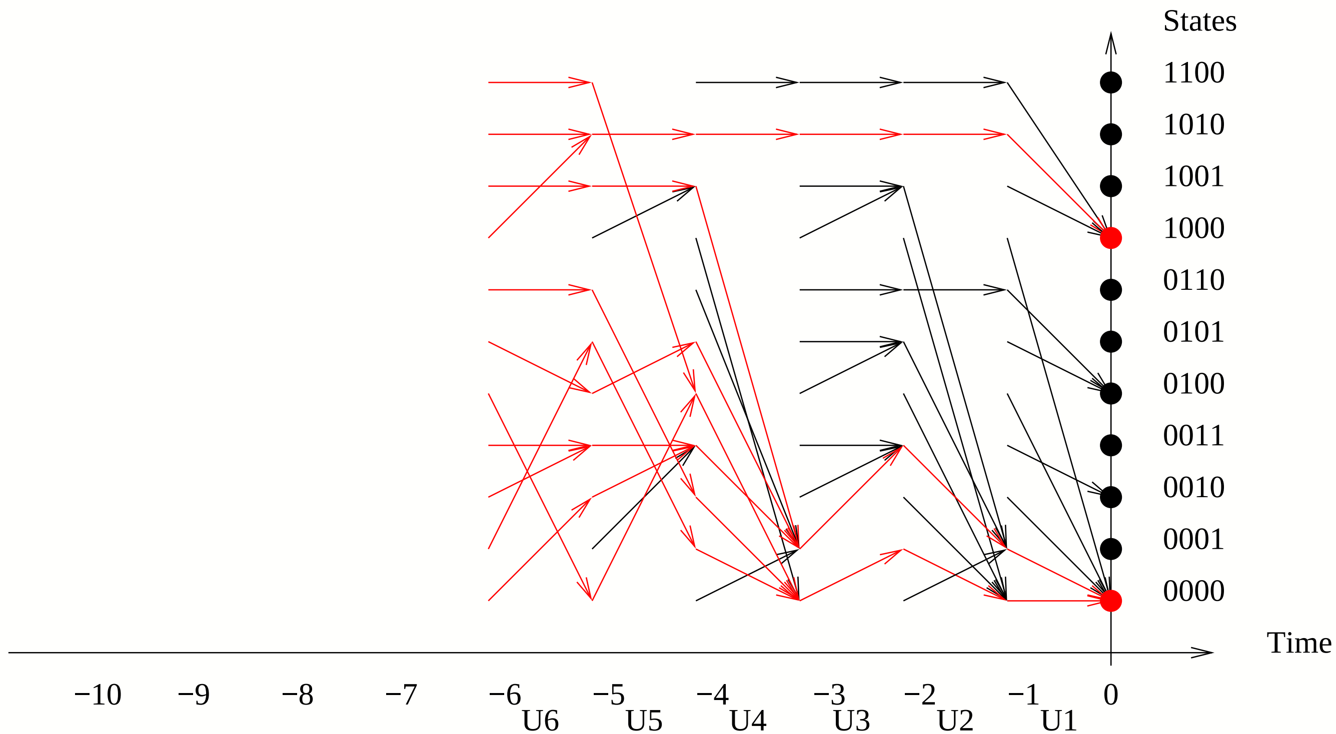
$\mathcal{Z}_2 = \{0000, 0100, 1000\}$

$\mathcal{Z}_3 = \{0000, 0100, 1000\}$

$\mathcal{Z}_4 = \{0000, 1000\}$

$\mathcal{Z}_5 = \{0000, 1000\}$

Backward simulation example



$\mathcal{Z}_0 = \mathcal{X}$

$\mathcal{Z}_1 = \{0000, 0010, 0100, 1000\}$

$\mathcal{Z}_2 = \{0000, 0100, 1000\}$

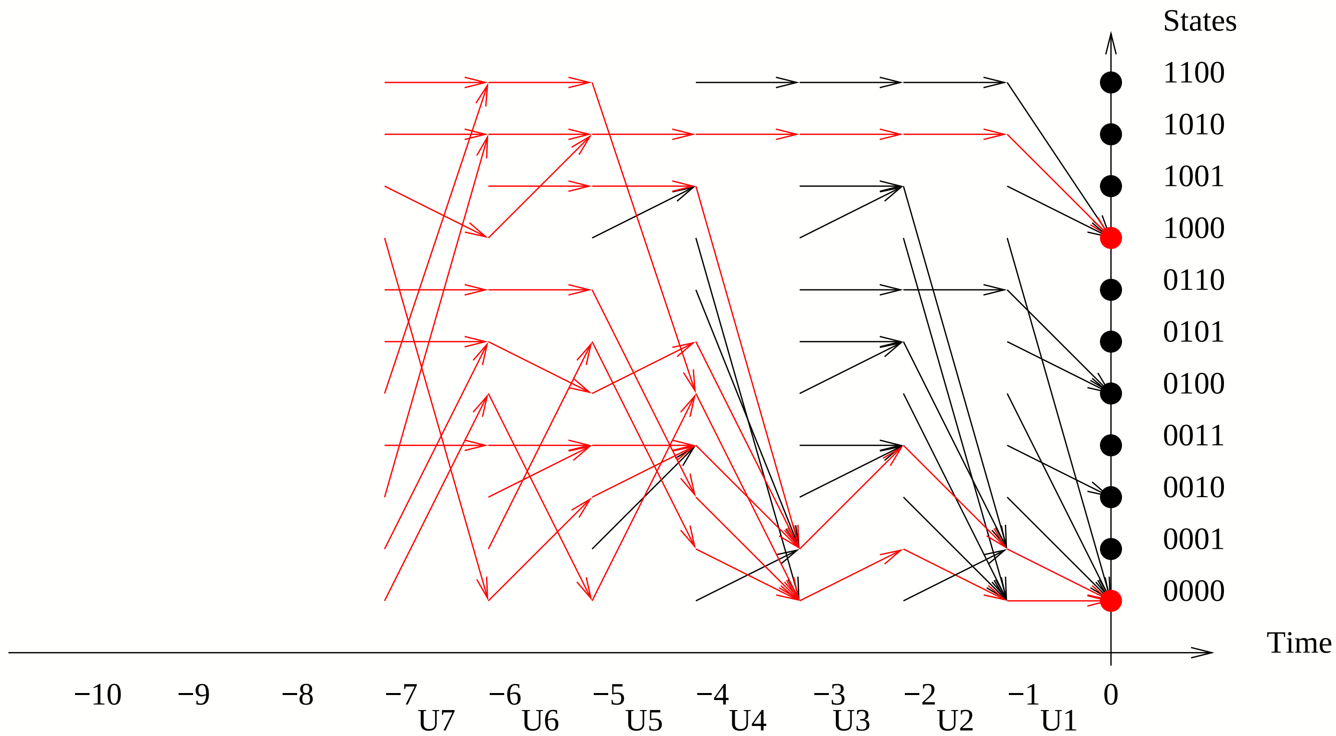
$\mathcal{Z}_3 = \{0000, 0100, 1000\}$

$\mathcal{Z}_4 = \{0000, 1000\}$

$\mathcal{Z}_5 = \{0000, 1000\}$

$\mathcal{Z}_6 = \{0000, 1000\}$

Backward simulation example



$\mathcal{Z}_0 = \mathcal{X}$

$\mathcal{Z}_1 = \{0000, 0010, 0100, 1000\}$

$\mathcal{Z}_2 = \{0000, 0100, 1000\}$

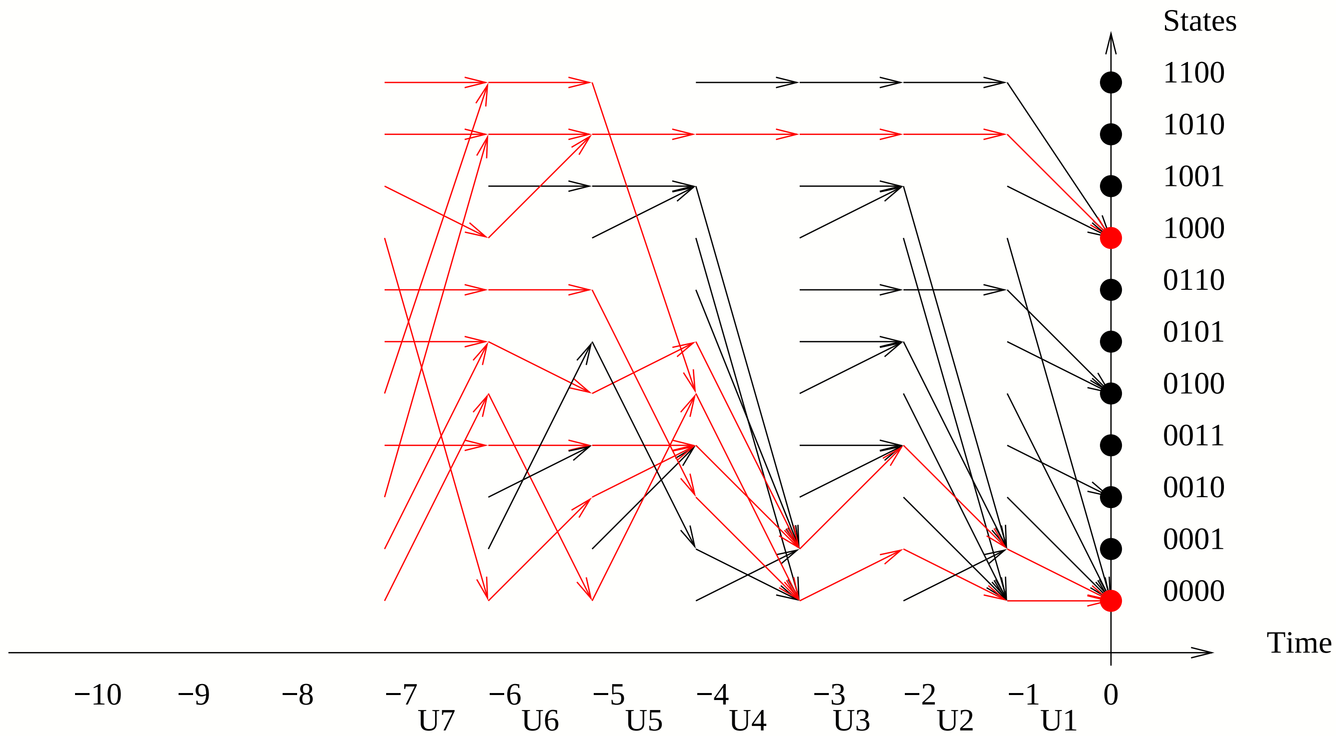
$\mathcal{Z}_3 = \{0000, 0100, 1000\}$

$\mathcal{Z}_4 = \{0000, 1000\}$

$\mathcal{Z}_5 = \{0000, 1000\}$

$\mathcal{Z}_6 = \{0000, 1000\}$

Backward simulation example



$\mathcal{Z}_0 = \mathcal{X}$

$\mathcal{Z}_1 = \{0000, 0010, 0100, 1000\}$

$\mathcal{Z}_2 = \{0000, 0100, 1000\}$

$\mathcal{Z}_3 = \{0000, 0100, 1000\}$

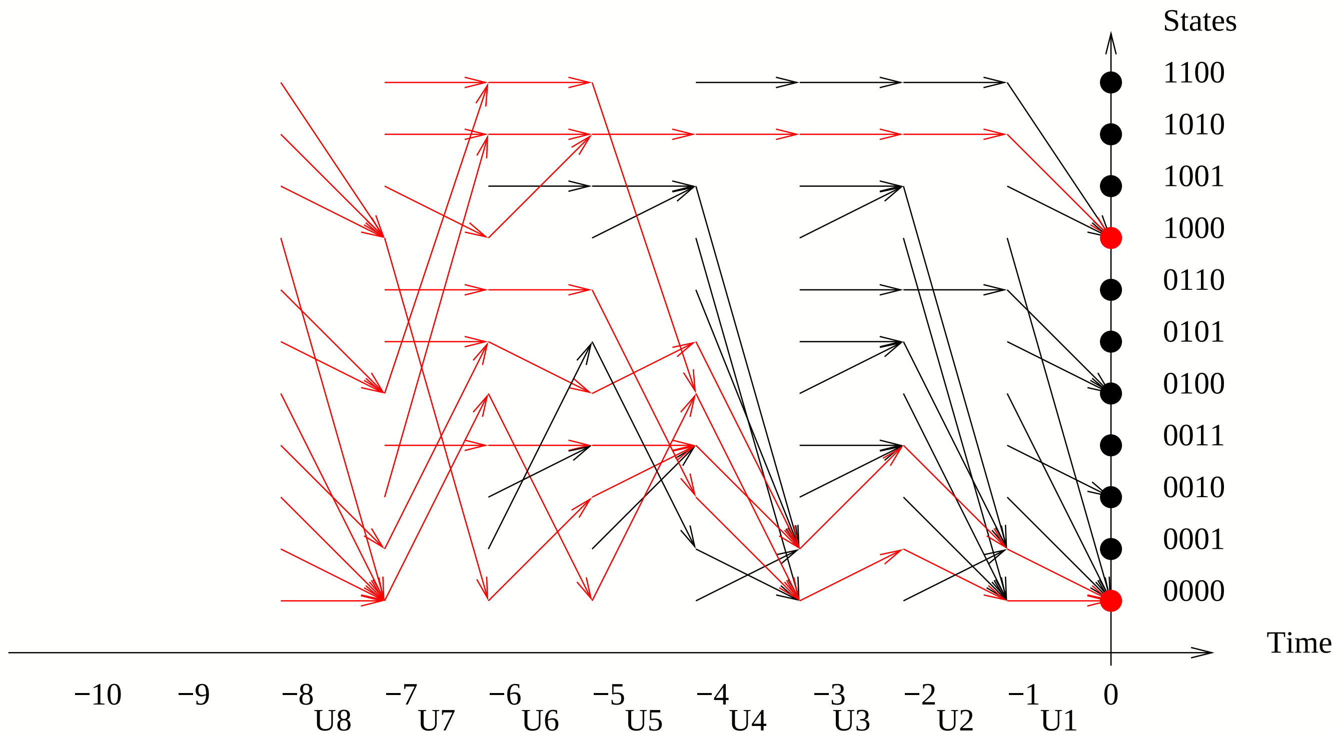
$\mathcal{Z}_4 = \{0000, 1000\}$

$\mathcal{Z}_5 = \{0000, 1000\}$

$\mathcal{Z}_6 = \{0000, 1000\}$

$\mathcal{Z}_7 = \{0000, 1000\}$

Backward simulation example



$\mathcal{Z}_0 = \mathcal{X}$

$\mathcal{Z}_1 = \{0000, 0010, 0100, 1000\}$

$\mathcal{Z}_2 = \{0000, 0100, 1000\}$

$\mathcal{Z}_3 = \{0000, 0100, 1000\}$

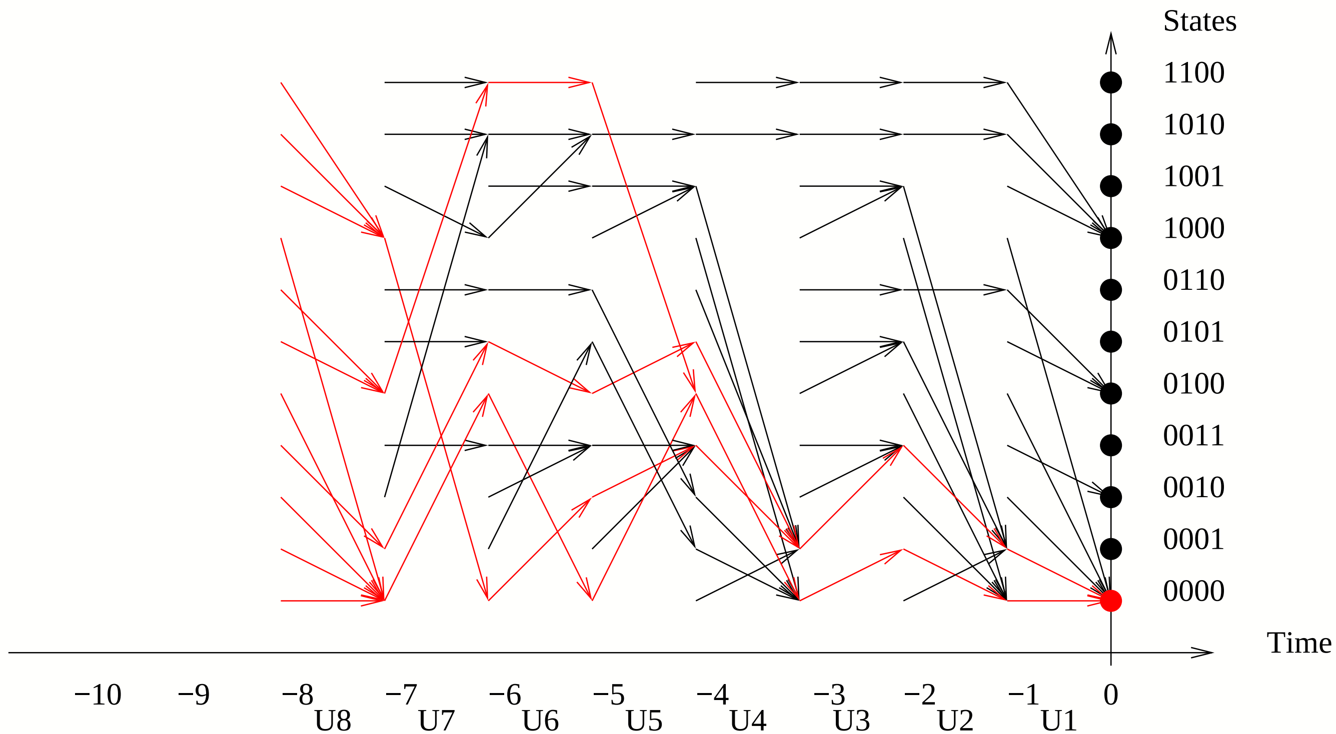
$\mathcal{Z}_4 = \{0000, 1000\}$

$\mathcal{Z}_5 = \{0000, 1000\}$

$\mathcal{Z}_6 = \{0000, 1000\}$

$\mathcal{Z}_7 = \{0000, 1000\}$

Backward simulation example



$\mathcal{Z}_0 = \mathcal{X}$

$\mathcal{Z}_1 = \{0000, 0010, 0100, 1000\}$

$\mathcal{Z}_2 = \{0000, 0100, 1000\}$

$\mathcal{Z}_3 = \{0000, 0100, 1000\}$

$\mathcal{Z}_4 = \{0000, 1000\}$

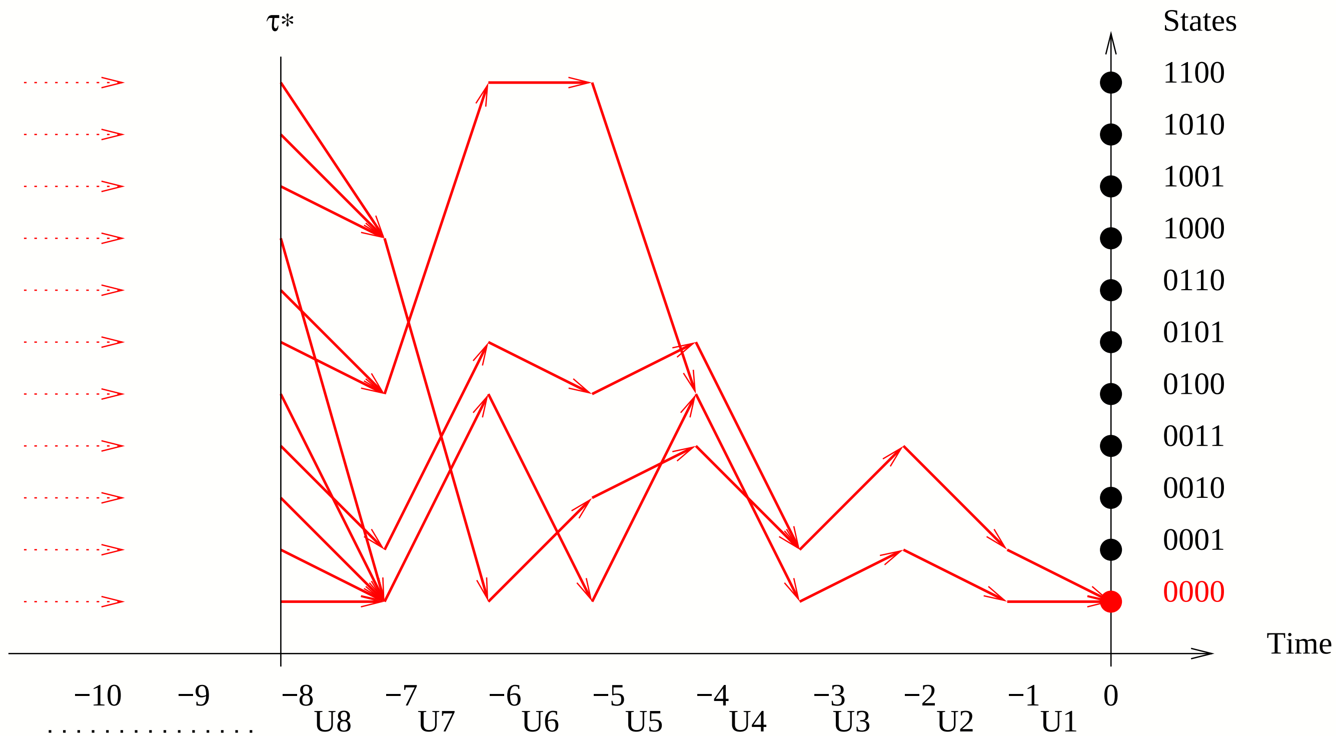
$\mathcal{Z}_5 = \{0000, 1000\}$

$\mathcal{Z}_6 = \{0000, 1000\}$

$\mathcal{Z}_7 = \{0000, 1000\}$

$\mathcal{Z}_8 = \{0000\}$

Backward simulation example



Process stops when $|\mathcal{Z}_n| = 1$

Stopping time $\tau^* = 8$

Number of computation of Φ (complexity) : $n.\tau^*$

Backward coupling simulation

Proposition 1 (Propp & Wilson) *If $\tau^* < +\infty$ a.s. then the returned value is stationary distributed.*

Proof :

$\{\mathcal{Z}_n\}_{n\in\mathbb{N}}$ is non-increasing and constant for n sufficiently large

$\Phi(\Phi(\cdots(\Phi(\mathcal{X}, U_{-n+1}), \cdots), U_{-1}), U_0) \overset{\mathcal{L}}{\sim} \Phi(\Phi(\cdots(\Phi(\mathcal{X}, U_1), \cdots), U_{n-1}), U_n)$

Remarks :

Stopping times τ and τ^* have the same law,

τ depends on Φ coding (not only on the transition matrix !)

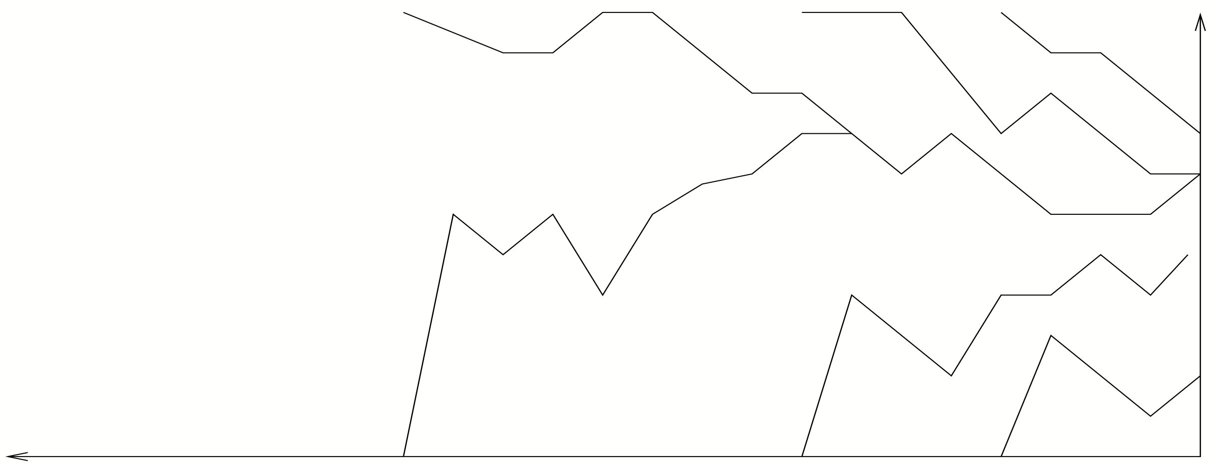
$\Rightarrow$ optimization problem

Partially ordered state space

$\mathcal{X}$ is partially ordered :

- set M of maximum
- set m of minimum
- example : vector and componentwise comparison

The global scheme is



Propp & Wilson : double period of simulation

Fill : interruptible forward algorithm

Monotonous Perfect

MONOTONICITY:

Propp & Wilson(1996)

- reverse time
- run parallel trajectories $|M| + |m|$
- wait for coupling.

$$\mathcal{Z}_n = \Phi(\Phi(\cdots(\Phi(M \cup m, U_{-n+1}), \cdots), U_{-1}), U_0).$$

potential set of reachable states from maxima and minima at time n.

n=0;
repeat
  n=n+1;
  R[n]=Random;
  for all $x \in M \cup m$ **do**
    $y(x) \leftarrow x$
  end for
  for i=n downto 1 **do**
    for all $x \in M \cup m$ **do**
      $y(x) \leftarrow \Phi(y(x), R[i])$
    end for
  end for
until All $y(x)$ are equal
return $y(x)$

Monotonous Perfect

MONOTONICITY:

Propp & Wilson(1996)

- reverse time
- run parallel trajectories $|M| + |m|$
- wait for coupling.

$$\mathcal{Z}_n = \Phi(\Phi(\cdots(\Phi(M \cup m, U_{-n+1}), \cdots), U_{-1}), U_0).$$

potential set of reachable states from maxima and minima at time n.

Optimal coupling time : complexity $\mathcal{O}(\mathbb{E}\tau)$

Storage of random sequence

n=1
repeat
  n=2n;
  for all $x \in M \cup m$ **do**
    $y(x) \leftarrow x$
  end for
  for i=n downto n/2 **do**
    R[i]=Random;
  end for
  for i=n downto 0 **do**
    for all $x \in M \cup m$ **do**
      $y(x) \leftarrow \Phi(y(x), R[i])$
    end for
  end for
until All $y(x)$ are equal
return $y(x)$

Monotone routing (1)

Queueing network

- K queues, capacity C_i for queue i
- State space $\mathcal{X} = \{0, \cdots, C_1\} \times \cdots \times \{0, \cdots, C_K\}$
- event e

Routing strategy (overflow):

- origin queue i
- destination list $j_1, \cdots, j_k$
- rate λ_e

according to a state x event e route a customer from queue i to the first non-full queue in the list of destinations. If all destinations are full the customer is routed out of the network (rejection).

Proposition 2 *The event e routing with rejection is monotonous.*

$$x \leqslant y \quad \Rightarrow \quad \Phi(x, e) \leqslant \Phi(y, e)$$

- If queue i is empty, nothing is done

- Arrival is a routing with overflow

Monotone routing (2)

Routing strategy (blocking):

- origin queue i
- destination list $j_1, \cdots, j_k$
- rate λ_e

according to a state x event e route a customer from queue i to the first non-full queue in the list of destinations. If all destinations are full the customer stay in its queue and run its service again.

Proposition 3 *The event e routing with blocking is monotonous.*

$$x \leqslant y \quad \Rightarrow \quad \Phi(x, e) \leqslant \Phi(y, e)$$

- If queue i is empty, nothing is done
- destination list $j_1, \cdots, j_k, i$

Monotone routing (3)

Routing strategy (JSQ):
- origin queue i
- destination list $j_1, \cdots, j_k$
- rate λ_e

according to a state x event e route a customer from queue i to the lowest non-full queue in the list of destinations. If all destinations are full the customer stay in its queue and run its service again or is rejected from the network.

Proposition 4 *The event e routing with JSQ policy is monotonous.*

$$x \leqslant y \quad \Rightarrow \quad \Phi(x, e) \leqslant \Phi(y, e)$$

- If queue i is empty, nothing is done

Uniformization

Each event is driven by a Poisson process

- λ_j rate of event e_j

Define $\Lambda = \sum \lambda_i$ the uniformized process with rate Λ

With probability $\frac{\lambda_i}{\Lambda}$ event e_i occurs, if e_i is not admissible the transition is skipped.

Theorem 1 *Markovian queueing networks with monotone routing policies have an uniformized version which is monotonous.*

$\Rightarrow$ Monotonous Perfect Simulation

General architecture of $\Psi 2$

Model description specification : textual file

Coupling function : C file (optional)

Simulation parameters : textual file

Unix-like command : psi2_unix -i model.txt -o result.txt -p param.txt -c couplage.c

Example queue : model

```
#nombre_de_files
1
#capacite_des_files
10
#etat_initial_mini_des_files
0
#etat_initial_maxi_des_files
10
#nombre_evenements
2
#tableau_des_evenements
#evt_id-evt_typ-taux-nb_fi_evt-origine-destin1-destin2-destina3
0       0       0.4     3      -1         0        -1
1       0       1.7     2       0        -1
```

Example queue : param

```
#nombre_echantillons
10000
#taille_trajectoire_maxi
100000000
#germe_generateur
5
```

Example queue : result

```
# numero d'echantillon ECHN
# nombre d'iterations NBITER
# etat final sup de la file de numero n  EFSFN
# etat final inf de la file de numero n  EFIFN?
#ECHN:  NBITER:  EFSFN:   0   EFIFN:   0
#==============================
     0       5              1             1
     1       5              1             1
     2       5              0             0
     3       4              0             0
     4       5              0             0
# taille    5  duree d'un tirage  : 136.200000 micro-secondes
# valeur initiale du randomize 5
```

Validation

Analytical models :

- single queue

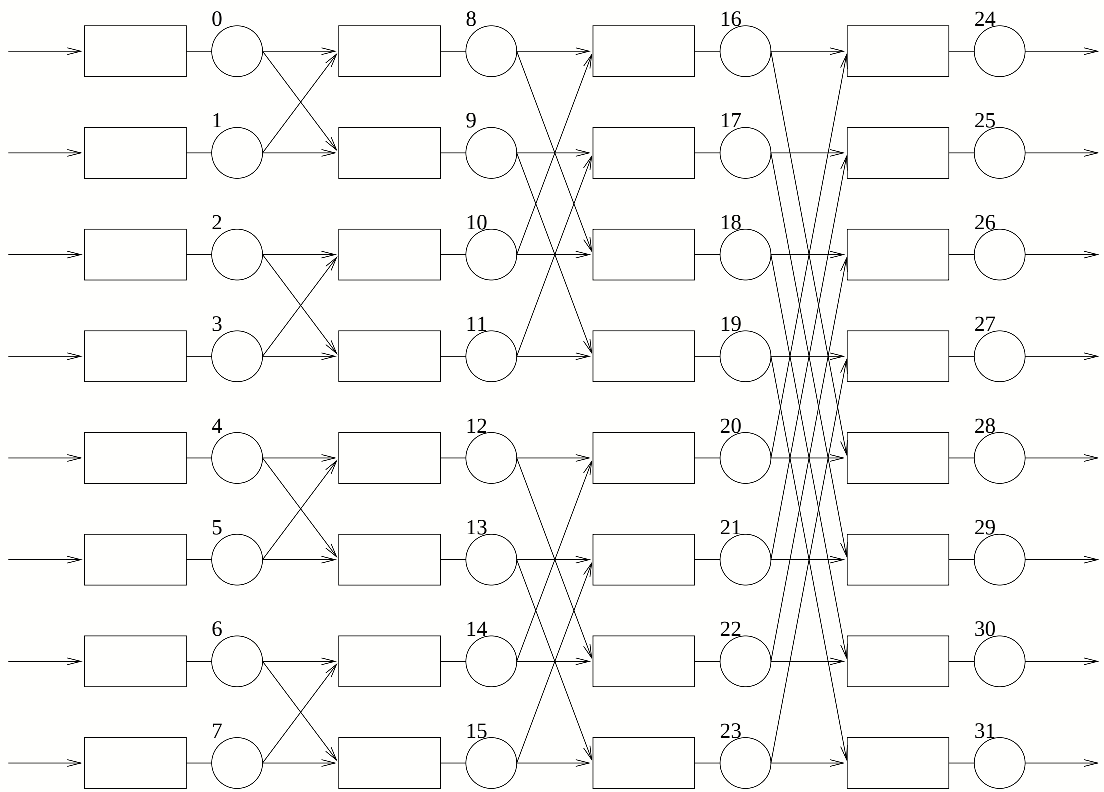
- erlang models

$\Rightarrow$ adequation statistical tests χ^2

Demo

Interconnexion networks

Delta 8×8 network, homogeneous



Interconnexion networks

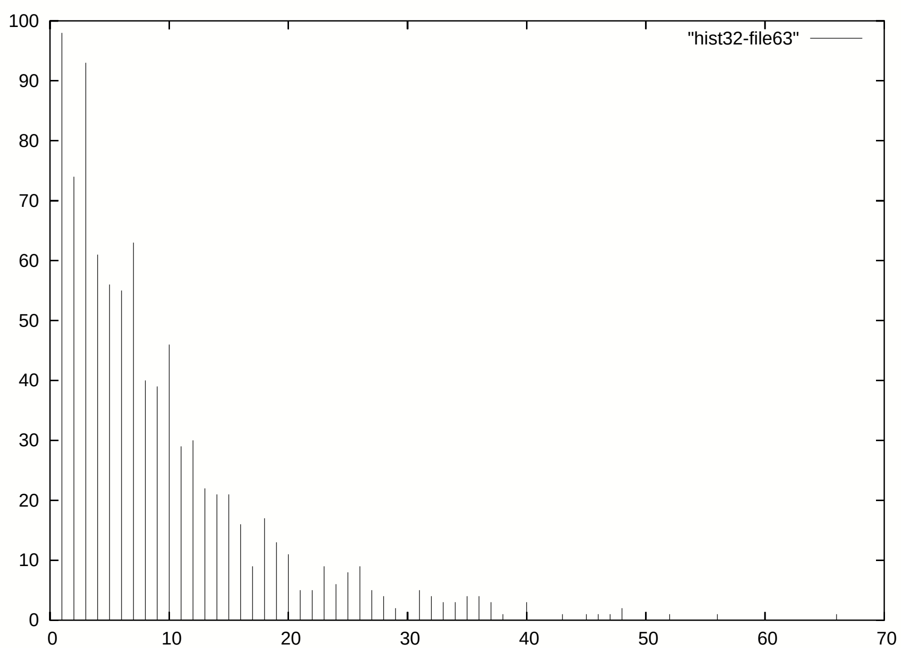
Results

```
# taille 1000  duree d'un tirage  : 135911.287000 micro-secondes
# valeur initiale du randomize 5
```

Coupling time : 2^{17} with probability 0.45 and 2^{18} with probability 0.55

Marginal distributions

Queue 63



Conclusion (1)

- Theoretical results :
  - reverse scheme + contracting operator
  - coupling condition
  - monotonous transitions
  - functional reduction
- Algorithmic results:
  - generic representation of QN
  - guaranteed coupling algorithm
  - compact representation
- Experimental results
  - complexity reduction
  - significant results (depending on the diameter and the coding of the network and capacities

Conclusion (2)

Software tool: **PSI 2 : Perfect Simulator**

http://www-id.imag.fr/Software/PSI2/

unix command / with a simple interface

`psi2_unix -i example.txt -p param.txt -c cost.c -o example.out`

`example.cost` associates to each state its cost

generates samples of costs stationary distributed (`example.sample`)

Future works

- Theoretical improvements:
  - deeper understanding of Φ properties and the spectrum of the transition matrix
  - evaluation or bounds on the coupling time
- Algorithmic perspectives:
  - storage of random sequences,
  - memory utilization,
  - parallelization
- Model based approach :
  - generalization to other modelling frameworks (QN, SAN, GSPN, PA,...)
  - non-monotone events
  - model properties : monotonicity, reversibility,...
- Experimental results
  - find limit models
  - significant results (estimation of rare events probability)
  - more examples