Bounds for the Coupling Time in Queueing Networks Perfect Simulation

J.G. Dopper[2], B. Gaujal and J.-M. Vincent[1]

[1]Laboratory ID-IMAG
MESCAL Project
Universities of Grenoble, France
{Bruno.Gaujal,Jean-Marc.Vincent}@imag.fr
[2]Mathematical Institute,
Leiden University, Nederland
jgdopper@math.leidenuniv.nl

Outline

Outline

1. Queueing Networks with finite capacity
2. Event modelling and monotonicity
3. Perfect simulation and coupling time
4. Acyclic networks
5. Synthesis and future works

Queueing networks with finite capacity

Network model

Finite set of resources :

- servers
- waiting room

Routing strategies :

- state dependent
- overflow strategy
- blocking strategy...

Average performance :

- load of the system
- response time
- loss rate ...

Markov model

Assumptions :
- Poisson arrival,
- exponential distribution for service times,
- probabilistic routing with overflow

⇒ **continuous time Markov chain**

Problem

Computation of the stationary distribution

⇒ **state space explosion**

Queueing networks with finite capacity

Network model

Finite set of resources :

- servers
- waiting room

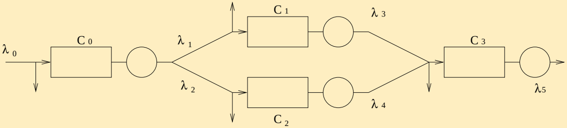
Routing strategies :

- state dependent
- overflow strategy
- blocking strategy...

Average performance :

- load of the system
- response time
- loss rate ...

Markov model



Assumptions :
- Poisson arrival,
- exponential distribution for service times,
- probabilistic routing with overflow

$\Rightarrow$ **continuous time Markov chain**

Problem

Computation of the stationary distribution

$\Rightarrow$ **state space explosion**

Queueing networks with finite capacity

Network model

Finite set of resources :

- servers
- waiting room

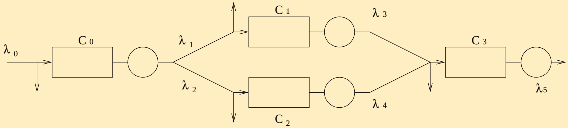
Routing strategies :

- state dependent
- overflow strategy
- blocking strategy...

Average performance :

- load of the system
- response time
- loss rate ...

Markov model



Assumptions :
- Poisson arrival,
- exponential distribution for service times,
- probabilistic routing with overflow

$\Rightarrow$ **continuous time Markov chain**

Problem

Computation of the stationary distribution
$\Rightarrow$ **state space explosion**

Related works

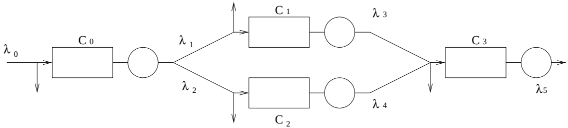
Non reversible systems (reverse event)
Product form solution ??
Widely studied domain

- Analytical solution [Perros 94]
  - specific cases
  - numerical computation of normalization constant
- Numerical computation [Stewart 94]
- Approximation techniques [Onvural 90, Perros 94,...]
- Simulation [Banks & al. 01,...]
  simulation of Markov models
  simulation of event graphs
  discrete event simulation
  perfect simulation [Mattson 04]

Outline

Event modelling

Queueing model :



Event description :

	rate	origin	destination	enabling condition	routing policy
e_0	λ_0	Q_{-1}	Q_0	none	rejection if Q_0 is full
e_1	λ_1	Q_0	Q_1	$s_0 > 0$	rejection if Q_1 is full
e_2	λ_2	Q_0	Q_2	$s_0 > 0$	rejection if Q_2 is full
e_3	λ_3	Q_1	Q_3	$s_1 > 0$	rejection if Q_3 is full
e_4	λ_4	Q_2	Q_3	$s_2 > 0$	rejection if Q_3 is full
e_5	λ_5	Q_3	Q_{-1}	$s_3 > 0$	none

Event modelling

Multidimensional state space

$$\mathcal{X} = \mathcal{X}_0 \times \cdots \times \mathcal{X}_{K-1}]$$

with $\mathcal{X}_i = \{0, \cdots, C_i\}$.
Event e :
$\rightsquigarrow$ transition function $\Phi(., e)$;
$\rightsquigarrow$ Poisson process λ_e

Poisson driven system

Uniformization $\Rightarrow$ GSMP representation

$$\Lambda = \sum_e \lambda_e \text{ and } \mathbb{P}(event\ e) = \frac{\lambda_e}{\Lambda}; \text{ Trajectory : } \{e_n\}_{n\in\mathbb{Z}} \text{ i.i.d.}$$

$\Rightarrow$ Homogeneous Discrete Time Markov Chain [Bremaud 99]

$$X_{n+1} = \Phi(X_n, e_{n+1}).$$

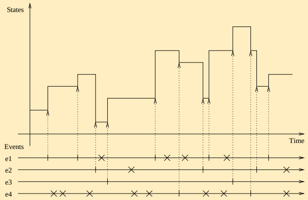
Event modelling

Multidimensional state space

$$\mathcal{X} = \mathcal{X}_0 \times \cdots \times \mathcal{X}_{K-1}]$$

with $\mathcal{X}_i = \{0, \cdots, C_i\}$.
Event e :
$\rightsquigarrow$ transition function $\Phi(., e)$;
$\rightsquigarrow$ Poisson process λ_e

Poisson driven system



Uniformization $\Rightarrow$ GSMP representation

$$\Lambda = \sum_e \lambda_e \text{ and } \mathbb{P}(event\ e) = \frac{\lambda_e}{\Lambda}; \text{ Trajectory : } \{e_n\}_{n\in\mathbb{Z}} \text{ i.i.d.}$$

$\Rightarrow$ Homogeneous Discrete Time Markov Chain [Bremaud 99]

$$X_{n+1} = \Phi(X_n, e_{n+1}).$$

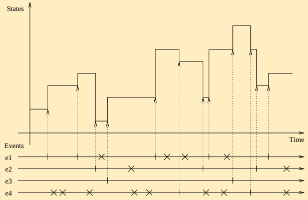
Event modelling

Multidimensional state space

$$\mathcal{X} = \mathcal{X}_0 \times \cdots \times \mathcal{X}_{K-1}]$$

with $\mathcal{X}_i = \{0, \cdots, C_i\}$.
Event e :
$\rightsquigarrow$ transition function $\Phi(., e)$;
$\rightsquigarrow$ Poisson process λ_e

Poisson driven system



Uniformization $\Rightarrow$ GSMP representation

$$\Lambda = \sum_e \lambda_e \text{ and } \mathbb{P}(event\ e) = \frac{\lambda_e}{\Lambda}; \text{ Trajectory : } \{e_n\}_{n\in\mathbb{Z}} \text{ i.i.d.}$$

$\Rightarrow$ Homogeneous Discrete Time Markov Chain [Bremaud 99]

$$X_{n+1} = \Phi(X_n, e_{n+1}).$$

Monotonicity of routing strategy

$(\mathcal{X}, \prec)$ partially ordered set (componentwise)

$$x = [x_0, x_1, \cdots, x_{K-1}] \prec y = [y_0, y_1, \cdots, y_{K-1}] \text{ iff } \forall i, \;\; x_i \leqslant y_i.$$

An event e is said to be monotone if

$$x \prec y \;\; \Rightarrow \;\; \Phi(x, e) \prec \Phi(y, e).$$

Examples [Glasserman and Yao]

All of these routing events are monotone:
- external arrival with overflow and rejection
- routing with overflow and rejection or blocking
- routing to the shortest available queue
- routing to the shortest mean available response time
- general index policies [Palmer-Mitrani]
- rerouting inside queues

...

Monotonicity of routing strategy

$(\mathcal{X}, \prec)$ partially ordered set (componentwise)

$$x = [x_0, x_1, \cdots, x_{K-1}] \prec y = [y_0, y_1, \cdots, y_{K-1}] \text{ iff } \forall i, \;\; x_i \leqslant y_i.$$

An event e is said to be monotone if

$$x \prec y \;\; \Rightarrow \;\; \Phi(x, e) \prec \Phi(y, e).$$

Examples [Glasserman and Yao]

All of these routing events are monotone:
- external arrival with overflow and rejection
- routing with overflow and rejection or blocking
- routing to the shortest available queue
- routing to the shortest mean available response time
- general index policies [Palmer-Mitrani]
- rerouting inside queues

...

Outline

Classical forward simulation

Forward

Representation : **transition fonction**

$$X_{n+1} = \Phi(X_n, e_{n+1}).$$

$x \leftarrow x_0$
{choice of the initial state at time =0}
$n = 0$;
repeat
  $n \leftarrow n + 1$;
  $e \leftarrow Random\_event()$;
  $x \leftarrow \Phi(x, e)$;
  {computation of the next state X_{n+1}}
until some empirical criteria
return x

Convergence : biased sample
Sampling : Warm-up period

Trajectory

Complexity

Related to the stabilization period
Estimation : replication or ergodic estimation

Classical forward simulation

Forward

Representation : **transition fonction**

$$X_{n+1} = \Phi(X_n, e_{n+1}).$$

$x \leftarrow x_0$
{choice of the initial state at time =0}
$n = 0$;
repeat
  $n \leftarrow n + 1$;
  $e \leftarrow Random\_event()$;
  $x \leftarrow \Phi(x, e)$;
  {computation of the next state X_{n+1}}
until some empirical criteria
return x

Convergence : biased sample
Sampling : Warm-up period

Trajectory

Complexity

Related to the stabilization period
Estimation : replication or ergodic estimation

Classical forward simulation

Forward

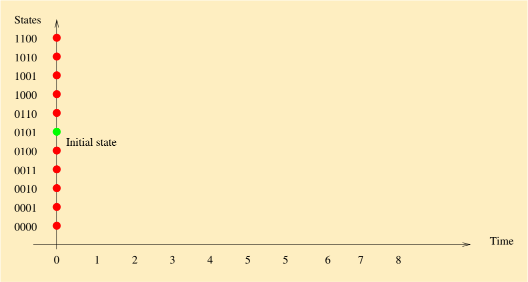
Representation : **transition fonction**

$$X_{n+1} = \Phi(X_n, e_{n+1}).$$

$x \leftarrow x_0$
{choice of the initial state at time =0}
$n = 0$;
repeat
  $n \leftarrow n+1$;
  $e \leftarrow Random\_event()$;
  $x \leftarrow \Phi(x, e)$;
  {computation of the next state X_{n+1}}
until some empirical criteria
return x

Convergence : biased sample
Sampling : Warm-up period

Trajectory



Complexity

Related to the stabilization period
Estimation : replication or ergodic estimation

Classical forward simulation

Forward

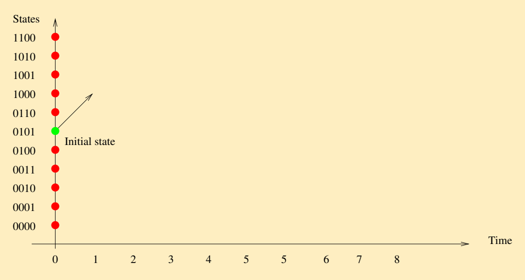
Representation : **transition fonction**

$$X_{n+1} = \Phi(X_n, e_{n+1}).$$

$x \leftarrow x_0$
{choice of the initial state at time =0}
$n = 0$;
repeat
  $n \leftarrow n+1$;
  $e \leftarrow Random\_event()$;
  $x \leftarrow \Phi(x, e)$;
  {computation of the next state X_{n+1}}
until some empirical criteria
return x

Convergence : biased sample
Sampling : Warm-up period

Trajectory



Complexity

Related to the stabilization period
Estimation : replication or ergodic estimation

Classical forward simulation

Forward

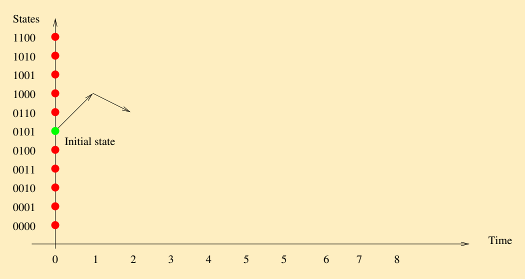
Representation : **transition fonction**

$$X_{n+1} = \Phi(X_n, e_{n+1}).$$

$x \leftarrow x_0$
{choice of the initial state at time =0}
$n = 0$;
repeat
  $n \leftarrow n + 1$;
  $e \leftarrow Random\_event()$;
  $x \leftarrow \Phi(x, e)$;
  {computation of the next state X_{n+1}}
until some empirical criteria
return x

Convergence : biased sample
Sampling : Warm-up period

Trajectory



Complexity

Related to the stabilization period
Estimation : replication or ergodic estimation

Classical forward simulation

Forward

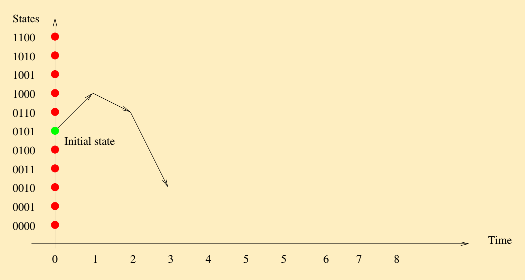
Representation : **transition fonction**

$$X_{n+1} = \Phi(X_n, e_{n+1}).$$

$x \leftarrow x_0$
{choice of the initial state at time =0}
$n = 0$;
repeat
  $n \leftarrow n+1$;
  $e \leftarrow Random\_event()$;
  $x \leftarrow \Phi(x, e)$;
  {computation of the next state X_{n+1}}
until some empirical criteria
return x

Convergence : biased sample
Sampling : Warm-up period

Trajectory



Complexity

Related to the stabilization period
Estimation : replication or ergodic estimation

Classical forward simulation

Forward

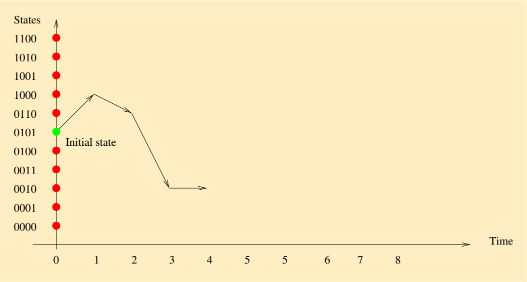
Representation : **transition fonction**

$$X_{n+1} = \Phi(X_n, e_{n+1}).$$

$x \leftarrow x_0$
{choice of the initial state at time =0}
$n = 0$;
repeat
  $n \leftarrow n + 1$;
  $e \leftarrow Random\_event()$;
  $x \leftarrow \Phi(x, e)$;
  {computation of the next state X_{n+1}}
until some empirical criteria
return x

Convergence : biased sample
Sampling : Warm-up period

Trajectory



Complexity

Related to the stabilization period
Estimation : replication or ergodic estimation

Classical forward simulation

Forward

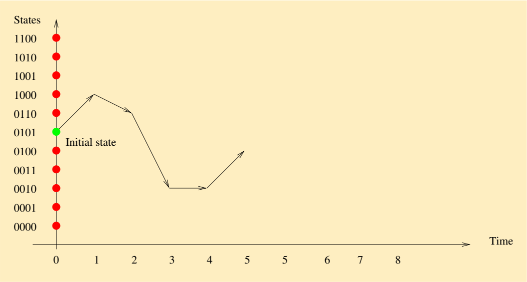
Representation : **transition fonction**

$$X_{n+1} = \Phi(X_n, e_{n+1}).$$

$x \leftarrow x_0$
{choice of the initial state at time =0}
$n = 0$;
repeat
  $n \leftarrow n + 1$;
  $e \leftarrow Random\_event()$;
  $x \leftarrow \Phi(x, e)$;
  {computation of the next state X_{n+1}}
until some empirical criteria
return x

Convergence : biased sample
Sampling : Warm-up period

Trajectory



Complexity

Related to the stabilization period
Estimation : replication or ergodic estimation

Classical forward simulation

Forward

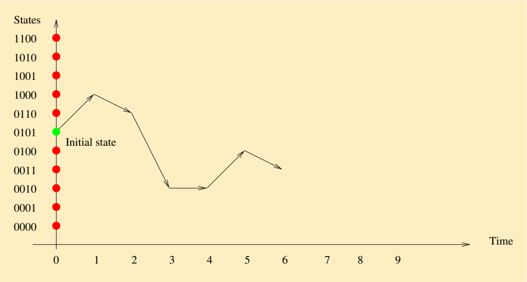
Representation : **transition fonction**

$$X_{n+1} = \Phi(X_n, e_{n+1}).$$

$x \leftarrow x_0$
{choice of the initial state at time =0}
$n = 0$;
repeat
  $n \leftarrow n+1$;
  $e \leftarrow Random\_event()$;
  $x \leftarrow \Phi(x, e)$;
  {computation of the next state X_{n+1}}
until some empirical criteria
return x

Convergence : biased sample
Sampling : Warm-up period

Trajectory



Complexity

Related to the stabilization period
Estimation : replication or ergodic estimation

Classical forward simulation

Forward

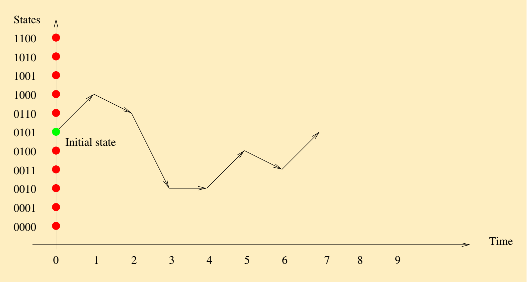
Representation : **transition fonction**

$$X_{n+1} = \Phi(X_n, e_{n+1}).$$

$x \leftarrow x_0$
{choice of the initial state at time =0}
$n = 0$;
repeat
  $n \leftarrow n+1$;
  $e \leftarrow Random\_event()$;
  $x \leftarrow \Phi(x, e)$;
  {computation of the next state X_{n+1}}
until some empirical criteria
return x

Convergence : biased sample
Sampling : Warm-up period

Trajectory



Complexity

Related to the stabilization period
Estimation : replication or ergodic estimation

Classical forward simulation

Forward

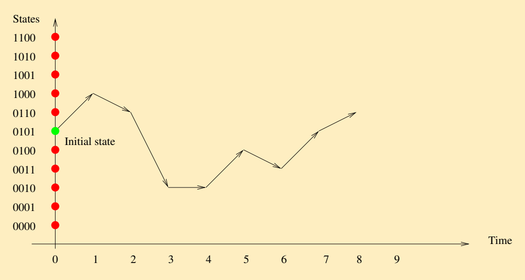
Representation : **transition fonction**

$$X_{n+1} = \Phi(X_n, e_{n+1}).$$

$x \leftarrow x_0$
{choice of the initial state at time =0}
$n = 0;$
repeat
  $n \leftarrow n + 1;$
  $e \leftarrow Random\_event();$
  $x \leftarrow \Phi(x, e);$
  {computation of the next state X_{n+1}}
until some empirical criteria
return x

Convergence : biased sample
Sampling : Warm-up period

Trajectory



Complexity

Related to the stabilization period
Estimation : replication or ergodic estimation

Classical forward simulation

Forward

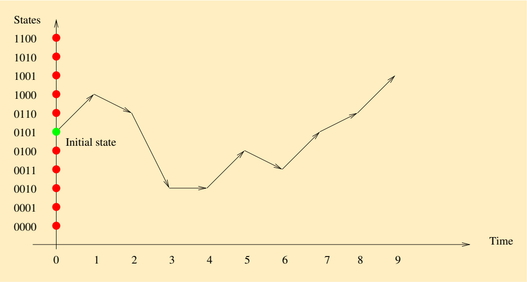
Representation : **transition fonction**

$$X_{n+1} = \Phi(X_n, e_{n+1}).$$

$x \leftarrow x_0$
{choice of the initial state at time =0}
$n = 0$;
repeat
  $n \leftarrow n + 1$;
  $e \leftarrow Random\_event()$;
  $x \leftarrow \Phi(x, e)$;
  {computation of the next state X_{n+1}}
until some empirical criteria
return x

Convergence : biased sample
Sampling : Warm-up period

Trajectory



Complexity

Related to the stabilization period
Estimation : replication or ergodic estimation

Classical forward simulation

Forward

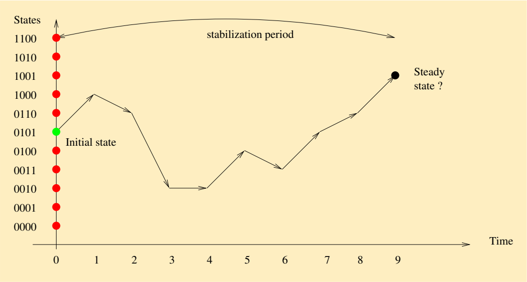
Representation : **transition fonction**

$$X_{n+1} = \Phi(X_n, e_{n+1}).$$

$x \leftarrow x_0$
{choice of the initial state at time =0}
$n = 0$;
repeat
  $n \leftarrow n + 1$;
  $e \leftarrow Random\_event()$;
  $x \leftarrow \Phi(x, e)$;
  {computation of the next state X_{n+1}}
until some empirical criteria
return x

Convergence : biased sample
Sampling : Warm-up period

Trajectory



Complexity

Related to the stabilization period
Estimation : replication or ergodic estimation

Classical forward simulation

Forward

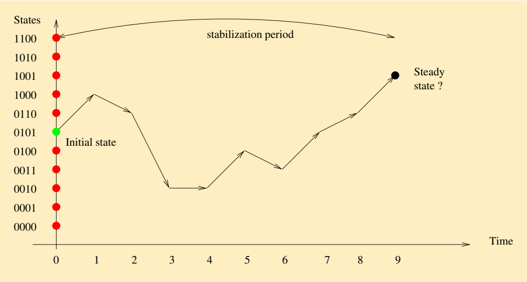
Representation : **transition fonction**

$$X_{n+1} = \Phi(X_n, e_{n+1}).$$

$x \leftarrow x_0$
{choice of the initial state at time =0}
$n = 0$;
repeat
  $n \leftarrow n + 1$;
  $e \leftarrow Random\_event()$;
  $x \leftarrow \Phi(x, e)$;
  {computation of the next state X_{n+1}}
until some empirical criteria
return x

Convergence : biased sample
Sampling : Warm-up period

Trajectory



Complexity

Related to the stabilization period
Estimation : replication or ergodic estimation

Perfect simulation : backward idea

Representation : **transition fonction**

$$X_{n+1} = \Phi(X_n, e_{n+1}), \{e_n\}_{n\in\mathbb{Z}} \text{ i.i.d. sequence.}$$

In what state could I be at time $n = 0$?

$$\begin{aligned}
X_0 &\in \mathcal{X} = \mathcal{Z}_0 \\
&\in \Phi(\mathcal{X}, e_0) = \mathcal{Z}_1 \\
&\in \Phi(\Phi(\mathcal{X}, e_{-1}), e_0) = \mathcal{Z}_2 \\
&\cdots \\
&\in \Phi(\Phi(\cdots\Phi(\mathcal{X}, e_{-n+1}), \cdots), e_0) = \mathcal{Z}_n
\end{aligned}$$

Theorem

Provided some condition on the sequence of events, the sequence of sets

$$\mathcal{Z}_0 \supseteq \mathcal{Z}_1 \supseteq \mathcal{Z}_2 \supseteq \cdots \supseteq \mathcal{Z}_n \supseteq \cdots \text{ is decreasing to a single state.}$$

The generated state is stationary distributed (steady state sample).

$$\tau^b = \inf\{n \in \mathbb{N};\ Card(\mathcal{Z}_n) = 1\}.$$

backward coupling time

Perfect simulation : backward idea

Representation : **transition fonction**

$$X_{n+1} = \Phi(X_n, e_{n+1}), \{e_n\}_{n\in\mathbb{Z}} \text{ i.i.d. sequence.}$$

In what state could I be at time $n = 0$?

$$\begin{array}{lll} X_0 & \in & \mathcal{X} = \mathcal{Z}_0 \\ & \in & \Phi(\mathcal{X}, e_0) = \mathcal{Z}_1 \\ & \in & \Phi(\Phi(\mathcal{X}, e_{-1}), e_0) = \mathcal{Z}_2 \\ & \cdots & \\ & \in & \Phi(\Phi(\cdots\Phi(\mathcal{X}, e_{-n+1}), \cdots), e_0) = \mathcal{Z}_n \end{array}$$

Theorem

Provided some condition on the sequence of events, the sequence of sets

$$\mathcal{Z}_0 \supseteq \mathcal{Z}_1 \supseteq \mathcal{Z}_2 \supseteq \cdots \supseteq \mathcal{Z}_n \supseteq \cdots \text{ is decreasing to a single state.}$$

The generated state is stationary distributed (steady state sample).

$$\tau^b = \inf\{n \in \mathbb{N};\ Card(\mathcal{Z}_n) = 1\}.$$

backward coupling time

Perfect simulation

Backward algorithm

Representation : **transition fonction**

$$X_{n+1} = \Phi(X_n, e_{n+1}).$$

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

Convergence : If the algorithm stops, the returned value is steady state distributed
Coupling time: $\tau < +\infty$, properties of Φ

Trajectories

Mean time complexity

c_Φ mean computation cost of $\Phi(x, e)$

$$C \leqslant Card(\mathcal{X}).\mathbb{E}\tau.c_\Phi.$$

Perfect simulation

Backward algorithm

Representation : **transition fonction**

$$X_{n+1} = \Phi(X_n, e_{n+1}).$$

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

Convergence : If the algorithm stops, the returned value is steady state distributed
Coupling time: $\tau < +\infty$, properties of Φ

Trajectories

Mean time complexity

c_Φ mean computation cost of $\Phi(x, e)$

$$C \leqslant Card(\mathcal{X}).\mathbb{E}\tau.c_\Phi.$$

Perfect simulation

Backward algorithm

Representation : **transition fonction**

$$X_{n+1} = \Phi(X_n, e_{n+1}).$$

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

Convergence : If the algorithm stops, the returned value is steady state distributed
Coupling time: $\tau < +\infty$, properties of Φ

Trajectories

Mean time complexity

c_Φ mean computation cost of $\Phi(x, e)$

$$C \leqslant Card(\mathcal{X}).\mathbb{E}\tau.c_\Phi.$$

Perfect simulation

Backward algorithm

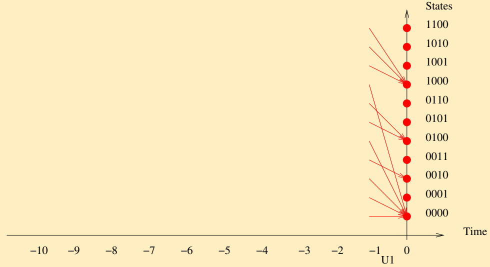
Representation : **transition fonction**

$$X_{n+1} = \Phi(X_n, e_{n+1}).$$

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

Convergence : If the algorithm stops, the returned value is steady state distributed
Coupling time: $\tau < +\infty$, properties of Φ

Trajectories



Mean time complexity

c_Φ mean computation cost of $\Phi(x, e)$

$$C \leqslant Card(\mathcal{X}).\mathbb{E}\tau.c_\Phi.$$

Perfect simulation

Backward algorithm

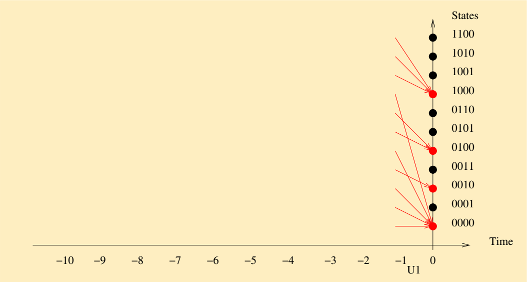
Representation : **transition fonction**

$$X_{n+1} = \Phi(X_n, e_{n+1}).$$

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

Convergence : If the algorithm stops, the returned value is steady state distributed
Coupling time: $\tau < +\infty$, properties of Φ

Trajectories



Mean time complexity

c_Φ mean computation cost of $\Phi(x, e)$

$$C \leqslant Card(\mathcal{X}).\mathbb{E}\tau.c_\Phi.$$

Perfect simulation

Backward algorithm

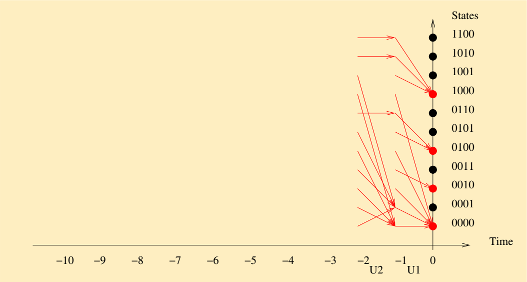
Representation : **transition fonction**

$$X_{n+1} = \Phi(X_n, e_{n+1}).$$

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

Convergence : If the algorithm stops, the returned value is steady state distributed
Coupling time: $\tau < +\infty$, properties of Φ

Trajectories



Mean time complexity

c_Φ mean computation cost of $\Phi(x, e)$

$$C \leqslant Card(\mathcal{X}).\mathbb{E}\tau.c_\Phi.$$

Perfect simulation

Backward algorithm

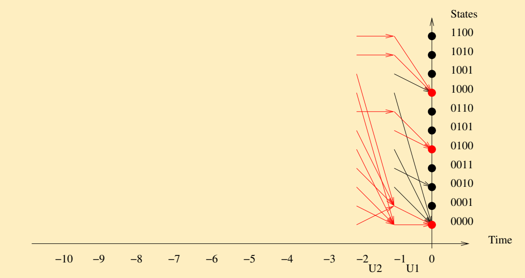
Representation : **transition fonction**

$$X_{n+1} = \Phi(X_n, e_{n+1}).$$

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

Convergence : If the algorithm stops, the returned value is steady state distributed
Coupling time: $\tau < +\infty$, properties of Φ

Trajectories



Mean time complexity

c_Φ mean computation cost of $\Phi(x, e)$

$$C \leqslant Card(\mathcal{X}).\mathbb{E}\tau.c_\Phi.$$

Perfect simulation

Backward algorithm

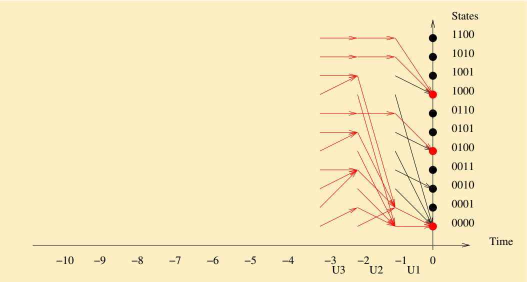
Representation : **transition fonction**

$$X_{n+1} = \Phi(X_n, e_{n+1}).$$

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

Convergence : If the algorithm stops, the returned value is steady state distributed
Coupling time: $\tau < +\infty$, properties of Φ

Trajectories



Mean time complexity

c_Φ mean computation cost of $\Phi(x, e)$

$$C \leqslant Card(\mathcal{X}).\mathbb{E}\tau.c_\Phi.$$

Perfect simulation

Backward algorithm

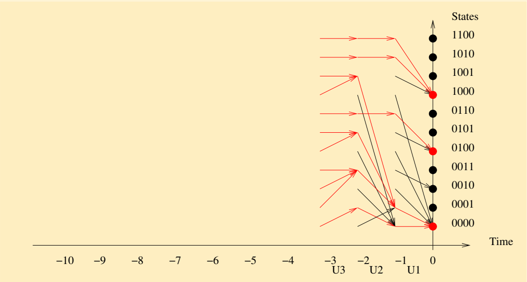
Representation : **transition fonction**

$$X_{n+1} = \Phi(X_n, e_{n+1}).$$

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

Convergence : If the algorithm stops, the returned value is steady state distributed
Coupling time: $\tau < +\infty$, properties of Φ

Trajectories



Mean time complexity

c_Φ mean computation cost of $\Phi(x, e)$

$$C \leqslant Card(\mathcal{X}).\mathbb{E}\tau.c_\Phi.$$

Perfect simulation

Backward algorithm

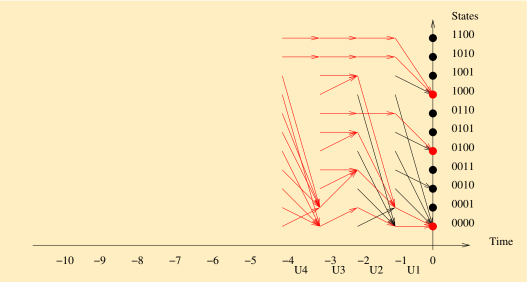
Representation : **transition fonction**

$$X_{n+1} = \Phi(X_n, e_{n+1}).$$

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

Convergence : If the algorithm stops, the returned value is steady state distributed
Coupling time: $\tau < +\infty$, properties of Φ

Trajectories



Mean time complexity

c_Φ mean computation cost of $\Phi(x, e)$

$$C \leqslant Card(\mathcal{X}).\mathbb{E}\tau.c_\Phi.$$

Perfect simulation

Backward algorithm

Representation : **transition fonction**

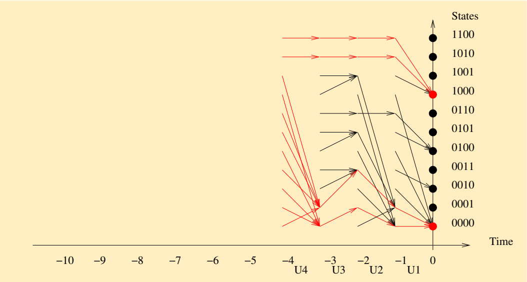
$$X_{n+1} = \Phi(X_n, e_{n+1}).$$

```
for all x ∈ 𝒳 do
  y(x) ← x
end for
repeat
  u ← Random;
  for all x ∈ 𝒳 do
    e ← Random_event();
    y(x) ← y(Φ(x, e));
  end for
until All y(x) are equal
return y(x)
```

Convergence : If the algorithm stops, the returned value is steady state distributed

Coupling time: $\tau < +\infty$, properties of Φ

Trajectories



Mean time complexity

c_Φ mean computation cost of $\Phi(x, e)$

$$C \leqslant Card(\mathcal{X}).\mathbb{E}\tau.c_\Phi.$$

Perfect simulation

Backward algorithm

Representation : **transition fonction**

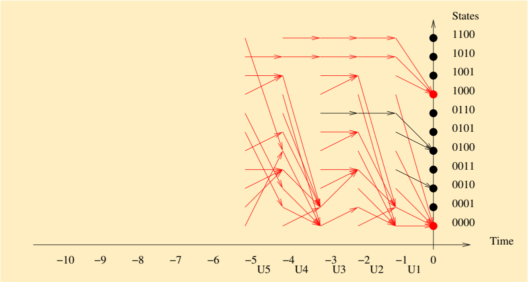
$$X_{n+1} = \Phi(X_n, e_{n+1}).$$

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

Convergence : If the algorithm stops, the returned value is steady state distributed

Coupling time: $\tau < +\infty$, properties of Φ

Trajectories



Mean time complexity

c_Φ mean computation cost of $\Phi(x, e)$

$$C \leqslant Card(\mathcal{X}).\mathbb{E}\tau.c_\Phi.$$

Perfect simulation

Backward algorithm

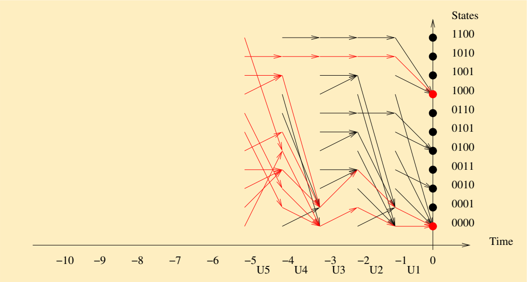
Representation : **transition fonction**

$$X_{n+1} = \Phi(X_n, e_{n+1}).$$

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

Convergence : If the algorithm stops, the returned value is steady state distributed
Coupling time: $\tau < +\infty$, properties of Φ

Trajectories



Mean time complexity

c_Φ mean computation cost of $\Phi(x, e)$

$$C \leqslant Card(\mathcal{X}).\mathbb{E}\tau.c_\Phi.$$

Perfect simulation

Backward algorithm

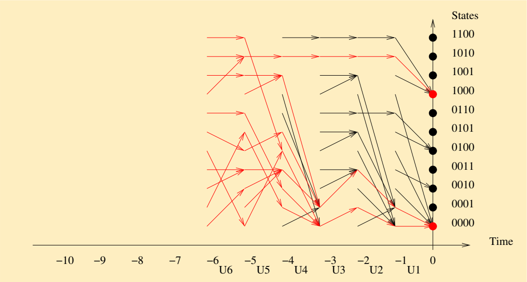
Representation : **transition fonction**

$$X_{n+1} = \Phi(X_n, e_{n+1}).$$

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

Convergence : If the algorithm stops, the returned value is steady state distributed
Coupling time: $\tau < +\infty$, properties of Φ

Trajectories



Mean time complexity

c_Φ mean computation cost of $\Phi(x, e)$

$$C \leqslant Card(\mathcal{X}).\mathbb{E}\tau.c_\Phi.$$

Perfect simulation

Backward algorithm

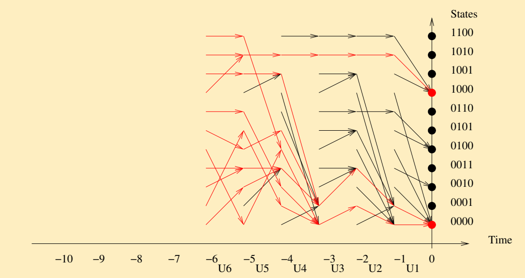
Representation : **transition fonction**

$$X_{n+1} = \Phi(X_n, e_{n+1}).$$

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

Convergence : If the algorithm stops, the returned value is steady state distributed
Coupling time: $\tau < +\infty$, properties of Φ

Trajectories



Mean time complexity

c_Φ mean computation cost of $\Phi(x, e)$

$$C \leqslant Card(\mathcal{X}).\mathbb{E}\tau.c_\Phi.$$

Perfect simulation

Backward algorithm

Representation : **transition fonction**

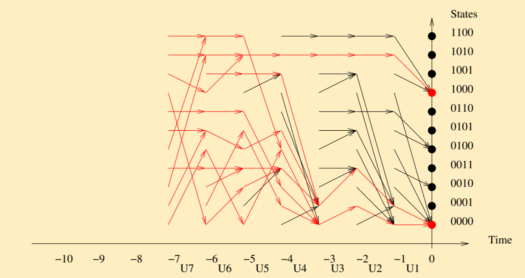
$$X_{n+1} = \Phi(X_n, e_{n+1}).$$

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

Convergence : If the algorithm stops, the returned value is steady state distributed

Coupling time: $\tau < +\infty$, properties of Φ

Trajectories



Mean time complexity

c_Φ mean computation cost of $\Phi(x, e)$

$$C \leqslant Card(\mathcal{X}).\mathbb{E}\tau.c_\Phi.$$

Perfect simulation

Backward algorithm

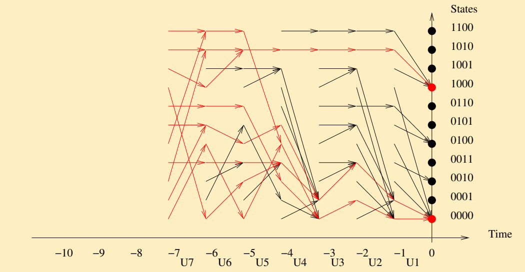
Representation : **transition fonction**

$$X_{n+1} = \Phi(X_n, e_{n+1}).$$

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

Convergence : If the algorithm stops, the returned value is steady state distributed
Coupling time: $\tau < +\infty$, properties of Φ

Trajectories



Mean time complexity

c_Φ mean computation cost of $\Phi(x, e)$

$$C \leqslant Card(\mathcal{X}).\mathbb{E}\tau.c_\Phi.$$

Perfect simulation

Backward algorithm

Representation : **transition fonction**

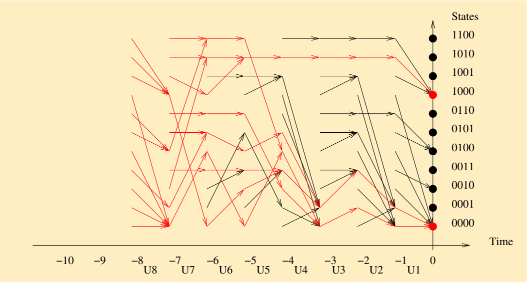
$$X_{n+1} = \Phi(X_n, e_{n+1}).$$

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

Convergence : If the algorithm stops, the returned value is steady state distributed

Coupling time: $\tau < +\infty$, properties of Φ

Trajectories



Mean time complexity

c_Φ mean computation cost of $\Phi(x, e)$

$$C \leqslant Card(\mathcal{X}).\mathbb{E}\tau.c_\Phi.$$

Perfect simulation

Backward algorithm

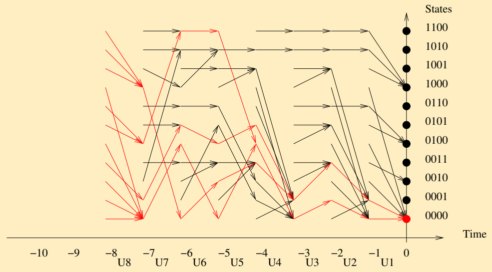
Representation : **transition fonction**

$$X_{n+1} = \Phi(X_n, e_{n+1}).$$

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

Convergence : If the algorithm stops, the returned value is steady state distributed
Coupling time: $\tau < +\infty$, properties of Φ

Trajectories



Mean time complexity

c_Φ mean computation cost of $\Phi(x, e)$

$$C \leqslant Card(\mathcal{X}).\mathbb{E}\tau.c_\Phi.$$

Perfect simulation

Backward algorithm

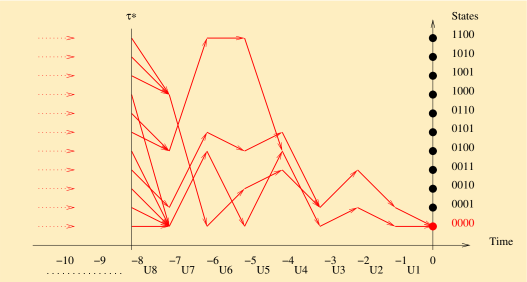
Representation : **transition fonction**

$$X_{n+1} = \Phi(X_n, e_{n+1}).$$

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

Convergence : If the algorithm stops, the returned value is steady state distributed
Coupling time: $\tau < +\infty$, properties of Φ

Trajectories



Mean time complexity

c_Φ mean computation cost of $\Phi(x, e)$

$$C \leqslant Card(\mathcal{X}).\mathbb{E}\tau.c_\Phi.$$

Perfect simulation

Backward algorithm

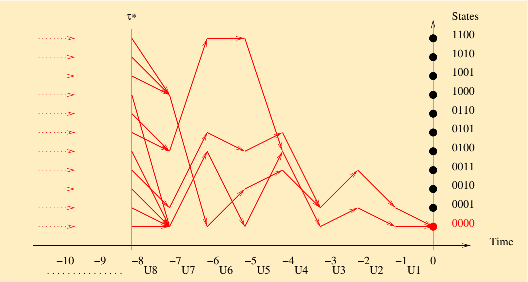
Representation : **transition fonction**

$$X_{n+1} = \Phi(X_n, e_{n+1}).$$

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

Convergence : If the algorithm stops, the returned value is steady state distributed
Coupling time: $\tau < +\infty$, properties of Φ

Trajectories



Mean time complexity

c_Φ mean computation cost of $\Phi(x, e)$

$$C \leqslant Card(\mathcal{X}).\mathbb{E}\tau.c_\Phi.$$

Monotonicity and perfect simulation : idea

$$min = (0, \cdots, 0) \text{ and } Max = (C_1, \cdots, C_n).$$

If all events are monotone then

$$X_0 \in \mathcal{Z}_n \subset [\Phi(min, e_{-n \to 0}), \Phi(Max, e_{-n \to 0})]$$

$\Rightarrow$ **2 trajectories**

Monotonicity and perfect simulation

Monotone PS

Doubling scheme

n=1;R[1]=Random_event;
repeat
n=2.n;
$y(min) \leftarrow min$
$y(Max) \leftarrow Max$
for i=n downto n/2+1 **do**
R[i]=Random_event;
end for
for i=n downto 1 **do**
$y(min) \leftarrow \Phi(y(min), R[i])$
$y(Max) \leftarrow \Phi(y(Max), R[i])$
end for
until $y(min) = y(Max)$
return $y(min)$

Trajectories

Mean time complexity

$C_n \leq 2.(2.\mathbb{E}\tau).C_\Phi$. Reduction factor : $\frac{4}{Card(\mathcal{X})}$.

Monotonicity and perfect simulation

Monotone PS

Doubling scheme

```
n=1;R[1]=Random_event;
repeat
   n=2.n;
   y(min) ← min
   y(Max) ← Max
   for i=n downto n/2+1 do
      R[i]=Random_event;
   end for
   for i=n downto 1 do
      y(min) ← Φ(y(min), R[i])
      y(Max) ← Φ(y(Max), R[i])
   end for
until y(min) = y(Max)
return y(min)
```

Trajectories

Mean time complexity

$C_m \leqslant 2.(2.\mathbb{E}\tau).c_\Phi$. Reduction factor : $\frac{4}{Card(\mathcal{X})}$.

Monotonicity and perfect simulation

Monotone PS

Doubling scheme

n=1;R[1]=Random_event;
repeat
  n=2.n;
  $y(min) \longleftarrow min$
  $y(Max) \longleftarrow Max$
  for i=n downto n/2+1 **do**
    R[i]=Random_event;
  end for
  for i=n downto 1 **do**
    $y(min) \longleftarrow \Phi(y(min), R[i])$
    $y(Max) \longleftarrow \Phi(y(Max), R[i])$
  end for
until $y(min) = y(Max)$
return $y(min)$

Trajectories

Mean time complexity

$C_m \leqslant 2.(2.\mathbb{E}\tau).c_\Phi$. Reduction factor : $\frac{4}{Card(\mathcal{X})}$.

Monotonicity and perfect simulation

Monotone PS

Doubling scheme

n=1;R[1]=Random_event;
repeat
  n=2.n;
  $y(min) \longleftarrow min$
  $y(Max) \longleftarrow Max$
  for i=n downto n/2+1 **do**
    R[i]=Random_event;
  end for
  for i=n downto 1 **do**
    $y(min) \longleftarrow \Phi(y(min), R[i])$
    $y(Max) \longleftarrow \Phi(y(Max), R[i])$
  end for
until $y(min) = y(Max)$
return $y(min)$

Trajectories

Mean time complexity

$C_m \leqslant 2.(2.\mathbb{E}\tau).c_\Phi$. Reduction factor : $\frac{4}{Card(\mathcal{X})}$.

Monotonicity and perfect simulation

Monotone PS

Doubling scheme

```
n=1;R[1]=Random_event;
repeat
   n=2.n;
   y(min) ← min
   y(Max) ← Max
   for i=n downto n/2+1 do
      R[i]=Random_event;
   end for
   for i=n downto 1 do
      y(min) ← Φ(y(min), R[i])
      y(Max) ← Φ(y(Max), R[i])
   end for
until y(min) = y(Max)
return y(min)
```

Trajectories

Mean time complexity

$C_m \leqslant 2.(2.\mathbb{E}\tau).c_\Phi$. Reduction factor : $\frac{4}{Card(\mathcal{X})}$.

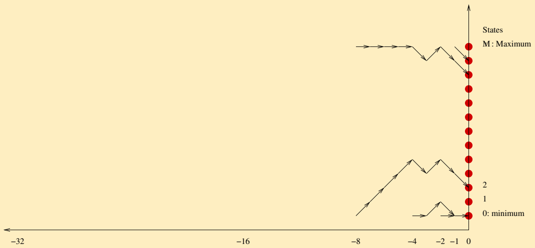
Monotonicity and perfect simulation

Monotone PS

Doubling scheme

n=1;R[1]=Random_event;
repeat
n=2.n;
$y(min) \leftarrow min$
$y(Max) \leftarrow Max$
for i=n downto n/2+1 **do**
R[i]=Random_event;
end for
for i=n downto 1 **do**
$y(min) \leftarrow \Phi(y(min), R[i])$
$y(Max) \leftarrow \Phi(y(Max), R[i])$
end for
until $y(min) = y(Max)$
return $y(min)$

Trajectories



Mean time complexity

$C_m \leqslant 2.(2.\mathbb{E}\tau).c_\Phi$. Reduction factor : $\frac{4}{Card(\mathcal{X})}$.

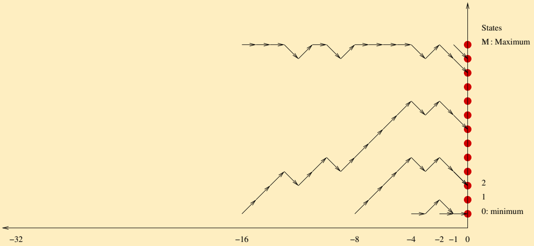
Monotonicity and perfect simulation

Monotone PS

Doubling scheme

n=1;R[1]=Random_event;
repeat
n=2.n;
$y(min) \leftarrow min$
$y(Max) \leftarrow Max$
for i=n downto n/2+1 **do**
R[i]=Random_event;
end for
for i=n downto 1 **do**
$y(min) \leftarrow \Phi(y(min), R[i])$
$y(Max) \leftarrow \Phi(y(Max), R[i])$
end for
until $y(min) = y(Max)$
return $y(min)$

Trajectories



Mean time complexity

$C_m \leqslant 2.(2.\mathbb{E}\tau).c_\Phi$. Reduction factor : $\frac{4}{Card(\mathcal{X})}$.

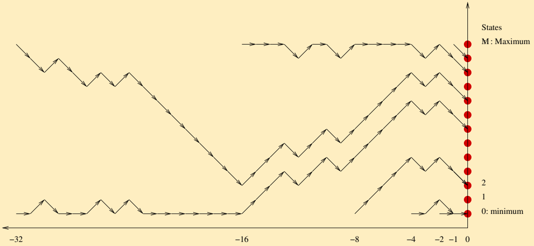
Monotonicity and perfect simulation

Monotone PS

Doubling scheme

n=1;R[1]=Random_event;
repeat
  n=2.n;
  $y(min) \leftarrow min$
  $y(Max) \leftarrow Max$
  for i=n downto n/2+1 **do**
    R[i]=Random_event;
  end for
  for i=n downto 1 **do**
    $y(min) \leftarrow \Phi(y(min), R[i])$
    $y(Max) \leftarrow \Phi(y(Max), R[i])$
  end for
until $y(min) = y(Max)$
return $y(min)$

Trajectories



Mean time complexity

$C_m \leqslant 2.(2.\mathbb{E}\tau).c_\Phi$. Reduction factor : $\frac{4}{Card(\mathcal{X})}$.

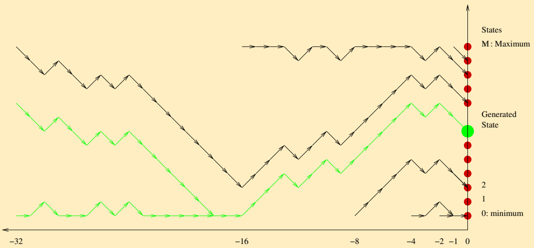
Monotonicity and perfect simulation

Monotone PS

Doubling scheme

```
n=1;R[1]=Random_event;
repeat
   n=2.n;
   y(min) ← min
   y(Max) ← Max
   for i=n downto n/2+1 do
      R[i]=Random_event;
   end for
   for i=n downto 1 do
      y(min) ← Φ(y(min), R[i])
      y(Max) ← Φ(y(Max), R[i])
   end for
until y(min) = y(Max)
return y(min)
```

Trajectories



Mean time complexity

$C_m \leqslant 2.(2.\mathbb{E}\tau).c_\Phi$. Reduction factor : $\frac{4}{Card(\mathcal{X})}$.

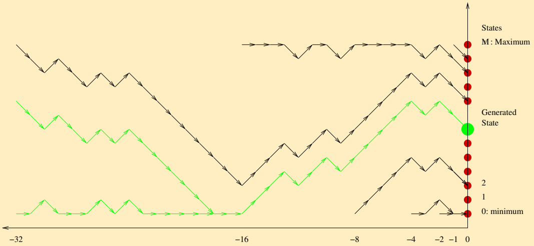
Monotonicity and perfect simulation

Monotone PS

Doubling scheme

n=1;R[1]=Random_event;
repeat
  n=2.n;
  $y(min) \leftarrow min$
  $y(Max) \leftarrow Max$
  for i=n downto n/2+1 **do**
    R[i]=Random_event;
  end for
  for i=n downto 1 **do**
    $y(min) \leftarrow \Phi(y(min), R[i])$
    $y(Max) \leftarrow \Phi(y(Max), R[i])$
  end for
until $y(min) = y(Max)$
return $y(min)$

Trajectories



Mean time complexity

$C_m \leqslant 2.(2.\mathbb{E}\tau).c_\Phi$. Reduction factor : $\frac{4}{Card(\mathcal{X})}$.

Coupling time

definition

$$\tau^b = \min\{n \in \mathbb{N};\ Card(\mathcal{Z}_n) = 1\};$$
$$= \min\{n \in \mathbb{N};\ |\Phi(\mathcal{X}, e_{-n\to 0}| = 1\}.$$

Properties

- Backward τ^b and forward τ^f coupling times have the same probability distribution;
- Marginal coupling : denote by τ_i^b the backward coupling time for Q_i

$$\tau^b = \max \tau_i^b.$$

Problem : compute the mean coupling time

Coupling time

definition

$$\begin{aligned}\tau^b &= \min\{n \in \mathbb{N};\ Card(\mathcal{Z}_n) = 1\};\\ &= \min\{n \in \mathbb{N};\ |\Phi(\mathcal{X}, e_{-n\to 0}| = 1\}.\end{aligned}$$

Properties

- Backward τ^b and forward τ^f coupling times have the same probability distribution;
- Marginal coupling : denote by τ_i^b the backward coupling time for Q_i

$$\tau^b = \max \tau_i^b.$$

Problem : compute the mean coupling time

Coupling time

definition

$$\begin{aligned} \tau^b &= \min\{n \in \mathbb{N};\ Card(\mathcal{Z}_n) = 1\}; \\ &= \min\{n \in \mathbb{N};\ |\Phi(\mathcal{X}, e_{-n \to 0}| = 1\}. \end{aligned}$$

Properties

- Backward τ^b and forward τ^f coupling times have the same probability distribution;
- Marginal coupling : denote by τ_i^b the backward coupling time for Q_i

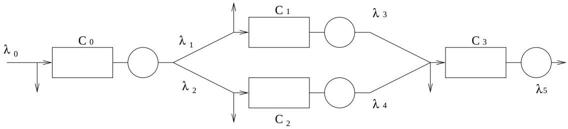
$$\tau^b = \max \tau_i^b.$$

Problem : compute the mean coupling time

Outline

Coupling experiment

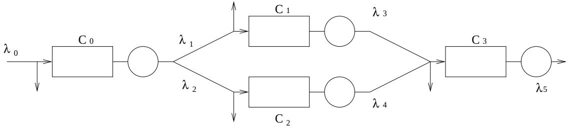
Queueing model :



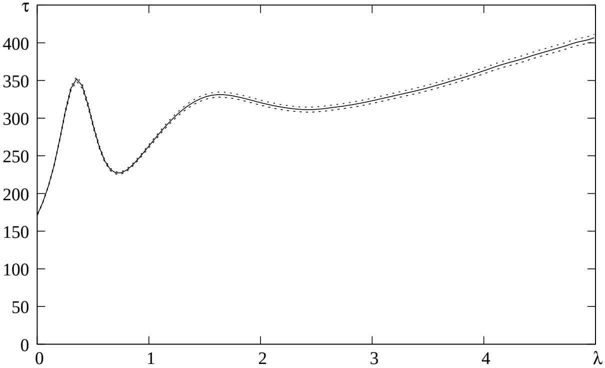
Estimation of $\mathbb{E}\tau$:

Coupling experiment

Queueing model :



Estimation of $\mathbb{E}\tau$:



Main result

Theorem (Bound on coupling time)

$$\mathbb{E}\tau \leqslant \sum_{i=1}^{K} \frac{\Lambda}{\Lambda_i} \frac{C_i + C_i^2}{2},$$

- Λ : *global event rate in the network,*
- Λ_i *the rate of events affecting* Q_i
- C_i *is the capacity of Queue* i.

Sketch of the proof

- Explicit computation for the $M/M/1/C$
- Computable bounds for the $M/M/1/C$
- Bound with isolated queues

Main result

Theorem (Bound on coupling time)

$$\mathbb{E}\tau \leqslant \sum_{i=1}^{K} \frac{\Lambda}{\Lambda_i} \frac{C_i + C_i^2}{2},$$

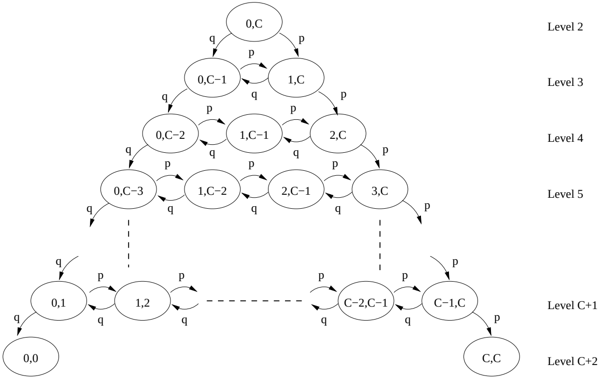
- *Λ : global event rate in the network,*
- *Λ_i the rate of events affecting Q_i*
- *C_i is the capacity of Queue i.*

Sketch of the proof

- Explicit computation for the $M/M/1/C$
- Computable bounds for the $M/M/1/C$
- Bound with isolated queues

Explicit computation for the $M/M/1/C$

$\mathbb{E}\tau^b = \mathbb{E}\min(h_{0\to C}, h_{C\to 0})$
Absorbing time in a finite Markov chain; $p = \frac{\lambda}{\lambda+\mu} = 1 - q$



Explicit recurrence equations
Case $\lambda = \mu$ $\mathbb{E}\tau^b = \frac{C+C^2}{2}$.

Computable bounds for $M/M/1/C$

If the stationary distribution is concentrated on 0 ($\lambda < \mu$),

$$\mathbb{E}\tau^b \leqslant \mathbb{E}h_{0\to C} \text{ is an accurate bound.}$$

Theorem

The mean coupling time $\mathbb{E}\tau^b$ of a M/M/1/C queue with arrival rate λ and service rate μ is bounded using $p = \lambda/(\lambda+\mu) = 1-q$.

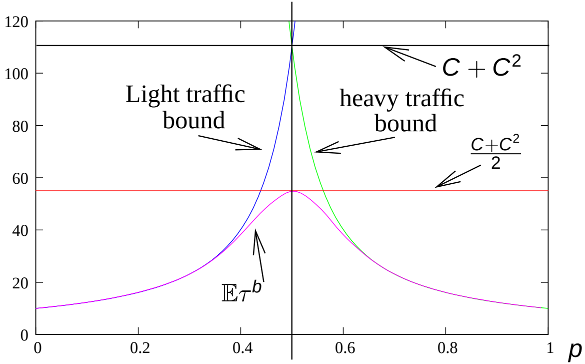
Critical bound: $\forall p \in [0,1], \quad \mathbb{E}\tau^b \leqslant \frac{C^2+C}{2}.$

Heavy traffic Bound: if $p > \frac{1}{2}, \quad \mathbb{E}\tau^b \leqslant \frac{C}{p-q} - \frac{q\left(1-\left(\frac{q}{p}\right)^C\right)}{(p-q)^2}.$

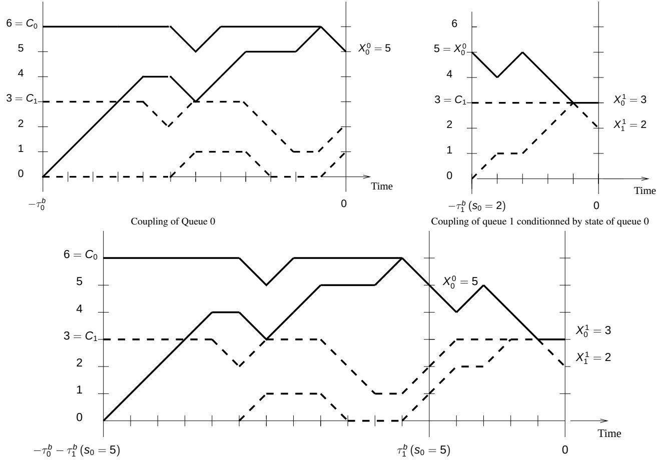
Light traffic bound: if $p < \frac{1}{2}, \quad \mathbb{E}\tau^b \leqslant \frac{C}{q-p} - \frac{p\left(1-\left(\frac{p}{q}\right)^C\right)}{(q-p)^2}.$

Computable bounds for $M/M/1/C$

Example with $C = 10$



Example for tandem queues



Then $\tau^b \leqslant_{st} \quad {}^{\infty}\tau_1^b + \tau_0^b$, normalized

Bound with isolated queues

Theorem

In an acyclic stable network of K $M/M/1/C_i$ queues with Bernoulli routing and losses in case of overflow, the coupling time from the past satisfies in expectation,

$$\mathbb{E}[\tau^b] \leqslant \sum_{i=0}^{K-1} \frac{\Lambda}{\ell_i + \mu_i} \left(\frac{C_i}{q_i - p_i} - \frac{p_i \left(1 - \left(\frac{p_i}{q_i}\right)^{C_i}\right)}{(q_i - p_i)^2} \right)$$

$$\leqslant \sum_{i=0}^{K-1} \frac{\Lambda}{\ell_i + \mu_i} (C_i + C_i^2).$$

Outline

Synthesis

Computable bound for the mean coupling time :

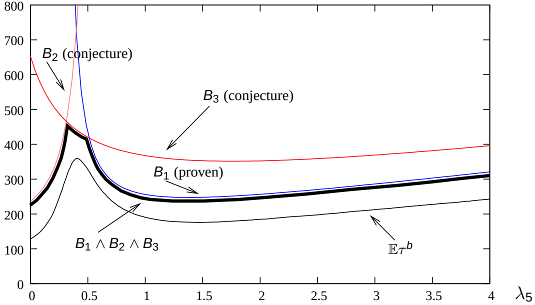
- linear in the number of component of the model;
- at most quadratic in queues sizes;
- large capacity queues (bound is accurate).

Practical impact

- Accurate bounds, dimensionning of trajectories length;
- Simulation useful even for low probability events;
- Coupling time is explained by the spread of the stationary distribution.

Future works

Conjecture for general networks.



Extension to cyclic networks,

Generalization to several types of events

Application : Grid and call centers