Perfect Sampling of Queuing Networks with Complex Routing

complexity and computational aspects

Jean-Marc Vincent

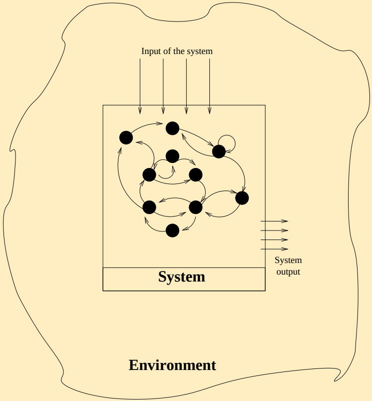
MESCAL-INRIA Project
Laboratoire d'Informatique de Grenoble
Universities of Grenoble, France
{Jean-Marc.Vincent}@imag.fr

This work was partially supported by ANR SETIN Checkbound

Outline

Application Problems : Modeling and Analysis of Complex Systems

Complex system



Basic model assumptions

System :
- automaton (discrete state space)
- **discrete** or continuous time

Environment : non deterministic
- time homogeneous
- stochastically regular

Problem

Generate “typical” states
- steady-state sampling
- ergodic simulation starting point
- state space exploring techniques

Queuing Networks with Finite Capacity

Network model

Finite set of resources :

- servers
- waiting rooms

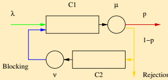
Routing strategies :

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

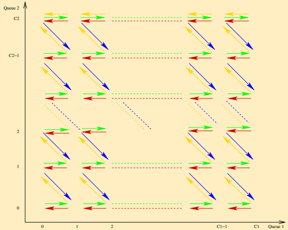
Average performance :

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

Markov model



Poisson arrival, exponential services distribution, probabilistic routing
$\Rightarrow$ continuous time Markov chain

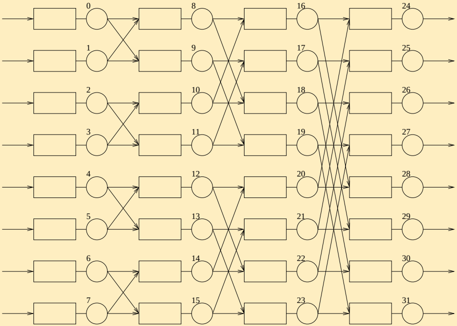


Problem

Computation of steady state distribution
$\Rightarrow$ state-space explosion

Interconnexion Networks

Delta network



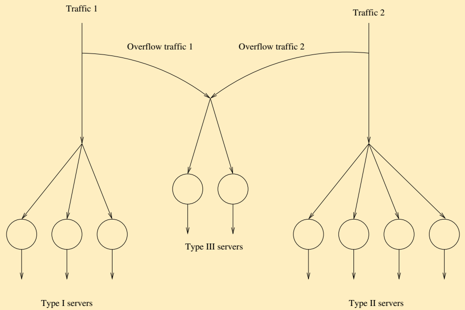
Input rates
Service rates
Homogeneous routing
Overflow strategy

Problem

Loss probability at each level
Analysis of hot spot
...

Call centers

Multilevel Erlang model



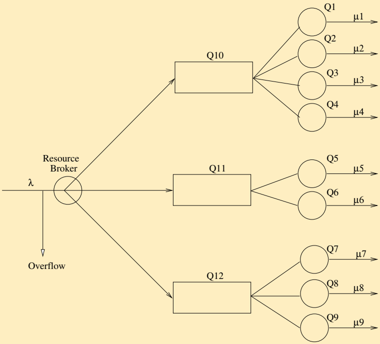
Types of requests
Input rates
Different service rates
Overflow strategy

Problem

Optimization of resources
Quality of service (waiting time, rejection probability,...)
...

Resource Broker

Grid model



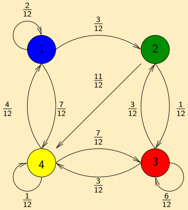
Input rates
Allocation strategy
State dependent allocation
Index based routing : destination minimize a criteria

Problem

Optimization of throughput, response time,...
Comparison of policies, analysis of heuristics
...

Formalization : Markov Chain

Quantification



Stochastic matrix : transition probability

$$P = \frac{1}{12} \begin{bmatrix} 2 & 3 & 0 & 7 \\ 0 & 0 & 1 & 11 \\ 0 & 3 & 6 & 3 \\ 4 & 0 & 7 & 1 \end{bmatrix}$$

Non-negative, if irreducible and aperiodic
Unique probability vector π satisfying $\pi = \pi P$,
$\pi = \frac{1}{350}[46, 47, 142, 115]$

1856-1922

Solving methods

Solving $\pi = \pi P$

- Analytical/approximatin methods
- Formal methods $N \leqslant 50$
  Maple, Sage,...
- Direct numerical methods $N \leqslant 1000$
  Mathematica, Scilab,...
- Iterative methods with preconditioning $N \leqslant 100,000$
  Marca,...
- Adapted methods (structured Markov chains) $N \leqslant 1,000,000$
  PEPS,...
- Monte-Carlo simulation $N \geqslant 10^7$

Postprocessing of the stationary distribution

Computation of rewards (expected stationary functions)
Utilization, response time,...

Ergodic Sampling(1)

Ergodic sampling algorithm

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
  $n \leftarrow n + 1$;
  $e \leftarrow Random\_event()$;
  $x \leftarrow \Phi(x, e)$;
  Store x
  {computation of the next state X_{n+1}}
until some empirical criteria
return the trajectory

Problem : Stopping criteria

Ergodic Sampling(2)

Start-up

Convergence to stationary behavior

$$\lim_{n \to +\infty} \mathbb{P}(X_n = x) = \pi_x.$$

Warm-up period : Avoid initial state dependence

Estimation error :

$$||\mathbb{P}(X_n = x) - \pi_x|| \leqslant C\lambda_2^n.$$

λ_2 second greatest eigenvalue of the transition matrix

- bounds on C and λ_2 (spectral gap)
- cut-off phenomena

λ_2 and C non reachable in practice
(complexity equivalent to the computation of π)
some known results (Birth and Death processes)

Ergodic Sampling(3)

Estimation quality

Ergodic theorem :

$$\lim_{n\to+\infty} \frac{1}{n}\sum_{i=1}^{n} f(X_i) = \mathbb{E}_\pi f.$$

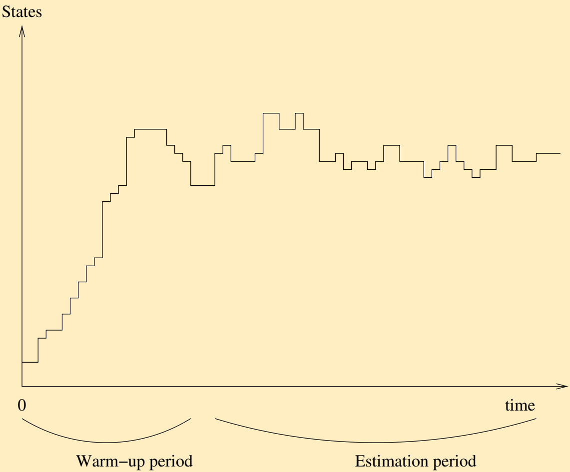
Length of the sampling : Error control (CLT theorem)
CLT for additive functionals of Markov chains

Complexity

Complexity of the transition function evaluation (computation of $\Phi(x, .)$)
Related to the stabilization period + Estimation time

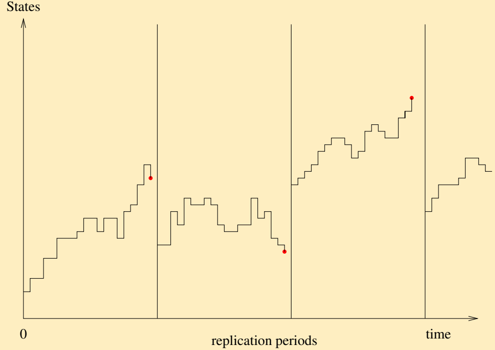
Ergodic sampling(4)

Typical trajectory



Replication Method

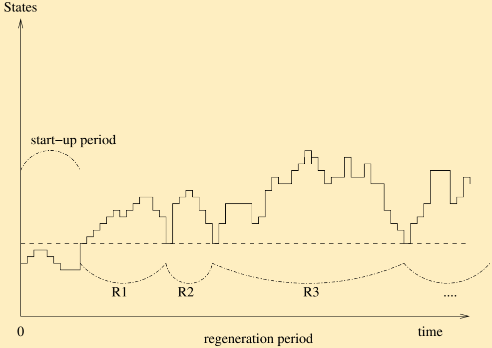
Typical trajectory



Sample of independent states
Drawback : length of the replication period (dependence from initial state)

Regeneration Method

Typical trajectory



Sample of independent trajectories
Drawback : length of the regeneration period (choice of the regenerative state)

Stochastic recursive sequences

Description [Borovkov et al]

- Discrete state space $\mathcal{X}$ (usually lattice, product of intervals,...)
- Innovation state space, and an innovation process
- Dynamic of the system : transition function

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

- Trajectory given by x_0 and $\{\xi_n\}$ an innovation process

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

Discrete event systems

- state space : usually lattice, product of intervals,...
- Innovations : usually a set of events $\mathcal{E}$
- Independent innovation process : Poisson systems (uniformization)

Markovian Modelling

Theorem (Markov process)

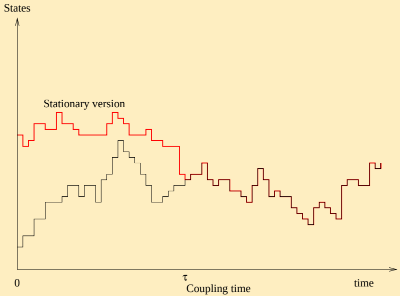
If $\{\xi_n\}$ is a sequence of iid random variables , the process $\{X_n\}$ is a homogeneous discrete time Markov chain.

Random Iterated system of functions

The trajectory X_n is the successive application of random functions taken in the set $\{\Phi(.,\xi),\ \xi \in \mathcal{E}\}$ according a probability measure on $\mathcal{E}$ [Diaconis and Friedman 98]

Coupling Inequality

Typical trajectory

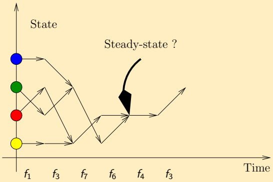


After τ the two processes are not distinguishable, then stationary
Scheme used to prove Markov convergence (coupling inequality)

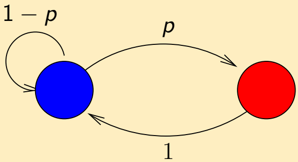
$$|\mathbb{P}(X_n \in A) - \pi_A| \leqslant \mathbb{P}(\tau \geqslant n)$$

Forward Sampling : avoid initial state dependence

Forward coupling



Example



Always couple in the blue state

Does not guarantee the steady state !

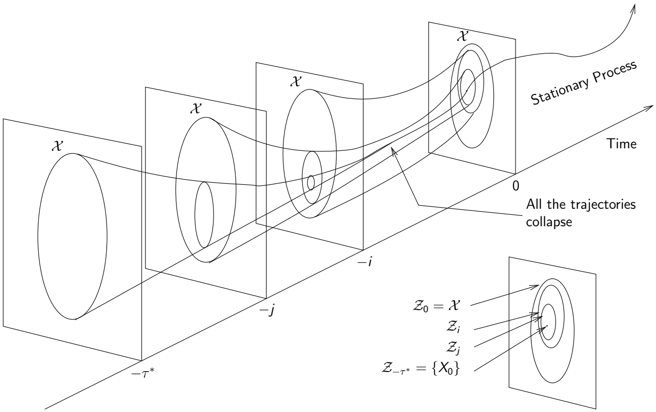
Perfect Sampling : Backward Idea

Set dynamic

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

$$
\begin{aligned}
X_0 &\in \mathcal{X} = \mathcal{Z}_0 \\
&\in \Phi(\mathcal{X}, e_{-1}) = \mathcal{Z}_1 \\
&\in \Phi(\Phi(\mathcal{X}, e_{-2}), e_{-1}) = \mathcal{Z}_2 \\
\vdots &\quad \vdots \\
&\in \Phi(\Phi(\cdots \Phi(\mathcal{X}, e_{-n}), \cdots), e_{-2}), e_{-1}) = \mathcal{Z}_n
\end{aligned}
$$

Perfect sampling : Backward Idea



Perfect Sampling : Convergence Theorem

Theorem

Provided some condition on the events the sequence of sets

$$\{\mathcal{Z}_n\}_{n\in\mathbb{N}}$$

is decreasing to a single state, stationary distributed.

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

backward coupling time

The set of possible states at time 0 is decreasing with regards to n

Perfect Sampling : Coupling Condition

Theorem

Suppose that the set of events is finite. Then the two conditions are equivalent:

- $\tau^* < +\infty$ *almost surely;*
- *There exist a finite sequence of events with positive probability* $\mathcal{S} = \{e_1, \cdots, e_M\}$ *such that*

$$|\Phi(\mathcal{X}, e_{1\to M})| = 1.$$

The sequence $\mathcal{S}$ is called a **synchronizing pattern** (synchronizing word, renovating event,...)

Perfect sampling : Coupling Condition (proof)

Proof

- $\Rightarrow$ If $\tau^* < +\infty$ almost surely there is a trajectory that couples in a finite time. This finite trajectory is a synchronizing pattern.
- $\Leftarrow$ Suppose there is a synchronizing pattern with length M. Because the sequence of events is iid, it occurs almost surely on every trajectory. Applying Borel-Cantelli lemma gives the result.

The forward and backward coupling time have the same distribution
τ^* has an exponentially dominated distribution tail

$$\mathbb{P}(\tau^* > M.n) \leqslant (1 - \mathbb{P}(e_{1\to M}))^n.$$

Practically efficient

Perfect sampling : convergence theorem (proof 1)

Proof based on the shift property

First, because $\tau < +\infty$ and the ergodicity of the chain there exists N_0 s.t.

$$|\mathbb{P}(\Phi(\mathcal{X}, e_{1\to n}) = \{x\}) - \pi_x| \leqslant \epsilon.$$

But the sequence of events is iid (stationary) then

$$\mathbb{P}(\Phi(\mathcal{X}, e_{1\to n}) = \{x\}) = \mathbb{P}(\Phi(\mathcal{X}, e_{-n+1\to 0}) = \{x\})$$

$\tau^* < +\infty$ then there exists N_1 such that $\mathbb{P}(\tau^* \geqslant N_1) \leqslant \epsilon$; then

$$\mathbb{P}(\Phi(\mathcal{X}, e_{-n+1\to 0}) = \{x\})$$

$$= \mathbb{P}(\Phi(\mathcal{X}, e_{-n+1\to 0}) = \{x\}, \tau^* < N_1) + \mathbb{P}(\Phi(\mathcal{X}, e_{-n+1\to 0}) = \{x\}, \tau^* \geqslant N_1)$$

$$= \mathbb{P}(\Phi(\mathcal{X}, e_{-\tau^*\to 0}) = \{x\}, \tau^* < N_1) + \epsilon',$$

$$= \mathbb{P}(X_0 = x, \tau^* < N_1) + \epsilon' = \mathbb{P}(X_0 = x) + \epsilon''.$$

Perfect Sampling : Convergence Theorem (proof 2)

Proof based on the coupling property [Haggstrom]

Consider N such that $\mathbb{P}(\tau^* \geqslant N) \leqslant \epsilon$
then consider a process $\{\overline{X}_n\}$ with the same events $e_{-N+1 \to 0}$ but with $\overline{X}_{-N+1}$ generated according π. The process $\{\overline{X}_n\}$ is stationary.
On the event $(\tau^* < N)$ we have $X_0 = \overline{X}_0$ and

$$\mathbb{P}(X_0 \neq \overline{X}_0) \leqslant \mathbb{P}(\tau^* \geqslant N) \leqslant \epsilon \text{ (coupling inequality).}$$

Finally

$$\mathbb{P}(X_0 = x) - \pi_x = \mathbb{P}(X_0 = x) - \mathbb{P}(\overline{X}_0 = x) \leqslant \mathbb{P}(X_0 \neq \overline{X}_0) \leqslant \epsilon;$$

$$\pi_x - \mathbb{P}(X_0 = x) = \mathbb{P}(\overline{X}_0 = x) - \mathbb{P}(X_0 = x) \leqslant \mathbb{P}(\overline{X}_0 \neq X_0) \leqslant \epsilon;$$

and the result follows.

Perfect Sampling : Algorithm

Backward algorithm

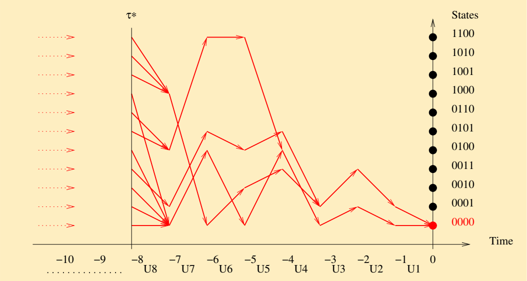
Representation : transition fonction

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

for all $x \in \mathcal{X}$ **do**
  $y(x) \longleftarrow x$
end for
repeat
  $e \longleftarrow Random\_event();$
  for all $x \in \mathcal{X}$ **do**
    $z(x) \longleftarrow y(\Phi(x, e));$
  end for
  $y \longleftarrow z$
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 Reward Sampling

Backward reward

Representation : transition fonction

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

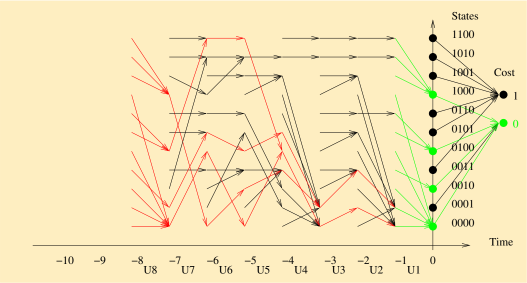
Arbitrary reward function

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

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

Coupling time: $\tau^r \leqslant \tau < +\infty$

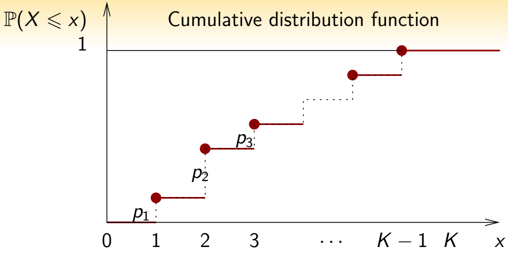
Trajectories



Mean time complexity

$C_{Reward} \leqslant Card(\mathcal{X}).\mathbb{E}\tau.c_{\Phi}$Depends on the reward function.

Inverse of PDF



Generation

Divide $[0, 1[$ in intervals with length p_k
Find the interval in which *Random* falls
Returns the index of the interval
Computation cost : $\mathcal{O}(\mathbb{E}X)$ steps
Memory cost : $\mathcal{O}(1)$

Inverse function algorithm

s=0; k=0;
u=random()
while u >s **do**
  k=k+1
  s=s+p_k
end while
return k

Searching optimization

Optimization methods

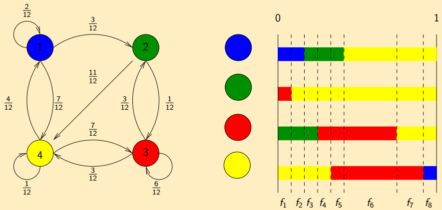
- pre-compute the pdf in a table
- rank objects by decreasing probability
- use a dichotomy algorithm
- use a tree searching algorithm (optimality = Huffmann coding tree)

Comments

- Depends on the usage of the generator (repeated use or not)
- pre-computation usually $\mathcal{O}(K)$ could be huge
-

Generation : Visual representation

$[0, 1]$ partitionning



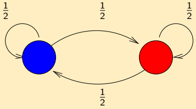
Random iterated system of functions

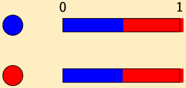
Function	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8
Probability	$\frac{1}{12}$	$\frac{1}{12}$	$\frac{1}{12}$	$\frac{1}{12}$	$\frac{1}{12}$	$\frac{4}{12}$	$\frac{2}{12}$	$\frac{1}{12}$

Stochastic matrix $P \Longrightarrow$ simulation algorithm = RIFS

The coupling problem

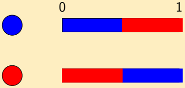
τ estimation





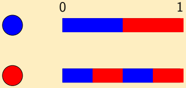
Couples with probability 1

$\tau = 1$



Never couples

$\tau = \infty$



Couples with probability $\frac{1}{2}$

$\mathbb{E}\tau = 2$

General problem

Objective

Given a stochastic matrix $P = ((p_{i,j}))$ build a system of function $(f_\theta, \theta \in \Theta)$ and a probability distribution $(p_\theta, \theta \in \Theta)$ such that :

1. the RIFS implements the transition matrix P,
2. ensures coupling in finite time
3. achieve the "best" mean coupling time : tradeoff between
   - choice of the transition function according to $((p_\theta))$,
   - computation of the transition

Remarks

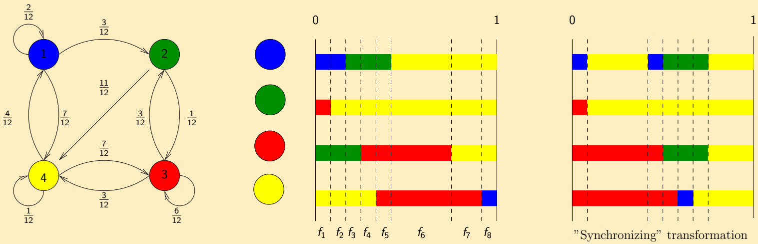
Usual method

$$|\Theta| = \text{number of non-negative elements of } P = \mathcal{O}(n^2)$$

choice in $\mathcal{O}(\log n)$

Non sparse matrices

Rearranging the system



Non sparse matrices

Convergence rate

When at least one column is non-negative $\Rightarrow$ one step coupling.
The RIFS ensures coupling and the coupling time τ is upper bounded by a geometric distribution with rate

$$\sum_j \min_i p_{i,j}$$

number of transition functions : could be more than the number of non-negative elements
at most n^2

Aliasing technique

Initialization

K objects

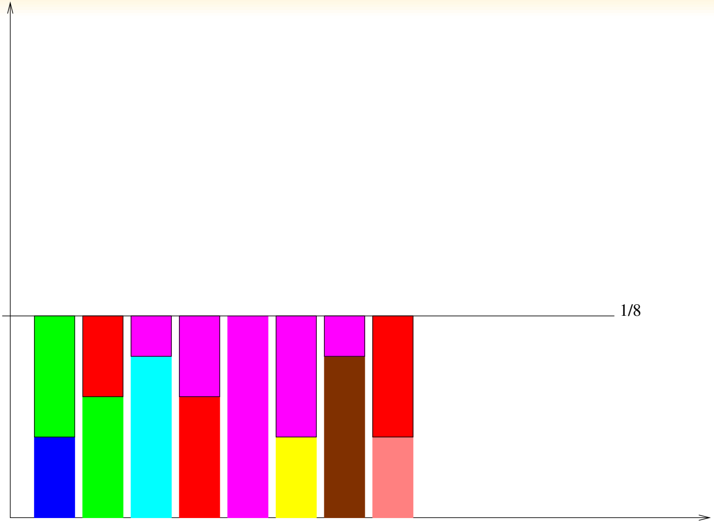
```
list L=∅,U=∅;
for k=1; k⩽ K; k++ do
    P[k]=p_k
    if P[k] ⩾ 1/K then
        U=U+{k};
    else
        L=L+{k};
    end if
end for
```

Alias and threshold tables

```
while L ≠ ∅ do
    Extract k ∈ L
    Extract i ∈ U
    S[k]=P[k]
    A[k]=i
    P[i] = P[i] - (1/K-P[k])
    if P[i] ⩾ 1/K then
        U=U+{i};
    else
        L=L+{i};
    end if
end while
```

Combine uniform and alias value when rejection

Aliasing technique : generation



Aliasing technique : generation

Generation

k=alea(K)
if Random . $\frac{1}{K} \leqslant$ S[k] **then**
  return k
else
  return A[k]
end if

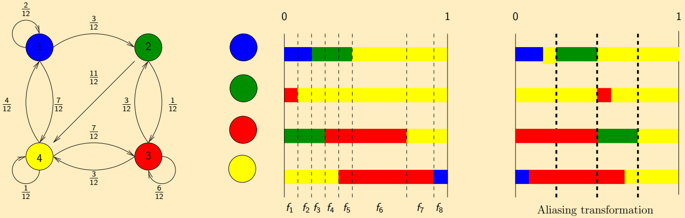
Complexity

Computation time :
- $\mathcal{O}(K)$ for pre-computation
- $\mathcal{O}(1)$ for generation

Memory :
- threshold $\mathcal{O}(K)$ (real numbers as probability)
- alias $\mathcal{O}(K)$ (integers indexes in a tables)

Sparse matrices

Rearranging the system

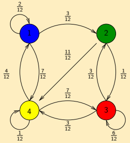


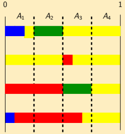
Complexity

$M =$ maximum out degree of states
p_θ uniform on $\{1, \cdots, M\}$, threshold comparison
$\mathcal{O}(1)$ to compute one transition
Combination with "Synchronizing" techniques

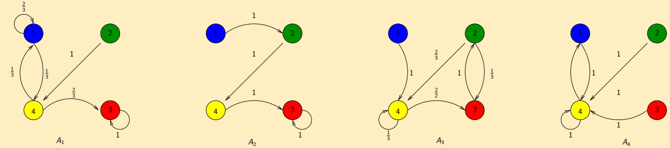
Uniform-binary decomposition

Uniform superposition





Aliasing transformation

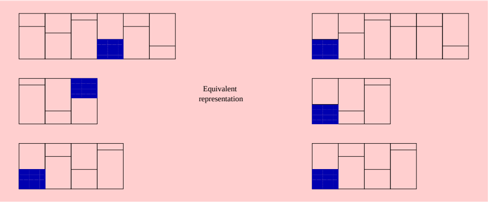


Decomposition

$$P = \frac{1}{M}\sum_{i=1}^{M} P_i, \quad P_i \text{ : stochastic matrix with at most 2 non negative elements per row}$$

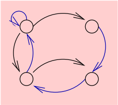
Coupling property

Exchange of columns or thresholds give an equivalent representative



Spanning tree

Irreducibility $\Longrightarrow$ there is a spanning tree going to a single state where coupling occurs.



$$\mathbb{P}(\tau^* < +\infty) = 1.$$

τ is geometrically bounded,
so τ^* and τ_C^*.

Ψ software

Example

Random transition coefficients:

Number of states	10	100	500	1000	3000
Mean coupling time	3.1	4.5	5.3	5.7	6.1
Mean execution time μs	3	17	170	360	1100

Pentium III 700MHz and 256Mb memory. Sample size 10000.

Remarks:

- very small coupling time
- Coefficients : same order of magnitude, aliasing enforces coupling

Comparison with birth and death process :

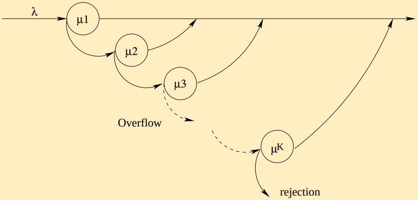
Number of states	10	100	500	1000	3000
Mean coupling time	41	557	2850	5680	17000
Mean execution time μs	28	1800	88177	366000	3.5s

Remarks:

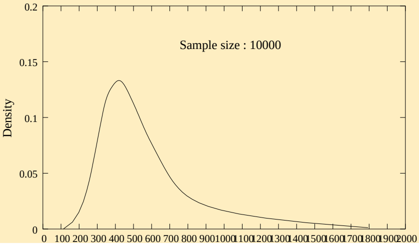
- large coupling time
- sparse matrix, large graph diameter

Overflow model

Model



Coupling time distribution



Parameters

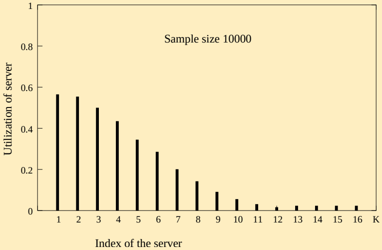
K servers,
priority on overflows
input rate λ,
different service rate
state $(x_1, \cdots, x_K)$, $x_i \in \{0, 1\}$,
size ~ 130000
low diameter
non product-form structure,

Statistics

Parameter	Value
minimum	113
maximum	1794
median	465
mean	498
Std	180

Overflow model (2)

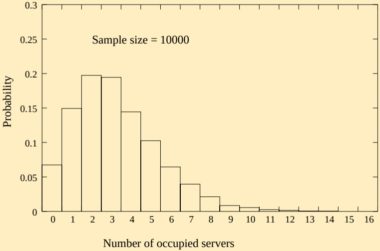
Marginal distribution



Marginal probability estimation

$$\mathbb{P}(X_i = 1)$$

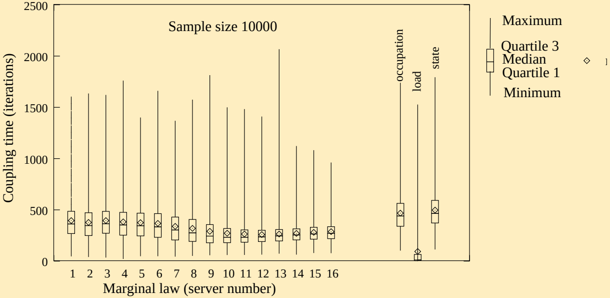
Occupied servers



Marginal distribution of the occupied servers

Overflow model (3)

Reward coupling



Reward coupling time
- gain 20% for the first marginals
- utilization : best reduction

Monotonicity and perfect sampling : idea

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

Typically componentwise ordering on products of intervals

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

An event e is **monotone** if $\Phi(., e)$ is monotone on $\mathcal{X}$
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

The Doubling Scheme

Complexity

- Need to store the backward sequence of events
- Consider 2 trajectories issued from $\{min, Max\}$ at time $-n$ and test if coupling

  One step backward $\Rightarrow$

  $$2.(1+2+\cdots+\tau^*) = \tau^*(\tau^*+1) = \mathcal{O}({\tau^*}^2)$$

  calls to the transition function.
- Consider 2 trajectories issued from $\{min, Max\}$ at time -2^k and test if coupling

  Doubling step backward $\Rightarrow$

  $$2.(1+2+\cdots+2^k) = 2^{k+2}-2$$

  calls to the transition function, with k such that $2^{k-1} < \tau^* \leqslant 2^k$,

  Number of calls : $\mathcal{O}(\tau^*)$

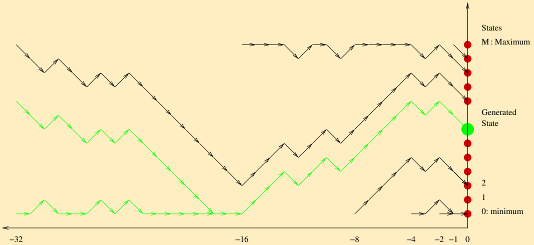
Monotonicity and Perfect Sampling

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

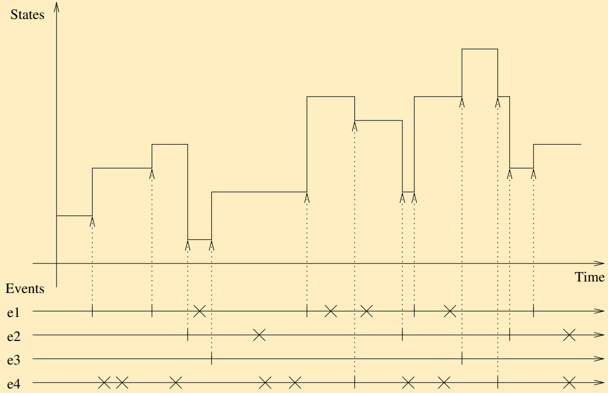
$C_m \leqslant 2.(2.\mathbb{E}\tau).c_\Phi$. Reduction factor : $\frac{4}{Card(\mathcal{X})}$.

Event Modelling

Multidimensional state space : $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_K$ with $\mathcal{X}_i = \{0, \cdots, C_i\}$. Event e :

$\rightsquigarrow$ transition function $\Phi(., e)$; (skip rule)

$\rightsquigarrow$ Poisson process λ_e



1781-1840

Event modelling

Uniformization

$$\Lambda = \sum_e \lambda_e \text{ and } \mathbb{P}(event\ e) = \frac{\lambda_e}{\Lambda};$$

Trajectory : $\{e_n\}_{n \in \mathbb{Z}}$ i.i.d. sequence.
⇒ Homogeneous Discrete Time Markov Chain [Bremaud 99]
$X_{n+1} = \Phi(X_n, e_{n+1})$.

Generation among a small finite space $\mathcal{E}$: $\mathcal{O}(1)$

Index Routing in Queuing Networks

Index functions for event e

For queue i $\quad I_i^e : \{0, \cdots, C_i\} \longrightarrow \mathcal{O}$ (totally ordered set).

Property : $\forall x_i, x_j \;\; I_i^e(x_i) \neq I_j^e(x_j)$.

ex: inverse of a priority,...

Routing algorithm:

if $x_{origin} > 0$ **then**
  { a client is available in the origin queue}
  $x_{origin} = x_{origin} - 1$; { the client is removed from the origin queue}
  $j = argmin_i \;\; I_i^e(x_i)$; { computation of the destination}
  if j $\neq$ -1 **then**
    $x_j = x_j + 1$; { arrival of the client in queue j }
    { in the other case, the client goes out of the network}
  end if
end if

Monotonicity of Index Routing Policies

Proposition

If all index functions I_i^e are monotone then event e is monotone.

Proof :

Let $x \prec y$ two states and let be an index routing event. Let i be the origin queue for the event.

$$j_x = argmin_j I_j^e(x_j) \text{ and } j_y = argmin_j I_j^e(y_j)$$

Case 1 $x_i = y_i = 0$ nothing happens and $\Phi(x, e) = x \prec y = \Phi(y, e)$

Case 2 $x_i = 0, \ y_i > 0$ then $\Phi(x, e) = x \prec y - e_i + e_{j_y} = \Phi(y, e)$

Case 3 $x_i > 0, \ y_i > 0$ then

$$I_{j_x}^e(x_{j_x}) < I_{j_y}^e(x_{j_y}) \leqslant I_{j_y}^e(y_{j_y}) < I_{j_x}^e(y_{j_x});$$

then $x_{j_x} < y_{j_x}$ and

$$\Phi(x, e) = x - e_i + e_{j_x} \leqslant y - e_i \leqslant y - e_i + e_{j_y} = \Phi(y, e)$$

Monotonicity of Routing

Examples [Glasserman and Yao]

All of these events could be expressed as index based routing policies :
- 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 : Examples

Stateless routing

Overflow routing

$$I_j^e(x_j) = \begin{cases} prio(j) & \text{if } x_j < C_j; \\ +\infty & \text{elsewhere} \end{cases}$$

$$I_{-1}^e = \max_j C_j.$$

Routing with blocking

$$I_j^e(x_j) = \begin{cases} prio(j) & \text{if } x_j < C_j; \\ +\infty & \text{elsewhere} \end{cases}$$

$$I_i^e = \max_j C_j.$$

State dependent routing

Join the shortest queue

$$I_j^e(x_j) = \begin{cases} x_j & \text{if } x_j < C_j; \\ +\infty & \text{elsewhere;} \end{cases}$$

$$I_{-1}^e = \max_j C_j.$$

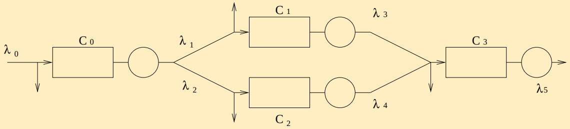
Join the shortest response time

$$I_j^e(x_j) = \begin{cases} \frac{x_j+1}{\mu_j} & \text{if } x_j < C_j; \\ +\infty & \text{elsewhere;} \end{cases}$$

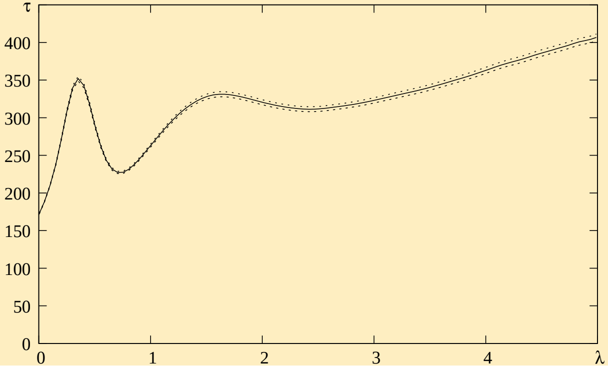
$$I_{-1}^e = \max_i C_i.$$

Coupling Experiment

Feed-forward queuing model



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



Main Result

Bound on coupling time

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

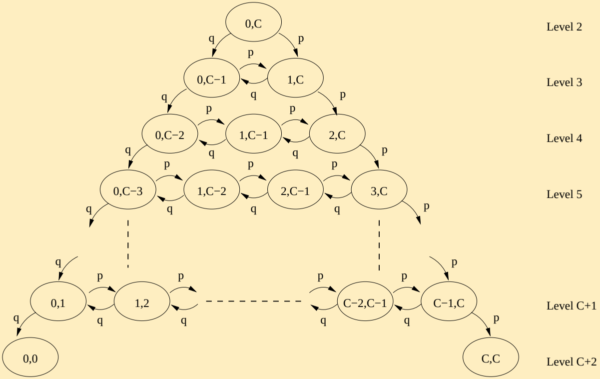
- Λ : global event rate in the network,
- Λ_i the rate of events affecting Q_i
- C_i is the capacity of Queue i.

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 ($\lambda < \mu$),

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

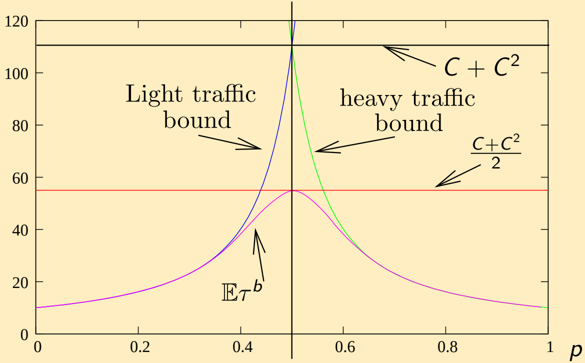
Theorem

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 $p = \lambda/(\lambda+\mu) = 1-q$.

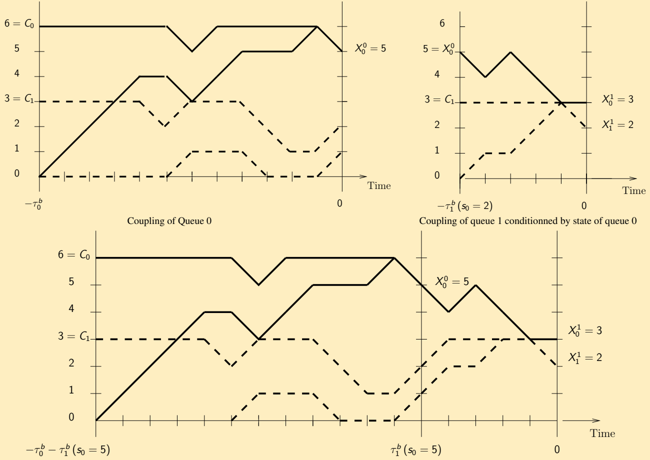
Critical bound:	$\forall p \in [0,1],$	$\mathbb{E}\tau^b \leqslant \frac{C^2+C}{2}.$
Heavy traffic Bound:	*if* $p > \frac{1}{2},$	$\mathbb{E}\tau^b \leqslant \frac{C}{p-q} - \frac{q(1-\left(\frac{q}{p}\right)^C)}{(p-q)^2}.$
Light traffic bound:	*if* $p < \frac{1}{2},$	$\mathbb{E}\tau^b \leqslant \frac{C}{q-p} - \frac{p(1-\left(\frac{p}{q}\right)^C)}{(q-p)^2}.$

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

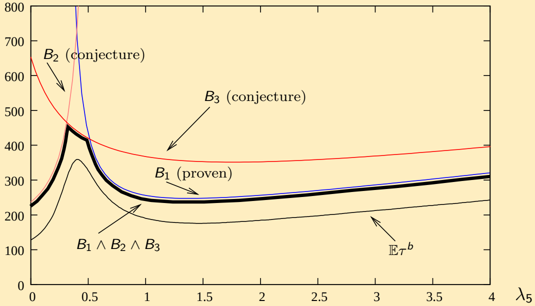
Theorem

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

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

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

Conjecture for General Networks



Extension to cyclic networks,
Generalization to several types of events
Application : Grid and call centers

Software architecture

Aim of the software

- finite capacity queuing network simulator
- rare events estimation (rejection, blocking,...)
- statistical guarantees (independence of samples)

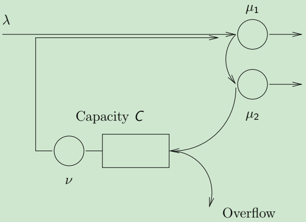
⇒ **Simulation kernel**

- open source (C, GPL licence)
- extensible library of events
- multiplatforms (Linux (Debian), mac OSX,...)

General architecture

Queueing Network Description

Constrained communications



Events types

type	action
1	Server departure
2	External arrival to the first empty room in the list DQ
3	Multi-server departure to DQ
4	Join the shortest queue in DQ
5	Index routing according an index table
6	Routing to the first empty room in the list DQ and overfl
7	Routing to the first empty room in the list DQ and blocking in the origin queue

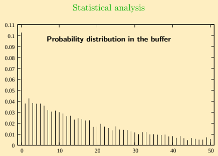
Description file

```
# Number of queues
3
# Queues capacities
1 1 50
# queues minimal initial state
0 0 0
# queues maximal initial state
1 1 50
# Number of events
4
# Index file - N for No index file
File:  N
# table of events
# id type rate nbq origin d1 d2 d3 d4
0 2 0.8 5 -1 :  0 1 2 -1
1 1 0.6 2 0 :  -1
2 1 0.4 2 1 :  -1
3 7 2.0 5 2 :  0 1 2 -1
```

Simulation control and output

Control parameters

```
# Sample number
10000
# Number of Antithetic variable
1
# Size of maximal trajectory
3000000
# Random generator seed
5
# Coupling file name
File:  No file
```

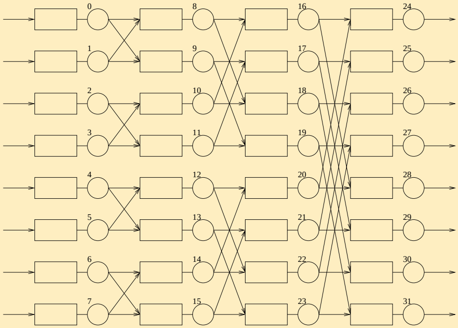


Output

```
# P.S.I.2 version 4.4.4
# Data Network model
# Number of queues
...
# Parameters
# Sample number
# 10000
# Number of Antithetic variates
...
# =============
0 [ [ 0 1 10 ] ]
1 [ [ 1 1 13 ] ]
2 [ [ 1 1 2 ] ]
3 [ [ 1 1 33 ] ]
...
9999 [ [ 1 1 2 ] ]
# Size 10000 Sampling time :
3809.202000 micro-seconds
# Seed Value 5
```

Example

Delta interconnection network, $C = 10$ $\rho = 0.9$




```
9999 [ [ 0 2 5 7 2 8 7 4 0 7 10 3 3 2 1 5 0 0 6 3 3 6 0 3
9 1 2 4 3 1 3 6 ] ]
# Size 10000 Sampling time :  4302.413600 micro-seconds
```

Monotonous reward sample

First server analysis

```
92 [ [ 0 ] ]
93 [ [ 1 ] ]
94 [ [ 1 ] ]
95 [ [ 1 ] ]
96 [ [ 1 ] ]
97 [ [ 1 ] ]
98 [ [ 1 ] ]
99 [ [ 1 ] ]
  # Size 100 Sampling time :  36.230000 micro-seconds
```

Time reduction

```
99 [ [ 1 1 1 ] ]
# Size 100 Sampling time :  308.100000 micro-seconds
```

Coupling time study (doubling scheme)

Coupling time for each queue

```
Perfect Simulation (with doubling period) started
0 6 2 1 6 [ 1 0 9 ]
1 7 5 1 7 [ 1 0 8 ]
2 8 2 6 8 [ 1 1 7 ]
3 8 1 2 8 [ 1 0 8 ]
4 8 1 1 8 [ 1 1 9 ]
5 7 7 4 7 [ 0 1 0 ]
6 6 1 2 6 [ 1 0 10 ]
7 8 1 1 8 [ 1 1 1 ]
```

Carefull : number of steps to couple ($\tau = 2^{\text{nb steps}}$)

Last queue have the largest coupling time.

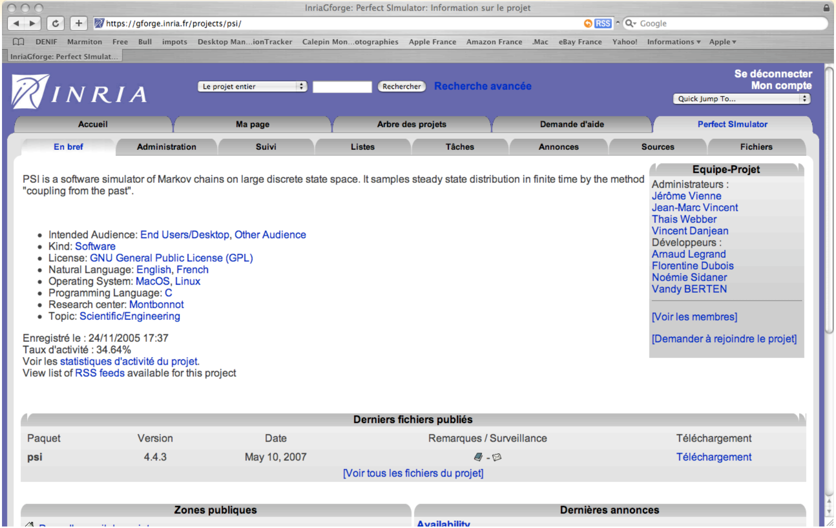
Coupling Time Study (one step scheme)

Coupling time for each queue

```
Perfect Simulation started 0 26 1 1 26 [ 1 0 10 ]
1 58 1 1 58 [ 0 0 10 ]
2 100 2 1 100 [ 1 0 8 ]
3 91 1 51 91 [ 1 1 2 ]
4 114 1 1 114 [ 1 1 6 ]
5 210 1 1 210 [ 1 1 9 ]
```

Distribution of the rewards coupling time

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



Synthesis

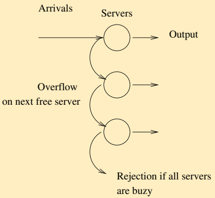
The psi2_unix command

```
USAGE : psi2_unix [-ipo] argument [-hdtv]
-i :  input file in ext directory
-p :  parameter file in ext directory
-o :  output file in ext directory
By default, output file has outputtest.txt name in ext
directory
-h :  help.
-d :  With details on output file
-t :  Without doubling period, show coupling time of each
queue
-v :  version
```

Enjoy !

Priority Servers

Erlang model



$\mathcal{X} = \{0, 1\}^3$
$\mathcal{E} = \{e_0, e_1, e_2, e_3\}$

$Card(\mathcal{X}) = 2^K$

Events

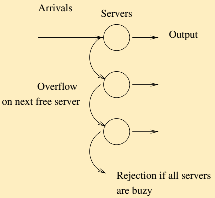
Event type	Rate	Origin	Destination list
Arrival	λ	-1	Q_1 ; Q_2 ; Q_3 ; -1
Departure	μ_1	Q_1	-1
Departure	μ_2	Q_2	-1
Departure	μ_3	Q_3	-1

Results

- Validation χ^2 test
- $K = 30$ μ_i decreasing
- Saturation probability 0.0579 ± 4.710^{-4}
- Simulation time $0.4ms$
- $\overline{\tau} = 577$

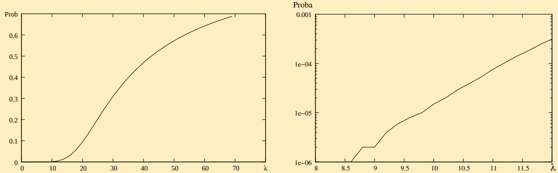
Priority servers

Erlang model

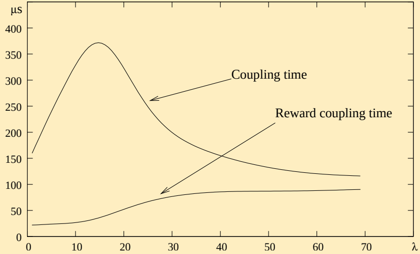


$\mathcal{X} = \{0,1\}^{40}$
$\mu_1 = 1,$
$\mu_2 = 0.8,$
$\mu_3 = 0.5$
Sample size 5.10^6
$Card(\mathcal{X}) = 2^K$

Saturation probability

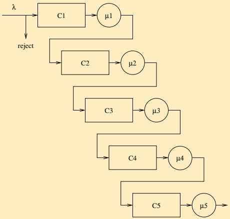


Coupling time



Line of Servers

Tandem queues



$\mathcal{X} = \{0, \cdots, 100\}^5$
$\mathcal{E} = \{e_0, \cdots, e_5\}$

$Card(\mathcal{X}) = C^K$

Events

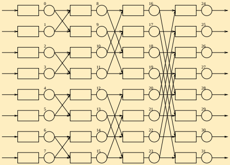
Event type	Rate	Origin	Dest. list
Arrival	λ	-1	Q_1 ; -1
Routing/block	μ_1	Q_1	Q_2 ; Q_1
Routing/block	μ_2	Q_2	Q_3 ; Q_2
...	...	...	...
Departure	μ_5	Q_5	-1

Results

- $C = 100$ $\lambda = 0.9$; $\mu = 1$ $p = \frac{1}{2}$
- Blocking probability $b_1 = 0.34$, $b_2 = 0.02$ $b_3 = 0.02$, $b_4, = 0.02$.
- Simulation time $< 1ms$

Multistage network

Delta network



$\mathcal{X} = \{0, \cdots, 100\}^{32}$
$\mathcal{E} = \{e_0, \cdots, e_{64}\}$

$Card(\mathcal{X}) = C^K$

Events

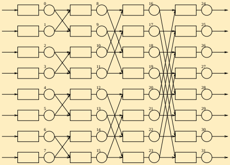
Event type	Rate	Origin	Dest. list
Arrival	λ	-1	Q_i ; -1
Routing/rejection	$\frac{1}{2}\mu$	Q_i	Q_j ; -1
...	...	...	...
Departure	μ	Q_k	-1

Results

- $C = 100$ $\lambda = 0.9$; $\mu = 1$
- Loss rate
- Simulation time 135ms
- $\overline{\tau} \simeq 400000$

Multistage network

Delta network

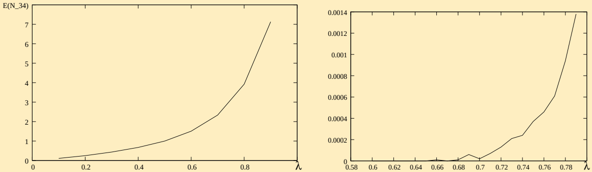


$$\mathcal{X} = \{0, \cdots, 100\}^{32}$$
$$\mathcal{E} = \{e_0, \cdots, e_{64}\}$$

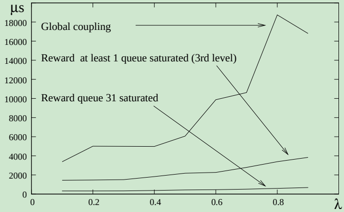
$Card(\mathcal{X}) = C^K$

Sample size 100000

Queue length and saturation proba at level 3

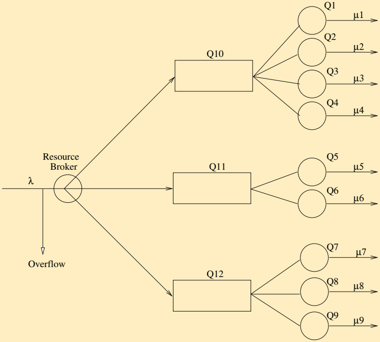


Coupling time



Resource Broker

Grid model



Input rates
Allocation strategy
State dependent allocation
Index based routing : destination minimize a criteria

Problem

Optimization of throughput, response time,...
Comparison of policies, analysis of heuristics
...

Routing Customers in Parallel Queues

The problem:

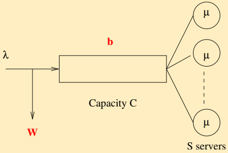
- Find a routing policy maximizing the expected (discounted) throughput of the system.
- Several variations on this problem depend on the information available to the controller: current size of all queues (and size of the arriving batch).

The applications:

- improve batch schedulers for cluster and grid infrastructures.
- Assert the value of information in such cases.

Index policies for routing

Optimal routing policy problem is still open for n different $M/M/1$
Heuristic : index policy inspired from the Multi-Armed Bandit
$\Rightarrow$ free parameter and compute an equilibrium point.
[Mitrani 2005] for routing and repair problems.



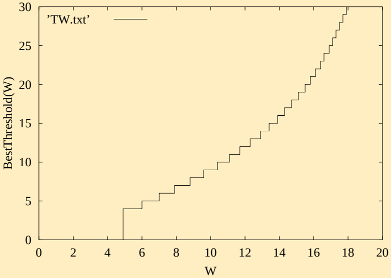
W is the rejection cost (free parameter).

Theorem

There is an optimal policy of threshold type:
there exists θ such that :Reject if $x \geqslant \theta$ and accept otherwise.
- θ does not depend on C as long as $C > \theta$ (including if C is infinite).
- θ is a non-decreasing function of W.

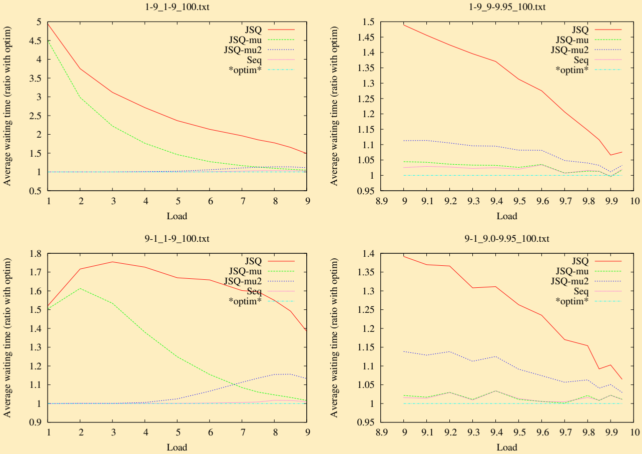
Index policies for Routing(II)

Computation of $\theta(W)$ linear system of corresponding to Bellman's equation, after uniformization.



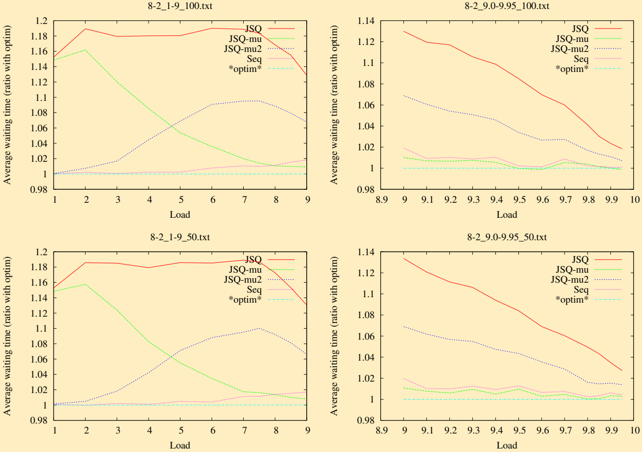
Index function $I(x) = \inf\{W \mid \theta(W) = x\}$.
Indifference case : when queue size is x, rejecting or accepting the next batch are both optimal choices if the rejection cost is $I(x)$.

Some numerical experiments(I)



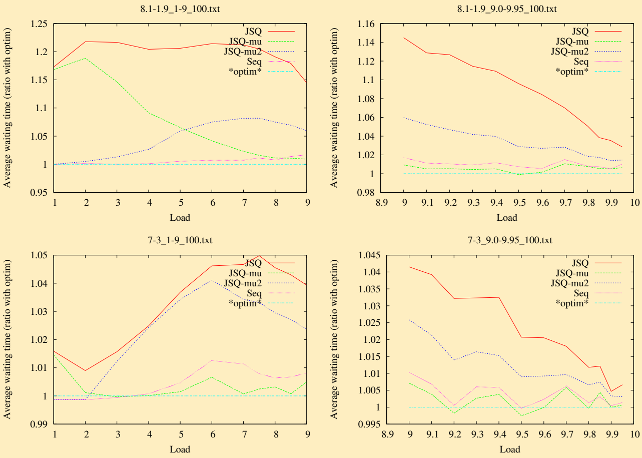
Several cases with two queues with respective parameters $(\mu = 9, 1)$, $C = 100$

Some numerical experiments(III)



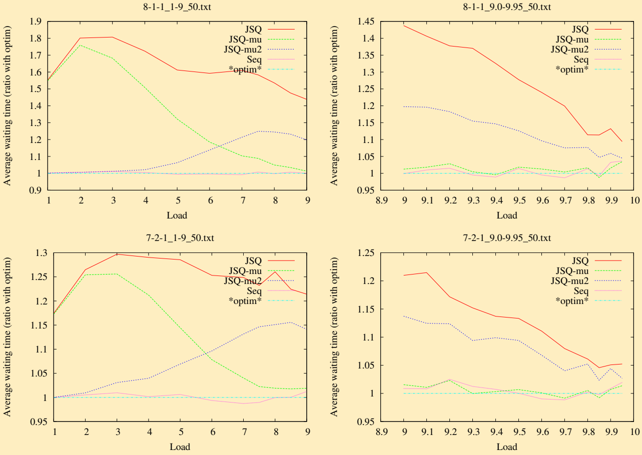
Several cases with two queues with respective parameters $(\mu = 8, 2)$, $C = 100$.

Some numerical experiments(IV)



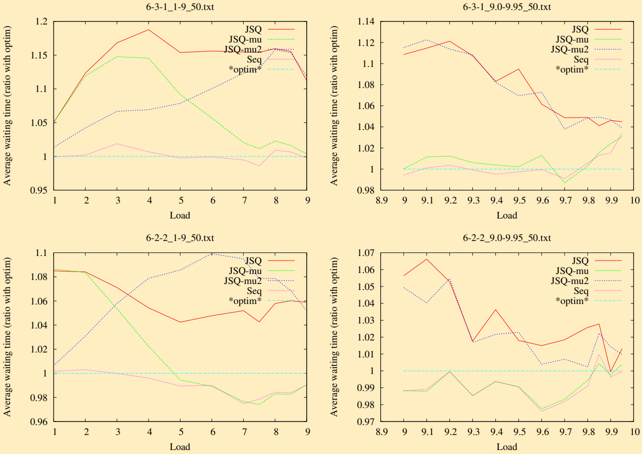
Some other cases

Some numerical experiments(V)



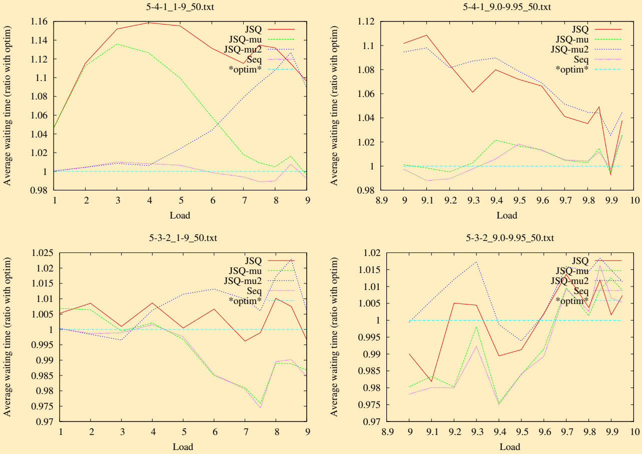
Three queues. Respectively, $\mu = 8, 1, 1$ and $\mu = 7, 2, 1$.

Numerical experiments(VI)



Now, $\mu = 6, 3, 1$ and $\mu = 6, 2, 2$.

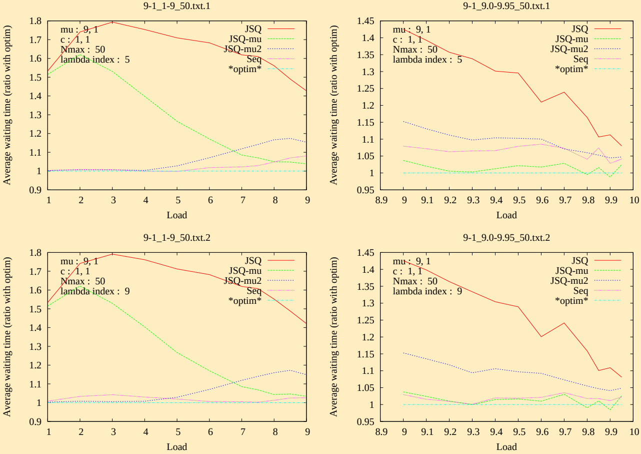
Numerical experiments(VII)



Now, $\mu = 5, 4, 1$ and $\mu = 5, 3, 2$.

Robustness of Index policies

The index policy was computed for $\lambda = 5$ or 9 and used over the whole range $\lambda = 1$ to 10.



Adaptation to Structured Models

Model Parameters

- set of uniformized events $E = \{e_1, .., e_p\}$
- global states are tuples of local states $\tilde{s} = (s_1, \ldots, s_K)$
- transition function: $\Phi(\tilde{s}, e_i) = \tilde{r}$

* each $\tilde{s} \in \mathcal{X}$ has a set of enabled events and its firing conditions and consequences

Constraints

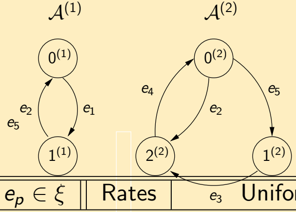
- Well-formed SAN models needed

* exploring the subset $\mathcal{X}^{\mathcal{R}}$ (Reachable state space)

- State space explosion still a problem

Analysis of Complex Discrete Systems

GRAPHICAL MODEL
(STATES + TRANSITIONS)



$\tilde{s} \in \mathcal{X}^{\mathcal{R}}$	$\Phi(\tilde{s}, e_1)$	$\Phi(\tilde{s}, e_2)$	$\Phi(\tilde{s}, e_3)$	$\Phi(\tilde{s}, e_4)$
	$\tilde{r} = \Phi(\tilde{s}, e_p), e_p \in \xi$			
{0;0}	**{1;0}**	{0;0}	{0;0}	{0;0}
{0;1}	**{1;1}**	{0;1}	**{0;2}**	{0;1}
{0;2}	**{1;2}**	{0;2}	{0;2}	**{0;0}**
{1;0}	{1;0}	**{0,2}**	{1;0}	{1;0}
{1;1}	{1;1}	{1;1}	**{1;2}**	{1;1}
{1;2}	{1;2}	{1;2}	{1;2}	**{1;0}**

$e_p \in \xi$	Rates	Uniformized Rates
e_1	λ_1	$\lambda_1/(\lambda_1+\lambda_2+\lambda_3+\lambda_4+\lambda_5)$
e_2	λ_2	$\lambda_2/(\lambda_1+\lambda_2+\lambda_3+\lambda_4+\lambda_5)$
e_3	λ_3	$\lambda_3/(\lambda_1+\lambda_2+\lambda_3+\lambda_4+\lambda_5)$
e_4	λ_4	$\lambda_4/(\lambda_1+\lambda_2+\lambda_3+\lambda_4+\lambda_5)$
e_5	λ_5	$\lambda_5/(\lambda_1+\lambda_2+\lambda_3+\lambda_4+\lambda_5)$

New solutions for huge SAN models

Monotonicity and Perfect Simulation Idea

- Monotonicity property for SAN related to the analysis of structural conditions
* component-wise state space formation

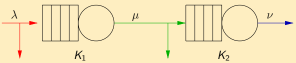
Families of SAN models

- SAN models with a natural partial order (canonical)
* e.g. derived from Queueing systems models [Vincent 2005]
- SAN models with a given component-wise partial order (non-lattice)

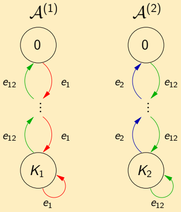
Partially Ordered State Spaces

Canonical component-wise ordering

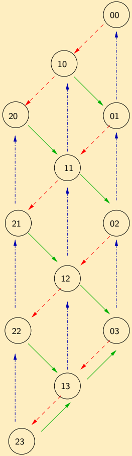
Queueing Network Model



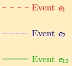
Equivalent SAN Model



Type	Event	Rate
loc	e_1	λ
syn	e_{12}	μ
loc	e_2	ν



Equivalent MC



Partially Ordered State Spaces

Non-lattice component-wise ordering

- Find a partial order of $\mathcal{X}$ demands a high c.c.
- Possible to find extremal global states in the underlying chain
- * $|\mathcal{X}^{\mathcal{M}}|$ states: more than two extremal states
- Complexity: related to τ, but also $|\mathcal{X}^{\mathcal{M}}|$

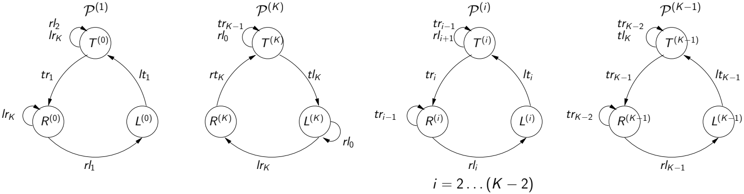
Extremal states

- Component-wise formation has ordered state indexes
- * consider an initial state composing $\mathcal{X}^{\mathcal{M}}$
- * add to $\mathcal{X}^{\mathcal{M}}$ the states without transitions to states with greater indexes

Non-lattice component-wise ordering

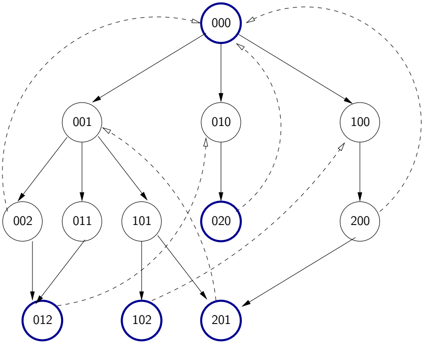
- Resource sharing model with reservation (Dining Philosophers)

Type	Event	Rate
loc	lt_i	μ
syn	tr_i	λ
syn	rl_i	λ
loc	rt_K	μ
syn	tl_K	λ
syn	lr_K	λ



Non-lattice component-wise ordering

- e.g. three philosophers with resources reservation, graphical model of the underlying transition chain, extremal states identification



SAN Perfect Simulation

Resource sharing model with reservation- K Philosophers

K	$\mathcal{X}$	$\mathcal{X}^{\mathcal{R}}$	$\mathcal{X}^{\mathcal{M}}$	*PEPS** (iteration)	*Perfect PEPS* (sample)*
8	6,561	985	43	0.003185 sec.	0.032354 sec.
10	59,049	5,741	111	0.038100 sec.	0.111365 sec.
12	531,441	33,461	289	0.551290 sec.	0.689674 sec.
14	4,782,969	195,025	755	5.712210 sec.	2.686925 sec.
16	43,046,721	1,136,689	1,975	68.704325 sec.	15.793501 sec.
18	**387,420,489**	6,625,109	**5,169**	—-	83.287321 sec.

Numerical results

- 3.2 GHz Intel Xeon processor under Linux, 1 GByte RAM
- times: for one iteration on PEPS and for one sample generation on Perfect PEPS
- Remarks: $\mathcal{X}$ contraction in $|\mathcal{X}^{\mathcal{M}}|$
- $\mathcal{X}$ limitation 6×10^7 onPEPS overcame

Non Monotonic Systems

Classic non monotonous events

- Batch arrivals
- Batch services
- Join procedures
- Negative customers
- ...

Almost monotonous events

Monotonicity is not verified at the frontier

- Batch arrivals : batch rejection (queue full)
- Batch services : system almost empty
- Join procedures : system almost empty
- Negative customers : system almost empty

Envelopes

Aim : Build a monotonous upper and lower process

Hypothesis : $\mathcal{X}$ is lattice, $T = \max \mathcal{X}$ and $B = \min \mathcal{X}$

$$\mathcal{U}(M, m, e) = \sup_{m \leqslant s \leqslant M} \Phi(s, e);$$

$$\mathcal{L}(M, m, e) = \inf_{m \leqslant s \leqslant M} \Phi(s, e).$$

Remark : if e is monotonous $\mathcal{U}(M, m, e) = \Phi(M, e)$ and $\mathcal{L}(M, m, e) = \Phi(m, e)$

Bounding process :

$$Y_0 = T \text{ and } Z_0 = B;$$

$$Y_n = \mathcal{U}(Y_{n-1}, Z_{n-1}, e_{1 \to n}) \text{ and } Z_n = \mathcal{L}(Y_{n-1}, Z_{n-1}, e_{1 \to n}).$$

Remark : Y_n and Z_n are not Markov chains but the couple is Markov.

$$Y_n \geqslant X_n \geqslant Z_n.$$

Envelopes (2)

Theorem (Convergence)

Either the process (Y_n, Z_n) hits the diagonal in finite time with probability 1 or nether hits the diagonal. When the process hits the diagonal the value is stationary distributed.

Problems

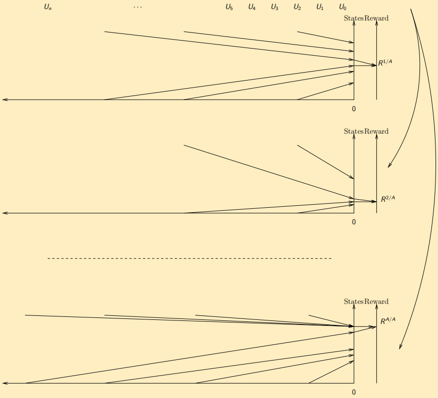
(P_1) The assumption that S_n hits the diagonal may not be verified.

(P_2) Even if convergence theorem, the coupling time could become prohibitively large.

(P_3) The time needed to compute $\mathcal{U}(M, m, e)$ and $\mathcal{L}(M, m, e)$ might depend on the number of states between m and M

Variance Reduction

Coupled trajectories



Coupled samples hope : negatively correlated

Variance Reduction

3 coupled trajectories

```
Perfect Simulation (with doubling period) -= Antithetic
with 3 variables =- started
0 [ [ 1 0 10 ] [ 1 1 0 ] [ 1 1 7 ] ]
1 [ [ 1 1 10 ] [ 1 1 3 ] [ 1 0 0 ] ]
2 [ [ 1 0 9 ] [ 1 1 0 ] [ 1 1 10 ] ]
3 [ [ 1 1 0 ] [ 1 1 7 ] [ 1 1 0 ] ]
4 [ [ 1 1 9 ] [ 1 0 0 ] [ 0 1 3 ] ]
5 [ [ 1 0 6 ] [ 1 1 7 ] [ 0 1 2 ] ]
6 [ [ 1 1 4 ] [ 1 1 1 ] [ 0 0 10 ] ]
7 [ [ 1 1 4 ] [ 1 0 6 ] [ 1 1 6 ] ]
8 [ [ 1 0 8 ] [ 0 1 10 ] [ 1 1 9 ] ]
9 [ [ 1 1 0 ] [ 1 1 0 ] [ 1 1 8 ] ]
10 [ [ 1 1 6 ] [ 1 1 6 ] [ 1 1 8 ] ]
11 [ [ 0 1 3 ] [ 1 1 6 ] [ 1 1 5 ] ]
```

Correlation analysis ⇒ variance reduction

example $Var X_0 > Var(X_0 + X_0' + X_0'')/3$

Synthesis

Exact sampling

- Poisson systems (uniformization) independence of events (SRS)
- Reversed process $\Rightarrow$ exact criteria (convergence proof)
- Polynomial coupling time in the size of the models
- Monotonicity and contraction on sets $\Rightarrow$ RIFS and fractals

$\Rightarrow$ **model structure**

Perfect samplers

- General DTMC : $\mathcal{O}(\mathbb{E}\tau^*.|\mathcal{X}|)$
- Monotone DTMC : $\mathcal{O}(\mathbb{E}\tau^*.|\mathcal{E}xt|)$
- Monotone DTMC lattice : $\mathcal{O}(\mathbb{E}\tau^*)$

$\Rightarrow$ **numerically tractable**

Other approaches in perfect sampling

Algorithms

- Forward/backward algorithm [Fill et al]
- Horizontal sampling [Foss et al]
- Read once samplers
- ...

Application contexts

- Interacting particle systems (statistical physics)
- Stochastic geometry
- Networking [Le Boudec et al]
- Samplers of complex distributions
- ...

Open problems

Method

- Optimal coupling problem (general case : event decomposition)
- Infinite state space : coupling condition
- Non-monotone systems
- Transformation of generators

Models

- Structured models : partial order construction
- Structured models : from description to efficient event decomposition
- $(Max, +)$ dynamical systems (Petri nets)
- $\ldots$

Software

Integration in general modeling framework

Bibliography

Books

- P. Brémaud Markov chains: Gibbs fields, Monte Carlo Simulation and Queues. Springer-Verlag, 1999
- O. Haggstrom. Finite Markov Chains and algorithmic applications. 2001. Matematisk Statistik, Chalmers teknisk hogshola och Goteborgs universitet.
- P. Glasserman and D. Yao. Monotone Structure in Discrete-Event Systems. John Wiley and Sons, 1994.
- T. Lindvall Lectures on the Coupling Method John Wiley and Sons 1992

Websites

- David Wilson's web site http://dimacs.rutgers.edu/∼dbwilson/exact

Basic references and application papers

Fundamental papers among (in my opinion)

- D. Propp, J.and Wilson. Exact sampling with coupled Markov chains and applications to statistical mechanics. *Random Structures and Algorithms*, 9(1&2):223–252, 1996.
- A. Borovkov and S. Foss. Two ergodicity criteria for stochastically recursive sequences. *Acta Appl. Math.*, 34, 1994.
- P. Diaconis and D. Freedman. Iterated random functions. *SIAM Review*, 41(1):45–76, 1999.
- V. Craiu and X.-L. Meng. Multiprocess parallel antithetic coupling for backward and forward Markov chain Monte Carlo. *Annals of Statistics*, 33(2):661–697, 2005.
- J.A. Fill. An interruptible algorithm for perfect sampling via Markov chains, *Annals of Applied Probability*, 8:131–162,1998.
- O. Stenflo. Ergodic theorems for markov chains represented by iterated function systems. *Bull. Polish Acad. Sci. Math*, 2001.
- A. Walker. An efficient method for generating discrete random variables with general distributions. *ACM Trans. Math. Software*, 3:253–256, 1974.

Some application papers in computer science

- B. Gaujal and F. Perronnin. Perfect simulation of stochastic hybrid systems with an application to peer to peer systems". *Journal of Discrete Event Dynamic Systems*, 2007. Special issue on hybrid systems.
- F. Baccelli, B. Blaszczyszyn, and F. Tournois. Spatial averages of coverage characteristics in large cdma networks. *Wirel. Netw.*, 8(6):569–586, 2002.
- V. Berten and B. Gaujal. Brokering strategies in computational grids using stochastic prediction models. *Parallel Computing*, 2007. Special Issue on Large Scale Grids.
- J.-Y. Le Boudec and M. Vojnovic. The Random Trip Model: Stability, Stationary Regime, and Perfect Simulation The Random Trip Model: Stability, Stationary Regime, and Perfect Simulation. *IEEE/ACM Transactions on Networking*, 14(6):1153–1166, 2006.
- A. Bouillard and B. Gaujal. Backward coupling for perfect simulation of free-choice nets. *Journal of Discrete Event Dynamics Systems, theory and applications*, 2006. Special issue of selected papers from the Valuetools conference.

Some references

Discrete time Markov chains

- J.-M. Vincent and C. Marchand. On the exact simulation of functionals of stationary markov chains. *Linear Algebra and its Applications*, 386:285–310, 2004.
- J.-M. Vincent. Markov chains, iterated systems of functions and coupling time for perfect simulation. In *Transgressive Computing*, pages 387–398, Grenada, Apr. 2006.
- I. Y. Kadi, J.-M. Fourneau, N. Pekergin, J. Vienne, and J.-M. Vincent. Perfect simulation and monotone stochastic bounds. In *VALUETOOLS*, Nantes, France, Oct. 2007

Finite queueing Networks

- J.-M. Vincent. Perfect simulation of queueing networks with blocking and rejection. In *Saint IEEE conference*, pages 268–271, Trento, 2005.
- J.-M. Vincent and J. Vienne. Perfect simulation of index based routing queueing networks. *Performance Evaluation Review*, 34(2):24–25, 2006.
- J. Dopper, B. Gaujal, and J.-M. Vincent. Bounds for the coupling time in queueing networks perfect simulation. In *Numerical Solutions for Markov Chain (NSMC06)*, pages 117–136, Charleston, June 2006.

Low probability events estimation

- J.-M. Vincent. Perfect simulation of monotone systems for rare event probability estimation. In *Winter Simulation Conference*, Orlando, Dec. 2005.
- J.-M. Vincent and J. Vienne. Perfect simulation of monotone systems with variance reduction. In *Proceedings of the 6th Int. Workshop on Rare Event Simulation*, pages 275–285, Bamberg, Oct. 2006.

Relaxing monotonicity properties

- P. Fernandes, J.-M. Vincent, and T. Webber. Perfect simulation of stochastic automata networks. In *ASMTA Conference*, Nicosie, jun 2008.
- A. Busic, B. Gaujal and J.-M. Vincent Perfect simulation and non-monotone Markovian systems Valuetools 2008

Software tools

- J.-M. Vincent and J. Vienne. Psi2 a software tool for the perfect simulation of finite queueing networks. In *QEST*, Edinburgh, Sept. 2007.

Thanks

- Searchers and students
  Vandy Berten, Ana Busic, Jantien Dopper, Florentine Dubois, Paulo Fernandes, Bruno Gaujal, Gaël Gorgo, Imène Kadi, Corine Marchand, Florent Morata, Nihal Pekerhin, Noémie Sidaner, Jérôme Vienne, Thaïs Webber,
- Research groups
  ANR Checkbound, ACI SurePath, ANR SMS.
- Technical suggestions and software support
  Vincent Danjean, Guillaume Huard, Arnaud Legrand, Nicolas Maillard, gforge.inria.fr,
- Images
  `http://www.gap-system.org/~history/Biographies/`