

# Filtered Gibbs Sampler for Estimating Blocking Probabilities in WDM Optical Networks

**Felisa J. Vázquez-Abad and Lachlan L. H. Andrew**

Département d'informatique et recherche opérationnelle

Université de Montréal, Québec, CANADA

courriel: vazquez@IRO.UMontreal.CA

Department of Electronic and Electrical Engineering

The University of Melbourne

email: {fva, lha}@ee.mu.oz.au

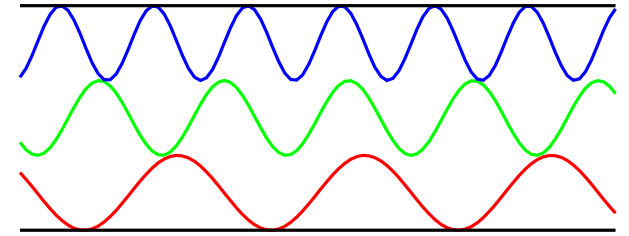
*European Simulation Multiconference, 25 May 2000.*

## Outline of Presentation

1. Motivation
  - WDM optical networks
2. Clique packing
  - Stationary measure
  - Blocking probability
3. Monte Carlo simulation
  - Accept/reject Monte Carlo
  - Markov chain Monte Carlo
4. The Gibbs sampler
  - Periodic Gibbs
  - Filtered sequential Gibbs sampler
5. Future work

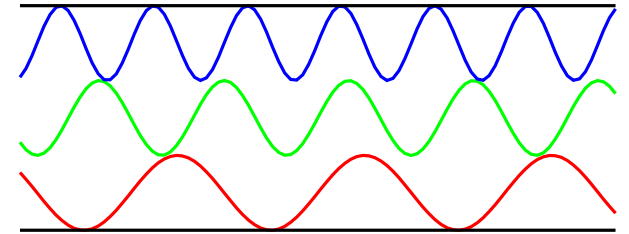
## 1.1 Motivation: WDM Optical Networks

- Optical bandwidth  $\gg$  electronic bandwidth.
- Wavelength division multiplexing (WDM):
  - $\Lambda$  independent wavelengths per fibre
  - Each wavelength modulated separately
- *Crossconnects*: at nodes act as space switches, they can **also switch wavelengths**.

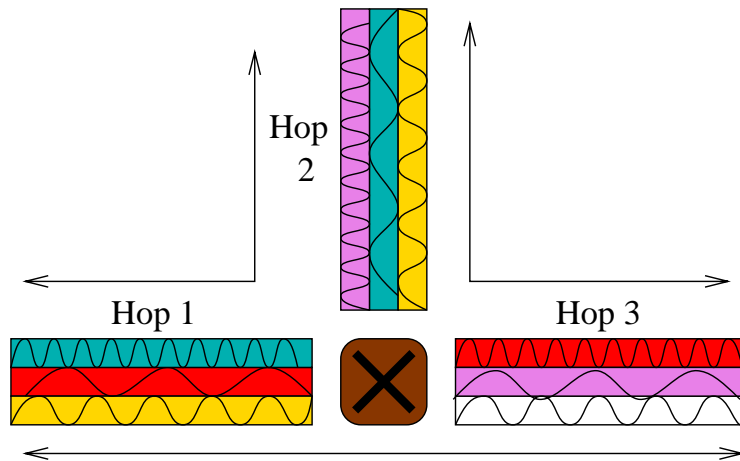


## 1.1 Motivation: WDM Optical Networks

- Optical bandwidth  $\gg$  electronic bandwidth.
- Wavelength division multiplexing (WDM):
  - $\Lambda$  independent wavelengths per fibre
  - Each wavelength modulated separately



- *Crossconnects*: at nodes act as space switches, they can **also switch wavelengths**.

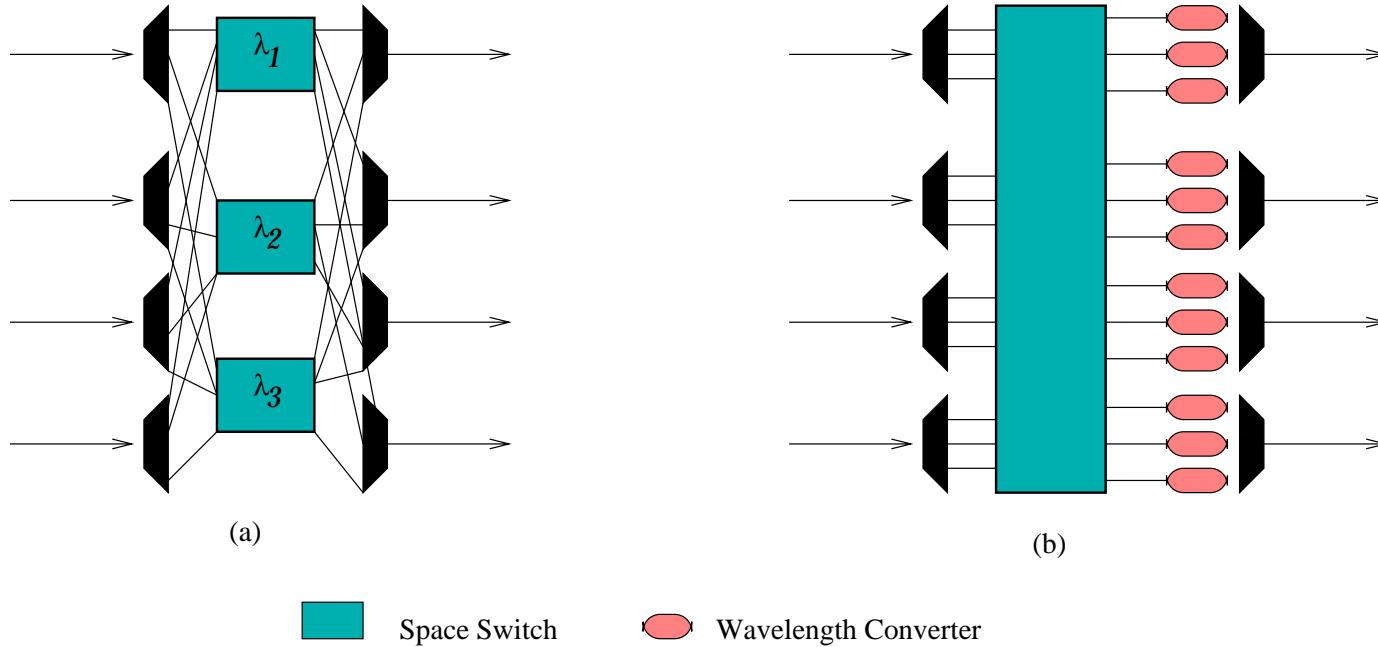


Lightpaths are shown in different shades of colour.

- Optical carriers within fibres are *wavelengths*.
- Calls are connected using optical carriers along the links on their paths: *lightpath*.
- Connected calls use the bandwidth of each carrier wavelength along the lightpath.

## 1.1 Motivation: Crossconnects

Full wavelength conversion  $\Rightarrow$  standard *circuit switched loss network*



$M$  input and output fibres with  $W$  wavelengths on each, requirements:

- wavelength *continuous* crossconnect:  $W$  different  $M \times M$  space switches,
- wavelength *conversion* crossconnect: a single  $MW \times MW$  space switch.

**VERY EXPENSIVE !!!!!**

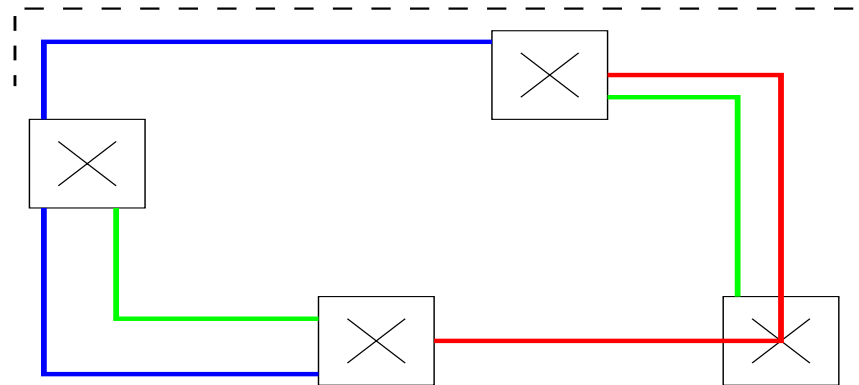
## 1.1 Motivation: Resource Allocation

### *Demand Model:*

Call *arrivals* to lightpath  $i$  follow a Point process  $N_i(t)$  with intensity  $\lambda_i$  (e.g. Poisson). Call durations: i.i.d *holding times* with mean  $1/\mu$ .

### *Resources:*

No (or partial) wavelength conversion : *wavelength continuity constraints*.  
Calls compete for bandwidth.



- *Dynamic allocation* of lightpaths
  - Several methods available to assign LPs to incoming calls
  - **Problem:** Analysis and evaluation difficult (unless full conversion)

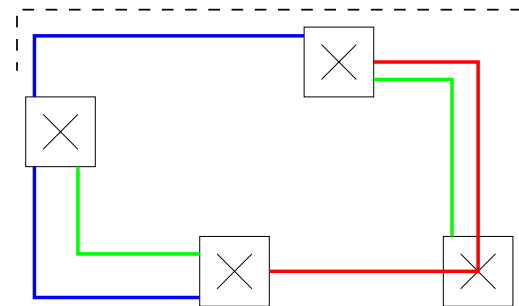
## 1.1 Motivation: Maximum and Clique Packing

*Dynamic lightpath allocation:* Wavelength continuity constraint (no conversion). How to assign lightpaths to incoming calls at route  $i$ ?

- Call arrives, search available wavelength (say First Fit assignment).
- No wavelength available on path .... *reject???*

### Fast tuning devices:

Optical carriers can (in principle) change wavelength of on-going connections without affecting QoS.



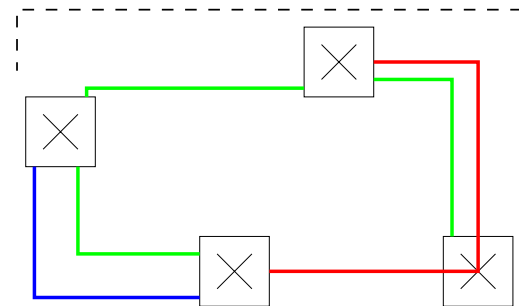
## 1.1 Motivation: Maximum and Clique Packing

*Dynamic lightpath allocation:* Wavelength continuity constraint (no conversion). How to assign lightpaths to incoming calls at route  $i$ ?

- Call arrives, search available wavelength (say First Fit assignment).
- No wavelength available on path .... *reject???*

### Fast tuning devices:

Optical carriers can (in principle) change wavelength of on-going connections without affecting QoS.



**Maximum packing:** Fast tuning devices  $\Rightarrow$  rearrangement of wavelengths. Calls on route  $i$  connected if, upon rearrangement, there is a wavelength available.



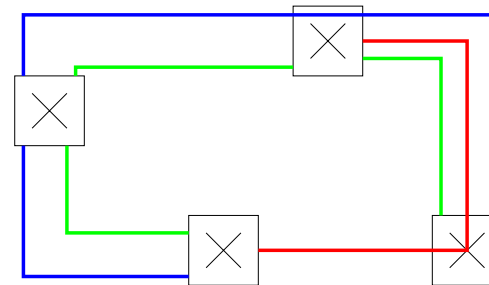
## 1.1 Motivation: Maximum and Clique Packing

*Dynamic lightpath allocation:* Wavelength continuity constraint (no conversion). How to assign lightpaths to incoming calls at route  $i$ ?

- Call arrives, search available wavelength (say First Fit assignment).
- No wavelength available on path .... *reject???*

### Fast tuning devices:

Optical carriers can (in principle) change wavelength of on-going connections without affecting QoS.



**Maximum packing:** Fast tuning devices  $\Rightarrow$  rearrangement of wavelengths. Calls on route  $i$  connected if, upon rearrangement, there is a wavelength available. State description: occupancy, complex coupling equations.

**Analysis:** Complex model for analytical results, state space too large.

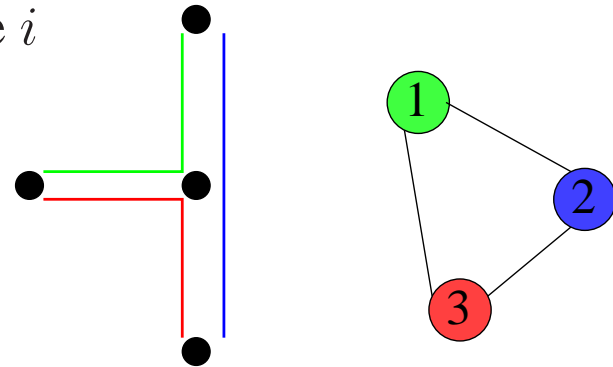
Simplified model: clique packing yields simple linear constraints

## 2.0 Clique Packing in WDM Optical Networks

- $R$  = number of routes in network (number of O/D pairs if fixed routing)
- $n_i$  = number of calls currently using route  $i$

**Cliques** Graph  $\mathcal{G} = (V, E)$

- $V$ : vertices = routes
- $E$ : edge if routes share a link
- **Clique**: fully connected subgraph of  $\mathcal{G}$ .



**Maximum packing** Fast tuning devices: Allocate incoming calls whenever possible, allowing rearrangement  $\Rightarrow$  ( $n$ -colouring of  $\mathcal{G}$ )

**Clique packing** assumes that incoming calls can be connected iff

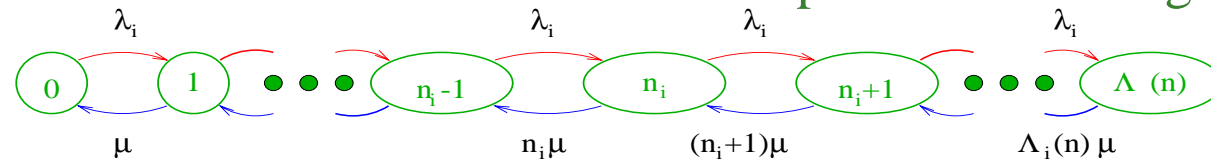
$$\sum_{j \in \mathcal{C}_l} n_j < \Lambda \text{ for all } l \text{ with } j \in \mathcal{C}_l$$

*Simplified Model:*

Occupancy vector  $n_i(t)$  follows stochastic process: independent Poisson arrivals and i.i.d. holding times (not  $M/G/\infty$  server... **boundaries!**)

## 2.1 Analysis of clique packing: stationary measure

**Model** Arrivals  $\sim \text{Poisson}(\lambda_i)$ , holding times  $\sim \exp(\mu)$ ,  $\{n(t)\}$  occupancy process: each dimension B&D with **state dependent reflecting boundaries**.



**Result** The limit occupancy distribution (stationary probabilities) are:

$$\pi(n) = \frac{1}{G} \prod_{i=1}^R \left( \frac{\rho_i^{n_i}}{n_i!} \right), \quad n \in \mathcal{S}$$

$$\mathcal{S} = \left\{ n \in \mathbf{N}^R : \sum_{j \in \mathcal{C}_l} n_j \leq \Lambda; \quad l = 1, \dots, L \right\}$$

$$G = \sum_{n \in \mathcal{S}} \prod_{i=1}^R \left( \frac{\rho_i^{n_i}}{n_i!} \right)$$

**Result:** This result may be generalised for other renewal arrival processes and holding time distribution.

## 2.2 Blocking probability

$$B = \lim_{t \rightarrow \infty} \sum_{i=1}^R \frac{Y_i(t)}{A(t)}$$

$Y_i(t)$  = number of lost arrivals on route  $i$  at time  $t$

$A(t)$  = total number of arrivals at time  $t$ .

**Blocking states on route  $i$ :** states  $n \in \mathcal{B}_i \Rightarrow$  incoming calls at  $i$  are lost:

$$\mathcal{B}_i = \left\{ n \in S : \max_{\{l:i \in \mathcal{C}_l\}} \sum_{j \in \mathcal{C}_l} n_j = \Lambda \right\}$$

$$B = \sum_{i=1}^R \left( \frac{\lambda_i}{\lambda} \right) \pi(\mathcal{B}_i)$$

...solved the problem?

## 2.2 Blocking probability

$$B = \lim_{t \rightarrow \infty} \sum_{i=1}^R \frac{Y_i(t)}{A(t)}$$

$Y_i(t)$  = number of lost arrivals on route  $i$  at time  $t$

$A(t)$  = total number of arrivals at time  $t$ .

**Blocking states on route  $i$ :** states  $n \in \mathcal{B}_i \Rightarrow$  incoming calls at  $i$  are lost:

$$\mathcal{B}_i = \left\{ n \in S : \max_{\{l:i \in \mathcal{C}_l\}} \sum_{j \in \mathcal{C}_l} n_j = \Lambda \right\}$$

$$B = \sum_{i=1}^R \left( \frac{\lambda_i}{\lambda} \right) \pi(\mathcal{B}_i)$$

...solved the problem?

Realistic network sizes:  $> 20$  nodes, 8–64 wavelengths,  $R = n^2/2 + o(n^2)$   
 # states  $\approx O(\Lambda^R)$ . For 10 nodes and 8 wavelengths, computation of  $G$  requires  
 $\approx 8^{45} \approx 10^{40}$  **multiplications**,

## 2.2 Blocking probability

$$B = \lim_{t \rightarrow \infty} \sum_{i=1}^R \frac{Y_i(t)}{A(t)}$$

$Y_i(t)$  = number of lost arrivals on route  $i$  at time  $t$   
 $A(t)$  = total number of arrivals at time  $t$ .

Blocking states on route  $i$ : states  $n \in \mathcal{B}_i \Rightarrow$  incoming calls at  $i$  are lost:

$$\mathcal{B}_i = \left\{ n \in S : \max_{\{l:i \in \mathcal{C}_l\}} \sum_{j \in \mathcal{C}_l} n_j = \Lambda \right\}$$

$$B = \sum_{i=1}^R \left( \frac{\lambda_i}{\lambda} \right) \pi(\mathcal{B}_i)$$

...solved the problem?

Realistic network sizes:  $> 20$  nodes, 8–64 wavelengths,  $R = n^2/2 + o(n^2)$   
 # states  $\approx O(\Lambda^R)$ . For 10 nodes and 8 wavelengths, computation of  $G$  requires  
 $\approx 8^{45} \approx 10^{40}$  multiplications, which takes

$10^{21}$  years of CPU time on a 1 TFlops computer...



### 3. Simulation methods: Monte Carlo

**Idea:** Estimate  $B$  directly, rather than find  $G$  then  $B$

$$B = \sum_{i=1}^R \left( \frac{\lambda_i}{\lambda} \right) \mathbf{E}(\mathbf{1}_{\{X \in \mathcal{B}_i\}}), \quad X \sim \pi$$

*Simulation:*

- Generate a sample  $\{X_1, \dots, X_N\}$  i.i.d.,  $X_i \sim \pi$
- Use the sample average:

$$\hat{Y}(N) = \frac{1}{N} \sum_{i=1}^R \left( \frac{\lambda_i}{\lambda} \right) \sum_{s=1}^N \mathbf{1}_{\{X_s \in \mathcal{B}_i\}}$$

### 3. Simulation methods: Efficiency

$$\hat{Y}(N) = \frac{1}{N} \sum_{i=1}^R \left( \frac{\lambda_i}{\lambda} \right) \sum_{s=1}^N \mathbf{1}_{\{X_s \in \mathcal{B}_i\}}$$

LLN and CLT  $\Rightarrow$  confidence intervals can be estimated to give approximate error  $\epsilon = z_{1-\alpha/2} \sqrt{\text{Var}(\hat{Y}(N))} \Rightarrow$

$$\text{Relative error} \approx \sqrt{\frac{\text{Var}\hat{Y}(N)}{B^2}}$$

**Definition:** *Relative efficiency* of estimator  $\hat{Y}(N)$ :

$$\mathcal{E}_r(\hat{Y}(N)) = \frac{B^2}{\text{CPU}[\hat{Y}(N)] \text{Var}[\hat{Y}(N)]}$$

Trade-off between *relative error* and *CPU time*.



## 4. Monte Carlo Simulation: Acceptance/Rejection

Generate  $X_k \sim \pi(\cdot)$ ,  $\{X_k\}$  i.i.d. and use

$$Y_k = \sum_{i=1}^R \left( \frac{\lambda_i}{\lambda} \right) \mathbf{1}_{\{X_k \in \mathcal{B}_i\}}$$

$\pi$  is a state-dependent truncated Poisson  $\pi(n) = \frac{1}{G} \prod_{i=1}^R \frac{\rho_i^{n_i}}{n_i!}$ ,  $n \in S$

Accept/Reject:

- Repeat
  - Generate  $(m_1, \dots, m_R)$  independent,  
 $m_i \sim \text{Poisson}(\rho_i)$
- until  $m \in S$
- Set  $X_k(i) = m_i$

$X_k$  are i.i.d.  $\sim \pi$ .

### 4. Monte Carlo Simulation: Acceptance/Rejection

Generate  $X_k \sim \pi(\cdot)$ ,  $\{X_k\}$  i.i.d. and use

$$Y_k = \sum_{i=1}^R \left( \frac{\lambda_i}{\lambda} \right) \mathbf{1}_{\{X_k \in \mathcal{B}_i\}}$$

$\pi$  is a state-dependent truncated Poisson

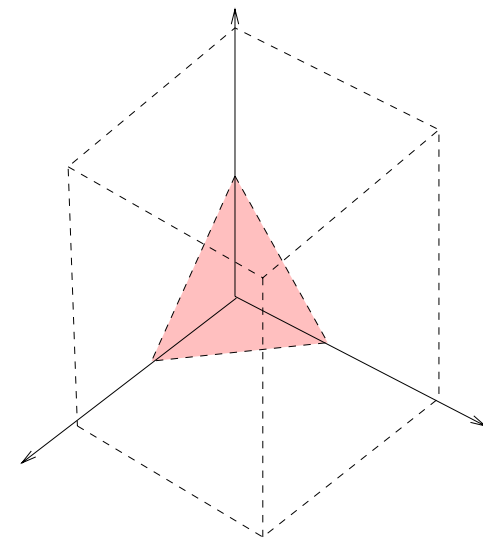
$$\pi(n) = \frac{1}{G} \prod_{i=1}^R \frac{\rho_i^{n_i}}{n_i!}, \quad n \in S$$

Accept/Reject:

- Repeat
  - Generate  $(m_1, \dots, m_R)$  independent,  $m_i \sim \text{Poisson}(\rho_i)$
- until  $m \in S$
- Set  $X_k(i) = m_i$

$X_k$  are i.i.d.  $\sim \pi$ .

$\rho_i$	$R = 7$	$R = 19$	$R = 37$
12	0.05	0.19	0.5
15	1.10	11.00	167.0
18	17.00	16.66	$5.3 \times 10^8$

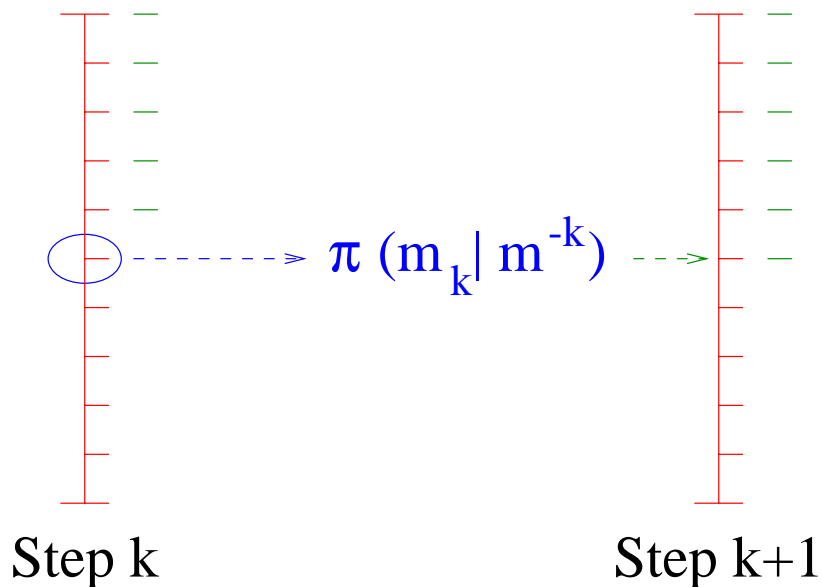


**Problem:**

$$E(\text{iterations}) = 1/G$$

## 5. The Gibbs sampler

**Periodic Gibbs sampler:** change one component of the vector at a time, using the **conditional distribution** to generate it (one-dimensional truncated Poisson).



After  $R$  steps the vector is updated in all components,

$$\pi(m|\zeta_{k+1}) = \frac{P_j(m)}{P_j(\Lambda_j(\zeta_k))},$$

$$\Lambda_j(\zeta_k) = \Lambda - \max_{i:j \in C_i} \sum_{c \in C_i} \zeta_k(c) \mathbf{1}_{\{j \neq c\}}.$$

- all samples are accepted: truncates to the feasible region
- one component at a time: no need to evaluate  $G$

**Remark:**  $\{\zeta_{kR}\}_{k=0}^{\infty}$  is a Markov chain.

## 5.2 Markov chain Monte Carlo: Gibbs Sampler

**Idea:** When direct generation of  $X \sim \pi$  is difficult or impractical, simulate a Markov chain  $\{\zeta_k\}$  with stationary probabilities:

$$\pi(n) = \lim_{k \rightarrow \infty} \mathbf{P}\{\zeta_k = n\}, \quad n \in S$$

Sample average satisfies CLT.

$$\zeta_{k+1}(i) \begin{cases} = \zeta_k(i) & \text{if } i \neq \sigma_k = k \bmod(R) + 1, \\ \sim \pi(\cdot | \zeta_{k+1}^{-\sigma_k}) & \text{otherwise.} \end{cases},$$

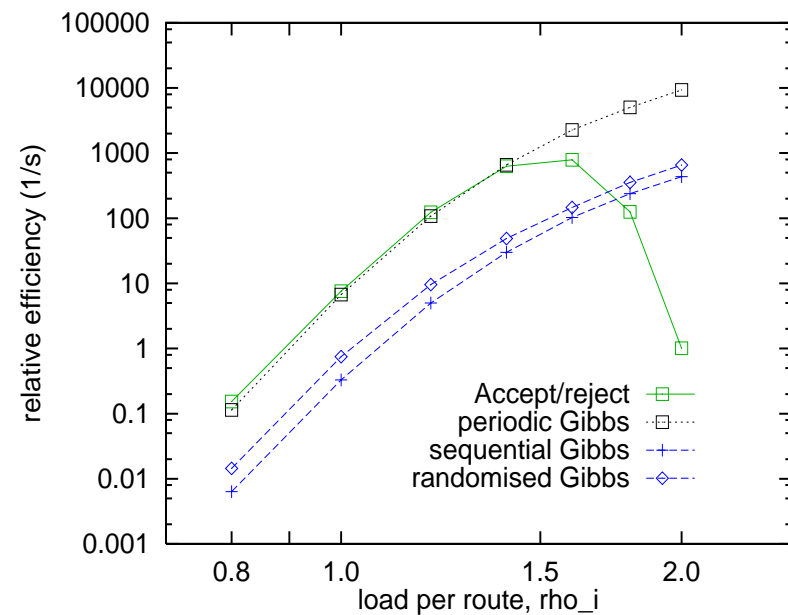
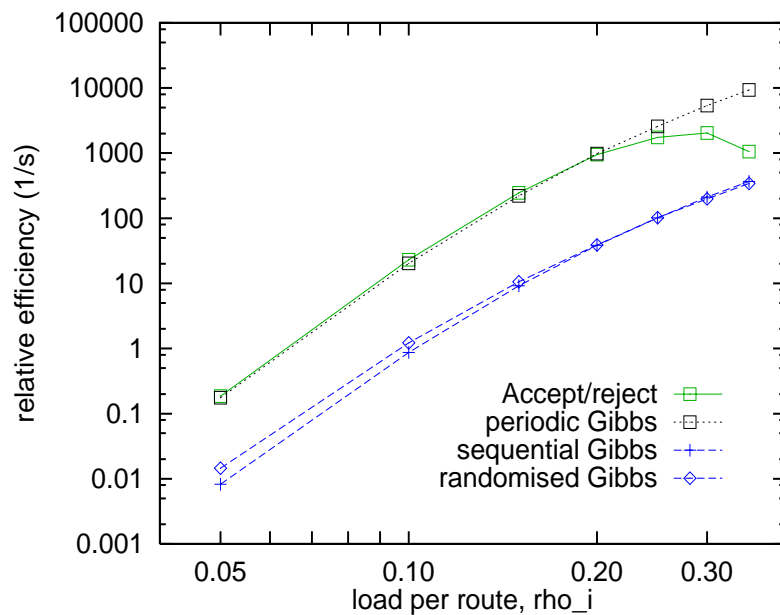
$$Y_k^{(r)} = \sum_{i=1}^R \left( \frac{\lambda_i}{\lambda} \right) \mathbf{1}_{\{\zeta_k^{(r)} \in \mathcal{B}_i\}} : \text{periodic } 1 \leq r \leq R,$$

$$Y_k = \sum_{i=1}^R \left( \frac{\lambda_i}{\lambda} \right) \mathbf{1}_{\{\zeta_k \in \mathcal{B}_i\}} : \text{sequential.}$$

**Result:** The periodic Gibbs sample has stationary probability  $\pi(\cdot)$ . The sequential Gibbs sampler is strongly consistent: at **each** step, calculate  $Y_k$  instead of only every  $R(\approx 300)$  steps.

## 5.2 Markov chain Monte Carlo: Gibbs Sampler

Efficiency against offered load for **direct simulation**, and **random**, **sequential** and **periodic** Gibbs samplers applied to a  $5 \times 5$  mesh-torus.



... except for **very high loads** Gibbs is not performing much better.

### 5.3 Filtered Gibbs sampler – Reduction of CPU time

Sequential Gibbs:  $\{\zeta_k\} \in \mathcal{S}$  a Markov chain. At each step use

$$Y_k = \sum_{i=1}^R \left( \frac{\lambda_i}{\lambda} \right) \mathbf{1}_{\{\zeta_k \in \mathcal{B}_i\}}$$

Many terms which are highly correlated between samples: component  $\sigma(k) = k \bmod R + 1$  is the only one that changes,  $\mathbf{1}_{\{\zeta_k \in \mathcal{B}_i\}} \in \{0, 1\}$  will be identical for many routes  $i$  from  $k - 1$  to  $k$ . For any  $i \in \{1, \dots, R\}$ :

$$Y_{k,i} = \mathbf{1}_{\{\zeta_{k+1} \in \mathcal{B}_i\}} \Rightarrow \mathbf{E}[Y_{k,i}] = B_i,$$

$$\lim_{N \rightarrow \infty} \frac{1}{N(i)} \sum_{k=1}^N Y_{k,i} \mathbf{1}_{\{\sigma_k = i\}} \rightarrow B_i$$

$R \approx 300$ , many computations are wasted.

Dedicate each iteration  $k$  to only *one component*  $i = \sigma_k \dots ?$

### 5.3 Filtered Gibbs sampler – Reduction of CPU time

Sequential Gibbs:  $\{\zeta_k\} \in \mathcal{S}$  a Markov chain. At each step use

$$Y_k = \sum_{i=1}^R \left( \frac{\lambda_i}{\lambda} \right) \mathbf{1}_{\{\zeta_k \in \mathcal{B}_i\}}$$

Dedicate each iteration  $k$  to only *one component*  $i = \sigma_k \dots?$   
 Greater effects on routes *sharing more cliques*.

Only determine blocking on updated route

$$Y_k = R \left( \frac{\lambda_{\sigma_k}}{\lambda} \right) \mathbf{1}_{\{\zeta_k \in \mathcal{B}_{\sigma_k}\}}, \quad \sigma_k = k \bmod R + 1.$$

**Result:** The sample average  $\hat{Y}(N)$  is strongly consistent for  $B$  and converges in the order  $\mathcal{O}(N^{-1/2})$ .

### 5.3 Filtered Gibbs sampler – Reduction of Variance

$$Y_k = R \left( \frac{\lambda_i}{\lambda} \right) \mathbf{1}_{\{\zeta_k \in \mathcal{B}_{\sigma_k}\}} \quad \sigma_k = k \bmod R + 1$$

Usual conditioning argument:  $\text{Var}(Y) = \text{Var}(\mathbf{E}(Y|X)) + \mathbf{E}(\text{Var}(Y|X))$ ,  
 $Z = \mathbf{E}(Y|X)$  satisfies  $\mathbf{E}(Z) = B$  and  $\text{Var}(Z) \leq \text{Var}(Y)$ .

$$\hat{Y}'(N) = \frac{1}{N} \sum_{k=1}^N \mathbf{E}[Y_k | \zeta_{k-1}]$$

conditions on **natural filtration** of the process (hence “filtered”).



### 5.3 Filtered Gibbs sampler – Reduction of Variance

$$Y_k = R \left( \frac{\lambda_i}{\lambda} \right) \mathbf{1}_{\{\zeta_k \in \mathcal{B}_{\sigma_k}\}} \quad \sigma_k = k \bmod R + 1$$

Usual conditioning argument:  $\text{Var}(Y) = \text{Var}(\mathbf{E}(Y|X)) + \mathbf{E}(\text{Var}(Y|X))$ ,  
 $Z = \mathbf{E}(Y|X)$  satisfies  $\mathbf{E}(Z) = B$  and  $\text{Var}(Z) \leq \text{Var}(Y)$ .

$$\hat{Y}'(N) = \frac{1}{N} \sum_{k=1}^N \mathbf{E}[Y_k | \zeta_{k-1}]$$

conditions on **natural filtration** of the process (hence “filtered”).

Instead of using  $\mathbf{1}_{\{\zeta_k \in \mathcal{B}_{\sigma}\}}$ , estimate the blocking using the conditional probability:

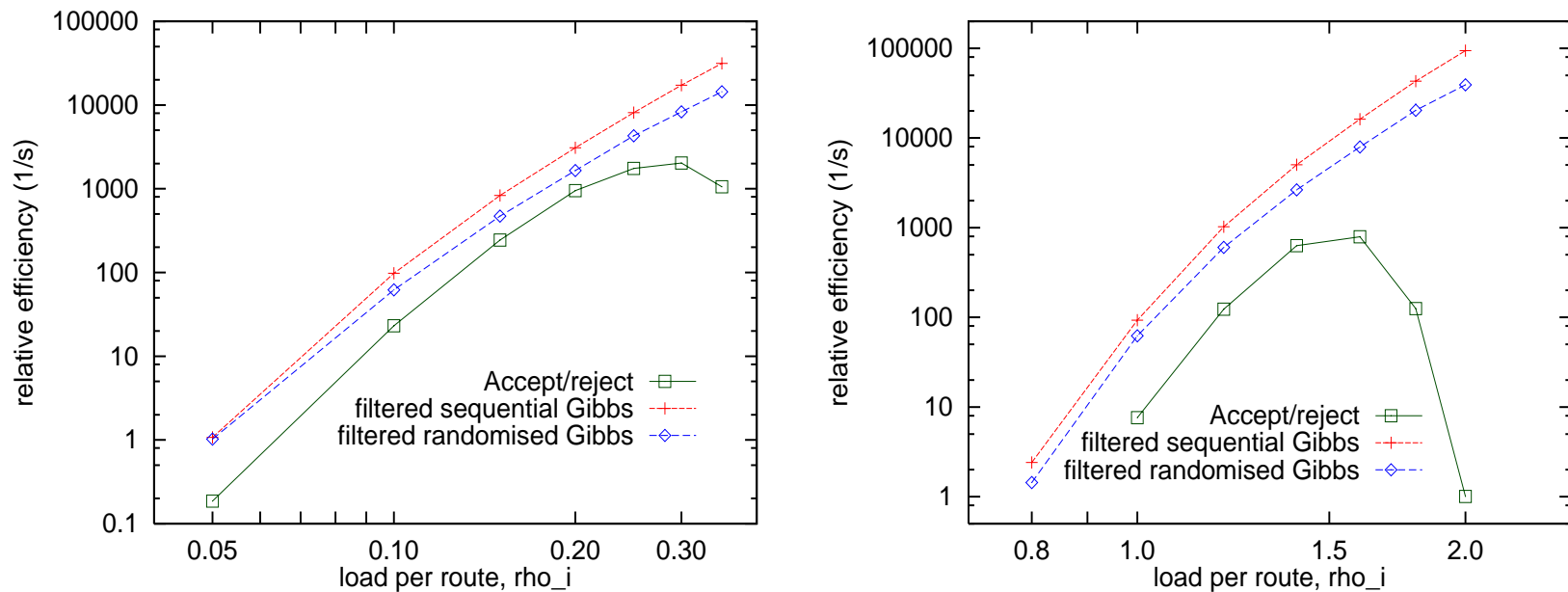
$$Y_k = R \left( \frac{\lambda_i}{\lambda} \right) \mathbf{P}(\zeta_k \in \mathcal{B}_{\sigma_k} | \zeta_{k-1})$$

**Result:** The Filtered Gibbs sampler is strongly consistent for  $B$  and converges in the order  $\mathcal{O}(N^{-1/2})$ .

$$\mathbf{P}[X_{k+1} \in \mathcal{B}_j | \zeta_k] = \frac{P_j(\Lambda_j(\zeta_k)) - P_j(\Lambda_j(\zeta_k) - 1)}{P_j(\Lambda_i(\zeta_k))}$$

### 5.3 Markov chain Monte Carlo: Gibbs Sampler

Efficiency against offered load for *direct simulation*, *filtered randomised* and *filtered sequential* Gibbs samplers applied to a  $5 \times 5$  mesh-torus.



Realistic network sizes and loads:  $\Lambda = 12$  at  $B = 0.12$ , the gain factor is 93,931: if our method requires **1 minute**, the usual A/R would need **65.2 days** to complete the simulation.

As network size grows it overcomes the curse of dimensionality.

## On-going and future work

- Use of stratification to choose  $\sigma_k$  in a judicious way, for variance reduction.
- Application of Filtered Gibbs sampler for more complex networks
  - partial conversion, asymmetric,
  - use efficient clique determination algorithms: graph theory.
- Application of similar techniques to realistic route assignment.
- Relative efficiency  $\downarrow$  as  $B \rightarrow 0$ : “rare events”
  - Use of fast simulation with importance sampling: direct simulation of the process.
  - Can we apply fast simulation to Gibbs sampler directly?