Analysis of coupled queues

Eline De Cuypere
Koen De Turck and Dieter Fiems

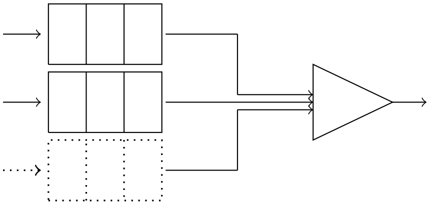
November 8, 2012

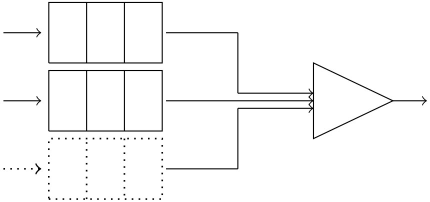

LEGO
Kingdoms
7-14
7946

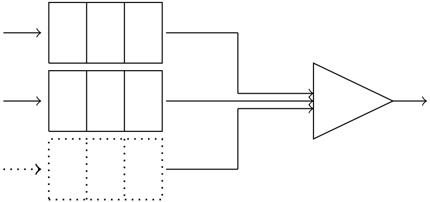

37

Coupled queues

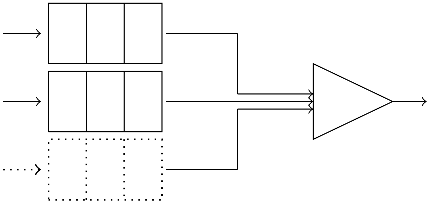
service is only possible if none of the queues are empty.



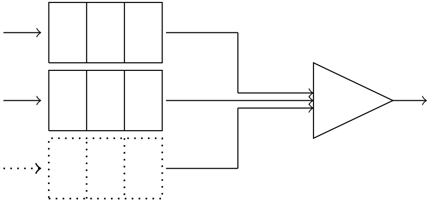




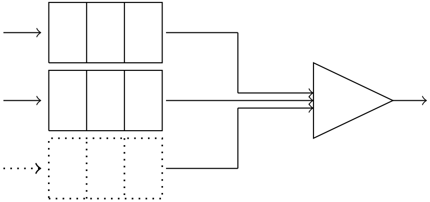
- slow convergence of simulation results



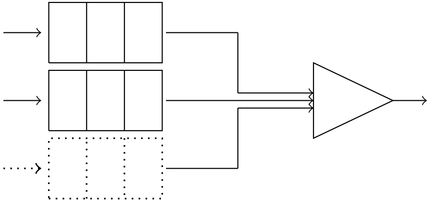
- slow convergence of simulation results
- null-recurrence



- slow convergence of simulation results
- null-recurrence
- state-space explosion



- slow convergence of simulation results
- null-recurrence
- state-space explosion
- rather complicated matrix structure



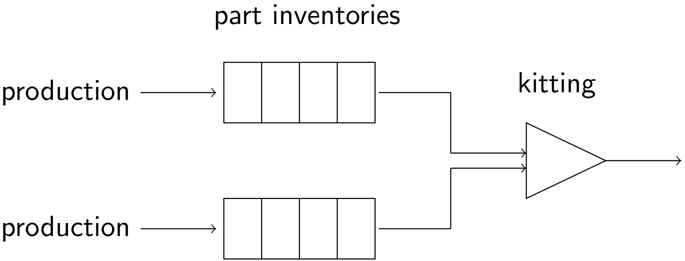
- slow convergence of simulation results
- null-recurrence
- state-space explosion
- rather complicated matrix structure

Table of contents

Applications

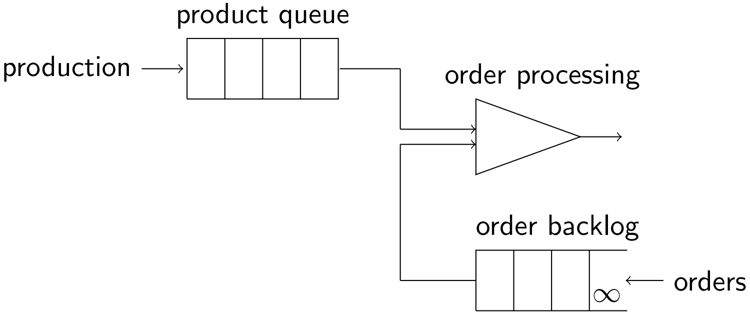
- Kitting process
- Decoupling buffer
- Energy harvesting process

Kitting process


part inventories
production
production
kitting

Decoupling inventory in a hybrid push-pull system

Raw materials are **pushed** into the semi-finished product inventory while customers **pulls** products by placing orders


product queue
production
order processing
order backlog
∞
orders

Energy harvesting process

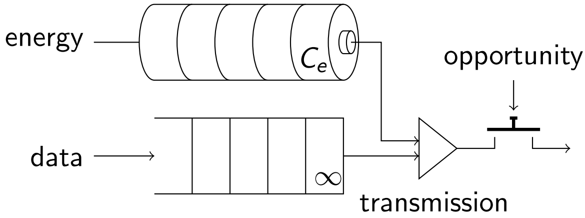
Energy harvesting is the process by which

- ambient energy is converted into electrical energy and

Energy harvesting process

Energy harvesting is the process by which

- ambient energy is converted into electrical energy and
- this energy is stored in small autonomous devices called **sensor nodes**.


energy
C_e
opportunity
data
∞
transmission

Numerical techniques

- **Sparse matrix techniques**
- Matrix-analytic methods
- Taylor series expansion approach

Sparse matrix techniques

Sparse method

- **sparse matrix**: matrix with most of its elements equal to zero

Sparse matrix techniques

Sparse method

- **sparse matrix**: matrix with most of its elements equal to zero
- generator matrix of paired queues where e.g. $C_1 = C_2 = 100$:
  - standard: 40804^2 elements
  - sparse: $3x40804$ elements

Sparse matrix techniques

Sparse method

- **sparse matrix**: matrix with most of its elements equal to zero
- generator matrix of paired queues where e.g. $C_1 = C_2 = 100$:
  - standard: 40804^2 elements
  - sparse: $3x40804$ elements
- less memory consumption and processing time than for standard matrices

Sparse matrix techniques

Sparse method

- **sparse matrix**: matrix with most of its elements equal to zero
- generator matrix of paired queues where e.g. $C_1 = C_2 = 100$:
  - standard: 40804^2 elements
  - sparse: $3x40804$ elements
- less memory consumption and processing time than for standard matrices

General Minimal Residual Method (GMRES)

- iterative method to solve sparse matrix equations
- fast and sufficiently accurate
- however, size of the state space is strictly limited

Numerical techniques

- Sparse matrix techniques
- **Matrix-analytic methods**
- Taylor series expansion approach

QBD Processes

We define a Markov process with state
$S = \{(n, i) : n > 0, 1 \leqslant i \leqslant m\}$

QBD Processes

We define a Markov process with state
$S = \{(n, i) : n > 0, 1 \leqslant i \leqslant m\}$

and with infinitesimal generator

$$Q = \begin{bmatrix} B_0 & A_0 & 0 & 0 & \cdots \\ A_2 & A_1 & A_0 & 0 & \cdots \\ 0 & A_2 & A_1 & A_0 & \cdots \\ 0 & 0 & A_2 & A_1 & \cdots \\ \vdots & \vdots & \vdots & \ddots & \end{bmatrix}$$

QBD Processes

We define a Markov process with state
$S = \{(n, i) : n > 0, 1 \leqslant i \leqslant m\}$

and with infinitesimal generator

$$Q = \begin{bmatrix} B_0 & A_0 & 0 & 0 & \cdots \\ A_2 & A_1 & A_0 & 0 & \cdots \\ 0 & A_2 & A_1 & A_0 & \cdots \\ 0 & 0 & A_2 & A_1 & \cdots \\ \vdots & \vdots & \vdots & \ddots & \end{bmatrix}$$

The one-step transitions are restricted to states in the same level $(n, *)$ or in two adjacent levels $(n + 1, *)$ or $(n - 1, *)$.

QBD Processes

We define a Markov process with state $S = \{(n, i) : n > 0, 1 \leqslant i \leqslant m\}$

and with infinitesimal generator

$$Q = \begin{bmatrix} B_0 & A_0 & 0 & 0 & \cdots \\ A_2 & A_1 & A_0 & 0 & \cdots \\ 0 & A_2 & A_1 & A_0 & \cdots \\ 0 & 0 & A_2 & A_1 & \cdots \\ \vdots & \vdots & \vdots & \ddots & \end{bmatrix}$$

The one-step transitions are restricted to states in the same level $(n, *)$ or in two adjacent levels $(n+1, *)$ or $(n-1, *)$.

The following equation determine the stationary distribution:

$$\pi_{\mathbf{n}} = \pi_{\mathbf{0}} \mathbf{R}^{\mathbf{n}}$$

QBD Processes

We define a Markov process with state
$S = \{(n, i) : n > 0, 1 \leqslant i \leqslant m\}$

and with infinitesimal generator

$$Q = \begin{bmatrix} B_0 & A_0 & 0 & 0 & \cdots \\ A_2 & A_1 & A_0 & 0 & \cdots \\ 0 & A_2 & A_1 & A_0 & \cdots \\ 0 & 0 & A_2 & A_1 & \cdots \\ \vdots & \vdots & \vdots & \ddots & \end{bmatrix}$$

The one-step transitions are restricted to states in the same level $(n, *)$ or in two adjacent levels $(n + 1, *)$ or $(n - 1, *)$.

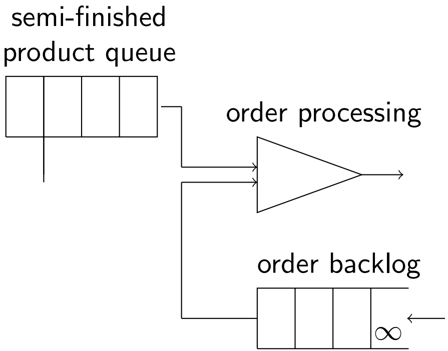
The following equation determine the stationary distribution:

$$\pi_{\mathbf{n}} = \pi_{\mathbf{0}} \mathbf{R}^{\mathbf{n}}$$

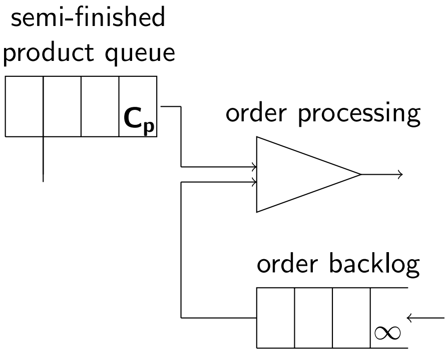
with the *rate matrix* **R**, solution of the equation :

$$\mathbf{R} = \mathbf{A_0} + \mathbf{R}\mathbf{A_1} + \mathbf{R}^2\mathbf{A_2}$$

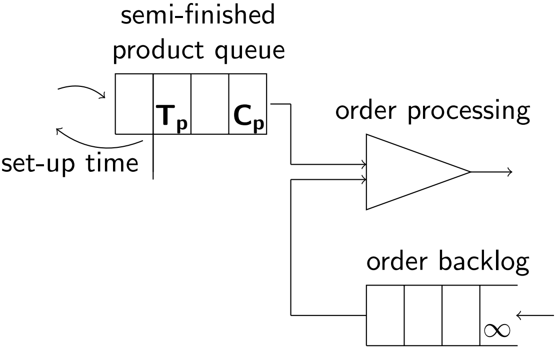
decoupling inventory model



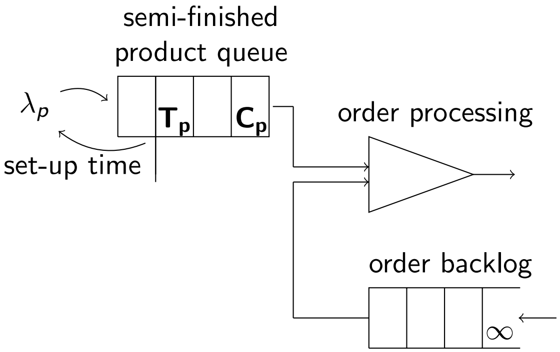
decoupling inventory model



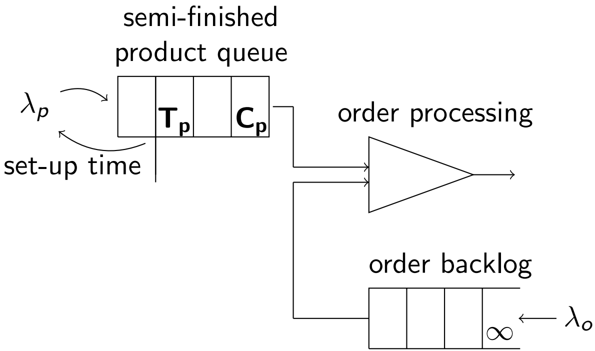
decoupling inventory model



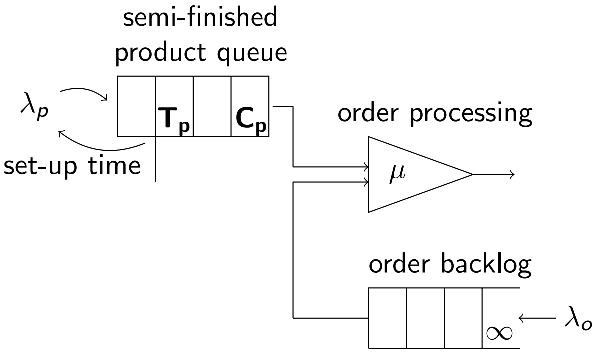
decoupling inventory model



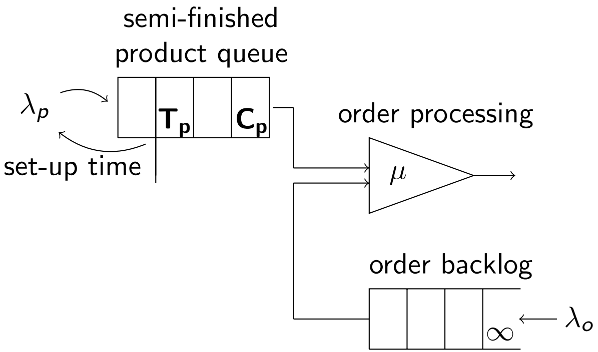
decoupling inventory model



decoupling inventory model

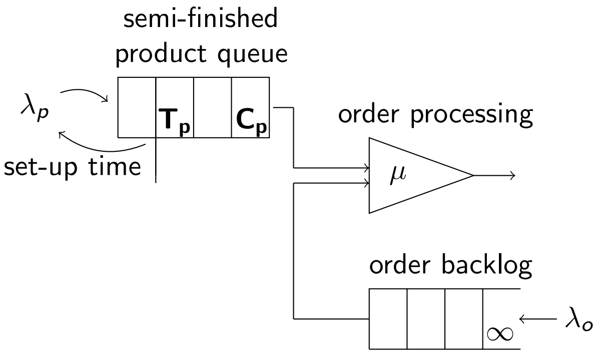


decoupling inventory model



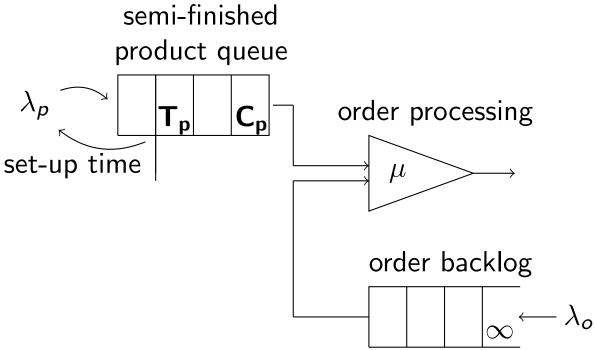
- 3-dimensional modulating Markov chain with state (n, m, i)

decoupling inventory model



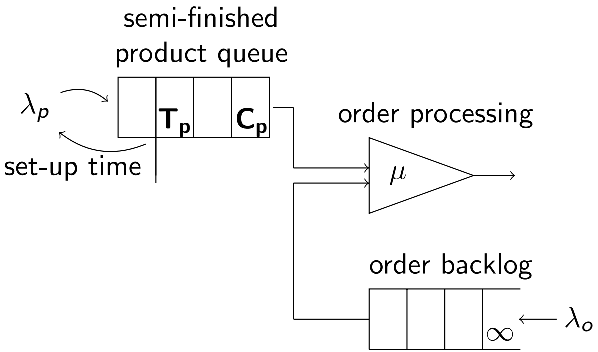
- 3-dimensional modulating Markov chain with state (n, m, i)
  - n = number of backlogged orders

decoupling inventory model



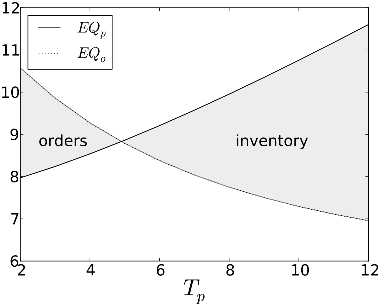
- 3-dimensional modulating Markov chain with state (n, m, i)
  - n = number of backlogged orders
  - m = number of semi-finished products

decoupling inventory model



- 3-dimensional modulating Markov chain with state (n, m, i)
  - n = number of backlogged orders
  - m = number of semi-finished products
  - i = state of the modulating chain

Numerical results



- capacity: $C_p = 20$
- product arrival rate: $\lambda_p = 1$
- order arrival rate: $\lambda_o = 0.85$
- order processing rate: $\mu = 1$
- no set-up time

block matrix structure for 3 finite capacity queues

$C_i = 2$ for queue $i = \{1, 2, 3\}$

generator matrix with a tridiagonal block structure

$$
\left(\begin{array}{ccc|ccc|ccc||ccc|ccc|ccc||ccc|ccc|ccc}
0 & \lambda_3 & 0 & \lambda_2 & 0 & 0 & 0 & 0 & 0 & \lambda_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \lambda_3 & 0 & \lambda_2 & 0 & 0 & 0 & 0 & 0 & \lambda_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \lambda_2 & 0 & 0 & 0 & 0 & 0 & \lambda_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & \lambda_3 & 0 & \lambda_2 & 0 & 0 & 0 & 0 & 0 & \lambda_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \lambda_3 & 0 & \lambda_2 & 0 & 0 & 0 & 0 & 0 & \lambda_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_2 & 0 & 0 & 0 & 0 & 0 & \lambda_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline\hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_3 & 0 & \lambda_2 & 0 & 0 & 0 & 0 & 0 & \lambda_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_3 & 0 & \lambda_2 & 0 & 0 & 0 & 0 & 0 & \lambda_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_2 & 0 & 0 & 0 & 0 & 0 & \lambda_1 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_3 & 0 & \lambda_2 & 0 & 0 & 0 & 0 & 0 & \lambda_1 & 0 & 0 & 0 & 0 & 0 \\
\mu & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_3 & 0 & \lambda_2 & 0 & 0 & 0 & 0 & 0 & \lambda_1 & 0 & 0 & 0 & 0 \\
0 & \mu & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_2 & 0 & 0 & 0 & 0 & 0 & \lambda_1 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_1 \\
0 & 0 & 0 & \mu & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_1 & 0 \\
0 & 0 & 0 & 0 & \mu & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_1 \\
\hline\hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_3 & 0 & \lambda_2 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_3 & 0 & \lambda_2 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_2 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_3 & 0 & \lambda_2 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mu & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_3 & 0 & \lambda_2 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mu & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_2 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_3 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mu & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_3 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mu & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}\right)
$$

1

Numerical techniques

- Sparse matrix techniques
- Matrix-analytic methods
- **Taylor series expansion approach**

Taylor series expansion approach

Consider a family of (continuous-time) Markov processes $\{X_\epsilon(t)\}$ over a finite state space $\mathcal{X}$ of size M with generator matrix

$$Q_\epsilon = Q^{(0)} + \epsilon Q^{(1)},$$

π_ϵ denotes the corresponding stationary distribution

$$\pi_\epsilon Q_\epsilon = 0$$

Taylor series expansion approach

Consider a family of (continuous-time) Markov processes $\{X_\epsilon(t)\}$ over a finite state space $\mathcal{X}$ of size M with generator matrix

$$Q_\epsilon = Q^{(0)} + \epsilon Q^{(1)},$$

π_ϵ denotes the corresponding stationary distribution

$$\pi_\epsilon Q_\epsilon = 0$$

Numerical computation of the steady-state vector has an asymptotic time complexity of $O(M^3)$

Models that suffer from state-space explosion stay generally out of reach of a numerical analysis

Taylor series expansion around $\epsilon = 0$

Vector π_ϵ is required to be analytic in a neighbourhood of $\epsilon = 0$

Taylor series expansion around $\epsilon = 0$

Vector π_ϵ is required to be analytic in a neighbourhood of $\epsilon = 0$

- we only consider finite state space
- entries of the generator matrix depend analytically on the parameter
- only one recurrent class for $\epsilon = 0$

"Regular perturbation"

From now on: assume $Q_0 = Q^{(0)}$ is a generator matrix with one recurrent class

Recursive solution of the series expansion

$$\pi^{(0)} Q^{(0)} = 0$$

$$\pi^{(n+1)} Q^{(0)} = -\pi^{(n)} Q^{(1)}$$

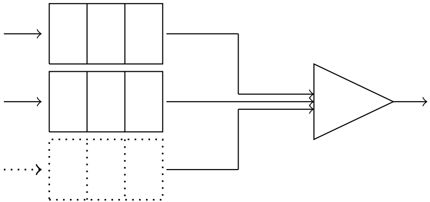
We have to solve a linear system of equations for each term $\pi^{(i)}$ in the expansion

- Impose the extra condition that $Q^{(0)}$ for some ordering of the state space be triangular

- Impose the extra condition that $Q^{(0)}$ for some ordering of the state space be triangular
  - Solution by backward substitutions $\rightarrow O(M^2)$

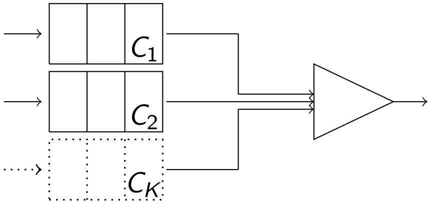
- Impose the extra condition that $Q^{(0)}$ for some ordering of the state space be triangular
  - Solution by backward substitutions $\rightarrow O(M^2)$
  - In practice $Q^{(0)}$ and $Q^{(1)}$ often sparse $\rightarrow O(M)$

Kitting process with K parts



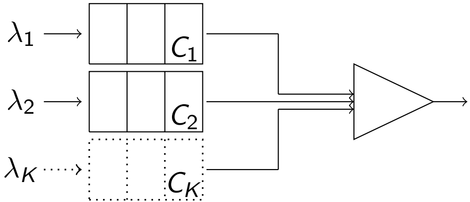
- system of K queues, queue i having finite capacity C_i
- Poisson arrivals in each of the queues at rate λ_i
- exponentially distributed service rate μ
- Lexicographical order
- Expansion around $\epsilon = \mu = 0$, all queues are full for $\epsilon = 0$

Kitting process with K parts



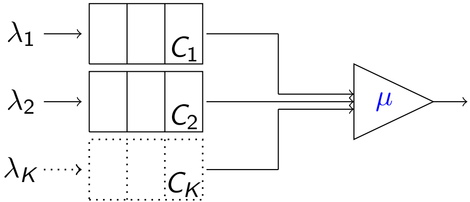
- system of K queues, queue i having finite capacity C_i
- Poisson arrivals in each of the queues at rate λ_i
- exponentially distributed service rate μ
- Lexicographical order
- Expansion around $\epsilon = \mu = 0$, all queues are full for $\epsilon = 0$

Kitting process with K parts



- system of K queues, queue i having finite capacity C_i
- Poisson arrivals in each of the queues at rate λ_i
- exponentially distributed service rate μ
- Lexicographical order
- Expansion around $\epsilon = \mu = 0$, all queues are full for $\epsilon = 0$

Kitting process with K parts



- system of K queues, queue i having finite capacity C_i
- Poisson arrivals in each of the queues at rate λ_i
- exponentially distributed service rate μ
- Lexicographical order
- Expansion around $\epsilon = \mu = 0$, all queues are full for $\epsilon = 0$

1. Balance equation

$$\pi(i_1, i_2, \ldots, i_K) \left(\mu \prod_{\ell=1}^{K} \mathbf{1}_{\{i_\ell > 0\}} + \sum_{\ell=1}^{K} \mathbf{1}_{\{i_\ell < C_\ell\}} \lambda_\ell \right) =$$

$$\pi(i_1 + 1, i_2 + 1, \ldots, i_K + 1)\, \mu \prod_{\ell=1}^{K} \mathbf{1}_{\{i_\ell < C_\ell\}}$$

$$+ \sum_{\ell=1}^{K} \pi(i_1, \ldots, i_{\ell-1}, i_\ell - 1, i_{\ell+1}, \ldots, i_K) \lambda_\ell \mathbf{1}_{\{i_\ell > 0\}}$$

2. Expansion

$$\pi(\mathbf{i}) = \sum_{n=0}^{\infty} \pi_n(\mathbf{i})\mu^n$$

3. Substitution

$$\sum_{n=0}^{\infty} \pi_n(i_1, i_2, \ldots, i_K)\mu^n \left(\mu \prod_{\ell=1}^{K} \mathbf{1}_{\{i_\ell > 0\}} + \sum_{\ell=1}^{K} \mathbf{1}_{\{i_\ell < C_\ell\}} \lambda_\ell \right) =$$

$$\sum_{n=0}^{\infty} \pi_n(i_1 + 1, i_2 + 1, \ldots, i_K + 1)\, \mu^{n+1} \prod_{\ell=1}^{K} \mathbf{1}_{\{i_\ell < C_\ell\}}$$

$$+ \sum_{n=0}^{\infty} \sum_{\ell=1}^{K} \pi_n(i_1, \ldots, i_{\ell-1}, i_\ell - 1, i_{\ell+1}, \ldots, i_K)\lambda_\ell \mu^n \mathbf{1}_{\{i_\ell > 0\}}$$

4. Compare terms in μ^n

$$\pi_n(i_1, i_2, \ldots, i_K) = \frac{1}{\sum_{\ell=1}^{K} \mathbf{1}_{\{i_\ell < C_\ell\}} \lambda_\ell} \times$$

$$\Big(\mathbf{1}_{\{n>0\}} \pi_{n-1}(i_1 + 1, i_2 + 1, \ldots, i_K + 1) \prod_{\ell=1}^{K} \mathbf{1}_{\{i_\ell < C_\ell\}}$$

$$+ \sum_{\ell=1}^{K} \pi_n(i_1, \ldots, i_{\ell-1}, i_\ell - 1, i_{\ell+1}, \ldots, i_K) \lambda_\ell \mathbf{1}_{\{i_\ell > 0\}}$$

$$- \mathbf{1}_{\{n>0\}} \pi_{n-1}(i_1, i_2, \ldots, i_K) \prod_{\ell=1}^{K} \mathbf{1}_{\{i_\ell > 0\}} \Big)$$

5. Normalisation condition

$$\pi_0([C_1, C_2, \ldots, C_K]) = 1$$

$$\pi_n([C_1, C_2, \ldots, C_K]) = -\sum_{i \in C^*} \pi_n(\mathbf{i})$$

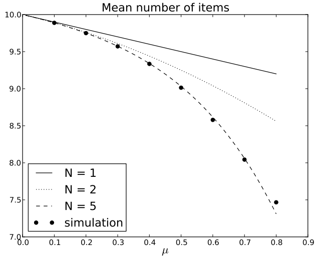
with $\mathcal{C}^* = \mathcal{C} \setminus \{[C_1, C_2, \ldots, C_K]\}$

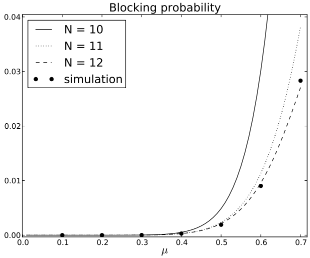
What's the speed?

C \ N	5	10	20	50	100
10	0.340	0.376	0.415	0.534	0.735
20	0.796	1.237	2.144	4.678	8.960
30	3.783	6.842	12.984	32.856	64.660
40	14.640	30.422	53.202	128.375	257.236

Table: Computation time in seconds for 5 queues

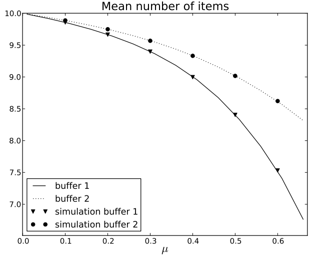
Numerical results

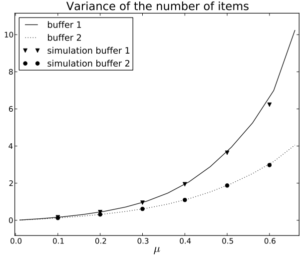




5 queues, $\lambda = 1$ and $C = 10$ for all queues

Numerical results





5 queues, $\lambda_1 = 0.8$ and $\lambda_{2-5} = 1$, $C = 10$

Conclusion

Conclusion

- Numerical analysis of coupled queues in a Markovian setting
  - Sparse-matrix techniques
  - Matrix-analytic methods
  - Taylor series expansion approach

Conclusion

Conclusion

- Numerical analysis of coupled queues in a Markovian setting
  - Sparse-matrix techniques
  - Matrix-analytic methods
  - Taylor series expansion approach

Future work

- Compare the analysis methodology

Conclusion

Conclusion

- Numerical analysis of coupled queues in a Markovian setting
  - Sparse-matrix techniques
  - Matrix-analytic methods
  - Taylor series expansion approach

Future work

- Compare the analysis methodology
- Investigate the matrix-analytic methods to cope with the tridiagonal block matrix structure of the generator matrix for the multidimensional case.

Conclusion

Conclusion

- Numerical analysis of coupled queues in a Markovian setting
  - Sparse-matrix techniques
  - Matrix-analytic methods
  - Taylor series expansion approach

Future work

- Compare the analysis methodology
- Investigate the matrix-analytic methods to cope with the tridiagonal block matrix structure of the generator matrix for the multidimensional case.
- Investigate other methods: mean field analysis, stochastic fluid model,. . .

Questions?