E.6 Approximate Analysis of Non-product-Form Queueing Networks

- Non exponentially distributed service times
- Priorities (time independent and time dependent)
- Different service times of different classes at Type-1 nodes -
- Finite queues (Queueing networks with blocking))
- Parallel processing, synchronisation
- Fork-Join-Systems
- Queueing networks with heterogeneous M/M/m- oder G/G/m- nodes
- Simultaneous resource possession

1 Solution Methods

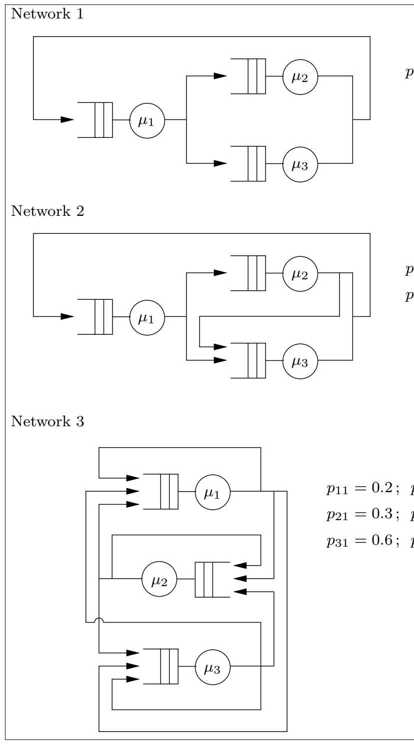
- Approximation of a non-product-form queueing network by a product-form queueing network
- Markovanalysis (expensive) → MOSEL
- Simulation (very expensive) → PEPSY
- Approximation methods → PEPSY
  - ◆ Extension of the product-form methods
  - ◆ Iterative use of product-form methods

2 Closed Non-Product-Form Queueing Networks

- At least one M/G/m-FCFS-node (-/G/m-FCFS-node)
  - Diffusionsapproximation
  - Maximum entropy method
  - Method of Marie
  - Summation method
  - Bottleneck Approximation
  - Robustness

- Robustness:
  - ➤ -/G/1-FCFS- and -/G/m-FCFS-nodes are replaced by M/M/1-FCFS- or M/M/m-FCFS-nodes
  - ➤ Only little influence of the coefficient of variation of the non-product-form nodes (-/G/1 und -/G/m) in case of **closed** networks (not valid for open networks !!).
  - ➤ This property is called "**Robustness**" .
  - ➤ The accuracy of the robustness is sufficient for many applications.

◆ Example



Network 1: $p_{12} = 0.6\,;\ p_{13} = 0.4$

Network 2: $p_{12} = 0.6\,;\ p_{13} = 0.4$

$p_{21} = 0.3\,;\ p_{23} = 0.7$

Network 3: $p_{11} = 0.2\,;\ p_{12} = 0.4\,;\ p_{13} = 0.4\,;$

$p_{21} = 0.3\,;\ p_{22} = 0.2\,;\ p_{23} = 0.5\,;$

$p_{31} = 0.6\,;\ p_{32} = 0.2\,;\ p_{31} = 0.2\,;$

➤ Coefficient of variation:

Combinations	Node 1 $c_{B_1}^2$	Node 2 $c_{B_2}^2$	Node 3 $c_{B_3}^2$
a	1.0	1.0	1.0
b	0.2	0.4	0.8
c	4.0	1.0	0.2
d	2.0	4.0	8.0

➤ Service rates:

	μ_1	μ_2	μ_3
Balanced network	0.5	0.333	0.666
Network with bottleneck	0.5	1.0	2.0

➤ Throughput of the "Balanced Network"

Squared Coefficient of Variations	Network 1		Network 2		Network 3	
	$K = 5$	$K = 10$	$K = 5$	$K = 10$	$K = 5$	$K = 10$
a	0.43	0.47	0.41	0.47	0.37	0.42
b	0.47	0.50	0.45	0.49	0.40	0.44
c	0.39	0.44	0.48	0.47	0.36	0.41
d	0.37	0.42	0.33	0.39	0.29	0.34

➤ Throughput of the "Network with Bottleneck":

Squared Coefficient of Variations	Network 1		Network 2		Network 3	
	$K = 5$	$K = 10$	$K = 5$	$K = 10$	$K = 5$	$K = 10$
a	0.50	0.50	0.51	0.50	0.50	0.52
b	0.50	0.50	0.52	0.50	0.50	0.52
c	0.49	0.50	0.52	0.50	0.49	0.52
d	0.47	0.50	0.47	0.50	0.48	0.50

➤ Throughput as a function of the number of jobs in the network:

K	3	4	5	10	20	50
a	0.842	0.907	0.940	0.996	1.00	1.00
b	–	0.970	0.991	1.00	1.00	1.00
c	–	0.856	0.894	0.972	0.998	1.00
d	0.716	0.766	0.805	0.917	0.984	1.00

3 Open Non-Product-Form Queueing Networks

- Methods:
  - ◆ Diffusion approximation
  - ◆ Maximum entropy method
  - ◆ Decomposition methods
    - ➤ Pujolle
    - ➤ Whitt
    - ➤ Gelenbe
    - ➤ Chylla
    - ➤ Kühn

- Decomposition methods
  - ➤ Interarrival times and service times are **arbitrarily** distributed and given by the first and second moment (Coefficient of variation).
  - ➤ Queueing discipline is FCFS, queue length is not limilted
  - ➤ Multiple jop classes (no class switching)
  - ➤ Multiple Server nodes (-/G/m) are possible.

◆ Algorithm (Example: method of Whitt):

- ***Step*** 1: Arrival rates and utilization of the nodes:

$$\lambda_i = \lambda_{0i} + \sum_{j=1}^{N} \lambda_j \cdot p_{ji}$$

$$\rho_i = \frac{\lambda_i}{m_i \cdot \mu_i}$$

- ***Step*** 2: Iterative computation of the coefficient of variation of inter arrival times of the individual nodes (see below).

- ***Step*** 3: Mean queue length and other performance measures

  ➤ M/M/m-FCFS:

$$\overline{Q}_{i\mathrm{M/M/m}} = \frac{\rho_i}{1-\rho_i} \cdot P_{mi}$$

Probability of waiting P_{mi} (= ρ_i für M/M/1)

  ➤ Allen-Cunneen formula for G/G/m-FCFS:

$$\overline{Q}_{i\mathrm{AC}} \approx \overline{Q}_{i\mathrm{M/M/m}} \cdot \frac{(c_{Ai}^2 + c_{Bi}^2)}{2}$$

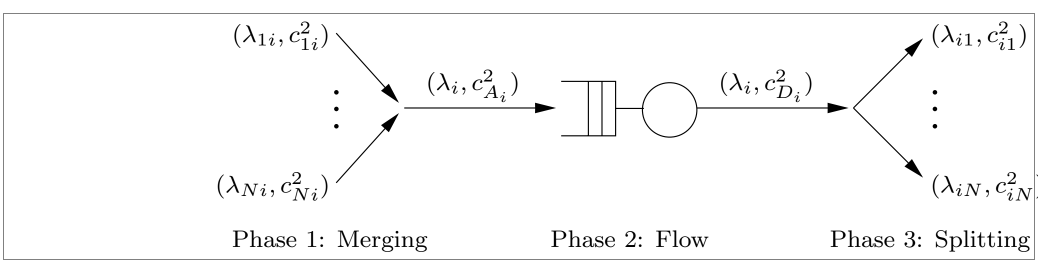
➤ Krämer/Langenbach-Belz formula for G/G/m-FCFS:

$$\overline{Q}_{i\mathrm{KLB}} \approx \overline{Q}_{i\mathrm{AC}} \cdot G_{\mathrm{KLB}}$$

with:

$$G_{\mathrm{KLB}} = \begin{cases} e^{\left(-\frac{2}{3} \cdot \frac{(1-\rho_i)}{P_{mi}} \cdot \frac{(1-c_{Ai}^2)^2}{(c_{Ai}^2 + c_{Bi}^2)}\right)}, & c_{Ai}^2 \leq 1\,, \\ e^{-\left((1-\rho_i) \cdot \frac{(c_{Ai}^2 - 1)}{(c_{Ai}^2 + c_{Bi}^2)}\right)}, & c_{Ai}^2 > 1 \end{cases}$$

• to ***Step*** 2: Iterative computation of the coefficient of variation of inter arrival times of the individual nodes (Whitt):



➤ Initialization: $c_{ij} = 1 \qquad i, j = 1, 2, \ldots, N$

➤ Merging: $i = 1, 2, \ldots, N$

$$c_{Ai}^2 = \frac{1}{\lambda_i} \cdot \left(\sum_{j=1}^{N} c_{ji}^2 \cdot \lambda_j \cdot p_{ji} + c_{0i}^2 \cdot \lambda_0 \cdot p_{0i} \right)$$

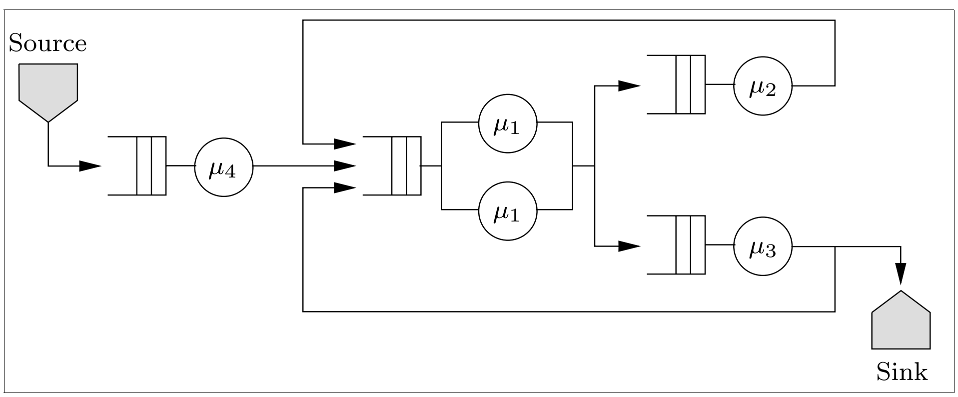
➤ Flow: $i = 1, 2, \ldots, N$

$$c_{Di}^2 = 1 + \frac{\rho_i^2 \cdot (c_{Bi}^2 - 1)}{\sqrt{m_i}} + (1 - \rho_i^2) \cdot (c_{Ai}^2 - 1)$$

➤ Splitting: $i, j = 1, 2, \ldots, N$

$$c_{ij}^2 = 1 + p_{ij} \cdot (c_{Di}^2 - 1)$$

◆ Example:



$$p_{12} = 0.5\,, \quad p_{13} = 0.5\,, \quad p_{31} = 0.6\,, \quad p_{21} = p_{41} = 1$$

$$\mu_1 = 12.5\,, \quad \mu_2 = 33.333 \quad \mu_3 = 16.666\,, \quad \mu_4 = 20$$

$$c_{B_1}^2 = 2.0\,, \quad c_{B_2}^2 = 6.0\,, \quad c_{B_3}^2 = 0.5\,, \quad c_{B_4}^2 = 0.2\,, \quad c_{04}^2 = 4.0$$

- ***Step*** 1: Arrival rates and utilizations;

$$\lambda_1 = \underline{20}, \quad \lambda_2 = \underline{10}, \quad \lambda_3 = \underline{10}, \quad \lambda_4 = \underline{4}$$

$$\rho_1 = \underline{0.8}, \quad \rho_2 = \underline{0.3}, \quad \rho_3 = \underline{0.6}, \quad \rho_4 = \underline{0.2}$$

- ***Step*** 2: Coefficient of variation of inter arrival times:

  ➤ Initialization:

$$c_{12}^2 = c_{13}^2 = c_{21}^2 = c_{31}^2 = c_{41}^2 = 1$$

E.6 Approximate Analysis of Non-product-Form Queueing Networks

1. Iteration:

➤ Merging:

$$c_{A_1}^2 = \frac{1}{\lambda_1}(c_{21}^2\lambda_2 p_{21} + c_{31}^2\lambda_3 p_{31} + c_{41}\lambda_4 p_{41})$$
$$= \frac{1}{20}(1\cdot 10\cdot 1 + 1\cdot 10\cdot 0.6 + 1\cdot 4\cdot 1) = \underline{1}\,.$$

$$c_{A_2}^2 = \underline{1}, \quad c_{A_3}^2 = \underline{1}, \quad c_{A_4}^2 = \underline{4}$$

➤ Flow:

$$
\begin{aligned}
c_{D_1}^2 &= 1 + \frac{\rho_1^2(c_{B_1}^2 - 1)}{\sqrt{m_1}} + (1-\rho_1^2)(c_{A_1}^2 - 1) \\
&= 1 + \frac{0.64(2-1)}{\sqrt{2}} + (1-0.64)(1-1) \\
&= \underline{1.453}
\end{aligned}
$$

$$c_{D_2}^2 = \underline{1.45}\,, \quad c_{D_3}^2 = \underline{0.82}\,, \quad c_{D_4}^2 = \underline{3.848}\,.$$

➤ Splitting:

$$c_{12}^2 = 1 + p_{12} \cdot (c_{D_1}^2 - 1) = \underline{1.226}$$

$$c_{13}^2 = \underline{1.226}\,, \quad c_{21}^2 = \underline{1.450}\,, \quad c_{31}^2 = \underline{0.892}\,, \quad c_{41}^2 = \underline{3.848}\,.$$

2. Iteration:

➤ Merging:

$$c_{A_1}^2 = \underline{1.762}\,, \quad c_{A_2}^2 = \underline{1.226}\,, \quad c_{A_3}^2 = \underline{1.226}\,, \quad c_{A_4}^2 = \underline{4.0}$$

➤ Flow:

$$c_{D_1}^2 = \underline{1.727}\,, \quad c_{D_2}^2 = \underline{1.656}\,, \quad c_{D_3}^2 = \underline{0.965}\,, \quad c_{D_4}^2 = \underline{3.848}$$

➤ Splitting:

$$
\begin{aligned}
&c_{12}^2 = \underline{1.363}\,, \quad c_{13}^2 = \underline{1.363}\,, \quad c_{21}^2 = \underline{1.656} \\
&c_{31}^2 = \underline{0.979}\,, \quad c_{41}^2 = \underline{3.848}
\end{aligned}
$$

7 Iterationen:

Iteration	$c^2_{A_1}$	$c^2_{A_2}$	$c^2_{A_3}$	$c^2_{A_4}$
1	1.0	1.0	1.0	4.0
2	1.762	1.226	1.226	4.0
3	1.891	1.363	1.363	4.0
4	1.969	1.387	1.387	4.0
5	1.983	1.401	1.401	4.0
6	1.991	1.403	1.403	4.0
7	1.992	1.405	1.405	4.0

- ***Step*** 3: Mean number of jobs

Iteration	$c^2_{A_1}$	$c^2_{A_2}$	$c^2_{A_3}$	$c^2_{A_4}$
Methods	$\overline{K}_1$	$\overline{K}_2$	$\overline{K}_3$	$\overline{K}_4$
AC	6.48	0.78	1.46	0.31
KLB	6.21	0.77	1.42	0.27
Sim	4.62	0.57	1.38	0.23

AC: Allen-Cunneen-Approximation

KLB: Krämer/Langenbach-Belz-Approximation

4 Priority Queueing Networks

◆ Description:

➤ Multiple job classes with priorities $1, 2, \ldots, R$

➤ 1 : highest priority; R : lowest priority

➤ Two additional node types:

M/M/1-FCFS-NONPRE (without preepmtion)

M/M/1-FCFS- PRE (with preemption)

➤ Only approximate solutions:

➤ Extended MVA (PRIOMVA)

➤ Shadow Method

■ PRIOMVA:

◆ MVA for multiple job classes:

➤ Mean response time (arrival theorem);

$$\overline{T}_{ir}(\mathbf{k}) = \begin{cases} \dfrac{1}{\mu_{ir}}\left[1 + \sum_{s=1}^{R} \overline{K}_{is}(\mathbf{k} - \mathbf{1}_r)\right] & \text{Type-1,2,4 } (m_i = 1), \\ \dfrac{1}{\mu_{ir}}, & \text{Type-3.} \end{cases}$$

$(\mathbf{k} - \mathbf{1}_r) = (k_1, \ldots, k_r - 1, \ldots, k_R)$ is the population vector with one class-r job less in the system

➤ Mean response time (M/M/1-PRE node):

$$\overline{T}_{ir}(\mathrm{k}) = \frac{\dfrac{1}{\mu_{ir}} + \sum\limits_{s=1}^{r} \dfrac{\overline{K}_{is}(\mathrm{k}-1_r)}{\mu_{is}}}{1 - \sum\limits_{s=1}^{r-1} \rho'_{is}},$$

➤ Mean response time (M/M/1-NONPRE node):

$$\overline{T}_{ir}(\mathrm{k}) = \frac{1}{\mu_{ir}} + \frac{\sum\limits_{s=1}^{r} \dfrac{\overline{K}_{is}(\mathrm{k}-1_r)}{\mu_{is}} + \sum\limits_{s=r+1}^{R} \dfrac{\rho_{is}(\mathrm{k}-1_r)}{\mu_{is}}}{1 - \sum\limits_{s=1}^{r-1} \rho'_{is}},$$

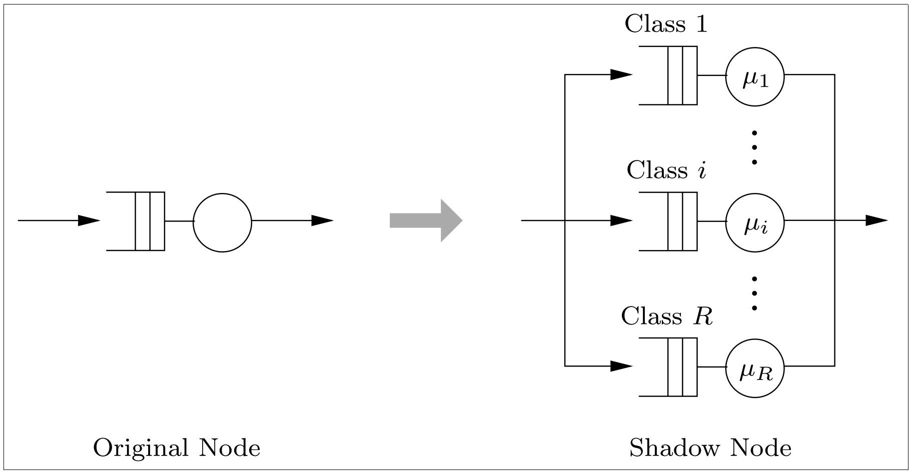
■ Shadow Method for M/M/1-FCFS-PRE Nodes:

➤ M/M/1-FCFS-PRE-Knoten is replaced by a "shadow node" with R parallel M/M/1-FCFS nodes.

➤ Thus a product-form queueing network arises, which can be analyzed by MVA, SCAT or convolution.

➤ Since the jobs of the different job classes are processed in parallel, the service times $s_{ir} = 1/\mu_{ir}$ have to be extended. This has to be done, that the mean response times of the shadow nodes are equal to those of the original M/M/1-FCFS-PRE nodes. This is done iteratively using an approximation formula. The initial value is s_{ir}.

◆ **Algorithm**:



- ***Step*** 1: Transform the original model into the shadow model
- ***Step*** 2: Set $\lambda_{i,r} = 0$
- ***Schritt*** 3: Iteration
  - ➤ *Step* 3.1: Compute the utilization for each shadow node:

$$\tilde{\rho}_{i,r} = \lambda_{i,r} \cdot \tilde{s}_{i,r}$$

  - ➤ *Step* 3.2: Compute the shadow service times:

$$\tilde{s}_{i,r} = \frac{s_{i,r}}{1 - \sum_{s=1}^{r-1} \tilde{\rho}_{i,s}}$$

$s_{i,r}$: original service time of a class r job at node i job

$\tilde{s}_{i,r}$: approximated service time in the shadow node

- *Step* 3.3: Compute the new values of the throughput λ_{ir} of class r at node i.

- ***Step*** 4: If the λ_{ir} differ less than ε in two successive iterations, then stop the iteration. Otherwise go back to Step 3.1