4 Approximation Algorithms for Product-Form Networks

Approximation based on the MVA (Bard-Schweitzer Method):

➤ Arrival theorem:

$$\overline{T}_i(K) = \frac{1}{\mu_i} \cdot \left[1 + \overline{K}_i(K-1)\right] \; , \qquad i = 1, \ldots , N$$

➤ Iteration using the following Approximation:

$$\overline{K}_i(K-1) = \frac{K-1}{K} \cdot \overline{K}_i(K)$$

➤ Initialization:

$$\overline{K}_1(K) = \overline{K}_2(K) = \cdots = \overline{K}_N(K) = \frac{K}{N}$$

◆ **Algorithm**:

- ***Step*** 1: Initialization:

$$\overline{K}_1(K) = \overline{K}_2(K) = \cdots = \overline{K}_N(K) = \frac{K}{N}$$

- ***Step*** 2: Approximation:

$$\overline{K}_i(K-1) = \frac{K-1}{K} \cdot \overline{K}_i(K) \quad i = 1, 2, \ldots , N$$

- ***Step*** 3: Mean response time:

$$\overline{T}_i(K) = \frac{1}{\mu_1}\left[1 + \overline{K}_i(K-1)\right] \quad i = 1, \cdots , N \quad \text{Typ 1,2,4}$$

$$\overline{T}_i(K) = \frac{1}{\mu_i} \quad i = 1, 2, \cdots , N \qquad \text{Typ 3}$$

- ***Step*** 4: Throughput:

$$\lambda(K) = \frac{K}{\sum\limits_{i=1}^{N} e_i \overline{T}_i(K)}$$

- ***Step*** 5: Mean number of jobs:

$$\overline{K}_i(K) = \overline{T}_i(K)\lambda(K)e_i \quad i = 1, 2, \cdots, N$$

- ***Step*** 6: Check the stopping condition:

$$\max_i \left|\overline{K}_i^{(n)}(K) - \overline{K}_i^{(n-1)}(K)\right| \quad < \epsilon$$

If the stopping condition is not fulfilled return back to ***Step*** 2

◆ **Example**: Queueing network with $K = 6$ and $\varepsilon = 0.06$ and:

i	e_i	$1/\mu_i$	m_i
1	1	0.02	1
2	0.4	0.2	1
3	0.2	0.4	1
4	0.1	0.6	1

- ***Step*** 1: Initialization:

$$\overline{K}_1(K) = \overline{K}_2(K) = \overline{K}_3(K) = \overline{K}_4(K) = \frac{K}{N} = \underline{1.5}$$

- ***Step*** 2: Approximation ($K = 6$):

$$\overline{K}_1(K-1) = \frac{K-1}{K} \cdot \overline{K}_1(K) = \underline{1.25},$$
$$\overline{K}_2(K-1) = \overline{K}_3(K-1) = \overline{K}_4(K-1) = \underline{1.25}.$$

- ***Step*** 3: Mean response time:

$$\overline{T}_1(K)=\frac{1}{\mu_1}\left[1+\overline{K}_1(K-1)\right]=\underline{0.045}\,,\quad \overline{T}2(K)=\frac{1}{\mu_2}\left[1+\overline{K}_2(K-1)\right]=\underline{0.45}\,,$$
$$\overline{T}_3(K)=\frac{1}{\mu_3}\left[1+\overline{K}_3(K-1)\right]=\underline{0.9}\,,\qquad \overline{T}_4(K)=\frac{1}{\mu_4}\left[1+\overline{K}_4(K-1)\right]=\underline{1.35}\,.$$

- ***Step*** 4: Throughput:

$$\lambda(K)=\frac{K}{\sum_{i=1}^{4} e_i\overline{T}_i(K)}=\underline{11.111}$$

- ***Step*** 5: Mean number of jobs:

$$\overline{K}_1(K)=\overline{T}_1(K)\lambda(K)e_1=\underline{0.5}\,,\quad \overline{K}_2(K)=\overline{T}_2(K)\lambda(K)e_2=\underline{2}\,,$$
$$\overline{K}_3(K)=\overline{T}_3(K)\lambda(K)e_3=\underline{2}\,,\qquad \overline{K}_4(K)=\overline{T}_4(K)\lambda(K)e_4=\underline{1.5}\,.$$

- ***Step*** 6: Stopping condition:

$$\max_i\left|\overline{K}_i^{(1)}(K)-\overline{K}_i^{(0)}(K)\right|=\underline{1}\quad >0.06\,.$$

<u>2. Iteration</u>:

- ***Step*** 2: Approximmation:

$$\overline{K}_1(K-1)=\frac{K-1}{K}\cdot 0.5=\underline{0.417}\,,$$
$$\overline{K}_2(K-1)=\underline{1.667}\,,\quad \overline{K}_3(K-1)=\underline{1.667}\,,\quad \overline{K}_4(K-1)=\underline{1.25}\,.$$

After 4 Iterations:

- ***Step*** 6: Stopping condition :

$$\max_i\left|\overline{K}_i^{(4)}(K)-\overline{K}_i^{(3)}(K)\right|=\underline{0.053}\quad <0.06\,.$$

➤ Final results:

Node:	1	2	3	4
Mean response time $\overline{T}_i$	0.024	0.573	1.145	1.240
Throughput λ_i	9.896	3.958	1.979	0.986
Mean number of jobs $\overline{K}_i$	0.239	2.267	2.267	1.240
Utilization ρ_i	0.198	0.729	0.729	0.594

➤ Exact results (MVA):

Node	1	2	3	4
Mean response time $\overline{T}_i$	0.025	0.570	1.140	1.244
Throughput λ_i	9.920	3.968	1.984	0.992
Mean number of jobs $\overline{K}_i$	0.244	2.261	2.261	1.234
Utilization ρ_i	0.198	0.794	0.794	0.595

➤ Validation:

- Accuracy is sufficient for many applications
  (Mean deviation 6%)
- No M/M/m - nodes considered
- Less computing time and less memory than **MVA**
- Multiple job classes
- Better accuracy and M/M/m - nodes using **SCAT**- Algorithm

➤ **SCAT** (Self-Correction Approximation Technique):

- Multiple Application of the BS-Algorithm
- Very good accuracy, M/M/m - nodes, few memory and computing time
- Core of the SCAT-Algorithm:
  - Help functions:
    $F_i(K) = \overline{K}_i(K)/K$ und $D_i(K) = F_i(K\text{-}1) - F_i(K)$
  - Improved Approximation:
    $\overline{K}_i(K\text{-}1) = (K\text{-}1)(F_i(K) + D_i(K))$
  - D_i: Korrection function, is improved iteratively
  - $D_i = 0 \rightarrow$ BS- Algorithm

Summation Method:

◆ Fundamentals: :

➤ Mean number of jobs: $\overline{K}_i = f_i(\lambda_i)$

mit:

$$f_i(\lambda_i) = \overline{K}_i = \begin{cases} \dfrac{\rho_i}{1 - \dfrac{K-1}{K}\rho_i}, & \text{Type-1,2,4 } (m_i = 1), \\ m_i\rho_i + \dfrac{\rho_i}{1 - \dfrac{K-m_i-1}{K-m_i}\rho_i} \cdot P_{m_i}, & \text{Type-1 } (m_i > 1), \\ \dfrac{\lambda_i}{\mu_i}, & \text{Type-3.} \end{cases}$$

➤ System equation:

$$\sum_{i=1}^{N} \overline{K}_i = \sum_{i=1}^{N} f_i(\lambda_i) = K$$

◆ Algorithm:

- ***Step*** 1: Initialization:

  Lower limit of the throughput: $\lambda_u = 0$

  Upper limit of the throughpu: λ_o = $\min_i \left\{ \frac{\mu_i m_i}{e_i} \right\}$

- ***Step*** 2: Bisection technique

  ➤ *Step* 2.: Throughput:

$$\lambda = \frac{\lambda_u + \lambda_o}{2}$$

  ➤ *Step* 2.2: Determination of:

$$g(\lambda) = \sum_{i=1}^{N} f_i(\lambda \cdot e_i)$$

➤ Step 2.3: Stopping condition:

$$g(\lambda) = K \pm \epsilon$$

Calculation of the performance measures using λ_i and ρ_i which are determined by the approximation formulas.

Else:

If $g(\lambda) > K$, set $\lambda_o = \lambda$ and go back to Step $\mathbf{2.1}$.
If $g(\lambda) < K$, set $\lambda_u = \lambda$ and go also back to Step $\mathbf{2.1}$.

© Gunter Bolch • Universität Erlangen-Nürnberg • Informatik 4 • 2002

Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

◆ **Example**: Closed queueing network with $N = 4$, $K = 3$, $\varepsilon = 0.001$ and:

i	e_i	$1/\mu_i$	m_i
1	1	0.5	2
2	0.5	0.6	1
3	0.5	0.8	1
4	1	1	∞

- ***Step*** 1: Initialization:

$$\lambda_u = \underline{0} \quad \text{und} \quad \lambda_o = \min_i \left\{ \frac{m_i \mu_i}{e_i} \right\} = \underline{2.5}$$

- ***Step*** 2:
  - ➤ *Step* 2.1: Throughput:

$$\lambda = \frac{\lambda_u + \lambda_o}{2} = \underline{1.25}$$

➤ Step 2.2: Computation of the functions $f_i(\lambda_i)$

$$\rho_1 = \frac{\lambda \cdot e_1}{\mu_1 \cdot m_1} = \underline{0.3125} \quad \text{and} \quad P_{m_1} = \underline{0.149}$$

$$f_1(\lambda_1) = 2\rho_1 + \rho_1 P_{m_1} = \underline{0.672}$$

$$f_2(\lambda_2) = \frac{\rho_2}{1 - \frac{2}{3}\rho_2} = \underline{0.5} \quad \text{with} \quad \rho_2 = \underline{0.375},$$

$$f_3(\lambda_3) = \frac{\rho_3}{1 - \frac{2}{3}\rho_3} = \underline{0.75} \quad \text{with} \quad \rho_3 = \underline{0.5},$$

$$f_4(\lambda_4) = \frac{\lambda_4}{\mu_4} = \underline{1.25}.$$

and:

$$g(\lambda) = \sum_{i=1}^{N} f_i(\lambda_i) = \underline{3.172}$$

➤ *Step* 2.3: Stopping condition:

$$g(\lambda) > K \text{ therefore } \lambda_o = \lambda = 1.25$$

2. Iteration:

➤ Step 2.1: Throughput:

$$\lambda = \frac{\lambda_u + \lambda_o}{2} = \underline{0.625}$$

etc..

Intervals for λ:

Step:	0	1	2	3	4	5	6	7	8	9	10
λ_l	0	0	0.625	0.9375	1.094	1.172	1.172	1.191	1.19 1	1.191	1.191
λ_u	2.5	1.25	1.25	1.25	1.25	1.25	1.211	1.211	1.20 1	1.196	1.194

Final value λ = 1.193 Exact value: λ = 1.217

Exact and approximate values:

i	1	2	3	4
$\lambda_{i_{\text{SUM}}}$	**1.193**	**0.596**	**0.596**	**1.193**
$\lambda_{i_{\text{MVA}}}$	**1.218**	**0.609**	**0.609**	**1.218**
$\overline{K}_{i_{\text{SUM}}}$	**0.637**	**0.470**	**0.700**	**1.193**
$\overline{K}_{i_{\text{MVA}}}$	**0.624**	**0.473**	**0.686**	**1.217**

◆ Summation method also applicable to networks with multiple job classes

Bounds Analysis:

Conditions:

- Only upper and lower bounds of throughput and response time are calculated:

$$\lambda_{\text{pes}} \leq \lambda \leq \lambda_{\text{opt}} \quad \text{and} \quad \overline{T}_{\text{opt}} \leq \overline{T} \leq \overline{T}_{\text{pes}}$$

- Only one job class

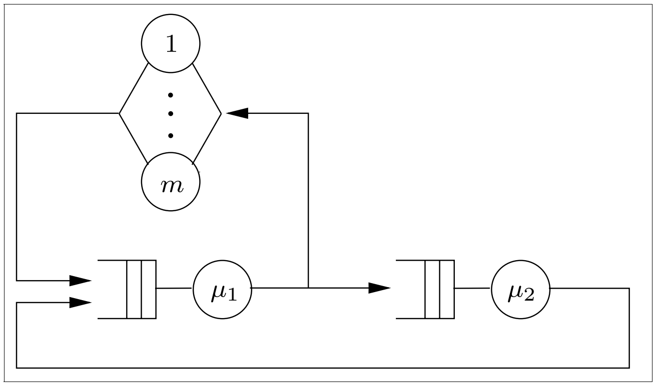
- Three different network types:
  - Typ A: Closed network **without** IS - nodes
  - Typ B: Closed network **with** IS - nodes
  - Typ C: Open network

ABA - Asymptotic Bounds Analysis:

	Network Type	ABA Bounds
λ	A	$\lambda(K) \leq \min\left\{\frac{K}{x_{\text{sum}}}, \frac{1}{x_{\text{max}}}\right\}$
	B	$\lambda(K) \leq \min\left\{\frac{K}{x_{\text{sum}}+Z}, \frac{1}{x_{\text{max}}}\right\}$
	C	$\lambda \leq \frac{1}{x_{\text{max}}}$
$\overline{T}$	A	$\overline{T}(K) \geq \max\{x_{\text{sum}}, K \cdot x_{\text{max}}\}$
	B	$\overline{T}(K) \geq \max\{x_{\text{sum}}, K \cdot x_{\text{max}} - Z\}$
	C	$\overline{T} \geq x_{\text{sum}}$

$$x_i = e_i/\mu_i \qquad x_{\text{max}} = \max_i(x_i) \quad \text{and} \quad x_{\text{sum}} = \sum_i x_i$$

➤ Example:



$$1/\mu_1 = 4.6\,, \quad 1/\mu_2 = 8\,, \qquad 1/\mu_3 = 120 = Z\,,$$
$$e_1 = 2\,, \qquad e_2 = e_3 = 1\,.$$

$$x_{\max} = \max\left\{\frac{e_1}{\mu_1}, \frac{e_2}{\mu_2}\right\} = \underline{9.2}\,, \quad x_{\text{sum}} = \frac{e_1}{\mu_1} + \frac{e_2}{\mu_2} = \underline{17.2}\,,$$
$$Z = \frac{e_3}{\mu_3} = \underline{120}\,.$$

Throughput:

$$\lambda(K) \le \min\left\{\frac{K}{x_{\text{sum}} + Z}, \frac{1}{x_{\max}}\right\} = \min\left\{\frac{20}{137.2}; \frac{1}{9.2}\right\} = \underline{0.109}\,.$$

Mean response time:

$$\overline{T}(K) \ge \max\,\{x_{\text{sum}}\,, K \cdot x_{\max} - Z\} = \max\,(17.2\,; 64) = \underline{64}\,.$$

Exact values:

$$\lambda(K) = \underline{0.100} \text{ and } \overline{T}(K) = \underline{80.28}$$

◆ BJB - Balanced Job Bounds Analysis: $x_{ave} = x_{sum}/N$

	Network Type	BJB Bounds
λ	A	$\frac{K}{x_{sum} + (K-1)x_{max}} \le \lambda(K) \le \frac{K}{x_{sum} + (K-1)x_{ave}}$
	B	$\frac{K}{x_{sum} + Z + \frac{(K-1)x_{max}}{1+\frac{Z}{K \cdot x_{sum}}}} \le \lambda(K) \le \frac{K}{x_{sum} + Z + \frac{(K-1)x_{ave}}{1+\frac{Z}{x_{sum}}}}$
	C	$\lambda \le \frac{1}{x_{max}}$
$\overline{T}$	A	$x_{sum} + (K-1)x_{ave} \le \overline{T}(K) \le x_{sum} + (K-1)x_{max}$
	B	$x_{sum} + \frac{(K-1)x_{ave}}{1+\frac{Z}{x_{sum}}} \le \overline{T}(K) \le x_{sum} + \frac{(K-1)x_{max}}{1+\frac{Z}{K \cdot x_{sum}}}$
	C	$\frac{x_{sum}}{1-\lambda x_{ave}} \le \overline{T} \le \frac{x_{sum}}{1-\lambda x_{max}}$

➤ Example:

- Throughput:

$$\underline{0.075} \le \lambda(K) \le \underline{0.127}$$

- Mean response time:

$$\underline{37.70} \le \overline{T}(K) \le \underline{146.8}$$

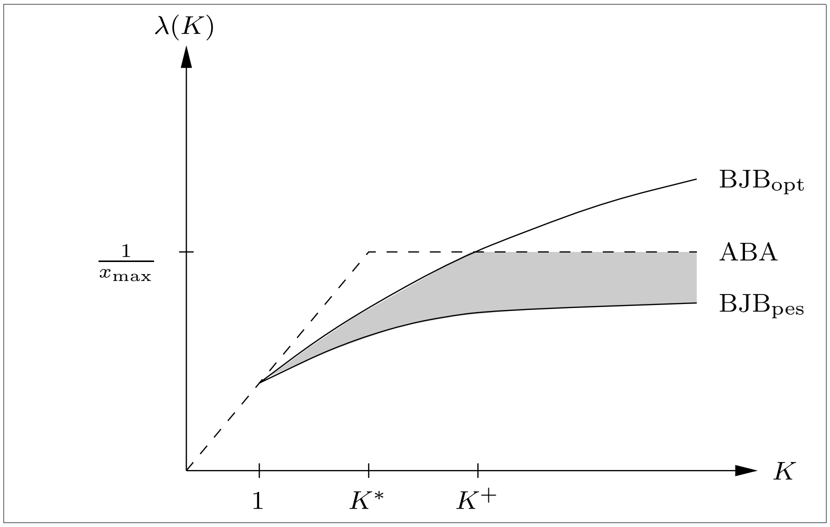
- Exact values:

$$\lambda(K) = \underline{0.100} \text{ and } \overline{T}(K) = \underline{80.28}$$

- ABA:

$$\lambda(K) = \underline{0.109} \text{ and } \overline{T}(K) = \underline{64}$$

- Throughput as a function of K:



- Mean response time as a function of K:

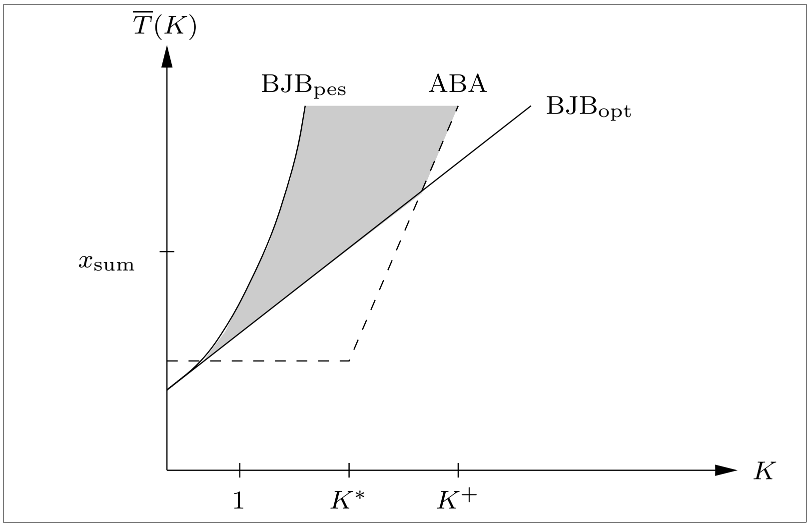


Table 0.1 Comparison of the approximation algorithms for product-form queueing networks.

Algorithms	Advantages	Disadvantages
Bard-Schweitzer (BS)	Very low storage and time requirement	No multiple server nodes Low accuracy
SCAT	Good accuracy Very low storage requirement compared with MVA or convolution	Needs more iterations than BS
SUM	Easy to understand and implement Low storage and time requirement Easy to extend to non-product-form networks	Accuracy is not very high (but sufficient for most applications)
ABA, BJB	Well suited for a bottleneck analysis In the design phase, to obtain a rough prediction (insight, understanding) of the performance of a system Extremely low storage and time requirement	Only for single class networks Only upper and lower bounds