# Lecture 9
# Multiple Class Models

Multiclass MVA

Approximate MVA

# Arrival Theorem for Multiple Classes

- With $\bar{N}$ jobs in the system, a job in class r arriving to any server i sees the server as in equilibrium with one job (himself!) removed from the system

$(N_1, N_2, \ldots, N_R)$ R job classes

$$A_{ir}(\bar{\mathbf{N}}) = \bar{n}_i(\bar{\mathbf{N}} - \bar{\mathbf{1}}_r) = \bar{n}^A_{ir}(\bar{N})$$

$r^{th}$ job class

$(0, 0, \ldots, 0, 1, 0, \ldots, 0)$

# Multiple Class MVA (4)

starting point: $A_{ir}(\bar{N}) = \bar{n}_i(\bar{N} - \bar{1}_r) = \bar{n}^A_{ir}(\bar{N})$ $\quad \bar{n}_{i,r}(\mathbf{0}) = 0$

residence time

$$R'_{i,r}(\bar{N}) = \begin{cases} D_{i,r}[1 + \bar{n}^A_{i,r}(\bar{N})] & \textit{queueing device} \\ D_{i,r} & \textit{delay device} \end{cases} \quad \forall i, r$$

$\bar{N}+1_r$

$$X_{0,r}(\bar{N}) \underset{system}{\overset{Little}{=}} \frac{N_r}{Z_r + \sum_i R'_{i,r}(\bar{N})} \quad \forall r$$

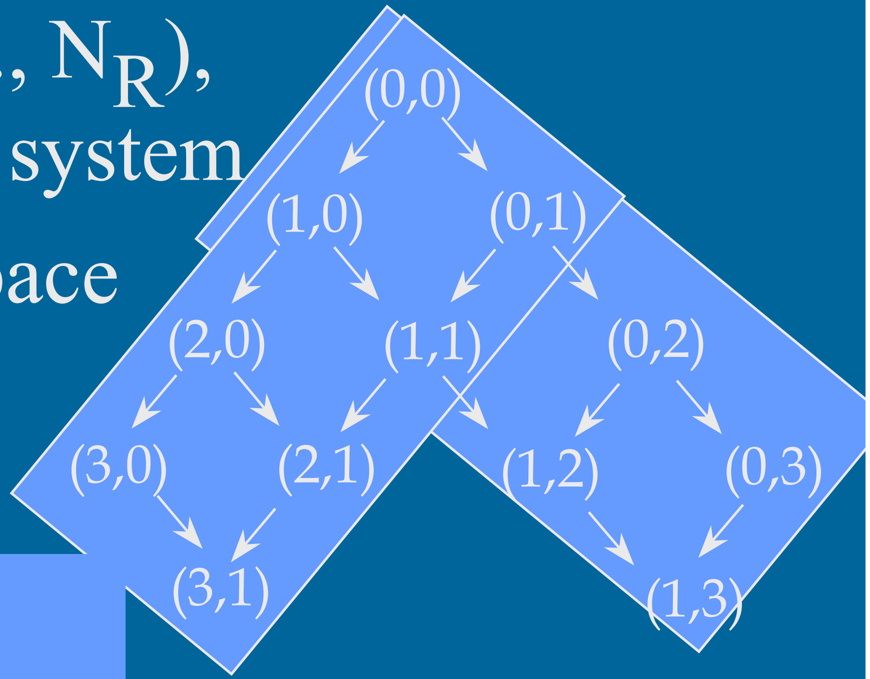
(response time law)

$$\bar{n}_{i,r}(\bar{N}) \underset{device}{\overset{Little}{=}} X_{i,r}(\bar{N}) R_{i,r}(\bar{N}) = X_{0,r}(\bar{N}) R'_{i,r}(\bar{N}) \quad \forall i, r$$

$$\bar{n}_i(\bar{N}) = \sum_r \bar{n}_{i,r}(\bar{N}) \quad \forall i$$

# Multiple Class Mean Value Analysis (MVA) (2)

- Compute solutions through class population space $N=(N_1, N_2, ..., N_R)$, starting from empty system
- population (state) space can still be large!




two job classes
target population (1,3)?
target population (3,1)?
target population (6, 15, 300)?

Alg 7.2 [LZGS 84]

# Example (4)

2-class closed model, Fig. 7.2 [LZGS 84]
Job classes A, B

Tbl 7.3 [LZGS 84]

(0,0) $Q_{CPU,A} = Q_{DISK,A} = Q_{CPU,B} = Q_{DISK,B} = 0$

(1,0) $R'_{CPU,A}$, $R'_{DISK,A}$, $X_{0,A}$, $Q_{CPU,A}$, $Q_{DISK,A}$

(0,1) $R'_{CPU,B}$, $R'_{DISK,B}$, $X_{0,B}$, $Q_{CPU,B}$, $Q_{DISK,B}$

(1,1) $R'_{CPU,A}$, $R'_{DISK,A}$, $R'_{CPU,B}$, $R'_{DISK,B}$,
$X_{0,A}$, $X_{0,B}$,
$Q_{CPU,A}$, $Q_{DISK,A}$, $Q_{CPU,B}$, $Q_{DISK,B}$

# Another Simple Example

Fig. 6.1 [Men 94]

| $D_{i,r}$ | query r=1 | update r=2 |
|---|---|---|
| cpu i=1 | 0.105 | 0.375 |
| d1 i=2 | 0.180 | 0.480 |
| d3 i=3 | 0 | 0.240 |

Watch out for index ordering:

$D_{1,2} = D_{2,\text{query}}$

Population in book figures in order (Update, Query)

# Simple Example (5)

update
query

$$(0,0) \quad \bar{n}_i(\bar{0}) = 0 = \bar{n}^{A}_{i,r}(1_r) \qquad \forall i, r$$

(1,0)

$$R'_{1,1}(1,0) = D_{1,1}[1+0] = 0.105 \qquad R'_{2,1}(1,0) = 0.180 \qquad R'_{3,1}(1,0) = 0$$

$$X_{0,1}(1,0) = \frac{1}{(0.105 + 0.180)} = 3.509$$

$$\bar{n}_{1,1}(1,0) = 3.509 * 0.105 = 0.368$$

$$\bar{n}_{2,1}(1,0) = 3.509 * 0.180 = 0.632 \qquad \bar{n}_{3,1}(1,0) = 0$$

$$R'_{1,2}(1,0) = ? \qquad \bar{n}_{1,2}(1,0) = \bar{n}_{2,2}(1,0) = \bar{n}_{3,2}(1,0) = 0$$

$$\bar{n}_1(1,0) = \bar{n}_{1,1}(1,0) + \bar{n}_{1,1}(1,0) = 0.368$$

$$\bar{n}_2(1,0) = 0.632 \qquad \bar{n}_3(1,0) = 0$$

Tbl 6.3

# Simple Example (contd) (6)

(0,1): $\bar{n}_1(0,1) = 0.343 \quad \bar{n}_2(0,1) = 0.438 \quad \bar{n}_3(0,1) = 0.219$

(1,1): $R'_{1,1}(1,1) = D_{1,1}\left[1 + \bar{n}^A_{1,1}(1,1)\right] = D_{1,1}\left[1 + \bar{n}_1(0,1)\right]$

$= 0.105[1 + 0.343] = 0.141$

$R'_{2,1}(1,1) = D_{2,1}\left[1 + \bar{n}_2(0,1)\right] = 0.180[1 + 0.438] = 0.259$ $\quad R'_{3,1}(1,1) = 0$

$X_{0,1}(1,1) = 1 / (0.141 + 0.258) = 2.500$

device index

class index

total popul.

| | | |
|---|---|---|
| $\bar{n}_{1,1}(1,1) = 2.5 * 0.141 = 0.352$ | $\bar{n}_{1,2}(1,1) = 0.334$ | $\bar{n}_1(1,1) = 0.686$ |
| $\bar{n}_{2,1}(1,1) = 2.5 * 0.259 = 0.648$ | $\bar{n}_{2,2}(1,1) = 0.519$ | $\bar{n}_2(1,1) = 1.158$ |
| $\bar{n}_{3,1}(1,1) = 0$ ... | $\bar{n}_{3,2}(1,1) = 0.156$ | $\bar{n}_3(1,1) = 0.156$ |

similarly for states (2,0), (3,0), (2,1), (3,1) Tbl 6.3

# Simple Example (contd)

- Baseline solution

  Table 6.3 [Men 94]

  – external: class *update* response time: 2.444 s
  – internal: Disk1 utilization too high

  Table 6.4

- Modification: move queries to Disk2
  – move query demand from Disk 1 to Disk2
    - $D_{21} = D_{d1,query} = 0, \; D_{31} = D_{d2,query} = 0.180$

(silly order, silly notation, sorry!)

  – solve again

  Table 6.4 [Men 94]

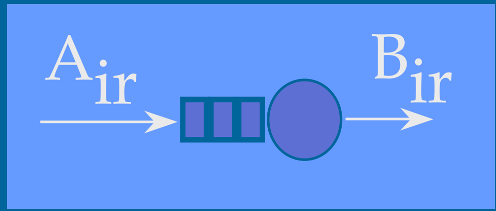
  – class *update* response time: 1.934 s

# Product Form Solution Exists: NW is Analytically Solvable

- BCMP-networks
  - Baskett, Chandy, Muntz & Palacios (1975)
- Service discipline
  - FCFS, PS, IS, LCFS-PR
- Job classes, class switching
- Service time distributions, interarrival times
  - Exponential interarrivals times for FCFS servers or for open job class
  - more generic for others (rational Laplace transformation exists)
- Load-dependent servers (LD-servers)
  - $S_{ir} = f(\bar{n}_i)$ for FCFS (I.e., same for each class)
  - $S_{ir} = f(\bar{n}_{ir})$ for others

# Product Form Solution Exists?

- Job Flow balance
  - flow in = flow out
  - per device, per system
- One step behavior
- Device homogeneity
  - single resource possession
  - no blocking
  - independent job behavior
  - local information
  - fair service
  - routing homogeneity




$\mu_{ir} = f\ (n_{ir})$
$\mu_i = f\ (n_i)$

$p_{ikr}$ constant

(prob for class change is load independent)

# How Useful are Exact Solution Methods?

- MVA & Convolution based algorithms
- Very good for single class cases
- Might be too time consuming for multiple class cases if nr of classes or class populations are (very) large

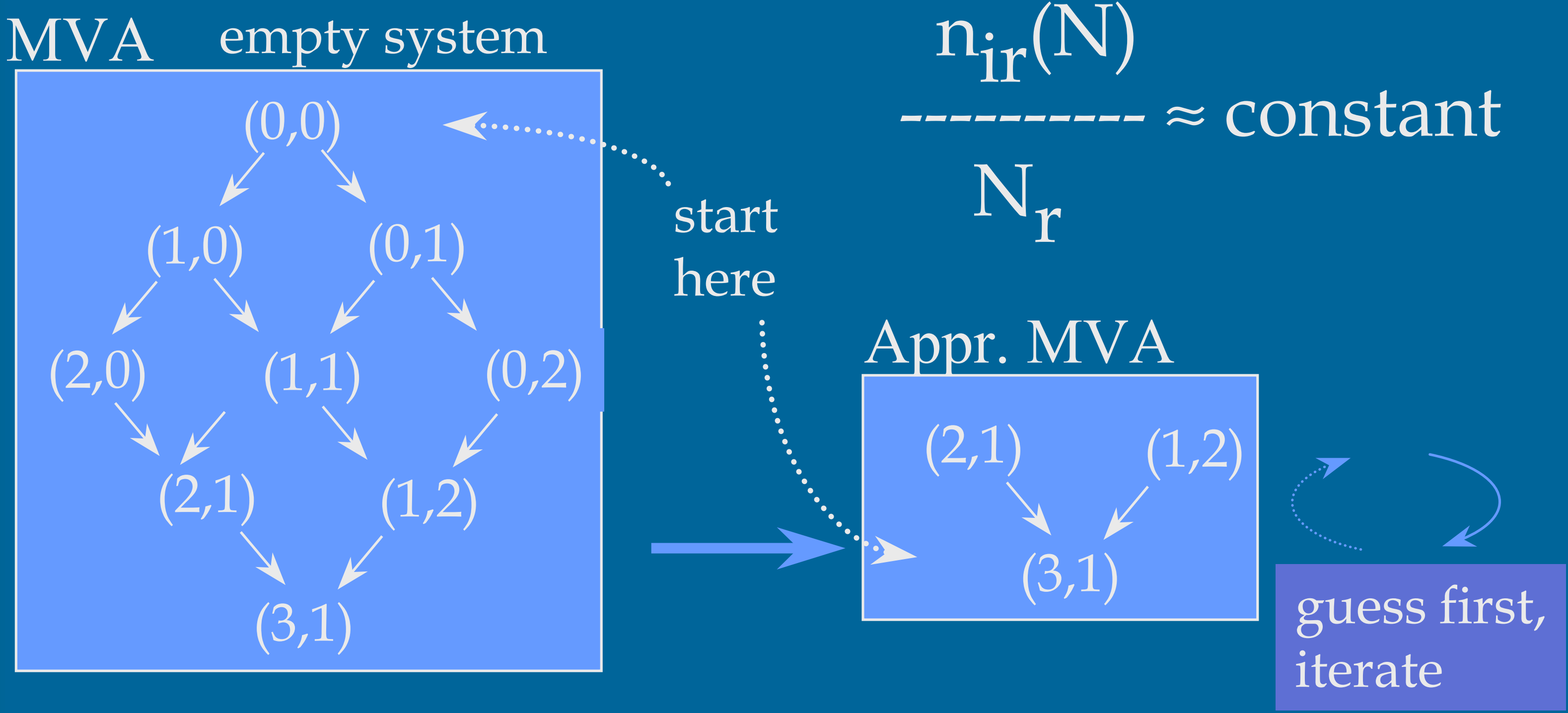
$$\mathbf{O}(\text{MVA}) = \mathbf{O}(K R \prod_{\mathbf{r}} (1 + N_r))$$

| K | R | $N_r$ | #ops |
|---|---|---|---|
| 3 | 2 | 2 | 54 |
| 5 | 20 | 1 | 100M |
| 20 | 5 | 1 | 3200 |
| 20 | 5 | 9 | 10M |

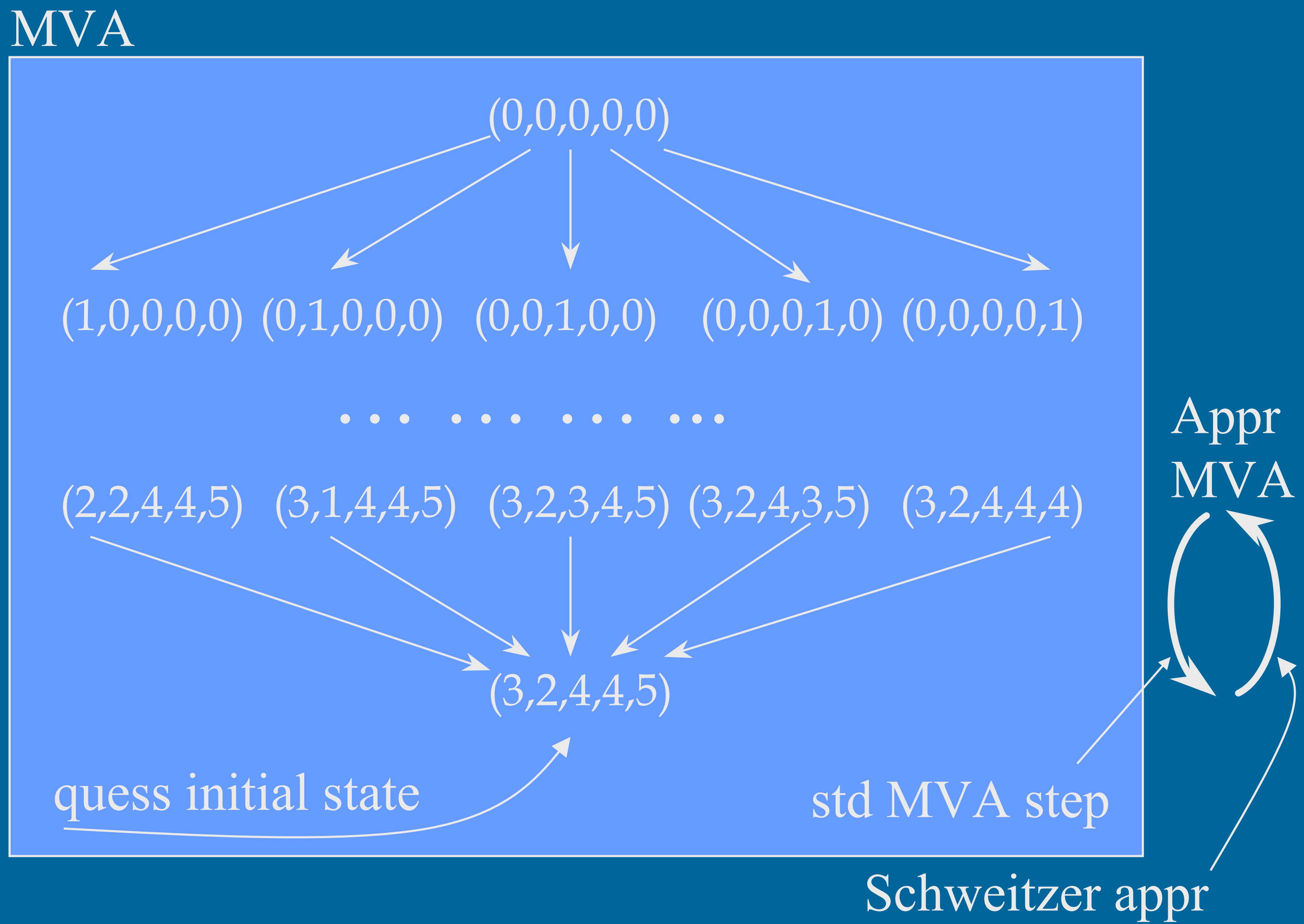
# Approximate MVA (1)

- Helps to solve MVA state space problem
- Based on Schweitzer-approximation:

$$\frac{n_{ir}(N)}{N_r} \approx \text{constant}$$



# MVA vs Approximate MVA



# Single Class
# Schweitzer Approximation

$$\frac{\overline{n}_i(N)}{N} \approx \text{constant}$$

$$\therefore \frac{\overline{n}_i(N)}{N} = \frac{\overline{n}_i(N-1)}{N-1} \qquad \Rightarrow \overline{n}_i(N-1) = \frac{N-1}{N}\,\overline{n}_i(N)$$

# Single Class Approximate MVA

$$guess\ initial: \quad \overline{n}_i(N)$$

$$compute: \quad \overline{n}_i(N-1) = \frac{N-1}{N}\overline{n}_i(N)$$

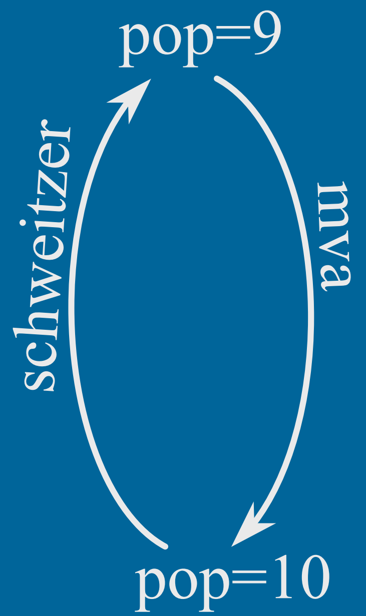
$$std\,MVA\ step:$$

$$R'_i(N) = D_i\left[1+\overline{n}_i(N-1)\right]$$

$$X_0(N) = N \Big/ \left(Z + \sum R'_i(N)\right)$$

$$\overline{n}_i(N) = X_0(N) * R'_i(N)$$

repeat until conver-gence



# Example (4)

$D_i$ = (10, 10, 15), K=3, N=2 [Fig. 5.1]

step 0 $\bar{n}_i(2) = 2/3 = 0.667 \quad \forall i$ initial guess: even distribution

step 1

$$R_1' = D_1\left[1 + \frac{2-1}{2}\bar{n}_1\right] = 10 * 1.333 = 13.333 = R_2' \qquad R_3' = 15 * 1.333 = 20$$

$$X_0 = \frac{2}{46.67} = 0.04286$$

$$\bar{n}_1 = 0.5714 = \bar{n}_2 \qquad \bar{n}_3 = 0.8572 \qquad \left(\sum \bar{n}_i = 2\right)$$

step 2

$$R_1' = 10\left[1 + \frac{0.5714}{2}\right] = 12.857 = R_2' \qquad R_3' = 15 * \left[1 + \frac{0.8572}{2}\right] = 21.429$$

$$X_0 = \frac{2}{47.143} = 0.042424$$

$$\bar{n}_1 = 0.5454 = \bar{n}_2 \qquad \bar{n}_3 = 0.9092 \qquad \left(\sum \bar{n}_i = 2\right)$$

0.5455 exact

0.9090 exact

why stop here?

# Questions (3)

- Q. How to quess initial job distribution

  A: $$\bar{n}_i(N) = \frac{N}{K}$$ (even distribution on all devices)

- Q. Does it always converge?

  A. No. Almost always. No quarantee.

- Q. If it converges, does it converge to the right value?

  A. We hope so. It seems to do it ….

# Questions (contd)

- Q. What is good measure of convergence?

  A. E.g., max relative change in $\overline{n}_i(N)$ must be less than 1%

# General Stationary Iterative Method (1)

(from Numerical Linear Algebra)

Fixed point equation $\bar{n} = B\bar{n} + \bar{c}$

converges from arbitrary initial $\bar{n}^{(0)}$

if $\rho(B) = \max|\lambda_i(B)| < 1$

spectral radius of B

i[th] eigenvalue of B

- Q. Do we check this before using approximate MVA?

  A. No.

# General Theory of Iteration (1)

$$\bar{n} = f(\bar{n})$$

*Thm*:

If $\bar{n} = f(\bar{n})$ has root $\bar{\alpha}$

and $f(\bar{n})$ exists close to $\alpha$, i.e., in $J = \{\bar{n}: |\bar{n} - \bar{\alpha}| < \rho\}$

and $|f'(\bar{n})| < 1 \quad \forall \bar{n} \in J$

Then (a) $\bar{n} \in J$ in each iteration

(b) $\bar{n}$ converges to $\bar{\alpha}$

(c) $\bar{\alpha}$ is the only root in $J$

- Q. Do we check this for approximate MVA?

  A. No.

# How Good is Approximate MVA

- Pretty good for throughput and response time
- Not so good for queue lengths at heavy loads

Figs 34.1-34.3 [Jain 91]

# Multiple Class Schweitzer Approximation (1)

single class:

$$\frac{\bar{n}_i(N)}{N} \approx \text{constant}$$

$$\therefore \quad \frac{\bar{n}_i(N)}{N} = \frac{\bar{n}_i(N-1)}{N-1} \quad \Rightarrow \quad \bar{n}_i(N-1) = \frac{N-1}{N}\,\bar{n}_i(N)$$

multiple class:

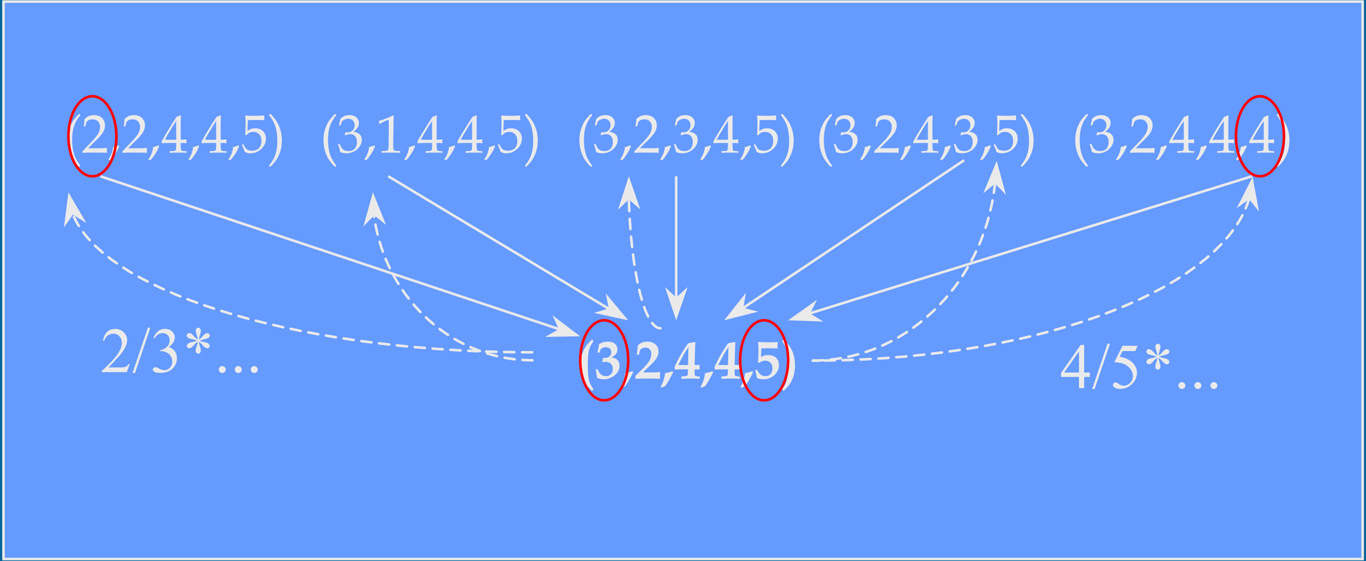
$$\frac{\bar{n}_{ir}(\overline{N})}{N_r} \approx \text{constant}$$

$$\therefore \quad \frac{\bar{n}_{ir}(\overline{N})}{N_r} = \frac{\bar{n}_{ir}(\overline{N-1_r})}{N_r - 1}$$

$$\Rightarrow \quad \bar{n}_{ir}(\overline{N-1_r}) = \frac{N_r - 1}{N_r}\,\bar{n}_{ir}(\overline{N})$$

# Multiple Class Approximate MVA

$$\bar{n}_{ir}(\bar{N} - 1_r) = \frac{N_r - 1}{N_r}\,\bar{n}_{ir}(\bar{N})$$



# Multiple Class Approximate MVA

(distribute evenly to every node visited)

- (1) guess initial $n_{ir}(\mathbf{N}) = \mathbf{N}_r / \mathbf{K}_r$ at target state
- (2) compute back $n_{ir}(\mathbf{N}\text{-}\mathbf{1}_r) = (\mathbf{N}_r\text{-}\mathbf{1})/\mathbf{N}_r \; n_{ir}(\mathbf{N})$
- (3) use standard MVA step to compute new estimate of $n_i(\mathbf{N})$
- iterate steps (2) and (3) until convergence
  - usually 4-6 iterations enough
  - std output from last iteration

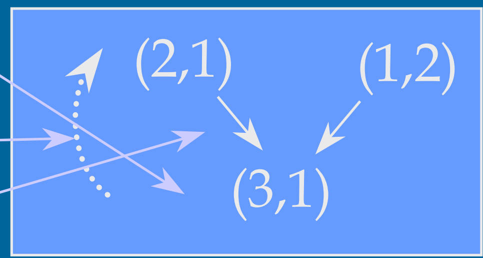



Fig. 6.5 [Men 94]

# Multiple Class Approximate MVA

- Does it converge?
  - Almost always. No quarantee.
- If it converges, does it converge to the right value?
  - It seems to do good work....
- What is good measure of convergence?
  - e.g., max relative change in $\bar{n}_{ir}(\bar{\mathbf{N}}) < 1\%$

# Example (2)

Model: Fig. 6.1 [Men 94]

step 0

$$\bar{n}_{11}(3,1) = 1.5 \qquad \bar{n}_{12}(3,1) = 1/3 = 0.333$$

$$\bar{n}_{21}(3,1) = 1.5 \qquad \bar{n}_{22}(3,1) = 0.333$$

$$\bar{n}_{31}(3,1) = 0 \qquad \bar{n}_{32}(3,1) = 0.333$$

step 1

$$R'_{11}(3,1) = D_{11}\left[1 + \bar{n}_{11}(2,1) + \bar{n}_{12}(3,0)\right]$$

$$\overset{Schweitzer}{=} 0.105\left[1 + \frac{2}{3}\bar{n}_{11}(3,1) + \frac{0}{1}\bar{n}_{12}(3,1)\right] = 0.105\left[1 + \frac{2}{3}1.5 + 0\right] = 0.210$$

$$R'_{21}(3,1) = 0.180\left[1 + \frac{2}{3}\bar{n}_{21}(3,1) + \frac{0}{1}\bar{n}_{22}(3,1)\right] = 0.180 * 2 = 0.360$$

$$R'_{31}(3,1) = 0$$

$$R'_{12}(3,1) = \ldots \qquad R'_{22}(3,1) = \ldots \qquad R'_{32}(3,1) = \ldots$$

→ Result: Tbl 6.6, Qsolver/1 output, PMVA output

= ?

# Problems with Analytical Solutions

- Load dependent servers
- Memory
- Some part of system not product form
  - however, solution method is robust
- Computational problems
  - accuracy, overflow, underflow

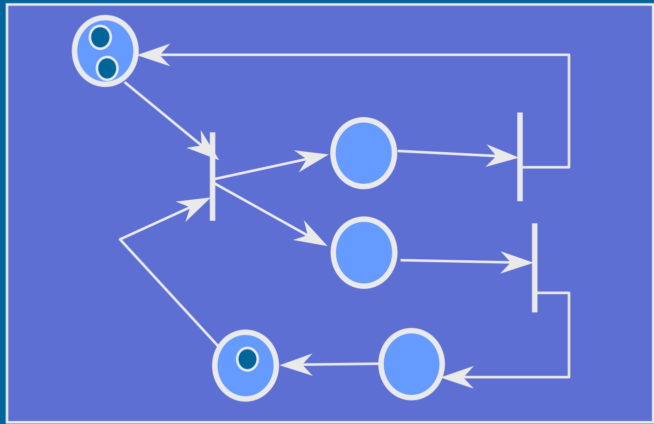
Figs 8.1, 8.3 [Men 94]

# Solution Packages (Software)

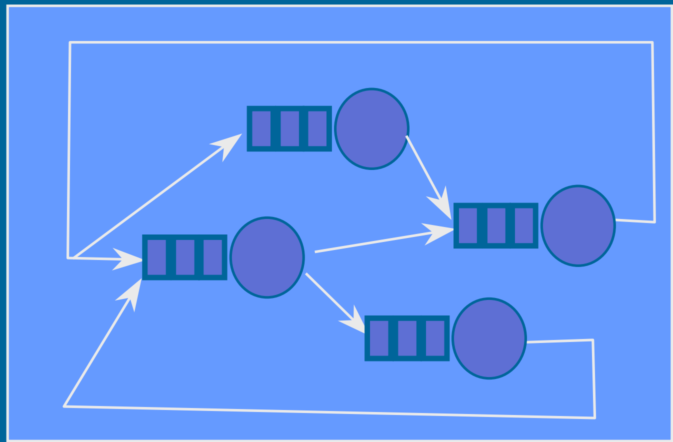
- Commercial
  - BGS, Boston, MA — Best/1
  - AT&T, Holmdale, NJ — Q+
  - SES, Austin, TX — PAWS
- University Projects
  - Jeff Brumfield, U of Austin, TX — PMVA
  - ...

# Petri Nets vs. Queueing Networks

- Petri Nets
  - good for concurrency
  - Stochastic Petri Nets (SPN)
  - Timed Petri Nets
  - simulators

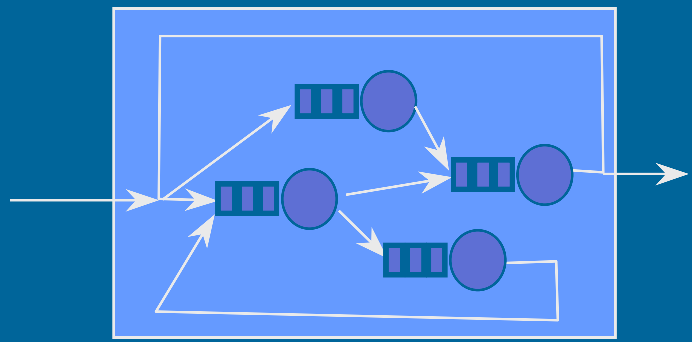


- Queuing Networks
  - good for queuing
  - simple bounds
    - Bottleneck Bounds (ABA)
    - Balanced Job Bounds (BJB)
  - exact solution methods
  - approximate solution methods
  - simulators



# Exact Solution Methods for Queuing Networks

- Open Networks



- Closed Networks
  - Convolution
  - MVA
  - Appr. MVA