MODULE 9: Queueing Networks (contd...)

LECTURE 34

Mean-Value Analysis Algorithm

Mean-Value Analysis (MVA) Algorithm

- Mean-value analysis (MVA) is an approach for calculation of expected value performance measures (mean number and mean sojourn time in each node), without actually computing the normalizing constant $G(N)$ or the state probability distribution.
- The MVA algorithm that recursively computes the performance measures by incrementing the number of customers until the desired number is reached is built on two basic principles:
  ▶ (Mean value theorem or Arrival theorem) The queue length observed by an arriving customer is the same as the general time queue length in a closed Jackson network with one less customer, i.e., $a_n(N) = p_n(N-1)$.
  ▶ Little's formula is applicable throughout the network.
- The algorithm is applicable to systems where the service times at each node are exponentially distributed.
- We also assume FCFS queueing discipline at each node.

- First principle $\Longrightarrow$
  - Recall, for $M/M/1$, $W = \dfrac{1+L}{\mu}$. That is, the average time an arriving customer must wait is the average time to serve the queue size as seen by an arriving customer plus itself.
  - For $M/M/c$ also no adjustment is needed as L is based on $a_n = p_n$.
  - The equivalent equation for our closed network, assuming that $c_i = 1 \quad \forall i$, is

$$W_i(N) = \frac{1 + L_i(N-1)}{\mu_i},$$

where

$$\begin{aligned} W_i(N) &= \text{ mean waiting time at node } i \text{ for a network with } N \text{ customers} \\ \mu_i &= \text{ mean service rate for the singer server at node } i \\ L_i(N-1) &= \text{ mean number at node } i \text{ in a network with } N-1 \text{ customers} \end{aligned}$$

- Second principle $\Longrightarrow$

$$L_i(N) = \lambda_i(N) W_i(N).$$

- If we can find $\lambda_i(N)$, then we have a method of calculating L_i and W_i starting with an empty network and building upto one with N customers $(L_i(0) = 0, W_i(1) = \dfrac{1}{\mu_i})$

To compute $\lambda_i(N)$:

- Let $D_i(N)$ represent the average delay per customer between successive visits to node i for a network with N customers. Then

$$\lambda_i(N) = \frac{N}{D_i(N)}$$

- To get $D_i(N)$: Take traffic equations, set $\nu_i = \mu_i \rho_i$. Then $\nu_i = \sum_{j=1}^{k} \nu_j r_{ji}$.
  - Set one ν_i to 1, say, $\nu_l = 1$ and solve for others.
  - Then ν_i are relative throughput through node i, i.e., $\nu_i = \dfrac{\lambda_i}{\lambda_l}$.
  - We can write $D_l(N) = \sum_{i=1}^{k} \nu_i W_i(N)$.

Mean-Value Analysis (MVA) Algorithm:

Objective: To find $L_i(N)$ and $W_i(N)$ in a k-node, single-server-per-node network with routing probability matrix $R = (r_{ij})$.

1. Solve $\nu_i = \sum_{j=1}^{k} \nu_j r_{ji}, \quad i = 1, 2, \ldots, k$, setting one of the ν_i's (say, ν_l) equal to 1.
2. Initialize $L_i(0) = 0, \quad i = 1, 2, \ldots k$
3. For $n = 1$ to N, calculate
   a) $W_i(n) = \dfrac{1 + L_i(n-1)}{\mu_i}, \qquad i = 1, 2, \ldots k$
   b) $\lambda_l(n) = \dfrac{n}{\sum_{i=1}^{k} \nu_i W_i(n)}$ (assume $\nu_l = 1$)
   c) $\lambda_i(n) = \lambda_l(n)\nu_i, \qquad i = 1, 2, \ldots k \quad i \neq l$
   d) $L_i(n) = \lambda_i(n) W_i(n) \qquad i = 1, 2, \ldots k$

Example (One-machine three-node network)

Consider the previously considered two-machines three-node network, but now with only one machine (to make it as a singer-server-at-all-nodes network).

The traffic equations are $(\nu_1, \nu_2, \nu_3) = (\nu_1, \nu_2, \nu_3) \begin{pmatrix} 0 & 3/4 & 1/4 \\ 2/3 & 0 & 1/3 \\ 1 & 0 & 0 \end{pmatrix}$.

Choosing $\nu_2 = 1$ (i.e., $l = 2$ here), we get $\nu_1 = 4/3$ and $\nu_3 = 2/3$.
Now, $i = 1, 2, 3$ and $n = N = 1$.
For step-3(a), $W_1(1) = (1 + L_1(0))/\lambda = 1/\lambda = 1/2, W_2(1) = 1/\mu_2 = 1, W_3(1) = 1/\mu_3 = 1/3$.
For step-3(b), $\lambda_2(1) = \frac{1}{\sum_{i=1}^{3} \nu_i W_i(1)} = \frac{9}{17}$.
For step-3(c), $\lambda_1(1) = \nu_1 \lambda_2(1) = 12/17, \quad \lambda_3(1) = \nu_3 \lambda_2(1) = 6/17$.
For step-3(d), $L_1(1) = \lambda_1(1) W_1(1) = 6/17, L_2(1) = 9/17, L_3(1) = 2/17$.
Since here $N = 1$, we are done.

Verification: Recall that the solution of traffic equations was $\rho_1 = 2/3, \rho_2 = 1, \rho_3 = 2/9$ and $p_{n_1,n_2,n_3} = (1/G(1))(2/3)^{n_1}(2/9)^{n_3}$. This implies that $G(1) = 17/9$ and $p_{100} = 6/17, p_{010} = 9/17, p_{001} = 2/17$ which in turn gives the above values of L_i's and W_i's.

Mean-Value Analysis (MVA) Algorithm (Multi-Server Case):

Denote the marginal probability of j at node i in an n-customer system by $p_i(j, n)$.

And, set $\alpha_i(j) = \begin{cases} j, & j \le c_i \\ c_i, & j \ge c_i \end{cases}$.

Objective: To find $L_i(N)$ and $W_i(N)$ in a k-node, c_i-servers-at-node-i network with routing probability matrix $R = (r_{ij})$.

1. Solve $\nu_i = \sum_{j=1}^{k} \nu_j r_{ji}, \quad i = 1, 2, \ldots, k$, setting one of the ν_i's (say, ν_l) equal to 1.
2. Initialize $L_i(0) = 0, p_i(0, 0) = 1, p_i(j, 0) = 0, j \neq 0, \quad i = 1, 2, \ldots k$
3. For $n = 1$ to N, calculate

   a) $W_i(n) = \dfrac{1}{c_i \mu_i} \left(1 + L_i(n-1) + \sum_{j=0}^{c_i - 2} (c_i - 1 - j) p_i(j, n-1) \right), \qquad i = 1, 2, \ldots k$

   b) $\lambda_l(n) = \dfrac{n}{\sum_{i=1}^{k} \nu_i W_i(n)}$ (assume $\nu_l = 1$)

   c) $\lambda_i(n) = \lambda_l(n) \nu_i, \qquad i = 1, 2, \ldots k \quad i \neq l$

   d) $L_i(n) = \lambda_i(n) W_i(n), \qquad i = 1, 2, \ldots k$

   e) $p_i(j, n) = \dfrac{\lambda_i(n)}{\alpha_i(j) \mu_i} p_i(j-1, n-1), \quad j = 1, 2, \ldots, n; \quad i = 1, 2, \ldots, k$

- *Exercise: Take the two-machines three-node closed network example considered earlier and apply the MVA algorithm (multi-server case). You can also carry out a verification of the obtained results.*

- Comparisons (Naive, Convolution and MVA):
  ▶ All three methods are easy for small problems
  ▶ Buzen's and MVA are computationally superior (over naive) with respect to efficiency (storage and speed) and stability, for larger problems.
  ▶ For a single-server-at-all-nodes networks, MVA is superior if we want only the mean waiting times and mean system sizes at each node.
  ▶ If we have multiserver nodes or if we want marginal probability distributions at the nodes, then it is not clear if MVA is really better than Buzen procedure.