Errata in Chapter 7

Page 402, Section 7.1.4.2 MVA Approximation for Small C

Changes marked in RED

Recall the MVA for product-form single-server closed queueing networks that we derived in Section 6.3.3. One can essentially go through that exact same algorithm with a modification in just one expression. However, for the sake of completeness we provide the entire analysis here. We first describe some notation (anything not defined here is provided prior to the bottleneck approximation). Define the following steady-state measures for all $i = 1, \ldots, N$ and $k = 0, \ldots, C$:

- $W_i(k)$: Average sojourn time in node i when there are k customers (as opposed to C) in the closed queueing network;
- $L_i(k)$: Average number in node i when there are k customers (as opposed to C) in the closed queueing network;
- $\lambda_i(k)$: Measure of average flow (sometimes also referred to as throughput) across node i ~~in the closed queueing network~~ when there are k customers (as opposed to C) in the network.

We do not have an expression for any of the above and the objective is to obtain them iteratively. However, before describing the iterative algorithm, we first explain the relationship between those parameters.

As a first approximation, we assume that the arrival theorem described in Remark 14 holds here too. Thus, in a network with k customers (such that $1 \leq k \leq C$) the expected number of customers that an arrival to node i (for any $i \in \{1, \ldots, N\}$) would see is $L_i(k-1)$. Note that $L_i(k-1)$ is the steady state expected number of customers in node i when there are $k-1$ customers in the system. Further, the net mean sojourn time experienced by that arriving customer in steady state is the average time to serve all those in the system upon arrival plus that of the customer. Note that the average service time is $1/\mu_i$ for all customers waiting and $(1+C_{S_i}^2)/(2\mu_i)$ for the customer in service (using the remaining time for an event in steady state for a renewal process). Thus we have

$$W_i(k) = \frac{1}{\mu_i}\left[\frac{1+C_{S_i}^2}{2} + L_i(k-1)\right].$$

The above expression is a gross approximation because it assumes there is always a customer at the server at an arrival epoch which is not true. However, since the server utilization is not known, we are unable to characterize the remaining service times more accurately and live with the above approximation hoping it would be conservative.

Let $v = [v_j]$ be a row vector which is the solution to $v = vP$ and that the v_j values sum to one. It is identical to the visit ratios in the *bottleneck* approximation given above. ~~The aggregate sojourn time weighted across the network using the visit ratios is given by $\sum_{i=1}^{N} v_i W_i(k)$ when there are k customers in the network.~~ Thereby we derive the average flow in the network using Little' law across the entire network as

$$\lambda_i(k) = \frac{kv_i}{\sum_{j=1}^{N} v_j W_j(k)}$$

when there are k customers in the network. Thereby applying Little's law across each node i we get

$$L_i(k) = \lambda_i(k) W_i(k)$$

when there are k customers in the network.

Using the above results we can develop an algorithm to determine $L_i(C)$, $W_i(C)$ and $\lambda_i(C)$ defined above. The input to the algorithm are N, C, P μ_i and $C_{S_i}^2$ for all $i \in \{1, \ldots, N\}$. For the algorithm, initialize $L_i(0) = 0$ for $1 \leq i \leq N$ and obtain the v_i values for all $i \in \{1, \ldots, N\}$. Then for $k = 1$ to C, iteratively compute for each i (such that $1 \leq i \leq N$):

$$\begin{aligned} W_i(k) &= \frac{1}{\mu_i}\left[\frac{1 + C_{S_i}^2}{2} + L_i(k-1)\right], \\ \lambda_i(k) &= \frac{kv_i}{\sum_{j=1}^{N} v_j W_j(k)}, \\ L_i(k) &= \lambda_i(k) W_i(k). \end{aligned}$$

Page 406, Section 7.1.4.2.2 Solution to Problem 68: MVA Approx.
Changes marked in RED

For the algorithm we initialized $L_i(0) = 0$ for $1 \leq i \leq 5$. We used the v_i values for all $i \in \{1, \ldots, 5\}$. Then for $k = 1$ to 30, iteratively compute for each i (such that $1 \leq i \leq 5$):

$$\begin{aligned} W_i(k) &= \frac{1}{\mu_i}\left[\frac{1 + C_{S_i}^2}{2} + L_i(k-1)\right], \\ \lambda_i(k) &= \frac{kv_i}{\sum_{j=1}^{5} v_j W_j(k)}, \\ L_i(k) &= \lambda_i(k) W_i(k). \end{aligned}$$

The approximation can be used to get L_i as $L_i(30)$. Writing a computer program we get $L_1 = 11.0655$, $L_2 = 0.4863$, $L_3 = 2.3016$, $L_4 = 15.8842$ and $L_5 = 0.2625$ with $\lambda_1(30) = 0.2736$ which is fairly close to the a_1 ~~λ~~ obtained in the bottleneck approximation. Notice that the L_i values are relatively close. Upon running simulations with service times according to an appropriate gamma distribution, we get $L_1 = 5.5803(\pm 0.0250)$, $L_2 = 0.7811(\pm 0.0010)$, $L_3 = 3.6819(\pm 0.0119)$, $L_4 = 19.7360(\pm 0.0333)$ and $L_5 = 0.2205(\pm 0.0002)$ with the numbers in the brackets denoting width of 95% confidence intervals based on 100 replications.

Page 416, Step 2 of QNA Algorithm

There should be no $\frac{\lambda_{i,r}}{\lambda_i}$ in the splitting equation. Thus the equation should be

$$C_{ij,r}^2 = 1 + \frac{\lambda_{i,r}}{\lambda_i} p_{ij,r}(C_{D_i}^2 - 1).$$

The above is identical to that in pages 414 and 420.

Thus some of the results in Problem 70 would be different from what is published.

Page 417, Figure 7.9

Arc from node 3 to node 5 is missing.