Theoretical Foundations: Requirements for a separable queuing network

- service center flow balance: at each service center, the number of arrivals equals the number of completions

- one step behavior: no two jobs in the system change state simultaneously

- routing homogeneity: the proportion of the time that a job completing service at center j proceeds to center k is independent of the current queue lengths at any of the centers (routing patterns do not affect performance in separable models)
- device homogeneity: the completion rate at a service center may vary with the number of jobs at that center, but may not depend on the number or placement of jobs in the network otherwise.

- homogeneous external arrivals: the arrival times must not depend on the number or placement of customers within the network

Algorithms So Far Assume:

- service time homogeneity: the rate of completions from a service center while busy, must be independent of the number of customers at that center

(7c load independent)

The MVA algorithms in this chapter apply only to networks consisting entirely of load independent centers and delay centers.

These assumptions are necessary for the solution using exact MVA to be the exact solution for the model.

Experience has shown that they need not be true in practice for a separable model to provide useful information.

Chapters 8 and 20 discuss the method of flow equivalent service centers, FESCs, in extending analysis to load dependent service centers. Essentially, at the stage in which the R_k's are computed, use a mean of the residence time calculated for different queue lengths and weighted for their probabilities. Using this, calculate the probable queue lengths for a system with one more job.

Chapter 7: Models with Multiple Job Classes

Advantages: greater accuracy for systems having jobs with significantly different patterns of demands, greater detail of output information

Disadvantages: more difficult to parametrize, more complex to implement

Queuing discipline becomes relevant with multiple job classes. The techniques here apply to several queuing disciplines, all class-blind: first come, first served (FCFS), processor sharing in which n jobs receive $\frac{1}{n}$ of the full power (PS); last come, first served pre-emptive resume (LCFS) in which the arriving job preempts service, which resumes with the most recently preempted job; and delay. The first three have identical performance in these models.

Open model results are very similar to single class results

processing capacity: $\max_k \left\{ \sum_{c=1}^{C} \lambda_c D_{c,k} \right\} < 1$

throughput $X_c(\vec{\lambda}) = \lambda_c$

utilization $U_{c,k}(\vec{\lambda}) = \lambda_c D_{c,k}$

residence time: $R_{c,k}(\vec{\lambda}) = \begin{cases} D_{c,k} & \text{delay} \\ \dfrac{D_{c,k}}{1 - \sum_{j=1}^{c} U_{j,k}(\vec{\lambda})} & \end{cases}$

queue length: $Q_{c,k}(\vec{\lambda}) = \lambda_c R_{c,k}(\vec{\lambda})$

system response time $R_c(\lambda) = \sum_{k=1}^{K} R_{c,k}(\vec{\lambda})$

average numbr in system $Q_c(\vec{\lambda}) = \lambda_c R_c(\vec{\lambda})$

Closed Model:

$A_{c,k}(\vec{N})$, the average arrival instant queue length seen at center k by an arriving class c customer, is equal to $Q_k(\overrightarrow{N-1_c})$ where $\overrightarrow{N-1_c}$ is the population vector $\vec{N}$ with the component corresponding to class c reduced by one.

$$A_{c,k}(\vec{N}) = Q_k(\overrightarrow{N-1_c})$$

$$R_{c,k}(\vec{N}) = \begin{cases} D_{c,k} & \text{delay centers} \\ D_{c,k}[1 + A_{c,k}(\vec{N})] & * \end{cases}$$

$$X_c(\vec{N}) = \frac{N_c}{Z_c + \sum_{k=1}^{K} R_{c,k}(\vec{N})}$$

$$Q_{c,k}(\vec{N}) = X_c(\vec{N}) R_{c,k}(\vec{N})$$

$$Q_k(\vec{N}) = \sum_c Q_{c,k}(\vec{N})$$

* for queuing centers,

$$R_{c,k}(\vec{N}) = V_{c,k} * (S_k (1 + A_{c,k}(\vec{N})))$$
$$= D_{c,k} * (1 + A_{c,k}(\vec{N})),$$

set up an iterative solution as before, except with an additional loop for c at each $\vec{n}$, and greater computational complexity.

To calculate $Q_{c,k}(\vec{N})$, $X_c(\vec{N})$ and $R_{c,k}(\vec{N})$,

$$Q_{j,l}(\vec{n}),\ X_j(\vec{n}) \text{ and } R_{j,l}(\vec{n})$$

must be computed for all $1 \le j \le C$, $1 \le l \le K$ and $\vec{n}$ with $n_i \le N_i$ for all i.

time complexity $CK \prod_{c=1}^{C} (N_c + 1)$

space complexity $K \prod_{\substack{c=1 \\ c \neq c_{max}}}^{C} (N_c + 1)$

Ex. to evaluate $\vec{N} = (1,2,1)$ need

(1,2,1)

(1,1,1) (1,2,0) (0,2,1)

(1,1,0) (0,2,0) (0,1,1)

(1,0,0) (0,1,0) (0,0,1)

(0,0,0)

The approximate MVA technique refines approximations for $Q_{c,k}(\vec{N})$ as before, but

$Q_k(\overrightarrow{N-1_c})$ is often approximated by

$$h_c[Q_{1,k}(\vec{N}), \dots Q_{C,k}(\vec{N})] = \frac{N_c - 1}{N_c} Q_{c,k}(\vec{N}) + \sum_{\substack{j=1 \\ j \neq c}}^{C} Q_{j,k}(\vec{N}).$$

Exact Mixed Model Technique:

$\{O\}$ = set of open classes, $\{C\}$ = set of closed classes

1. Find utilization of each center by open classes

$$U_{\{O\},k}(\vec{I}) = \sum_{c \in \{O\}} U_{c,k}$$

using the forced flow law and the open classes' demands.

2. Solve the closed model using the K centers and just the classes in $\{C\}$ using demands adjusted for the availability of the centers after the time required per unit time for the open classes is removed:

$$D^*_{c,k} = \frac{D_{c,k}}{1 - U_{\{O\},k}(\vec{I})}$$

3. For $c \in \{O\}$

$$R_{c,k}(\vec{I}) = \frac{D_{c,k}\left[1 + Q_{\{C\},k}(\vec{I})\right]}{1 - U_{\{O\},k}(\vec{I})}$$

$$Q_{c,k}(\vec{I}) = \lambda_c R_{c,k}(\vec{I})$$