

Chapter 2

Modeling Automated Warehouses Using Semi-Open Queueing Networks

Xiao Cai, Sunderesh S. Heragu, and Yang Liu

2.1 Introduction

A typical warehouse consists of three areas: reserve area, forward area, and cross-dock area. The reserve area is a high-density, narrow-aisle storage area with unit loads (pallets, totes or bins) stored on racks that extend from floor to ceiling and wall to wall. Because the aisles are narrow to maximize storage density, full pallet loads are typically handled in the reserve area and the throughput is high, this area of a warehouse is typically automated. Two main types of automated material handling technologies have been used in the reserve area of a warehouse. One of them, the automated storage and retrieval system (AS/RS), has been widely used for decades. The other technology, the autonomous vehicle storage and retrieval system (AVS/RS), is relatively new and has been installed in over fifty warehouses in Europe and other parts of the world.

The AS/RS consists of narrow aisles with storage racks usually located on both sides of the aisle. A storage/retrieval (S/R) crane capable of handling one or two unit loads traverses the entire depth and height of the aisle to store or retrieve units loads in or from their respective storage locations. Each crane is mounted on a mast and there are two sets of motors, one driving the crane up and down a mast and another moving the mast in and out of an aisle. This allows a crane to access any storage location anywhere on the rack. The crane-mast system can be designed so they are aisle-captive or can move from aisle to aisle. The aisle-captive designs are more

X. Cai
FedEx Corporation, Memphis, TN, USA
e-mail: xiao.cai@fedex.com

S.S. Heragu (✉)
University of Louisville, Louisville, KY, USA
e-mail: s.heragu@louisville.edu

Y. Liu
Chrysler Group LLC, Belvidere, IL, USA
e-mail: y.liu@chrysler.com

common. After all, the primary reason warehouse managers invest in automation is due to the high levels of throughput requirement they face and thus minimizing the number of cranes by making them travel aisle to aisle could reduce the throughput that might be achieved by aisle-captive systems. An example of an AS/RS is provided in Fig. 2.1.

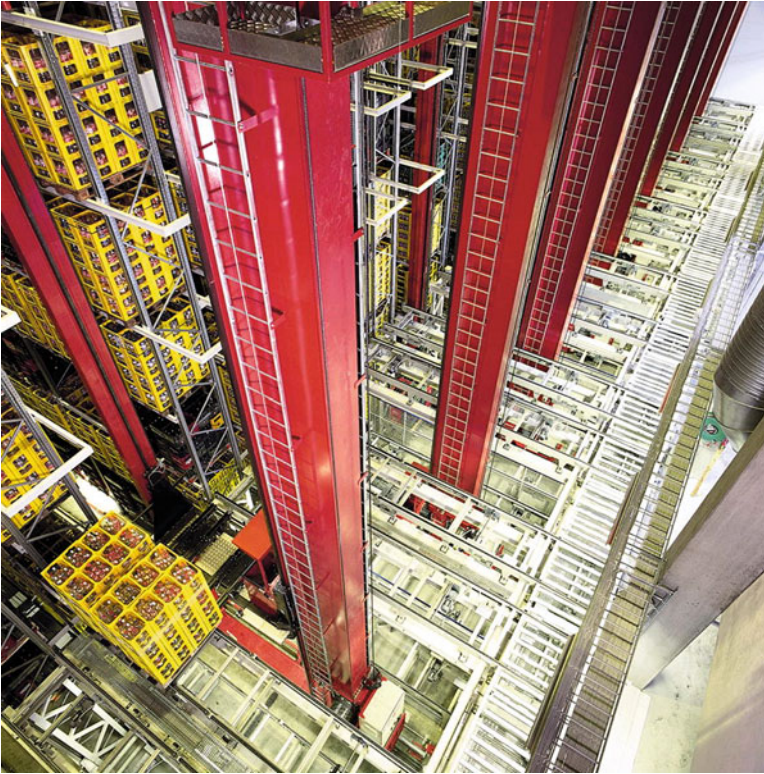


Fig. 2.1 An example of the AS/RS

An AVS/RS is an alternative automated material handling system also used in the reserve area of warehouse, but it uses a combination of autonomous vehicles to move pallets within a tier and lifts to transport empty or loaded vehicles between tiers (see Fig. 2.2). As shown in Fig. 2.3, the autonomous vehicle has two sets of motors one for travel in the x-dimension and another for the y-dimension.

If the pallet is to be stored in a floor other than the one where it was picked up (which typically is on the ground floor), the autonomous vehicle travels to the nearest elevator with the pallet load and summons a lift. When the lift arrives, the vehicle and pallet are transported to the destination tier. Once at the tier, the vehicle travels to the specific storage location to store the inbound pallet load. If the pallet load is to be stored on a rack located on the ground floor, lift travel is not necessary.

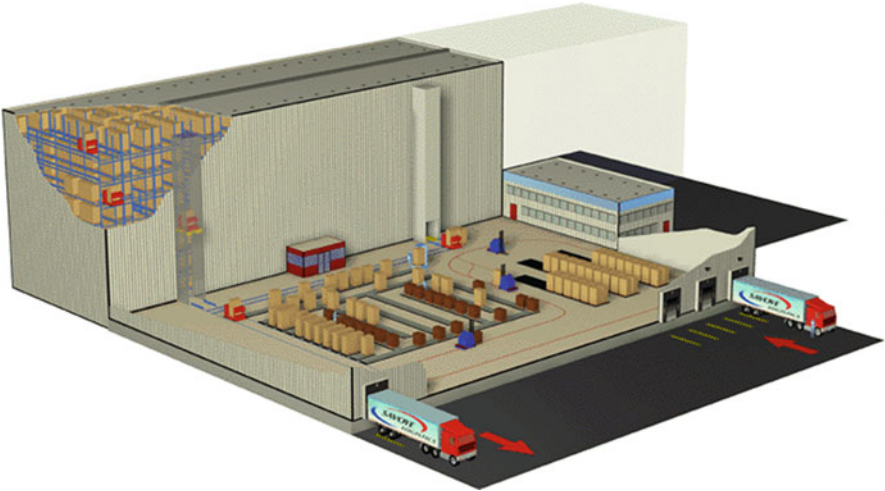


Fig. 2.2 An example of the AVS/RS

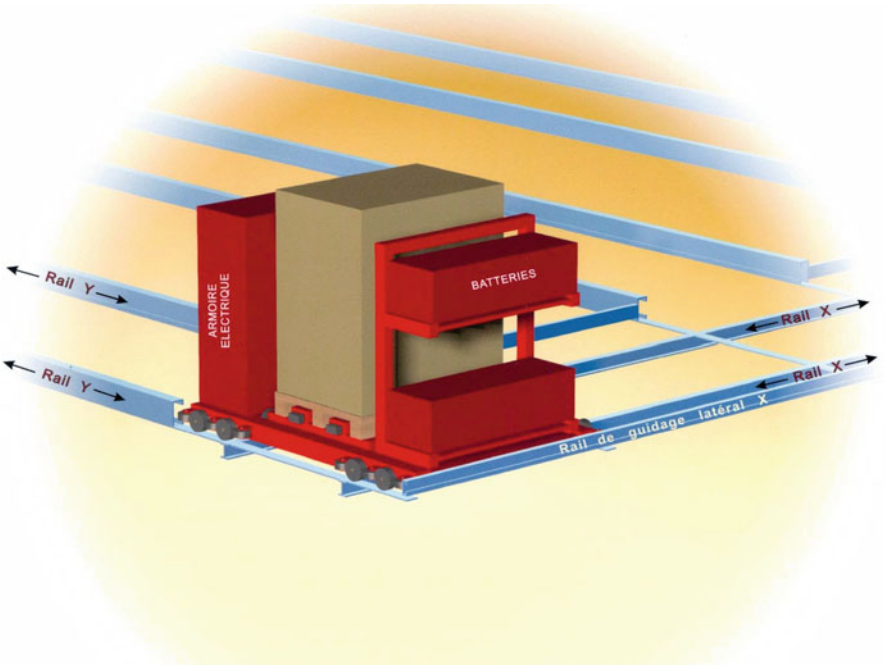


Fig. 2.3 Autonomous vehicle with two sets of motors

In a retrieval transaction, outbound pallets are retrieved from their storage locations and transported one at a time using a vehicle-lift combination as done in a storage transaction, but in reverse order.

Modeling systems in which an incoming customer must be paired with another resource and the two must stay together until service to the customer is completed, as an open queuing network (OQN) or closed queuing network (CQN) leads to an underestimation of the true sojourn time of the customer. The reason for this is that the OQN implicitly assumes there is an infinite number of the secondary resource and so an incoming customer never has to wait outside the system. A CQN, on the other hand, assumes there is an infinite number of customers waiting externally and so a resource assigned to a customer who has just completed service immediately reenters the system with a new customer. In reality, a customer must sometimes wait for a resource or vice-versa. For these systems, a semi-open queuing network is a preferred model because it captures the fact that sometimes a customer must wait for a resource or a resource must wait for a customer. In the AVS/RS considered in our paper, a storage or retrieval transaction is a customer and the autonomous vehicle is the secondary resource. Because we have a finite number of each, it is important to model the AVS/RS as an SOQN in which a vehicle (secondary resource) must be synchronized or paired with a storage or retrieval transaction and stay with it until the transaction is completed.

An SOQN represents a queueing network with an additional resource. Initially, all the resources wait in a resource queue. A new customer is required to be synchronized or paired with a resource before entering the service network. If there is no resource available, the customer has to wait in an external customer queue until a resource becomes available. Once the customer is synchronized or paired with a resource, the service process begins. When the customer exits the network, the resource associated with this customer returns to the resource queue and waits for the next customer. A general SOQN is shown in Fig. 2.4.

In this chapter, we model the AVS/RS as an SOQN and propose two efficient algorithms based on a state space method and the matrix geometric method (MGM), to evaluate the performance of the AVS/RS. A set of steady-state results can be obtained for semi-open queuing networks via an approximate, but tractable method via the use of Norton's theorem. This successful application of MGM is due to the unique lack-of-memory property of the exponential distribution. However, assuming such exponential distributions on the inter-arrival and service times does not reflect many real world scenarios. On the other hand, analyzing an SOQN with general inter-arrival and service times through simulation is very time-consuming. A compromise is to develop a method that approximates general distributions so that the MGM can still be applied. To that end, we use Phase-type distributions to approximate the general distribution and utilize the MGM to solve the general SOQN problems (see [7–9]).

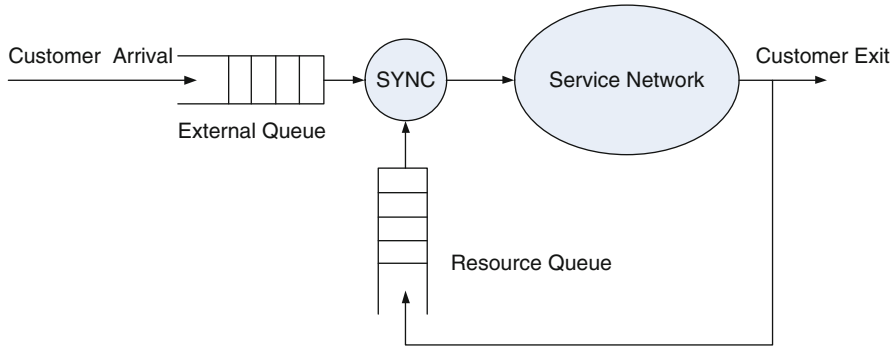


Fig. 2.4 A general SOQN

2.2 SOQN Notation

The main parameters and system performance measures of the AVS/RS used in this chapter are:

- S : number of service stages in the network
- V : number of vehicles (resources) in the system
- λ : overall external customer arrival rate
- μ_j : service rate of j th service stage, $j = 1, \dots, S$
- L_{eq} : average number of customers waiting in the external queue
- L_{pq} : average number of vehicles in the vehicle queue
- L_j : average number of customers at j th service stage, $j = 1, \dots, S$
- L_n : average number of customers in the network
- W_s : average waiting time per customer in the system

In this chapter, we assume the number of vehicles is known and the route of customers is fixed. The service rate of each server is also assumed to be known and the same for all customers.

2.3 Single-Class SOQN with Two Stages of Exponential Servers and Poisson Arrivals

2.3.1 State Space Solution

Figure 2.5 shows a two-stage, single-class SOQN with exponential servers and interarrival times. The state (i, j) denotes that there are a total of i customers in the external queue and the first server, and j customers in the second service stage. The state space S_s is the infinite set $\{(0, 0), (0, 1), \dots, (0, V), (1, 0), (1, 1), \dots\}$, and each state s_m in S_s is:

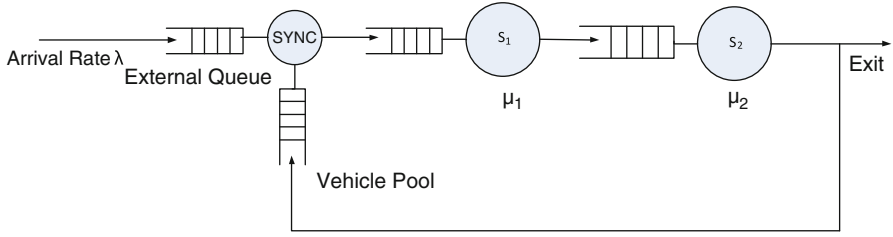


Fig. 2.5 Single-class, two-stage SOQN

$$s_m = (i, j), \text{ where } i \geq 0, 0 \leq j \leq V, \text{ and } m = i(V + 1) + j.$$

Figure 2.6 shows the state space of this SOQN.

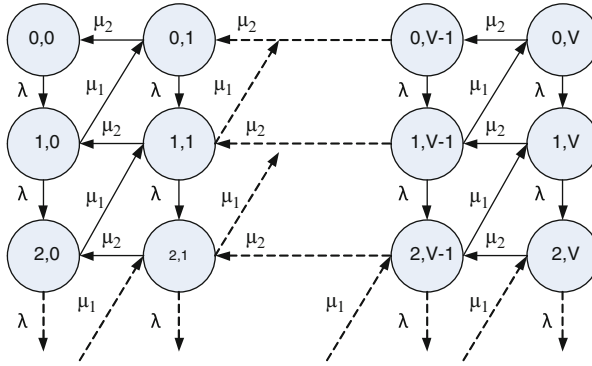


Fig. 2.6 The state space of single-class, two-stage SOQN with two variables

This two-stage, single-class SOQN with exponential servers and Poisson arrivals is a continuous-time Markov chain (CTMC) process, which means the conditional probability mass function (pmf) of this process satisfies:

$$p_{mn}(t) = P\{X(s+t) = s_m | X(s) = s_n\}, \forall s, t > 0, \text{ and } s_m, s_n \in S_s. \quad (2.1)$$

Here $p_{mn}(t)$ is the transition probability from state s_m to state s_n at time t and $\sum_{s_n \in S_s} p_{mn} = 1$. The p_{mn} s are usually summarized in a nonnegative transition matrix $\mathbf{P}(t)$:

$$\mathbf{P}(t) = [p_{mn}(t)] = \begin{bmatrix} p_{00}(t) & p_{01}(t) & p_{02}(t) & \cdots \\ p_{10}(t) & p_{11}(t) & p_{12}(t) & \cdots \\ p_{20}(t) & p_{21}(t) & p_{22}(t) & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}.$$

The unconditional state probability $\pi_n(t)$ can be expressed by $p_{mn}(t)$ and the initial condition $\pi_m(0)$:

$$\pi_n(t) = \sum_{s_m \in S_s} p_{mn}(t) \pi_m(0), \quad (2.2)$$

or

$$\pi(t) = \pi(0)\mathbf{P}(t), \quad (2.3)$$

where $\pi(t) = [\pi_0(t), \pi_1(t), \dots]$.

The main result of homogeneous CTMCs is Kolmogorov's forward differential equation:

$$p'_{mn}(t) = \sum_{s_k \in S_s} p_{mk}(t) q_{kn}, \quad (2.4)$$

where $q_{mn}(t)$ is the instantaneous transition rate. The definition of q_{mn} is:

$$q_{mn}(t) = \begin{cases} \lim_{\Delta t \rightarrow 0} \frac{p_{mn}(t, t + \Delta t) - p_{mn}(t, t)}{\Delta t} & m \neq n, \\ \lim_{\Delta t \rightarrow 0} \frac{p_{mm}(t, t + \Delta t) - 1}{\Delta t} & \text{otherwise.} \end{cases} \quad (2.5)$$

For example, from state $s_0(0, 0)$ to state $s_V(1, 0)$, q_{0V} denotes the arrival process of a customer, so $q_{0V} = \lambda$. Since s_0 can only arrive to s_V , the value of q_{00} can be calculated as:

$$\begin{aligned} q_{00}(t) &= \lim_{\Delta t \rightarrow 0} \frac{p_{00}(t + \Delta t) - 1}{\Delta t} \\ &= \lim_{\Delta t \rightarrow 0} \frac{1 - \sum_{s_n \in S_s} p_{0n}(t + \Delta t) - 1}{\Delta t} \\ &= \lim_{\Delta t \rightarrow 0} \frac{-\sum_{s_n \in S_s} p_{0n}(t + \Delta t)}{\Delta t} \\ &= \lim_{\Delta t \rightarrow 0} \frac{-p_{0V}(t + \Delta t)}{\Delta t} \\ &= -q_{0V} = -\lambda. \end{aligned}$$

We combine (2.2) and (2.4):

$$\dot{\pi}(t) = \pi(t)\mathbf{Q}, \quad (2.6)$$

where the matrix \mathbf{Q} is:

$$\mathbf{Q} = [q_{mn}], \forall s_m, s_n \in S_s. \quad (2.7)$$

For example, the \mathbf{Q} of the SOQN with two vehicles is:

$$\mathbf{Q} = \begin{bmatrix} -\lambda & 0 & 0 & \lambda & 0 & 0 & 0 & \cdots \\ \mu_2 & -(\mu_2 + \lambda) & 0 & 0 & \lambda & 0 & 0 & \cdots \\ 0 & \mu_2 & -(\mu_2 + \lambda) & 0 & 0 & \lambda & 0 & \cdots \\ 0 & \mu_1 & 0 & -(\mu_1 + \lambda) & 0 & 0 & \lambda & \cdots \\ 0 & 0 & \mu_1 & \mu_2 & -(\mu_1 + \mu_2 + \lambda) & 0 & 0 & \cdots \\ 0 & 0 & 0 & 0 & \mu_2 & -(\mu_2 + \lambda) & 0 & \cdots \\ 0 & 0 & 0 & 0 & \mu_1 & 0 & -(\mu_1 + \lambda) & \cdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \mu_2 & \cdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

If the unconditional steady state π of the CTMC exists, it should be independent of time:

$$\lim_{t \rightarrow \infty} \dot{\pi}(t) = 0.$$

Finally,

$$\pi \mathbf{Q} = \mathbf{0}. \quad (2.8)$$

Additionally, the normalization condition holds:

$$\pi \mathbf{1} = 1. \quad (2.9)$$

Since the state space of SOQN is infinite, there is no closed form expression for this stochastic process. An alternative method is to truncate the state space at a certain level k to obtain an approximate solution.

Algorithm based on State Space

$\pi_V(0) = 0.5, \pi(0) = [0, \dots, \pi_V(0), \dots, 0]_{1 \times k(V+1)};$

$\pi(1) = \pi(0) \mathbf{Q}_{k(V+1) \times k(V+1)};$

$n = 0;$

while $|\pi_V(n+1) - \pi_V(n)| \geq \varepsilon$

$n++;$

$\pi(n+1) = \pi(n) \mathbf{Q}_{k(V+1) \times k(V+1)};$

end

$\pi = \pi(n+1);$

$\pi_m = \frac{\pi_m}{\sum \pi_m}.$

The performance measures can be obtained directly from these unconditional state probabilities ((2.10)–(2.15)).

$$L_{eq} = \sum_{i=0}^k \sum_{j=\max(0, V+1-j)}^V (i+j-V) \pi_{i(V+1)+j} \quad (2.10)$$

$$L_1 = \sum_{i=0}^k \sum_{j=0}^V L_{ij}, \quad \text{where } L_{ij} = \begin{cases} i\pi_{i(V+1)+j} & \text{if } i+j \leq V \\ (V-j)\pi_{i(V+1)+j} & \text{otherwise} \end{cases} \quad (2.11)$$

$$L_2 = \sum_{i=0}^k \sum_{j=0}^V j\pi_{i(V+1)+j} \quad (2.12)$$

$$L_n = L_1 + L_2 \quad (2.13)$$

$$L_{pq} = V - L_n \quad (2.14)$$

$$W_s = \frac{L_n + L_{eq}}{\lambda}. \quad (2.15)$$

2.3.2 Matrix Geometric Method Solution

In the method described in the previous section, it is rather difficult to determine the unconditional stationary state probabilities of a Markov process with infinite number of states in a closed form solution. However, if the state space of a Markov process can be expressed by a repetitive structure, the unconditional stationary state probabilities can be obtained exactly. The unconditional stationary state probabilities of this repetitive structure thus has a geometric form. Neuts [12] developed a body of results of this repetitive structure that is called matrix geometric form. We develop an algorithm based on this MGM to solve the two-stage, single-class SOQN with exponential servers and Poisson arrivals.

First, we construct a state space of this SOQN with three parameters. The first parameter is the number of customers waiting in the external queue i . The second parameter is the number of customers j at the first server and the last parameter is the number of customers k at the second server.

$$s_m = (i, j, k) \text{ where } i, j, k \geq 0, (j+k) \leq V,$$

$$m = \begin{cases} \frac{(j+k)(j+k+1)}{2} + k, & \text{if } i = 0, \\ i(V+1) + \frac{(N+1)N}{2} + k, & \text{otherwise.} \end{cases}$$

The instantaneous transition rates matrix \mathbf{Q} is obtained by (2.5). Figure 2.7 shows the state space which can be used to construct the matrix \mathbf{Q} .

Next, we observe the behavior of this Markov process and find the following properties:

1. If $i \geq 1, j+k = V$. This property means that all vehicles are busy if there are customers waiting outside.

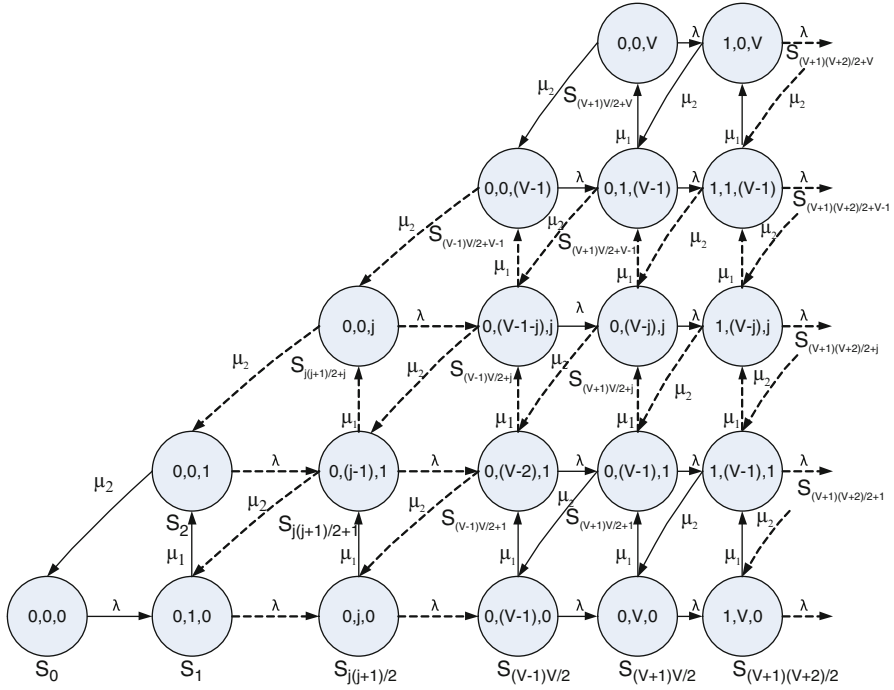


Fig. 2.7 The state space of single-class, two-stage SOQN with three variables

2. It is impossible to travel from state (i, j, k) to (i', j, k) when $|i - i'| \geq 2$. Obviously, during an infinitesimal time interval $[t, t + \Delta t]$, only one customer enters or exits the system.

3. In \mathbf{Q} , q_{mn} s are independent of i .

Since this Markov process satisfies these properties, it is a continuous time, irreducible, homogeneous quasi-birth-death (QBD) process. The original problem now is treated as determining the unconditional stationary state probabilities of a QBD process. In a QBD process, the number of customers in the external queue i is the i th level, and number of customers at each service stage (j, k) is the phase (j, k) . According to this, we denote π_i as the vector of unconditional stationary state probabilities of all phases at the i th level. This QBD has a repetitive structure of \mathbf{Q} like this:

$$\mathbf{Q} = \begin{bmatrix} \mathbf{B}_{00} & \mathbf{B}_{01} & \mathbf{0} & \mathbf{0} & \cdots \\ \mathbf{B}_{10} & \mathbf{A}_1 & \mathbf{A}_0 & \mathbf{0} & \cdots \\ \mathbf{0} & \mathbf{A}_2 & \mathbf{A}_1 & \mathbf{A}_0 & \cdots \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_2 & \mathbf{A}_1 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}, \quad (2.16)$$

where \mathbf{B}_{00} , \mathbf{B}_{01} and \mathbf{B}_{10} are instantaneous transition matrixes to determine the initial state of the system.

\mathbf{B}_{00} denotes the transition rates from level 0 to level 0:

$$\mathbf{B}_{00} = \begin{bmatrix} -\lambda & \lambda & 0 & & & & & & & & \\ 0 & -(\mu_1 + \lambda) & \mu_1 & \lambda & 0 & & & & & & \\ \mu_2 & 0 & -(\mu_2 + \lambda) & 0 & \lambda & & & & & & \\ \vdots & & \ddots & \ddots & \ddots & & & & & & \\ & & & 0 & \cdots & 0 & 0 & -(\mu_1 + \lambda) & \mu_1 & 0 & \cdots & 0 \\ & & & \mu_2 & & & & -(\mu_1 + \mu_2 + \lambda) & \mu_1 & & & \\ & & & & \ddots & & & & \ddots & & & \vdots \\ & & & & & \mu_2 & & & & -(\mu_1 + \mu_2 + \lambda) & \mu_1 & \\ & & & & & & \mu_2 & & & & -(\mu_2 + \lambda) \end{bmatrix}$$

\mathbf{B}_{01} denotes the transition rates from level 0 to level 1:

$$\mathbf{B}_{01} = \begin{bmatrix} 0 & \cdots & 0 \\ \lambda & & \\ & \ddots & \\ & & \lambda \end{bmatrix}.$$

\mathbf{B}_{10} denotes the transition rates from level 1 to level 0:

$$\mathbf{B}_{10} = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ 0 & \mu_2 & & & \vdots \\ \vdots & & \ddots & & \vdots \\ 0 & & & \mu_2 & 0 \end{bmatrix}.$$

The repetitive structure includes \mathbf{A}_0 , \mathbf{A}_1 and \mathbf{A}_2 .

$$\mathbf{A}_0 = \begin{bmatrix} \lambda & & & \\ & \lambda & & \\ & & \ddots & \\ & & & \lambda \end{bmatrix}_{(V+1) \times (V+1)},$$

$$\mathbf{A}_1 = \begin{bmatrix} -(\mu_1 + \lambda) & & \mu_1 & & \\ & -(\mu_1 + \mu_2 + \lambda) & \mu_1 & & \\ & & \ddots & \ddots & \\ & & & -(\mu_1 + \mu_2 + \lambda) & \mu_1 \\ & & & & -(\mu_2 + \lambda) \end{bmatrix}_{(V+1) \times (V+1)},$$

$$\mathbf{A}_2 = \begin{bmatrix} 0 & \cdots & \cdots & 0 \\ \mu_2 & & & \vdots \\ & \ddots & & \vdots \\ & & \mu_2 & 0 \end{bmatrix}_{(V+1) \times (V+1)}.$$

According to (2.8), the following repetitive balance equation holds:

$$\pi_{i-1} \mathbf{A}_0 + \pi_i \mathbf{A}_1 + \pi_{i+1} \mathbf{A}_2 = \mathbf{0}, \quad i \geq 2. \quad (2.17)$$

The QBD has an important property described in Theorem 2.1 (Proof can be found in [13]).

Theorem 2.1. *If the QBD is positive recurrent ($\pi_i \mathbf{A}_0 \mathbf{e} < \pi_i \mathbf{A}_2 \mathbf{e}$), then*

$$\pi_{i+1} = \pi_i \mathbf{R} \text{ for } i \geq 1, \quad (2.18)$$

or

$$\pi_i = \pi_1 \mathbf{R}^{i-1} \text{ for } i \geq 1, \quad (2.19)$$

where \mathbf{R} is a rate matrix.

Substituting (2.18) into (2.17) and simplifying yields

$$\mathbf{A}_0 + \mathbf{R} \mathbf{A}_1 + \mathbf{R}^2 \mathbf{A}_2 = \mathbf{0}. \quad (2.20)$$

If we can get \mathbf{R} and π_1 , we can get all π_i . A simple heuristic procedure is applied to get \mathbf{R} . First, (2.20) can be written as

$$\mathbf{R} = -(\mathbf{A}_0 + \mathbf{R}^2 \mathbf{A}_2) \mathbf{A}_1^{-1}. \quad (2.21)$$

Then, the procedure to obtain \mathbf{R} is:

```

R0 = 0
R1 = -(A0 + R02A2)A1-1
k = 0
while ||Rk+1|| - ||Rk|| > ε
    k++;
    Rk+1 = -(A0 + Rk2A2)A1-1;
end
R = Rk.

```

π_1 can be obtained from the boundary part of the balance equations (2.8):

$$\begin{cases} \pi_1 \mathbf{B}_{00} + \pi_1 \mathbf{B}_{10} = \mathbf{0}, \\ \pi_0 \mathbf{B}_{01} + \pi_1 \mathbf{A}_1 + \pi_2 \mathbf{A}_2 = \mathbf{0}. \end{cases} \quad (2.22)$$

$$\pi_2 = \pi_1 \mathbf{R}.$$

Substituting this fact into (2.22) and simplifying in matrix form, we get:

$$\begin{bmatrix} \pi_0 & \pi_1 \end{bmatrix} \begin{bmatrix} \mathbf{B}_{00} & \mathbf{B}_{01} \\ \mathbf{B}_{10} & \mathbf{A}_1 + \mathbf{R} \mathbf{A}_2 \end{bmatrix} = \mathbf{0}. \quad (2.23)$$

Since the coefficient matrix is not full rank, equation (2.23) is not sufficient to determine the values of π_0 and π_1 . We can use the normalization condition (2.9) to determine these values:

$$1 = \pi_0 \mathbf{e} + \pi_1 \sum_{i=1}^{\infty} \mathbf{R}^{i-1} \mathbf{e} = \pi_0 + \pi_1 (\mathbf{I} - \mathbf{R})^{-1} \mathbf{e}. \quad (2.24)$$

Adding (2.24) to (2.23), we get:

$$[\pi_0 \ \pi_1] \begin{bmatrix} \mathbf{e} & \mathbf{B}_{00} & \mathbf{B}_{01} \\ (\mathbf{I} - \mathbf{R})^{-1} \mathbf{e} & \mathbf{B}_{10} & \mathbf{A}_1 + \mathbf{R} \mathbf{A}_2 \end{bmatrix} = [1 \ 0], \quad (2.25)$$

or

$$[\pi_0 \ \pi_1] = [1 \ 0] / \begin{bmatrix} \mathbf{e} & \mathbf{B}_{00} & \mathbf{B}_{01} \\ (\mathbf{I} - \mathbf{R})^{-1} \mathbf{e} & \mathbf{B}_{10} & \mathbf{A}_1 + \mathbf{R} \mathbf{A}_2 \end{bmatrix}. \quad (2.26)$$

The performance measures can be obtained from these unconditional stationary state probabilities :

$$L_{eq} = \sum_{i=1}^{\infty} i \pi_i \mathbf{e} = \pi_1 (\mathbf{I} - \mathbf{R})^{-2} \mathbf{e}, \quad (2.27)$$

$$L_n = \mathbf{n}_0 \pi_0^T + V \sum_{i=1}^{\infty} \pi_i \mathbf{e} = \mathbf{n}_0 \pi_0^T + V \pi_1 (\mathbf{I} - \mathbf{R})^{-1} \mathbf{e}, \quad (2.28)$$

$$L_{pq} = V - L_n, \quad (2.29)$$

$$W_s = \frac{L_n + L_{eq}}{\lambda}. \quad (2.30)$$

2.3.3 Numerical Example 1

Consider the two-stage, single-class SOQN with two exponential servers. The service rate of the first stage μ_1 is 12 and the service rate of the second stage μ_2 is 13. The arrival process is Poisson and the arrival rate λ is 10. We conduct experiments by varying the number of vehicles (V) in the system. Results as well as computing times from simulation (S), the algorithm based on state space (A1) and the algorithm based on the matrix geometric method (A2) are provided in Tables 2.1 and 2.2.

Table 2.1 Comparison of A1 and S for SOQN with two stages

	L_{eq}		L_{pq}		L_n		Utilization		W_s		Computing time	
	A1	S	A1	S	A1	S	A1	S	A1	S	A1	S
V = 5	18.42	19.36	0.38	0.38	4.62	4.62	92.4%	92.4%	146.52	143.78	40.04	27.00
V = 10	2.51	2.48	3.40	3.41	6.60	6.59	66.0%	65.9%	54.65	54.48	12.16	27.00
V = 20	0.36	0.32	12.01	12.05	7.99	7.95	40.0%	39.8%	50.14	49.63	18.91	27.00
V = 40	0.01	0.01	31.61	31.70	8.39	8.33	21.0%	20.8%	40.78	40.18	32.42	27.00

We see from these results that the algorithm based on state space and the algorithm based on MGM provide estimates of performance measures (e.g., L_{eq} , W_s) that are very close to those of simulation when the utilization of the vehicles is reason-

Table 2.2 Comparison of A2 and S for SOQN with two stages

	L_{eq}		L_{pq}		L_n		Utilization		W_s		Computing time	
	A2	S	A2	S	A2	S	A2	S	A2	S	A2	S
$V = 5$	18.50	19.36	0.38	0.38	4.62	4.62	92.4%	92.4%	138.68	143.78	0.00	27.00
$V = 10$	2.51	2.48	3.40	3.41	6.60	6.59	66.0%	65.9%	54.67	54.48	0.00	27.00
$V = 20$	0.36	0.32	12.01	12.05	7.99	7.95	40.0%	39.8%	50.14	49.63	0.00	27.00
$V = 40$	0.00	0.01	31.82	31.70	8.18	8.33	20.5%	20.8%	49.06	40.18	0.00	27.00

able (utilization $< 90\%$). When the utilization exceeds 90%, the number of states that must be considered in the truncation process increases exponentially. The algorithm based on state space is not efficient and is either unstable or it takes too long to converge.

2.4 Single-Class SOQN with Multiple Stages of Exponential Servers and Poisson Arrivals

2.4.1 Decomposition-Aggregation Method

For multiple stages of service, neither the state space based method nor a direct application of the MGM is practical. An approximation approach is used to solve this problem. The main idea is to convert the original multi-stage SOQN into an equivalent two-stage SOQN and then apply the algorithms we discussed in Sect. 2.3.

First, we combine stages other than the bottleneck stage as a closed queueing network (CQN). Then, we apply the mean value analysis (MVA) to solve this CQN to get load-dependent throughput. This CQN can be treated as an equivalent load-dependent server S_e whose service rate $\mu_e(n)$ is the throughput of this CQN. Now, the original network can be replaced by a two-stage SOQN where the first stage is the bottleneck stage, and the second stage is a load-dependent server.

This decomposition-aggregation method is based on Norton's theorem—an important theorem in electrical circuit theory. According to this theorem, the behavior of a subsystem σ between two points is the same when other parts of this circuit are replaced by a single current source and a parallel internal resistance. The value of the current source equals the current flowing between these two points when the subsystem σ is short-circuited [2]. Chandy et al. [4] proved that Norton's theorem holds for queueing networks with local balance. In order to study the behavior of a subsystem σ between two points, other parts can be replaced by a single composite queue. The service rate for this composite queue is equal to the rate at which customers pass between the two points.

Figure 2.8 shows how to apply this method to a multi-stage SOQN. Here we assume the first stage is the bottleneck stage.

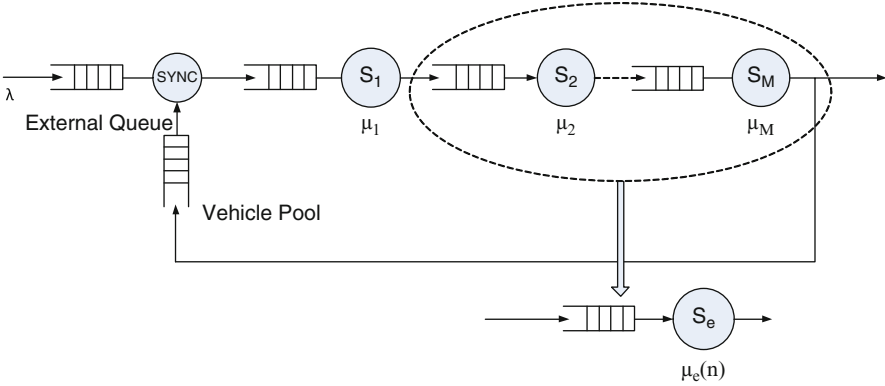


Fig. 2.8 Approximation method based on Norton's theorem

2.4.2 Numerical Example 2

We conduct a five-stage, single-class SOQN with exponential servers and Poisson arrival. The service rates for these five stages are: $\mu_1 = 12$, $\mu_2 = 13$, $\mu_3 = 15$, $\mu_4 = 14$ and $\mu_5 = 13.5$. The arrival rate λ is 10. As before, we conduct experiments by varying the number of vehicles in the system. Results from simulation (S), algorithm based on state space (A1) and algorithm based on matrix geometric method (A2) are listed in Tables 2.3 and 2.4.

Table 2.3 Comparison of A1 and simulation for SOQN with multiple stages

	L_{eq}		L_{pq}		L_n		Utilization		W_s		Computing time	
	A1	S	A1	S	A1	S	A1	S	A1	S	A1	S
$V = 15$	12.02	10.27	1.84	1.83	13.16	13.17	87.7%	87.8%	151.08	140.64	99.34	53.12
$V = 20$	2.80	2.69	5.66	5.62	14.34	14.38	71.7%	71.9%	102.83	102.42	31.37	53.12
$V = 25$	1.02	0.91	10.00	10.02	15.00	14.98	60.0%	59.9%	96.14	95.34	27.68	53.12
$V = 30$	0.41	0.46	14.64	14.56	15.36	15.44	51.2%	51.5%	94.62	95.40	19.62	53.12

Table 2.4 Comparison of A2 and simulation for SOQN with multiple stages

	L_{eq}		L_{pq}		L_n		Utilization		W_s		Computing time	
	A2	S	A2	S	A2	S	A2	S	A2	S	A2	S
$V = 15$	12.07	10.27	1.83	1.83	13.17	13.17	87.8%	87.8%	151.41	140.64	0.00	53.12
$V = 20$	2.81	2.69	5.66	5.62	14.34	14.38	71.7%	71.9%	102.90	102.42	0.00	53.12
$V = 25$	0.99	0.91	10.02	10.02	14.98	14.98	59.9%	59.9%	95.78	95.34	0.00	53.12
$V = 30$	0.42	0.46	14.45	14.56	15.55	15.44	51.8%	51.5%	86.83	95.40	0.00	53.12

2.5 Phase-Type Distribution

In order to evaluate the SOQN with general arrival and service times, we need to introduce the Phase-type distribution (PH distribution) first.

2.5.1 Definition

To analyze the property of a random variable S , we usually need the first two moments, $\mathbb{E}[S]$ and $\text{Var}[S]$, or $\mathbb{E}[S]$ and the squared coefficient of variation (SCV) of S , $C_X^2(S) = \text{Var}(S)/(\mathbb{E}[S])^2$. When S is exponentially distributed, $C_X^2(S)$ is equal to 1. If all random variables of a queueing network model are exponentially distributed, we can analyze this network as a Markov system. Otherwise, the queueing network model is a non-Markovian system. A phase-type distribution is useful in approximating a non-Markovian system as a Markovian system. After this approximation process, we can then use MGM to analyze the equivalent Markov process.

Reference [6] is the earliest paper that introduced the phase concept to approximate general distributions. In this chapter, the well-known Erlang- k distribution could be decomposed into k independent and identical exponential distributions. These k exponential distributions are called k phases of the Erlang- k distribution. Figure 2.9 shows a random variable with an Erlang- k distribution.

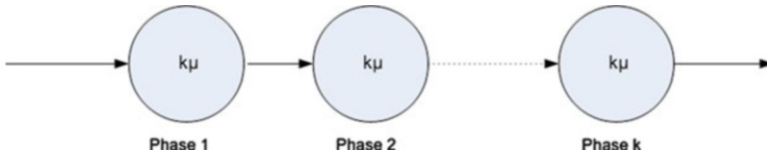


Fig. 2.9 A random variable with Erlang- k distribution

Cox [5] generalized the result of Erlang [6] and presented the set of PH distributions. The definition of PH distributions is given below:

Definition 2.1. A probability distribution $F(x)$ is a PH distribution if and only if the stochastic process of the time until absorption is a finite Markov process Q . The pair (α, \mathbf{T}) is a representation of the PH distribution.

In Definition 2.1, Q is the transition matrix of a finite Markov process with $m+1$ states. States 1 to m are transient and absorbed into state $m+1$.

$$Q = \begin{bmatrix} \mathbf{T} & \mathbf{T}^0 \\ \mathbf{0} & 0 \end{bmatrix}. \quad (2.31)$$

The distribution $F(x)$ is

$$F(x) = 1 - \alpha \exp(\mathbf{T}x)\mathbf{e}, \quad x \geq 0. \quad (2.32)$$

The Laplace-Stieltjes transform $f(s)$ of $F(x)$ is:

$$f(s) = \mathbb{E}[\exp(-sX)] = \int_{-\infty}^{\infty} e^{-sx} dF(x) = \alpha_{m+1} + \alpha(s\mathbf{I} - \mathbf{T})^{-1}\mathbf{T}^0, \quad (2.33)$$

where the real part of s is bigger than 0.

Additionally, the generator Q^* is $\mathbf{T} + \mathbf{T}^0\mathbf{A}^0$, where $\mathbf{A}^0 = (1 - \alpha_{m+1})\mathbf{T}^0\alpha$. Q^* is used to find the stationary probability vector π of m states:

$$\begin{aligned} \pi Q^* &= \pi(\mathbf{T} + \mathbf{T}^0\mathbf{A}^0) = \mathbf{0}, \\ \pi \mathbf{e} &= 1. \end{aligned} \quad (2.34)$$

The $m \times m$ matrix \mathbf{T} is the transition matrix of m transient states and \mathbf{T}^0 is a m transition vector from m transient states to the absorbing state $m+1$. T and T^0 satisfy

$$\mathbf{T}\mathbf{e} + \mathbf{T}^0 = \mathbf{0}, \quad (2.35)$$

where \mathbf{e} is a $m \times 1$ standard unit vector.

The other essential factor to define this Markov process is the initial probability of $m+1$ states, which is given by (α, α_{m+1}) . Obviously, α and α_{m+1} should satisfy the following equation:

$$\alpha\mathbf{e} + \alpha_{m+1} = 1. \quad (2.36)$$

From (2.35) and (2.36), we can see a pair of (α, \mathbf{T}) is sufficient to represent a PH distribution.

We give two examples to indicate how to define PH distributions. The first example is the classic Erlang- k distribution with parameters $\lambda_1, \dots, \lambda_k$ and the initial probabilities of the k states are $\alpha = \{1, 0, \dots, 0\}$. Then, the transition matrix of k states is given by

$$\mathbf{T} = \begin{bmatrix} -\lambda_1 & \lambda_1 & & & \\ & -\lambda_2 & \lambda_2 & & \\ & & \dots & & \\ & & & -\lambda_{k-1} & \lambda_{k-1} \\ & & & & -\lambda_k \end{bmatrix}.$$

The transition vector to the absorbed state $k+1$ $\mathbf{T}^0 = -\mathbf{T}\mathbf{e}$ is $\{0, \dots, -\lambda_m\}'$. The initial probability of absorbed state $k+1$ $\alpha_{k+1} = 1 - \alpha\mathbf{e}$ is 0. If $\lambda_1 = \lambda_2 = \dots = \lambda_k$, C_X^2 of this PH distribution is $1/k$.

The second example is the Coxian distribution or Coxian- k distribution. This is also the PH distribution used in this chapter. As the name of this distribution suggests, the Coxian- k distribution is represented by a k -phase Markov process. Each phase has an exponentially distributed rate μ_k . After the i th phase, the probability of entering the next phase is a_i , and the probability of being absorbed is b_i , where $a_i + b_i = 1$. Figure 2.10 shows a random variable with Coxian- k distribution.

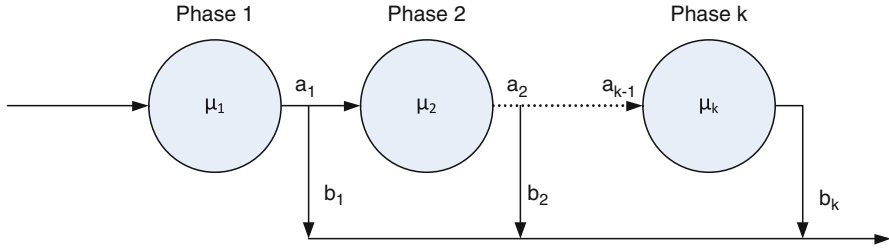


Fig. 2.10 Coxian- k distribution

This Coxian- k distribution can be represented by a pair (α, \mathbf{T}) where $\alpha = \{1, 0, \dots, 0\}$ and

$$\mathbf{T} = \begin{bmatrix} -\mu_1 & a_1\mu_1 & & & \\ & -\mu_2 & a_2\mu_2 & & \\ & & \dots & & \\ & & & -\mu_{k-1} & a_{k-1}\mu_{k-1} \\ & & & & -\mu_k \end{bmatrix}.$$

There are two cases of Coxian- k distribution.

Case I: $C_X^2 \leq 1$. In this case, all phases have same service rate μ , and the probability of entering the next phase is 1 except for the first phase. This case is shown in Fig. 2.11.

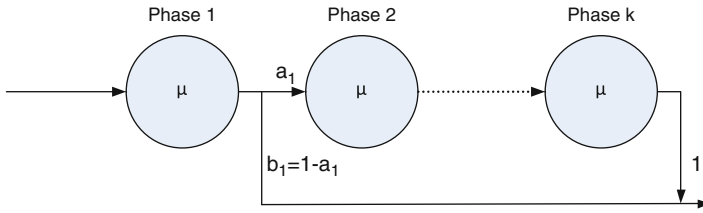


Fig. 2.11 Coxian- k distribution with $C_X^2 \leq 1$

The representation of this case is $\alpha = \{1, 0, \dots, 0\}$ and \mathbf{T} is

$$\begin{bmatrix} -\mu & a_1\mu & & & \\ & -\mu & \mu & & \\ & & \dots & & \\ & & & -\mu & \mu \\ & & & & -\mu \end{bmatrix}.$$

According to Sauer and Chandy [14], μ and a_1 can be estimated by (2.37),

$$\mu = \frac{k - (1 - a_1)(k - 1)}{\bar{X}},$$

$$a_1 = 1 - \frac{2kC_X^2 + (k - 1) - \sqrt{k^2 + 4 - 4kC_X^2}}{2(C_X^2 + 1)(k - 1)}, \quad (2.37)$$

where \bar{X} is the mean value. The number of phases k can be estimated by (2.38),

$$k = \lceil \frac{1}{C_X^2} \rceil. \quad (2.38)$$

Case II: $C_X^2 > 1$. In this case, the number of phases is fixed to 2. Therefore, it is also called a Coxian-2 distribution. The service rate of the first stage is μ_1 and the service rate of the second stage is μ_2 . Figure 2.12 shows the Coxian-2 distribution.

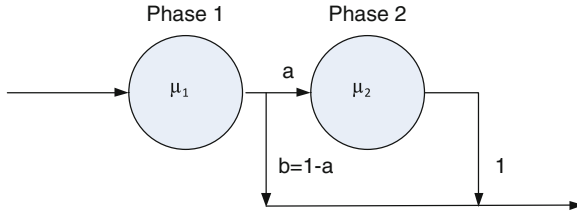


Fig. 2.12 Coxian- k distribution with $C_X^2 > 1$ (Coxian-2 distribution)

According to Sauer and Chandy [14], μ_1 , μ_2 and a are estimated by (2.39),

$$\mu_1 = \frac{2}{\bar{X}},$$

$$\mu_2 = \frac{1}{\bar{X}C_X^2}, \quad (2.39)$$

$$a = \frac{1}{2C_X^2}.$$

2.5.2 Closure Properties and Kronecker Product

2.5.2.1 Closure Properties

We can estimate general distributions with different C_X^2 s by a PH distribution. We start from a single stage queueing model where the inter-arrival and service times are generally distributed. Now we can approximate a simple GI/G/1 queue as a PH/PH/1 queue, in which the arrival procedure is represented by the pair (α, \mathbf{T}) and the service procedure is represented by the pair (β, \mathbf{S}) . Neuts [13] proved that the PH distribution property holds even after the mixture.

Theorem 2.2. If F is a PH distribution of $m + 1$ states with representation (α, \mathbf{T}) and G is also a PH distribution of n states with representation (β, \mathbf{S}) , then the convolution $F * G$ is still a PH distribution with representation (γ, \mathbf{L}) , where

$$\begin{aligned} \gamma &= [\alpha, \alpha_{m+1}\beta] \\ \mathbf{L} &= \begin{bmatrix} \mathbf{T} & \mathbf{T}^0 \mathbf{B}^0 \\ \mathbf{0} & \mathbf{S} \end{bmatrix}. \end{aligned} \quad (2.40)$$

Figure 2.13 shows the process of a $PH/PH/1$ queue with the arrival procedure (α, \mathbf{T}) and the service procedure (β, \mathbf{S}) . According to Theorem 2.2, the distribution of this process is still a PH distribution.

Here we assume C_X^2 s of both inter-arrival and service times are greater than 1.

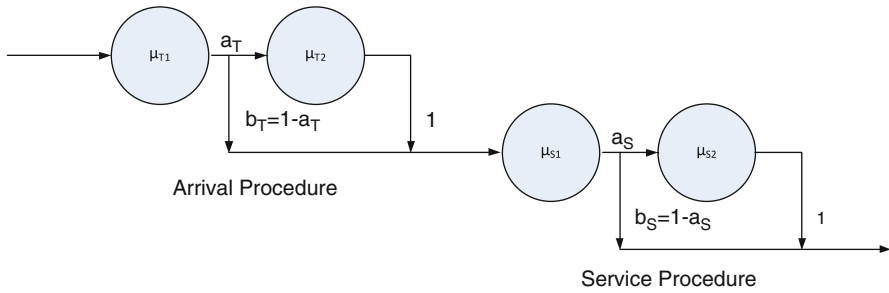


Fig. 2.13 A $PH/PH/1$ queue

There are 4 stages in this $PH/PH/1$ queue. Initially, there is no customer at any stage. Once a customer is generated, there is one customer at the first phase of the arrival process. The next moment, the probability that this customer is transferred to the second phase is a_T , and the probability the customer is absorbed is b_T . Here, the arrival procedure is renewed when the absorption state is reached. At the same time, the customer is transferred to the first phase of the service process. We can denote the state of this $PH/PH/1$ queue as (n, a_l, s_l) , where n is the number of customers in the service node or the level, a_l is the l th phase of the arrival process and s_l is the l th phase of the service process.

The state space of this $PH/PH/1$ queue is complex. However, the Markov process of the $PH/PH/1$ queue can be viewed as an embedded finite Markov process (PH distribution) in a $M/M/1$ queue. The $M/M/1$ queue is an example of the birth-death process. Hence, Neuts [13] discussed the $PH/PH/1$ as a direct example of a QBD process. We divide the state space into two parts. The first part is the initial part, or state space between level 0 and level 0, as well as between level 0 and level 1. The second part is the repetitive state space between levels 1 and $n - 1$, as well as between levels $n - 1$ and n .

Since this Markov process satisfies these properties, it is a continuous time, irreducible, homogeneous QBD process. The original problem now is treated as determining the unconditional stationary state probabilities of QBD. In a QBD process,

the number of customers in the external queue i is the i th level, and number of customers at each service stage (j, k) is the phase (j, k) . According to this, we denote π_i as the vector of unconditional stationary state probabilities of all phases at the i th level. This QBD has a repetitive structure of \mathbf{Q} like this:

$$\mathbf{Q} = \begin{bmatrix} \mathbf{B}_{00} & \mathbf{B}_{01} & \mathbf{0} & \mathbf{0} & \cdots \\ \mathbf{B}_{10} & \mathbf{A}_1 & \mathbf{A}_0 & \mathbf{0} & \cdots \\ \mathbf{0} & \mathbf{A}_2 & \mathbf{A}_1 & \mathbf{A}_0 & \cdots \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_2 & \mathbf{A}_1 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}, \quad (2.41)$$

where \mathbf{B}_{00} , \mathbf{B}_{01} and \mathbf{B}_{10} are instantaneous transition rate matrices that determine the initial state of the system.

$$\mathbf{B}_{00} = \begin{matrix} & (0,1) & (0,2) \\ \begin{matrix} (0,1) \\ (0,2) \end{matrix} & \begin{pmatrix} -\mu_{1T} & a_T \mu_{1T} \\ 0 & \mu_{2T} \end{pmatrix} \end{matrix},$$

$$\mathbf{B}_{01} = \begin{matrix} & (1,1,1) & (1,1,2) & (1,2,1) & (1,2,2) \\ \begin{matrix} (0,1) \\ (0,2) \end{matrix} & \begin{pmatrix} (1-a_T)\mu_{1T}\alpha_1\beta_1 & (1-a_T)\mu_{1T}\alpha_1\beta_2 & (1-a_T)\mu_{1T}\alpha_2\beta_1 & (1-a_T)\mu_{1T}\alpha_2\beta_2 \\ \mu_{2T}\alpha_1\beta_1 & \mu_{2T}\alpha_1\beta_2 & \mu_{2T}\alpha_2\beta_1 & \mu_{2T}\alpha_2\beta_2 \end{pmatrix} \end{matrix},$$

$$\mathbf{B}_{10} = \begin{matrix} & (0,1) & (0,2) \\ \begin{matrix} (1,1,1) \\ (1,1,2) \\ (1,2,1) \\ (1,2,2) \end{matrix} & \begin{pmatrix} (1-a_S)\mu_{1S} & 0 \\ \mu_{2S} & 0 \\ 0 & (1-a_S)\mu_{1S} \\ 0 & \mu_{2S} \end{pmatrix} \end{matrix}.$$

Similarly, we can get the transition matrices of the repetitive part of the state space of this $PH/PH/1$ queue.

$$\mathbf{A}_0 = \begin{matrix} & (n-1,1,1) & (n-1,1,2) & (n-1,1,1) & (n-1,1,2) \\ \begin{matrix} (n,1,1) \\ (n,1,2) \\ (n,2,1) \\ (n,2,2) \end{matrix} & \begin{pmatrix} (1-a_T)\mu_{1T}\alpha_1 & 0 & (1-a_T)\mu_{1T}\alpha_2 & 0 \\ 0 & (1-a_T)\mu_{1T}\alpha_1 & 0 & (1-a_T)\mu_{1T}\alpha_2 \\ \mu_{2T}\alpha_1 & 0 & \mu_{2T}\alpha_2 & 0 \\ 0 & \mu_{2T}\alpha_1 & 0 & \mu_{2T}\alpha_2 \end{pmatrix} \end{matrix},$$

$$\mathbf{A}_1 = \begin{matrix} & \begin{matrix} (n, 1, 1) & (n, 1, 2) & (n, 1, 1) & (n, 1, 2) \end{matrix} \\ \begin{matrix} (n, 1, 1) \\ (n, 1, 2) \\ (n, 2, 1) \\ l(n, 2, 2) \end{matrix} & \begin{pmatrix} -\mu_{1T} - \mu_{1S} & a_S \mu_{1S} & a_T \mu_{1T} & 0 \\ 0 & -\mu_{1T} - \mu_{2S} & 0 & a_T \mu_{1T} \\ 0 & 0 & -\mu_{2T} - \mu_{1S} & a_S \mu_{1S} \\ 0 & 0 & 0 & -\mu_{2T} - \mu_{2S} \end{pmatrix} \end{matrix},$$

$$\mathbf{A}_2 = \begin{matrix} & \begin{matrix} (n, 1, 1) & (n, 1, 2) & (n, 1, 1) & (n, 1, 2) \end{matrix} \\ \begin{matrix} (n-1, 1, 1) \\ (n-1, 1, 2) \\ (n-1, 2, 1) \\ (n-1, 2, 2) \end{matrix} & \begin{pmatrix} (1-a_S)\mu_{1S}\beta_1 & (1-a_S)\mu_{1S}\beta_2 & 0 & 0 \\ \mu_{2S}\beta_1 & \mu_{2S}\beta_2 & 0 & 0 \\ 0 & 0 & (1-a_S)\mu_{1S}\beta_1 & (1-a_S)\mu_{1S}\beta_2 \\ 0 & 0 & \mu_{2S}\beta_1 & \mu_{2S}\beta_2 \end{pmatrix} \end{matrix}.$$

Now, we can get a similar generator Q as (2.41), and apply the MGM to analyze this $PH/PH/1$ queue.

2.5.2.2 Kronecker Product

Although $PH/PH/1$ is a very simple queue, the generator Q is very complex. Moreover, Theorem 2.2 can be extended to the convolution of multiple PH distributions. The generator Q of this case will be even more complex.

Fortunately, an important property of matrices called the Kronecker product of matrices can be used to simplify Q . The detail and proof of the Kronecker product of matrices can be found in [1].

Definition 2.2. Let \mathbf{A} be an $m_1 \times n_1$ matrix and \mathbf{B} be an $m_2 \times n_2$ matrix. Then the Kronecker product of \mathbf{A} and \mathbf{B} , $\mathbf{A} \otimes \mathbf{B}$, is

$$\begin{bmatrix} A_{11}B & A_{12}B & \dots & A_{1n_1}B \\ A_{21}B & A_{22}B & \dots & A_{2n_1}B \\ \vdots & \vdots & \ddots & \vdots \\ A_{m_1 1}B & A_{m_1 2}B & \dots & A_{m_1 n_1}B \end{bmatrix}_{m_1 m_2 \times n_1 n_2}. \quad (2.42)$$

According to Neuts [13], the generator Q of the $PH/PH/1$ queue can be rewritten as follows:

$$\begin{aligned} \mathbf{B}_{00} &= \mathbf{T}, \\ \mathbf{B}_{01} &= \mathbf{T}^0 \mathbf{A}^0 \otimes \beta, \\ \mathbf{B}_{10} &= \mathbf{I}_T \otimes \mathbf{S}^0, \\ \mathbf{A}_0 &= \mathbf{T}^0 \mathbf{A}^0 \otimes \mathbf{I}_S, \\ \mathbf{A}_1 &= \mathbf{T} \otimes \mathbf{I}_S + \mathbf{I}_T \otimes \mathbf{S}, \\ \mathbf{A}_2 &= \mathbf{I}_T \otimes \mathbf{S}^0 \mathbf{B}^0. \end{aligned}$$

Here \mathbf{I}_T is the diagonal matrix with the same size of \mathbf{T} and \mathbf{I}_S is the diagonal matrix with the same size of \mathbf{S} .

2.6 Single-Class SOQN with Two Stages of General Servers and General Arrival

2.6.1 State Space Analysis

Compared to the $PH/PH/1$ queue we discussed in Sect. 2.5, the second stage brings in new phases. In order to simplify the notation, we still assume C_X^2 of the service process at the second stage is greater than 1. It is easy to extend to the $C_X^2 \leq 1$ case. Figure 2.14 shows this two-stage SOQN with PH distributions.

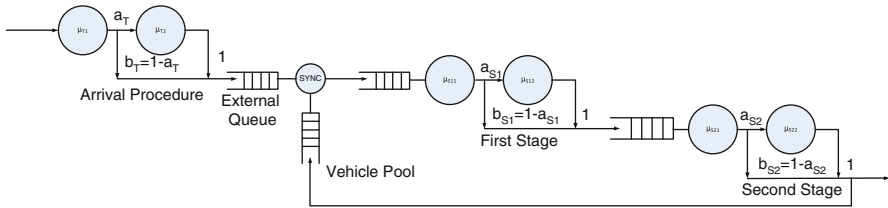


Fig. 2.14 A two-stage SOQN with PH distributions

The arrival process is represented by the pair (α, \mathbf{T}) . The service processes at the first and second stages are represented by the pairs (β, \mathbf{S}_1) and (ν, \mathbf{S}_2) respectively. Since we do not want these processes to begin in the absorption phase, we set $\alpha_3 = \beta_3 = \nu_3 = 0$. Therefore,

$$\mathbf{T}^0 \mathbf{A}_T^0 = \mathbf{T}^0 \alpha,$$

$$\mathbf{S}_1^0 \mathbf{A}_{S_1}^0 = \mathbf{S}_1^0 \beta,$$

$$\mathbf{S}_2^0 \mathbf{A}_{S_2}^0 = \mathbf{S}_2^0 \nu.$$

We extend the notation of the $PH/PH/1$ queue in Sect. 2.5 to describe the SOQN with PH distributed arrival and service processes. Each state s_m in the state space $(i, j, a_l, s_{1l}, s_{2l})$ denotes that there are i customers at the external and first queue, or level i , there are j customers at the second queue, the current phase of arrival process is a_l and the current phases of the two service processes are s_{1l} and s_{2l} respectively:

$$s_m = (i, j, a_l, s_{1l}, s_{2l}), \text{ where } 0 \leq i, 0 \leq j \leq N.$$

Similar to the $PH/PH/1$ queue, the Markov process of this SOQN can be viewed as a QBD process with several embedded finite state Markov processes. The gener-

ator Q of this process is similar to (2.41), but much more elaborate. As before, we analyze the initial and repetitive parts separately.

\mathbf{B}_{00} is the transition matrix of level 0, where j is changed from 0 to N . This transition matrix can be viewed as a part of the generator of $PH/PH/1$ queue of the first N levels. The only difference is that it is impossible to travel from j to $j+1$. This is reasonable because if there is no customer at the external queue and the first stage, the number of customers at the second stage cannot be increased. This slight difference does not hurt the QBD property. \mathbf{B}_{00} still contains the initial and repetitive parts.

In \mathbf{B}_{00} , $(0,0)$ denotes two states $(0,0,1,\square,\square)$ and $(0,0,2,\square,\square)$. $(0,j)$ denotes four states $(0,0,1,\square,1)$, $(0,0,1,\square,2)$, $(0,0,2,\square,1)$, and $(0,0,2,\square,2)$. \square means the states of this process does not change in this part.

$$\mathbf{B}_{00} = \begin{matrix} & \begin{matrix} (0,0) & (0,1) & (0,2) & \dots & (0,N) \end{matrix} \\ \begin{matrix} (0,0) \\ (0,1) \\ (0,2) \\ \vdots \\ (0,N) \end{matrix} & \begin{pmatrix} \mathbf{T} & & & & \\ \mathbf{I}_T \otimes \mathbf{S}_2^0 & \mathbf{T} \otimes \mathbf{I}_{S_2} + \mathbf{I}_T \otimes \mathbf{S}_2 & & & \\ & \mathbf{I}_T \otimes \mathbf{S}_2^0 \gamma & \mathbf{T} \otimes \mathbf{I}_{S_2} + \mathbf{I}_T \otimes \mathbf{S}_2 & & \\ & & \ddots & \ddots & \\ & & & \mathbf{I}_T \otimes \mathbf{S}_2^0 \gamma & \mathbf{T} \otimes \mathbf{I}_{S_2} + \mathbf{I}_T \otimes \mathbf{S}_2 \end{pmatrix} \end{matrix}.$$

\mathbf{B}_{01} is the transition matrix from level 0 to level 1, where j is changed from 1 to N . In this part, the situation is more complicated than \mathbf{B}_{00} . The initial part is from $(0,0)$ to $(1,0)$. $(1,0)$ denotes four states $(1,0,1,1,\square)$, $(1,0,1,2,\square)$, $(1,0,2,1,\square)$, and $(1,0,2,2,\square)$.

The transition matrix from $(0,j)$ to $(1,j)$ is different because it involves three PH distributed processes. $(1,j)$ denotes eight states: $(1,j,1,1,1)$, $(1,j,1,1,2)$, $(1,j,1,2,1)$, $(1,j,1,2,2)$, $(1,j,2,1,1)$, $(1,j,2,1,2)$, $(1,j,2,2,1)$, and $(1,j,2,2,2)$. Neuts [13] proved that Theorem 2.2 can be extended to a more general conclusion: a finite mixture of PH distributions is still a PH distribution. Therefore, the transition matrix from $(0,1)$ to $(1,1)$ can be extended from the transition matrix of $PH/PH/1$ queue from level 0 to level 1.

The second difference is the last part of \mathbf{B}_{01} from $(0,N)$ to $(1,N)$. $(1,N)$ denotes four states $(1,N,1,\square,1)$, $(1,N,1,\square,2)$, $(1,N,2,\square,1)$ and $(1,N,2,\square,2)$. Since there are at most N customers at two stages and the number of customers at the second stage is N , the number of customers at the first stage must be 0. Hence, this one customer of $(1,N)$ must be at the external queue waiting for the next available resource.

\mathbf{B}_{10} is the transition matrix from level 1 to level 0, where j is changed from 1 to N . The initial part is the transition matrix from $(1,0)$ to $(0,1)$. Similar to \mathbf{B}_{01} , the convolution of three PH distributions is still a PH distribution. Hence, the initial part is the mixture of the initial part from level 1 to level 0 at the external queue, the first stage and the initial part from level 0 to level 1 at the second stage.

$$\mathbf{B}_{10} = \begin{matrix} & (0,0) & (0,1) & (0,2) & \dots & (0,N) \\ \begin{matrix} (1,0) \\ (1,1) \\ \vdots \\ (1,N-1) \\ (1,N) \end{matrix} & \begin{pmatrix} & \mathbf{I}_T \otimes \mathbf{S}_1^0 \otimes \gamma & & & \\ & & \mathbf{I}_T \otimes \mathbf{S}_1^0 \otimes \mathbf{I}_{S_2} & & \\ & & & \ddots & \\ & & & & \mathbf{I}_T \otimes \mathbf{S}_1^0 \otimes \mathbf{I}_{S_2} \end{pmatrix} \end{matrix}.$$

The repetitive part can also be separated into three parts. \mathbf{A}_0 is the transition matrix from level $i-1$ to level i , where j is changed from 0 to N . \mathbf{A}_0 has a layout similar to \mathbf{B}_{01} . The transition matrix of the service process at the second stage is the same in \mathbf{A}_0 and \mathbf{B}_{01} . Note that although \mathbf{A}_0 and \mathbf{B}_{01} look similar, \mathbf{A}_0 is the repetitive part and \mathbf{B}_{01} is the boundary part of the generator.

$$\mathbf{A}_0 = \begin{matrix} & (i,0) & (i,1) & (i,2) & \dots & (i,N) \\ \begin{matrix} (i-1,0) \\ (i-1,1) \\ (i-1,2) \\ \vdots \\ (i-1,N) \end{matrix} & \begin{pmatrix} \mathbf{T}^0 \alpha \otimes \mathbf{I}_{S_1} & & & & \\ & \mathbf{T}^0 \alpha \otimes \mathbf{I}_{S_1} \otimes \mathbf{I}_{S_2} & & & \\ & & \mathbf{T}^0 \alpha \otimes \beta \otimes \mathbf{I}_{S_2} & & \\ & & & \ddots & \\ & & & & \mathbf{T}^0 \alpha \otimes \mathbf{I}_{S_2} \end{pmatrix} \end{matrix}.$$

\mathbf{A}_1 is the transition matrix from level i to level i , where j is changed from 0 to N . \mathbf{A}_1 should have a layout similar to \mathbf{B}_{00} . In \mathbf{B}_{00} , the states of the service processes at the first stage do not change. However, \mathbf{A}_1 is more complicated because of the mixture of three PH distributed processes.

$$\mathbf{A}_1 = \begin{matrix} & (i,0) & (i,1) & (i,2) & \dots & (i,N) \\ \begin{matrix} (i,0) \\ (i,1) \\ (i,2) \\ \vdots \\ (i,N) \end{matrix} & \begin{pmatrix} \mathbf{T} \otimes \mathbf{I}_{S_1} + \mathbf{I}_T \otimes \mathbf{S}_1 & & & & \\ \mathbf{I}_T \otimes \mathbf{I}_{S_1} \otimes \mathbf{S}_2^0 & (\mathbf{T} \otimes \mathbf{I}_{S_1} + \mathbf{I}_T \otimes \mathbf{S}_1) \otimes \mathbf{I}_{S_2} + \mathbf{I}_{TS_1} \otimes \mathbf{S}_2 & & & \\ & \mathbf{I}_T \otimes \mathbf{I}_{S_1} \otimes \mathbf{S}_2^0 \gamma & (\mathbf{T} \otimes \mathbf{I}_{S_1} + \mathbf{I}_T \otimes \mathbf{S}_1) \otimes \mathbf{I}_{S_2} + \mathbf{I}_{TS_1} \otimes \mathbf{S}_2 & & \\ & & & \ddots & \\ & & & & \mathbf{I}_T \otimes \beta \otimes \mathbf{S}_2^0 \gamma \quad \mathbf{T} \otimes \mathbf{I}_{S_2} + \mathbf{I}_T \otimes \mathbf{S}_2 \end{pmatrix} \end{matrix}.$$

\mathbf{A}_2 is the transition matrix from level $i+1$ to level i , where j is changed from 0 to N . \mathbf{A}_2 has a layout similar to \mathbf{B}_{10} . The only difference is that the initial part in \mathbf{B}_{10} should be replaced by the repetitive part in \mathbf{A}_2 .

$$\mathbf{A}_2 = \begin{matrix} & (i,0) & (i,1) & (i,2) & \dots & (i,N) \\ \begin{matrix} (i+1,0) \\ (i+1,1) \\ \vdots \\ (i+1,N-1) \\ (i+1,N) \end{matrix} & \begin{pmatrix} & \mathbf{I}_T \otimes \mathbf{S}_1^0 \beta \otimes \gamma & & & \\ & & \mathbf{I}_T \otimes \mathbf{S}_1^0 \beta \otimes \mathbf{I}_{S_2} & & \\ & & & \ddots & \\ & & & & \mathbf{I}_T \otimes \mathbf{S}_1^0 \otimes \mathbf{I}_{S_2} \end{pmatrix} \end{matrix}.$$

From the state space analysis of single-class SOQN with two general stages and arrival process, we find the generator Q is very complex. Hence, the state space

solution is not a good choice to solve stationary probabilities. The MGM is used to get stationary probability vectors.

2.6.2 Numerical Example 3

The system considered is a warehouse where autonomous vehicles are paired with storage or retrieval transactions. Thus, the storage and retrieval transactions are the customers and the vehicles are the resources. We conduct numerical experiments to show the effectiveness of the approximation method. The results from the approximation method (A) are compared with those from simulation (S).

The first part is to examine the accuracy of our method for systems with low and high variances. The first case is a Coxian- k distribution with low variance and the second case is a Coxian-2 distribution with high variance. We construct a one-stage generalized SOQN or a $PH/PH/1$ queue with population restriction. There are two sets of experiments.

In the first set of experiments, we set the distribution of the inter-arrival time as exponential with a mean value of 1.5 and the distribution of the service time as Erlang-2 with a mean value of 1. The exponential distribution is an example of moderate variance with $C_X^2 = 1$. The Erlang-2 distribution is an example of low variance since $C_X^2 = 0.5$ in this case. We conduct experiments by varying the number of vehicles V in the system. Table 2.5 shows the number of customers in the external queue L_{eq} , the number of customers at service stage L_{pq} and the utilization of the vehicles.

Table 2.5 Results of Exponential/Erlang-2/1

	$V = 10$			$V = 5$			$V = 2$		
	L_{eq}	L_{pq}	Utilization	L_{eq}	L_{pq}	Utilization	L_{eq}	L_{pq}	Utilization
A	0.25	3.36	33.6%	0.95	2.66	53.2%	2.12	1.49	74.5%
S	0.19	3.43	34.3%	0.86	2.78	55.6%	1.97	1.64	82.0%
error%	24.0	2.08	1.05	9.47	4.51	5.13	7.08	10.1	10.1

From Table 2.5, we can see that our proposed approximation method works well.

In the second set of experiments, we assign distribution with higher variance (Gamma with a mean value of 1.5 and $C_X^2 = 1.2$) to the arrival process and distribution with lower variance (Erlang-3 with a mean value of 1 and $C_X^2 = 1/3$) to the service process.

As shown in Table 2.6, the accuracy is not as good as the accuracy in Table 2.5. One possible explanation is that it is caused by the approximate estimation of distributions with high and low variances.

The second part of our experiment is to examine our approximation method of the two-stage, single-class SOQN with PH distributions. We construct a two-stage

Table 2.6 Results of Gamma/Erlang-3/1

	$V = 10$			$V = 5$			$V = 2$		
	L_{eq}	L_{pq}	Utilization	L_{eq}	L_{pq}	Utilization	L_{eq}	L_{pq}	Utilization
A	0.009	1.69	16.9%	0.13	1.57	31.4%	0.62	1.08	54.0%
S	0.007	1.93	19.3%	0.11	1.83	36.6%	0.80	1.59	79.5%
error%	22.2	14.2	2.89	15.4	16.6	7.60	12.9	47.2	55.4

SOQN as follows: the inter-arrival time is exponentially distributed with a mean of 1.5. The distributions of service time at the first and second stages are Erlang-2 with a mean of 1 and C_X^2 of 0.5, Gamma with a mean of 1 and C_X^2 of 1.2, respectively.

From Table 2.7, we can see that the result of the second stage is better than the result of the first stage. It appears that the approximation method of distributions with low variance needs to be improved.

Table 2.7 Results of two-stage SOQN

	$V = 10$				$V = 5$				$V = 2$			
	L_{eq}	L_1	L_2	Utilization	L_{eq}	L_1	L_2	Utilization	L_{eq}	L_1	L_2	Utilization
A	0.014	0.45	1.54	19.9%	0.18	0.44	1.39	36.5%	1.12	0.38	0.91	64.5%
S	0.012	0.34	1.39	17.3%	0.13	0.33	1.29	32.4%	0.78	0.3	0.88	59.0%
error%	14.3	30.9	9.74	3.24	27.7	25.0	7.2	6.45	30.4	21.1	3.30	15.5

2.6.3 Multiple Servers

If there are multiple servers at a service stage and the service time of each server is exponentially distributed, the service time of the entire stage is no longer exponentially distributed. Neuts [13] proved that the MGM can give a complete generator for a $PH/PH/c$ queue with heterogeneous servers. However, the critical matrix \mathbf{R} used in MGM is rather difficult to compute when the number of parallel servers is large. In order to get results in reasonable computation time, we make two assumptions on the SOQNs. The first assumption is that the servers on the same stage are identical, which means all servers have the same service time distribution. This assumption allows a major simplification in the state space description. It is also a reasonable assumption in real world applications because servers in the same service node often execute the same task. The second assumption is that the number of servers is not greater than 10, which is due to the limitation of the MGM.

The algorithm for the multiple servers situation was first introduced by Mayhugh and McCormick [11] for the $PH/PH/c$ queue model. Let c_1 and c_2 being the number of parallel servers at the first and second stage respectively. Each state s_m now can be described as $(i, j, a_l, s_{1l1}, \dots, s_{1lc_1}, s_{2l1}, \dots, s_{2lc_2})$ or $(i, j, a_l, \mathbf{s}_{1l}, \mathbf{s}_{2l})$,

where \mathbf{s}_{11} and \mathbf{s}_{21} are vectors of current phases of all possible busy servers at the two stages. It is straightforward to extend the generator of the SOQN of the single-server case to the generator of the SOQN of the multi-server case. We know that if all servers at a stage are busy, the behavior of this stage should be the same as that of the single-server stage because customers have to wait in the queue in front of that stage. Hence, the only difference is in the initial part when some servers are idle. The generator Q is rewritten as

$$Q = \begin{bmatrix} \mathbf{A}_{10} & \mathbf{A}_{00} & & & & \\ \mathbf{A}_{21} & \mathbf{A}_{11} & \mathbf{A}_{01} & & & \\ & \mathbf{A}_{22} & \mathbf{A}_{12} & \mathbf{A}_{02} & & \\ & & \ddots & \ddots & \ddots & \\ & & & \mathbf{A}_{2c_1-1} & \mathbf{A}_{1c_1-1} & \mathbf{A}_{0c_1-1} \\ & & & & \mathbf{A}_{2c_1} & \mathbf{A}_{1c_1} & \mathbf{A}_{0c_1} \\ & & & & & \mathbf{A}_{2c_1+1} & \mathbf{A}_{1c_1} & \mathbf{A}_{0c_1} \\ & & & & & & \ddots & \ddots & \ddots \end{bmatrix}. \quad (2.43)$$

Before we analyze this generator, we introduce an additional notation called Kronecker sum, which is a simple extension of Kronecker product (Definition 2.2).

Definition 2.3. The Kronecker sum of matrices \mathbf{A} and \mathbf{B} is

$$\mathbf{A} \oplus \mathbf{B} = \mathbf{A} \otimes \mathbf{I}_B + \mathbf{I}_A \otimes \mathbf{B}. \quad (2.44)$$

Additionally, the Kronecker product and Kronecker product of multiple matrices can be expressed as:

$$\begin{aligned} \mathbf{A}_0 \otimes \mathbf{A}_1 \otimes \dots \otimes \mathbf{A}_N &= \otimes^N \mathbf{A}_n \\ \mathbf{A}_0 \oplus \mathbf{A}_1 \oplus \dots \oplus \mathbf{A}_N &= \oplus^N \mathbf{A}_n \end{aligned} \quad (2.45)$$

In Q , \mathbf{A}_{2t} , \mathbf{A}_{1t} and \mathbf{A}_{0t} are extended from \mathbf{A}_2 , \mathbf{A}_1 and \mathbf{A}_0 of the single-server case. They indicate the transition behavior when there are t servers busy at the first stage. We choose \mathbf{A}_{1t} to discuss in detail, and give the result of \mathbf{A}_{0t} and \mathbf{A}_{2t} directly.

Similar to the \mathbf{A}_1 of single-server case, \mathbf{A}_{1t} has two parts.

$$\mathbf{A}_{1t} = \begin{matrix} & \begin{matrix} (t,0) & (t,1) & \dots & (t,V) \end{matrix} \\ \begin{matrix} (t,0) \\ (t,1) \\ \vdots \\ (t,V) \end{matrix} & \begin{pmatrix} \mathbf{A}_{1t}^{(0,0)} & & & \\ \mathbf{A}_{1t}^{(1,0)} & \mathbf{A}_{1t}^{(1,1)} & & \\ & \ddots & \ddots & \\ & & \mathbf{A}_{1t}^{(v,v-1)} & \mathbf{A}_{1t}^{(v,v)} \end{pmatrix} \end{matrix}.$$

The first part contains sub-matrices on the diagonal, $\mathbf{A}_{1t}^{(v,v)}$, where v is the number of customers at stage 2. By using Definition 2.3, $\mathbf{A}_{1t}^{(v,v)}$ can be written as $\mathbf{T} \oplus \mathbf{S}_1 \oplus \mathbf{S}_2$.

Hence,

$$\mathbf{A}_{1t}^{(v,v)} = \mathbf{T} \oplus (\oplus^{\min(t, V-v)} \mathbf{S}_1) \oplus (\oplus^{\min(v, c_2)} \mathbf{S}_2).$$

The second part contains sub-matrices from (t, v) to $(t, v-1)$. In \mathbf{A}_1 , $\mathbf{A}_1^{(1,0)}$ is $\mathbf{I}_T \otimes \mathbf{I}_{S_1} \otimes \mathbf{S}_2^0$, $\mathbf{A}_1^{(v, v-1)}$ is $\mathbf{I}_T \otimes \mathbf{I}_{S_1} \otimes \mathbf{S}_2^0 \gamma$ and $\mathbf{A}_1^{(v, V-1)}$ is $\mathbf{I}_T \otimes \beta \otimes \mathbf{S}_2^0 \gamma$. \mathbf{A}_{1t} is more complicated because we must consider different scenarios of busy servers at the second stage.

- $1 \leq v \leq c_2$

In this scenario, there is no customer waiting for service at the second stage. Hence, there is no change in the arrival process and the service process at the first stage transitions from (t, v) to $(t, v-1)$.

$$\mathbf{A}_{1t}^{(v, v-1)} = \mathbf{I}_T \otimes (\otimes^t \mathbf{I}_{S_1}) \otimes \left(\sum_{h=v-1}^1 \mathbf{S}_2^0 \otimes (\otimes^h \mathbf{I}_{S_2}) + \sum_{h=1}^{v-1} (\otimes^h \mathbf{I}_{S_2}) \otimes \mathbf{S}_2^0 \right).$$

- $c_2 \leq v \leq V$ and $t+v \leq V$

In this scenario, there are some customers waiting in front of the second stage and no customer is waiting outside. When a customer leaves the system, the first customer in the queue in front of the second stage enters into the second stage when the system transitions from (t, v) to $(t, v-1)$.

$$\mathbf{A}_{1t}^{(v, v-1)} = \mathbf{I}_T \otimes (\otimes^t \mathbf{I}_{S_1}) \otimes (\oplus^{c_2} \mathbf{S}_2^0 \gamma).$$

- $c_2 \leq v \leq V$ and $t+v > V$

In this scenario, there are customers waiting in front of the second stage and outside. When the system transitions from (t, v) to $(t, v-1)$, a customer leaves the system from the second stage, the first customer in the queue in front of the second stage enters into the stage, and the first customer waiting outside obtains the released resource to be served at the first stage.

$$\mathbf{A}_{1t}^{(v, v-1)} = \mathbf{I}_T \otimes (\otimes^{V-v} \mathbf{I}_{S_1}) \otimes \beta \otimes (\oplus^{c_2} \mathbf{I}_{S_2} \gamma).$$

\mathbf{A}_{0t} is the transition matrix from the current level to the next level, which is similar to \mathbf{A}_0 of the single-server case. \mathbf{A}_{0t} has sub-matrices only on the diagonal.

$$\mathbf{A}_{0t} = \begin{matrix} & (t+1, 0) & (t+1, 1) & \dots & (t+1, V) \\ \begin{matrix} (t, 0) \\ (t, 1) \\ \vdots \\ (t, V) \end{matrix} & \left(\begin{array}{cccc} \mathbf{A}_{0t}^{(0,0)} & & & \\ & \mathbf{A}_{0t}^{(1,1)} & & \\ & & \ddots & \\ & & & \mathbf{A}_{0t}^{(V,V)} \end{array} \right) \end{matrix}.$$

- $0 \leq v \leq c_2$

$$\mathbf{A}_{0t}^{(v,v)} = \mathbf{T}^0 \alpha \otimes (\otimes^t \mathbf{I}_{S_1}) \otimes \beta \otimes (\otimes^v \mathbf{I}_{S_2}).$$

- $c_2 \leq v \leq V$ and $t + v \leq V$

$$\mathbf{A}_{0t}^{(v,v)} = \mathbf{T}^0 \alpha \otimes (\otimes^t \mathbf{I}_{S_1}) \otimes \beta \otimes (\otimes^{c_2} \mathbf{I}_{S_2}).$$

- $c_2 \leq v \leq V$ and $t + v > V$

$$\mathbf{A}_{0t}^{(v,v)} = \mathbf{T}^0 \alpha \otimes (\otimes^{(V-v)} \mathbf{I}_{S_1}) \otimes (\otimes^{c_2} \mathbf{I}_{S_2}).$$

\mathbf{A}_{0c_1} is a special case because all servers at the first stage are busy. The next incoming customer has no impact on states of the two stages.

$$\mathbf{A}_{0c_1}^{(v,v)} = \mathbf{T}^0 \alpha \otimes (\otimes^{\min(c_1, V-v)} \mathbf{I}_{S_1}) \otimes (\otimes^{\min(c_2, v)} \mathbf{I}_{S_2}).$$

\mathbf{A}_{2t} is the transition matrix from the current level to the previous level, which is similar to \mathbf{A}_2 of the single-server case.

$$\mathbf{A}_{2t} = \begin{matrix} (t+1, 0) \\ (t+1, 1) \\ \vdots \\ (t+1, V-1) \\ (t+1, V) \end{matrix} \begin{pmatrix} (t, 0) & (t, 1) & (t, 2) & \dots & (t, V) \\ & \mathbf{A}_{2t}^{(0,1)} & & & \\ & & \mathbf{A}_{2t}^{(1,2)} & & \\ & & \ddots & \ddots & \\ & & & & \mathbf{A}_{0t}^{(V-1, V)} \end{pmatrix}.$$

- $0 \leq v < c_2$

There is no customer waiting outside and in front of the second stage.

$$\mathbf{A}_{2t}^{(v, v+1)} = \mathbf{I}_T \otimes \left(\sum_{h=t-1}^1 \mathbf{S}_1^0 \otimes (\otimes^h \mathbf{I}_{S_1}) + \sum_{h=1}^{t-1} (\otimes^h \mathbf{I}_{S_1}) \otimes \mathbf{S}_1^0 \right) \otimes (\otimes^v \mathbf{I}_{S_2}) \otimes \gamma.$$

- $c_2 \leq v \leq V$

In this situation, all servers at the second stage are busy. Customers have to wait in front of the second stage. A customer who is leaving the system will not change the states of the second stage.

$$\mathbf{A}_{2t}^{(v, v+1)} = \mathbf{I}_T \otimes \left(\sum_{h=\min(t, V-v)-1}^1 \mathbf{S}_1^0 \otimes (\otimes^h \mathbf{I}_{S_1}) + \sum_{h=1}^{\min(t, V-v)-1} (\otimes^h \mathbf{I}_{S_1}) \otimes \mathbf{S}_1^0 \right) \otimes (\otimes^v \mathbf{I}_{S_2}).$$

\mathbf{A}_{2c_1+1} is special because all servers are busy at the first stage for both the current level and the previous level.

- $0 \leq v < c_2$

In this situation, there is a customer waiting in front of the first stage. When a customer leaves the first stage, the released server begins to serve the waiting customer immediately.

$$\mathbf{A}_{2c_1+1}^{(v, v+1)} = \mathbf{I}_T \otimes (\oplus^{c_1} \mathbf{S}_0 \beta) \otimes (\otimes^v \mathbf{I}_{S_2}) \otimes \gamma.$$

- $c_2 \leq v \leq V - c_1$

In this situation, the first stage is the same as in the previous situation. Customers have to wait in front of the second stage because all the servers at the second stage are busy.

$$\mathbf{A}_{2c_1+1}^{(v,v+1)} = \mathbf{I}_T \otimes (\oplus^{c_1} \mathbf{S}_0 \beta) \otimes (\otimes^{c_2} \mathbf{I}_{S_2}) \otimes \gamma.$$

- $V - c_1 < v \leq V$

All servers at both stages are busy.

$$\mathbf{A}_{2c_1+1}^{(v,v+1)} = \mathbf{I}_T \otimes \left(\sum_{h=v}^1 \mathbf{S}_1^0 \otimes (\otimes^h \mathbf{I}_{S_1}) \right) + \sum_{h=1}^V (\otimes^h \mathbf{I}_{S_1}) \otimes \mathbf{S}_1^0 \otimes (\otimes^v \mathbf{I}_{S_2}).$$

2.6.4 Numerical Example 4

We construct a single-class SOQN with two service stages. The servers at each stage are parallel and identical. The inter-arrival time distribution is Gamma with a mean of 2 and C_X^2 of 1.2. The first stage has one server, and the distribution of its service time is exponential with a mean value of 1.5. The second stage has 2 parallel servers, and each server has a Erlang-2 distribution for service time with a mean of 3 and C_X^2 of 0.5.

Similar to the experiments conducted for single-server case, we conduct experiments by varying the number of vehicles V in the system. Table 2.8 shows the number of customers outside L_{eq} , the number of customers at the first stage L_1 , the number of customers at the second stage L_2 and the utilization of vehicles.

Results in Table 2.8 show that our method is relatively accurate for both high variance and low variance distributions.

Table 2.8 Results of two-stage SOQN with multiple servers

$V = 10$				
	L_{eq}	L_1	L_2	Utilization
A	0.05	0.91	2.04	29.5%
S	0.07	0.97	1.87	28.3%
error%	40.0	6.59	8.33	4.07
$V = 5$				
A	0.80	0.71	1.85	51.2%
S	0.65	0.67	1.70	47.4%
error%	18.8	5.63	8.33	7.42
$V = 3$				
A	2.58	0.42	1.66	69.2%
S	2.89	0.40	1.45	61.6%
error%	12.0	4.76	12.7	11.0

2.7 Single-Class SOQN with Multiple Stages of General Servers and General Arrival

2.7.1 Modified Decomposition-Aggregation Method

We can apply a decomposition-aggregation method to approximate a multi-stage SOQN as an equivalent two-stage SOQN. In this approximation process, the arrival process is PH distributed, one of the two stages is a single-load dependent server with exponentially distributed service time and the other is a multi-server stage with PH distributed service time.

Marie [10] discussed a method to solve non-product-form CQNs. We apply this method to get the load-dependent throughput of the CQN that contains the stages we want to aggregate.

The next modification of the decomposition-aggregation method relates to the representation of the load-dependent exponential distribution as a PH distribution. We can then apply the algorithm of the two-stage SOQN with PH distributions to analyze this equivalent two-stage SOQN. One possible solution is to view the exponential distribution as a PH distribution with one transient phase. Assume (α, \mathbf{T}) is the representation of a service stage with a load-dependent exponential distribution. According to (2.37), the initial probability of the transient state α is 1, the transition matrix \mathbf{T} is $-\mu(v)$ and the transition matrix of absorbed state \mathbf{T}^0 is $\mu(v)$. Here v is the number of customers being served, or the load of this stage.

Figure 2.15 shows the equivalent two-stage SOQN with a PH distributed arrival process, a load-dependent exponentially distributed service stage and a PH distributed service stage.

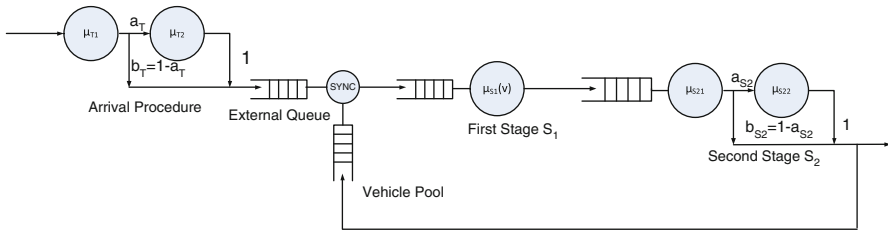


Fig. 2.15 Equivalent two-stage SOQN of multi-stage SOQN

2.7.2 Numerical Example 5

We conduct the experiment based on a four-stage, single-class SOQN with generally distributed servers and arrival processes. The distribution of the inter-arrival time is Erlang-2 with a mean of 1.5. The first stage has a single server with an exponentially distributed service process, where the mean of service time is 1. The second

stage has two identical parallel servers. The distribution of the service time of each server is Erlang-3 with a mean value of 2. The third stage has three identical parallel servers. The distribution of the service time of each server is Gamma with a mean value of 3 and C_X^2 of 1.2. The last stage has a single server. The service time has a Gamma distribution with a mean value of 1 and C_X^2 of 2. Table 2.9 shows the configuration of this four-stage SOQN.

Table 2.9 Four-stage single-class SOQN

Stage i	# of servers c_i	Mean value μ	SCV C_X^2
1	1	1	1
2	2	2	0.33
3	3	3	1.2
4	1	1	2

Table 2.10 Results of four-stage SOQN

$V = 12$					
A	1.02	0.79	0.56	1.20	0.63
S	1.14	0.84	0.50	1.24	0.60
error%	12.1	6.33	10.7	3.33	4.76
$V = 10$					
	L_{eq}	L_1	L_2	L_3	L_4
A	2.31	0.85	0.51	0.92	0.55
S	2.50	0.81	0.45	0.97	0.52
error%	8.23	4.71	11.7	5.43	5.45
$V = 7$					
A	27.3	0.55	0.31	0.43	0.32
S	25.8	0.58	0.28	0.46	0.30
error%	5.50	5.45	9.67	6.98	6.25

The results in Table 2.10 show that our proposed method works well for heavy, normal and lightly loaded networks. Again, our method is expected to improve for the low variance distributions. For example, the error is greater when our method is used to estimate the queue length of the second stage with an Erlang-3 distribution.

2.8 Multi-Class SOQN with Multiple Stages of General Servers and General Arrivals

In the real world, there are often more than one class of customers in a queueing network. For example, a manufacturing facility needs to process multiple types of products. Each type of product has its own product routing policy and processing times. So it is important to extend the algorithm to multi-class SOQN.

2.8.1 Aggregation Method

The algorithm to evaluate multi-class SOQN is inspired by Buitenhek et al. [3]. An aggregation method is presented to evaluate performance measures of a multi-class SOQN. The basic idea is to aggregate multiple classes of customers into one equivalent class of customers. After this aggregation, we can apply the algorithm of a single-class SOQN to get the performance measures of this equivalent class of customers. Finally, we can get the performance measures of each class of customers.

The number of customer classes in the multi-class SOQN is denoted as R . The r th class of customer has a generally distributed arrival process with an arrival rate λ_r and the SCV of the inter-arrival time is C_{Xr}^2 , where $r = 1, \dots, R$. Whitt [15] presented a set of formulae to aggregate multiple arrival processes into a compound arrival process with the arrival rate $\hat{\lambda}$ and the SCV of inter-arrival time \hat{C}_X^2 :

$$\begin{aligned}\hat{\lambda} &= \sum_{r=1}^R \lambda_r, \\ \hat{C}_X^2 &= \sum_{r=1}^R \frac{\lambda_r}{\hat{\lambda}} C_{Xr}^2.\end{aligned}\tag{2.46}$$

We use M to denote the number of service stages. For each class of customers, we assume it has its unique and deterministic route, which means a customer cannot change its routing within the network. The server in each stage has different service processes for different classes of customers, the service rate is $c_m \mu_{rm}$ and the SCV of service time is C_{Xrm}^2 / c_m , where c_m is the number of parallel servers at the m th stage, $r \in R_m$ and $m = 1, \dots, M$. Here, R_m is the set of classes of customers who visit the m th stage. Whitt [15] also presented a set of formulae to aggregate these service processes into one service process for the compound class:

$$\begin{aligned}\hat{\mu}_m &= \frac{\sum_{r \in R_m} \lambda_r}{\sum_{r \in R_m} \lambda_r / c_m \mu_{rm}}, \\ \widehat{C_{Xm}^2} &= \frac{\sum_{r \in R_m} \lambda_r (C_{Xrm}^2 / c_m + 1) / (c_m \mu_{rm})^2}{\sum_{r \in R_m} \lambda_r} \hat{\mu}_m^2 - 1.\end{aligned}\tag{2.47}$$

Each class of customers has its own deterministic path and each server has its own set of classes of customers. We aggregate these classes into one compound class. However, this compound class is different from the class in the single-class SOQN. In the single-class SOQN, the layout of service stages is tandem. In the multi-class SOQN, the compound class visits each stage with a probability. We use routing probability p_{ij} to denote the probability that a customer is transferred from the i th stage to the j th stage.

Another important parameter is the visit ratio vi_m , which is the mean number of visits of a customer to the m th stage.

$$vi_m = \frac{\widehat{\lambda}_m}{\widehat{\lambda}}, \quad (2.48)$$

where $\widehat{\lambda}_m = \sum_{r \in R_m} \lambda_r$ is the aggregated arrival rate at the m th stage.

The vi_m can also be expressed by routing probabilities,

$$vi_m = \sum_{j=1}^M vi_j p_{jm}, \text{ for } m = 1, \dots, M. \quad (2.49)$$

So far, we have already replaced the original multi-class SOQN with an equivalent single-class SOQN. However, it is still difficult to apply the decomposition-aggregation method we used in the single-class SOQN. In the single-class SOQN, we can divide the network into two subnetworks from any node. The average throughput rate of first subnetwork is equal to the arrival rate of the second subnetwork. This fact does not hold in the multi-class SOQN after aggregation because each stage has a certain visit ratio and these visit ratios may not be equal to 1. In other words, the throughput rate of the i th node may not be equal to the arrival rate of the j th node. Hence, we cannot divide the network into two parts.

Buitenhek et al. [3] suggested a simplified decomposition-aggregation method. We can simply aggregate all service stages and replace it with a load dependent stage. The problem is then reduced to a simple queue with general arrival process and a load dependent service stage. We can use the PH distribution to approximate the general distribution of the arrival process. Finally, the problem becomes a $PH/\mu(v)$ queue.

The performance measures of each single class are easy to obtain from the performance measures of the compound class. The external queue length of the r th class of customers is

$$L_{eqr} = L_{eq} \frac{\lambda_r}{\widehat{\lambda}}. \quad (2.50)$$

The expected number of customer of r th class at the m th stage L_{mr} can be divided into two parts. The first part is the expected number of customers of r th class in the m th service stage $\rho_{rm} = \frac{\lambda_r}{\mu_{rm}}$. The second part is the expected number of customers of the r th class in front of the m th service stage. It is known that the ratio of the expected number of r th class of customers in the queue should be the same as the ratio of the expected number of arrivals of r th class of customers.

$$L_{mr} = \rho_{rm} + (L_m - \sum_{r \in R_m} \rho_{rm}) \frac{\lambda_r}{\lambda_m}, \text{ for } r \in R_m. \quad (2.51)$$

2.8.2 Numerical Example 6

We construct an SOQN with six service stages (S_1, \dots, S_6) and five different classes of customers. Table 2.11 shows the deterministic routes of these five classes of customers.

Table 2.11 Routes of the five classes of customers in the SOQN

Class #	Route
1	$S_1 \rightarrow S_2 \rightarrow S_5 \rightarrow S_6$
2	$S_1 \rightarrow S_4 \rightarrow S_3 \rightarrow S_6$
3	$S_2 \rightarrow S_4 \rightarrow S_6$
4	$S_1 \rightarrow S_3 \rightarrow S_5$
5	$S_2 \rightarrow S_1 \rightarrow S_4 \rightarrow S_3 \rightarrow S_6 \rightarrow S_5$

The first set of experiments assumes that arrival processes of all the five classes are Poisson processes. The arrival rates are 0.6, 0.6, 0.8, 0.8, and 1, respectively. Each of the six service stages has only one server. Table 2.12 shows the mean and C_X^2 of service times of each stage for five classes. “N/A” means the particular class does not visit the corresponding service stage.

From Table 2.12, we can see that there are three kinds of distributions among the service times of the six stages: exponential distribution, Erlang-2 distribution and Coxian-2 distribution. These three kinds of distributions represent moderate, low and high variance cases respectively. Similar to the experiments conducted in previous sections, we vary the number of vehicles in the network from 18 to 25. The expected number of customers at the external queue and each stage for the aggregated class as well as the five classes are shown in Tables 2.13–2.15.

Results in these tables show that our approximation method works well when compared to the simulation models. Relative errors of expected number of customers in front of the six stages are very small. Although the relative error of expected number of customers outside is greater in the heavy load case, our method works well for moderate and light load cases.

In the second set of experiments, we examine the accuracy of our method for general arrival processes. We keep the arrival rates of five classes the same, but change the distribution type. The distribution of the arrival processes of classes 1 and 2 are Coxian-2 distributions with $C_X^2 = 2$. The distribution of the arrival processes of classes 3 and 4 are still exponential. The distribution of the arrival process of class 5 is Erlang-2. We conduct this set of experiments by changing the number of

Table 2.12 The first two moments of service times

Class #	S_1	S_2	S_3	S_4	S_5	S_6
1	0.2, 2	0.5, 0.8	N/A	N/A	0.4, 1.5	0.25, 0.5
2	0.3, 1.5	N/A	0.3, 1	0.25, 1	N/A	0.25, 0.5
3	N/A	0.3, 1	N/A	0.5, 1	N/A	0.5, 1
4	0.25, 1	N/A	0.4, 1	N/A	0.2, 3	N/A
5	0.15, 0.75	0.26, 1.2	0.2, 1	0.2, 1	0.2, 0.5	0.15, 2

Table 2.13 Result of 5-class Poisson arrival 6-stage single-server SOQN with 18 pallets

Aggregated	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	34.3	1.98	2.55	2.39	3.16	1.77	4.72
A	45.3	2.00	2.50	2.29	2.98	1.77	4.69
error%	24.4	1.00	2.00	4.37	6.04	0.00	0.64
Class 1	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	5.42	0.39	0.74	N/A	N/A	0.53	0.92
A	7.15	0.39	0.73	N/A	N/A	0.53	0.92
Class 2	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	5.42	0.45	N/A	0.60	0.75	N/A	0.92
A	7.15	0.45	N/A	0.58	0.71	N/A	0.92
Class 3	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	7.22	N/A	0.82	N/A	1.20	N/A	1.43
A	9.54	N/A	0.81	N/A	1.14	N/A	1.42
Class 4	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	7.22	0.55	N/A	0.88	N/A	0.55	N/A
A	9.54	0.56	N/A	0.85	N/A	0.55	N/A
Class 5	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	9.03	0.59	0.99	0.90	1.20	0.69	1.44
A	11.9	0.60	0.97	0.86	1.13	0.69	1.43

vehicles in the network from 20 to 25. Tables 2.16–2.18 show the expected number of customers at the external queue and each stage for the aggregated class and the five classes.

Similar to the first set of experiments, these tables show that our method works very well for estimating the expected number of customers in the network. If the load of this SOQN is moderate, the accuracy of our method is also good for estimating the expected number of customers outside.

From the previous two sets of experiments we notice that the queue lengths of stages 2, 4 and 6 are long. Therefore, we add some parallel servers in these stages and conduct the last set of experiments. We set the numbers of servers at stages 2 and 4 as 2, and the number of servers at stage 6 as 3. The number of vehicles in the network ranges from 7 to 10. The arrival processes of five classes are still generally

Table 2.14 Result of 5-class Poisson arrival 6-stage single-server SOQN with 22 pallets

Aggregated	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	8.87	2.04	2.73	2.57	3.24	1.88	5.25
A	9.92	2.06	2.78	2.39	3.16	1.82	5.21
error%	10.6	0.97	1.80	7.53	2.53	3.30	0.77
Class 1	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	1.40	0.40	0.78	N/A	N/A	0.56	1.03
A	1.57	0.40	0.79	N/A	N/A	0.55	1.02
Class 2	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	1.40	0.46	N/A	0.65	0.77	N/A	1.03
A	1.57	0.46	N/A	0.61	0.75	N/A	1.02
Class 3	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	1.87	N/A	0.88	N/A	1.23	N/A	1.57
A	2.09	N/A	0.90	N/A	1.20	N/A	1.56
Class 4	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	1.87	0.57	N/A	0.94	N/A	0.57	N/A
A	2.09	0.58	N/A	0.88	N/A	0.57	N/A
Class 5	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	2.33	0.61	1.06	0.98	1.24	0.73	1.62
A	2.61	0.62	1.09	0.90	1.20	0.71	1.60

Table 2.15 Result of 5-class Poisson arrival 6-stage single-server SOQN with 25 pallets

Aggregated	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	4.36	2.09	2.90	2.56	3.25	1.85	5.75
A	4.91	2.09	2.93	2.43	3.24	1.84	5.51
error%	11.2	0.00	1.02	5.35	0.31	0.54	4.00
Class 1	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	0.69	0.41	0.83	N/A	N/A	0.55	1.13
A	0.77	0.41	0.83	N/A	N/A	0.55	1.08
Class 2	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	0.69	0.47	N/A	0.65	0.78	N/A	1.13
A	0.77	0.47	N/A	0.61	0.77	N/A	1.08
Class 3	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	0.92	N/A	0.94	N/A	1.23	N/A	1.70
A	1.03	N/A	0.95	N/A	1.23	N/A	1.64
Class 4	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	0.92	0.58	N/A	0.94	N/A	0.58	N/A
A	1.03	0.58	N/A	0.90	N/A	0.57	N/A
Class 5	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	1.15	0.63	1.14	0.98	1.24	0.72	1.78
A	1.29	0.63	1.15	0.92	1.24	0.72	1.70

Table 2.16 Result of 5-Class general arrival 6-stage single-server SOQN with 20 pallets

Aggregated	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	16.8	2.03	3.66	2.33	3.23	1.83	4.93
A	20.7	2.04	3.66	2.35	3.08	1.80	4.98
error%	18.8	0.49	0.00	0.85	4.87	1.67	1.00
Class 1	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	2.65	0.40	1.02	N/A	N/A	0.55	0.97
A	3.27	0.40	1.02	N/A	N/A	0.54	0.98
Class 2	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	2.65	0.46	N/A	0.59	0.77	N/A	0.97
A	3.27	0.46	N/A	0.59	0.73	N/A	0.98
Class 3	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	3.54	N/A	1.19	N/A	1.23	N/A	1.49
A	4.36	N/A	1.19	N/A	1.18	N/A	1.50
Class 4	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	3.54	0.57	N/A	0.86	N/A	0.57	N/A
A	4.36	0.57	N/A	0.87	N/A	0.56	N/A
Class 5	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	4.42	0.61	1.45	0.88	1.23	0.71	1.51
A	5.45	0.61	1.45	0.89	1.17	0.70	1.53

distributed and have the same parameters as in the second set of experiments. Tables 2.19–2.21 show the expected number of customers at the external queue and each stage for the aggregated class and the five classes.

Similar to the first two sets of experiments, these tables show that our method still works well for estimating the expected number of customers in the network. However, the accuracy for estimating the expected number of customers in the external queueing is not very good. There are two sources of error. The first source is from the aggregation process of multiple classes of customers. The second one is from the aggregation process of parallel servers.

2.9 Conclusions

In this chapter, we discuss how to model the automated warehouse by using semi-open queueing networks. We present two algorithms for solving SOQN with exponential interarrival and service times. The first method is the state space based method. The key point of this method is to truncate the state space of two-stage, single-class SOQN at a certain level, then estimate the steady state probabilities. However, if the number of resources (i.e., the number of vehicles in AVS/RS) is large, this method is time consuming because the size of the state space is large. The

Table 2.17 Result of 5-class general arrival 6-stage single-server SOQN with 22 pallets

Aggregated	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	10.7	2.07	4.00	2.40	3.18	1.79	5.14
A	11.2	2.07	3.80	2.39	3.17	1.83	5.23
error%	4.30	0.00	5.26	0.42	0.32	2.19	1.72
Class 1	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	1.69	0.40	1.10	N/A	N/A	0.54	1.01
A	1.77	0.40	1.05	N/A	N/A	0.55	1.03
Class 2	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	1.69	0.46	N/A	0.61	0.76	N/A	1.01
A	1.77	0.46	N/A	0.60	0.76	N/A	1.03
Class 3	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	2.26	N/A	1.31	N/A	1.21	N/A	1.54
A	2.36	N/A	1.24	N/A	1.21	N/A	1.57
Class 4	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	2.26	0.58	N/A	0.89	N/A	0.56	N/A
A	2.36	0.58	N/A	0.88	N/A	0.57	N/A
Class 5	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	2.82	0.62	1.59	0.91	1.21	0.70	1.58
A	2.95	0.62	1.51	0.90	1.21	0.71	1.61

Table 2.18 Result of 5-class general arrivals 6-stage single server SOQN with 25 pallets

Aggregated	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	5.51	2.10	3.98	2.68	3.28	1.81	5.72
A	5.68	2.10	3.96	2.44	3.26	1.85	5.55
error%	3.00	0.00	0.51	9.84	0.61	2.16	3.06
Class 1	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	0.87	0.41	1.10	N/A	N/A	0.54	1.12
A	0.90	0.41	1.09	N/A	N/A	0.55	1.09
Class 2	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	0.87	0.47	N/A	0.68	0.78	N/A	1.12
A	0.90	0.47	N/A	0.62	0.78	N/A	1.09
Class 3	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	1.16	N/A	1.30	N/A	1.24	N/A	1.70
A	1.20	N/A	1.29	N/A	1.24	N/A	1.65
Class 4	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	1.16	0.59	N/A	0.98	N/A	0.56	N/A
A	1.20	0.59	N/A	0.90	N/A	0.58	N/A
Class 5	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	1.45	0.63	1.59	1.03	1.25	0.70	1.77
A	1.49	0.63	1.58	0.93	1.25	0.72	1.72

Table 2.19 Result of 5-class general arrival 6-stage multiple-server SOQN with 7 pallets

Aggregated	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	36.1	1.66	0.66	1.89	0.62	1.44	0.44
A	30.4	1.64	0.65	1.85	0.61	1.47	0.41
error%	18.8	1.22	1.54	2.16	1.64	2.04	7.32
Class 1	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	5.70	0.32	0.27	N/A	N/A	0.45	0.07
A	4.80	0.32	0.26	N/A	N/A	0.46	0.06
Class 2	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	5.70	0.38	N/A	0.48	0.12	N/A	0.07
A	4.80	0.38	N/A	0.47	0.12	N/A	0.06
Class 3	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	7.60	N/A	0.19	N/A	0.36	N/A	0.29
A	6.40	N/A	0.19	N/A	0.35	N/A	0.28
Class 4	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	7.60	0.47	N/A	0.72	N/A	0.44	N/A
A	6.40	0.46	N/A	0.70	N/A	0.45	N/A
Class 5	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	8.50	0.49	0.20	0.70	0.15	0.55	0.01
A	8.00	0.48	0.20	0.68	0.14	0.56	0.00

Table 2.20 Result of 5-class general arrival 6-stage multiple-server SOQN with 8 pallets

Aggregated	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	7.43	1.76	0.69	1.95	0.64	1.53	0.43
A	9.71	1.73	0.66	1.97	0.62	1.54	0.41
error%	23.5	1.73	4.55	1.02	3.23	0.65	4.88
Class 1	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	1.17	0.34	0.27	N/A	N/A	0.47	0.07
A	1.53	0.34	0.27	N/A	N/A	0.48	0.06
Class 2	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	1.17	0.40	N/A	0.49	0.12	N/A	0.07
A	1.53	0.40	N/A	0.50	0.12	N/A	0.06
Class 3	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	1.56	N/A	0.20	N/A	0.36	N/A	0.29
A	2.04	N/A	0.19	N/A	0.36	N/A	0.28
Class 4	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	1.56	0.50	N/A	0.74	N/A	0.47	N/A
A	2.04	0.49	N/A	0.74	N/A	0.47	N/A
Class 5	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	1.96	0.52	0.21	0.72	0.15	0.59	0.01
A	2.56	0.51	0.20	0.73	0.15	0.59	0.00

Table 2.21 Result of 5-class general arrival 6-stage multiple-server SOQN with 10 pallets

Aggregated	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	3.20	1.83	0.69	2.13	0.63	1.72	0.44
A	3.18	1.87	0.67	2.17	0.63	1.65	0.41
error%	0.63	2.14	2.98	1.84	0.00	4.24	7.32
Class 1	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	0.51	0.36	0.27	N/A	N/A	0.52	0.07
A	0.50	0.36	0.27	N/A	N/A	0.50	0.06
Class 2	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	0.51	0.42	N/A	0.54	0.12	N/A	0.07
A	0.50	0.42	N/A	0.55	0.12	N/A	0.06
Class 3	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	0.67	N/A	0.20	N/A	0.36	N/A	0.29
A	0.67	N/A	0.20	N/A	0.36	N/A	0.28
Class 4	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	0.67	0.51	N/A	0.80	N/A	0.53	N/A
A	0.67	0.53	N/A	0.81	N/A	0.51	N/A
Class 5	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	0.84	0.54	0.21	0.80	0.15	0.67	0.01
A	0.84	0.56	0.21	0.81	0.15	0.64	0.00

second method is the matrix geometric method, which develops a generator matrix with repetitive structures that can be solved exactly via an iterative procedure. We then solve the two-stage SOQN with generally distributed service time and arrival processes. The structure of the generator matrix is discussed in detail. We also discuss the structure of the generator matrix in detail, especially how to extend the single-server case to the multi-server case. We then extend the two-stage SOQN to multi-stage SOQN by applying the decomposition-aggregation method. We apply Marie's method for the general distributions. Finally, we discuss the approximation algorithm for multiple-class SOQN. The basic idea is to aggregate multiple classes into a single equivalent class and then aggregate the network into a single load-dependent stage by using Marie's method.

References

1. Bellman, R. (1960). *Introduction to matrix analysis*. New York: McGraw-Hill.
2. Bird, J. (2007). *Electrical circuit theory and technology*. Newnes, Burlington, MA.
3. Buitenhek, R., Houtum, G., & Zijm, H. (2000). An mva-based solution procedures for open queueing networks with population constraints. *Annals of Operations Research*, 93, 15–40.
4. Chandy, K., Herzog, U., & Woo, L. (1975). Parametric analysis of queueing networks. *IBM Journal of Research and Development*, 19, 36–42.

5. Cox, D. (1955). A use of complex probabilities in the theory of stochastic processes. *Proceedings of Cambridge Philosophical Society*, 51, 313–319.
6. Erlang, A. (1917). Solution of some problems in the theory of probabilities of some significance in automatic telephone exchanges. *Post Office Electrical Engineering's Journal*, 10, 189–197.
7. Heragu, S., Cai, X., Krishnamurthy, A., & Malmberg, C. (2011). Analytical models for analysis of automated warehouse material handling systems. *International Journal of Production Research*, 49, 6833–6861.
8. Heragu, S., & Srinivasan, M. (2011). Analysis of manufacturing systems via semi-open queueing networks. *International Journal of Production Research*, 49, 295–319.
9. Jia, J., & Heragu, S. (2009). Solving semi-open queueing networks. *Operations Research*, 57(2), 391–401.
10. Marie, R. (1980). Calculating equilibrium probabilities for $\lambda(n)/c_k/1/n$ queues. *ACM SIGMETRICS Performance Evaluation Review*, 9, 117–125.
11. Mayhugh, J. O., & McCormick, R. E. (1968). Steady-state solution of the queue $m/ek/r$. *Management Science*, 14, 692–712.
12. Neuts, M. (1981). *Matrix-geometric solutions in stochastic models: An algorithmic approach*. Baltimore: John Hopkins University Press.
13. Neuts, M. (1995). *Matrix-geometric solutions in stochastic models: An algorithmic approach*. Courier Dover Publications, Mineola, NY.
14. Sauer, C. & Chandy, K. (1981). *Computer systems performance modeling*. Englewood Cliffs: Prentice-Hall.
15. Whitt, W. (1983). The queueing network analyzer. *The Bell System Technical Journal*, 62, 2779–2815.

Handbook of Stochastic Models and Analysis of
Manufacturing System Operations

Smith, J.M.; Tan, B. (Eds.)

2013, XXVIII, 373 p., Hardcover

ISBN: 978-1-4614-6776-2