APPROXIMATE MVA ALGORITHMS FOR OPEN QUEUEING NETWORKS WITH RESTRICTED CAPACITY

Ronald Buitenhek, Geert-Jan van Houtum and Henk Zijm

Production and Operations Management Group,
Faculty of Mechanical Engineering,
University of Twente,
P.O. Box 217, 7500 AE Enschede, The Netherlands,
Phone: +31 53 489 25 54
Fax: +31 53 489 34 71
E-mail: r.buitenhek@wb.utwente.nl

March 28, 1997

Abstract

We propose the application of approximate Mean Values Analysis algorithms for the analysis of Open Queueing Networks with Restricted Capacity. This consitutes a modification of the method as described by Dallery [6] (for single classes) and by Perros, Dallery and Pujolle [11] (for multiple classes served by exponential stations). The modified method is computationally more attractive, because it saves on iterations. We describe our method extensively for the single class case with exponential servers. Furthermore we show how the method can be extended to deal with more general cases, such as networks with non-exponential servers and/or multiple customer classes.

1 Introduction

We consider the performance evaluation of *Open Queueing Networks with Restricted Capacity* (OQNs-RC). Consider an OQN-RC with a *single job class*. Such a network consists of an inner system with service stations and an external queue. The restricted capacity concerns the number of jobs inside the inner system: there can only be N jobs in the inner system at the same time. Jobs that arrive at the system enter the inner system if there are less than N jobs there. An arriving job that finds the inner system full, joins the external queue at the back. Once a job is inside

the inner system, it follows its Markovian routing until it has completed its final operation. Upon the external departure of a job the first in line job of the external queue is allowed to enter the inner system.

The capacity constraint of the inner system can be represented by a set of *tokens*. To be allowed to enter the inner system a job must first get a token. The job keeps the token during its stay in the inner system and releases it upon its departure.

An obvious generalization of the single class OQN-RC is the multiclass OQN-RC. For the multiclass OQNs-RC, we distinguish between three cases. Either each job class has a set of dedicated tokens, or there is a common set tokens that are shared among the classes, or there are subsets of classes that share tokens while those tokens are dedicated to the subset.

The OQN-RC can be used for the performance evaluation of computer systems and manufacturing systems. Consider, for example, a Flexible Manufacturing System (FMS) in which jobs need to be clamped on a pallet while they are processed at the work stations. The work stations are then modeled by the service stations and the pallets are modeled by the tokens. Modeling an FMS by an OQN-RC may be more desirable than by a Closed Queueing Network if the assumption that a completed part can always be replaced by a new part is not always fulfilled.

The behavior of the *single class* OQN-RC with *exponential* servers is described by a Markov process with the states $(n_E, \vec{n}) = (n_E, (n_1, \ldots, n_M))$, where n_E denotes the number of jobs in the external queue and n_m denotes the number of jobs at station m and M denotes the number of service stations. It can be easily seen that this Markov process is a Quasi Birth and Death (QBD) process: there can only be transitions from the state $(n_E, (n_1, \ldots, n_M))$ to states with the number in the external queue equal to $n_E - 1$, n_E or $n_E + 1$. Unfortunately, the stationary distribution of the Markov process does not possess a product-form solution. However, the Markov process can be solved by numerical methods. One can use the matrix-geometric approach (see [10]) or aggregation-disaggregation techniques (see [13] and [14]) if the number of work stations M and the number of tokens N are not too large. For the larger values of M and N, approximate solution procedures have been developed to obtain the desired performance measures.

A well-known approximate solution procedure is the *aggregation method* as already proposed by Avi-Itzhak and Heyman in 1972. This method basically considers the aggregate Markov process with the states $n = n_E + n_I$, where $n_I = n_1 + \ldots + n_M$. The downcrossing rate from the state n is estimated by the throughput of the CQN that is constructed from the OQN-RC by properly redirecting the external departures, and that has the population n_I. If there are non-exponential servers, approximate Mean Value Analysis algorithms can be used to approximate the requried values of the throughput. A generalization of the aggregation method to networks with multiple job classes and shared tokens is described in Buitenhek, van Houtum and Zijm [4]. The direct generalization for the case with dedicated pallets yields a multi-dimensional random walk that is difficult to solve. Brandwajn [1], Lazowska and Zahorjan [8], and Thomasian and Bay [15] give generalizations that involve extra approximations to keep the computational effort tractable. In [3] we have given a solution procedure for the two-dimensional case of the direct generalization, which is based on the matrix-geometric approach.

Another approximation method for the single class OQN-RC with general servers was proposed by Dallery [6] in 1990. His method is based on the idea that the OQN-RC is equivalent to a CQN if the roles of the jobs and the tokens are interchanged. The equivalent CQN has a population equal to the number of tokens and consists of the same stations as the OQN-RC plus an extra station modeling the external queue. This CQN is then analyzed by an extension of the method of Marie (see [9]) as follows. The external queue and all non-exponential server stations are, in an iterative procedure, replaced by load-dependent exponential servers. The resulting product-form CQN is solved to obtain the internal queue lengths. The number of jobs in the external queue is approximated by solving a random walk with as upcrossing rates the external arrival rate of the jobs and as downcrossing rate the load-dependent arrival rate of the tokens as obtained by Marie's method. The method of Dallery and the aggregation method yield identical results if the inner system of the OQN-RC consists of only exponential servers. A generalization for the multiclass OQN-RC with dedicated tokens is described in Perros, Dallery and Pujolle [11], albeit that they restrict themselves to networks with a BCMP-type inner system.

In this paper, we propose to apply (approximate) Mean Value Analysis algorithms to solve the CQN. For the general single class OQN-RC this provides an algorithm that avoids the iterations as required by the method of Marie to find the load-dependent service rates of the replacing service stations. Furthermore, this allows us to form a framework of solution procedures for the all generalizations mentioned above.

The paper is organized as follows. In Section 2 we describe our procedure for the single class OQN-RC with exponential servers. In Section 3 we describe how the procedure must be modified for the different generalizations and which extra approximations are involved. Finally, we give some concluding remarks in Section 4.

2 The single class OQN-RC with exponential service times and Poisson arrivals

2.1 The queueing network

Consider the OQN-RC with a single job class. A schematic representation of the single class OQN-RC is given in Figure 1. The OQN-RC consist of two elements: the *external queue* and the *inner system*. The external queue regulates the capacity constraint of the network or availability of the tokens. It exists of a queue for the jobs and a queue for the tokens. The inner system consists of the M service stations. Furthermore, one may distinguish two flows of entities: the flow of jobs and the flow of the N tokens. The jobs arrive at the network according to a Poisson process with rate λ. A job moves through the network according to a Markovian routing, characterized by the routing matrix P. At service station m a job receives service. The corresponding service time has an exponential distribution with rate μ_m.

The regulation of the capacity constraint is as follows. If upon arrival a job

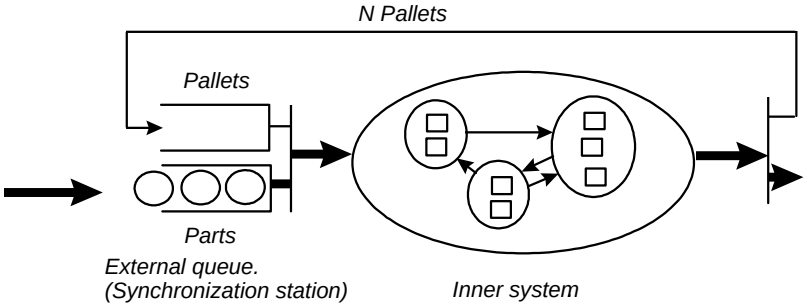

Figure 1: *The open queueing network with restricted capacity.*

finds a token in the external queue, the job and the token synchronize and together enter the inner system. Otherwise, the job joins the external queue at the back. Upon an external departure of a job, the token that was synchronized with that job returns to the external queue. The first in line job in the external queue (if any) now synchronizes with the token or (if there is no job in the external queue) the token waits until a job arrives. The synchronization of the jobs and the tokens ensures that the number of jobs in the inner system is never larger than the number of tokens or the capacity of the network. In the sequel, we also refer to the external queue as the *synchronization station*.

The characteristics of the single class OQN-RC with exponential servers and Poisson arrivals can now be listed as follows:

1. There is *one* job class.
2. There are M *exponential* service stations; that is, at service station m, the service time has an exponential distribution with rate μ_m, $m = 1, \ldots, M$.
3. There is *one* synchronization station. The jobs arrive at the synchronization station according to a *Poisson* process with rate λ. Identify the synchronization station by the index $M + 1$.
4. Jobs have a Markovian routing that is characterized by the routing matrix P.
5. There are N tokens.

2.2 The solution procedure

We propose to solve the single class OQN-RC with exponential servers and Poisson arrivals with respect to the mean queue lengths by a procedure similar to the method of Dallery. Consider the CQN view of the OQN-RC, in which the tokens are the customers. Then our procedure is given by the following three steps.

1. Replace the synchronization station by a load-dependent exponential server. Let the service rates of the replacing station be such that the throughput $TH(N)$ of the CQN is equal to the external arrival rate λ of the jobs.

2. Solve the resulting product-form CQN to find the mean number of tokens (and hence the mean number of jobs) at the service stations. To this aim, use a *Mean Value Analysis* (MVA) algorithm.

3. Find the mean number of jobs in the external queue by considering the external queue in isolation and applying a Markovian analysis.

These steps only differ from the method of Dallery in that we insist on the application of the MVA algorithm. However, the difference between the methods and the significance of our method become clear when we consider the generalizations in Section 3. Below we first explain step 1 and step 3 of the steps that constitute the procedure. An MVA algorithm as used in step 2 can be found in the textbook of Buzacott and Shanthikumar [5] or in the original paper by Reiser and Lavenberg [12]. We present (approximate) MVA algorithms in the Appendix, that can deal with all cases in this paper. The algorithms combine parts of the algorithm of Buzacott and Shanthikumar and approximations for the $M|G|c$ queue (for a comparison of AMVA algorithms, see [2]).

In step 1, we modify the original network in such a way that the behavior is modified as little as possible while we are able to solve the resulting network. The load-dependent service rate $\nu_{M+1}(n)$ of the synchronization station is chosen equal to λ for $n > 1$: if there are tokens at the synchronization station then the first in line token has to wait for an exponential time with rate λ until it synchronizes with a job and leaves the station. We set $\nu_{M+1}(1)$ such that the throughput $TH(N)$ of the CQN with N tokens is equal to the external arrival rate of the jobs λ. Thus, the resulting CQN is consistent with the original OQN-RC.

The detemination of $\nu_{M+1}(1)$ proceeds as follows. Consider the CQN in which the synchronization station has been replaced by a load-dependent exponential server. This network has a product-form stationary distribution and therefore Norton's theorem can be applied. In this case Norton's theorem states that the behavior of the synchronization station is described by the birth and death process with states $n = 0, 1, \ldots, N$ (denoting the number of tokens at the station). The downcrossing rates are equal to the service rates $\nu_{M+1}(n)$ and the upcrossing rates are given by the throughput values $TH^{C_{M+1}}(N-n)$ of the complement network of the synchronziation station, C_{M+1}. Therefore, for the marginal probabilities $p_{M+1}(0)$ and $p_{M+1}(1)$ that there are 0 or 1 job(s) at the synchronization station, respectively, it must hold that

$$\frac{p_{M+1}(0)}{p_{M+1}(1)} = \frac{\nu_{M+1}(1)}{TH^{C_{M+1}}(N)} \ . \tag{1}$$

Furthermore, the number of jobs that arrive at the synchronziation station per time unit must be equal to the number of jobs that depart from the synchronization station per time unit:

$$\lambda = p_{M+1}(1)\ \nu_{M+1}(1) + [1 - p_{M+1}(0) - p_{M+1}(1)]\ \ \lambda \ , \tag{2}$$

which can be rewritten as

$$\nu_{M+1}(1) = \lambda\ \left(1 + \frac{p_{M+1}(0)}{p_{M+1}(1)}\right). \tag{3}$$

Combining Equations (1) and (3) yields:

$$\nu_{M+1}(1) = \frac{1}{q}\ \lambda\ ,$$

where q is given by

$$q = 1 - \frac{\lambda}{TH^{C_{M+1}}(N)}\ . \tag{4}$$

In Step 3, we analyze the synchronization station in isolation in order to find the mean number of jobs present there. Consider the Markov process with the states (n_t, n_j), where n_t denotes the number of tokens and n_j denotes the number of jobs at the synchronization station. The arrival rate of the jobs is λ and, according to Norton's theorem, the arrival rate of the tokens is $TH^{C_{M+1}}(N - n_t)$. Since there can never be jobs and tokens at the synchronzation station at the same time (they would have been synchronized), the only feasible states are $(n_t, 0)$ for $n_t = 0, \ldots, N$, and $(0, n_j)$, for $n_j \geq 0$. The Markov process is thus a random walk and can be solved by solving the local balance equations (5) and (6).

$$\lambda\ p(n_t, 0) = TH^{C_{M+1}}(N - n_t + 1)\ p(n_t - 1, 0), \quad \text{for } n_t = 1, \ldots, N, \tag{5}$$

$$\lambda\ p(0, n_j) = TH^{C_{M+1}}(N)\ p(0, n_j), \quad \text{for } n_j > 0. \tag{6}$$

From Equations (5) and (6) we find the probabilies $p(0, n_j)$, for $n_j > 0$, and from them we find the mean number of jobs in the external queue EL_{ext}.

$$EL_{ext} = \sum_{n_j=1}^{\infty} n_j\ p(0, n_j) = p(0, 0)\ \frac{\rho}{1 - \rho}\ , \tag{7}$$

where $\rho = \lambda\ /\ TH^{C_{M+1}}(N)$ and $p(0, 0)$ is given by

$$p(0, 0) = \prod_{k=1}^{N} \frac{\lambda}{TH^{C_{M+1}}(k)}\ p(N, 0)$$

and

$$p(N, 0) = \left[\sum_{n_j=1}^{\infty} \prod_{k=1}^{N-n_j} \frac{\lambda}{TH^{C_{M+1}}(k)} + \prod_{k=1}^{N} \frac{\lambda}{TH^{C_{M+1}}(k)}\ \frac{\rho}{1 - \rho} \right]^{-1}$$

We recall that the only approximation made in the procedure described above is constituted by the replacement of the synchronziation by a load-dependent exponential server.

3 Generalizations

The procedure described in Section 2 can be generalized to deal with several other situations. Generalizations mainly concern the service times and the number of job classes. First, the service times at the service station may be non-exponential. This leads to an inner system and a CQN (after replacing the synchronization station) which are not of the product-form type. Second, there may be multiple job

classes. In this case, the inner system and the resulting CQN are only product-form (BCMP-type) networks if all service times are exponential *and* the service rates of the different classes at one stations are the same. We therefore distinguish between BCMP-type networks and general (non-BCMP-type) networks. Furthermore, we have to distinguish between networks with dedicated tokens and networks with shared tokens. Finally, we consider the hybrid case, in which subsets of the classes share a set of tokens that are dedicated to that subset. The possible combinations of these features of the networks are summarized in Table 1.

	single class	multiple classes		
		shared tokens	dedicated tokens	hybrid case
BCMP type network	Case 1	Case 3	-	-
General type network	Case 2	Case 4	Case 5	Case 6

Table 1: The diiferent cases to be distinguished.

Case 1 has been dealt with in Section 2. The other cases are discussed in this section. We show how the procedure must be modified for these cases and what extra approximations are introduced. Furthermore, we indicate which methods have been described in the literature to deal with the specific case.

It should be noted that in all cases we assume Poisson arrival process(es). This assumption can also be relaxed, as was shown by Di Mascolo [7].

3.1 Case 2

Consider the single class OQN-RC with *non-exponential* service stations. For this case, only characteristic 2 in Subsection 2.1 is violated. This characteristic becomes:

2. There are M *general* service stations; that is, at service station m, the service time has an general distribution with known first two moments ES_m and ES_m^2, $m = 1, \ldots, M$.

The fact that the inner system is not BCMP-type, has two implications. First, the complement network of the synchronization station is not a product-form network. This means that we cannot apply Norton's theorem in Step 1, and if we choose the load-dependent service rates in the same way as for Case 1, this does not guarantee that the throughput of the network is equal to the external arrival rate. Second, the network resulting after the synchronization station has been replaced by a load-dependent exponential server is not a product-form network. Therefore, an MVA algorithm does not yield exact results.

We propose to generalize the procedure for this case by applying an approximate MVA algorithm in Step 2, while Step 1 remains unchanged. Thus, this introduces two approximations. The approximate MVA algorithm for this case is formulated in Appendix A.2.

Other methods that can approximately solve this case are the method of Dallery and the traditional aggregation method when approximate MVA is used to compute

the throughput of the complement network of the synchronization station. The advantage of the procedure presented here when compared to the method of Dallery is that its complexity is not affected by the presence of (and, more specifically, the number of) non-exponential servers. The procedure has the same complexity as the aggregation method. The quality of these three methods highly depends on the approximation of the external queue length (see [4]). Since the three methods use the same way to approximate this quantity, their approximations are of about the same quality.

3.2 Case 3

Consider the *multiclass* OQN-RC with *shared tokens*. Let the inner system be a BCMP-type network; that is, at station m the different classes have the same exponential service time with rate μ_m. For this network characteristics 1, 3 and 4 are violated. These characteristic become:

1. There are $R > 1$ job classes.

3. There is *one* synchronization station. Class r jobs arrive at the synchronization station according to a *Poisson* process with rate $\lambda^{(r)}$, $r = 1, \ldots, R$. Identify the synchronization station by the index $M + 1$.

4. Class r jobs have a Markovian routing that is characterized by the routing matrix $P^{(r)}$, $r = 1, \ldots, R$.

Note, that due to the shared tokens, there is still only *one synchronization station*.

This network can be transformed into an equivalent single chain network due to the facts that all job classes visit the synchronization station and that they share the tokens. Within the chain one distinguishes between the different classes. A token changes class at the synchronization station. Denote by $\lambda^{(a)} = \sum_{r=1}^{R} \lambda^{(r)}$ the total arrival rate. With probability $\lambda^{(r)} \ / \ \lambda^{(a)}$, any job arriving at the synchronization station, leaves this station as a class r job. In order to be able to apply an MVA algorithm, we need to know the visit ratio's $V_m^{(a)}$ of an arbitrary job to station m. (We assumed that $\mu_m^{(r)} = \mu_m$, for $m = 1, \ldots, M$. Thus, $\mu_m^{(a)} = \mu_m$.) Denote by $V_m^{(r)}$ the visit ratio of class r to station m. The visit ratio's follow directly from the routing matrices $P^{(r)}$ and satisfy

$$V_{M+1}^{(r)} = 1 \text{ and } V_m^{(r)} = \frac{\lambda_m^{(r)}}{\lambda^{(r)}} \text{ for } m = 1, \ldots, M,$$

where $\lambda_m^{(r)}$ denotes the arrival rate of class r jobs at station m. For the visit ratio's of an arbitrary job it holds that

$$V_m^{(a)} = \frac{\sum_{r=1}^{R} \lambda_m^{(r)}}{\sum_{r=1}^{R} \lambda^{(r)}} = \frac{\lambda_m^{(a)}}{\lambda^{(a)}} \ .$$

These visit ratio's may be renormalized to facilitate the application of the MVA algorithm. In the main procedure for this case we can now perform the computations

for the aggregate class a, and only afterwards obtain the class dependent quantities by weighing the aggregate quantities with the class dependent arrival rates.

It is important to stress that the transformation described here does not change the behavior of the network. The original network and the transformed network are exactly identical and thus the transformation does not introduce an approximation.

Once we have obtained a single chain network, the procedure to find the mean number of jobs in the external queue and the mean number of jobs in the inner system is basically the same as the procedure used for the single class OQN-RC. Here, we need one additional step to compute the class dependent quantities. For clarity, we give the procedure in full.

1. Replace the synchronization station by a load-dependent exponential server. The service rates only depend on the *total* number of tokens present. Let the service rates of the replacing station be such that the throughput $TH(N)$ of the CQN is equal to the *total* external arrival rate $\lambda^{(a)}$ of the jobs.

2. Solve the resulting product-form CQN to find the mean number of tokens (and hence the mean number of jobs) at the service stations. To this aim, use a *Mean Value Analysis* (MVA) algorithm for CQN's with a single class (the aggregate class). This yields the mean total number of jobs at each service station ($EL_m^{(a)}$).

3. Compute the mean *total* number of jobs in the external queue ($EL_{ext}^{(a)}$) by considering the external queue in isolation and applying a Markovian analysis.

4. Compute the mean number of class r jobs in the external queue and the mean number of class r jobs at station m.

$$EL_{ext}^{(r)} = \frac{\lambda^{(r)}}{\lambda^{(a)}}\ EL_{ext}^{(a)} \quad \text{and} \quad EL_m^{(r)} = \frac{\lambda_m^{(r)}}{\lambda_m^{(a)}}\ EL_m^{(a)}\ .$$

In order to obtain the load-dependent service rate of the replacing station in Step 1, we need to solve the single chain CQN without the synchronization station. The relative visit ratio's of the aggregate class in this network are equal to the visit ratio's $V_m^{(a)}$ (it is of course possible to renormalize these values). The Markovian analysis to be performed in Step 3 is identical to that of Step 3 in Subsection 2.2.

3.3 Case 4

Consider the multiclass OQN-RC with shared tokens and *general* servers. The main difference between this case and Case 3 is that the CQN that results after the synchronization station has been replaced by a load-dependent exponential server is not a product-form network. This prevents that the exact single class MVA algorithm yields the exact results. Therefore, we propose to apply an approximate single chain multiple class MVA algorithm, that directly takes into account the fact that there are multiple classes. Thus, we end up with an approximate MVA algorithm that has to perform the same number of iterations as the algorithm in Case 3. The approximate MVA algorithm for Case 4, however, needs more variables

and more iteration steps per iteration. For example, there is a variable for the lead time of a class r job at station m given that there is a total number of n tokens in the system. The resulting approximate MVA algorithm is given in Appendix A.3.

In order to be able to apply the suggested AMVA algorithm, we need to relate the visit ratio's of all classes to each other. This can be achieved by setting $V_{M+1}^{(a)} = 1$. The class dependent visit ratio's are then given by

$$V_m^{(r)} = \frac{\lambda_m^{(r)}}{\lambda^{(r)}} \ .$$

The resulting procedure for this case is given by the following three steps.

1. Replace the synchronization station by a load-dependent exponential server. The service rates only depend on the total number of tokens present. Let the service rates of the replacing station be such that the throughput $TH(N)$ of the CQN is equal to the *total* external arrival rate $\lambda^{(a)}$ of the jobs.

2. Approximately solve the resulting CQN to find the mean number of jobs per class at the service stations. To this aim, use an *approximate MVA* algorithm for CQN's with a single clain (the aggregate class) and multiple classes (see Appendix A.3). This yields the mean number of jobs per class at each service station.

3. Compute the mean total number of jobs in the external queue ($EL_{ext}^{(a)}$) by considering the external queue in isolation and applying a Markovian analysis.

4. Compute the mean number of class r jobs in the external queue:

$$EL_{ext}^{(r)} = \frac{\lambda^{(r)}}{\lambda^{(a)}} \ EL_{ext}^{(a)} \ .$$

For this case we developed three algorithms based on the aggregation method in [4]. The third of these algorithms is very similar to the procedure proposed here, in the sense that it also creates a single chain with multiple classes. The efficiency of that algorithm and this procedure both mainly depend on the MVA algorithm. The others are less efficient but slightly more accurate than.

3.4 Case 5

Consider the multiclass OQN-RC with dedicated tokens. All characteristics as listed in Subsection 2.1 are now violated. The characteristics for this case are listed below.

1. There are R job classes.

2. There are M *general* service stations; that is, at service station m, the service time of class r has a general distribution with known first two moments $ES_m^{(r)}$ and $E\left(S_m^{(r)}\right)^2$, $m = 1, \ldots, M$.

3. There are R synchronization stations; one for each class. The synchronization station for class r is identified by the index $M+r$. Class r jobs arrive according to a Poisson arrival process with rate $\lambda^{(r)}$, $r = 1, \ldots, R$.

4. Class r jobs have a Markovian routing that is characterized by the routing matrix $P^{(r)}$, $r = 1, \ldots, R$.

5. There are $N^{(r)}$ dedicated tokens for class r, $r = 1, \ldots, R$.

It is important to notice two things. First, notice that there are $R > 1$ synchronization stations. This means that the complement network of a synchronization station contains $R - 1 > 0$ other synchronization stations. Such a complement network cannot be a BCMP-type network. For our procedure, it is therefore of no use to distinguish between BCMP-type inner systems and non-BCMP-type inner systems. Second, notice that each of the synchronization stations is only visited by one class; station $M + r$ is visited by class r only. This allows us to replace the synchronziation stations by load-dependent exponential servers, where the load is given by the number of class r jobs, and to generalize the Markovian analysis required to obtain the number of jobs in the external queues, in a straightforward way.

We must now compute the load-dependent service rates if there is 1 token, for R synchronization stations. Unfortunately, these service rates depend on each other and therefore an iterative procedure is required to find them. The procedure for this case is given by the following steps:

1. Replace the synchronziation stations by load-dependent exponential servers:

   (a) Set the load-dependent service rates for stations $M + 1$ to $M + R$ to some initial value. For example, set $v_{M+r}(1) = \lambda^{(r)}$; this corresponds to rejecting jobs that upon their arrival do not find a token of their class in the external queue.

   (b) For $r = 1, \ldots, R$, solve the complement CQN of station $M + r$, C_{M+r}, by an approximate MVA algorithm. This yields the class r throughput of the complement CQN of station $M + r$, with population $\vec{N}$, $TH^{C_{M+r}}(\vec{N})$.

   (c) Reset the load-dependent service rates according to

   $$v_{M+r}(n) = \frac{1}{q^{(r)}} \lambda^{(r)} ,$$

   where $q^{(r)}$ is given by

   $$q^{(r)} = 1 - \frac{\lambda^{(r)}}{TH^{C_{M+r}}(\vec{N})} .$$

   (d) Repeat Steps 1b and 1c until convergence of the service rates.

2. Solve the resulting CQN with an approximate MVA algorithm.

3. Compute the number of jobs in each synchronzation station by performing a Markovian analysis.

Observe that the procedure for this case is again very similar to the procedure for the simple Case 1. The extra difficulty arises mainly in Step 1. To replace the synchronization station by load-dependent exponential servers we need to perform an iterative (sub)-procedure. This of course greatly affects the efficiency of the procedure. We come back to the efficiency later.

The approximate MVA algorithm to be applied in Step 2 is formulated in Appendix A.4.

The analysis done in Step 3 is basically identical to that of Step 3 in the procedure for Case 1. This is due to the fact that each synchronziation station is only visited by one class.

The procedure described here is basically the same as the one described by Perros, Dallery and Pujolle [11]. The difference is that by applying an approximate MVA algorithm, we are able to allow non-exponential service times at the service stations in the inner system. Perros, Dallery and Pujolle avoid non-exponential servers, because in their algorithm the presence of these servers would cause the number of iterations to grow too much, as they would also have to be replaced by load-dependent exponential servers.

It is important to know whether the proposed iteration procedure converges. Since the procedure is equivalent to that of Perros, Dallery and Pujolle who claim that their algorithm has converged on all their test sets (with only exponential servers), we presume that this is the case for our procedure too. Further research is required to verify this.

3.5 Case 6

Consider a multiclass OQN-RC in which the classes can be partioned into subsets such that the classes in a subset share one set of tokens dedicated to that subset. We call this the hybrid case or Case 6. For this case a procedure that is a combination of the procedures for the cases 5 and 6 can be formulated. The extensions are straightforward and we do not describe them explicitely here.

4 Concluding remarks

We have provided a framework of procedures for 6 variations of the OQN-RC. The procedures are based on the CQN view of the OQN-RC and the application of (approximate) MVA algorithms to solve the resulting network. Like for the traditional aggregation method, the main approximation error for these procedures will be in the approximation of the external queue lengths. Extensive numerical experiments will have to verify whether the procedures yield accurate approximations.

References

[1] A. Brandwajn. Fast approximate solution of multiprogramming models. *Performance Evaluation Review*, pages 141–149, 1982.

[2] R. Buitenhek, G. V. Houtum, and W. Zijm. An evaluation of approximate mean value analysis algorithms. Working paper LPOM-95-18, Faculty of Mechanical Engineering, University of Twente, 1995.

[3] R. Buitenhek, G. V. Houtum, and W. Zijm. A queueing model of a dual resource production problem. Working paper LPOM-95-17, Faculty of Mechanical Engineering, University of Twente, 1995.

[4] R. Buitenhek, G. V. Houtum, and W. Zijm. An open queueing model for flexible manufacturing systems with multiple part types and general purpose pallets. Working paper LPOM-96-11, Faculty of Mechanical Engineering, University of Twente, 1996.

[5] J. Buzacott and J. Shanthikumar. *Stochastic Models of Manufacturing Systems.* Prentice Hall, 1993.

[6] Y. Dallery. Approximate analysis of general open queueing networks with restricted capacity. *Performance Evaluation*, 11:209–222, 1990.

[7] M. Di Mascolo. Analysis of a synchronization station for the performance evaluation of a kanban system with a general arrival process of demands. *European Journal of Operational Research*, 89:147–163, 1996.

[8] E. Lazowska and J. Zahorjan. Multiple class memory constrained queueing networks. *Performance Evaluation Review*, pages 130–140, 1982.

[9] R. Marie. An approximate method for general queueing networks. *IEEE Transactions on Software Engineering*, 5:530–538, 1979.

[10] M. Neuts. *Matrix-Geometric Solutions in Stochastic Models - An Algorithmic Approach.* The John Hopkins University Press, 1981.

[11] H. Perros, Y. Dallery, and G. Pujolle. Analysis of a queueing network model with class dependent window flow control. *IEEE InfoCom*, pages 968–977, 1992.

[12] M. Reiser and S. Lavenberg. Mean-value analysis of closed multichain queueing networks. *Journal of the Association for Computing Machinery*, 27:313–322, 1980.

[13] L. Seelen. An algorithm for PH|PH|c queues. *European Journal of Operational Research*, 23:118–127, 1986.

[14] U. Sumita and M. Rieders. Numerical comparison of the replacement process approach with the aggregation-disaggregation algorithm for row-continuous markov chains. In W. J. Stewart, editor, *Numerical Solution of Markov Chains*, pages 287–302. Marcel Dekker Inc., 1991.

[15] A. Thomasian and P. Bay. Analysis of queueing network models with population size constraints and delayed blocked customers. In *Proceedings of the ACM SIGMETRICS Conference, Cambridge*, pages 202–216, 1984.

A Appendix: MVA algorithms

In this appendix we give (approximate) MVA algorithms that can be used in the procedures described in this paper. We like to stress that the approximate MVA algorithms are by no means the only possible algorithms that one may use. They just serve as an illustration of what is possible.

As a basis for the algorithms given here, we take the algorithms as described in the text of Buzacott and Shanthikumar [5] and in [2]. For the non-product-form networks we will assume, however, that except for the stations replacing the synchronziation stations, there are no load-dependent servers. This assumption is not necessary, but allows us to keep a clearer presentation of the algorithms.

A.1 Single class exact MVA

The algorithm for single class product-form CQN is given on pages 373-374 in [5].

1. Set $p_m(0|0) = 1$, $m = 1, \ldots, M+1$.
2. For $n = 1, \ldots, N$, compute
   (a) for $m = 1, \ldots, M$,

$$ET_m(n) = \sum_{k=0}^{n-1} \frac{k+1}{\mu_m(k+1)}\, p_m(k|n-1) \tag{8}$$

   (b)

$$TH(n) = \frac{n}{\sum_{m=1}^{M+1} \nu_m\, ET_m(n)} \tag{9}$$

   (c) for $m = 1, \ldots, M$,

$$p_m(k|n) = \frac{\nu_m TH(n)}{\mu_m(k)}\, p_m(k-1|n-1) \tag{10}$$

$$p_0(k|n) = 1 - \sum_{k=1}^{n} p_m(k|n) \tag{11}$$

A.2 Single class approximate MVA

Approximate MVA algorithms for the general single class CQN are given on pages 378-380 of [5]. We give a slightly different algorithm. Let station $M+1$ be a load-dependent exponential server with rates

$$\begin{aligned} \mu(1) &= \frac{\lambda}{q} \text{ and } q = 1 - \frac{\lambda}{TH^{C_{M+1}}(N)} \\ \mu(n) &= \lambda, \text{ for } n > 1. \end{aligned}$$

Furthermore, let stations m be a general single or multiple server station with service time S, with known first two moments ES_m and ES_m^2.

We first rewrite Equation (8) for station $M+1$.

$$\begin{aligned} ET_{M+1}(n) &= \sum_{k=1}^{n-1} \frac{k+1}{\lambda}\ p_m(k|n-1) + p_m(0|n-1)\ q\ \frac{1}{\lambda} \\ &= [EL_{M+1}(n-1) + Q_{M+1}(n-1) + p_m(0|n-1)\ ]\ \frac{1}{\lambda}\ , \end{aligned}$$

where

$$EL_{M+1}(n) = \sum_{k=1}^{n} k\ p_m(k|n),$$

and

$$Q_{M+1}(n) = \sum_{k=1}^{n}\ p_m(k|n).$$

1. Initialize. Set $p_m(0|0) = 1$, $Q_m(0) = 0$, $E\tilde{L}_m(0) = 0$, $m = 1, \ldots, M$ and $p_{M+1}(0|0) = 1$, $Q_{M+1}(0) = 0$, $EL_{M+1}(0) = 0$.

2. Preprocessing, for $m = 1, \ldots, M$,

$$ES_{rem,m} = \frac{c_m - 1}{c_m + 1}\ \frac{ES_m}{c_m} + \frac{2}{c_m + 1}\ \frac{1}{c_m}\ \frac{ES_m^2}{2ES_m}\ .$$

3. For $n = 1, \ldots, N$, compute

   (a) for $m = 1, \ldots, M$,

$$ET_m(n) = Q_m(n-1)\ ES_{rem,m} + E\tilde{L}_m(n-1)\ \frac{ES_m}{c_m} + ES_m,$$

   and

$$ET_{M+1}(n) = [EL_{M+1}(n-1) + Q_{M+1}(n-1) + p_m(0|n-1)\ ]\ \frac{1}{\lambda}$$

   (b)

$$TH(n) = n\ /\ \sum_{m=1}^{M+1} \nu_m ET_m(n),$$

   (c) for $m = 1, \ldots, M$,

$$Q_m(n) = \frac{ES_m}{c_m}\ \nu_m TH(n)\ \ [Q_m(n-1) + p_m(c_m - 1|n-1)]$$

   and

$$Q_{M+1}(n) = \frac{TH(n)}{\lambda}\ \ [Q_{M+1}(n-1) + p_{M+1}(0|n-1)\ q]$$

   (d) for $m = 1, \ldots, M$,

$$E\tilde{L}_m(n) = \frac{ES_m}{c_m}\ \nu_m TH(n)\ \ \left[E\tilde{L}_m(n-1) + Q_m(n-1)\right]$$

   (e)

$$EL_{M+1}(n) = TH_{M+1}(n)\ ET_{M+1}(n).$$

A.3 Approximate single chain multiple class MVA

1. Initialize. Set $p_m(0|0) = 1$, $Q_m(0) = 0$, $E\tilde{L}_m(0) = 0$, $m = 1, \ldots, M$ and $p_{M+1}(0|0) = 1$, $Q_{M+1}(0) = 0$, $EL_{M+1}(0) = 0$.

2. For $n = 1, \ldots, N$, compute

   (a) for $m = 1, \ldots, M$,

$$ET_m^{(r)}(n) = Q_m(n-1)\ ES_{rem,m}(n) + E\tilde{L}_m(n-1)\ \frac{ES_m(n)}{c_m} + ES_m^{(r)},$$

   where

$$ES_{rem,m}(n) = \frac{c_m - 1}{c_m + 1}\ \frac{ES_m(n)}{c_m} + \frac{2}{c_m + 1}\ \frac{1}{c_m}\ \frac{ES_m^2(n)}{2ES_m(n)}\ ,$$

   and

$$ET_{M+1}(n) = [EL_{M+1}(n-1) + Q_{M+1}(n-1) + p_m(0|n-1)\ ]\ \frac{1}{\lambda}$$

   (b)

$$TH^{(r)}(n) = n^{(r)}\ /\ \sum_{r=1}^{R}\sum_{m=1}^{M+1} \nu_m ET_m(n),$$

   (c) for $m = 1, \ldots, M$, and for $k = 1, 2$,

$$ES_m^k(n) = \sum_{r=1}^{R} \frac{TH^{(r)}(n)}{\sum_{r=1}^{R} TH^{(r)}(n)}\ E\left(S_m^{(r)}\right)^k,$$

   (d) for $m = 1, \ldots, M$, and for $l = 1, \ldots, \min(c_m - 1, n)$,

$$p_m(l|n) = \frac{ES_m(n)}{c_m}\nu_m\ \sum_{r=1}^{R} TH^{(r)}(n)\ p_m(l-1|n-1)\ ,$$

   (e) for $m = 1, \ldots, M$,

$$Q_m(n) = \frac{ES_m(n)}{c_m}\ \nu_m\ \sum_{r=1}^{R} TH^{(r)}(n)\ \ [Q_m(n-1) + p_m(c_m - 1|n-1)]$$

   and

$$Q_{M+1}(n) = \frac{TH(n)}{\lambda}\ \ [Q_{M+1}(n-1) + p_{M+1}(0|n-1)\ q]$$

   (f) for $m = 1, \ldots, M$

$$p_m(0|n) = 1 - \sum_{l=1}^{\min(c_m-1,n)} p_m(l|n) - Q_m(n)$$

   and

$$p_{M+1}(0|n) = 1 - Q_{M+1}(n)\ ,$$

(g) for $m = 1, \ldots, M$,

$$E\tilde{L}_m(n) = \frac{ES_m(n)}{c_m}\ \nu_m TH(n)\ \left[E\tilde{L}_m(n-1) + Q_m(n-1)\right]$$

(h)

$$EL_{M+1}(n) = TH_{M+1}(n)\ ET_{M+1}(n).$$

A.4 Multiple class approximate MVA

1. Initialize. Set $p_m(0|\vec{0}) = 1$, $Q_m(\vec{0}) = 0$, $E\tilde{L}_m(\vec{0}) = 0$, $m = 1, \ldots, M$ and for $r = 1, \ldots, R$, $p_{M+r}(0|\vec{0}) = 1$, $Q_{M+r}(\vec{0}) = 0$, $EL_{M+r}(\vec{0}) = 0$.

2. For $n = 1, \ldots, N$, for all states $\vec{n}$ for which $\sum_{r=1}^{R} n^{(r)} = n$, and $n^{(r)} \leq N^{(r)}$ compute,

   (a) for $m = 1, \ldots, M$,

   $$ET_m^{(r)}(\vec{n}) = Q_m(\vec{n} - \vec{e}_r)\ ES_{rem,m}^{(r)}(\vec{n}) + E\tilde{L}_m(\vec{n} - \vec{e}_r)\ \frac{ES_m(\vec{n} - \vec{e}_r)}{c_m} + ES_m^{(r)},$$

   where

   $$ES_{rem,m}^{(r)}(\vec{n}) = \frac{c_m - 1}{c_m + 1}\ \frac{ES_m(\vec{n} - \vec{e}_r)}{c_m} + \frac{2}{c_m + 1}\ \frac{1}{c_m}\ \frac{ES_m^2(\vec{n} - \vec{e}_r)}{2ES_m(\vec{n} - \vec{e}_r)}\ .$$

   and for $r = 1, \ldots, R$,

   $$ET_{M+r}(\vec{n}) = [EL_{M+r}(\vec{n} - \vec{e}_r) + Q_{M+r}(\vec{n} - \vec{e}_r) + p_{M+r}(0|\vec{n} - \vec{e}_r)\ ]\ \frac{1}{\lambda}$$

   (b)

   $$TH^{(r)}(\vec{n}) = n^{(r)}\ /\ \sum_{r=1}^{R}\sum_{m=1}^{M+1} \nu_m ET_m(\vec{n}),$$

   (c) for $m = 1, \ldots, M$, and for $k = 1, 2$,

   $$ES_m^k(\vec{n}) = \sum_{r=1}^{R} \frac{TH^{(r)}(\vec{n})}{\sum_{r=1}^{R} TH^{(r)}(\vec{n})}\ E\left(S_m^{(r)}\right)^k,$$

   (d) for $m = 1, \ldots, M$, and for $l = 1, \ldots, \min(c_m - 1, n)$,

   $$p_m(l|\vec{n}) = \frac{ES_m(\vec{n})}{c_m}\nu_m\ \sum_{r=1}^{R} TH^{(r)}(\vec{n})\ p_m(l-1|\vec{n} - \vec{e}_r)\ ,$$

   (e) for $m = 1, \ldots, M$,

   $$Q_m(\vec{n}) = \frac{ES_m(\vec{n})}{c_m}\ \nu_m\ \sum_{r=1}^{R} TH^{(r)}(\vec{n})\ [Q_m(\vec{n} - \vec{e}_r) + p_m(c_m - 1|\vec{n} - \vec{e}_r)]$$

   and for $r = 1, \ldots, R$,

   $$Q_{M+r}(\vec{n}) = \frac{TH(n)}{\lambda}\ [Q_{M+r}(\vec{n} - \vec{e}_r) + p_{M+r}(0|\vec{n} - \vec{e}_r)\ q]$$

(f) for $m = 1, \ldots, M$

$$p_m(0|\vec{n}) = 1 - \sum_{l=1}^{c_m - 1} p_m(l|\vec{n}) - Q_m(\vec{n})$$

and for $r = 1, \ldots, R$,

$$p_{M+r}(0|\vec{n}) = 1 - Q_{M+r}(\vec{n}) \ ,$$

(g) for $m = 1, \ldots, M$,

$$E\tilde{L}_m(\vec{n}) = \frac{ES_m(\vec{n})}{c_m} \ \nu_m TH(\vec{n}) \ \left[E\tilde{L}_m(\vec{n} - \vec{e}_r) + Q_m(\vec{n} - \vec{e}_r)\right]$$

(h) For $r = 1, \ldots, R$,

$$EL_{M+r}(\vec{n}) = TH_{M+r}(\vec{n}) \ ET_{M+r}(\vec{n}).$$