OR Spectrum (2013) 35:563–583
DOI 10.1007/s00291-012-0282-7

REGULAR ARTICLE

Analysis of finite-buffer state-dependent bulk queues

Remco Germs · Nicky van Foreest

Published online: 1 February 2012

Abstract In this paper, we consider a general state-dependent finite-buffer bulk queue in which the rates and batch sizes of arrivals and services are allowed to depend on the number of customers in queue and service batch sizes. Such queueing systems have rich applications in manufacturing, service operations, computer and telecommunication systems. Interesting examples include batch oven processes in the aircraft and semiconductor industry; serving of passengers by elevators, shuttle buses, and ferries; and congestion control mechanisms to regulate transmission rates in packet-switched communication networks. We develop a unifying method to study the performance of this general class of finite-buffer state-dependent bulk queueing systems. For this purpose, we use semi-regenerative analysis to develop a numerically stable method for calculating the limiting probability distribution of the queue length process. Based on the limiting probabilities, we present various performance measures for evaluating admission control and batch service policies, such as the loss probability for an arriving group of customers and for individual customers within a group. We demonstrate our method by means of numerical examples.

Keywords Bulk-arrival and bulk-service queue · Finite buffer · State-dependent control · Loss probabilities

1 Introduction

Group arrival and batch service queues (usually called *bulk queues*) have many applications in manufacturing, service operations, computer and telecommunication systems.

R. Germs (✉) · N. van Foreest
Faculty of Economics and Business, P.O. Box 800, 9700 AV Groningen, The Netherlands
e-mail: r.germs@rug.nl

N. van Foreest
e-mail: n.d.van.foreest@rug.nl

Since most of these systems have finite buffer capacity, it is of interest to study queueing systems with finite queue size. For example, in manufacturing systems, there is limited waiting room before workstations in assembly lines, material handling systems, or cellular manufacturing cells. In service systems such as facilities, there are limited circulation systems (elevators, stairways, and corridors) and finite storage areas (MacGregor Smith and Cruz 2005). Finally, in computer and telecommunication systems, routers and switches that regulate the transmission of information packages have finite buffer capacity.

In many of these applications, the arrival and service rate depend on the state of the queue. For example, a long queue can "discourage" arriving customers (Dshalalow 1997) leading to queue-length dependent balking. Another example consists of systems where the server is a human being and the perception of the workload may directly influence the server's productivity (Bekker 2004; Bekker et al. 2004). Besides the arrival and service rate, the size of arriving group and service batches may be queue length dependent. For instance, when the queue length hits the maximum buffer capacity, a situation can occur that a newly arriving group of customers does not find enough room in the queue and that a part of the group has to be refused from entering the system. Furthermore, service batch sizes are typically determined by the capacity of the server (i.e. the maximum number of customers that can be served simultaneously) and the number of customers waiting in queue, e.g., in the serving of people by elevators, shuttle buses, and ferries. Finally, the batch service time can also depend on the batch size; typically larger batches require more service time. In all these applications, it is helpful to be able to compute relevant performance measures, such as average time in system, moments of the number of customers in queue and loss probabilities for arriving groups of customers, or individual customers within a group. This allows operators to determine optimal system configuration, good admission control policies, or optimal batch sizing policies.

In this paper, we develop a simple, numerically stable, and efficient algorithmic method that allows the performance evaluation of a general queueing system that contains all of the above examples as special cases. The queueing system that we study for our purpose is the finite-buffer state-dependent bulk queue: $M(n)^{X(n)}/G(n)^{Y(n)}/1/K+B$. Here $M(n)$ and $G(n)$ correspond to the state-dependent arrival and batch service processes, the exponents $X(n)$ and $Y(n)$ represent the (random) state-dependent sizes of the arriving groups and service batches, the capacity of the queue is limited by K, and, finally, the maximal service capacity is B. The formal analysis of this queueing system is considered an open problem in the queueing literature (Dshalalow 1997) and thereby our research makes a start to fill a gap in literature. To do so, we use a semi-regenerative analysis to obtain the limiting probabilities of the queueing process, which in turn allows the computation of many performance measures relevant for selecting the best system configuration.

The paper is organized as follows. In Sect. 2, we provide applications of the finite-buffer state-dependent bulk queue and review literature related to the analysis of the model. After introducing the $M(n)^{X(n)}/G(n)^{Y(n)}/1/K+B$ model in Sect. 3 we illustrate in Sect. 4 how various special cases and applications are covered by the model. In Sect. 5, we present the semi-regenerative analysis of the model and obtain the limiting probabilities in terms of recurrence relations. Section 6 presents the algorithmic

aspects of our solution method and Sect. 7 defines various performance measures of the model. In Sect. 8, we use numerical examples to demonstrate our method.

2 Applications and literature review

Before reviewing related literature, we sketch two practical scenarios leading to bulk queue models with finite buffers and state-dependent arrival or service processes.

A typical batch-wise process in the aircraft industry concerns the hardening of synthetic parts (Hodes et al. 1992; Van der Zee et al. 2001). These parts arrive in groups from preceding manufacturing steps and are hardened in an oven in a batch-wise manner. Upon arrival the parts enter a buffer where they wait until they are loaded into the oven. The maximum time parts can stay in the buffer is limited due to strict quality constraints. In particular, if parts stay more than T time units in the buffer, the products become worthless for any further use. The time limit is operationalized by constraining the capacity of the buffer to K parts. Furthermore, service batch sizes are limited by the physical size of the oven, and processing times (including preparation times) are independent of the number of parts in a batch. Once processing has started, no interruption is allowed, i.e. no addition or extraction of parts is possible during the production process. Given these characteristics, a control policy is required that determines, once a service batch is finished, when to start a new batch service in such a way that logistical costs and product loss are minimized and a given service level is reached. This process can be modeled as a finite-buffer state-dependent bulk queue. The arrival group sizes correspond to the synthetic parts which are state-dependent due to the finite capacity of the buffer. A service batch corresponds to the parts that are hardened in the oven in a batch-wise manner and the service batch size is also dependent on the number of parts waiting in the buffer. Other production systems that possess more or less similar characteristics are ovens that are used for the diffusion/oxidation process in the manufacture of semiconductor wafers (Fowler et al. 1992; Uzsoy et al. 1994) and the burn-in operation of a manufacture of medical diagnostic units (Hopp and Spearman 2008).

Bulk queueing systems are also often found in transportation since mass transit vehicles are natural batch servers to which passengers arrive in groups of varying size. Furthermore, arriving passengers may decide to take another mode of transport when the queue length becomes excessive which makes state-dependent arrival rates a realistic assumption. The single server system is generally found in the form of a shuttle between two or more campuses of an institution, see e.g. (Deb 1978; Weiss 1979). The travel time does not depend on the number of passengers aboard and the fixed travel cost is only minimally affected by the number of passengers carried. Given these characteristics, an operating policy is required that determines when to dispatch the shuttle such that service cost and passenger waiting time are minimized. The state-dependent finite-buffer bulk queue is a reasonable model to evaluate dispatching rules for the shuttle bus problem.

Besides for practical examples, the formal analysis of the $M(n)^{X(n)}/G(n)^{Y(n)}/1/K+B$ model is also theoretically a challenging problem. There is a long tradition in the development of algorithmic methods for computing the limiting probabilities of

generalizations of the $M/G/1/K$ queueing process, e.g. cf. Neuts (1977) and Takagi (1993). The $M/G/1/K$ queue with state-dependent arrival and service rates was first analyzed by Courtois and Georges (1971) using the embedded Markov chain approach. However, as Gupta and Rao (1998) pointed out, the method presented by Courtois and Georges (1971) is numerically unstable. A stable recursive algorithm for computing the limiting probabilities of the $M/G/1/K$ queue with state-dependent arrival rates has been given by Tijms and Van Hoorn (1981). Schellhaas (1983) and Gupta and Rao (1998) generalized the model of Tijms and Van Hoorn (1981) by allowing state-dependent service times, using respectively a semi-regenerative approach and the supplementary variable method.

Comparatively less work has been done to introduce state dependencies into finite-buffer $M/G/1$ bulk queues. In the survey on queueing systems with state-dependent parameters, Dshalalow (1997) mentions that it is still an open problem to generalize the state-dependent $M/G/1/K$ model for group arrivals and batch services. In recent years, however, significant contributions have been made to the development of algorithmic methods for computing the limiting probability of $M^X/G^Y/1/K+B$ bulk queues under various rejection policies, cf. Nobel (1989), Dudin et al. (2005), Chang et al. (2004) and Germs and Van Foreest (2010). Also the literature on queueing models with different types of batch service policies has grown over the years. In Medhi (2003) and Chaudry and Templeton (1983) a comprehensive treatment of bulk queues with batch service can be found. However, in all of the aforementioned research on bulk queues, none of the input or service parameters of the queueing models are state-dependent.

3 Model

We consider a single server queue at which groups of customers arrive according to a state-dependent Poisson process with finite rate λ_i when the queue contains i customers. We note that λ_i denotes the rate at which groups of customers *arrive*; note that, due to the finite capacity K of the queue, the arrival rate can be different from the rate at which groups of customers are *accepted*. The sizes of the *arriving* groups form a sequence of independent integer random variables, distributed as the generic random variable X_i with probability mass function $\mathrm{P}\{X_i = k\} = x_i(k)$, $k \geq 1$. Here and in the sequel, the subscript i will always refer to the dependence on the queue length (number of customers waiting for service) at the moment of customer arrival or service completion (the context will always clarify which of the two cases apply).

Due to the limited capacity of the queue, it can occur that a newly arriving group does not find enough room in the queue. As a consequence, a decision has to be made which part of the group is to be refused from entering the system. Hence, dependent on the *rejection policy* in use and the queue length, the distribution of the size of an *accepted* group may differ from the distribution of the size of an *arriving* group. Let the sizes of the accepted groups be distributed as the generic random variable $\hat{X}_i$ with $\mathrm{P}\{\hat{X}_i = k\} = \hat{x}_i(k)$, $k \geq 0$. We refer to Sect. 4 for examples that illustrate how to define the $\hat{x}_i(k)$ for various rejection policies.

Customers are served in FCFS order in service batches. Service batch sizes are independent integer random variables, distributed as the generic random variable Y_i with distribution $\mathrm{P}\{Y_i = k\} = y_i(k)$, for $k = 0, \ldots, B$, where B is the maximal server capacity. Here, $y_i(0)$ denotes the probability that no customers are taken into service and that, as a consequence, the server enters an idle period. A situation in which it is reasonable to keep the server idle while there are customer waiting in queue is when the aim is to minimize average waiting time of customers in the system. In fact, Aalto (2000) and Deb and Serfozo (1973) prove that it is optimal to start serving customers only when the number of customers in queue exceeds some threshold a. Note that $y_i(k) = 0$ if $k > i$, since it is impossible to take k customers into service when there are only $i < k$ customers in queue. We assume that any arrival during a service joins the queue, if accepted. (Thus, if a group arrives to find $k > 0$ customers in service, the group cannot join the batch already undergoing service.) Batch service times $S_{i,k}$ are assumed to be independent of the arrival process, but may depend on the service batch size k and on the queue length i, and form a set of independent random variables distributed as $G_{i,k}(s) = \mathrm{P}\{S_{i,k} \leq s\}$. We assume $\mathrm{E}(S_{i,k}) < \infty$ for all i, k.

4 Special cases

In this section, we illustrate that $M(n)^{X(n)}/G(n)^{Y(n)}/1/K+B$ model covers a large class of well-known finite-buffer single server queueing models. The models are loosely ranked in order of complexity. As later models are in most cases extensions of previous models, we only specify the parameter settings in which these models differ from the previous models.

We extend the finite-buffer single server model mainly in two directions: different service batching policies, and rejection (blocking) policies. We choose to implement the service (rejection) policies by means of specific choices for $y_i(k)$ ($\hat{x}_i(k)$).

4.1 $M/G/1/K+1$ queue

This queue is the base model for the other models and can be derived by taking $\lambda_i = \lambda$, $x_i(1) = 1$ and $G_{i,k}(\cdot) = G(\cdot)$, for $i, k \geq 0$, in the $M(n)^{X(n)}/G(n)^{Y(n)}/1/K+B$ model. Since customers are blocked when K customers are in queue, it follows that $\hat{x}_i(1) = 1$ if $i < K$ and $\hat{x}_i(0) = 1$ if $i \geq K$. Observe that the server does not idle if the queue is not empty and serves the customers one by one. Therefore, $y_0(0) = 1$ and $y_i(1) = 1$ for all $i > 0$.

4.2 $M/G^Y/1/K+B$ queue with random batch service

In this model, the server has a random capacity Y. The actual number of customers accepted in a given service period equals the whole queue, or the current batch capacity, whichever is less (see Bagchi and Templeton (1973)). To implement the policy, we set

$$y_i(k) = \begin{cases} \mathrm{P}\{Y = k\}, & \text{if } k < i, \\ \sum_{k=i}^{\infty} \mathrm{P}\{Y = k\}, & \text{if } k = i, \\ 0, & \text{otherwise.} \end{cases}$$

A practical example of the random batch-service policy can be found in the semiconductor industry, where it is frequently observed that circuit boards are processed in random batches (Hochbaum and Landy 1997).

4.3 $M/G^{[a,b]}/1/K+b$ queue with minimal batch service

With the *minimal batch service* policy, the server only serves batches of size at least a and not larger than b, that is, $\mathrm{P}\{Y_i = \min\{i, b\}\} = 1$ only when $i \geq a$. To implement the policy, we set $y_i(0) = 1$ if $i < a$ and $y_i(k) = 1\{k = \min\{i, b\}\}$ if $i \geq a$.

Deb and Serfozo (1973) show that the minimal batch service policy is optimal for a batch service queue where costs are incurred for serving the customers and for holding them in the system. Aalto (2000) generalizes the result to queueing systems with compound Poisson arrivals. Note that if the cost of serving is set to zero, minimizing the expected averaged cost is equivalent to minimizing the average waiting time. Applications of this batch service policy are abundant and can be found in the serving of people by elevators, ferries, and shuttle buses; the transhipment of mail, and military supplies; the processing of computer programs, job applications and library books; and the production, inventory control and shipment of commercial products (Deb and Serfozo 1973).

4.4 $M/G^{[b,b]}/1/K+b$ queue with full batch service

The *full batch service policy* is contained in the previous model by setting $a = b$.

4.5 $M^X/G/1/K+1$ queue with partial acceptance

Since the queue length is bounded, and group sizes may be larger than 1, we need to decide how to handle arriving groups whose size exceeds the free capacity. In case of *partial acceptance*, whenever the size of the arriving group and the queue length i at an arrival epoch exceed K, only the part of the batch that fits into the buffer is accepted (i.e. $K - i$ customers). Hence, for $i < K$

$$\hat{x}_i(k) = \begin{cases} x_i(k), & \text{if } i + k < K, \\ \sum_{l \geq k} x_i(l), & \text{if } i + k = K, \end{cases}$$

and $\hat{x}_i(0) = 1$ for $i = K$.

This policy has many applications in manufacturing, service, computer and telecommunication systems, as the partial batch acceptance policy utilizes the buffer space in an optimal manner so that the loss probability of customers is rather low.

4.6 $M^X/G/1/K+1$ queue with complete rejection

In a make-to-order situation where a group of customers represents a batch of products belonging to one order, it is often not possible to allow partial acceptance of individual products. The same holds for telecommunication systems where a group of customers is interpreted as a set of packages belonging to one information unit (Dudin et al. 2005). For these situations, it is more realistic to select the *complete rejection* or the *complete acceptance* admission policy.

Under the complete rejection policy the complete arriving group is rejected if its size exceeds the available buffer space. It is not difficult to see that the distribution of $\hat{X}_i$ for the complete rejection model is given by $\hat{x}_i(k) = x_i(k)1\{i+k \leq K\}$, for $k \geq 1$. Observe that under the complete rejection policy, $\hat{x}_i(0)$ is the probability that at an arrival epoch all customers in the group are rejected. Hence, $\hat{x}_i(0) = \sum_{k>K-i} x_i(k)$.

4.7 $M^X/G/1/K+1$ queue with complete acceptance

In situations where customers arrive in large groups, the complete rejection policy has a rather high loss probability. The complete acceptance policy may provide in these cases a much better performance. Under this policy, a group is completely accepted whenever part of it can be accepted and therefore $\hat{x}_i(k) = x_i(k)$ if $i < K$ and $\hat{x}_i(0) = 1$ if $i \geq K$.

The complete acceptance discipline suggest a presence of some additional place for admitting a whole group which can not be completely placed into the buffer. This is however not a problem in many real life systems. For instance, if we model a computer system we can consider RAM (Random Access Memory) as a finite buffer. In case of buffer overflow, the information that does not fit into the RAM can be placed into extended or expanded memory (Dudin et al. 2005).

4.8 $M(n)/G(n)/1/K$ queue

In this model, the arrival and service process are dependent on the number of customers in the system (i.e. the number of customers in the queue plus the one in service in case the server is busy). The model, and some special cases of it (e.g. the machine repairman problem), has been discussed extensively by Schellhaas (1983) and Gupta and Rao (1998). As we will discuss in Remark 2, we can let the arrival rate depend on the status of the server by replacing the λ_i in the model by $\lambda_{i,l}$, where $l = 1$ if the server is busy and $l = 0$ otherwise. Now, let the index m denote the number of customers in the system, then we can cover the $M(n)/G(n)/1/K$ queue by defining $\lambda_{0,0} = \lambda_0$, if $m = 0$, and $\lambda_{m-1,1} = \lambda_m$, if $m > 0$.

5 Semi-regenerative analysis of the model

We start with characterizing the state of the $M(n)^{X(n)}/G(n)^{Y(n)}/1/K+B$ queue and defining the limiting probabilities of the queue length process. Next, we derive a procedure to compute these limiting probabilities.

5.1 Preliminaries

To characterize the state of the $M(n)^{X(n)}/G(n)^{Y(n)}/1/K+B$ system at an arbitrary point in time t, we need to specify both the queue length and the server state at t. To see this, note that in the present model, the service policy may idle the server even when customers are present in queue. Therefore, knowing the number of customers in the queue at time t is not sufficient to determine whether the server is idle or busy at t. Let the queue length process $\{Q(t), t \geq 0\}$ take values in the finite set $E \subset \mathbb{N}$, while the busy process $\{B(t), t \geq 0\}$ takes values in $\{0, 1\}$, so that $B(t) = 1$ when the server is busy at time t and $B(t) = 0$ otherwise. The system is now characterized by the right continuous, bi-variate process $\{Q(t), B(t)\}$, which is assumed to have left limits in t.

The server observes the queue length at service completion epochs and at arrival epochs of customers when the server is idle. Let $0 = T_0 < T_1 < T_2 < \cdots$ be the ordered sequence of these epochs, and let $\{Q_n, n \geq 0\}$ denote the (embedded) queue length process as observed by the server at these times, that is, we define

$$Q_n = \begin{cases} Q(T_n-), & \text{if } T_n \text{ is a service completion epoch,} \\ Q(T_n-) + \hat{X}_{Q(T_n-)}, & \text{if } T_n \text{ is an arrival epoch and the server is idle.} \end{cases}$$

Thus Q_n is *either* the queue length just before service completion *or* the queue length just after the acceptance of (part of) the group of customers. Then, it is clear (although it requires some technical arguments, see e.g. Çinlar (1975) or Asmussen (2003)) that $\{Q_n, T_n\}$ is a Markov renewal process embedded in $\{Q(t), B(t)\}$, so that $\{Q(t), B(t)\}$ is a semi-regenerative process. This means that for any n, the conditional distribution of $\{Q(t+T_0+\cdots+T_n), B(t+T_0+\cdots+T_n)\}_{t\geq 0}$ given $T_0, \ldots, T_n, Q_0, \ldots, Q_n = i$ is the same as the conditional distribution of $\{Q(t), B(t)\}$ given $T_0 = 0$ and $Q_0 = i$. Hence, to characterize the conditional distribution of $\{Q(t), B(t)\}$ it suffices to specify the behavior $Q(t)$ and $B(t)$ on the interval $[T_0, T_1)$. Let $T_0 = 0$ and $Q_0 = i$, then $\{Q(t), B(t)\}$ must satisfy for $t \in [0, T_1)$,

$$B(t) = 1\{Y_i > 0\}, \tag{1a}$$

where $1\{A\}$ is the indicator function of the set $\{A\}$, and

$$Q(t) = \begin{cases} i, & \text{if } Y_i = 0, \\ i - Y_i + Z_{i-Y_i}(t), & \text{if } Y_i > 0, \end{cases} \tag{1b}$$

since if $Y_i = 0$ the server remains idle during $[0, T_1)$ and if $Y_i > 0$ it takes a batch of size Y_i into service while the random variable $Z_i(s)$ represents the number of *accepted* arrivals during $[0, s]$ given that at the start of the interval the queue length is i and the server is busy. Note that $Q(t) \geq 0$ for all $t \geq 0$, since $y_i(k) = 0$ if $k > i$, and $\{Z_i(s)\}$ is a pure birth process.

We assume that the embedded Markov chain $\{Q_n\}$ with state space E is irreducible and aperiodic and that the Markov renewal process $\{Q_n, T_n\}$ is aperiodic. Since E is also finite, it follows that $\{Q_n\}$ is positive recurrent.

Assuming that all these conditions are satisfied, the limiting distributions π of the embedded Markov chain $\{Q_n\}$ and p of the semi-regenerative process $\{Q(t), B(t)\}$ exist. That is, for $j \in E$,

$$\begin{aligned}
\pi_j &= \lim_{n\to\infty} \mathrm{P}\{\text{at time } T_n,\ j \text{ customers wait in queue}\} \\
&= \lim_{n\to\infty} \mathrm{P}\{Q_n = j\} \\
p_{j,0} &= \lim_{t\to\infty} \mathrm{P}\{\text{at time } t,\ j \text{ customers wait in queue and the server is idle}\} \\
&= \lim_{t\to\infty} \mathrm{P}\{Q(t) = j, B(t) = 0\} \\
p_{j,1} &= \lim_{t\to\infty} \mathrm{P}\{\text{at time } t,\ j \text{ customers wait in queue and the server is busy}\} \\
&= \lim_{t\to\infty} \mathrm{P}\{Q(t) = j, B(t) = 1\}. \qquad (2)
\end{aligned}$$

5.2 Analysis

We next derive a method to compute the limiting distributions of the embedded Markov chain $\{Q_n\}$ and the semi-regenerative process $\{Q(t), B(t)\}$. We start with deriving a numerically stable procedure to compute the semi-Markov kernel $H = \{H_i(j,t); i, j \in E, t \geq 0\}$ corresponding to the Markov renewal process $\{Q_n, T_n\}$. We recall from Çinlar (1975) or Asmussen (2003) that the elements $H_i(j,t)$ of H over E are defined as

$$\begin{aligned}
H_i(j,t) &= \mathrm{P}\{Q_{n+1} = j, T_{n+1} - T_n \leq t \mid Q_n = i\} \\
&= \mathrm{P}\{Q_1 = j, T_1 \leq t \mid Q_0 = i\}. \qquad (3)
\end{aligned}$$

To start the computation of H we expand the definition of $H_i(j,t)$ by conditioning on Y_i;

$$H_i(j,t) = \sum_{k=0}^{i} \mathrm{P}\{Y_i = k\}\, \mathrm{P}\{Q_1 = j, T_1 \leq t \mid Q_0 = i, Y_i = k\}.$$

Observe that when $Y_i = 0$, T_1 corresponds to an idle period that starts with i customers in queue and ends within t time units with the arrival of a group of customers from which $j - i$ are accepted. Hence,

$$\mathrm{P}\{Q_1 = j, T_1 \leq t \mid Q_0 = i, Y_i = 0\} = (1 - \mathrm{e}^{-\lambda_i t})\, \hat{x}_i(j - i).$$

Otherwise, when $Y_i = k > 0$, T_1 corresponds to a batch service of size k that starts with i customers in queue and ends within t time units during which $j - i + k$ customers are accepted. Writing $R_i(m,s) = \mathrm{P}\{Z_i(s) = m\}$ for the probability to accept m customers during a service interval of duration s that starts with i customers in queue, we have that for $k \geq 1$

$$\mathrm{P}\{Q_1 = j, T_1 \le t \mid Q_0 = i, Y_i = k\} = \int_0^t R_{i-k}(j - i + k, s)\, dG_{i,k}(s).$$

Now we can expand the expression for $H_i(j,t)$ as

$$H_i(j,t) = y_i(0)(1 - \mathrm{e}^{-\lambda_i t})\,\hat{x}_i(j-i) + \sum_{k=1}^{i} y_i(k) \int_0^t R_{i-k}(j - i + k, s)\, dG_{i,k}(s). \tag{4}$$

From (4), it is obvious that it remains to find a suitable expression to compute $R_i(m,s)$. In the following lemma we present an efficient recursion for this purpose.

Lemma 1 *The probability that in a service period of duration s, m customers are accepted, given that just after the start of the service i customers are in queue, can be written as*

$$R_i(m,s) = \sum_{n=0}^{\infty} U_i(m,n)\mathrm{e}^{-\lambda s}\frac{(\lambda s)^n}{n!}, \tag{5}$$

for some (finite) $\lambda \ge \max_{i\in E}\lambda_i$, and where $U_i(m,n)$ satisfies the following recursion for $i \in E$ and $n, m \ge 0$,

$$\begin{aligned} U_i(m,n+1) = U_i(m,n) &+ \frac{\lambda_{i+m}}{\lambda}\left[\hat{x}_{i+m}(0) - 1\right] U_i(m,n) \\ &+ \sum_{l=0}^{m-1} \frac{\lambda_{i+l}}{\lambda}\,\hat{x}_{i+l}(m-l) U_i(l,n), \end{aligned} \tag{6}$$

with initial conditions

$$U_i(m,0) = \begin{cases} 1, & \text{if } m = 0, \\ 0, & \text{if } m > 0. \end{cases}$$

Proof Since the group inter-arrival times are exponentially distributed it follows for sufficiently small $h > 0$ that

$$\begin{aligned} R_i(m, s+h) = [1 - \lambda_{i+m} h(1 - \hat{x}_{i+m}(0))] R_i(m,s) \\ + h \sum_{l=0}^{m-1} \lambda_{i+l}\,\hat{x}_{i+l}(m-l) R_i(l,s) + o(h). \end{aligned}$$

Subtracting $R_i(m,s)$ at both sides, dividing by h, and taking the limit $h \downarrow 0$ we arrive at the Kolmogorov forward equation

$$\frac{\mathrm{d}}{\mathrm{d}s}R_i(m,s) = \lambda_{i+m}(\hat{x}_{i+m}(0)-1)R_i(m,s) + \sum_{l=0}^{m-1}\lambda_{i+l}\,\hat{x}_{i+l}(m-l)R_i(l,s). \tag{7}$$

By the finiteness of the λ_i, there exist a finite λ such that $\lambda \geq \max_{i\in E}\lambda_i$. Therefore, we can use the uniformization method and substitute the form

$$R_i(m,s) = \sum_{n=0}^{\infty} U_i(m,n)\mathrm{e}^{-\lambda s}\frac{(\lambda s)^n}{n!} \tag{8}$$

in (7) for any such λ. After simplifying the result, we obtain (6).

The initial conditions follow from observing in (8) that $R_i(m,0) = U_i(m,0)$, and that $R_i(m,0) = 1\{m=0\}$.

Remark 1 Observe that $U_i(m,n)$ can be interpreted as the probability to accept m customers given that n groups of customers arrived since the start of the service epoch and given that the number of customers in queue just after the start of the service epoch was i.

Now we have all the tools to compute the semi-Markov kernel H and obtain the transition matrix $P = \{P(i,j); i,j \in E\}$ of the embedded Markov chain $\{Q_n\}$ by taking the limit of H as $t \to \infty$. From the assumption that $\{Q_n\}$ is an ergodic Markov chain, it follows that the limiting distribution π exists and is the unique solution (up to normalization) of

$$\begin{aligned}\pi_j &= \sum_{i\in E}\pi_i H_i(j,\infty)\\ &= \sum_{i\in E}\pi_i\left[y_i(0)\hat{x}_i(j-i) + V(i,j)\right],\end{aligned} \tag{9}$$

where

$$V(i,j) = \sum_{k=1}^{i} y_i(k)\sum_{n=0}^{\infty} U_{i-k}(j-i+k,n)a_{i,k}(n)$$

which we obtain after substituting (5) for $R_i(k,s)$ in (4) and reorganizing so that the integrations reduce to the mixed Poisson probabilities

$$a_{i,k}(n) = \int_0^{\infty}\mathrm{e}^{-\lambda s}\frac{(\lambda s)^n}{n!}dG_{i,k}(s). \tag{10}$$

To proceed from π to p, we use semi-regeneration in the following theorem. First, let C denote the length of the interval between two successive embedded Markov points T_n and T_{n+1}. Supposing that $Q_n = i$, observe that C is a service interval of length $S_{i,k}$ when $Y_i = k \geq 1$, and an inter-arrival time when $Y_i = 0$. Therefore, the expected cycle time C_i is

$$\mathrm{E}\,(S_i) = \sum_{k=1}^{\infty} y_i(k)\mathrm{E}\,(S_{i,k}),$$

if a service starts with a queue length $Q_n = i$ while it is $y_i(0)/\lambda_i$ when the server idles. Hence,

$$\mathrm{E}\,(C) = \mathrm{E}\,(\mathrm{E}\,(C|Q)) = \sum_{i\in E} \pi_i \left[\frac{y_i(0)}{\lambda_i} + \mathrm{E}\,(S_i)\right].$$

Note that it may occur that $\lambda_i = 0$ for some $i \in E$. We require in such states that $y_i(0) = 1$ to prevent that i is an absorbing state. In such cases set $y_i(0)/\lambda_i \equiv 0$.

Theorem 1 *The limiting distribution p satisfies*

$$p_{j,0} = \frac{y_j(0)}{\lambda_j}\frac{\pi_j}{\mathrm{E}\,(C)}, \tag{11a}$$

$$p_{j,1} = \sum_{i\in E} \frac{\pi_i}{\mathrm{E}\,(C)} V_e(i,j) \tag{11b}$$

where

$$V_e(i,j) = \sum_{k=1}^{i} y_i(k) \sum_{n=0}^{\infty} U_{i-k}(j-i+k,n)\, a^e_{i,k}(n), \tag{11c}$$

$$a^e_{i,k}(n) = \int_0^{\infty} \mathrm{e}^{-\lambda s} \frac{(\lambda s)^n}{n!}[1 - G_{i,k}(s)]\,\mathrm{d}s. \tag{11d}$$

Proof To prove (11) we use Çinlar (1975, Theorem 6.6.12) which states that for $j \in E$ and $l \in \{0, 1\}$

$$p_{j,l} = \frac{1}{\mathrm{E}\,(C)} \sum_{i\in E} \pi_i \int_0^{\infty} \psi_i(t,j,l)\,\mathrm{d}t, \tag{12}$$

provided that $\{Q_n, T_n\}$ is an ergodic process, $\mathrm{E}\,(C) < \infty$, and the function $t \to \psi_i(t,j,l) = \mathrm{P}\{Q(t) = j, B(t) = l, T_1 > t \mid Q_0 = i\}$ is directly Riemann integrable for every $i, j \in E$ and $l \in \{0, 1\}$.

To check these conditions, note that the first two conditions are true by the assumptions made in Sect. 3. From (1), (4), and the fact that T_1 equals a service time $S_{i,k}$ when $Y_i = k \geq 1$, and an inter-arrival time when $Y_i = 0$, it follows that

$$\psi_i(t,j,l) = \begin{cases} y_i(0)\,\mathrm{e}^{-\lambda_i t}, & \text{if } l = 0,\ j = i, \\ \sum_{k=1}^{i} y_i(k) R_{i-k}(j-i+k,t)[1 - G_{i,k}(t)], & \text{if } l = 1, \\ 0, & \text{otherwise.} \end{cases} \tag{13}$$

It is clear that $\psi_i(t, i, 0) = y_i(0)\,\mathrm{e}^{-\lambda_i t}$ is directly Riemann integrable for any $i \in E$. From (13) we have that $\psi_i(t, j, 1) \leq \max_{k \leq i}\{1 - G_{i,k}(t)\}$ for $i, j \in E$. Note that $1 - G_{i,k}(t)$ is directly Riemann integrable since it is non-increasing and $\int_0^\infty [1 - G_{i,k}(t)]dt = \mathrm{E}(S_{i,k}) < \infty$. Hence, $t \to \psi_i(t, j, l)$ is directly Riemann integrable for every $i, j \in E$ and $l \in \{0, 1\}$.

For $l = 0$, (11a) follows directly from (12) and (13). Let $l = 1$ and $j \in E$. Then by (12), (13), (5),

$$
\begin{aligned}
p_{j,1} &= \frac{1}{\mathrm{E}(C)} \sum_{i \in E} \pi_i \sum_{k=1}^{i} y_i(k) \int_0^\infty R_{i-k}(j - i + k, t)[1 - G_{i,k}(t)]\,\mathrm{d}t \\
&= \sum_{i \in E} \frac{\pi_i}{\mathrm{E}(C)} \sum_{k=1}^{i} y_i(k) \sum_{n=0}^{\infty} U_{i-k}(j - i + k, n) \int_0^\infty \mathrm{e}^{-\lambda t} \frac{(\lambda t)^n}{n!}[1 - G_{i,k}(t)]\,\mathrm{d}t.
\end{aligned}
$$

This proves (11b).

Remark 2 So far, the arrival process may only depend on the number of customers in queue and not on the status of the server (i.e. whether the server is idle or busy) at the moment of customer arrival. Dependence of the arrival process on the status of the server can easily be included in our model at the expense of an additional index l, where $l = 1$ if the server is busy and $l = 0$ otherwise. Now, we define $\lambda_{i,1}$ ($\lambda_{i,0}$) to be the rate at which customers arrive when there are i customers in queue and the server is busy (idle). In a similar way, we extend the definitions of the probabilities $x_{i,l}(k)$ and $\hat{x}_{i,l}(k)$, for $l = 0, 1$.

The equations needed for the computation of the limiting probabilities π and p, which we derived in this section, can now be adapted to cover the described extension. First observe that the λ_i and $\hat{x}_i(k)$ in Eq. (6) all correspond to the arrival rates and group sizes during a busy period and therefore can be replaced by $\lambda_{i,1}$ and $\hat{x}_{i,1}(k)$, respectively. Furthermore, the $y_i(0)\,\hat{x}_i(j - i)$ part in (9) and λ_i in (11a) (and in $\mathrm{E}(C)$) correspond to the group size and arrival rate of customers to an idle server and can be replaced by $y_i(0)\,\hat{x}_{i,0}(j - i)$ and $\lambda_{i,0}$, respectively. With these small modifications, we can generalize our model to include the dependence of the arrival process on the status of the server. In Sect. 4, we showed that this generalization enables us to study the $M(n)/G(n)/1/K$ where the number of customers in the *system* (instead of in the *queue*) is limited by K.

6 Algorithmic aspects

Now, we summarize the approach for computing the limiting probabilities at embedded, i.e., π_j, and arbitrary epochs, i.e., p_j, and we show how the precision of our numerical method can be specified in advance.

The numerical method that we have developed in the previous section leads to the following algorithm

Step 1 Compute (by numerical integration or if possible explicitly) the mixed Poisson probabilities $a_{i,k}(n)$ and $a^e_{i,k}(n)$ from relations (10) and (11d).
Step 2 Compute $U_i(k,n)$ for $i \in E$ and $k, n \geq 0$, by means of the recursion (6).
Step 3 Use standard numerical procedures to compute π from (9).
Step 4 Compute p using the relations in Theorem 1.

To compute π_j to a given precision $\epsilon > 0$ it suffices to compute $U_i(k,n)$ and the probabilities $a_{i,k}(n)$ up to some finite N_i, where

$$N_i = \min\left\{m; \sum_{n=0}^{m}\sum_{k=1}^{i} y_i(k)a_{i,k}(n) \geq 1-\epsilon\right\}.$$

This follows, since, c.f. (9),

$$V(i,j) = V_{N_i}(i,j) + e_{N_i},$$

where

$$V(i,j) = \sum_{k=1}^{i} y_i(k) \sum_{n=0}^{\infty} U_{i-k}(j-i+k,n)\, a_{i,k}(n),$$
$$V_{N_i}(i,j) = \sum_{k=1}^{i} y_i(k) \sum_{n=0}^{N_i} U_{i-k}(j-i+k,n)\, a_{i,k}(n),$$

and e_{N_i} satisfies

$$e_{N_i} = \sum_{k=1}^{i} y_i(k) \sum_{n\geq N_i+1} U_{i-k}(j-i+k,n)\, a_{i,k}(n) \leq \sum_{n\geq N_i+1}\sum_{k=1}^{i} y_i(k)a_{i,k}(n),$$

since the probabilities $U_{i-k}(j-i+k,n) \leq 1$. Therefore,

$$e_{N_i} \leq 1 - \sum_{n=0}^{N_i}\sum_{k=1}^{i} y_i(k)a_{i,k}(n) \leq \epsilon.$$

Similar reasoning applies to the computation of $p_{j,1}$.

Finally, note that the computations in our approach only involve additions and multiplications of positive and bounded numbers, thereby preventing a loss a significant digits. Observe also that explicit expressions for the $a_{i,k}(n)$ and $a^e_{i,k}(n)$ can be given for the cases of deterministic and phase-type services.

7 Performance measures

In this section, we derive a set of relevant performance measures such as the average number of customers in queue (L_q), the average waiting time in queue (W_q),

the server utilization (ρ) and the loss probability of a group of customers and of an arbitrary customer within a group. All these performance measures can be obtained from the limiting probabilities $p_{j,k}$, c.f. the definition in Eq. (2). Since $p_{j,k}$ is the probability that the queue contains j customers and the server is in state $k \in \{0, 1\}$, $p_j = p_{j,0} + p_{j,1}$ is the limiting probability that at an arbitrary point in time j customers are waiting in queue. Now it easily follows that

$$\rho = 1 - \sum_{j \in E} p_{j,0} = \sum_{j \in E} p_{j,1} \qquad \lambda' = \sum_{j \geq 0} \lambda_j \, p_j \, \mathrm{E}\,(\hat{X}_j) \tag{14a}$$

$$L_q = \sum_{j \in E} j \, p_j \qquad W_q = \frac{L_q}{\lambda'} \tag{14b}$$

where λ' is the *acceptance* rate of customers.

The loss probability of a group of customers and of an arbitrary customer within a group clearly depend on the rejection policy. Common rejection policies are the ones we discussed in Sect. 4, i.e., partial acceptance, complete rejection and complete acceptance. In what follows, we discuss the computation of the loss probabilities for these three rejection policies.

7.1 Complete acceptance

In the sequel, let Γ and γ, respectively, correspond to the event that a group is lost and that an arbitrary customer is lost. Then, it is not difficult to check that for the complete acceptance policy

$$\mathrm{P}\{\Gamma\} = \mathrm{P}\{\gamma\} = 1 - \sum_{j=0}^{K-1} p_j .$$

7.2 Complete rejection

Recall from Sect. 4 that under the complete rejection policy the probability that customers in an arriving group are rejected is $\hat{x}_i(0) = \sum_{k \geq K-i+1} x_i(k)$. Therefore,

$$\mathrm{P}\{\Gamma\} = \sum_{i=0}^{K} p_i \, \hat{x}_i(0). \tag{15}$$

To calculate the rejection probability for an arbitrary customer, we use the following renewal-theoretic result (Burke 1975):

$$q_k = \mathrm{P}\{\text{an arbitrary customer belongs to a group of size } k\} = \frac{k \, x(k)}{\mathrm{E}\,(X)}, \tag{16}$$

with $x(k) = \sum_{i \in E} p_i\, x_i(k)$ and $\mathrm{E}\,(X) = \sum_{k \geq 1} k\, x(k)$. Define $\bar{q}_k = \sum_{m=k}^{\infty} q_m$ to be the probability that an arbitrary customer belongs to a group of size greater or equal to k. Then

$$\begin{aligned} \mathrm{P}\{\gamma\} &= \sum_{i=0}^{K} \mathrm{P}\left\{ \gamma \,\middle|\, \begin{array}{l} \text{customer sees } i \text{ customers} \\ \text{in queue upon arrival} \end{array} \right\} p_i \\ &= \sum_{i=0}^{K} \mathrm{P}\left\{ \begin{array}{l} \text{customer belongs to a group} \\ \text{of size larger than } K - i \end{array} \right\} p_i \\ &= \sum_{i=0}^{K} p_i\, \bar{q}_{K-i+1}. \end{aligned} \tag{17}$$

7.3 Partial acceptance

Under the partial acceptance policy, it is preferable to interpret Γ as the event that a group of customers *overflows*, i.e. when an arriving group does not fit completely into the queue (see Nobel (1989)). Then, it is easy to check that

$$\mathrm{P}\{\Gamma\} = \sum_{i=0}^{K} p_i \sum_{k \geq K+1-i} x_i(k).$$

To calculate the rejection probability for an arbitrary customer, we define for $k \geq 1$

$$\begin{aligned} \eta_k &= \mathrm{P}\{\text{an arbitrary customer occupies the } k\text{th position in the group}\} \\ &= \sum_{m \geq k} \frac{x(m)}{\mathrm{E}\,(X)}, \end{aligned}$$

where the last equation follows by conditioning on the event that "an arbitrary customer belongs to a group of size k" and then using (16). Let $\bar{\eta}_k = \sum_{m=k}^{\infty} \eta_m$ denote the probability that an arbitrary customers occupies a position greater or equal to k in his group. Then, analogous to the derivation of (17) we obtain

$$\mathrm{P}\{\gamma\} = \sum_{i=0}^{K} p_i\, \bar{\eta}_{K-i+1}.$$

8 Numerical examples

In this section, we apply the model to the numerical analysis of three examples, a batch queueing process with queue length dependent balking, a queuing process subject to holding cost and loss, and a batch arrival/service process subject to queue

length dependent batch arrival sizes and batch size dependent service rates. Our code is available at the second author's homepage.[1]

8.1 Bulk queues with state dependent balking and service rates

Consider a single server shop. Customers require varying amounts of service. With little amount of work in the system, all customers are prepared to enter the system, but when there is a large amount of work, the 'large' customers still enter while most of the 'small' customers balk. As is commonly the case (see, e.g., Bekker 2004; Bekker et al. 2004), the server increases the service rate when the queue becomes longer. We assume complete acceptance, and K to be so large that the probability of overflow is negligible.

As a concrete example, suppose that the service requirement of a large customer is 10 times that of a small customer. We model this by letting the service requirement of a small (large) customer correspond to a batch size of $k = 1$ ($k = 10$) units. Large customers arrive at the system at rate $\lambda_l = 5$ per hour. We implement the balking behavior of the small customers by taking $\lambda_{s,i} = \max\{0, 10 - i\}$. Then, take $\lambda_i = \lambda_{s,i} + \lambda_l$, and set

$$x_i(k) = \begin{cases} \lambda_{s,i}/\lambda_i, & \text{for } k = 1, \\ \lambda_l/\lambda_i, & \text{for } k = 10, \\ 0, & \text{otherwise.} \end{cases} \tag{18}$$

Since we assume complete acceptance: $\hat{x}_i(k) = x_i(k)$ if $i < K$ and $\hat{x}_i(0) = 1$ if $i \geq K$. Let service take place in single units, thus, $y_i(1) = 1$ for all $i > 0$ and $y_0(0) = 1$. Note that the queue length corresponds now to the workload in the queue. For simplicity, we assume deterministic service times. When the queue length is long, however, the employee feels more stress, and therefore works at a higher rate. This is implemented by taking $S_{i,k} \equiv (90 + i/5)^{-1}$ for all i, k.

For the case with $K = 50$, we find that the acceptance rate, see Eq. 14a, $\lambda' = 56.1$ per hour, $\rho = 0.6119$, $L_q = 5.678$, and $\gamma = \Gamma = 0.0009$. As a simple reference, we compare this system to an $M/D/1$ queue with load $\rho = (5 \cdot 10 + 10)/90 = 2/3$, which leads to $L_q(M/D/1) = 2/3$. Clearly, this value is much lower than 5.678, leading us to conclude that simpler queueing models are not accurate models for general batch queueing processes.

8.2 Minimal batch service queues with holding and setup costs

Consider a batch service system subject to setup and service costs, holding costs and rejection costs. A natural batch service policy for this system is to start service only when the queue length exceeds some threshold a and then serve as many customers as possible. In this section, we compute the performance measures for

[1] http://nicky.vanforeest.com/batchqueues/batchModel.html.

the $M^X/G^{[a,b]}/1/K+b$ queueing process, for which the costs are a function of the threshold parameter a (see Sect. 4.3). We consider also three loss policies: complete rejection, complete acceptance and partial acceptance. As in Aalto (2000) we assume that the holding cost is c_h per customer in the queue per unit time, a service cost $c_k + c_s\, j$ is incurred at each service epoch when j is the batch service size, and a rejection cost of c_r per unit.

Let π and p denote the limiting distribution of the queue length process at embedded and arbitrary epochs, respectively. It is easy to check that the average rejection costs per time unit under the complete acceptance ($\mathrm{RC_{ca}}$), complete rejection ($\mathrm{RC_{cr}}$) and partial acceptance ($\mathrm{RC_{pa}}$) policy can be expressed as follows

$$\begin{aligned}
\mathrm{RC_{ca}} &= c_r \sum_{i\geq K} p_i \sum_{j=1}^{\infty} x_i(j)\, j = c_r \sum_{i\geq K} p_i\, \mathrm{E}\,(X_i)\\
\mathrm{RC_{cr}} &= c_r \sum_{i=0}^{K} p_i \sum_{j=K+1-i}^{\infty} x_i(j)\, j\\
\mathrm{RC_{pa}} &= c_r \sum_{i=0}^{K} p_i \sum_{j=K+1-i}^{\infty} x_i(j)\,(j-K+i).
\end{aligned}$$

Now the average cost per time unit under the minimal batch service policy and rejection policy $\xi \in \{\mathrm{ca}, \mathrm{cr}, \mathrm{pa}\}$ can be expressed as $\mathrm{AC}_\xi(a) = \mathrm{HC}(a) + \mathrm{SC}(a) + \mathrm{RC}_\xi$, where the average holding cost

$$\mathrm{HC}(a) = c_h \sum_{i\geq 0} p_i\, i,$$

and average service cost per time unit

$$\mathrm{SC}(a) = \sum_{i=a}^{B} \pi_i\,(c_k + c_s\, i) + \sum_{i=B+1}^{\infty} \pi_i\,(c_k + c_s\, B).$$

As a concrete example, suppose for all $i \in E$, $\lambda_i = 0.2$, $x_i(1) = 0.25$, $x_i(3) = 0.5$, $x_i(5) = 0.25$, $\mathrm{E}\,(X_i) = 3$, service is deterministic, i.e., $S_{i,k} \equiv 10$ for all k, i, $B = 10$, and $K = 10$. The parameters $\hat{x}_i(k)$ and $y_i(k)$ are defined as in Sect. 4 for the three rejection policies and the minimal batch service policy. Furthermore, the cost parameters are given by $c_h = 5$, $c_k = 10$, $c_s = 5$ and $c_r = 50$. In Table 1, we present the costs per unit time for different values of the threshold a. It is easy to find that the optimal minimal batch service threshold value $a^\star$ is 6 for the complete rejection policy, 7 for the partial acceptance and complete acceptance policies. Thus, the threshold value increases when the acceptance policy is less 'strict'. This is as expected, since the policy makes a trade-off between set-up costs, i.e., a cost c_k is incurred for each service interval, and the rejection costs. Setting the threshold a to a lower value increases the long run average setup costs but lowers the rejection costs.

Table 1 Long run average holding cost (HC), service cost (SC), rejection cost (RC) and total cost (AC) for the rejection policies Partial rejection (pr), Complete rejection (cr) and Complete admission (ca) as functions of the minimal batch service threshold a

a	1	2	3	4	5	6	7	8	9	10
HC_{ca}	14.90	14.47	14.37	14.42	14.78	15.44	16.94	18.49	21.15	24.52
SC_{ca}	35.37	33.75	33.02	30.61	28.99	27.60	26.05	25.06	24.22	23.52
RC_{ca}	10.75	10.28	10.05	9.26	8.66	8.11	7.61	7.50	7.91	9.21
AC_{ca}	61.02	58.49	57.45	54.30	52.43	51.14	50.60	51.06	53.28	57.25
HC_{pr}	13.24	12.86	12.81	13.00	13.45	14.19	15.62	16.84	18.61	20.70
SC_{pr}	34.14	32.57	31.99	29.53	28.02	26.74	25.01	23.89	22.30	20.46
RC_{pr}	15.32	14.63	14.36	13.11	12.26	11.47	11.57	12.41	16.23	22.87
AC_{pr}	62.70	60.06	59.15	55.64	53.74	52.41	52.20	53.15	57.13	64.03
HC_{cr}	12.83	12.47	12.43	12.65	13.13	13.89	15.44	16.73	18.84	21.27
SC_{cr}	32.89	31.51	30.96	28.65	27.24	26.03	24.30	23.19	21.40	19.38
RC_{cr}	21.46	20.48	20.10	18.36	17.17	16.07	20.53	23.31	33.63	43.31
AC_{cr}	67.27	64.47	63.49	59.66	57.54	55.99	60.27	63.24	73.87	83.97

8.3 Queueing at thrill rides at fairs

Consider now the queueing process at a thrill ride such as the 'Freak Out' (see Wikipedia[2] for a description). Customers arrive in groups, and are served in batches. Larger groups tend to balk less quickly, as the customers in one group also take pleasure (hopefully) in each other's company. The service time of a batch depends on the batch size, since each customer in the rider requires a safety check before the ride can take off. The problem is to determine the minimal batch size, i.e., the a parameter of the previous model, that maximizes the number of persons entering, i.e., paying.

As a simple numerical illustration, suppose couples, i.e., two customers, arrive at rate $\lambda_{s,i} = \max\{0, 1 - i/14\}$ per minute, while groups of four persons arrive as $\lambda_{l,i} = 0.25\, 1\{i \leq 20\}$, where $1\{\cdot\}$ is the indicator function. Set $\lambda_i = \lambda_{s,i} + \lambda_{l,i}$, take

$$x_i(k) = \begin{cases} \lambda_{s,i}/\lambda_i, & \text{for } k = 2, \\ \lambda_{l,i}/\lambda_i, & \text{for } k = 4, \\ 0, & \text{else.} \end{cases} \tag{19}$$

and assume complete acceptance. The service time of a batch consists of the time of the actual ride, 2 (very long) minutes, 1 min of loading and unloading, and 5 s per safety check. Assuming that the service time does not depend on the queue length, the service distribution then becomes $S_{i,k} = 3 + k/12$ minutes, where k is the batch size. Finally, the Freak Out has 16 seats, so the maximal batch size is 16.

[2] http://en.wikipedia.org/wiki/Freak_Out_(ride).

Table 2 Revenue rate λ_e as a function of the minimal batch service threshold a

a	2	4	6	8	10	12	14	16
λ_e	2.3204	2.3239	2.3230	2.2988	2.2319	2.1148	1.9495	1.6844

Clearly, the revenue rate equals the rate λ_e at which customers enter the system, which is given by

$$\lambda_e = \sum_{i \in E} p_i (2\lambda_{s,i} + 4\lambda_{l,i})$$

Note that the maximal entering rate occurs when $Q(t) = 0$, which in this case becomes $2\lambda_{s,0} + 4\lambda_{l,0} = 3$ per minute. Table 2 shows the dependency of λ_e on the minimal threshold parameter a. Interestingly, greedy service leads for this model to higher revenues than full batch service.

Acknowledgments The authors like to thank J. Riezebos and J. Slomp for helpful discussions, and the (anonymous) referees for providing valuable suggestions to improve the paper.

References

Aalto S (2000) Optimal control of batch service queues with finite service capacity and linear holding costs. Math Methods Oper Res 51(2):263–285

Asmussen S (2003) Applied probability and queues. Springer, New York

Bagchi TP, Templeton JGC (1973) A note on the $M^X/G^Y/1, K$ bulk queueing system. J Appl Probab 10(4):901–906

Bekker R (2004) Finite buffer queues with workload-dependent service and arrival rates. PhD thesis, Eindhoven University of Technology, The Netherlands

Bekker R, Borst SC, Boxma OJ, Kella O (2004) Queues with workload-dependent arrival and service rates. Queueing Syst 46(3–4):537–556

Burke PJ (1975) Delays in single-server queues with batch inputs. Oper Res 23(4):830–833

Çinlar E (1975) Introduction to stochastic processes. Prentice-Hall, Englewood Cliffs

Chang SH, Choi DW, Kim TS (2004) Performance analysis of a finite-buffer bulk-arrival bulk-service queue with variable server capacity. Stoch Anal Appl 22(5):1151–1173

Chaudry M, Templeton JGC (1983) A first course in bulk queues. Wiley, New York

Courtois PJ, Georges J (1971) On a single server finite queuing model with state-dependent arrival and service processes. Oper Res 19(2):424–435

Deb RK (1978) Optimal dispatching of a finite capacity shuttle. Manag Sci 24(13):1362–1372

Deb RK, Serfozo RF (1973) Optimal control of batch service queues. Adv Appl Probab 5(2):340–361

Dshalalow JH (1997) Queueing systems with state dependent parameters. In: Dshalalow J (ed) Frontiers in queueing: models and applications in science and engineering. CRC Press, Boca Raton, pp 61–116

Dudin AN, Shaban AA, Klimenok VI (2005) Analysis of a queue in the $BMAP/G/1/N$ system. Int J Simul Syst Sci Technol 6(1–2):13–23

Fowler JW, Hogg GL, Phillips DT (1992) Control of multiproduct bulk service diffusion/oxidation processes. IIE Trans 24(4):84–96

Germs R, Van Foreest ND (2010) Loss probabilities for the $M^X/G^Y/1/K + B$ bulk queue. Probab Eng Inf Sci 24(4):457–471

Gupta UC, Rao TSSS (1998) On the analysis of single server finite queue with state dependent arrival and service processes: $M(n)/G(n)/1/K$. OR Spectr 20(2):83–89
Hochbaum DS, Landy D (1997) Scheduling semiconductor burn-in operations to minimize total flowtime. Oper Res 45(6):874–885
Hodes B, Schoonhoven B, Swart R (1992) On line planning van ovens. Tech. rep., University of Twente, Enschede, The Netherlands
Hopp W, Spearman M (2008) Factory physics. McGraw-Hill, New York
MacGregor Smith J, Cruz FRB (2005) The buffer allocation problem for general finite buffer queueing networks. IIE Trans 37(4):343–365
Medhi J (2003) Stochastic models in queueing theory. Academic Press, San Diego
Neuts MF (1977) Algorithmic methods in probability theory. North-Holland, Amsterdam
Nobel RD (1989) Practical approximations for finite-buffer queueing models with batch arrivals. Eur J Oper Res 38(1):44–55
Schellhaas H (1983) Computation of the state dependent probabilities in $M/G/1$ queues with state dependent input and state dependent service. OR Spectr 5(4):223–228
Takagi H (1993) Queueing analysis: finite systems. North-Holland, Amsterdam
Tijms HC, Van Hoorn MH (1981) Algorithms for the state probabilities and waiting times in single server queueing systems with random and quasirandom input and phase-type service times. OR Spectr 2(3):145–152
Uzsoy R, Lee CY, Martin-Vega LA (1994) A review of production planning and scheduling models in the semiconductor industry, part ii: shop-floor control. IIE Trans 26(5):44–55
Van der Zee DJ, Van Harten A, Schuur P (2001) On-line scheduling of multi-server batch operations. IIE Trans 33(7):569–586
Weiss HJ (1979) The computation of optimal control limits for a queue with batch services. Manag Sci 25(4):320–328