Near-optimal switching strategies for a tandem queue

van Leeuwen, D.; Núñez-Queija, R.

DOI
10.1007/978-3-319-47766-4_17

Publication date
2017

Document Version
Submitted manuscript

Published in
Markov Decision Processes in Practice

Citation for published version (APA):
van Leeuwen, D., & Núñez-Queija, R. (2017). Near-optimal switching strategies for a tandem queue. In R. J. Boucherie, & N. M. van Dijk (Eds.), *Markov Decision Processes in Practice* (pp. 439-459). (International Series in Operations Research and Management Science; Vol. 248). Springer. https://doi.org/10.1007/978-3-319-47766-4_17

Near-optimal switching strategies for a tandem queue

Daphne van Leeuwen, Rudesindo Núñez-Queija

December 23, 2015

Abstract

Motivated by various applications in logistics, road traffic and production management, we investigate two versions of a tandem queueing model in which the service rate of the first queue can be controlled. The objective is to keep the mean number of jobs in the second queue as low as possible, without compromising the total system delay (i.e. avoiding starvation of the second queue). The balance between these objectives is governed by a linear cost function of the queue lengths.

In the first model, the server in the first queue can be either switched on or off, depending on the queue lengths of both queues. This model has been studied extensively in the literature. Obtaining the optimal control is known to be computationally intensive and time consuming. We are particularly interested in the scenario that the first queue can operate at larger service speed than the second queue. This scenario has received less attention in literature. We propose an approximation using an efficient mathematical analysis of a near-optimal threshold policy based on a matrix-geometric solution of the stationary probabilities that enables us to compute the relevant stationary measures more efficiently and determine an optimal choice for the threshold value.

In some of our target applications, it is more realistic to see the first queue as a (controllable) batch-server system. We follow a similar approach as for the first model and obtain the structure of the optimal policy as well as an efficiently computable near-optimal threshold policy.

We illustrate the appropriateness of our approximations using simulations of both models.

1 Introduction

We investigate a dynamic flow control problem arising in various applications. As a motivating illustration consider road traffic control, where trucks enter a crowded metropolitan area to supply goods in the city center. More often than not, such a scenario leads to clustering of transportation traffic near distribution facilities in the city. Our specific aim here would be to develop a control method that reduces long waiting lines of trucks at distribution centers located in or near cities. As a solution we investigate the effectiveness of a buffer location (i.e. a parking facility for trucks) near a distribution center to reduce the number of waiting trucks in busy areas, thereby giving more space to other traffic and reducing emissions.

Indeed, the buffer location will temporarily 'store' trucks and prevent overly crowded areas near the distribution center. On the down side, the introduction of the buffer location introduces an additional hop in the route for the trucks, creating potential inefficiency. When poorly operated, trucks may be waiting at the buffer location, while the service location at the distribution center may have cleared all the local backlog.

The problem setting described here illustrates a generic challenge in transportation logistics, manufacturing and production management. One may for example think of an asphalting machine that must be supplied with liquid asphalt at the correct pace, avoiding too long storage of the perishable material, but also maintaining sufficient supply to avoid expensive shutdown of the machine due to lack of material. In a production assembly line one can also imagine the necessity to balance the local inventory of assembly parts with the available space. And in road congestion management the traffic density near bottleneck junctions must be kept low enough to avoid traffic dead-lock, but on the other hand in the upstream direction the traffic flow should be sufficient to prevent unnecessary delay. We discuss and describe such control problems and design a generic strategy with practical applicability.

The above sketched situations are modeled by a controllable two-stage tandem queue. Referring to our initial motivation, the first stage represents the buffer and has infinite capacity. We first concentrate on the setting in which the server at the first stage can be controlled by an on-off switch. The second stage represents the distribution center for which we want to reduce the number of trucks. We seek an optimal trade-off between the reduction in the number of vehicles at the second stage and the additional delay caused by the on/off policy at the first stage. The optimal operation point is determined by the minimization of a cost function. This function accounts for waiting time costs in the buffering stage as well as costs for waiting at the distribution center. Arrivals to this system are modeled by a Poisson process and "service times" in both queues are exponentially distributed, which facilitates a formulation as a Markov Decision Problem (MDP).

This model has been studied extensively in literature and it is known that it is optimal to serve either at full speed or not to serve at all[14]. This type of strategies are referred to as "bang-bang strategies". Under certain cost assumptions it has been shown that the structure of the optimal service rate gives a switching strategy dividing the state space into two regions: one where the service rate is at its maximum and one where service is paused. Such structural properties of optimality can be proven by showing the convexity of the value function in the Bellman optimality equation, see for example [14] and [6]. Similar convexity properties also hold for networks of queues [15]. Adopting the optimal control, several strategic questions can be answered as well, such as the desirable distance of the buffer location from the distribution center in order to regulate trucks optimally. In our model this distance is captured by the service rate at the first stage.

Full evaluation of the theoretically optimal strategy is often numerically infeasible, or at the least numerically overly time consuming for practical purposes. Several papers have looked at approximation techniques for this model. The fluid approximation developed in [1] works well when the second station works at a higher rate than the first station. We will summarize and complement that analysis with the opposite case (when the second station is the slowest) and show that then the optimal strategy can well be approximated by a threshold-based decision rule. The author in [10, p.439-441] developed a method to determine an approximation for the threshold value for the discounted reward MDP, which does not have a direct counterpart in the average reward problem. Alternatively, the authors in [5] propose a fluid approximation, focusing on a large deviations analysis. They determine the most probable evolution of the system toward rarely visited states, whereas we are concerned with controlling the system for optimal *average* behavior, i.e., we concentrate on states that are visited frequently.

In this chapter we discuss an approximation technique for the best threshold value in order to reduce the computational effort. This method uses matrix geometric analysis as described in [11]. Various papers, such as [4], have previously applied this method to speed up computation. The exposition in [7] for a tandem queue similar to ours is particularly relevant to develop our approximation as it gives an explanation of the blocks which are necessary to capture the details of our model.

In practice, service at the first station may not be limited to one job at a time. Indeed, in our primary example several trucks may jointly leave the parking facility if the waiting line at the distribution center is very short. Similarly, in manufacturing and production planning, several items may be produced or delivered at the same time. We therefore proceed to study an extension of the tandem queue with controllable service rate, allowing for batch service in the first station. A service batch corresponds, for example, to platoons of trucks jointly driving from the buffer location to the distribution center. In this setting, it is reasonable to maintain the service rate

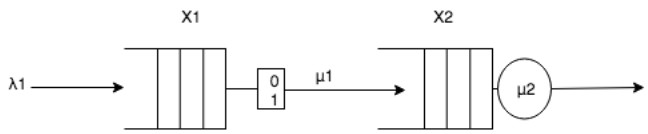

Figure 1: *The tandem queue with an on-off controlling mechanism.*

independent of the number of jobs that are jointly processed. We study the impact of batch service at the first server and determine the structure of the optimal policy. It turns out that the optimal queue level at the second queue is fully determined by the aggregate number of jobs in the two queues (i.e., the sum of the two individual queues). If the optimal levels can be determined, the optimal batch size is then easily computed for all states in the state space.

The batch model will be approximated in the same manner as the basic on-off model. The extension of the matrix geometric method for use in the batch model follows along the same lines as [12]. That reference focuses on a system which requires a minimum batch size to initiate service, and additionally, service can be granted up to a predefined maximum batch size. In various other papers optimal batch sizes are determined via a trade-off between startup costs for service and costs per unit time for jobs in the system, see e.g. [3],[16]. Our model, however, does not have startup costs for batch service. We determine the optimal batch size leading to an optimal threshold level based on properties arising from the MDP formulation.

This chapter is structured as follows. We give a detailed description of the basic model in Section 2. In Section 3 we study the structure of the optimal policy for various choices of the parameters and observe that for most cases of interest in our context, the shape of the optimal switching curve suggests it can well be approximated by a threshold policy. In Section 4 we set out to determine the best choice for the threshold value using matrix-geometric analysis techniques. We then turn our attention to the batch-service model in Section 5 and follow the same program: we investigate the structural properties of the optimal strategy and develop a matrix-geometric representation to numerically determine a near-optimal strategy. In Section 6 we numerically study the appropriateness of the proposed strategies for both models. We conclude the chapter in Section 7.

2 Model description: Single service model

Our model consists of two queues in tandem. As alluded to before, the second queue represents the actual service facility (distribution center, production plant or assembly line), whereas the first queue serves as a temporary buffer to alleviate congestion in the second queue. For analysis purposes we assume that jobs arrive to the first queue according to a Poisson process at rate λ and jobs *can be* processed at rate μ_1. After service in the first queue, jobs proceed to the second queue, for which the service rate is denoted with μ_2. For stability we assume both $\lambda < \mu_1$ and $\lambda < \mu_2$.

So as to control the number of jobs at the second station we introduce a binary decision at the first station, depicted in Figure 1. The control mechanism may be interpreted as an on/off switch at the first station with two states: $\{0, 1\}$. State 0 represents a service rate of 0, i.e. all jobs waiting at the first station will be blocked for service. State 1 represents the situation where each job at station 1 is served by rate μ_1 and continues to stage 2.

To formulate this as an optimization problem we introduce constant waiting costs c_{wait}, which are incurred *per job* and *per unit of time*. Jobs queueing at the second station encounter additional costs indicated by c_{loc} which, in our introductory example, represents the costs of residing in the distribution area. Thus, the waiting cost at the first server is $c_1 = c_{wait}$ per job per unit of time, and at the second station it is $c_2 = c_{wait} + c_{loc}$. Naturally, we assume $0 < c_1 < c_2$. Due to larger costs at station 2, it is more advantageous to hold customers in queue 1 rather than in queue 2. However, one should avoid an empty station 2 when station 1 still has a backlog. We seek an efficient trade-off between these two effects.

We formulate the problem as a Markov Decision Process. The system will be observed at epochs of arrivals and service completions i.e. in discrete time. We use uniformization to discretize the Markov chain as described in Lippman [9]. Our discrete-time Markov Decision Process consists of the quadruple $\{\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{C}\}$. $\mathcal{S}$ represents the state space of the system, and is defined as $i = (x_1, x_2) \in \mathbb{N}^2$. Here x_i is the number of jobs at stations $i = 1$ and $i = 2$, respectively. $\mathcal{A}$ represents the action space; the set of actions that decision makers can take. In this problem $\mathcal{A} = \{0, 1\}$ representing either a blocked or an unblocked first server, respectively. Action 0 blocks service at station 1, i.e. no jobs can move from station 1 to station 2. For action 1 jobs are served at the first station at rate μ_1 and then move from station 1 to station 2. $\mathcal{P}$ contains the transition probabilities from state i to state j for action $a \in \mathcal{A}$; these can be written as $p^a(i, j)$. Finally, $\mathcal{C}$ denotes the cost function and will be written as $c^a(i)$ which is the expected cost per unit of time for each state $i = (x_1, x_2) \in \mathcal{S}$ and action $a \in \{0, 1\}$.

An optimal strategy satisfies Bellman's equation [2, 13]:

$$V^*(i) + g^* = \min_{a \in \mathcal{A}} \Big\{ c^a(i) + \sum_{j \in \mathcal{S}} p^a(i, j) V^*(j) \Big\} \text{ for } i \in \mathcal{S}, \tag{1}$$

where g^* and $V^*(i)$ give the optimal average reward and value function. The decision rule can be determined by:

$$f(i) = \text{argmin}_{a \in \mathcal{A}} \Big\{ c^a(i) + \sum_{j \in \mathcal{S}} p^a(i, j) V^*(j) \Big\} \text{ for } i \in \mathcal{S}, \tag{2}$$

where $V^*(j)$ satisfies $V^*(i) + g^* = c^f(i) + \sum_{j \in \mathcal{S}} p^f(i, j) V^*(j)$. Note the slight abuse in notation in writing $c^f(i)$ and $p^f(i, j)$ instead of $c^{f(i)}(i)$ and $p^{f(i)}(i, j)$ as we should have according to our earlier notation. Our goal is to minimize the average cost and determine an optimal decision for each state.

To determine the optimal strategy in our tandem queue we use equation (2) where $c^a(i)$, with $i = (x_1, x_2)$, is given by $c_1 x_1 + c_2 x_2$. Recall that the cost c_1 consists only of the waiting cost per job at station 1 and c_2 is a combination of the waiting costs and additional costs for station 2, and that we assume $0 < c_1 < c_2$.

The transition probabilities $p^a(i, j)$ are determined by the transition rates in each state, applying uniformization as described in Lippman [9]. For action $a = 1$ (service in queue 1), we have for $x_1 \geq 0$ and $x_2 \geq 0$: $p^1((x_1, x_2), (x_1 + 1, x_2)) = \lambda/(\lambda + \mu_1 + \mu_2)$, $p^1((x_1 + 1, x_2), (x_1, x_2 + 1)) = \mu_1/(\lambda + \mu_1 + \mu_2)$, and $p^1((x_1, x_2 + 1), (x_1, x_2) = \mu_2/(\lambda + \mu_1 + \mu_2)$. On the boundary we have "dummy transitions" leading to $p^1((0, x_2 + 1), (0, x_2 + 1)) = \mu_1/(\lambda + \mu_1 + \mu_2)$, $p^1((x_1 + 1, 0), (x_1 + 1, 0)) = \mu_2/(\lambda + \mu_1 + \mu_2)$, and $p^1((0, 0), (0, 0)) = (\mu_1 + \mu_2)/(\lambda + \mu_1 + \mu_2)$.
Similarly, when $a = 0$ (no service in queue 1), we have for $x_1 \geq 0$ and $x_2 \geq 0$: $p^0((x_1, x_2), (x_1 + 1, x_2)) = \lambda/(\lambda + \mu_1 + \mu_2)$, and $p^0((x_1, x_2 + 1), (x_1, x_2) = \mu_2/(\lambda + \mu_1 + \mu_2)$. Now there can be no service in queue 1, giving $p^0((x_1, x_2 + 1), (x_1, x_2 + 1)) = \mu_1/(\lambda + \mu_1 + \mu_2)$. Finally, the remaining transitions on the boundary are $p^0((x_1, 0), (x_1, 0)) = (\mu_1 + \mu_2)/(\lambda + \mu_1 + \mu_2)$.

Successive Approximation (SA) will be used to calculate the optimal decision for each state so as to minimize average costs:

$$V_n(i) = \min_{a \in \mathcal{A}_i} \Big\{ c^a(i) + \sum_{j \in \mathcal{S}} p^a(i, j) V^*_{n-1}(j) \Big\},$$

and

$$f_n(i) = \text{argmin}_{a \in \mathcal{A}_i} \Big\{ c^a(i) + \sum_{j \in \mathcal{S}} p^a(i, j) V^*_{n-1}(j) \Big\}.$$

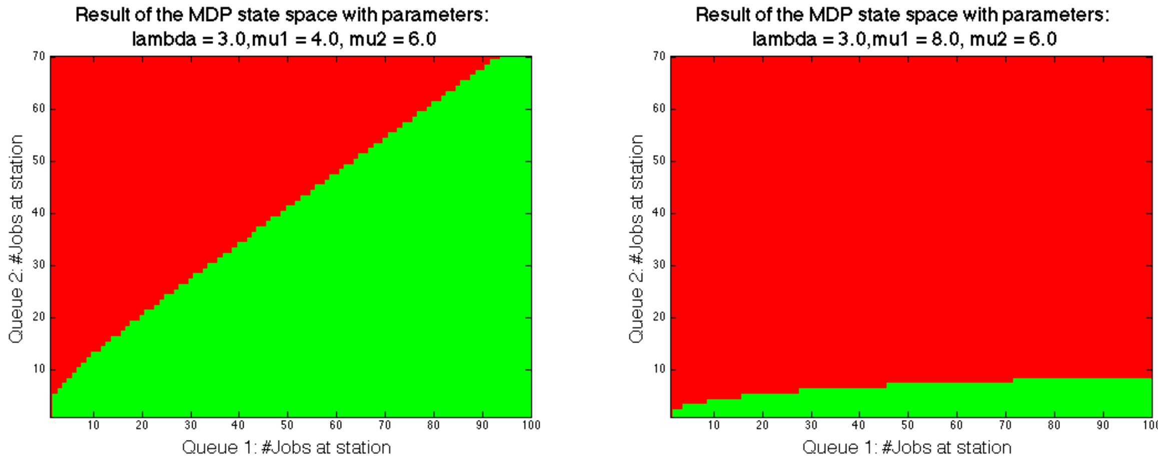

Figure 2: *An illustration of the optimal actions for all states. In the figures red indicates blocking (jobs are not served in queue 1) and green indicates that jobs at server 1 are served at rate* μ_1.

3 Structural properties of an optimal switching curve

We start our discussion of the optimal strategy with a numerical illustration for a particular example. Here and in the remainder of the chapter we will use $c_1 = 1$ and $c_2 = 3$ meaning that jobs incur waiting costs of 1 per unit of time and, only in queue 2, an additional location cost of 2 units. These values were also chosen in the example used by Meyn [10] for the discounted case. Our structural results hold for all values that satisfy $0 < c_1 < c_2$.

First we illustrate a dichotomy that occurs between the cases $\mu_1 < \mu_2$ i.e. the first server serves jobs at lower speed than the second server, and the opposite $\mu_1 > \mu_2$. The two graphs in Figure 2 show the optimal strategy for the MDP under these two settings. A red color indicates that it is optimal to block service at the first stage. The green color implies a system working at maximum service speed at both stages. We observe that in both cases, the optimal action is prescribed by a so-called "switching curve" separating the green area from the red area. The shapes of the switching curves in the two graphs are a bit different. On the left, the curve eventually grows with a constant slope (this will be explained), whereas the graph on the right flattens for larger values of x_1. This difference appears to be fundamental to the two chosen parameter sets: one where the first server is slower than the second, and the opposite case. We will discuss this in more detail below.

To have a better understanding of the dynamics of the system operating under such a switching curve, we include the drift and trajectory diagrams displayed in Figures 3 and 4. Irrespective of the shape of the switching curve, the drift above the curve is positive in the horizontal direction (due to arrivals at rate λ) and negative in the vertical direction (by departures from the second queue at rate μ_2). Note that because of the stability condition $\lambda < \mu_2$, the horizontal component of the drift is smaller than that in the vertical direction, but that is irrelevant for our discussion here. Below the curve, the horizontal drift changes sign and has magnitude $\mu_1 - \lambda_1$, which is positive due to the stability condition $\lambda < \mu_1$. In the vertical direction the drift is $\mu_1 - \mu_2$. Here we observe a distinction between the case $\mu_1 < \mu_2$ in Figure 3 and the case $\mu_1 > \mu_2$ in Figure 4. In the first case ($\mu_1 < \mu_2$), we obtain a negative vertical drift and a corresponding direction toward the horizontal axis. If $\mu_1 > \mu_2$, the vertical drift is positive and the trajectory is directed toward the switching curve from both sides.

We now return to Figure 2. The graph on the left suggests a close approximation of the switching curve by a linear function with positive intercept at the vertical axis. The graph on the right rather suggests an approximation by a horizontal line. The difference in behavior can be explained by

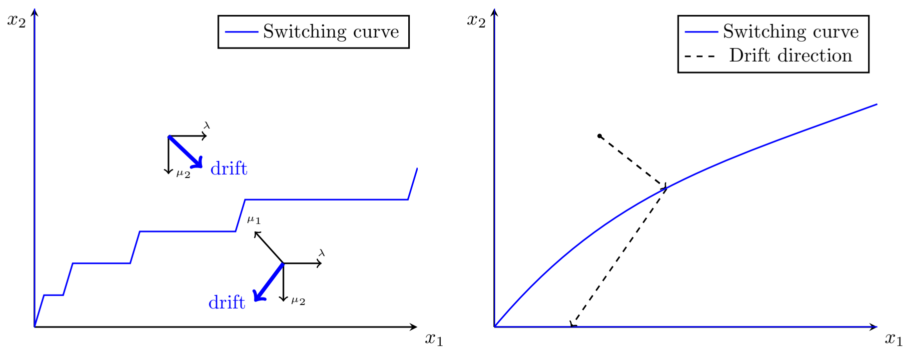

Figure 3: *An illustration of the drift above and below the switching curve (left graph) and a typical trajectory (right graph) ;* $\mu_1 < \mu_2$.

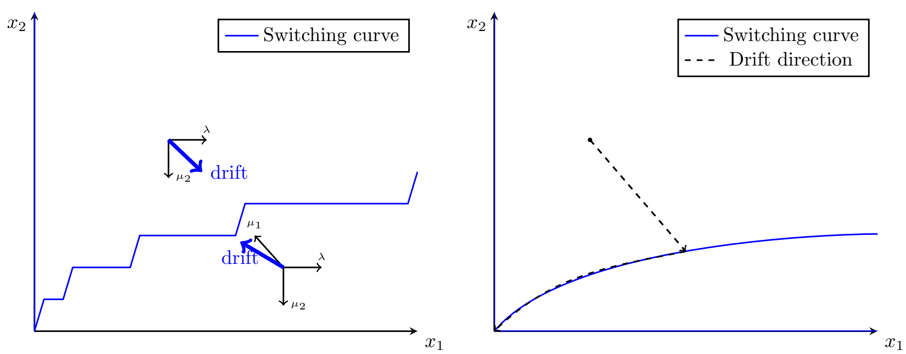

Figure 4: *An illustration of the drift above and below the switching curve (left graph) and a typical trajectory (right graph);* $\mu_1 > \mu_2$.

the parameter choice. The linear increasing curve is the effect of a larger service capacity at the second stage, $\mu_1 < \mu_2$. Intuitively, an optimal strategy must aim at avoiding an empty queue 2, when there are jobs in queue 1. The undesirable states are therefore located on the horizontal axis. If $\mu_1 < \mu_2$, the first queue can not "catch up" with the second queue, and therefore it should always provide sufficient inflow for queue 2 even at large system states. To further explain this fact, we refer to Figure 3. The typical trajectory leads to the horizontal axis, which is the set of undesirable states. The linearly increasing switching curve avoids that the horizontal axis is hit at a very large level. When $\mu_1 > \mu_2$, the first server *can* catch up with the second server, because it serves at higher speed, thereby decreasing the probability of starvation of the second stage. All trajectories lead to the switching curve and then continue along the switching curve toward the origin. Hence, the size of the second queue can be maintained at a low level and consequently the switching curve flattens for larger levels of the first queue.. This fundamental difference leads to a likewise fundamentally different analysis of these two cases.

The first case, $\mu_1 < \mu_2$, has been investigated thoroughly in [1] using a fluid approximation. In this fluid approximation, the random trajectories are replaced by deterministic ones, characterized by their (expected) drifts. The fluid approximation can be shown to be the exact limit of the stochastic process under an appropriate scaling. The limiting fluid process can be shown to have an optimal linear switching curve, which translates into the optimal action for large system states, but lacks information about the optimal strategy near the origin. As we have observed, at the origin, the optimal switching curve for the stochastic model has a vertical offset. That offset can be approximated using perturbation methods [1], and turns out to give a good representation of the optimal strategy.

Unfortunately this method does not work when $\mu_1 > \mu_2$, which is the more relevant setting for many of our motivating applications. For example, the "buffer" location for the distribution center will likely not be located very far from the distribution center, which corresponds to relatively large values of μ_1. The above fluid approximation applied to $\mu_1 > \mu_2$ gives a switching curve that lies on the horizontal axis which suggests that the first server should never be operated. This is well explained by the sub-linear shape of the switching curve in the right graph of Figure 2. On a linear scale, this graph vanishes for large system states.

We therefore set out to obtain an approximate analysis for the case $\mu_1 > \mu_2$. Theoretically it can be seen that the switching curve still increases indefinitely, albeit at a sub-linear pace. The flat shape however, implies that over large ranges of buffer levels in queue 1, the optimal action switches at a common buffer level of queue 2. This suggests that the optimal switching curve may well be approximated by a horizontal line, i.e. that a fixed threshold-based strategy should be close to optimal. Obtaining the optimal threshold value from the Bellman equations is computationally hard. In the sequel we use a matrix geometric analysis [11] to compute the best threshold value and compare it to the optimal strategy. An alternative method to approximate near-optimal threshold values for the discounted reward MDP was developed in [10, p.439-441]. Unfortunately, when applied to the average reward problem (under the usual limiting argument for discounted reward models [13, 8.2.2]) the threshold value becomes equal to infinity. For the purpose of this chapter our focus lies on an approximation for the average reward case. Alternatively, the authors of [5] approximate the curve using a large-deviations analysis. In that scaling, they obtain an asymptotically optimal switching level.

4 Matrix geometric method for fixed threshold policies

We have argued that for the case $\mu_1 > \mu_2$ the optimal switching curve can perhaps be well approximated by a horizontal line. In order to compare the effectiveness of such fixed-threshold strategies, we set out to determine the relevant performance measures as functions of the threshold parameter K and then pick the best value of K. In this section we show that the resulting model falls into the class of Quasi-Birth-Death processes that allow for a matrix geometric solution. In order to cast our model into the framework of [11], we partition the state space into levels and phases, resulting in the generic structure of the generator matrix displayed in 3. In our model, each level will correspond with a fixed number of jobs in the first queue, and the phase within a level represents the number of jobs in the second queue. Thus, the generator matrix can be written in the block form of 3. Transitions between blocks correspond to a change in level (queue 1) and transitions within a block represent a change in phase (queue 2). The number of levels is therefore unbounded and the size of the block matrices (corresponding to the number of phases) is $K + 1$, where K is the fixed threshold level.

Formally the state space can be described by $\mathcal{S} = \{(x_1, x_2) : x_1 \in \mathbb{Z}^+, 0 \leq x_2 \leq K + 1\}$. The level index x_1 denotes the number of jobs at station 1 and x_2, the phase index, represents the number of jobs at station 2. The maximum number of jobs at station 2 is now bounded by the threshold K.

The generator matrix Q for this system is given by:

$$Q = \begin{bmatrix} B_0 & \Lambda & 0 & 0 & 0 & 0 & \cdots \\ M & D & \Lambda & 0 & 0 & 0 & \cdots \\ 0 & M & D & \Lambda & 0 & 0 & \cdots \\ 0 & 0 & M & D & \Lambda & 0 & \cdots \\ 0 & 0 & 0 & M & D & \Lambda & \cdots \\ 0 & 0 & 0 & 0 & M & D & \cdots \\ 0 & 0 & 0 & 0 & 0 & M & \cdots \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \end{bmatrix} \tag{3}$$

In this representation all blocks are square matrices of order $K+1$, and $M+D+\Lambda$ is a generator of a $K+1$ dimensional Markov process that follows the transitions of the second queue, *conditional on a non-empty first queue*. The stability condition is given by Neuts' mean drift criterion [11]: We define π to be the equilibrium distribution of a Markov process with generator $M+D+\Lambda$:

$$\underline{\pi}(M + D + \Lambda) = \underline{0} \text{ where } \pi\mathbf{e} = 1,$$

$\mathbf{e}$ being a $K+1$ dimensional vector with all entries equal to 1. The process with generator Q is stable if and only if $\underline{\pi}M\mathbf{e} > \underline{\pi}\Lambda\mathbf{e}$, i.e., the drift to higher levels should be strictly less than the drift to lower levels to guarantee stability of the system. For a fixed threshold level K the blocks are defined as follows:

$$B_0 = \begin{bmatrix} -\lambda & \cdots & \cdots & \cdots & 0 \\ \mu_2 & -a_1 & \cdots & \cdots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \mu_2 & -a_1 & \vdots \\ 0 & \cdots & \cdots & \mu_2 & -a_1 \end{bmatrix}, \Lambda = \begin{bmatrix} \lambda & \cdots & \cdots & \cdots & 0 \\ \vdots & \lambda & \cdots & \cdots & \cdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \cdots & \cdots & \lambda & \vdots \\ 0 & \cdots & \cdots & \cdots & \lambda \end{bmatrix},$$

with $a_1 = \lambda + \mu_2$,

$$D = \begin{bmatrix} -a_2 & \cdots & \cdots & \cdots & 0 \\ \mu_2 & -a_2 & \cdots & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \mu_2 & -a_2 & \cdots \\ 0 & \cdots & \cdots & \mu_2 & -a_1 \end{bmatrix}, M = \begin{bmatrix} 0 & \mu_1 & \cdots & \cdots & 0 \\ \vdots & \cdots & \mu_1 & \cdots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \cdots & \cdots & \cdots & \mu_1 \\ 0 & \cdots & \cdots & \cdots & 0 \end{bmatrix},$$

with $a_2 = \lambda + \mu_1 + \mu_2$.

The repetitive block structure implies a matrix geometric form for the stationary probabilities corresponding to Q. Defining the $K+1$ dimensional vectors $\underline{\pi_i} = (\pi_{i0} \ldots \pi_{iK})$, where $\pi_{x_1\,x_2}$ is the stationary probability of having x_1 jobs in the first and x_2 jobs in the second queue, the balance equations are:

$$\underline{\pi}_{i-1}\Lambda + \underline{\pi}_i D + \underline{\pi}_{i+1} M = \underline{0} \text{ for } i \geq 1,$$

and we can write:

$$\underline{\pi}_i = \underline{\pi}_{i-1} R \rightarrow \underline{\pi}_i = \underline{\pi}_0 R^i,$$

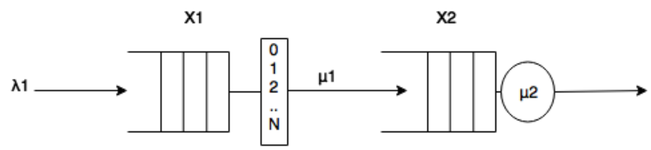

Figure 5: *Graphical representation of the tandem queue with batch service in the first queue.*

where the matrix R is the minimal non-negative solution to the following quadratic matrix equation:

$$R^2M + RD + \Lambda = R.$$

Via iteration we may determine R (there are alternative and more efficient routines, see [8]). Once we have determined a solution for R we can include the boundary conditions. It then remains to compute $\underline{\pi}_0$ via the remaining boundary equation:

$$\underline{\pi}_0 B_0 + \underline{\pi}_1 \Lambda = 0.$$

For a unique solution we impose the normalization condition that the probabilities sum to 1. This gives:

$$\underline{\pi}_0 \mathbf{e} + \sum_{i=1}^{\infty} \underline{\pi}_0 R^i \mathbf{e} = 1 \text{ or equivalently } \underline{\pi}_0 (I - R)^{-1} \mathbf{e} = 1.$$

In order to compute the cost function we determine the average queue length for both queues:

$$E[x_1] = \pi_0 R(I - R)^{-2}\mathbf{e},$$
$$E[x_2] = \pi_0 (I - R)^{-1} J,$$

where J is the column vector $(0, 1, \ldots, K-1)^T$.

To determine the optimal threshold we simply minimize costs over all thresholds:

$$\min_{K \in N} \{\pi_0 (I - R)^{-1} * (c_1 R(I - R)^{-1}\mathbf{e} + c_2 J).\} \quad (4)$$

Now we can compute the best threshold level with respect to the costs and compare it with the MDP policy. From now on we will refer to this policy as the "optimal threshold" policy, not implying that this policy is overall optimal, but rather that it is optimal among the threshold policies. Determining the optimal threshold is much less computationally demanding than finding the optimal strategy using the MDP approach.

5 Model description: Batch transition model

We now extend the model allowing for batch services in the first queue. As we elaborated on in the introduction, processing multiple jobs at the first server to prevent starvation at the second server is a logical choice for various applications of this model. To clarify in what ways this model extends the previous one, we will describe it while referring to the details of the first model. To gain understanding of the new model a graphical representation is shown in Figure 5. Compared to Figure 1 we can see that the first queue is serving N jobs in one single service instead of handling jobs individually.

So as to allow the first queue to serve in batches we extend the decision space from $a \in \{0, 1\}$ to $a \in \{0, \ldots, x_1\}$ when the number of jobs in the first queue is x_1 (the set of possible actions is thus dependent on the current state). The value of a corresponds to the chosen batch size, which is naturally limited by the number of jobs in the first queue. Next, we adapt the transition probabilities described in Section 2. For $i = (0, 0)$ we have

$$p^a(i,j) = \begin{cases} \frac{\lambda}{\lambda+\mu_1+\mu_2} & \text{if } j = (1,0) \\ \frac{\mu_1+\mu_2}{\lambda+\mu_1+\mu_2} & \text{if } j = (0,0) \end{cases},$$

and for $i = (x_1, x_2) \neq (0, 0)$:

$$p^a(i,j) = \begin{cases} \frac{\lambda}{\lambda+\mu_1+\mu_2} & \text{if } j = (x_1+1, x_2) \\ \frac{\mu_1}{\lambda+\mu_1+\mu_2} & \text{if } j = (x_1 - a, x_2 + a) \text{ for } a \in \{0, \ldots, x_1\} \\ \frac{\mu_2}{\lambda+\mu_1+\mu_2} & \text{if } j = (x_1, x_2 - 1) \qquad \text{or } i = j = (x_1, 0) \end{cases}. \tag{5}$$

For this system there's always a strategy that stabilizes it as long as $\lambda < \mu_2$, irrespective of the value of $\mu_1 > 0$. This is obvious, since we can always choose to serve all jobs in queue 1 in a single batch, no mater how many there are. Different from the single service model, there will be no clear distinction between the cases $\mu_1 > \mu_2$ and $\mu_1 < \mu_2$, because the first station is always able to 'catch up' with the second station, even for $\mu_1 < \mu_2$. In the batch transition model we will see that the switching curve always flattens for larger x_1 values as is illustrated in Figure 6. More details on this figure will be given when we investigate the structural properties of the batch service model.

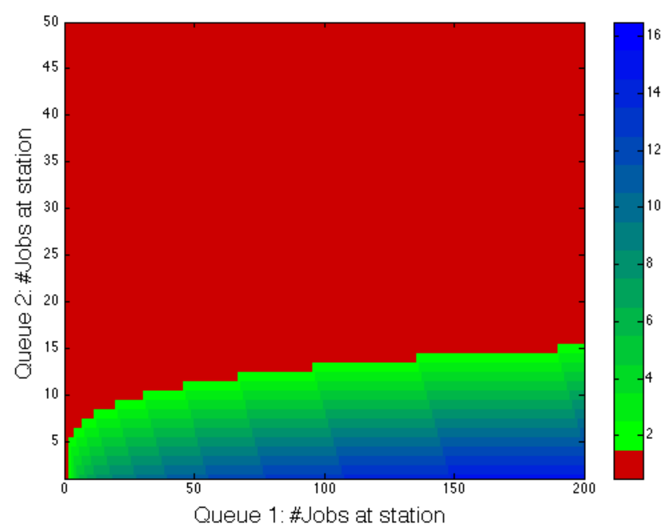

Figure 6: *Output of the MDP for the batch service model for parameter set $\lambda = 4$, $\mu_1 = 2$ and $\mu_2 = 6$. Each color represents a different optimal batch size for the current state.*

5.1 Structural properties of the batch service model

We will apply similar numerical calculations to show the main structural properties of the batch service model and compare with the single service model. For the single service model, the state

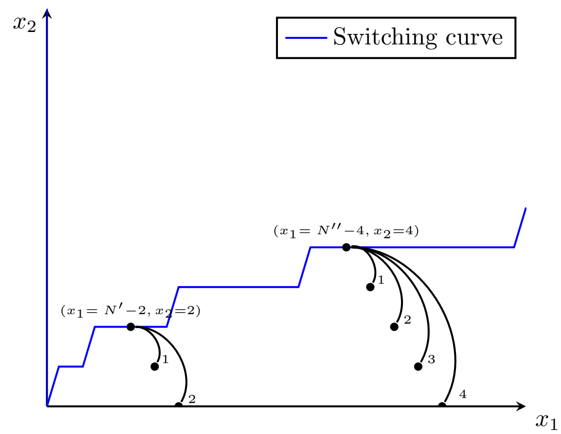

Figure 7: *A graphical representation of the optimal batch size for two examples.*

space was divided into two regions depending on the optimal decision. For the batch transition model similar shapes are encountered when grouping states with the same optimal decision. In Figure 6 states colored in red correspond as before to blocking of service at the first queue. The next layer corresponds to states in which the optimal batch size is one, then we have states with an optimal batch size of two, etcetera. This figure suggests that the optimal trajectories of the process are near the curve dividing red from green colored states. In this numerically obtained graph, the shape of this curve is again sublinear; we will indeed show that this is confirmed when investigating a scaled version of the process. We note once more, that this shape is not restricted to particular parameter settings, as was the case in the single batch model where the sublinear shape corresponded to the choice $\mu_1 > \mu_2$. The larger jumps now allow the system to move to the switching curve in one step from any state.

Although this is rather difficult to see from Figure 6, we observe that the optimal strategy can again be characterized by the *single* switching curve separating the red states from all others. Given the total number of jobs in the system, say $x_1 + x_2 = N$, the optimal action is to serve a jobs in the first queue such that $(x_1 - a, x_2 + a)$, which also has N jobs in total, is *on* the switching curve. Should this value of a be negative (this happens when (x_1, x_2) is in the red area), then no jobs should be served in the first queue.

A graphical representation of the optimal transitions can be seen in Figure 7 for two different values of the total number of jobs in the system: $N = N'$ and $N = N''$. All states on a diagonal $x_1 + x_2 = N$ "point" to the same destination state at the intersection of this line and the switching curve.

5.2 Matrix Geometric method with batch services

Similar to the case $\mu_1 > \mu_2$ in the single service model, we wish to approximate the (sublinear) switching strategy with a horizontal line, thereby implementing a threshold-based strategy with, say, threshold value K. This model falls into the class of G/M/1 type Markov chains that admit a matrix geometric solution for the stationary distribution. Note that, for a fixed threshold value, the condition $\lambda < \mu_2$ is not sufficient for stability. For sure, the system can not be stable if $\lambda \geq K\mu_1$, because $K\mu_1$ is the maximum rate at which jobs can be pushed from the first station. The additional condition $\lambda < K\mu_1$ is therefore necessary, but certainly not sufficient either. The precise stability condition can be shown to be

$$\lambda < \mu_1 \left(K \left(\frac{\mu_2}{\mu_1 + \mu_2} \right)^K + \sum_{k=1}^{K-1} k \frac{\mu_1}{\mu_1 + \mu_2} \left(\frac{\mu_2}{\mu_1 + \mu_2} \right)^k \right). \tag{6}$$

This can be obtained by interpreting the right hand side of this inequality as the exact departure rate from the first station if that station were saturated (i.e., starting with infinitely many jobs in station 1). We will not make this precise here, as we can obtain this equation from Neuts' mean drift condition, which we will do below.

To define the batch transition model in matrix geometric form extra blocks must be added into the generator matrix, that allow for the larger transition jumps. As was described in the previous section, the batch size can be derived from the switching curve, effectively redistributing the total number of jobs over the two queues (with the obvious limitation that no jobs can be moved from the second to the first queue).

The generator matrix Q_{batch} now has the following structure:

$$Q_{batch} = \begin{bmatrix} B_0 & \Lambda & 0 & 0 & 0 & \cdots & \cdots \\ B_1 & D & \Lambda & 0 & 0 & \cdots & \cdots \\ B_2 & M_1 & D & \Lambda & 0 & \cdots & \cdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \cdots \\ B_{K-1} & M_{K-2} & M_{K-3} & \cdots & \cdots & \ddots & \cdots \\ B_K & M_{K-1} & M_{K-2} & \cdots & \cdots & \ddots & \cdots \\ 0 & M_K & M_{K-1} & \ddots & \ddots & \cdots & \cdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \end{bmatrix} \tag{7}$$

The 0^{th} level of the process represents the boundary states, comparable to the single service model. The matrices B_0, Λ and D are defined as before. Because of the batch services, the block matrices below the diagonal must be adapted. The matrices B_k, $k - 1, 2, \ldots, K$ correspond to transitions for which the first queue is emptied. This is only possible when there are 1 up to K jobs in the first queue, and the second queue has sufficient space left to accommodate the batch size:

$$B_1 = M = \begin{bmatrix} 0 & \mu_1 & 0 & \cdots & 0 \\ 0 & 0 & \mu_1 & 0 & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & \mu_1 \\ 0 & \cdots & \cdots & \cdots & 0 \end{bmatrix}, B_2 = \begin{bmatrix} 0 & 0 & \mu_1 & \cdots & 0 \\ 0 & 0 & 0 & \mu_1 & \vdots \\ \vdots & \ddots & \ddots & 0 & \mu_1 \\ \vdots & \cdots & \cdots & \cdots & 0 \\ 0 & \cdots & \cdots & \cdots & 0 \end{bmatrix}, \cdots, B_K = \begin{bmatrix} 0 & \cdots & \cdots & 0 & \mu_1 \\ 0 & \cdots & \cdots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \cdots & \cdots & \cdots & \vdots \\ 0 & \cdots & \cdots & \cdots & 0 \end{bmatrix},$$

The transitions corresponding to batch services that do not lead to an empty first station are grouped in the matrices M_k:

$$M_K = B_K, M_{K-1} = \begin{bmatrix} 0 & \cdots & \cdots & 0 \\ 0 & \cdots & 0 & \mu_1 \\ 0 & \cdots & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & 0 \end{bmatrix}, \cdots, M_1 = \begin{bmatrix} 0 & \cdots & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \mu_1 \\ 0 & \cdots & \cdots & 0 \end{bmatrix}$$

From Neuts' mean drift criterion [11] we can obtain the stability criterion in (6). Similar to the single service model we now define π to be the equilibrium distribution of a Markov process with generator $\Lambda + D + \sum_{k=1}^{K} M_k$:

$$\underline{\pi}(\Lambda + D + \sum_{k=1}^{K} M_k) = \underline{0} \text{ where } \underline{\pi}\mathbf{e} = \underline{1}.$$

The process with generator Q is stable if and only if the drift condition $\underline{\pi} \sum_{k=1}^{K} M_k \mathbf{e} > \underline{\pi} \Lambda \mathbf{e}$ is satisfied.

Again, following [11], the stationary distribution has a matrix-geometric structure $\underline{\pi}_i = \underline{\pi}_0 R^i$, for $i = 1, 2, \ldots,$ where the matrix R is the minimal non-negative solution of

$$\Lambda + RD + \sum_{k=1}^{K} R^{k+1} \cdot M_k = 0.$$

The boundary equations now read

$$\underline{\pi}_0 \sum_{k=0}^{K} R^k \cdot B_k = \underline{0},$$

and the normalization condition is

$$\underline{\pi}_0 \sum_{k=0}^{\infty} R^k \cdot \mathbf{e} = \underline{\pi}_0 \cdot (I - R)^{-1} \cdot \mathbf{e} = 1.$$

Again, by computing the stationary distributions for varying values of the threshold K, we may determine the best value of the threshold in terms of the average cost as we did for the single service model using (4). Finally we can compare this result with the optimal MDP policy.

6 Simulation experiments

In this section we illustrate the effectiveness of the threshold policies, obtained using the matrix geometric representation, with the optimal strategies from the MDP formulation. For our comparison, we will simulate both classes of strategies, although for the threshold strategies the reported results can also be directly obtained after computing the stationary distribution.

In Figure 8 the costs and the average queue lengths are plotted for varying service rate at station 1. We observe that the performance of the best threshold policy is almost identical to that of the optimal MDP policy. The right hand graph also shows that the two policies are very close to each other in terms of the average queue lengths. The discontinuities in the curves corresponding to the threshold policies correspond to parameter choices where the optimal threshold value shifts by one. As can be expected, the discontinuities for the MDP policies are much less pronounced, as the optimal switching curve may shift only for a small number of states.

For the batch service model, the simulation results of the MDP and the threshold strategies are reported in Figure 9. We may now take μ_1 to be smaller than λ without compromising the stability of the optimal policy as long as condition (6) is satisfied.

We again observe a remarkable fit in terms of cost, for almost all values of the service rate at the first station. As could be expected, the costs are lower compared to the single service model. As for the queue lengths, we observe that the batch service mode allows to keep the first station at lower levels, but the queue lengths at the second station remain at roughly the same level. For now we defer further comparisons between the models with and without batch services and focus on comparisons between the MDP and the threshold strategies. Due to the more aggressive service mode, the costs of the threshold policies are much less smooth than in Figure 8 and the optimal mean queue lengths oscillate more for larger values of μ_1. Indeed, changing the threshold value by one has a much larger impact on the resulting policy (that aims to bring the queue length back to the horizontal switching curve in a single service run). The strong fluctuations in queue lengths, make it all the more surprising that the costs of the best threshold policy remain close to those of the optimal MDP strategy.

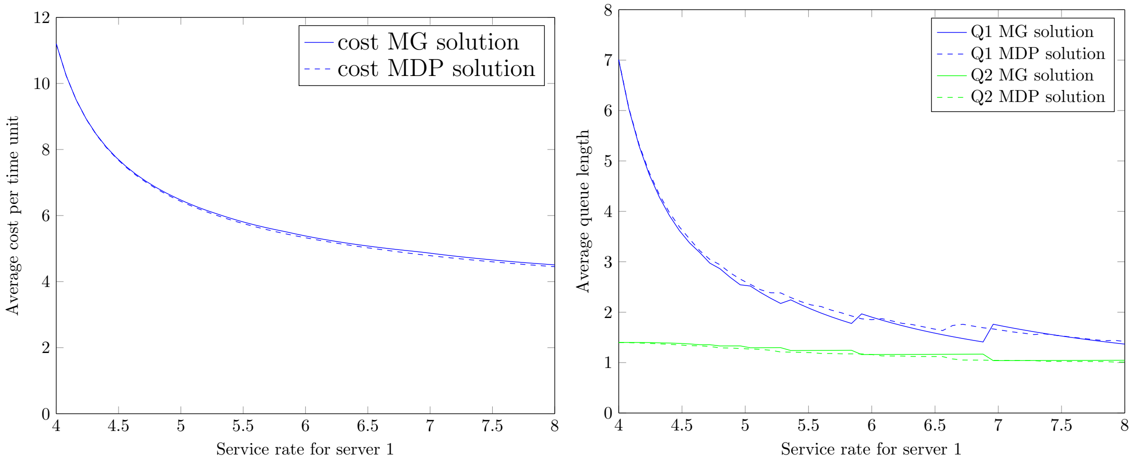

Figure 8: *Comparison of the optimal MDP policy model and the threshold-based approximation for the single service model for varying service rate μ_1. The parameters are: $\lambda = 3, \mu_1 \in [4, 8], \mu_2 = 6$.*

	$\lambda = 4, \mu_1 = 5, \mu_2 = 6$		$\lambda = 4, \mu_1 = 7, \mu_2 = 6$	
K	Single service	Batch transition	Single service	Batch transition
3	-	18.39	13.04	7.34
4	50.17	7.16	7.20	6.00
5	14.01	6.87	6.75	6.19
6	11.23	7.06	6.78	6.47
7	10.42	7.29	6.90	6.70
8	10.12	7.47	7.00	6.87
9	10.00	7.61	7.10	6.99
10	9.96	7.71	7.17	7.07
11	9.95	7.78	7.22	7.12
12	9.96	7.83	7.25	7.16

Table 1: *Comparing the costs of single and batch services for various threshold levels.*

We have now compared the rightfulness of the approximating threshold strategies. Next we compare the gain of having batch service in the first station. In Figure 10 the best threshold values are determined, again for increasing service rate at station 1. The single service model is not stable for $\lambda \geq \mu_1$. We observe that the threshold strategies only perform badly in the single service model when the system approaches instability. For a large range of values with $\mu_1 < \mu_2$, the single-service threshold strategy performs almost as well as the batch-service threshold strategy, although in that case the optimal switching curve for the single-service model has a rather steep (linear) ascent. It is quite surprising that the costs are comparable for the two models, as long as μ_1 does not approach the stability limit (i.e., remains $> \lambda$). The optimal threshold levels do, naturally, differ: that of the single service model is considerably larger than in the batch service model (as could be expected).

Table 1 displays the costs for the two models for various threshold levels. The batch transition model performs better for all threshold levels, also non-optimal levels, but the difference is not very pronounced. The main advantage of the batch-service model is that the costs are not that sensitive to the exact value of the threshold. For the single-service model the costs are much more sensitive and small errors in the threshold value may lead to considerable loss of efficiency.

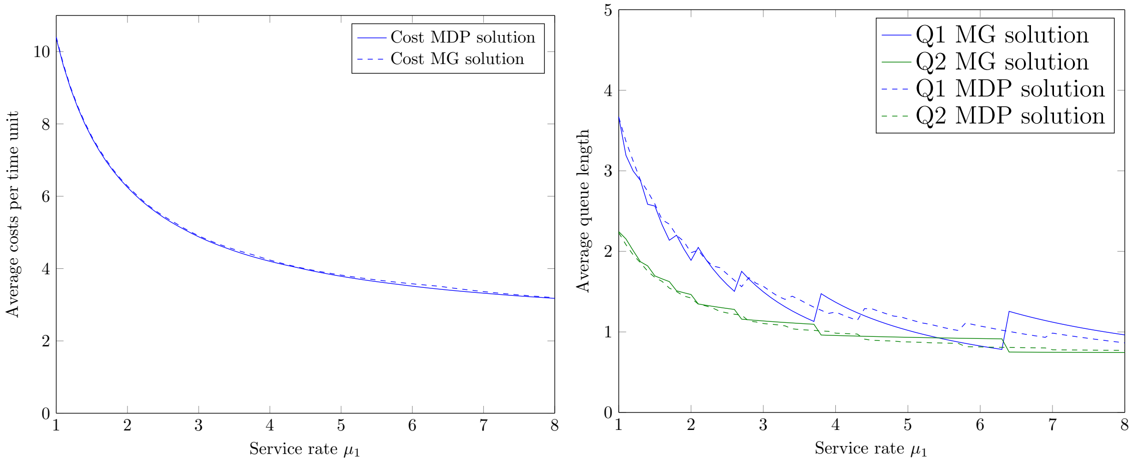

Figure 9: *Comparison of the optimal MDP strategy and the threshold-based approximation for the batch service model for varying service rate* μ_1. *The parameters are:* $\lambda = 3, \mu_1 \in [1, 8], \mu_2 = 6$.

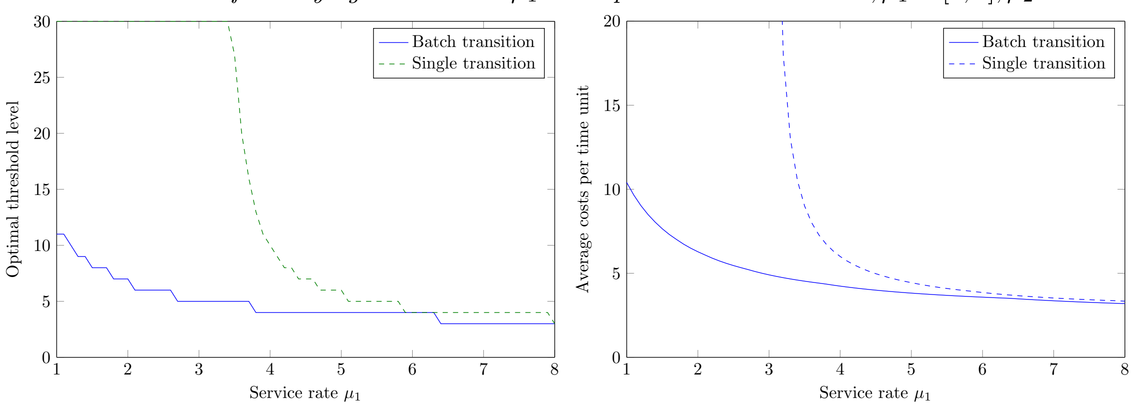

Figure 10: *Comparison of optimal threshold strategies for the single service and batch service model;* $\lambda = 3, \mu_1 \in [1, 8], \mu_2 = 6$.

7 Conclusion

We have investigated the optimal control of the number of jobs in an expensive service station, by regulating the flow from a preceding buffer station. We started by determining the optimal control policy using a Markov Decision Problem formulation. The optimal strategy can in general be characterized by a switching curve; the shape of this curve is determined by whether or not the first station has a larger service rate than the second. If so, then the optimal switching curve is rather flat, otherwise it increases approximately linearly. When the optimal switching curve is flat, it can well be approximated by a horizontal one, which corresponds to a fixed threshold strategy. Besides their practical relevance due to the simplicity of implementation, threshold-based strategies have the advantage that they allow a much more detailed analysis. By casting a threshold based control into the framework of Markov models with a matrix geometric stationary distribution, we can efficiently compute the best threshold level. For this "optimal" threshold level, we indeed verified that it performs very closely to the optimal MDP strategy.

For some of our motivating examples, the "feeding" process from the buffer station need not necessarily be by individual jobs only. It is quite natural to allow multiple jobs to be served in a single service run from the first station. For this model we again formulated and studied the corresponding MDP and established that the optimal switching curve always has a flat shape, irrespective of the speeds of the servers. Again, threshold based strategies were shown to be much more efficiently solvable and have close to optimal performance.

Surprisingly, the best single-service threshold and the best batch-service threshold policies were found to give comparable performance, unless the single-service threshold policies were near their stability limit (the arrival rate being near the service rate of the first station). In the latter case, batch-service threshold strategies profit from their larger stability region.

Our results also translate into practical design rules. First of all, the simplicity of threshold based rules make them much easier to implement in practical scenarios. Note that the optimal switching curve policy requires to operate a different threshold level depending on the load in the first station.

A further insight is that for single service mode at the first station, it is important that the service rate in that station is large enough (preferably larger than, or at least comparable to that at the second station). When applied to the distribution center setting, this implies that the parking facility should not be too far from the distribution center; in fact, the travel time between the two should be smaller than, or comparable to, the unloading time at the distribution center. If multiple jobs can be simultaneously transferred between the two stations, the distance is not a major issue. In that case, performance is rather insensitive to the service speed in the first station (unless that speed is very low).

References

[1] F. Avram, *Optimal Control of Fluid Limits of Queueing Networks and Stochasticity Corrections.* Lectures in Applied Mathematics, vol 33, p. 1-36, 1997.

[2] R.E. Bellman, *Dynamic Programming.* Princeton University Press, Princeton, NJ, Republished 2003: Dover.

[3] G.L. Curry, R.M. Feldman *An M/M/1 queue with a general bulk service rule.* Naval research logistics quarterly, vol 32.4 p.595-603, 1985.

[4] A. El-Rayes, M. Kwiatkowska, and G. Norman, *Solving infinite stochastic process algebra models through matrix-geometric methods.* School of computer science research reports, university of Birmingham, 1999.

[5] A. Gajrat, A. Hordijk, A. Ridder, *Large-deviations analysis of the fluid approximation for a controllable tandem queue.* Annals of Applied Probability p.1423-1448, 2003.

[6] G. Koole, *Convexity in tandem queues.* Probability in the Engineering and Informational Sciences 18.01: 13-31.

[7] G. Latouche, M. Neuts, *Efficient algorithmic solutions to exponential tandem queues with blocking.* SIAM Journal on Algebraic Discrete Methods, vol 1, p. 93-106,1980.

[8] G. Latouche, V. Ramaswami, *A logarithmic reduction algorithm for quasi-birth-and-death processes.* J. Appl. Probab., 30:650–674, 1993.

[9] S.A. Lippman, *Applying a new device in the optimisation of exponential queueing systems.* Operations research, 1975.

[10] S.P. Meyn, *Control techniques for complex networks.* Cambridge ISBN 978-0-521-88441-9 hardback, 2008.

[11] M. Neuts, *Matrix-geometric Solutions in Stochastic Models: An Algorithmic Approach.* Courier Dover Publications, 1981.

[12] M. Neuts, *A general class of bulk queues with Poisson input.* The Annals of Mathematical Statistics p.759-770, 1967.

[13] M.L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming.* John Wiley & Sons, Inc. 1994.

[14] Z. Rosberg, P.P. Varaiya, J. Walrand, *Optimal control of service in tandem queues.* Automatic Control, IEEE Transactions in vol.27 Issue: 3, 1982.

[15] R.R. Weber, S. Stidham, *Optimal control of service rates in networks of queues.* Advances in Applied Probability, vol. 19 p.202-218, 1987.

[16] J. Weiss, *The computation of optimal control limits for a queue with batch services.* Management Science 25.4 p.320-328, 1979.