Systems Engineering Group
Department of Mechanical Engineering
Eindhoven University of Technology
PO Box 513
5600 MB Eindhoven
The Netherlands
http://se.wtb.tue.nl/

SE Report: Nr. 2008-09

Optimal control of a deterministic multiclass queuing system by serving several queues simultaneously

Erjen Lefeber, Stefan Lämmer, and J.E. Rooda

ISSN: 1872-1567

SE Report: Nr. 2008-09
Eindhoven, September 2008
SE Reports are available via http://se.wtb.tue.nl/sereports

Abstract

We consider the optimal control problem of emptying a deterministic single server multiclass queuing system without arrivals. We assume that the server is able to serve several queues simultaneously, each at its own rate, independent of the number of queues being served.

We show that the optimal sequence of modes is ordered by rate of cost decrease. However, queues are not necessarily emptied. We propose a dynamic programming approach for solving the problem, which reduces the multiparametric QP (mpQP) to a series of problems that can be solved readily.

1 Introduction

Consider a model of N queues competing for a single server. The buffer capacity at each queue is unlimited. The server is able to serve queue i at a rate μ_i. The cost of operation per unit time is a linear function of the queue sizes. For this system it is well-known [1, 2, 4, 6] that the optimal policy is a μc-rule: allocate service attention to the non-empty queue with the largest rate of cost decrease.

The above mentioned papers assume that the server can serve only one queue simultaneously. In this report we assume that the server is able to serve several queues simultaneously, each queue at rate μ_i, independent of the number of queues being served. These kind of models arise when studying multiclass queueing networks. In Figure 1 we depicted three illustrative

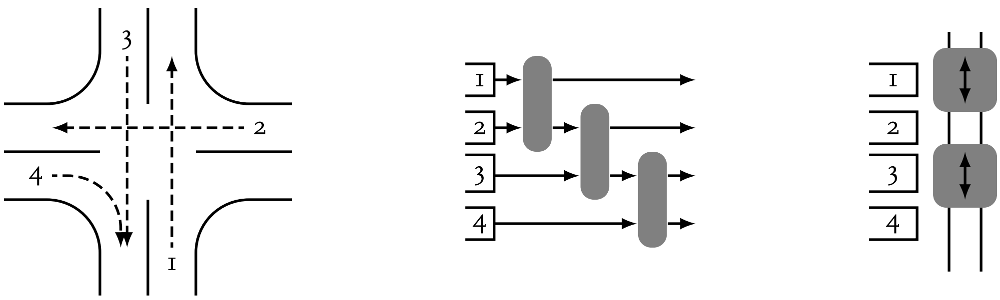

Figure 1: Three examples of multiclass queueing networks which can be modeled as a single server that can serve several queues simultaneously

examples, which all can be modeled similarly.

The first example is an intersection which needs to switch between flows from four different directions. For this intersection the directions 1 and 2 can not be served simultaneously. The same holds for directions 2 and 3, and also for the directions 3 and 4. However, the directions 1 and 3 can be served simultaneously. The same holds for directions 1 and 4, and also for the directions 2 and 4.

The second example is a multiclass queueing tandem network consisting of three servers, where each server serves two classes, but can serve only one class at the same time. There are no buffers between the servers. Class 1 needs only service at the 1st server, class 2 needs service at server 1, followed by service at server 2. Class 3 needs service at server 2, followed by service at server 3, and class 4 needs only service at the 3rd server. This model can also be used as an approximation for cases where buffers between servers are negligibly small.

The third example is a two-server polling system with physical constraints: the servers can not serve to consecutive queues and can not overtake.

All three examples can be modeled as a single server with the modes

mode $\{1, 3\}$: serve class 1 and class 3 simultaneously,

mode $\{1, 4\}$: serve class 1 and class 4 simultaneously,

mode $\{2, 4\}$: serve class 2 and class 4 simultaneously,

Table 1: An overview of the rate of cost decrease per mode for the example with service rate $\mu_i = 1$, $c_1 = 4$, $c_2 = 3$, $c_3 = 2$, and $c_4 = 5$

mode	rate of cost decrease
mode $\{1, 4\}$	9
mode $\{2, 4\}$	8
mode $\{1, 3\}$	6
mode $\{4\}$	5
mode $\{1\}$	4
mode $\{2\}$	3
mode $\{3\}$	2
mode $\emptyset$	0

and the additional modes

mode $\{1\}$: serve only class 1,

mode $\{2\}$: serve only class 2,

mode $\{3\}$: serve only class 3,

mode $\{4\}$: serve only class 4,

mode $\emptyset$: idle,

since it is also possible to serve a subset of classes.

In this report we consider the optimal control of this multiclass queueing system when the cost of operation per unit time is a linear function of the queue sizes. Throughout, we consider a fluid model with negligible setup times.

As an illustrative example, consider the above mentioned system and assume no arrivals. Furthermore, assume that for each class the service rate $\mu_i = 1$. Let $x_i(t)$ denote the queue size of class i at time t, and assume that the cost of operation per unit time is given by $c_1x_1 + c_2x_2 + c_3x_3 + c_4x_4$ with $c_1 = 4$, $c_2 = 3$, $c_3 = 2$, and $c_4 = 5$. So the problem we consider is to minimize

$$\int_0^\infty 4x_1(t) + 3x_2(t) + 2x_3(t) + 5x_4(t)\,\mathrm{d}\,t. \tag{1}$$

Assume that the system initially starts at $(x_1, x_2, x_3, x_4) = (6, 6, 6, 6)$.

We can make an overview of the rate of cost decrease per mode, as shown in Table 1. According to the μc-rule, a good policy seems to be to first use mode $\{1, 4\}$ for a duration of 6 time units, bringing the system in $(x_1, x_2, x_3, x_4) = (0, 6, 6, 0)$, followed by mode $\{2\}$ for a duration of 6 time units, bringing the system in $(x_1, x_2, x_3, x_4) = (0, 0, 6, 0)$. Finally, use mode $\{3\}$ for a duration of 6 time units to empty the system, after which the system can idle. If we substitute the resulting trajectories for the queue lengths in (1), the total costs for this policy become $4 \cdot 18 + 3 \cdot 54 + 2 \cdot 90 + 5 \cdot 18 = 504$.

An alternative policy would be to first use mode $\{2, 4\}$ for a duration of 6 time units, bringing the system in $(x_1, x_2, x_3, x_4) = (6, 0, 6, 0)$. Next, serve in mode $\{1, 3\}$ for 6 time units to empty the system, and then idle. If we substitute the resulting trajectories for the queue lengths in (1), the total costs for the alternative policy become $4 \cdot 54 + 3 \cdot 18 + 2 \cdot 54 + 5 \cdot 18 = 468$, which is less.

Clearly the μc-rule does not hold for this system. Furthermore, the alternative policy is not optimal either. As we show in the remainder of this report, the optimal policy in this case yields a total costs of 456.

2 The problem

We consider N queues competing for a single server which can serve several queues simultaneously.

Assumption 2.1. We assume that no new jobs arrive to this system.

To model classes that can not be served simultaneously, let $S = (\mathcal{N}, \mathcal{C})$ be an undirected graph, with vertices $\mathcal{N} = \{1, 2, \dots, N\}$ corresponding with the classes, and edges $\mathcal{C} \subset \mathcal{N} \times \mathcal{N}$ corresponding to conflicting classes. That is, a pair $(i, j) \in \mathcal{C}$ $(i < j)$ when classes i and j can not be served simultaneously. For the example in the previous section we have

$$\mathcal{N} = \{1, 2, 3, 4\} \qquad \text{and} \qquad \mathcal{C} = \{(1, 2), (2, 3), (3, 4)\}, \tag{2}$$

see also Figure 2.

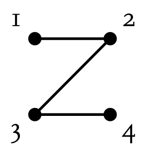

Figure 2: The graph $S = (\mathcal{N}, \mathcal{C})$ with $\mathcal{N}$ and $\mathcal{C}$ as in (2)

Definition 2.2. We call a set $m \subset \mathcal{N}$ an *allowed mode* when $m \times m \cap \mathcal{C} = \emptyset$. That is, all of the classes in m can be served simultaneously.

Corollary 2.3. *When a set $m \subset \mathcal{N}$ is an allowed mode, any subset of m is also an allowed mode.*

Let $\mathcal{M}_S$ denote the set of all allowed modes for the multiclass single server system described by the graph S. Furthermore, let $x(t) = [x_1(t), x_2(t), \dots, x_N(t)]^T$ denote the queue lengths at time t.

The system dynamics is given by the hybrid fluid model:

$$\dot{x}(t) = -B_m u(t) \qquad m \in \mathcal{M}_S, \tag{3}$$

where

$$B_m = \begin{bmatrix} \mathbb{I}_m(1) & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \mathbb{I}_m(N) \end{bmatrix} \qquad \mathbb{I}_m(i) = \begin{cases} 1 & \text{if } i \in m \\ 0 & \text{if } i \notin m, \end{cases}$$

and $u(t) = [u_1(t), u_2(t), \dots, u_N(t)]^T$ denotes the vector of used service rates at time t.

The system dynamics is subject to the constraints

$$x_i(t) \geq 0 \qquad\qquad 0 \leq u_i(t) \leq \mu_i \qquad\qquad \forall i \in \mathcal{N}, \forall t \geq 0. \tag{4}$$

Let $c = [c_1, c_2, \dots, c_N]^T$ be a costs vector satisfying $c_i > 0$.

Problem 2.4. Find a feedback $u(x)$, $m(x)$ for the system (3) which guarantees (4) and minimizes

$$J(x_0) = \int_0^\infty c^T x(s; u, m, x_0)\, \mathrm{d}\, s, \tag{5}$$

where $x(t; u, m, x_0)$ denotes the resulting queue lengths at time t when using feedback $u(x)$ and $m(x)$ if the system starts in $x(0) = x_0$ at time 0.

Lemma 2.5. *For an optimal policy, the rate of service of class $i \in \mathcal{N}$ is given by $u_i(x) = \mu_i$.*

Proof. We prove the result by contradiction. Suppose that an optimal policy is given for which class $i \in \mathcal{N}$ is served from t_0 to t_f in consecutive modes $m_1, m_2, \dots, m_n$ with $i \in m_j$ for $j = 1, 2, \dots, n$. But assume that during this interval $u_i(t) < \mu_i$. Let x_i^0 and x_i^f denote the queue length for class i at t_0 and t_f respectively. Consider an alternative policy which mimics this optimal policy, but first serves class i at rate μ_i for a duration of $(x_i^0 - x_i^f)/\mu_i$, after which it serves class i at rate 0 for the remaining duration of $(t_f - t_0) - (x_i^0 - x_i^f)/\mu_i$. Notice that the alternative policy is feasible, since we have no arrivals. Clearly, the queue length of class i can not decrease at a faster rate than in this alternative policy. Therefore, at each time instant the queue length of class i is strictly less than for the optimal policy, whereas the queue length of all other classes remains the same. In particular this implies that the alternative policy results in strictly lower total costs, which contradicts the optimality of the given optimal policy. $\square$

Lemma 2.6. *For an optimal policy the value of $\sum_{i \in m_j} \mu_i c_i$ is non-increasing for two consecutive modes m_j.*

Proof. We prove the result by contradiction. Suppose that an optimal policy is given with two consecutive modes m_1 and m_2 for which $\sum_{i \in m_1} \mu_i c_i < \sum_{i \in m_2} \mu_i c_i$. Let τ_{m_1} and τ_{m_2} denote the corresponding durations of these modes. Consider the alternative policy where this sequence of modes is interchanged, while keeping the durations of the modes the same. That is, in the alternative policy first mode m_2 is used for a duration of τ_{m_2}, after which mode m_1 is used for a duration of τ_{m_1}. Notice that the alternative policy is feasible, since we have no arrivals. Clearly, the costs initially decrease at a faster rate for the alternative policy, resulting in strictly lower total costs, which contradicts the optimality of the given optimal policy. $\square$

Remark 2.7. Notice that Lemma 2.6 does not contradict our observation that the μc-rule does not hold for the example in the previous section. Apparently the sequence of modes during transient is in accordance with their μc-values, but the duration of modes is not determined by buffers becoming empty.

Remark 2.8. Notice that we have not yet addressed the possibility of switching infinitely fast between several modes. This is something we in principle could do, as setup times are assumed to be zero. By assuming that at time t we are in mode m for a fraction of time $\alpha_m(t) \geq 0$, with $\sum_{m \in \mathcal{M}_S} \alpha_m(t) = 1$, instead of the dynamics (3) we could consider the dynamics

$$\dot{x}(t) = -\sum_{m \in \mathcal{M}_S} \alpha_m(t) B_m u(t) \qquad\qquad m \in \mathcal{M}_S.$$

In a similar way as the proofs of lemmas 2.5 and 2.6 it can be shown be means of contradiction that $\alpha_m(t) \in \{0, 1\}$.

Suppose that an optimal policy is given which does not satisfy this property on the interval $[t_1, t_2]$. For each mode $m \in \mathcal{M}_S$ we define $\tau_m = \int_{t_1}^{t_2} \alpha_m(s)\, \mathrm{d}\, s$, and additionally for each class $i \in \mathcal{N}$ we define $\tau_m^i = \frac{1}{\mu_i} \int_{t_1}^{t_2} \alpha_m(s) u_i(s)\, \mathrm{d}\, s$. Consider an alternative policy which is successively in each mode m for a duration of τ_m, where the modes are in the order such that $\mu_m c_m$ is non-increasing for two consecutive modes. During mode m, class i is first served at rate μ_i for a duration of τ_m^i, after which it is served at rate 0 for a duration of $\tau_m - \tau_m^i$. This alternative is not only feasible, but also improves on the given optimal policy.

3 A worked out example

In the previous section we not only introduced the problem, but also derived two lemmas that are helpful in determining the optimal feedback. Before solving the general problem we first consider the example introduced in Section 1, i.e. the system depicted in Figure 1 which can be parameterized by means of (2), $\mu_i = 1$ for all $i \in \mathcal{N}$, and $c = [4, 3, 2, 5]^T$.

As a first step in solving the problem we first consider the open loop optimal control problem. Let an initial condition $x(0) = [x_{10}, x_{20}, x_{30}, x_{40}]^T$ be given. From Lemma 2.6 we know that the system subsequently visits the modes $\{1, 4\}$, $\{2, 4\}$, $\{1, 3\}$, $\{4\}$, $\{1\}$, $\{2\}$, and $\{3\}$, after which the system stays in mode $\emptyset$ forever. Let τ_{14}, τ_{24}, τ_{13}, τ_4, τ_1, τ_2, and τ_3 denote the durations of the successive modes. From Lemma 2.5 we know that during each mode, each class is served at maximal rate.

Using the results from lemmas 2.5 and 2.6 we can now determine the resulting costs as a function of these durations:

$$\begin{aligned}
\int_0^\infty x_1(s)\, \mathrm{d}\, s &= \frac{1}{2} x_{10}^2 + (x_{10} - \tau_{14})\tau_{24} + (x_{10} - \tau_{14} - \tau_{13})\tau_4 \\
\int_0^\infty x_2(s)\, \mathrm{d}\, s &= \frac{1}{2} x_{20}^2 + x_{20}\tau_{14} + (x_{20} - \tau_{24})(\tau_{13} + \tau_4 + \tau_1) \\
\int_0^\infty x_3(s)\, \mathrm{d}\, s &= \frac{1}{2} x_{30}^2 + x_{30}(\tau_{14} + \tau_{24}) + (x_{30} - \tau_{13})(\tau_4 + \tau_1 + \tau_2) \\
\int_0^\infty x_4(s)\, \mathrm{d}\, s &= \frac{1}{2} x_{40}^2 + (x_{40} - \tau_{14} - \tau_{24})\tau_{13}
\end{aligned}$$

where we also have

$$\begin{aligned}
x_{10} &= \tau_{14} + \tau_{13} + \tau_1 \\
x_{20} &= \tau_{24} + \tau_2 \\
x_{30} &= \tau_{13} + \tau_3 \\
x_{40} &= \tau_{14} + \tau_{24} + \tau_4
\end{aligned} \tag{6}$$

The problem of minimizing the costs (5) for a given initial condition x_0 can be reduced to solving the following quadratic program:

$$\min_{\tau \geq 0} \frac{1}{2} \tau^T H \tau - x_0^T F \tau + \frac{1}{2} x_0^T Y x_0 \tag{7a}$$

subject to

$$G\tau \leq x_0 \tag{7b}$$

where $\tau = [\tau_{14}, \tau_{24}, \tau_{13}]^T$ and

$$F = \begin{bmatrix} 4 & 3 & 2 \\ 3 & 3 & 2 \\ 2 & 2 & 2 \\ 4 & 3 & 1 \end{bmatrix} \qquad G = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \tag{7c}$$

$$H = \begin{bmatrix} 8 & 6 & 3 \\ 6 & 6 & 3 \\ 3 & 3 & 4 \end{bmatrix} \qquad Y = \begin{bmatrix} 4 & 3 & 2 & 4 \\ 3 & 3 & 2 & 3 \\ 2 & 2 & 2 & 2 \\ 4 & 3 & 2 & 5 \end{bmatrix}. \tag{7d}$$

For any given initial condition x_0, (7) is a QP. The quadratic program (7) is a so-called multi parametric quadratic program (mpQP) and can be solved for an arbitrary parameter x_0 [3, 7]. An mpQP solver is included in the (free) Multi-Parametric Toolbox for Matlab [5].

The solution of the mpQP (7) is given by:

$$\tau = \begin{cases}
\begin{bmatrix} \frac{1}{2} & -\frac{1}{3} & -\frac{1}{3} & \frac{1}{2} \\ -\frac{1}{2} & \frac{1}{3} & \frac{1}{3} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{3} & -\frac{1}{2} \end{bmatrix} x_0 & \text{for } \begin{bmatrix} -3 & 2 & 2 & -3 \\ 3 & -2 & -2 & -3 \\ -3 & -2 & -2 & 3 \\ -3 & -4 & 2 & 3 \\ 3 & 2 & -4 & -3 \end{bmatrix} x_0 \leq 0 \\
\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} x_0 & \text{for } \begin{bmatrix} 0 & -1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 3 & -2 & -2 & 3 \end{bmatrix} x_0 \leq 0 \\
\begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 \end{bmatrix} x_0 & \text{for } \begin{bmatrix} 1 & 0 & -1 & -1 \\ -3 & 2 & 2 & 3 \end{bmatrix} x_0 \leq 0 \\
\begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} x_0 & \text{for } \begin{bmatrix} -1 & -1 & 0 & 1 \\ 3 & 2 & 2 & -3 \end{bmatrix} x_0 \leq 0 \\
\begin{bmatrix} 0 & -1 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & -1 \end{bmatrix} x_0 & \text{for } \begin{bmatrix} 0 & 1 & 0 & -1 \\ -1 & -1 & 0 & 1 \\ 3 & 4 & -2 & -3 \end{bmatrix} x_0 \leq 0 \\
\begin{bmatrix} 1 & 0 & -1 & 0 \\ -1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} x_0 & \text{for } \begin{bmatrix} -1 & 0 & 1 & 0 \\ 1 & 0 & -1 & -1 \\ -3 & -2 & 4 & 3 \end{bmatrix} x_0 \leq 0 \\
\begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} x_0 & \text{for } \begin{bmatrix} -1 & 0 & 1 & 1 \end{bmatrix} x_0 \leq 0 \\
\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} x_0 & \text{for } \begin{bmatrix} 1 & 1 & 0 & -1 \end{bmatrix} x_0 \leq 0.
\end{cases} \tag{8}$$

So the parameter space for x_0 is divided into 8 regions, and for each region the duration of the first three modes is specified as a linear function of x_0. The duration of the other four modes follows from (6).

From this solution we can obtain the optimal controller for the example studied in Section 1, i.e. starting from the initial condition $x_0 = [6, 6, 6, 6]^T$. Notice that we are in the first region of (8). This gives that we should first use mode $\{1, 4\}$ for a duration of 2, bringing the system in $x = (4, 6, 6, 4)$. Next, use mode $\{2, 4\}$ for a duration of 4, bringing the system in $x = (4, 2, 6, 0)$. Subsequently, use mode $\{1, 3\}$ for a duration of 4, bringing the system in $x = (0, 2, 2, 0)$. Then, use mode $\{2\}$ for a duration of 2, bringing the system in $x = (0, 0, 2, 0)$. Finally, use mode $\{3\}$ for a duration of 2, bringing the system in $x = (0, 0, 0, 0)$, resulting in the mentioned total costs of 456.

4 A dynamic programming approach

The approach introduced in the previous section solves the example problem for a given cost vector $c = [c_1, c_2, c_3, c_4]^T$. However, by means of a dynamic programming approach it is possible to solve the problem for a given sequence of modes. Furthermore, a different formulation of the controller is obtained, which is also easier to implement. To illustrate this, we again consider the system depicted in Figure 1 which can be parameterized by means of (2), but this time we consider arbitrary $\mu_i > 0$ and $c_i > 0$. We only assume that the sequence of modes is as described in Table 1, i.e. we assume that $0 < \mu_3 c_3 \le \mu_2 c_2 < \mu_1 c_1 \le \mu_4 c_4 \le \mu_1 c_1 + \mu_3 c_3$.

In our dynamic programming approach we first solve the subproblem for the case where we only have the final five modes $\{4\}$, $\{1\}$, $\{2\}$, $\{3\}$, and $\emptyset$ available. The solution to this problem is given by the μc-rule. First serve class 4 exhaustively, then class 1, followed by class 2 and finally class 3. The resulting cost-to-go is given by

$$\frac{1}{2} x^T \begin{bmatrix} \frac{c_1}{\mu_1} & \frac{c_2}{\mu_1} & \frac{c_3}{\mu_1} & \frac{c_1}{\mu_4} \\ \frac{c_2}{\mu_1} & \frac{c_2}{\mu_2} & \frac{c_3}{\mu_2} & \frac{c_2}{\mu_4} \\ \frac{c_3}{\mu_1} & \frac{c_3}{\mu_2} & \frac{c_3}{\mu_3} & \frac{c_3}{\mu_4} \\ \frac{c_1}{\mu_4} & \frac{c_2}{\mu_4} & \frac{c_3}{\mu_4} & \frac{c_4}{\mu_4} \end{bmatrix} x. \tag{9}$$

For the next subproblem, we assume that we have the final six modes available. That is, in addition to the five modes we assume to have available in the previous subproblem, we assume to have mode $\{1, 3\}$ available as well. From Lemma 2.6 we know that we start in this mode, and from the previous subproblem we know how to proceed after leaving this mode. The only thing that we need to determine is how long to stay in mode $\{1, 3\}$. Assume that we stay in this mode for a duration of τ_{13}. The costs made during mode $\{1, 3\}$ are

$$c_1 \tau_{13} \left(x_1 - \frac{1}{2} \mu_1 \tau_{13} \right) + c_2 \tau_{13} x_2 + c_3 \tau_{13} \left(x_3 - \frac{1}{2} \mu_3 \tau_{13} \right) + c_4 \tau_{13} x_4. \tag{10}$$

The remaining cost to go is given by

$$\frac{1}{2} \begin{bmatrix} x_1 - \tau_{13}\mu_1 \\ x_2 \\ x_3 - \tau_{13}\mu_3 \\ x_4 \end{bmatrix}^T \begin{bmatrix} \frac{c_1}{\mu_1} & \frac{c_2}{\mu_1} & \frac{c_3}{\mu_1} & \frac{c_1}{\mu_4} \\ \frac{c_2}{\mu_1} & \frac{c_2}{\mu_2} & \frac{c_3}{\mu_2} & \frac{c_2}{\mu_4} \\ \frac{c_3}{\mu_1} & \frac{c_3}{\mu_2} & \frac{c_3}{\mu_3} & \frac{c_3}{\mu_4} \\ \frac{c_1}{\mu_4} & \frac{c_2}{\mu_4} & \frac{c_3}{\mu_4} & \frac{c_4}{\mu_4} \end{bmatrix} \begin{bmatrix} x_1 - \tau_{13}\mu_1 \\ x_2 \\ x_3 - \tau_{13}\mu_3 \\ x_4 \end{bmatrix}. \tag{11}$$

Adding (10) and (11) and subtracting (9) results in an additional cost to go of

$$\mu_3 c_3 \tau_{13} \left(\tau_{13} - \left[\frac{x_1}{\mu_1} + \frac{x_2}{\mu_2} + \frac{x_3}{\mu_3} + \frac{\mu_1 c_1 + \mu_3 c_3 - \mu_4 c_4}{\mu_3 c_3} \frac{x_4}{\mu_4} \right] \right). \tag{12}$$

which needs to be minimized over τ_{13} subject to the constraint $0 \le \tau_{13} \le \min(x_1/\mu_1, x_3/\mu_3)$.

The minimum of (12) as a function of τ_{13} is achieved for

$$\tau_{13}^* = \frac{1}{2}\left(\frac{x_1}{\mu_1} + \frac{x_2}{\mu_2} + \frac{x_3}{\mu_3} + \frac{(\mu_1 c_1 + \mu_3 c_3) - \mu_4 c_4}{\mu_3 c_3}\frac{x_4}{\mu_4}\right)$$
$$\geq \frac{1}{2}\left(\frac{x_1}{\mu_1} + \frac{x_3}{\mu_3}\right) \geq \min\left(\frac{x_1}{\mu_1}, \frac{x_3}{\mu_3}\right).$$

Therefore, the end of mode $\{1,3\}$ is determined by either buffer 1 or buffer 3 becoming empty.

Similarly we can analyze the next subproblem, in which we assume that in addition to the final six modes we have mode $\{2,4\}$ available too. Let τ_{24} denote the duration of this mode. For the additional costs we get for $\frac{x_1}{\mu_1} \geq \frac{x_3}{\mu_3}$

$$\mu_2 c_2 \tau_{24}\left(\tau_{24} - \left[\frac{x_2}{\mu_2} + \frac{x_4}{\mu_4} + \frac{\mu_2 c_2 + \mu_4 c_4 - \mu_1 c_1 - \mu_3 c_3}{\mu_2 c_2}\frac{x_3}{\mu_3} + \left(\frac{x_1}{\mu_1} - \frac{x_3}{\mu_3}\right)\right]\right),$$

whereas for $\frac{x_2}{\mu_2} \leq \frac{x_4}{\mu_4}$ we obtain:

$$\mu_2 c_2 \tau_{24}\left(\tau_{24} - \left[\frac{x_2}{\mu_2} + \frac{x_4}{\mu_4} + \frac{\mu_2 c_2 + \mu_4 c_4 - \mu_1 c_1 - \mu_3 c_3}{\mu_2 c_2}\frac{x_1}{\mu_1} + \frac{\mu_3 c_3}{\mu_2 c_2}\left(\frac{x_3}{\mu_3} - \frac{x_1}{\mu_1}\right)\right]\right).$$

For both expressions the minimum as a function of τ_{24} is achieved for $\tau_{24}^* \geq \min(x_2/\mu_2, x_4/\mu_4)$, which implies that mode $\{2,4\}$ is finished by either $x_2 = 0$ or $x_4 = 0$.

The final step in our dynamic programming approach is to consider the full problem, i.e. assume that all allowed modes are available. We need to determine the duration of mode $\{1,4\}$: τ_{14}. When either $\frac{x_4}{\mu_4} \geq \frac{x_2}{\mu_2}$ or $\frac{x_1}{\mu_1} \geq \frac{x_3}{\mu_3}$ we obtain $\tau_{14}^* \geq \min(x_1/\mu_1, x_4/\mu_4)$. However, for $\frac{x_4}{\mu_4} \leq \frac{x_2}{\mu_2}$ and $\frac{x_1}{\mu_1} \leq \frac{x_3}{\mu_3}$ we obtain

$$\tau_{14}^* = \frac{1}{2}\left(\frac{x_1}{\mu_1} + \frac{x_4}{\mu_4}\right) - \frac{\mu_3 c_3}{2(\mu_1 c_1 - \mu_2 c_2 + \mu_3 c_3)}\left(\frac{x_2}{\mu_2} + \frac{x_3}{\mu_3}\right). \tag{13}$$

This implies that mode $\{1,4\}$ is either terminated when $x_1 = 0$ or $x_4 = 0$, or when all of the following three conditions are satisfied:

- $\frac{x_4}{\mu_4} \leq \frac{x_2}{\mu_2}$,
- $\frac{x_1}{\mu_1} \leq \frac{x_1}{\mu_3}$, and
- $(\mu_1 c_1 - \mu_2 c_2 + \mu_3 c_3)\left(\frac{x_1}{\mu_1} + \frac{x_4}{\mu_4}\right) \leq \mu_3 c_3\left(\frac{x_2}{\mu_2} + \frac{x_3}{\mu_3}\right)$.

To summarize, from the dynamic programming approach we obtain the following controller for the system depicted in Figure 1, parameterized by means of (2), with $0 < \mu_3 c_3 \leq \mu_2 c_2 < \mu_1 c_1 \leq \mu_4 c_4 \leq \mu_1 c_1 + \mu_3 c_3$:

Initialization: Start in mode $\{1,4\}$.

mode $\{1,4\}$: Stay in this mode until either $x_1 = 0$, or $x_4 = 0$, or $x_4 \leq x_2 \wedge x_1 \leq x_3 \wedge (\mu_1 c_1 - \mu_2 c_2 + \mu_3 c_3)\left(\frac{x_1}{\mu_1} + \frac{x_4}{\mu_4}\right) \leq \mu_3 c_3\left(\frac{x_2}{\mu_2} + \frac{x_3}{\mu_3}\right)$ then switch to mode $\{2,4\}$.

mode $\{2,4\}$: Stay in this mode until either $x_2 = 0$ or $x_4 = 0$, then switch to mode $\{1,3\}$.

mode $\{1, 3\}$: Stay in this mode until either $x_1 = 0$ or $x_3 = 0$, then switch to mode $\{4\}$.

mode $\{4\}$: Stay in this mode until $x_4 = 0$, then switch to mode $\{1\}$.

mode $\{1\}$: Stay in this mode until $x_1 = 0$, then switch to mode $\{2\}$.

mode $\{2\}$: Stay in this mode until $x_2 = 0$, then switch to mode $\{3\}$.

mode $\{3\}$: Stay in this mode until $x_3 = 0$.

Given an arbitrary initial condition x_0, the duration of each mode can be derived from the above description. Doing so for the case where $c_1 = 4$, $c_2 = 3$, $c_3 = 2$, $c_4 = 5$, and $\mu_i = 1$ results in (8).

5 The dynamic programming approach for the general problem

In the previous section we introduced a dynamic programming approach to solving the problem for a specific example. In this section we deal with the dynamic programming approach for Problem 2.4 as introduced in Section 2.

Consider the set of allowed modes

$$\mathcal{M}_S = \{m_1, m_2, \dots, m_M\}$$

and assume without loss of generality that

$$\sum_{i \in m_j} \mu_i c_i \geq \sum_{i \in m_k} \mu_i c_i \qquad \forall j < k.$$

From Lemmas 2.5 and 2.6, we know that classes are served at maximal rate, and subsequent modes are ordered by the rate of cost decrease. That is, the system visits first mode m_1, then m_2, etc. and finally the system visits mode $m_M = \emptyset$.

Remark 5.1. Notice that not necessarily the modes in which a single class is served are the modes $m_{M-N}, m_{M-N+1}, \dots, m_{M-1}$. For example, in the system parameterized by (2), we might have $c_4 > c_1 + c_3$.

Our dynamic programming approach consists of solving a sequence of subproblems.

The i^{th} subproblem P_i can be formulated as follows:

Problem 5.2 (subproblem P_i). Consider system dynamics described by the hybrid fluid model

$$\dot{x}(t) = -B_m \mu \qquad\qquad m \in \{m_{M-i+1}, m_{M-i+2}, \dots, m_M\} \tag{14}$$

where

$$B_m = \begin{bmatrix} \mathbb{I}_m(1) & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \mathbb{I}_m(N) \end{bmatrix} \qquad\qquad \mathbb{I}_m(i) = \begin{cases} 1 & \text{if } i \in m \\ 0 & \text{if } i \notin m, \end{cases}$$

and $\mu(t) = [\mu_1, \mu_2, \dots, \mu_N]^T$ denotes the vector of service rates.

Find a feedback $m(x)$ which guarantees

$$\begin{aligned} x_i(t) &\geq 0 \quad \text{for all } i \in \bigcup_{j=M-i+1}^{M} m_j, \forall t \geq 0 \\ x_i(t) &= 0 \quad \text{for all } i \in \mathcal{N} \setminus \bigcup_{j=M-i+1}^{M} m_j, \forall t \geq 0 \end{aligned} \tag{15}$$

and minimizes

$$J(x_0) = \int_0^\infty c^T x(s; u, m, x_0)\, \mathrm{d}\, s, \tag{16}$$

where $x(t; u, m, x_0)$ denotes the resulting queue lengths at time t when using feedback $m(x)$ if the system starts in $x(0) = x_0$ at time 0, where x_0 satisfies (15).

The solution of subproblem P_1 is trivial.

Let the solution of subproblem P_i be given, consisting not only of the feedback $m(x)$, but also of the cost to go $J(x)$. From Lemma 2.6 we know that the solution to subproblem P_{i+1} follows from first staying in mode m_{M-i} for a duration τ_{M-i}, after which the solution of subproblem P_i can be applied. Therefore, in order to solve subproblem P_{i+1}, only the duration $\tau_{M-i} \geq 0$ needs to be determined. This duration follows from minimizing a second order polynomial in τ_{M-i} subject to an upperbound on τ_{M-i} due to the fact that buffers are not allowed to become negative during mode m_{M-i}.

In this way, starting from the solution of subproblem P_1, we can consecutively solve the subproblems P_2, P_3, …, P_{M-1}, and finally also subproblem P_M, which actually is equivalent to Problem 2.4 which we need to solve.

6 Conclusions and future work

In this report we considered the optimal control problem of emptying a deterministic single server multiclass queuing system without arrivals. We considered that case where the server is able to serve several queues simultaneously, where queue i can be served at a rate μ_i. The cost of operation per unit time is a linear function of the queue sizes.

We showed that the optimal sequence of modes is ordered by rate of cost decrease. However, contrary to the μc-rule, queues are not necessarily emptied. Let M denote the number of modes. We proposed a dynamic programming approach for solving the problem, which reduces the M-dimensional multiparametric QP (mpQP) to a series of M problems that can be solved readily.

So far, we considered a system without arrivals. We are currently working on extending our solution to constant arrival rates. Lemmas 2.5 and 2.6 can be extended easily. The main difference is that the server can serve class i not only at rate 0 or μ_i, but also at rate λ_i (only when buffer i is empty). When considering infinitely fast switching between modes, the problem can also be formulated as a separated continuous linear problem for which a solution method has recently been presented in [8].

The next step to pursue is an extension to the stochastic setting, cf. [1, 2, 4, 6]. Lemmas 2.5 and 2.6 can also be extended to the setting of stochastic inter arrival times and stochastic service times. Subsequently, the dynamic programming approach can be extended.

The above mentioned extensions are relatively straightforward. A more difficult extension is to include non-zero setup times, since Lemma 2.6 does not hold anymore. This can be illustrated by means of the system depicted in Figure 1, parameterized by means of (2). Let the service rates $\mu_i = 1$, the cost vector $c = (0.34, 0.33, 0.32, 0.35)^T$, and the initial state $x_0 = (30, 20, 20, 40)$. In addition, assume that setup times are not negligible anymore, and that they are all equal to 1.

Based on the foregoing, one might assume it is optimal to first serve the system in mode $\{1, 4\}$, then in mode $\{2, 4\}$, next in mode $\{1, 3\}$, and finally in mode $\{3\}$, since that happens for negligible setup times. Notice that during the setup from mode $\{1, 4\}$ to mode $\{2, 4\}$, the system can keep on serving class 4, so in between these two modes the system can operate in mode $\{4\}$. During the setup from mode $\{2, 4\}$ to mode $\{1, 3\}$ no class can be served, i.e. the system is in mode $\emptyset$. Finally, the setup from mode $\{1, 3\}$ to mode $\{3\}$ does not take any time as the system has been set up for serving class 3 already. The total costs by applying this policy are 1039.68.

Unfortunately, this turns out not to be the optimal policy. It is better to first serve the system in mode $\{2, 4\}$, then in mode $\{1, 4\}$, next in mode $\{1, 3\}$, and finally in mode $\{3\}$. Even though in mode $\{1, 4\}$ the rate of cost decrease is larger than in mode $\{2, 4\}$ (0.69 versus 0.68), it is more beneficial to start with mode $\{2, 4\}$, followed by mode $\{1, 4\}$. The reason for that is due to non-negligible setup times and the fact that during setups the system might still partially serve certain classes. By interchanging the modes $\{2, 4\}$ and $\{1, 4\}$ one accomplishes that during the setup to mode $\{1, 3\}$ the system can still serve class 1 instead of idle. For the given parameters it turns out that the cost reduction by being able to serve class 1 during the setup to mode $\{1, 3\}$ outweighs the additional costs by serving first in mode $\{2, 4\}$ instead of mode $\{1, 4\}$. For this alternative strategy the total costs are 1039.60, which is less. This makes the problem with non negligible setup times challenging.

7 Acknowledgement

This work was supported by the Netherlands Organization for Scientific Research (NWO-VIDI grant 639.072.072).

Bibliography

[1] J.S. Baras, D.-J. Ma, and A.M. Makowski. K competing queues with geometric service requirements and linear costs: The μc-rule is always optimal. *Systems and Control Letters*, 6:173–180, 1985.

[2] J.S. Baras, D.-J. Ma, and A.M. Makowski. Two competing queues with linear costs and geometric service requirements: The μc-rule is often optimal. *Advances in Applied Probability*, 17:186–209, 1985.

[3] A. Bemporad, M. Morari, V. Dua, and E. Pistikopoulos. The explicit Linear-Quadratic regulator for constrained systems. *Automatic*, 38(1):3–20, 2002.

[4] C. Buyukkoc, P. Varaiya, and J. Walrand. The μc-rule revisited. *Advances in Applied Probability*, 17:237–238, 1985.

[5] M. Kvasnica, P. Grieder, and M. Baotić. Multi-Parametric Toolbox (MPT), 2004.

[6] P. Nain, P. Tsoucas, and J. Walrand. Interchange arguments in stochastic scheduling. *Journal of Applied Probability*, 27:815–826, 1989.

[7] P. Tøndel, T.A. Johansen, and A. Bemporad. An algorithm for multi-parametric quadratic programming and explicit MPC solutions. In *Proceedings of the 40th IEEE Conference on Decision and Control*, pages 1199–1204, Orlando, Florida, USA, December 2001.

[8] G. Weiss. A simplex based algorithm to solve separated continuous linear programs. *Mathematical Programming*, 115(1):151–198, September 2008.