Netherlands Organisation for Scientific Research

# Optimal Control of a Deterministic Multiclass Queuing System Simultaneously Serving Several Queues

Erjen Lefeber, Stefan Lämmer

Technische Universiteit
**Eindhoven**
University of Technology

Where innovation starts

November 14, 2011

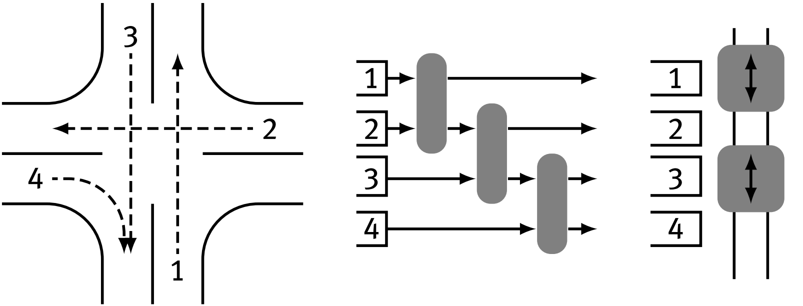
## Acknowledgment

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

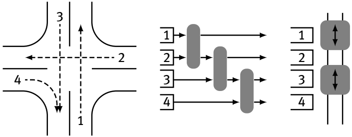

## Paper

E. Lefeber, S. Lämmer, J.E. Rooda, Optimal control of a deterministic multiclass queuing system by serving several queues simultaneously, Systems and Control Letters 60(7), 524-529, 2011.

## MCQS simultaneously serving several queues




- Intersection
- Multiclass tandem queue without buffers, e.g. hot ingots
- Polling system with physical constraints, e.g. (un)loading container vessels




Systems can be modeled as single server with 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,

and the additional modes

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

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

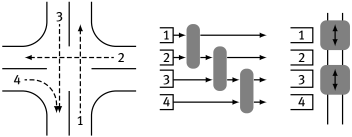
mode $\{3\}$: serve only class 3,

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

mode $\emptyset$: idle,

# Introduction






## Assumptions

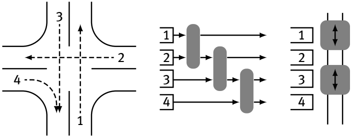
- Deterministic fluid model
- No setup times
- Unit service rate, i.e. $\mu_i = 1$ (w.l.o.g.).
- No arrivals, i.e. $\lambda_i = 0$.



Technische Universiteit Eindhoven University of Technology

# Introduction






## Assumptions

- Deterministic fluid model
- No setup times
- Unit service rate, i.e. $\mu_i = 1$ (w.l.o.g.).
- No arrivals, i.e. $\lambda_i = 0$.

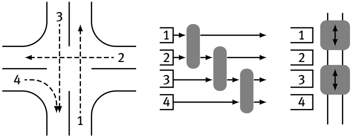
## Objective

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

where $x_i(t)$ denotes the length of queue $i$ at time $t$.

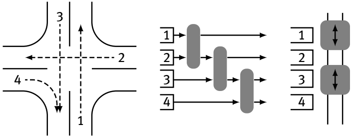

Technische Universiteit Eindhoven University of Technology

# Introduction




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

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

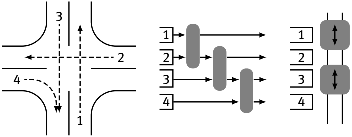



$$\min \int_0^\infty 4x_1(t) + 3x_2(t) + 2x_3(t) + 5x_4(t)\,\mathrm{d}t$$

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

| 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 |

# Introduction



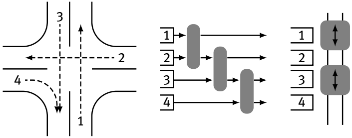

$$\min \int_0^\infty 4x_1(t) + 3x_2(t) + 2x_3(t) + 5x_4(t)\,\mathrm{d}t$$

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

| 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 |

$\mu c$-rule Total costs: 504.

- mode $\{1, 4\}$ for 6
- mode $\{2\}$ for 6
- mode $\{3\}$ for 6




$$\min \int_0^\infty 4x_1(t) + 3x_2(t) + 2x_3(t) + 5x_4(t)\,dt$$

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

| 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 |

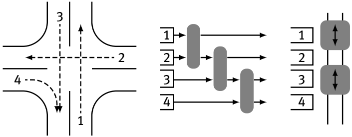
$\mu c$-rule Total costs: 504.

Min. time Total costs: 468.

- mode $\{2,4\}$ for 6
- mode $\{1,3\}$ for 6

# Introduction






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

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

| 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 |

$\mu c$-rule Total costs: 504.

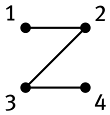
Min. time Total costs: 468.

Optimal Total costs: 456.

# Problem

- $S = (\mathcal{N}, \mathcal{C})$ undirected graph which models classes that cannot be served simultaneously
- Vertices $\mathcal{N} = \{1, 2, \ldots, N\}$: classes
- Edges $\mathcal{C} \subset \mathcal{N} \times \mathcal{N}$: conflicting classes.

For the example:



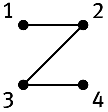

$$\mathcal{N} = \{1, 2, 3, 4\} \text{ and } \mathcal{C} = \{(1, 2), (2, 3), (3, 4)\}$$

# Problem



- $S = (\mathcal{N}, \mathcal{C})$ undirected graph which models classes that cannot be served simultaneously
- Vertices $\mathcal{N} = \{1, 2, \ldots, N\}$: classes
- Edges $\mathcal{C} \subset \mathcal{N} \times \mathcal{N}$: conflicting classes.

For the example:




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

- A set $m \subset \mathcal{N}$ is an allowed mode when $m \times m \cap \mathcal{C} = \emptyset$.
- $\mathcal{M}_S$ set of all allowed modes for system $S$.

# Problem



Dynamics:

$$\dot{x}(t) = -B_m u(t) \qquad m \in \mathcal{M}_S, \tag{1}$$

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

Constraints:

$$x_i(t) \geq 0 \qquad 0 \leq u_i(t) \leq \mu_i \qquad \forall i \in \mathcal{N}, \quad \forall t \geq 0. \tag{2}$$

# Problem



Dynamics:

$$\dot{x}(t) = -B_m u(t) \qquad\qquad m \in \mathcal{M}_S, \tag{1}$$

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

Constraints:

$$x_i(t) \geq 0 \qquad 0 \leq u_i(t) \leq \mu_i \qquad \forall i \in \mathcal{N}, \quad \forall t \geq 0. \tag{2}$$

Problem: Find feedback $u(x)$, $m(x)$ for (1) guaranteeing (2), minimizing

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

# Some basic lemmas



## Lemma (max rate)

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

# Some basic lemmas



## Lemma (max rate)

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

## Lemma ($\mu c$)

*For an optimal policy: $\sum_{i \in m_k} \mu_i c_i$ is nonincreasing for two consecutive modes $m_k$.*

# Some basic lemmas



## Lemma (max rate)

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

## Lemma ($\mu c$)

*For an optimal policy: $\sum_{i \in m_k} \mu_i c_i$ is nonincreasing for two consecutive modes $m_k$.*

## Lemma

*Switching infinitely fast between several modes can be ignored w.l.o.g.*

# Optimization problem



Let $\tau = \begin{bmatrix}\tau_{14} & \tau_{24} & \tau_{13} & \tau_4 & \tau_1 & \tau_2 & \tau_3\end{bmatrix}^T$ denote the durations of the successive modes.

# Optimization problem



Let $\tau = \begin{bmatrix} \tau_{14} & \tau_{24} & \tau_{13} & \tau_4 & \tau_1 & \tau_2 & \tau_3 \end{bmatrix}^T$ denote the durations of the successive modes.

Given $\tau$ we can determine the resulting costs, e.g.

$$\int_0^\infty x_1(s)\,\mathrm{d}s = \frac{1}{2}x_{10}^2 + (x_{10} - \tau_{14})\tau_{24} + (x_{10} - \tau_{14} - \tau_{13})\tau_4$$



Technische Universiteit Eindhoven University of Technology

# Optimization problem



Let $\tau = \begin{bmatrix} \tau_{14} & \tau_{24} & \tau_{13} & \tau_4 & \tau_1 & \tau_2 & \tau_3 \end{bmatrix}^T$ denote the durations of the successive modes.

Given $\tau$ we can determine the resulting costs, e.g.

$$\int_0^\infty x_1(s)\,\mathrm{d}s = \frac{1}{2}x_{10}^2 + (x_{10} - \tau_{14})\tau_{24} + (x_{10} - \tau_{14} - \tau_{13})\tau_4$$

In addition we have constraints like

$$x_{10} = \tau_{14} + \tau_{13} + \tau_1 \qquad \text{and} \qquad \tau_i \geq 0$$

# Optimization problem



The problem can be written as an mpQP:

$$\min_{\tau} \frac{1}{2}\tau^T H\tau - x_0^T F\tau + \frac{1}{2}x_0^T Y x_0$$

subject to

$$G\tau \leq x_0$$

which can be solved for an arbitrary parameter $x_0$.

Note that solving for $\tau_{14}$, $\tau_{24}$, and $\tau_{13}$ suffices.

# Solution (1)



$$
\begin{bmatrix} \tau_{14} \\ \tau_{24} \\ \tau_{13} \end{bmatrix} =
\begin{cases}
\begin{bmatrix} \frac{1}{2} & -\frac{1}{3} & -\frac{1}{3} & \frac{1}{2} \\ -\frac{1}{2} & \frac{1}{3} & \frac{1}{3} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{3} & -\frac{1}{2} \end{bmatrix} x_0 & \text{for } \begin{bmatrix} -3 & 2 & 2 & -3 \\ 3 & -2 & -2 & -3 \\ -3 & -2 & -2 & 3 \\ -3 & -4 & 2 & 3 \\ 3 & 2 & -4 & -3 \end{bmatrix} x_0 \leq 0 \\
\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} x_0 & \text{for } \begin{bmatrix} 0 & -1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 3 & -2 & -2 & 3 \end{bmatrix} x_0 \leq 0 \\
\begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 \end{bmatrix} x_0 & \text{for } \begin{bmatrix} 1 & 0 & -1 & -1 \\ -3 & 2 & 2 & 3 \end{bmatrix} x_0 \leq 0 \\
\begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} x_0 & \text{for } \begin{bmatrix} -1 & -1 & 0 & 1 \\ 3 & 2 & 2 & -3 \end{bmatrix} x_0 \leq 0 \\
\vdots &
\end{cases}
$$

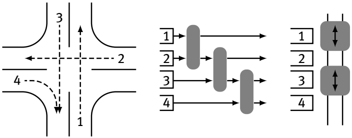
# Solution (2)



$$
\begin{bmatrix} \mathcal{T}_{14} \\ \mathcal{T}_{24} \\ \mathcal{T}_{13} \end{bmatrix} = \begin{cases} \vdots \\
\begin{bmatrix} 0 & -1 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & -1 \end{bmatrix} x_0 & \text{for } \begin{bmatrix} 0 & 1 & 0 & -1 \\ -1 & -1 & 0 & 1 \\ 3 & 4 & -2 & -3 \end{bmatrix} x_0 \leq 0 \\
\begin{bmatrix} 1 & 0 & -1 & 0 \\ -1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} x_0 & \text{for } \begin{bmatrix} -1 & 0 & 1 & 0 \\ 1 & 0 & -1 & -1 \\ -3 & -2 & 4 & 3 \end{bmatrix} x_0 \leq 0 \\
\begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} x_0 & \text{for } \begin{bmatrix} -1 & 0 & 1 & 1 \end{bmatrix} x_0 \leq 0 \\
\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} x_0 & \text{for } \begin{bmatrix} 1 & 1 & 0 & -1 \end{bmatrix} x_0 \leq 0.
\end{cases}
$$

# Example: Optimal solution





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

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

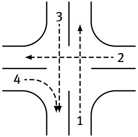
| Mode | Rate |
|---|---|
| mode $\{1,4\}$ | 9 |
| mode $\{2,4\}$ | 8 |
| mode $\{1,3\}$ | 6 |
| mode $\{4\}$ | 5 |
| mode $\{1\}$ | 4 |
| mode $\{2\}$ | 3 |
| mode $\{3\}$ | 2 |
| mode $\emptyset$ | 0 |

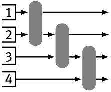
$\mu c$-rule Total costs: 504.

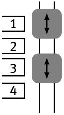
Min. time Total costs: 468.

Optimal Total costs: 456.

# Example: Optimal solution




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

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

| Mode | Rate |
|---|---|
| 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 |

$\mu c$-rule Total costs: 504.

Min. time Total costs: 468.

Optimal Total costs: 456.

- mode $\{1,4\}$ for 2 (4,6,6,4)
- mode $\{2,4\}$ for 4 (4,2,6,0)
- mode $\{1,3\}$ for 4 (0,2,2,0)
- mode $\{2\}$ for 2 (0,0,2,0)
- mode $\{3\}$ for 2 (0,0,0,0)



TU/e Technische Universiteit Eindhoven University of Technology

# Summary



Using the mpQP-approach we can solve the problem for

- Given cost vector $c$
- Arbitrary initial condition

The controller becomes a "lookup table".

# Summary



Using the mpQP-approach we can solve the problem for

- Given cost vector $c$
- Arbitrary initial condition

The controller becomes a "lookup table".

## Remaining questions

- Can we solve the problem for arbitrary $c$?
- Can we describe the controller more elegantly?

# A dynamic programming like approach



Let $\mu_i > 0$ and $c_i > 0$ be given such that the sequence of modes remains the same, i.e. $0 < \mu_3 c_3 \leq \mu_2 c_2 < \mu_1 c_1 \leq \mu_4 c_4 \leq \mu_1 c_1 + \mu_3 c_3$

## Step 1

Assume that only the final five modes are given, i.e. only mode $\{4\}$, mode $\{1\}$, mode $\{2\}$, mode $\{3\}$, and mode $\emptyset$.

# A dynamic programming like approach



Let $\mu_i > 0$ and $c_i > 0$ be given such that the sequence of modes remains the same, i.e. $0 < \mu_3 c_3 \leq \mu_2 c_2 < \mu_1 c_1 \leq \mu_4 c_4 \leq \mu_1 c_1 + \mu_3 c_3$

## Step 1

Assume that only the final five modes are given, i.e. only mode $\{4\}$, mode $\{1\}$, mode $\{2\}$, mode $\{3\}$, and mode $\emptyset$.

Solution is simple: $\tau_i = x_{i0}/\mu_i$.

# A dynamic programming like approach



Let $\mu_i > 0$ and $c_i > 0$ be given such that the sequence of modes remains the same, i.e. $0 < \mu_3 c_3 \leq \mu_2 c_2 < \mu_1 c_1 \leq \mu_4 c_4 \leq \mu_1 c_1 + \mu_3 c_3$

## Step 1

Assume that only the final five modes are given, i.e. only mode $\{4\}$, mode $\{1\}$, mode $\{2\}$, mode $\{3\}$, and mode $\emptyset$.

Solution is simple: $\tau_i = x_{i0}/\mu_i$.

Cost to go:

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

### Step 2

Assume that only the final *six* modes are given, i.e. only mode $\{1, 3\}$ mode $\{4\}$, mode $\{1\}$, mode $\{2\}$, mode $\{3\}$, and mode $\emptyset$.


Technische Universiteit Eindhoven University of Technology

## Step 2

Assume that only the final *six* modes are given, i.e. only mode $\{1,3\}$ mode $\{4\}$, mode $\{1\}$, mode $\{2\}$, mode $\{3\}$, and mode $\emptyset$.

We only need to determine $0 \leq \tau_{13} \leq \min(x_1/\mu_1, x_3/\mu_3)$

# A dynamic programming approach



## Step 2

Assume that only the final *six* modes are given, i.e. only mode $\{1,3\}$ mode $\{4\}$, mode $\{1\}$, mode $\{2\}$, mode $\{3\}$, and mode $\emptyset$.

We only need to determine $0 \leq \tau_{13} \leq \min(x_1/\mu_1, x_3/\mu_3)$

Costs made during mode $\{1,3\}$:

$$c_1\tau_{13}\frac{x_1+(x_1-\tau_{13}\mu_1)}{2}+c_2\tau_{13}x_2+c_3\tau_{13}\frac{x_3+(x_3-\tau_{13}\mu_3)}{2}+c_4\tau_{13}x_4$$

Remaining cost to go:

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

# A dynamic programming approach



## Step 2 (continued)

We need to minimize the additional cost to go:

$$\mu_3 c_3 \tau_{13} \left( \tau_{13} - \left[ \frac{x_1}{\mu_1} + \frac{x_2}{\mu_2} + \frac{x_3}{\mu_3} + \underbrace{\frac{\mu_1 c_1 + \mu_3 c_3 - \mu_4 c_4}{\mu_3 c_3}}_{\geq 0} \frac{x_4}{\mu_4} \right] \right)$$

subject to $0 \leq \tau_{13} \leq \min(x_1/\mu_1, x_3/\mu_3)$.

## Step 2 (continued)

We need to minimize the additional cost to go:

$$\mu_3 c_3 \tau_{13} \left( \tau_{13} - \left[ \frac{x_1}{\mu_1} + \frac{x_2}{\mu_2} + \frac{x_3}{\mu_3} + \underbrace{\frac{\mu_1 c_1 + \mu_3 c_3 - \mu_4 c_4}{\mu_3 c_3}}_{\geq 0} \frac{x_4}{\mu_4} \right] \right)$$

subject to $0 \leq \tau_{13} \leq \min(x_1/\mu_1, x_3/\mu_3)$.

Optimal value: $\tau_{13}^* = \min(x_1/\mu_1, x_3/\mu_3)$.



Technische Universiteit Eindhoven University of Technology

## Step 2 (continued)

We need to minimize the additional cost to go:

$$\mu_3 c_3 \tau_{13} \left( \tau_{13} - \left[ \frac{x_1}{\mu_1} + \frac{x_2}{\mu_2} + \frac{x_3}{\mu_3} + \underbrace{\frac{\mu_1 c_1 + \mu_3 c_3 - \mu_4 c_4}{\mu_3 c_3}}_{\geq 0} \frac{x_4}{\mu_4} \right] \right)$$

subject to $0 \leq \tau_{13} \leq \min(x_1/\mu_1, x_3/\mu_3)$.

Optimal value: $\tau_{13}^* = \min(x_1/\mu_1, x_3/\mu_3)$.

## Step 3 and 4

Along the same lines (add remaining two modes one at a time)

# Result if $0 < \mu_3 c_3 \leq \mu_2 c_2 < \mu_1 c_1 \leq \mu_4 c_4 \leq \mu_1 c_1 + \mu_3 c_3$



Stay in a mode until a condition is satisfied, then move to the next one

mode $\{4\}$: $x_4 = 0$

mode $\{1\}$: $x_1 = 0$

mode $\{2\}$: $x_2 = 0$

mode $\{3\}$: $x_3 = 0$

mode $\emptyset$: Stay in this mode.



Technische Universiteit Eindhoven University of Technology

# Result if $0 < \mu_3 c_3 \leq \mu_2 c_2 < \mu_1 c_1 \leq \mu_4 c_4 \leq \mu_1 c_1 + \mu_3 c_3$



Stay in a mode until a condition is satisfied, then move to the next one

mode $\{1,3\}$: $x_1 = 0$ or $x_3 = 0$

mode $\{4\}$: $x_4 = 0$

mode $\{1\}$: $x_1 = 0$

mode $\{2\}$: $x_2 = 0$

mode $\{3\}$: $x_3 = 0$

mode $\emptyset$: Stay in this mode.

# Result if $0 < \mu_3 c_3 \leq \mu_2 c_2 < \mu_1 c_1 \leq \mu_4 c_4 \leq \mu_1 c_1 + \mu_3 c_3$



Stay in a mode until a condition is satisfied, then move to the next one

mode $\{2,4\}$: $x_2 = 0$ or $x_4 = 0$

mode $\{1,3\}$: $x_1 = 0$ or $x_3 = 0$

mode $\{4\}$: $x_4 = 0$

mode $\{1\}$: $x_1 = 0$

mode $\{2\}$: $x_2 = 0$

mode $\{3\}$: $x_3 = 0$

mode $\emptyset$: Stay in this mode.

# Result if $0 < \mu_3 c_3 \leq \mu_2 c_2 < \mu_1 c_1 \leq \mu_4 c_4 \leq \mu_1 c_1 + \mu_3 c_3$



Stay in a mode until a condition is satisfied, then move to the next one

mode $\{1,4\}$: $x_1 = 0$ or $x_4 = 0$ or $x_4 \leq x_2 \wedge x_1 \leq x_3 \wedge$
$\wedge(\mu_1 c_1 - \mu_2 c_2 + \mu_3 c_3)\left(\frac{x_1}{\mu_1} + \frac{x_4}{\mu_4}\right) \leq \mu_3 c_3 \left(\frac{x_2}{\mu_2} + \frac{x_3}{\mu_3}\right)$,

mode $\{2,4\}$: $x_2 = 0$ or $x_4 = 0$

mode $\{1,3\}$: $x_1 = 0$ or $x_3 = 0$

mode $\{4\}$: $x_4 = 0$

mode $\{1\}$: $x_1 = 0$

mode $\{2\}$: $x_2 = 0$

mode $\{3\}$: $x_3 = 0$

mode $\emptyset$: Stay in this mode.

# Dynamic programming approach (general case)



Notice that if $c_4\mu_4 > c_1\mu_1 + c_3\mu_3$ mode $\{4\}$ has a higher rate of cost decrease than mode $\{1, 3\}$.

# Dynamic programming approach (general case)



Notice that if $c_4\mu_4 > c_1\mu_1 + c_3\mu_3$ mode $\{4\}$ has a higher rate of cost decrease than mode $\{1, 3\}$.

Consider modes $m_1, m_2, \ldots, m_M = \emptyset$, ordered by rate of cost decrease. We define the $i$th subproblem as follows:

- only modes $m_{M-i+1}, m_{M-i+2}, \ldots m_M$ are allowed
- (initial) state restrictions:

$$x_j(t) \geq 0 \qquad \text{for all } j \in \bigcup_{k=M-i+1}^{M} m_k \qquad \forall t \geq 0$$

$$x_j(t) = 0 \qquad \text{for all } j \notin \bigcup_{k=M-i+1}^{M} m_k \qquad \forall t \geq 0$$

# Conclusions



## Summary

- Optimal control problem: emptying deterministic single server multiclass queueing system without arrivals
- Server serves several queues simultaneously
- Sequence of modes: $\mu c$
- Buffers not necessarily empty at end of mode.
- Presented mpQP approach
- Presented dynamic programming approach

# Conclusions



## Summary

- Optimal control problem: emptying deterministic single server multiclass queueing system without arrivals
- Server serves several queues simultaneously
- Sequence of modes: $\mu c$
- Buffers not necessarily empty at end of mode.
- Presented mpQP approach
- Presented dynamic programming approach

## Future work

- Extend to systems with arrivals (SCLP (Gideon Weiss); stability)
- Extend to stochastic setting
- Include setup times (more challenging)

# Separate Continuous Linear Program



$$\min_u \int_0^T \begin{bmatrix} 4 & 3 & 2 & 5 \end{bmatrix} x(s)\,\mathrm{d}s$$

subject to

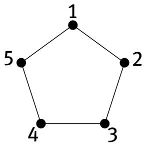
$$\dot{x}(t) = \begin{bmatrix} 0.2 \\ 0.2 \\ 0.4 \\ 0.4 \end{bmatrix} - \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} u(t) \qquad x(0) = \begin{bmatrix} 6 \\ 6 \\ 6 \\ 6 \end{bmatrix}$$

$$\begin{aligned} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} u(t) &= 1 \qquad && \forall t \geq 0 \\ x_i(t), u_i(t) &\geq 0 && \forall t \geq 0 \end{aligned}$$

# Stability



- Arrival rate $\lambda_i$. Define $\rho_i = \lambda_i/\mu_i$.
- Necessary condition for stability: $\max_{C \in \mathcal{C}} \sum_{i \in C} \rho_i < 1$
- Not sufficient




For stability not only:

$$\rho_1 + \rho_2 < 1 \quad \rho_2 + \rho_3 < 1 \quad \rho_3 + \rho_4 < 1 \quad \rho_4 + \rho_5 < 1 \quad \rho_5 + \rho_1 < 1$$

but also:

$$\rho_1 + \rho_2 + \rho_3 + \rho_4 + \rho_5 < 2.$$

# Setup times



Consider $\mu_i = 1$, setup times: 1, $c = (0.34, 0.33, 0.32, 0.35)^T$.

- First mode $\{1, 4\}$, then mode $\{2, 4\}$, next mode $\{1, 3\}$, finally mode $\{3\}$. Resulting cost: 1039.68
- First mode $\{2, 4\}$, then mode $\{1, 4\}$, next mode $\{1, 3\}$, finally mode $\{3\}$. Resulting cost: 1039.60

Reduction due to fact that during setups, the system might still partially serve certain classes.