Optimal Scheduling services in a queueing system with impatience and setup costs

Alain Jean-Marie\textsuperscript{1} Emmanuel Hyon\textsuperscript{2}

\textsuperscript{1}INRIA
LIRMM CNRS/Univ. Montpellier 2

\textsuperscript{2}Université Paris Ouest Nanterre la Défense
LIP6

ISCIS 2010, London, 22 September 2010
Outline

1. Introduction
   - The Model
   - The Question
   - The Literature
   - This talk

2. Dynamic Programming representation

3. The case \( B = 1 \)
   - Case \( B = 1 \)
   - Results
   - Propagation
   - Computation of the threshold

4. The case \( B \geq 2 \)
Scheduling with Impatience

Jean-Marie & Hyon

Introduction

The Model

The Question

The Literature

This talk

1 Introduction

Dynamic Programming representation

2 The case $B = 1$

$B = 1$

Case $B = 1$

Results

Propagation

Computation of the threshold

$B \geq 2$

Questions

3 The case $B \geq 2$

Progress
Arrival

- Customers arrive to an infinite-buffer queue.
- Time is discrete.
- The distribution of arrivals in each slot $A_t$, arbitrary with mean $\lambda$ (customers/slot)
The Model

Service
- Service occurs by *batches* of size $B$.
- Service time is one slot.
The Model

Deadline

Customers are *impatient*: they may leave before service.
- the individual probability of being impatient in each slot: $\alpha$
- memoryless, *geometrically distributed* patience
The Model

Control

Service is *controlled*.

- The controller knows the number of customers but not their amount of patience: just the distribution.
- It decides whether to serve a batch or not.
The Question

What is the optimal policy $\pi^*$ of the controller, so as to minimize the $\theta$-discounted global cost:

$$v_{\theta}^{\pi}(x) = \mathbb{E}_{x}^{\pi} \left[ \sum_{n=0}^{\infty} \theta^n c(x_n, q_n) \right],$$

where:

- $x_n$: number of customers at step $n$;
- $q_n$: decision taken at step $n$;

and $c(x, q)$ is the cost incurred, involving:

- $c_B$: cost for serving a batch (setup cost)
- $c_H$: per capita holding cost of customers
- $c_L$: per capita loss cost of impatient customers.
Related Literature

Control of queues and/or impatience (or reneging, abandonment) has a long history.

Optimal, deadline-based scheduling:

- Bhattacharya & Ephremides, 1989
- Towsley & Panwar, 1990

Optimal admission/service control (without impatience)

- Deb & Serfozo, 1973
- Altman & Koole, 1998 (admission)
- Papadaki & Powell, 2002 (service)

Optimal routing control with impatience

- Kocaga & Ward, 2009
- Movaghar, 2005

No optimal control of batch service in presence of stochastic impatience, so far.
In the talk, we:

- give the solution to this problem for $B = 1$
- explain what goes wrong when $B \geq 2$
Introduction

The Model

The Question

The Literature

This talk

Dynamic Programming representation

The case $B = 1$

Case $B = 1$

Results

Propagation

Computation of the threshold

The case $B \geq 2$
State dynamics

\[ x_n: \text{number of customers in the queue at time } n. \]
\[ q_n = 1 \text{ is service occurs, } q_n = 0 \text{ if not, at time } n. \]

Sequence of events (at each slot)

1. Begining of the slot
2. Admission in service
3. Impatience on remaining customers
4. Arrivals
State dynamics (ctd.)

The sequence of events leads to:

\[ x_{n+1} = S ([x_n - q_n B]^+) + A_{n+1} . \]

\( S(x) \): the (random) number of "survivors" after impatience, out of \( x \) customers initially present.

\( I(x) \): the number of \textit{impatient} customers.

\( B = 1 \)

\( B \geq 2 \)

\( \Rightarrow \text{ binomially distributed random variables} \)
Costs

The cost at step $n$ is:

$$c_B q_n + c_L I([x_n - q_nB]^+) + c_H [x_n - q_nB]^+$$

**Average Cost**

$$c(x, q) = q c_B + (c_L \alpha + c_H) (x - qB)^+ = q c_B + c_Q (x - qB)^+.$$  

Optimization criterion:

$$\nu_\theta^\pi (x) = \mathbb{E}_x^\pi \left[ \sum_{n=0}^{\infty} \theta^n c(x_n, q_n) \right].$$
The optimal *value function* $V(x)$ is solution to:

**The dynamic programming equation**

$$V(x) = \min_{q \in \{0, 1\}} \left\{ c_B q + c_Q [x - Bq]^+ + \theta \mathbb{E} \left( V(S([x - Bq]^+) + A) \right) \right\}.$$

The optimal policy is *Markovian* and feedback: there exists a function of the state $x$, $d(x)$, such that

$$\pi^* = (d, d, \ldots, d, \ldots)$$

and $d(x)$ is given by:

**The optimal policy**

$$d(x) = \arg \min_{q \in \{0, 1\}} \left\{ \ldots \right\}.$$
Progress

1. Introduction
   - The Model
   - The Question
   - The Literature
   - This talk

2. Dynamic Programming representation

3. The case $B = 1$
   - Case $B = 1$
   - Results
   - Propagation
   - Computation of the threshold

4. The case $B \geq 2$
Optimality Results

**Theorem**

The optimal policy is of threshold type: there exists a \( \nu \) such that \( d(x) = 1_{\{x \geq \nu\}} \).

**Theorem**

Let \( \psi \) be the number defined by

\[
\psi = c_B - \frac{c_Q}{1 - \alpha \theta}.
\]

Then,

1. If \( \psi > 0 \), the optimal threshold is \( \nu = +\infty \).
2. If \( \psi < 0 \), the optimal threshold is \( \nu = 1 \).
3. If \( \psi = 0 \), any threshold \( \nu \geq 1 \) gives the same value.
Method of Proof

Framework: propagation of properties through the dynamic programming operator (Puterman, Glasserman & Yao).

Requirement 1

\[ \exists w(\cdot) \geq 0, \sup_{(x,q)} \frac{|c(x, q)|}{w(x)} < +\infty , \]

\[ \sup_{(x,q)} \frac{1}{w(x)} \sum_y \mathbb{P}(y|x, q)w(y) < +\infty , \]

and \( \forall \mu, 0 \leq \mu < 1, \exists \eta, 0 \leq \eta < 1, \exists J, \) such that: \( \forall J\)-uple of Markov Deterministic decision rules \( \pi = (d_1, \ldots, d_J) \), and \( \forall x, \)

\[ \mu^J \sum_y P_{\pi}(y|x)w(y) \leq \eta w(x) . \]

\[ \rightarrow \] works with \( w(x) = C + cQx \)
Method of Proof

Framework: propagation of properties through the dynamic programming operator (Puterman, Glasserman & Yao).

Requirement 2

\[ \exists V^\sigma, D^\sigma \]

1. \( v \in V^\sigma \) implies \( Lv \in V^\sigma \),
2. \( v \in V^\sigma \) implies there exists a decision \( d \) such that \( d \in D^\sigma \cap \arg\min_d L_d v \),
3. \( V^\sigma \) is a closed by simple convergence.

works with:

\[ V^\sigma = \{ \text{increasing and convex} \} \] and
\[ D^\sigma = \{ \text{monotone controls} \} \]
### Propagation of structure

**Theorem**

Let, for any function \( v \), \( \tilde{v}(x) = \min_q T_v(x, q) \). Then:

1. If \( v \) increasing, then \( \tilde{v} \) increasing
2. If \( v \) increasing and convex, then \( \tilde{v} \) increasing convex

**Theorem**

If \( v \) is increasing and convex, then \( T_v(x, q) \) is submodular over \( \mathbb{N} \times Q \). As a consequence, \( x \mapsto \arg \min_q T_v(x, q) \) is increasing.

**Submodularity (Topkis, Glasserman & Yao, Puterman)**

\( g \) submodular if, for any \( \bar{x} \geq x \in \mathcal{X} \) and any \( \bar{q} \geq q \in Q \):

\[
g(\bar{x}, \bar{q}) - g(x, \bar{q}) \leq g(\bar{x}, q) - g(x, q).
\]
The system under threshold $\nu$ evolves as:

$$x_{n+1} = R_\nu(x_n) := S \left( [x_n - 1_{\{x \geq \nu\}}]^+ \right) + A_{n+1}.$$ 

A direct computation gives:

$$V_\nu(x) = \frac{cQ}{1 - \theta \bar{\alpha}} \left( x + \frac{\theta \lambda}{1 - \theta} \right) + \psi \Phi(\nu, x)$$

$$\Phi(\nu, x) = \sum_{n=0}^{\infty} \theta^n \mathbb{P}(R^{(n)}_\nu(x) \geq \nu)$$

$$\psi = c_B - \frac{cQ}{1 - \bar{\alpha} \theta}.$$
Lemma

The function $\Phi(\nu, x)$ is decreasing in $\nu \geq 1$, for every $x$.

Proof by a coupling argument. If $O_\nu^{(n)} = \text{set of customers present at time } n \text{ under threshold } \nu$, starting from $x_0 = x$:

Lemma

For every trajectory, we have

$$O_{\nu+1}^{(n)} = \begin{cases} 
\text{either} & O_\nu^{(n)} \\
\text{or} & O_\nu^{(n)} \cup \{j_n\}
\end{cases}$$

where $j_n$ is the customer of smaller index in $O_{\nu+1}^{(n)}$.

$$\implies \{R_{\nu+1}^{(n)}(x) \geq \nu + 1\} \subset \{R_\nu^{(n)}(x) \geq \nu\}.$$
Progress

1. Introduction
   - The Model
   - The Question
   - The Literature
   - This talk

2. Dynamic Programming representation

3. The case $B = 1$
   - Case $B = 1$
   - Results
   - Propagation
   - Computation of the threshold

4. The case $B \geq 2$
What goes wrong when $B \geq 2$

Numerical experiments and exact results in special cases reveal that:

- The value function $V(x)$ is not convex in general
- The function $TV(x, q)$ is not submodular in general

Examples with $B = 10$, $\alpha = 1/10$, $\theta = 8/10$: $V$ not convex
What goes wrong when $B \geq 2$, ctd.

Submodularity: if $Tv(x,1)$ is submodular, then $x \mapsto Tv(x,1) - Tv(x,0)$ is decreasing.

A counterexample with $B = 2$, $\lambda = 1/10$, $\alpha = 9/10$, $\theta = 9/10$. 

![Graph showing increments of $Tv(x,1) - Tv(x,0)$]
What goes wrong when $B \geq 2$, end.

Papadaki & Powell study the same problem without impatience.

**Dynamics without impatience**

$$x_{n+1} = [x_n - q_nB]^+ + A_{n+1}.$$

They show that the following “K-convexity” propagates:

**K-convexity**

$$V(x + K) - V(x) \geq V(x - 1 + K) - V(x - 1).$$

Also used in Altman & Koole for batch arrivals.

$\implies$ does not work here.

Questions?
Bibliography

The complete paper


References on optimal Markovian control theory


References on the control of queues

- **R. K. Deb and R. F. Serfozo.**
  Optimal control of batch service queues.

- **E. Altman and G. Koole.**
  On submodular value functions and complex dynamic programming.
  *Stochastic Models, 14:1051–1072, 1998.*

- **K. P. Papadaki and W. B. Powell.**
  Exploiting structure in adaptative dynamic programming algorithms for a stochastic batch service problem.

- **Y. L. Koçaga and A. R. Ward.**
  Admission control for a multi-server queue with abandonment.
  Forthcoming, *Queueing Systems, July 2009.*
Control of queues with deadlines


Extensions to the model

Average case / no discount: \( \theta = 1 \).

\[ \implies \text{should work as long as } \alpha \neq 0 (\overline{\alpha} \neq 1) \]

Critical value:

\[
\psi = c_B - c_Q \frac{1}{\alpha} = c_B - c_L - \frac{c_H}{\alpha}. 
\]

Branching processes: at each step, each customer is replaced by \( X \) customers. \( \overline{\alpha} = \mathbb{E}X \), must be \( \overline{\alpha} < \theta^{-1} \).

\[ \implies \text{same formula for the optimal policy} \]

Critical value:

\[
\psi = c_B - \frac{c_Q}{1 - \overline{\alpha} \theta}. 
\]