Appointment-driven queueing systems
ORAHS: Leuven 2009

Stefan Creemers & Marc Lambrecht
Research Center for Operations Management
Katholieke Universiteit Leuven

July 17, 2009
Presentation outline

- Introduction
- problem setting
- Optimization
- Methodology
- Questions
Introduction: Traditional Appointment Scheduling

Traditional appointment scheduling policies:

- Fixed number of customers served during a given service session
- Use of static/dynamic scheduling rules or procedures
- Customers are scheduled well in advance of the service session
Introduction: Traditional Appointment Scheduling

Traditional appointment scheduling policies:

- Fixed number of customers served during a given service session
- Use of static/dynamic scheduling rules or procedures
- Customers are scheduled well in advance of the service session

Outpatient scheduling in health care: a review of literature

Production and Operations Management, 12(4), 519-549
Introduction: Traditional Appointment Scheduling

Traditional appointment scheduling policies:

- Fixed number of customers served during a given service session
- Use of static/dynamic scheduling rules or procedures
- Customers are scheduled well in advance of the service session

Outpatient scheduling in health care: a review of literature
Production and Operations Management, 12(4), 519-549

⇒ Cancelation and no-shows result in process variability
Introduction: Open-Access Scheduling

Open-access appointment scheduling policies:

- Random number of customers served during a service session
- Appointments are made during the same day
- Trade-off versus traditional appointment scheduling:
  - Less no-show and cancelation
  - Increased variability in day-to-day workload
Introduction: Open-Access Scheduling

Open-access appointment scheduling policies:

- Random number of customers served during a service session
- Appointments are made during the same day
- Trade-off versus traditional appointment scheduling:
  - Less no-show and cancelation
  - Increased variability in day-to-day workload

Dynamic Scheduling of Outpatient Appointments under Patient No-shows and Cancellations
To appear in MSOM
Introduction: Open-Access Scheduling

Open-access appointment scheduling policies:

- Random number of customers served during a service session
- Appointments are made during the same day
- Trade-off versus traditional appointment scheduling:
  - Less no-show and cancelation
  - Increased variability in day-to-day workload
Introduction: Open-Access Scheduling

Open-access appointment scheduling policies:

- Random number of customers served during a service session
- Appointments are made during the same day
- Trade-off versus traditional appointment scheduling:
  - Less no-show and cancelation
  - Increased variability in day-to-day workload

A Comparison of Traditional and Open-Access Appointment Scheduling Policies

Working Paper
Introduction:

Existing approaches:

- In traditional appointment scheduling the randomness results from no-shows and cancelations.
- In Open-access appointment scheduling scheduling randomness results from the variation in the number of arriving customers.
Introduction:

Existing approaches:

- In traditional appointment scheduling the randomness results from no-shows and cancelations
- In Open-access appointment scheduling randomness results from the variation in the number of arriving customers

⇒ Existing approaches only observe a single service session
Introduction:

Existing approaches:

- In traditional appointment scheduling the randomness results from no-shows and cancelations
- In Open-access appointment scheduling randomness results from the variation in the number of arriving customers

⇒ Existing approaches only observe a single service session

Questions unanswered:

- How long do customers have to wait in the waiting list?
- What is the origin of the variability of the arrival process?
- How to assess the variability of the arrival process?
- . . .
Problem description: Setting

January 15
Problem description: Setting

- January 15
- April 15

Appointment-driven queueing systems
Problem description: Setting

Introduction
Problem Setting
Performance Measures
Strategic Questions

Problem description: Setting

Appointment-driven queueing systems
Problem description: Setting

Stefan Creemers
Appointment-driven queueing systems
Problem description: Setting

January 15

January 15

April 15

Waiting time at waiting list

Waiting time at service facility

8AM 10AM 12PM 2PM 4PM 6PM

LUNCH
Problem description: Setting

- January 15
- January 15
- April 15
- Server idle time
- Waiting time at waiting list
- Waiting time at service facility

Stefan Creemers
Appointment-driven queueing systems
Problem description: Setting

- January 15
- April 15
- Server idle time
- Server overtime
- Waiting time at waiting list
- Waiting time at service facility

Stefan Creemers
Appointment-driven queueing systems
Problem description: Performance measures

- Waiting time at the waiting list
- Waiting time at the service facility
- Server idle time
- Server overtime
Problem description: Strategic questions

- Waiting time at the waiting list
- Waiting time at the service facility
- Server idle time
- Server overtime

- How many service sessions should be installed
- When should a service session be installed
- How many customers should be served during each session
- How should customers be scheduled during a session
- ...

Stefan Creemers
Appointment-driven queueing systems
Problem description: Strategic questions

- Waiting time at the waiting list
- Waiting time at the service facility
- Server idle time
- Server overtime

- How many service sessions should be installed
- When should a service session be installed
- How many customers should be served during each session
- How should customers be scheduled during a session

...
Optimization: Insights

Number of customers allowed during a session

Performance
Optimization: Insights

Waiting time at the waiting list increases as the number of customers allowed during a session increases.
Optimization: Insights

Waiting time at the waiting list

Number of customers allowed during a session

Performance

MON | TUE | WED | THU | FRI | SAT | Time

Appointment-driven queueing systems
Optimization: Insights

Waiting time at the waiting list

Number of customers allowed during a session

MON TUE WED THU FRI SAT Time

Performance
Optimization: Insights

Waiting time at the waiting list

Number of customers allowed during a session

Performance

MON TUE WED THU FRI SAT

Time

Appointment-driven queueing systems
Optimization: Insights

- Waiting time at the waiting list
- Number of customers allowed during a session

Stefan Creemers
Appointment-driven queueing systems
Optimization: Insights

- **Waiting time at the waiting list**

Number of customers allowed during a session

- **Performance**

- **MON**
- **TUE**
- **WED**
- **THU**
- **FRI**
- **SAT**

Time

Stefan Creemers

Appointment-driven queueing systems
Optimization: Insights

Performance vs. Number of customers allowed during a session

Waiting time at the waiting list

MON TUE WED THU FRI SAT

Time

Appointment-driven queueing systems
Optimization: Insights

- Waiting time at the waiting list

Number of customers allowed during a session

Performance

MON TUE WED THU FRI SAT

Stefan Creemers
Appointment-driven queueing systems
Optimization: Insights

Waiting time at the waiting list

Number of customers allowed during a session
Optimization: Insights

Performance

Waiting time at the waiting list

Number of customers allowed during a session

MON, TUE, WED, THU, FRI, SAT

Time

Stefan Creemers
Appointment-driven queueing systems
Optimization: Insights

Waiting time at the waiting list

Number of customers allowed during a session

Performance
Optimization: Insights

- Waiting time at the waiting list

Number of customers allowed during a session

Performance

MON | TUE | WED | THU | FRI | SAT | Time

Appointment-driven queueing systems
Optimization: Insights

Waiting time at the waiting list

Number of customers allowed during a session

Performance

MON  TUE  WED  THU  FRI  SAT  Time

Appointment-driven queueing systems
Optimization: Insights

- **Performance**
- **Waiting time at the service facility**
- **Number of customers allowed during a session**

**Areas of application**

Appointment-driven queueing systems
Optimization: Insights

- Server idle time vs. number of customers allowed during a session.

Stefan Creemers
Appointment-driven queueing systems
Optimization: Insights

Server idle time

Number of customers allowed during a session

MON TUE WED THU FRI SAT

Server idle time:

Versus

Appointment-driven queueing systems
Optimization: Insights

- Cost Function
- Methodology
- Areas of application

Stefan Creemers
Appointment-driven queueing systems
Optimization: Insights

![Graph showing performance over time]

- Server overtime

Number of customers allowed during a session

MON | TUE | WED | THU | FRI | SAT

Server overtime: [ ] [ ] [ ] [ ] [ ] [ ]

Versus

[ ] [ ]
Optimization: Insights

- Server overtime

Number of customers allowed during a session

Server overtime:  

MON | TUE | WED | THU | FRI | SAT
---|---|---|---|---|---
[ ] | [ ] | [ ] | [ ] | [ ] | [ ]

Stefan Creemers
Appointment-driven queueing systems
Optimization: Cost function

- Waiting at list
- Waiting at facility
- Server overtime
- Server idle time

Number of customers allowed during a session vs. Costs
Optimization: Cost function

- Costs
- Total cost function
- Waiting at list
- Waiting at facility
- Server overtime
- Server idle time

Number of customers allowed during a session
Optimization: Cost function

- Total cost function
- Optimum
- Waiting at list
- Waiting at facility
- Server overtime
- Server idle time

Number of customers allowed during a session vs Costs
Areas of application

- Health care and services in general (e.g. Cayirli & Veral 2003)
- Manufacturing systems (e.g. Biskup, Herrmann & Gupta 2008)
- Transportation (e.g. Namboothiri & Erera 2008)
- Telecommunications and computing (e.g. van Leeuwaarden, Denteneer & Resing 2006)
- Open Access appointment systems (e.g. Liu, Ziya & Kulkarni 2009)
- ...
Modeling approach

An appointment-driven system is a combination of two distinct queueing systems:

- **Service Allocation Model (SAM)**: observes the queueing behavior of customers from the making of an appointment until the start of the service session in which service is administered.

- **Customer Appointment System (CAS)**: observes the queueing behavior of customers during a service session itself.
Modeling approach

From the SAM we obtain:

- The expected waiting time of a customer from the making of an appointment until the start of the service session in which service is administered
- The distribution of the number of customers to be served during a given service session

This latter parameter is used to weigh the performance measures resulting from the analysis of the CAS:

- Customer waiting time at the service facility
- Server idle time
- Server overtime
Link between SAM and CAS

- Service Allocation Model (SAM)
- Customer Appointment System (CAS)

**Problem description**

**Optimization**

**Methodology**

**Questions**

**Modeling approach**

**Link between SAM and CAS**

**Appointment-driven queueing systems**

Stefan Creemers
Service Allocation Model (SAM)
Synopsis

- Model: set of two-dimensional DTMC
- Methodology: matrix-analytical techniques and efficient algorithms implemented in Visual C++
- Results: Numerically exact results, limited computational requirements allow the study of complex, real-life problems

Queueing models for appointment-driven systems
Accepted for publication in Annals of OR

An advanced queueing model to analyze appointment-driven service systems
Computers & operations research, 36(10), 2773-2785
Assumptions

- Customer arrival and service processes are assumed to be cyclic (hence server vacations are cyclic as well).
- Customers are allowed during an arrival session $i$; each arrival session $i$ is characterized by: a deterministic length $T_{a_i}$, a $PH$ arrival process $i$.
- At the start of a service session $i$, up to $k_i$ customers are removed from the queue (FCFS).
- After removal of up to $k_i$ customers, a vacation of deterministic length $T_i$ is initiated.
- $k_i$ and $T_i$ are time-dependent.
General model

Inefficient approach: model the SAM as a Markov chain of four dimensions:

- Queue size $Q$
- Vacation type $j$
- Phase of the arrival process $m$
- Phase of the vacation process $\nu$

Efficient approach: use a set of two-dimensional DTMC $X_j$ that observe the number of customers in queue at the start of a vacation of type $j$. 
DTMC $X_j$

Solution procedure:

• Counting process to obtain the probability of a number of customers arriving during a deterministic time interval

• Efficient algorithm to obtain the transition probabilities of the DTMC $X_j$

• Matrix analytical methods to obtain the stationary distribution of the DTMC

Two-dimensional statespace represented by pairs $(Q, m)$:

• $Q$; the number of customers in queue

• $m$; the phase of the arrival process
DTMC $X_j$

Transition matrix $\mathcal{P}$ may be represented as a non-skip-free $M/G/1$ Markov chain:

$$
\mathcal{P} = \begin{bmatrix}
B_{0,0} & B_{0,1} & B_{0,2} & B_{0,3} & \cdots \\
B_{1,0} & B_{1,1} & B_{1,2} & B_{1,3} & \cdots \\
\vdots & \vdots & \vdots & \vdots & \ddots \\
B_{Q_c-1,0} & B_{Q_c-1,1} & B_{Q_c-1,2} & B_{Q_c-1,3} & \cdots \\
A_0 & A_1 & A_2 & A_3 & \cdots \\
0 & A_0 & A_1 & A_2 & \cdots \\
0 & 0 & A_0 & A_1 & \cdots \\
\vdots & \vdots & \vdots & \vdots & \ddots
\end{bmatrix}
$$
DTMC $X_j$

Using Matrix analytical methods, we obtain:

- the waiting time of a customer at the waiting list
- $\pi[s|j]$; the probability of having $s$ customers in queue at the start of a vacation of type $j$ (prior to the removal of $k_j$ customers); the probability of having to service $\text{min}(s, k_j)$ customers during a service session $j$

Both parameters are computed in a numerical exact manner
Results

Solution procedures are implemented in Visual C (adopting GSL and BLAS routines to perform matrix operations)

<table>
<thead>
<tr>
<th>$\rho$</th>
<th>$CI(X_j)$</th>
<th>sec($CP$)</th>
<th>sec($X_j$)</th>
<th>sec($SIM$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.75</td>
<td>1.00</td>
<td>0.25</td>
<td>0.23</td>
<td>6.64</td>
</tr>
<tr>
<td>0.85</td>
<td>1.00</td>
<td>0.26</td>
<td>0.42</td>
<td>7.44</td>
</tr>
<tr>
<td>0.90</td>
<td>1.00</td>
<td>0.27</td>
<td>0.72</td>
<td>7.86</td>
</tr>
<tr>
<td>0.95</td>
<td>1.00</td>
<td>0.28</td>
<td>7.78</td>
<td>8.31</td>
</tr>
<tr>
<td>0.99</td>
<td>0.99</td>
<td>0.29</td>
<td>10.4</td>
<td>17.9</td>
</tr>
</tbody>
</table>

Table: Confidence en CPU times for various traffic intensities
Customer Appointment System (CAS)
Customer Appointment System (CAS)

- Model: Four- to five-dimensional DTMC depending on the assumptions imposed
- Methodology: efficient algorithms implemented in Visual C++
- Results: Depending on the granularity used in the algorithms near exact results may be obtained

Appointment-driven queueing systems
Creemers S. (2009)
PhD Thesis
Assumptions

- Customers are allowed to arrive early, late or may even fail to show up (overtaking of customers is possible)
- Customers have a unique arrival process characterization (i.e. unique probability to show up, unique probability to arrive early/late and unique corresponding distributions)
- All arriving customers are served
- Customers are served at a single server
- We assume general customer service times
- The service process of a customer is allowed to be interrupted (preemptive and non-preemptive interrupts)
- The start of a service session may be delayed
Definition

The goal of any Appointment System (AS) is to schedule customers as to optimize some set of performance measures. In order to obtain the scheduled arrival times of customers, one may resort to the use of:

- Appointment Scheduling Rules (ASRs)
- Scheduling procedures

We adopt a set of 314 ASR and optimize performance over:

- Customer waiting time at the waiting list
- Server idle time
- Server overtime
Different ASR

INDIVIDUAL APPOINTMENT SCHEDULING RULES

SCHEDULED ARRIVAL TIMES

μ^−1  μ^−1  μ^−1  μ^−1  μ^−1  μ^−1  μ^−1  μ^−1  μ^−1

1  2  3  4  5  6  7  8  9  10
Different ASR

**BLOCK APPOINTMENT SCHEDULING RULES**

1. 2  2. 4  3. 6  4. 8

---

**SCHEDULED ARRIVAL TIMES**

\[2\mu^{-1}\]

---

**Diagram:**

- Numbers 1 to 10
- Blocks indicating scheduled arrival times

---

Stefan Creemers

Appointment-driven queueing systems
Different ASR

EL APPOINTMENT SCHEDULING RULES

SCHEDULED ARRIVAL TIMES

Stefan Creemers
Appointment-driven queueing systems
Model Definition

- The CAS is modeled as a DTMC and observes the system over unit time interval ($\Delta$)
- Service and arrival distributions need to be discretized
- Discretization yields:
  - $P[S = x]$; the probability of completing service during time interval $[x\Delta, (x + 1)\Delta)$
  - $P[A = x]$; the probability of a particular customer arriving during time interval $[x\Delta, (x + 1)\Delta)$
Discretization

CUMULATIVE DISTRIBUTION

0  Δ  2Δ  3Δ  4Δ
Discretization

CUMULATIVE DISTRIBUTION

\[ P[S=0] \
\[ P[S=1] \]
\[ P[S=2] \]
\[ P[S=3] \]

\[ 0 \Delta \]
\[ P[S=0] \]
\[ P[S=1] \]
\[ P[S=2] \]
\[ P[S=3] \]
\[ 4\Delta \]
Statespace definition

The statespace of the DTMC may be represented using quadruples \((x, Q, y, S)\):

- \(x\); the current time in the system
- \(Q\); the number of customers in queue at time instance \(x\Delta\)
- \(y\); the phase of the service process
- \(S\); the set of units that still have to arrive

\(S \subseteq T_x \subseteq N\) where:

- \(T_x\) denotes the set of customers allowed to arrive at a time instance \(x\Delta\)
- \(N\) denotes the set of all customers

Further define \((E_x = T_x \setminus T_{x-1})\); the set of units that has become eligible to arrive at a time instance \(x\Delta\)
State transitions (simplified)

- Service is ongoing:
  - \((x, Q, y, S) \rightarrow ((x+1), \max(0, (Q+|U|−1)), −1, (S \setminus U))\) upon service completion and arrival of a set of customers \(U\)
  - \((x, Q, y, S) \rightarrow ((x+1), (Q + |U|), (y + 1), (S \setminus U))\) upon service advancement and arrival of a set of customers \(U\)

- Server is idle yet \((S \neq \emptyset)\):
  - \((x, 0, −1, S) \rightarrow ((x + 1), (|U| − 1), 0, (S \setminus U))\) upon arrival of a set of customers \((U \neq \emptyset)\)
  - \((x, Q, y, S) \rightarrow ((x + 1), 0, −1, S)\) if no customers arrive during \([x\Delta, (x + 1)\Delta)\)

- Server is idle and \((S = \emptyset)\); all customers have been served, an absorbing state has been entered
Obtaining performance measures

An efficient algorithm is developed to assess:

- the probability of visiting a transient state
- the probability of ending up in an absorbing state at time instance $x\Delta$

A state $(x, Q, y, S)$ is associated with:

- A total customer waiting time of $Q\Delta$ time units
- A server idle time of $\Delta$ time units if $(y = -1)$
- A server idle time of $(O - x\Delta)$ time units if $(x, Q, y, S)$ is an absorbing state and $(x\Delta < O)$
- A server overtime of $(x\Delta - O)$ time units if $(x, Q, y, S)$ is an absorbing state and $(x\Delta > O)$
Time for questions