Appointment scheduling and resource planning under uncertainty

DAVID PHILLIPS

Department of Mathematics
United States Naval Academy

Joint work with
MAJ Marcus Colyer, M.D. (Walter Reed)
MIDN Marisa Molkenbuhr (USNA)

November 13, 2014
Have you heard this one? *So a mathematician walks into a room full of healthcare providers...*
Have you heard this one? So a mathematician walks into a room full of healthcare providers...

- Appointment scheduling at a glaucoma clinic
- A stochastic integer program
- Discussion: Did it help? How could it be more helpful?
Appointment scheduling with uncertain process times

- $n$ jobs
- $m$ machines where jobs require some amount of time on each machine (may be zero)
- $P_{ij}$ is the random time job $j$ requires on machine $i$
Appointment scheduling with uncertain process times

- $n$ jobs
- $m$ machines where jobs require some amount of time on each machine (may be zero)
- $P_{ij}$ is the random time job $j$ requires on machine $i$
- $S_i$ is the set of appointment slots, i.e., a finite set of possible scheduled start times for jobs on machine $i$
  - The start time is not necessarily the scheduled start time!
Appointment scheduling with uncertain process times

- $n$ jobs
- $m$ machines where jobs require some amount of time on each machine (may be zero)
- $P_{ij}$ is the random time job $j$ requires on machine $i$
- $S_i$ is the set of appointment slots, i.e., a finite set of possible scheduled start times for jobs on machine $i$
  - The start time is not necessarily the scheduled start time!

- Multiple objectives
  - maximize jobs complete
  - minimize machine idle time
  - minimize job wait time (scheduled start time - start time)
  - minimize average completion time
Problem description and assumptions

- Patient flow through the glaucoma clinic:
  1. Technicians administer Visual Fields testing (VF)
  2. One of two doctors then see patients

- Two groups of patients:
  - SPEC & PROC: new and repeat patients requiring VF
  - EST: repeat patient not requiring VF

- Patients are already assigned doctors.

- Objectives we modeled:
  - Maximize the number of patients seen
  - Minimize the idle time of doctors/VF
  - Minimize the wait time of patients between doctors/VF
  - Minimize the average completion time

- Time required by doctors ($P^D_{ij}$), VF ($P^{VF}$) is random.
Decision variables
Decision variables

Which doctor do patients see and when?

\[ x_{ijt} = \begin{cases} 
1 & \text{if doctor } i \text{ sees patient } j \text{ at time } t \\
0 & \text{otherwise} 
\end{cases} \]
Decision variables

Which doctor do patients see and when?

\[ x_{ijt} = \begin{cases} 
1 & \text{if doctor } i \text{ sees patient } j \text{ at time } t \\
0 & \text{otherwise}
\end{cases} \]

For patients requiring VF, when?

\[ y_{jt} = \begin{cases} 
1 & \text{if patient } j \text{ has a VF at time } t \\
0 & \text{otherwise}
\end{cases} \]
Gist of the IP

- **Maximize** (patients seen) - (idle time) - (wait times) - (completion times)

- Some of the constraints:
  - Can only schedule a patient at most once.
  - Patients that need VF get one if seen.
  - Doctors can only see one patient at a time.
  - Up to three VFs at any one time.
  - Patients are scheduled into time slots
Gist of the IP

- **Maximize** (patients seen) - (idle time) - (wait times) - (completion times)

- Some of the constraints:
  - Can only schedule a patient at most once.
    \[
    \sum_{i,t} x_{jti} \leq 1, \forall j
    \]
  - Patients that need VF get one if seen.
  - Doctors can only see one patient at a time.
  - Up to three VFs at any one time.
  - Patients are scheduled into time slots
Gist of the IP

- **Maximize** (patients seen) - (idle time) - (wait times) - (completion times)

- Some of the constraints:
  - Can only schedule a patient at most once.
  - Patients that need VF get one if seen.

\[
\sum_{t} y_{jt} \geq \sum_{i,t} x_{ijt}, \forall j
\]

- Doctors can only see one patient at a time.
- Up to three VFs at any one time.
- Patients are scheduled into time slots
Gist of the IP

- **Maximize** (patients seen) - (idle time) - (wait times) - (completion times)

- Some of the constraints:
  - Can only schedule a patient at most once.
  - Patients that need VF get one if seen.
  - Doctors can only see one patient at a time.

\[
\sum_{j} \sum_{s=t}^{t+P_{ij}^D} x_{ijs} \leq 1, \forall i
\]

- Up to three VFs at any one time.
- Patients are scheduled into time slots
Gist of the IP

- **Maximize** (patients seen) - (idle time) - (wait times) - (completion times)

- Some of the constraints:
  - Can only schedule a patient at most once.
  - Patients that need VF get one if seen.
  - Doctors can only see one patient at a time.
  - Up to three VFs at any one time.

\[
\sum_{j} \sum_{s=t}^{t+P^{VF}} y_{js} \leq 3, \forall i
\]

- Patients are scheduled into time slots
Gist of the IP

- **Maximize** (patients seen) - (idle time) - (wait times) - (completion times)

- Some of the constraints:
  - Can only schedule a patient at most once.
  - Patients that need VF get one if seen.
  - Doctors can only see one patient at a time.
  - Up to three VFs at any one time.
  - Patients are scheduled into time slots

Restrict time slots for each variable
Computations

- Our model size: for a two-week schedule (4 days), \( \approx 2000 - 9000 \) constraints, \( \approx 13,000 - 30,000 \) variables

- Our computer: 1.7 gigahertz machine with 8 GB of RAM, solves in \( \approx 10 \) minutes

- Our solver: Gurobi 5.6.3 (using Python)
Computations

- Our model size: for a two-week schedule (4 days), \( \approx 2000 – 9000 \) constraints, \( \approx 13,000 – 30,000 \) variables

- Our computer: 1.7 gigahertz machine with 8 GB of RAM, solves in \( \approx 10 \) minutes

- Our solver: Gurobi 5.6.3 (using Python)

- Uncertainty: We use a Monte-Carlo simulation:
  - Randomly generate a set of process times
  - Solve the optimization model
  - Analyze the average of the results
  - Requires solving the model many times
  - Requires data on the process times
### Processing times (minutes)

<table>
<thead>
<tr>
<th>Entity</th>
<th>n</th>
<th>Mean</th>
<th>Sample Error</th>
<th>95% Conf. Int.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Doctor 1</td>
<td>51</td>
<td>17.6</td>
<td>1.2</td>
<td>[15.7,19.3]</td>
</tr>
<tr>
<td>Doctor 2</td>
<td>11</td>
<td>12.5</td>
<td>2.3</td>
<td>[8.3,14.3]</td>
</tr>
<tr>
<td>Visual Fields</td>
<td>60</td>
<td>25.9</td>
<td>1.0</td>
<td>[24.2,27.6]</td>
</tr>
</tbody>
</table>
Distribution fitting

- Doctor 1: logistic distribution
- Doctor 2: exponential distribution
- VF: Inverse gaussian

Thanks to CDR David Ruth, Ph.D. for statistical help!
Growth of one IP solve to .03% opt.
2 days = one week, 1000+ IPs required
96% C.I. is ±10%.
Growth of one IP solve to .03% opt. 
1000+ solves required 
2 days = one week
So how can the model help us?

- Analyze tradeoff in appointment slot sizes
- Analyze resource allocation questions
- Analyze the impact of variance on scheduling
- Optimal “average processing time” schedules
### Slot size tradeoffs

#### Doctor 1:

<table>
<thead>
<tr>
<th>Objective</th>
<th>10</th>
<th>15</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pts/day</td>
<td>19.2 ± 0.1</td>
<td>17.2 ± 0.2</td>
<td>16.8 ± 0.2</td>
</tr>
<tr>
<td>Wait time</td>
<td>11.5 ± 3.4</td>
<td>10.6 ± 3.5</td>
<td>9.8 ± 3.2</td>
</tr>
<tr>
<td>Utilization</td>
<td>80% ± 0.5%</td>
<td>68% ± 0.7%</td>
<td>66% ± 0.9%</td>
</tr>
</tbody>
</table>

#### Doctor 2:

<table>
<thead>
<tr>
<th>Objective</th>
<th>10</th>
<th>15</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pts/day</td>
<td>20.0 ± 0.04</td>
<td>17.4 ± 0.2</td>
<td>12.9 ± 0.2</td>
</tr>
<tr>
<td>Wait time</td>
<td>12.1 ± 3.4</td>
<td>7.7 ± 5.7</td>
<td>6.1 ± 2.3</td>
</tr>
<tr>
<td>Utilization</td>
<td>65% ± 2.2%</td>
<td>43% ± 1.1%</td>
<td>24% ± 1.4%</td>
</tr>
</tbody>
</table>
How was it useful?

- Probably the most useful insight was the VF processing time...
  - Prevalent belief was that a VF requires $\approx 20$ minutes.
  - Seemed to convince techs that they needed to keep track of “difficult patients”
How was it useful?

- Probably the most useful insight was the VF processing time...
  - Prevalent belief was that a VF requires \( \approx 20 \) minutes.
  - Seemed to convince techs that they needed to keep track of “difficult patients”

- The next most useful insight was that there was evidence that they want the impossible: their desired throughput doesn’t seem achievable under idealized conditions.
  - Naturally, more analysis required, i.e., more runs
How was it useful?

- Probably the most useful insight was the VF processing time...
  - Prevalent belief was that a VF requires \( \approx 20 \) minutes.
  - Seemed to convince techs that they needed to keep track of “difficult patients”

- The next most useful insight was that there was evidence that they want the impossible: their desired throughput doesn’t seem achievable under idealized conditions.
  - Naturally, more analysis required, i.e., more runs

- Finally, the slot size analysis seemed helpful in that allocating more time per patient than they previously had been seemed like a good tradeoff.
  - Again, more analysis required
Enhancing the model

- Add-on patients?
- Doctors sharing patients
- Data limitations - Doctor 2 with only 11 observations
- Patient differences? SPEC versus PROC versus EST? Even more specific?
Discussion 1:

Suggested future directions to Walter Reed:

- Impact of adding/subtracting resources (extra doctor, VF)
- Dedicated tool to help with real-time scheduling
- Strategic insights into scheduling with uncertainty to help guide the current scheduling process
- Better understand the current uncertainty in the model
- What are your priorities with respect to idle times, patient wait times, number of patients seen, and overtime? Are there other objectives of interest?
Discussion 2:

**Researcher interests:**

- About that runtime...
  - Approximation approaches – rounding the LP
  - Tightening the IP with better constraints
  - Set up parallel computation

- Uncertainty
  - Augment model to two stages so that add-ons are modeled
  - Use a chance constraint instead of Monte Carlo?
  - Robust optimization approach?

- Approximation problems: abstract to a clean model
  - Investigate best approximation algorithms (e.g., Skutella, et al, 2014)
  - Yields strategic insights

- Your thoughts?