A Data-driven Stochastic Optimization Approach to the Colonoscopy Scheduling Problem

Karmel S. Shehadeh, PhD Candidate
Amy E.M. Cohn, Professor

Department of Industrial and Operations Engineering
Center for Healthcare Engineering and Patient Safety
University of Michigan, Ann Arbor

November 5, 2018
INFORMS 2018 | Phoenix
Acknowledgements
Colonoscopy Procedure

- The most common screening test for Colorectal Cancer, the third leading cause of cancer-related death in the US

- Allows for direct visual examination of the entire colon and rectum
  - Spot existing cancer, prompting treatment
  - Prevent future cancer, by detecting precancerous growth

- Can help reduce CRC incidence by about 40% and mortality by about 50%
Significant variability in procedure duration due to the quality of the pre-procedure bowel prep that the patient must undergo.
Challenges to Colonoscopy Scheduling at Michigan Medicine

- **Significant variability** in procedure duration due to the quality of the pre-procedure **bowel prep** that the patient must undergo.

- Patient **non-punctuality** (future talk)

Shehadeh, K.S., Cohn, A.
University of Michigan
Challenges to Colonoscopy Scheduling at Michigan Medicine

- **Significant variability** in procedure duration due to the quality of the pre-procedure *bowel prep* that the patient must undergo.

- Patient **non-punctuality** (*future talk)*

- **Multiple** and **competing** performance criteria.
Research Goal

- **Optimize** daily schedule **templates** and policies for filling these templates, to best account for patient characteristics and the associated **variability**.
Research Goal

- **Optimize** daily schedule **templates** and policies for filling these templates, to best account for patient characteristics and the associated variability.

- By building **higher-quality schedules**, it is possible to:
  - Reduce **costs**
  - Improve patient and provider **satisfaction**
  - Achieve **better clinical outcomes**, and more
Research Goal

- **Optimize** daily schedule templates and policies for filling these templates, to best account for patient characteristics and the associated variability.

- By building higher-quality schedules, it is possible to:
  - Reduce costs
  - Improve patient and provider satisfaction
  - Achieve better clinical outcomes, and more

- A valuable tool in creating such templates is the ability to solve the Stochastic Outpatient Procedure Scheduling Problem (SOPSP) as an embedded subproblem.
Stochastic Outpatient Procedure Scheduling Problem (SOPSP)

A clinic manager must schedule the start times for a set of procedures, where

- Each procedure is of a known type and associated probability distribution of (non-negative) random duration
- Although patients may fail to show up to their appointments, we assume that those who do show up are punctual
Stochastic Outpatient Procedure Scheduling Problem (SOPSP)

A clinic manager must **schedule** the **start times** for a set of procedures, where

- Each procedure is of a **known type** and associated probability distribution of (non-negative) **random duration**
- Although patients may fail to show up to their appointments, we assume that those who do show up are **punctual**
- The **provider** is **always available** at the start of the day, and immediately after each procedure
- There is **no opportunity to modify the schedule** on the day of service
- The performance metric is the **weighted sum** of total patient **waiting time** cost, total provider **idle time** cost, and clinic **overtime**
A clinic manager must **schedule** the **start times** for a set of procedures, where

- Each procedure is of a **known type** and associated probability distribution of (non-negative) **random duration**
- Although patients may fail to show up to their appointments, we assume that those who do show up are **punctual**
- The **provider is always available** at the start of the day, and immediately after each procedure
- There is **no opportunity to modify the schedule** on the day of service
- The performance metric is the **weighted sum** of total patient **waiting time** cost, total provider **idle time** cost, and clinic **overtime**

A basic (yet still **challenging**) single-server stochastic appointment sequencing and scheduling (**SASS**) problem
So, We’ve Got Ourselves a Complex Appointment Scheduling Problem!!

What to do Next?
“Curse of Uncertainty"

Sample-Based Optimization and Mixed-Integer Programming (SMIP) to Rescue
SOPSP Complexity

Welch (1952)
Denton et al (2010)
Gupta and Denton (2008)
Mancilla and Storer (2012)
Gupta (2007)
Ahmadi-Javid et al (2017)
Berg et al (2014)
Weiss (1990)

Complex \{combinatorial, stochastic, multi-criteria\} Problem
SOPSP Complexity

Complex \{combinatorial, stochastic, multi-criteria\} Problem

History of SMIP for SOPSP
- Not easy to solve, require specially-developed algorithms/heuristics,...
SOPSP Complexity

History of SMIP for SOPSP

- Not easy to solve, require specially-developed algorithms/heuristics,....

What we need
SOPSP Complexity

History of SMIP for SOPSP

- Not easy to solve, require specially-developed algorithms/heuristics,....

What we need

- **Tractable SMIP**: can solve SOPSP instances of realistic sizes in an acceptable amount of time
SOPSP Complexity

History of SMIP for SOPSP

- Not easy to solve, require specially-developed algorithms/heuristics,....

What we need

- **Tractable SMIP**: can solve SOPSP instances of realistic sizes in an acceptable amount of time

- **Implementable SMIP**: can be easily translated into standard optimization software packages, not requiring customized algorithmic development or tuning
SOPSP Complexity

History of SMIP for SOPSP
- Not easy to solve, require specially-developed algorithms/heuristics,....

What we need
- **Tractable SMIP**: can solve SOPSP instances of realistic sizes in an acceptable amount of time
- **Implementable SMIP**: can be easily translated into standard optimization software packages, not requiring customized algorithmic development or tuning

**Sorry, Superman, it’s the #KnightsOfMIP business**
Subject to:  
Each procedure is assigned to one appointment  
Each appointment is assigned one procedure
Stochastic Mixed-Integer Program (SMIP) for SOPSP

Subject to:

\[
\sum_{i=1}^{P} x_{i,p} = 1 \quad \forall p \tag{2}
\]

\[
\sum_{p=1}^{P} x_{i,p} = 1 \quad \forall i \tag{3}
\]

\text{Actual start time}_i^n = \max \{\text{scheduled time}_i, \text{completion time}_{i-1}^n\}
Subject to:
\[ \sum_{i=1}^{P} x_{i,p} = 1 \quad \forall p \]  
\[ \sum_{p=1}^{P} x_{i,p} = 1 \quad \forall i \]  
\[ s_i^n \geq t_i \quad \forall (i, n) \]  
\[ s_i^n \geq s_{i-1}^n + \sum_{p=1}^{P} d_p \cdot x_{i-1,p} \quad \forall (i \geq 2, n) \]  

Idle Time\(_i^n\) = max \{actual start time\(_{i+1}^n\), completion time\(_i^n\)\}

Overtime\(_i^n\) = max \{completion time of last appt \(_P^n\) – scheduled closing time\}
Subject to:

\[ \sum_{i=1}^{P} x_{i,p} = 1 \quad \forall p \]  
\[ \sum_{p=1}^{P} x_{i,p} = 1 \quad \forall i \]  
\[ s_{i}^{n} \geq t_{i} \quad \forall (i,n) \]  
\[ s_{i}^{n} \geq s_{i-1}^{n} + \sum_{p=1}^{P} d_{p}^{m} \cdot x_{i-1,p} \quad \forall (i \geq 2,n) \]  
\[ g_{i}^{n} = s_{i+1}^{n} - \left( s_{i}^{n} + \sum_{p=1}^{P} d_{p}^{m} \cdot x_{i,p} \right) \quad \forall (i < P,n) \]  
\[ o^{n} \geq \left( s_{p}^{n} + \sum_{p=1}^{P} d_{p}^{m} \cdot x_{p,p} \right) - \mathcal{L} \quad \forall n \]  
\[ (g_{i}^{n}, s_{i}^{n}, o^{n}) \geq 0 \quad \forall (i,n) \]  
\[ x_{i,p} \in \{0,1\}, t_{i} \geq 0 \quad \forall (i,p) \]
Stochastic Mixed-Integer Program (SMIP) for SOPSP

\[ \text{SOPSP}(\Omega) \quad \text{minimize} \quad \mathbb{E} \left[ \lambda^w \cdot \text{Total Waiting} + \lambda^g \cdot \text{Total Idle} + \lambda^g \cdot \text{Overtime} \right] \]

Subject to:

\[ \sum_{i=1}^{P} x_{i,p} = 1 \quad \forall p \]  

\[ \sum_{p=1}^{P} x_{i,p} = 1 \quad \forall i \]  

\[ s_i^n \geq t_i \quad \forall (i,n) \]  

\[ s_i^n \geq s_{i-1}^n + \sum_{p=1}^{P} d_p^n \cdot x_{i-1,p} \quad \forall (i \geq 2,n) \]  

\[ g_i^n = s_{i+1}^n - \left( s_i^n + \sum_{p=1}^{P} d_p^n \cdot x_{i,p} \right) \quad \forall (i < P,n) \]  

\[ o^n \geq \left( s_P^n + \sum_{p=1}^{P} d_p^n \cdot x_{P,p} \right) - \mathcal{L} \quad \forall n \]  

\[ (g_i^n, s_i^n, o^n) \geq 0 \quad \forall (i,n) \]  

\[ x_{i,p} \in \{0,1\}, \quad t_i \geq 0 \quad \forall (i,p) \]
Stochastic Mixed-Integer Program (SMIP) for SOPSP

\[
\text{SOPSP}(N) \quad \text{minimize}\quad \frac{1}{N} \sum_{n=1}^{N} \left[ \sum_{i=1}^{P} \lambda_i^w \cdot (s_i^n - t_i) + \sum_{i=1}^{P} \lambda_i^g \cdot g_i^n + \lambda^o \cdot o^n \right]
\]  

(1)

Subject to:

\[
\sum_{i=1}^{P} x_{i,p} = 1 \quad \forall p
\]  

(2)

\[
\sum_{p=1}^{P} x_{i,p} = 1 \quad \forall i
\]  

(3)

\[
s_i^n \geq t_i \quad \forall (i, n)
\]  

(4)

\[
s_i^n \geq s_{i-1}^n + \sum_{p=1}^{P} d_{p}^n \cdot x_{i-1,p} \quad \forall (i \geq 2, n)
\]  

(5)

\[
g_i^n = s_{i+1}^n - \left( s_i^n + \sum_{p=1}^{P} d_{p}^n \cdot x_{i,p} \right) \quad \forall (i < P, n)
\]  

(6)

\[
o^n \geq \left( s_P^n + \sum_{p=1}^{P} d_{p}^n \cdot x_{P,p} \right) - \mathcal{L} \quad \forall n
\]  

(7)

\[
(g_i^n, s_i^n, o^n) \geq 0 \quad \forall (i, n)
\]  

(8)

\[
x_{i,p} \in \{0, 1\}, \quad t_i \geq 0 \quad \forall (i, p)
\]  

(9)
Theoretical Analysis of the SMIP for SOPSP

(1) **Sizes** of SOPSP Formulations$^2$

---

(1) Sizes of SOPSP Formulations

Table 1: Sizes of formulations of the SOPSP with \( P \) procedures and \( N \) scenarios

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td># Binary variables</td>
<td>( P^2 )</td>
<td>( P^2 )</td>
<td>( P^2 )</td>
</tr>
<tr>
<td># Continuous variables</td>
<td>( P + N(2P^2 + 2) )</td>
<td>( P + N(2P + 1) )</td>
<td></td>
</tr>
<tr>
<td># First-stage constraints</td>
<td>( P^2 + 3P )</td>
<td>( P^2 + 3P )</td>
<td></td>
</tr>
<tr>
<td># Second-stage constraints</td>
<td>( N(4P^2 + P + 2) )</td>
<td>( 5NP )</td>
<td></td>
</tr>
</tbody>
</table>
Theoretical Analysis of the SMIP for SOPSP

(1) Sizes of SOPSP Formulations

Table 1: Sizes of formulations of the SOPSP with $P$ procedures and $N$ scenarios

|                        | Mancilla et al. (2012) | Shehadeh et al. (201)
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$P^2$</td>
<td>$P^2$</td>
<td>$P^2$</td>
</tr>
<tr>
<td>$P + N(2P^2 + 2)$</td>
<td>$P + N(2P + 1)$</td>
<td></td>
</tr>
<tr>
<td>$P^2 + 3P$</td>
<td>$P^2 + 3P$</td>
<td></td>
</tr>
<tr>
<td>$N(4P^2 + P + 2)$</td>
<td>$5NP$</td>
<td></td>
</tr>
</tbody>
</table>

(Mancilla) minimize

$$\frac{1}{N} \sum_{n=1}^{N} \left[ \sum_{i=1}^{P} \sum_{p=1}^{P} \lambda_{w}^{n} \cdot w_{i,p}^{n} + \sum_{i=1}^{P} \sum_{p=1}^{P} \lambda_{g}^{n} \cdot g_{i,p}^{n} + \lambda_{o}^{n} \cdot o^{n} \right]$$

subject to

$$\sum_{i=1}^{P} x_{i,p} = 1 \quad \forall p$$

$$\sum_{p=1}^{P} x_{i,p} = 1 \quad \forall i$$

$$t_{i} - t_{i+1} - \sum_{p=1}^{P} w_{i+1,p}^{n} + \sum_{p=1}^{P} g_{i,p}^{n} + \sum_{p=1}^{P} w_{i,p}^{n} = - \sum_{p=1}^{P} d_{p}^{n} \cdot x_{i,p} \quad \forall (i < P, n)$$

$$t_{P} + \sum_{p=1}^{P} w_{P,p}^{n} - o^{n} + e^{n} = - \sum_{p=1}^{P} d_{p}^{n} \cdot x_{P,p} + \mathcal{L} \quad \forall n$$

$$w_{i,p}^{n} \leq M_{1}^{i} \cdot x_{i,p} \quad \forall (i, p, n)$$

$$g_{i,p}^{n} \leq M_{2}^{i} \cdot x_{i,p} \quad \forall (i, p, n)$$

$$(w_{i,p}^{n}, g_{i,p}^{n}, o^{n}, e^{n}) \geq 0 \quad \forall (i, p, n)$$

$$t_{i} \geq 0 \quad \forall i$$

$$x_{i,p} \in \{0, 1\} \quad \forall (i, p)$$
Theoretical Analysis of the SMIP for SOPSP

(1) Sizes of SOPSP Formulations

Table 1: Sizes of formulations of the SOPSP with $P$ procedures and $N$ scenarios

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td># Binary variables</td>
<td>$2P^2 + 4P + 2$</td>
<td>$P^2$</td>
<td>$P^2$</td>
</tr>
<tr>
<td># Continuous variables</td>
<td>$P + 1 + N(2P^2 + 4P + 4)$</td>
<td>$P + N(2P^2 + 2)$</td>
<td>$P + N(2P + 1)$</td>
</tr>
<tr>
<td># First-stage constraints</td>
<td>$P^3 + 5P^2 + 11P + 10$</td>
<td>$P^2 + 3P$</td>
<td>$P^2 + 3P$</td>
</tr>
<tr>
<td># Second-stage constraints</td>
<td>$N(4P^2 + 9P + 5)$</td>
<td>$N(4P^2 + P + 2)$</td>
<td>$5NP$</td>
</tr>
</tbody>
</table>

[*] An enhancement of Denton et al. (2007)
SMIP for SOPSP

Theoretical Analysis of the SMIP for SOPSP

Table 1: Sizes of formulations of the SOPSP with $P$ procedures and $N$ scenarios

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>(Berg) minimize</td>
<td>$\frac{1}{N} \sum_{n=1}^{N} \left[ \sum_{p=1}^{P+1} \sum_{p'=1}^{P} \lambda_{p,p'}^{n} \cdot A_{p,p'}^{n} + \sum_{p'=1}^{P+1} \sum_{p=1}^{P} \lambda_{p,p'}^{n} \cdot g_{p,p'}^{n} + \lambda^{n} \cdot \alpha^{n} \right]$</td>
<td>$P^2$</td>
<td>$P^2 + 3P$</td>
</tr>
<tr>
<td>subject to</td>
<td>$\sum_{p'=1}^{P+1} r_{p,p'} \leq 1$</td>
<td>$\forall p$</td>
<td>$5NP$</td>
</tr>
<tr>
<td></td>
<td>$\sum_{p=1}^{P+1} r_{p,p'} = N$</td>
<td>$\forall (p, p', i \leq P)$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$x_{i,p} + x_{i+1',p'} - 1 \leq r_{p,p'}$</td>
<td>$\forall p$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$\sum_{p=1}^{P+1} x_{p,i} = 1$</td>
<td>$\forall i$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$\sum_{p=1}^{P+1} r_{p} = 1$</td>
<td>$\forall i$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$\sum_{p=1}^{P+1} r_{p,p+1} = 1$</td>
<td>$\forall p$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$\sum_{p=1}^{P+1} r_{p+1,p} = 0$</td>
<td>$\forall p$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$x_{P+1,p} = 1$</td>
<td>$\forall (p, p', n)$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$w_{p,p'}^{n} \leq M_{1} r_{p,p'}$</td>
<td>$\forall (p, p', n)$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$g_{p,p'} \leq M_{2} r_{p,p'}$</td>
<td>$\forall (p, p', n)$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$- \sum_{p'=1}^{P+1} w_{p,p'}^{n} - \sum_{p=1}^{P+1} w_{p,p'}^{n} - \sum_{p'=1}^{P+1} g_{p,p'}^{n} = A_{p,p'}^{n} - y_{p}$</td>
<td>$\forall (p : p \leq P, n)$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$\sum_{p=1}^{P+1} \sum_{p'=1}^{P+1} g_{p,p'}^{n} - \alpha^{n} + \epsilon^{n} = N - \sum_{p=1}^{P+1} A_{p,p'}^{n}$</td>
<td>$\forall n$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$r_{p,p'}, x_{p,i} \in {0, 1}; y_{p} \geq 0$</td>
<td>$\forall (p, p', i)$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$w_{p,p'}^{n}, g_{p,p'}^{n}, \alpha^{n}, \epsilon^{n} \geq 0$</td>
<td>$\forall (p, p', n)$</td>
<td></td>
</tr>
</tbody>
</table>

Shehadeh, K.S., Cohn, A.

University of Michigan
Theoretical Analysis of the SMIP for SOPSP

(1) Our SMIP formulation is **smaller** ✓

(2) The **tightness** of the SOPSP Formulations

**Theorem 1.** Suppose $\lambda^w > 0$, and $\lambda^g > 0$ and/or $\lambda^o > 0$.

- The linear programming relaxation (LPR) of Shehadeh et al. and Mancilla et al. are **equivalent**.

---

Theoretical Analysis of the SMIP for SOPSP

(1) Our SMIP formulation is smaller ✓

(2) The tightness of the SOPSP Formulations

Theorem 1. Suppose $\lambda^w > 0$, and $\lambda^g > 0$ and/or $\lambda^o > 0$.

- The linear programming relaxation (LPR) of Shehadeh et al. and Mancilla et al. are equivalent.
- Furthermore, Shehadeh et al. is a tighter formulation than Berg et al. ✓
Do We Really Have a Good Model?
Is it Tractable?
Can We Implement it in Practice?
## Description of Experiments

- 14 different SOPSP instances

### Table 2: Characteristics of SOPSP instances.

<table>
<thead>
<tr>
<th>Instance</th>
<th># of Procedures</th>
<th># of Types</th>
<th>Procedures to be scheduled (by type)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1(^a)</td>
<td>4 procedures</td>
<td>3 types</td>
<td>(2A, 1C, 1J )</td>
</tr>
<tr>
<td>2(^a)</td>
<td>5 procedures</td>
<td>4 types</td>
<td>(2A, 1G, 1H,1J)</td>
</tr>
<tr>
<td>3(^a)</td>
<td>5 procedures</td>
<td>4 types</td>
<td>(1A, 1D,2G, 1J)</td>
</tr>
<tr>
<td>4(^a)</td>
<td>6 procedures</td>
<td>5 types</td>
<td>(1A, 1B, 1F, 2G, 1H)</td>
</tr>
<tr>
<td>5(^a)</td>
<td>7 procedures</td>
<td>5 types</td>
<td>(1C, 1D, 1F, 1H, 3J)</td>
</tr>
<tr>
<td>6(^a)</td>
<td>7 procedures</td>
<td>6 types</td>
<td>(1A, 1B, 1D, 1E, 2G, 1J)</td>
</tr>
<tr>
<td>7(^a)</td>
<td>10 procedures</td>
<td>6 types</td>
<td>(3A, 1C, 1D, 1G, 1I,3J)</td>
</tr>
<tr>
<td>8(^a)</td>
<td>10 procedures</td>
<td>6 types</td>
<td>(2A, 1B, 1D, 2G, 2I, 2J)</td>
</tr>
<tr>
<td>9(^b)</td>
<td>10 procedures</td>
<td>2 types</td>
<td>(6CP, 4CPU)</td>
</tr>
<tr>
<td>10(^a)</td>
<td>11 procedures</td>
<td>8 types</td>
<td>(2A, 1C, 2E, 1F, 1G, 1H, 2I, 1J)</td>
</tr>
<tr>
<td>11(^a)</td>
<td>11 procedures</td>
<td>6 types</td>
<td>(2A, 2F, 1G, 2H, 2I, 2J)</td>
</tr>
<tr>
<td>12(^c)</td>
<td>12 procedures</td>
<td>2 types</td>
<td>(9R, 3N)</td>
</tr>
<tr>
<td>13(^c)</td>
<td>16 procedures</td>
<td>2 types</td>
<td>(12R, 4N)</td>
</tr>
<tr>
<td>14(^c)</td>
<td>20 procedures</td>
<td>2 types</td>
<td>(15R, 5N)</td>
</tr>
</tbody>
</table>

\(a\) From the AIMMS-MOPTA 5th Optimization Modeling Competition

\(b\) From Berg et al. (2014)

\(c\) From Deceuninck et al. (2018)
Description of Experiments

- 14 different SOPSP instances

Table 3: Distribution information for procedure duration, by type.

<table>
<thead>
<tr>
<th>Procedure type</th>
<th>Mean</th>
<th>Variance</th>
<th>Distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>9.83</td>
<td>12.08</td>
<td>Lognormal</td>
</tr>
<tr>
<td>B</td>
<td>81.46</td>
<td>804.56</td>
<td>Normal</td>
</tr>
<tr>
<td>C</td>
<td>59.75</td>
<td>652.69</td>
<td>Lognormal</td>
</tr>
<tr>
<td>D</td>
<td>34.53</td>
<td>303.94</td>
<td>Lognormal</td>
</tr>
<tr>
<td>E</td>
<td>120.84</td>
<td>2.38e+3</td>
<td>Lognormal</td>
</tr>
<tr>
<td>F</td>
<td>47.76</td>
<td>232.06</td>
<td>Lognormal</td>
</tr>
<tr>
<td>G</td>
<td>43.94</td>
<td>469.86</td>
<td>Gamma</td>
</tr>
<tr>
<td>H</td>
<td>39.90</td>
<td>129.28</td>
<td>Lognormal</td>
</tr>
<tr>
<td>I</td>
<td>95.13</td>
<td>2.430e+3</td>
<td>Lognormal</td>
</tr>
<tr>
<td>J</td>
<td>19.51</td>
<td>99.36</td>
<td>Lognormal</td>
</tr>
<tr>
<td>CPU</td>
<td>12.05</td>
<td>188.57</td>
<td>Weibull</td>
</tr>
<tr>
<td>CP</td>
<td>30.96</td>
<td>58.75</td>
<td>Weibull</td>
</tr>
<tr>
<td>R</td>
<td>20.00</td>
<td>256.00</td>
<td>Lognormal</td>
</tr>
<tr>
<td>N</td>
<td>30.00</td>
<td>576.00</td>
<td>Lognormal</td>
</tr>
</tbody>
</table>
Description of Experiments

- 14 different SOPSP instances
- Three different sets of weights for the multi-criteria objective function
  
  1. \( \lambda^w = \lambda^g = \lambda^o \)
  2. \( \lambda^w = 1, \lambda^g = 0, \lambda^o = 10 \) (Berg et al., 2014)
  3. \( \lambda^w = 1, \lambda^g = 5, \lambda^o = 7.5 \) (Deceuninck et al., 2018)
Description of Experiments

- 14 different SOPSP instances
- Three different sets of weights for the multi-criteria objective function
  1. $\lambda^w = \lambda^g = \lambda^o$
  2. $\lambda^w = 1, \lambda^g = 0, \lambda^o = 10$ (Berg et al., 2014)
  3. $\lambda^w = 1, \lambda^g = 5, \lambda^o = 7.5$ (Deceuninck et al., 2018)
- For each of the 14 SOPSP instances and 3 sets of weights, we generated 10 sample average approximations, each with $N = 1,000$ scenarios
Description of Experiments

- 14 different SOPSP instances

- Three different sets of weights for the multi-criteria objective function
  
  (1) $\lambda^w = \lambda^g = \lambda^o$

  (2) $\lambda^w = 1, \lambda^g = 0, \lambda^o = 10$ (Berg et al., 2014)

  (3) $\lambda^w = 1, \lambda^g = 5, \lambda^o = 7.5$ (Deceuninck et al., 2018)

- 420 sample average approximations (SAA), each with $N = 1,000$ scenarios
Description of Experiments

- 14 different SOPSP instances
- Three different sets of weights for the multi-criteria objective function
  1. $\lambda^w = \lambda^g = \lambda^o$
  2. $\lambda^w = 1, \lambda^g = 0, \lambda^o = 10$ (Berg et al., 2014)
  3. $\lambda^w = 1, \lambda^g = 5, \lambda^o = 7.5$ (Deceuninck et al., 2018)
- 420 sample average approximations (SAA), each with $N = 1,000$ scenarios
- Symmetry-breaking constraints\(^3\)

\(^3\)Ostrowski et al. 2011, Berg et al. 2014
Description of Experiments

- 14 different SOPSP instances
- Three different sets of weights for the multi-criteria objective function
  1. \( \lambda^w = \lambda^g = \lambda^o \)
  2. \( \lambda^w = 1, \lambda^g = 0, \lambda^o = 10 \) (Berg et al., 2014)
  3. \( \lambda^w = 1, \lambda^g = 5, \lambda^o = 7.5 \) (Deceuninck et al., 2018)
- 420 sample average approximations (SAA), each with \( N = 1,000 \) scenarios
- Symmetry-breaking constraints\(^3\)

\[
    x_{i,p} - \sum_{j>i}^{P} x_{j,p+1} \leq 0, \quad \forall i = 1, \ldots, P, \quad \forall p: p, p + 1 \in P_q, \quad \forall q = 1, \ldots, Q,
\]

---

\(^3\) Ostrowski et al. 2011, Berg et al. 2014
Description of Experiments

- 14 different SOPSP instances
- Three different sets of weights for the multi-criteria objective function
  1. \( \lambda^w = \lambda^g = \lambda^o \)
  2. \( \lambda^w = 1, \lambda^g = 0, \lambda^o = 10 \) (Berg et al., 2014)
  3. \( \lambda^w = 1, \lambda^g = 5, \lambda^o = 7.5 \) (Deceuninck et al., 2018)
- 420 sample average approximations (SAA), each with \( N = 1,000 \) scenarios
- Symmetry-breaking constraints
  \[
  x_i,p - \sum_{j>i}^P x_{j,p+1} \leq 0, \quad \forall i = 1, \ldots, P, \forall p : p, p+1 \in P_q, \forall q = 1, \ldots, Q,
  \]
  e.g., procedure mix = (2 Type A, 1 Type B)
  
\[ A_1 \rightarrow B_1 \rightarrow A_2 \]
Description of Experiments

- 14 different SOPSP instances
- Three different sets of weights for the multi-criteria objective function
  1. $\lambda^w = \lambda^g = \lambda^o$
  2. $\lambda^w = 1, \lambda^g = 0, \lambda^o = 10$ (Berg et al., 2014)
  3. $\lambda^w = 1, \lambda^g = 5, \lambda^o = 7.5$ (Deceuninck et al., 2018)
- 420 sample average approximations (SAA), each with $N = 1,000$ scenarios
- Symmetry-breaking constraints
  \[
  x_{i,p} - \sum_{j>i}^P x_{j,p+1} \leq 0, \quad \forall i = 1, \ldots, P, \ \forall p : p, p+1 \in P_q, \ \forall q = 1, \ldots, Q,
  \]
  e.g., procedure mix = (2 Type A, 1 Type B)

\[
A_1 \rightarrow B_1 \rightarrow A_2 \iff A_2 \rightarrow B_1 \rightarrow A_1
\]
Description of Experiments

- 14 different SOPSP instances
- Three different sets of weights for the multi-criteria objective function
  1. $\lambda^w = \lambda^g = \lambda^o$
  2. $\lambda^w = 1, \lambda^g = 0, \lambda^o = 10$ (Berg et al., 2014)
  3. $\lambda^w = 1, \lambda^g = 5, \lambda^o = 7.5$ (Deceuninck et al., 2018)
- 420 sample average approximations (SAA), each with $N = 1,000$ scenarios
- Symmetry-breaking constraints\(^3\)
- Time limit: 2 hours
- Using a standard optimization modeling tool (AMPL), and a commercial MILP solver (CPLEX), with default settings

\(^3\)Ostrowski et al. 2011, Berg et al. 2014
Using our model, we were able to solve all of the 420 SAAs in less than 25 minutes

Table 4: Solution times (in seconds) using Shehadeh et al. model

<table>
<thead>
<tr>
<th>SOPS Instance</th>
<th>$\lambda^w = \lambda^g = \lambda^o$</th>
<th>$\lambda^w = 1, \lambda^g = 0, \lambda^o = 10$</th>
<th>$\lambda^w = 1, \lambda^g = 5, \lambda^o = 7.5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance</td>
<td>Min</td>
<td>Avg±stdv</td>
<td>Max</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>3 ±0.34</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>13±2</td>
<td>17</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>9±0.9</td>
<td>11</td>
</tr>
<tr>
<td>4</td>
<td>33</td>
<td>41±6</td>
<td>55</td>
</tr>
<tr>
<td>5</td>
<td>53</td>
<td>65±9</td>
<td>77</td>
</tr>
<tr>
<td>6</td>
<td>99</td>
<td>111±7</td>
<td>122</td>
</tr>
<tr>
<td>7</td>
<td>215</td>
<td>276±46</td>
<td>334</td>
</tr>
<tr>
<td>8</td>
<td>237</td>
<td>284±24</td>
<td>310</td>
</tr>
<tr>
<td>9</td>
<td>57</td>
<td>70±8</td>
<td>85</td>
</tr>
<tr>
<td>10</td>
<td>588</td>
<td>769±105</td>
<td>937</td>
</tr>
<tr>
<td>11</td>
<td>254</td>
<td>357±61</td>
<td>460</td>
</tr>
<tr>
<td>12</td>
<td>83</td>
<td>107±12</td>
<td>123</td>
</tr>
<tr>
<td>13</td>
<td>363</td>
<td>466±59</td>
<td>551</td>
</tr>
<tr>
<td>14</td>
<td>862</td>
<td>1218±164</td>
<td>1464</td>
</tr>
</tbody>
</table>
Comparison with Berg et al. (2014)

- Using Berg et al. model, we were able to solve 160 SAAs to optimality within the 2 hrs time limit.

- Berg takes from 6 to 138 times longer than our model to solve such instances.

- The remaining 260 SAAs that were not solved terminated with poor quality solutions.

- 180 SAAs terminated with a relative MIP gap \((UB - LB) / UB \times 100\%\) between 16\% and 70\%.

- 80 SAAs terminated without any feasible MIP solutions.

So, WHY?
Comparison with Berg et al. (2014)

- Using Berg et al. model, we were able to solve **160** SAAs to optimality within the 2 hrs time limit.

<p>| Table 5: Ratios of solution times of Berg et al. and Shehadeh et al. on SAAs solved by both. |
|-------------------------------------------------|-------------------------------------------------|-------------------------------------------------|
| $\lambda^w = \lambda^g = \lambda^o$ (a)       | $\lambda^w = 1, \lambda^g = 0, \lambda^o = 10$ (b) | $\lambda^w = 1, \lambda^g = 5, \lambda^o = 7.5$ (b) |</p>
<table>
<thead>
<tr>
<th>Min</th>
<th>Avg±stdv</th>
<th>Max</th>
<th>Min</th>
<th>Avg±stdv</th>
<th>Max</th>
<th>Min</th>
<th>Avg±stdv</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>31±29</td>
<td>116</td>
<td>4</td>
<td>33±27</td>
<td>107</td>
<td>8</td>
<td>51±35</td>
<td>138</td>
</tr>
</tbody>
</table>

[a] SOPS Instances 1–6, 10 SAA instances each.
[b] SOPS Instances 1–5, 10 SAA instances each.
Comparison with Berg et al. (2014)

- Using Berg et al. model, we were able to solve 160 SAAs to optimality within the 2 hrs time limit.
  - Berg takes from 6 to 138 times longer than our model to solve such instances
- The remaining 260 SAAs that were not solved terminated with poor quality solutions
Comparison with Berg et al. (2014)

- Using Berg et al. model, we were able to solve 160 SAAs to optimality within the 2 hrs time limit.
  - Berg takes from 6 to 138 times longer than our model to solve such instances
- The remaining 260 SAAs that were not solved terminated with poor quality solutions
  - 180 SAAs terminated with a relative MIP gap \( \left( \frac{UB - LB}{UB} \times 100\% \right) \) between 16% and 70%

<table>
<thead>
<tr>
<th>( \lambda^w = \lambda^g = \lambda^o ) (a)</th>
<th>( \lambda^w = 1, \lambda^g = 0, \lambda^o = 10 ) (b)</th>
<th>( \lambda^w = 1, \lambda^g = 5, \lambda^o = 7.5 ) (b)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min</td>
<td>Avg±stdv</td>
<td>Max</td>
</tr>
<tr>
<td>41%</td>
<td>54±0.08%</td>
<td>70%</td>
</tr>
</tbody>
</table>

[a] SOPS Instances 7–12, 10 SAA instances each. 
[b] SOPS Instances 6–11, 10 SAA instances each.
Comparison with Berg et al. (2014)

- Using Berg et al. model, we were able to solve 160 SAAs to optimality within the 2 hrs time limit.

  - Berg takes from 6 to 138 times longer than our model to solve such instances

- The remaining 260 SAAs that were not solved terminated with poor quality solutions

  - 180 SAAs terminated with a relative MIP gap ($\frac{UB-LB}{UB} \times 100\%$) between 16% and 70%
  - 80 SAAs terminated without any feasible MIP solutions
Comparison with Berg et al. (2014)

- Using Berg et al. model, we were able to solve 160 SAAs to optimality within the 2 hrs time limit.
  - Berg takes from 6 to 138 times longer than our model to solve such instances

- The remaining 260 SAAs that were not solved terminated with poor quality solutions
  - 180 SAAs terminated with a relative MIP gap \( \frac{UB - LB}{UB} \times 100\% \) between 16% and 70%
  - 80 SAAs terminated without any feasible MIP solutions

- So, WHY?

| Table 7: Ratios of optimal objective values of LP relaxations of Shehadeh et al. and Berg et al. |
|---------------------------------|---------------------------------|---------------------------------|
| \( \lambda^w = \lambda^g = \lambda^o \) | \( \lambda^w = 1, \lambda^g = 0, \lambda^o = 10 \) | \( \lambda^w = 1, \lambda^g = 5, \lambda^o = 7.5 \) |
| Min | Avg±stdv | Max | Min | Avg±stdv | Max | Min | Avg±stdv | Max |
| 1.95 | 2.62±0.41 | 3.48 | 1.11 | 1.38±0.26 | 2.08 | 1.27 | 1.64±0.33 | 2.49 |
Comparison with Mancilla et al. (2012)

- Using our model, we were able to solve all of the 420 SAAs in less than 25 minutes.

- Using Mancilla et al. model, we were able to solve 340 of the 420 SAAs to optimality within the two hour time limit.
Comparison with Mancilla et al. (2012)

- Using **our model**, we were **able to solve all of the 420 SAAs**
- Using **Mancilla et al.** model, we were **able to solve 340** of the 420 SAAs to optimality within the two hour time limit.

Table 8: Comparison of performance of Mancilla et al. and Shehadeh et al. on SAAs solved by both

<table>
<thead>
<tr>
<th>Ratio</th>
<th>$\lambda^w = \lambda^g = \lambda^o$</th>
<th>$\lambda^w = 1, \lambda^g = 0, \lambda^o = 10$</th>
<th>$\lambda^w = 1, \lambda^g = 5, \lambda^o = 7.5$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Min</td>
<td>Avg±stdv</td>
<td>Max</td>
</tr>
<tr>
<td>(M) sol. time</td>
<td>1.2</td>
<td>7±4</td>
<td>21</td>
</tr>
<tr>
<td>(S) sol. time</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Comparison with Mancilla et al. (2012)

- Using our model, we were able to solve all of the 420 SAAs.
- Using Mancilla et al. model, we were able to solve 340 of the 420 SAAs to optimality within the two hour time limit.

Table 8: Comparison of performance of Mancilla et al. and Shehadeh et al. on SAAs solved by both

<table>
<thead>
<tr>
<th>Ratio</th>
<th>$\lambda^w = \lambda^g = \lambda^o$</th>
<th>$\lambda^w = 1, \lambda^g = 0, \lambda^o = 10$</th>
<th>$\lambda^w = 1, \lambda^g = 5, \lambda^o = 7.5$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Min</td>
<td>Avg±stdv</td>
<td>Max</td>
</tr>
<tr>
<td>(M) sol. time</td>
<td>1.2</td>
<td>7±4</td>
<td>21</td>
</tr>
<tr>
<td>(S) sol. time</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(M) nodes</td>
<td>0.5</td>
<td>1±0.2</td>
<td>1.4</td>
</tr>
<tr>
<td>(S) nodes</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Comparison with Mancilla et al. (2012)

- Using **our model**, we were able to solve all of the 420 SAAs.
- Using **Mancilla et al.** model, we were able to solve 340 of the 420 SAAs to optimality within the two hour time limit.

<table>
<thead>
<tr>
<th>Ratio</th>
<th>$\lambda^w = \lambda^g = \lambda^o$</th>
<th>$\lambda^w = 1, \lambda^g = 0, \lambda^o = 10$</th>
<th>$\lambda^w = 1, \lambda^g = 5, \lambda^o = 7.5$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Min</td>
<td>Avg±stdv</td>
<td>Max</td>
</tr>
<tr>
<td>(M) sol. time</td>
<td>1.2</td>
<td>7±4</td>
<td>21</td>
</tr>
<tr>
<td>(S) sol. time</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(M) nodes</td>
<td>0.5</td>
<td>1±0.2</td>
<td>1.4</td>
</tr>
<tr>
<td>(S) nodes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(M) iterations</td>
<td>1</td>
<td>11±15</td>
<td>119</td>
</tr>
<tr>
<td>(S) iterations</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Comparison with Mancilla et al. (2012)

- Using our model, we were able to solve all of the 420 SAAs.
- Using Mancilla et al. model, we were able to solve 340 of the 420 SAAs to optimality within the two hour time limit.

<table>
<thead>
<tr>
<th>Ratio</th>
<th>$\lambda^w = \lambda^g = \lambda^o$</th>
<th>$\lambda^w = 1, \lambda^g = 0, \lambda^o = 10$</th>
<th>$\lambda^w = 1, \lambda^g = 5, \lambda^o = 7.5$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Min</td>
<td>Avg±stdv</td>
<td>Max</td>
</tr>
<tr>
<td>(M) sol. time</td>
<td>1.2</td>
<td>7±4</td>
<td>21</td>
</tr>
<tr>
<td>(S) sol. time</td>
<td>0.5</td>
<td>1±0.2</td>
<td>1.4</td>
</tr>
<tr>
<td>(M) nodes</td>
<td>0.5</td>
<td>1±0.2</td>
<td>1.4</td>
</tr>
<tr>
<td>(S) nodes</td>
<td>1</td>
<td>11±15</td>
<td>119</td>
</tr>
<tr>
<td>(M) iterations</td>
<td>1</td>
<td>11±15</td>
<td>119</td>
</tr>
<tr>
<td>(S) iterations</td>
<td>1</td>
<td>11±15</td>
<td>119</td>
</tr>
</tbody>
</table>

- The remaining 80 SAAs that were not solved by Mancilla et al. terminated with relative MIP gaps between 15% and 25%.
Do We Really Have a Good Model? YES ✓
Is it Tractable? YES ✓
Can We Implement it in Practice? YES ✓

- Scheduling of surgeries in an operating room
- Scheduling of ships in a port
- Scheduling of exams in an examination facility, and more....
Conclusion

★ Outpatient colonoscopy scheduling is challenging primarily due to uncertainty in procedure time

★ We developed a new SMIP model for the Stochastic Outpatient Procedure Scheduling Problem
  – a basic (yet still challenging) offline SASS problem
Conclusion

★ Outpatient colonoscopy scheduling is challenging primarily due to uncertainty in procedure time

★ We developed a new SMIP model for the Stochastic Outpatient Procedure Scheduling Problem
  – a basic (yet still challenging) offline SASS problem

★ We compared our model with the existing models and analyzed them both empirically and theoretically
Conclusion

★ Outpatient colonoscopy scheduling is challenging primarily due to uncertainty in procedure time

★ We developed a new SMIP model for the Stochastic Outpatient Procedure Scheduling Problem
   – a basic (yet still challenging) offline SASS problem

★ We compared our model with the existing models and analyzed them both empirically and theoretically

★ In addition to empirical tractability, our model have the advantage of being implementable
   - necessary for optimization-based decision support tools to gain broad adoption in outpatient clinics (and other service-providing industry)
Conclusion

★ Outpatient colonoscopy scheduling is challenging primarily due to uncertainty in procedure time

★ We developed a new SMIP model for the Stochastic Outpatient Procedure Scheduling Problem
  – a basic (yet still challenging) offline SASS problem

★ We compared our model with the existing models and analyzed them both empirically and theoretically

★ In addition to empirical tractability, our model have the advantage of being implementable
  - necessary for optimization-based decision support tools to gain broad adoption in outpatient clinics (and other service-providing industry)

.... And we SOLVED SASS
Absolutely Not! The Rest is Still Coming

⋆ Karmel Shehadeh, Amy Cohn, Ruiwei Jiang. A Stochastic Programming Approach for Appointment Scheduling with Heterogenous and Random Arrivals
Absolutely Not! The Rest is Still Coming

⋆ Karmel Shehadeh, Amy Cohn, Ruiwei Jiang. A Stochastic Programming Approach for Appointment Scheduling with Heterogenous and Random Arrivals

- Proposition 1. Appointment order policy provides an upper bound

- Proposition 2. Rescheduling under perfect information provides a lower bound.
Absolutely Not! The Rest is Still Coming

- Proposition 1. Appointment order policy provides an upper bound.
- Proposition 2. Rescheduling under perfect information provides a lower bound.
Absolutely Not! The Rest is Still Coming

- Proposition 1. Appointment order policy provides an upper bound
- Proposition 2. Rescheduling under perfect information provides a lower bound.

All in all, we developed tractable operations research frameworks for stochastic outpatient appointment scheduling
And for Colonoscopy Scheduling at Michigan Medicine
And for Colonoscopy Scheduling at Michigan Medicine

Well,

Figure 1: Variability of colonoscopy duration with (a) adequately prep, and (b) inadequate prep (2013-2017)
And for Colonoscopy Scheduling at Michigan Medicine

Well,

Figure 1: Variability of colonoscopy duration with (a) adequately prep, and (b) inadequate prep (2013-2017)

Shehadeh et al. A Data-Driven Distributionally Robust Optimization (DRO) Approach for Colonoscopy Scheduling
“The theory of sequencing and scheduling, more than any other area in operations research, is characterized by a virtually unlimited number of problem types.”

Thank You!

THE QUESTION MARK

IS IT ALWAYS SO UNCERTAIN?
I'M SO GLAD YOU ASKED.

Karmel Shehadeh
Ksheha@umich.edu

Professor Amy Cohn
amycohn@umich.edu
We interpret the randomness in bowel prep-adequacy by a 0-1 Bernoulli random variable, $q_p$

$$d_p = q_p \cdot d^A + (1 - q_p) \cdot d^I$$
A Data-Driven DRO Approach for Colonoscopy Scheduling

★ We interpret the randomness in bowel prep-adequacy by a 0-1 Bernoulli random variable, $q_p$

$$d_p = q_p \cdot d^A + (1 - q_p) \cdot d^I$$

★ Accordingly, we can assume that the planner has access to (or knows) the lower and upper bounds of procedure duration and arrival time

$$S^A := \{d^A \geq 0 : d^A_{iL} \leq d^A_i \leq d^A_{iU}, \ \forall i \in [P], \ d^A_{P+1} = 0\}, \quad S^I := \{d^I \geq 0 : d^I_{iL} \leq d^I_i \leq d^I_{iU}, \ \forall i \in [P]\}$$

$$S^u := \{u : u^L_i \leq u \leq u^U_i, \ i \in [P], \ d^I_{P+1} = 0, \ u_{P+1} = 0\}, \quad S^q : \{q : q \in \{0,1\}^P\}$$
A Data-Driven DRO Approach for Colonoscopy Scheduling

★ We interpret the randomness in bowel prep-adequacy by a 0-1 Bernoulli random variable, $q_p$

$$d_p = q_p \cdot d^A + (1 - q_p) \cdot d^I$$

★ Accordingly, we can assume that the planner has access to (or knows) the lower and upper bounds of procedure duration and arrival time

★ Therefore, we define the ambiguity set $\mathcal{F}(S, \mu^q, \mu^A, \mu^I, \mu^u)$
A Data-Driven DRO Approach for Colonoscopy Scheduling

★ We interpret the randomness in bowel prep-adequacy by a 0-1 Bernoulli random variable, \( q_p \)

\[
d_p = q_p \cdot d^A + (1 - q_p) \cdot d^I
\]

★ Accordingly, we can assume that the planner has access to (or knows) the lower and upper bounds of procedure duration and arrival time

★ Therefore, we define the ambiguity set \( \mathcal{F}(S, \mu^q, \mu^A, \mu^I, \mu^u) \)

★ With expectation as a risk measure

\[
\min_{x \in X, t \in T} \sup_{\mathbb{P} \in \mathcal{F}(S, \mu^q, \mu^A, \mu^I, \mu^u)} \mathbb{E}_{\mathbb{P}}[Q(x, t, q, d^A, d^I, u)]
\]
A Data-Driven DRO Approach for Colonoscopy Scheduling

★ We interpret the randomness in bowel prep-adequacy by a 0-1 Bernoulli random variable, $q_p$

$$d_p = q_p \cdot d^A + (1 - q_p) \cdot d^I$$

★ Accordingly, we can assume that the planner has access to (or knows) the lower and upper bounds of procedure duration and arrival time

★ Therefore, we define the ambiguity set $\mathcal{F}(S, \mu^q, \mu^A, \mu^I, \mu^u)$

★ With expectation as a risk measure

$$\min_{x \in \mathcal{X}, t \in \mathcal{T}} \sup_{\mathbb{P} \in \mathcal{F}(S, \mu^q, \mu^A, \mu^I, \mu^u)} \mathbb{E}_{\mathbb{P}}[Q(x, t, q, d^A, d^I, u)]$$

★ OR Journey: NMIP $\Rightarrow$ IP $\Rightarrow$ LP