Robust Optimization of MILPs under Decision–Dependent Uncertainty, with an application to Scheduling

Dr. Robin Vujanic

Rio Tinto Centre for Mine Automation

May 12, 2016
Background

- B.Sc. and M.Sc. in Mechanical Engineering
  ETH Zurich and Georgia Tech

- Ph.D. in Control Systems and Automation
  Prof. Manfred Morari
  Automatic Control Lab (Institut für Automatik)
  Electrical Engineering Dept.

- **topic**: approximation schemes in mixed–integer optimization

- **application domains**: power systems, supply chains
Co-Authors / Collaborators

Prof. Manfred Morari
Automatic Control Lab
ETH Zurich

Prof. Paul Goulart
Control Engineering
University of Oxford

Dr. Peyman M. Esfahani
Risk Analytics and Optimization
EPFL Lausanne

Prof. Sébastien Mariéthoz
Institute for Energy and Mobility Research
Bern University
Mixed–Integer Optimization

• many practical and industrial systems entail continuous quantities
  • physical measurements of voltages
  • concentrations
  • positions in space

as well as discrete components
  • on/off decisions
  • switches
  • logic reasoning (if, or, …)
  • scheduling assignments

• when the associated control/operation tasks are addressed using optimization, mixed-integer optimization problems (MIPs) arise

• generally computationally hard

• today: MIPs affected by uncertainty, and robust approaches to address them
Outline

Uncertain Problem Considered and its Robust Counterpart

Application to Scheduling under Uncertainty

Ongoing and Future Projects
Outline

Uncertain Problem Considered and its Robust Counterpart

Application to Scheduling under Uncertainty

Ongoing and Future Projects
Primer on Robust Optimization

• obtain an **uncertain optimization problem** (UP), integrating a nominal model and a characterization of the uncertainty affecting it

• derive *another* optimization problem, the **robust counterpart** (RC)
  • deterministic
  • its solutions remain feasible in UP for any possible uncertainty realization
  • and attain the “best objective”

• nowadays RCs are known for a number of important cases
  • computational tractability remains an issue in the non–convex case

• today: a new uncertainty structure, its RC and how it can be used
Uncertain Problem Considered

- consider the uncertain mixed-integer linear program

\[
\begin{align*}
\text{UP} : \quad & \min & c_x^\top x + c_y^\top y \\
\text{subject to} & & Ax + By + Dw \leq b \quad w \in \mathcal{W}(x) \\
& & x \in \{0, 1\}^n
\end{align*}
\]

- \(x\) boolean, \(y\) continuous
- \((A, B, b, c)\) is deterministic; \(\text{UP}\) encodes uncertainties directly affecting \(x\) (we will see how)
- \(\mathcal{W} : \mathbb{R}^{nx} \rightsquigarrow \mathbb{R}^{nw}\) is the set–valued map

\[
\mathcal{W}(x) = \bigoplus_{k=1}^{nx} x_k \cdot \mathcal{W}_k,
\]

\(x_k\) is \(k\)-th component of \(x\), \(\mathcal{W}_k \subseteq \mathbb{R}^{nw}\) polyhedral
- our goal: find the robust counterpart to \(\text{UP}\)
Robust Optimization – Affine Recourse Model

• we wish to robustify \( x \) (boolean) “statically”
  • independently of the uncertainty outcome, \( x \) should remain unchanged and feasible
• and allow for recourse on \( y \) (continuous)
• in the scheduling example, this will mean that the core of the schedule (assignments) is immune to uncertainty (preventive), but batchsizes can be regulated as uncertainty is revealed (reactive)
• ideally, we want an optimal recourse policy (e.g., using DP), but this is computationally intractable
• use affine policy model instead, see AARCS (Nemirovski 2009)

\[
y = Yw + v \quad w \in \mathcal{W}(\cdot)
\]

• causality \( \rightarrow \) structured \( Y \)
Robust Counterpart (1/2)

**Theorem:** the explicit robust counterpart to UP under affine recourse

\[ y = Yw + v \]

is

\[
\begin{aligned}
\text{min}_{x,v,Y,\Phi,\Psi} & \quad c_x^T x + c_y^T v \\
\text{subject to} & \quad Ax + Bv + \Phi 1^{n_x} \leq b \\
& \quad 1^{n_k \times m} \cdot \text{diag}(\psi_k) \geq [(BY + D)W_k]^T \\
& \quad 0 \leq \phi_k \leq x_k \psi_k \\
& \quad 0 \leq \psi_k - \phi_k \leq (1 - x_k) \psi_k \\
& \quad x \in \{0,1\}^{n_x},
\end{aligned}
\]

RCP:

with new variables \( \Phi, \Psi \in \mathbb{R}^{m \times n_x} \) and a constant \( \overline{\psi}_k \in \mathbb{R}^m \) such that

\[(BY + D)w \leq \overline{\psi}_k \quad \forall w \in \mathcal{W}_k, \forall k = 1, \ldots, n_x.\]

**Input:**
- nominal problem data, uncertainty sets \( \mathcal{W}_k \)

**Output:**
- \( x,v \), robust solutions
- \( Y \), matrix telling us how to modify \( v \) when \( w \neq 0 \)
Robust Counterpart (2/2)

\[
\begin{align*}
\text{min}_{x,v,Y,\Phi,\Psi} \quad & c_x^\top x + c_y^\top v \\
\text{subject to} \quad & Ax + Bv + \Phi 1^{nx} \leq b \\
& 1^{nk \times m} \cdot \text{diag}(\psi_k) \geq [(BY + D)W_k]^\top \\
& 0 \leq \phi_k \leq x_k \psi_k \\
& 0 \leq \psi_k - \phi_k \leq (1 - x_k)\overline{\psi}_k \\
x \in \{0, 1\}^{nx},
\end{align*}
\]

- RCP is still a MILP
- same number of integers as UP
- its size can be further reduced in many situations
- simpler result for \( Y = 0 \)
- can be combined with traditional approaches of robust optimization
Outline

Uncertain Problem Considered and its Robust Counterpart

Application to Scheduling under Uncertainty

Ongoing and Future Projects
Nominal Model: State–Task–Networks

- ○ = states (material quantities)
- □ = tasks

Application to Scheduling under Uncertainty
Explanations

- given: a set of physical units, a set of tasks that the units can perform and which transform states (material quantities), and a production network
- e.g., reactors 1 and 2 may perform reactions 1, 2, and 3; reaction 2 transforms a mix of states “hot A” and “intermediate BC” into the final product 1 and the intermediate state AB
- the scheduling problem is to establish what unit to assign to which task and at what time, in order to maximize a given performance index
STN Core Model

Variables:
- \( x_{ijt} \in \{0, 1\} \) is 1 if unit \( i \) starts processing task \( j \) at time step \( t \)
- \( y_{ijt}^{\text{batch}}, y_{st}^{\text{state}} \in \mathbb{R} \) batchsize and state (material) amount

Constraints (core):
- unique unit allocation
  \[
  \sum_{j \in \mathcal{J}_i} x_{ijt} \leq 1 \\
  \sum_{j \in \mathcal{J}_i} \sum_{t'=t}^{t+P_j-1} x_{ijt'} - 1 \leq M_{ij} (1 - x_{ijt}) \quad \forall i, t, j \in \mathcal{J}_i
  \]
- capacity of processing units and storages
  \[
  x_{ijt} \cdot V_{ij}^{\min} \leq y_{ijt}^{\text{batch}} \leq x_{ijt} \cdot V_{ij}^{\max} \quad \forall j, t, i \in \mathcal{I}_j
  \]
  \[
  0 \leq y_{st}^{\text{state}} \leq C_s \quad \forall s, t
  \]
- states update equations
  \[
  y_{st}^{\text{state}} = y_{s,t-1}^{\text{state}} + \sum_{j \in \mathcal{J}_s} \tilde{\rho}_{js} \sum_{i \in \mathcal{I}_j} y_{ij,t-P_j-1}^{\text{batch}} - \sum_{j \in \mathcal{J}_s} \rho_{js} \sum_{i \in \mathcal{I}_j} y_{ijt}^{\text{batch}} + R_{st} - D_{st}
  \]
- actual model may contain several “addon constraints”
Remarks on STNs

- generic model for scheduling batch production networks
- STNs and RTNs are very common in practice
- presented the discrete–time form
- continuous time formulation also exists (horrible eqns)
- “used to be” computationally intensive
Scheduling Uncertainty

- a number of sources of uncertainty affect schedules
  - time delays
  - unit malfunctioning or outage
  - natural events
  - human errors
  - unexpected changes in production requirements
  - . . .

- these have a substantial impact on the usefulness of a schedule
- issue is exacerbated by the fact that optimization routines tend to “pack” tasks at times when, e.g., prices are lowest
- how to handle?
Handling Uncertainty

Option 1:
- recompute a new schedule when event occurs (reactive)
- impractical, since the new schedule can be completely different

Option 2:
- use robust optimization to obtain flexible schedules
- these can support uncertain events without being disrupted
- and (optionally) allow for reactivity on $y$
  - stockpile size is regulated according to revealed uncertain events
  - while remaining within constraints
Modelling the Uncertainty – Example

• suppose that $x^* = [0, 1, 0, 0]^\top$
• however, at planning time we do not know whether we will actually be able to implement $x^*$
• e.g., a delay may result in

$$\hat{x} = x^* + w = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ -1 \\ 1 \\ 0 \end{bmatrix}$$

• clearly, the value of $w$ depends on the choice of $x$; and $w = [0, -1, 1, 0]$ should only be active with $x^* = [0, 1, 0, 0] \rightarrow \mathcal{W}(x)$
• on more complex decision problems, such a construction allows one to encode a rich variety of uncertain events
• quantification from existing data conceptually easy: check the difference between plan and actual execution to construct $w$
1. heating delay by one hour
2. execution of reaction 2 may be swapped from reactor 1 to reactor 2, but only after the first four hours
Heating Delay – Nominal vs Robust

Nominal Obj.: 2744.4, Robust Obj.: 2744.4

Heater

reactor 1

reactor 2

Separation

Application to Scheduling under Uncertainty
Heating Delay – Delay Occurred

Application to Scheduling under Uncertainty
# Unit Swap – Nominal vs Robust

Nominal Obj.: 2744.4, Robust Obj.: 2513.8

<table>
<thead>
<tr>
<th>Time (t)</th>
<th>Heater</th>
<th>Reactor 1</th>
<th>Reactor 2</th>
<th>Separation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>H. 52</td>
<td>Rea.1 76</td>
<td>Rea.1 50</td>
<td>Sep. 50</td>
</tr>
<tr>
<td>2</td>
<td>H. 32</td>
<td>Rea.2 80</td>
<td>Rea.2 50</td>
<td>Sep. 114</td>
</tr>
<tr>
<td>4</td>
<td>H. 4</td>
<td>Rea.2 80</td>
<td>Rea.3 50</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>H. 12</td>
<td>Rea.2 80</td>
<td>Rea.3 14</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>Rea.3 78</td>
<td>Rea.3 50</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>Rea.3 80</td>
<td>Rea.3 50</td>
<td></td>
</tr>
</tbody>
</table>

Application to Scheduling under Uncertainty 23
Further Remarks and Summary

The proposed method:

- is an approach to robust optimization that combines a preventive and a reactive action
- can be used to generate flexible schedules
- allows the incorporation of a rich variety of events
- is more broadly applicable
  - not tied to any specific feature of STNs; application to other scheduling systems possible
  - may be of use outside of scheduling altogether
- is computationally OK as far as we can tell
  - solve times for the STN examples: 0.4s for P, 1.1sec for RCP
- can be combined with existing techniques of robust optimization
Outline

Uncertain Problem Considered and its Robust Counterpart

Application to Scheduling under Uncertainty

Ongoing and Future Projects
Ongoing and Future Projects – Applications

- application of RO to scheduling models related to mining
- optimization of the energy management for large industrial loads
- “data driven optimization”
“Decomposition Methods for Large–Scale Non-Convex Models”:

- previous work on large scale instances coupled through constraints (e.g., shared resources)
- applications: power systems control, supply chains (portfolio optimization)
- new: models coupled through variables (e.g., SIPs)
- new insights on the tightness of decompositions based on duality
- use of techniques based on ergodic sequences (averaging) for primal recovery
- use of new first-order methods (Nesterov)
Thank you for your attention!

Questions?
Counterexample

encode possible delay of 1 unit by assuming task A is length 2

the algorithm produces...