Optimisation and uncertainty: comparing stochastic and robust programming

THIBAUT CUVELIER
UNIVERSITÉ DE LIÈGE
Why uncertainty in optimisation?

- How to make the best decisions? Optimisation!
- Example: schedule deliveries

What if there is unexpected congestion?
  - Penalties when deliveries are late
  - Drivers working overtime

Source: ArcGIS documentation
Table of contents

- Modelling approaches
  - Stochastic
  - Robust

- Applications
  - Facility location
  - Unit-commitment
  - Dam management
Modelling: stochastic approach

• First idea: probabilities, statistics
  ◦ Common tool with uncertainty!

• Hence: **stochastic programming**
  ◦ Best solution **on average**
  ◦ Use probabilities and scenarios

• Recourse action:
  ◦ In the future, **adapt** your decisions to what you have seen
  ◦ Uncertainty partly revealed (second stage)
Modelling: robust approach

• How to deal with the curse of dimensionality?
• Second idea: have the best solution in the worst case

• Worst case defined through an uncertainty set
  ◦ For example: confidence intervals
  ◦ Optimise within a set of values
  ◦ Ensure the solution works for all values
Application: Facility Location

• Where to place a hospital?
  ◦ Demand to meet — with uncertainty!
  ◦ Number of possible locations
Maximum instance size

• Compare the various uncertainty models
  ◦ And the way to solve them

• Maximum problem size under constraints?
  ◦ Number of instances that could be solved
  ◦ 10.5GB of RAM, 4 hours
Time to convergence

• Average time until solved on the same instances

• Results?
  ◦ Best? Average or robust: a few seconds at most!
  ◦ Different algorithms have various properties, but are not always able to solve large instances!
Application: Unit-Commitment

• How to meet some electrical demand?
  ◦ Choose which power plants to use
  ◦ Uncertainty lies in the demand

  ◦ Under constraints:
    ◦ Time to start, to stop
    ◦ Minimum up/down time

• Two-stage model:
  ◦ Plan today (first stage)
  ◦ Prepare for tomorrow (second stage)
A look at uncertainty

• Stochastic scenarios and uncertainty set:
  ◦ Intervals surround the scenarios

• Test case: 40 machines, 48-hour first stage, 48-hour second stage
Robustness comparison

• Compare solutions based on two criteria:
  ◦ **Stochastic**: objective value (average on scenarios)
  ◦ **Robust**: maximum interval size

• First test: robust criterion
  ◦ Robust: much better!
  ◦ Interval size larger than asked

• What about cost?
  ◦ Difference below one percent!
Robustness comparison

• What about failures?
  ◦ Stochastic: great impact
    ◦ Perceptibly better
  ◦ Robust: similar to previous

• Moving the recourse sooner?
  ◦ Plan for six hours (instead of two days)
  ◦ Same conclusions: stochastic fits better
Robustness comparison: Monte Carlo analysis

• Methodology to evaluate a solution:
  ◦ Draw scenarios (95% chance within interval)
  ◦ Evaluate the solution: check feasibility

• Robust feasible much more often!
Application: Dam Management

• Belgian dams’ main role: provide drinking water

• Uncertainty in the inflows, i.e. rain

• How to ensure no shortage?
  ◦ Lower bound for water depth
  ◦ If lower than this: no more guaranteed

• Currently used: computed once with data from the 1970s
Current rule too conservative?

• Stochastic: not so smooth
• Robust: very conservative

Still work in progress!
Concluding remarks

• About robustness of solutions:
  ◦ Robust programming? Most conservative
  ◦ Stochastic programming? Risk of over-fitting

• About performance of solving:
  ◦ Robust programming: little impact
  ◦ Stochastic programming: quickly intractable
    ◦ Dedicated algorithm: not to speed up!
And so?

• Each has its strengths!
  ◦ Provides different insights into the properties of the solution
  ◦ Not incompatible within a given model

• Little research comparing those approaches
  ◦ Nor a common modelling layer!

• Going further:
  ◦ Look at other problems
  ◦ Test different uncertainty sets
Back-up
Scenario tree

$t = t_0$

$t = t_r$
Ellipsoids
Solving stochastic programs

- Optimisation can be represented as a matrix

- Two-stage:

  \[
  \begin{array}{ccc}
  \text{T}_1 & \text{U}_1 \\
  \text{T}_2 & \text{U}_2 \\
  \vdots & \vdots \\
  \text{T}_s & \text{U}_s
  \end{array}
  \]

- Exploit this special structure!
Benders’ decomposition

• Decompose this matrix along stages

• Consider only what happens now
  ◦ And retrieve information from the future iteratively
Progressive hedging

• Decompose this matrix along scenarios

• Optimal solution for each scenario
  ◦ Then bring them together

\[
\begin{bmatrix}
T_1 & U_1 \\
T_2 & U_2 \\
\ldots & \ldots \\
T_s & U_s \\
\end{bmatrix}
\]

Right now

Evaluate in the future