Recent Advances in Solving Multistage Stochastic Mixed-integer Programs

Merve Bodur

University of Edinburgh

School of Mathematics

Outline

Introduction

Deterministic Equivalent

Tree Decomposition based Bounds

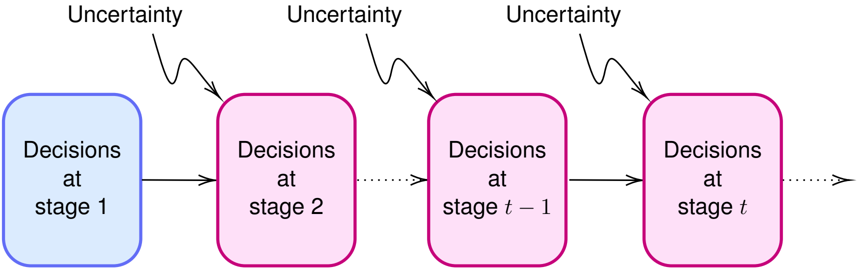
Exact Approaches

Partially Extended Formulations

Policy Development
- Aggregation Policies
- Decision Rules

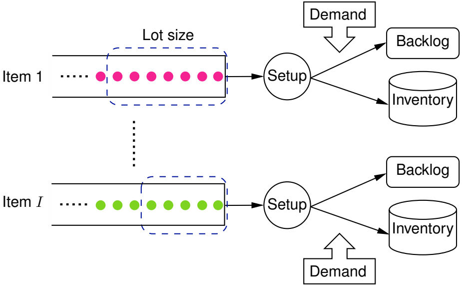
Conclusion

Sequential Decision-making Under Uncertainty

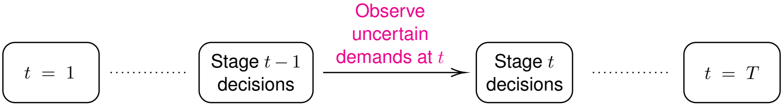


- Uncertainty is gradually observed
- Decisions are dynamically adapted to:
  - Observed uncertainty
  - Previous decisions

Example: Production Planning



Multi-item multi-period multi-stage lot-sizing problem under uncertainty

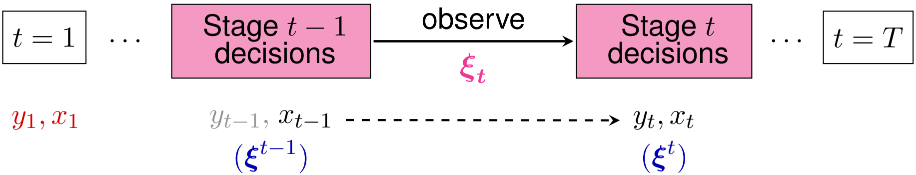


production levels	production levels$_{t-1}$	production levels$_t$	production levels$_T$
setups	setups$_{t-1}$	setups$_t$	setups$_T$
inventory levels	inventory levels$_{t-1}$	inventory levels$_t$	inventory levels$_T$
backlogs	backlogs$_{t-1}$	backlogs$_t$	backlogs$_T$

Multi-stage Stochastic Programs

Finite-horizon sequential decision making problems under uncertainty

- $T \geq 2$ decision stages $\qquad [T] := \{1, 2, \ldots, T\}$
- Stochastic process: $\{\boldsymbol{\xi}_t\}_{t=1}^T$
- History of the process: $\boldsymbol{\xi}^t := (\xi_1, \ldots, \boldsymbol{\xi}_t)$
- Dynamics:



- Decision variables: (nonanticipative)
  - ◇ State variables: $x_t(\boldsymbol{\xi}^t)$
  - ◇ Recourse (stage) variables: $y_t(\boldsymbol{\xi}^t)$
- For convenience: ξ_1 is constant (i.e., deterministic first stage)

MS(I)LP Formulation

- Uncertainty: $\{\boldsymbol{\xi}_t\}_{t\in[T]}$ has probability distribution $\mathbb{P}$ and support Ξ
- Decision variables: $y_t(\boldsymbol{\xi}^t) \in \mathbb{R}^{n_t},\ x_t(\boldsymbol{\xi}^t) \in \mathbb{R}^{d_t}$
- Objective:

$$\min\ \mathbb{E}_{\xi^T}\Big[\sum_{t\in[T]} c_t(\xi^t)^\top y_t(\boldsymbol{\xi}^t) + h_t(\xi^t)^\top x_t(\boldsymbol{\xi}^t)\Big]$$

- Constraints: For all $t \in [T]$, $\mathbb{P}$-a.s.,
  - State equations

$$A_t(\xi^t)x_t(\boldsymbol{\xi}^t) + B_t(\xi^t)x_{t-1}(\boldsymbol{\xi}^{t-1}) + C_t(\xi^t)y_t(\boldsymbol{\xi}^t) = b_t(\xi^t)$$

  - Recourse constraints (+ integrality)

$$D_t(\xi^t)x_t(\boldsymbol{\xi}^t) + E_t(\xi^t)y_t(\boldsymbol{\xi}^t) \geq d_t(\xi^t)$$

Infinite-dimensional problem!

Common Approaches

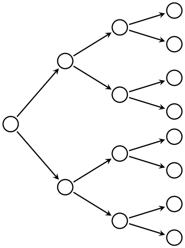
Approximate!

- Restrict the functional form of the policy **Decision rules**

$$\big(x_t(\boldsymbol{\xi}_t), y_t(\boldsymbol{\xi}_t)\big) \in \mathcal{F}_t\big(x_{t-1}(\boldsymbol{\xi}^{t-1}), \boldsymbol{\xi}_t\big)$$

- Model the underlying stochastic process in a structured way

Scenario trees



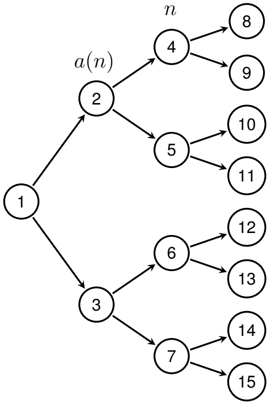
- Further assumptions ⇒ Exact methods
- Otherwise ⇒ Bounding techniques
  - Policy development
  - Dual bounding

Deterministic Equivalent

Scenario-tree Setting

Usually an *exponentially large* tree

Uncertainty model



MSP model

Extensive Form (node-based):

$$\min \sum_{n \in \mathcal{N}} p_n f_n(x_n, y_n)$$

$$\text{s.t. } (x_n, y_n) \in \mathcal{X}_n(x_{a(n)}) \quad \forall n \in \mathcal{N}$$

$y_n \rightarrow$ local variables

$x_n \rightarrow$ state variables

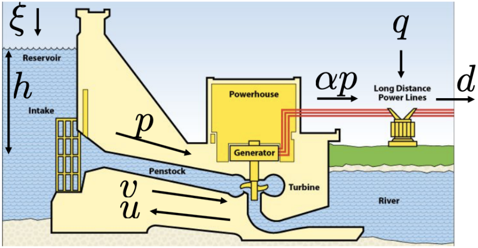
- Suppose fully linear

$\Rightarrow$ **Large-scale MILP**

Example: Hydropower planning

[Ahmed, 2016]

How much hydro power to generate in each period to satisfy demand?

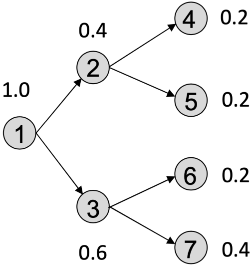


$$\min \sum_{t \in [T]} (b_t q_t + c_t u_t + g_t v_t)$$

$$\begin{aligned} \text{s.t. } & h_t = h_{t-1} + \xi_t - p_t + u_t - v_t && t \in [T] \\ & \alpha p_t + q_t = d_t && t \in [T] \\ & 0 \le h_t \le h^{\max} && t \in [T] \\ & p_t, q_t, u_t, v_t \ge 0 && t \in [T] \end{aligned}$$

Hydropower Example

Suppose inflows ξ are stochastic with the following scenario tree



$$
\begin{aligned}
\min\ & 1.0 \cdot (b_1 q_1 + c_1 u_1 + g_1 v_1) + \\
& 0.4 \cdot (b_2 q_2 + c_2 u_2 + g_2 v_2) + \\
& 0.6 \cdot (b_2 q_3 + c_2 u_3 + g_2 v_3) + \\
& \dots \\
& 0.4 \cdot (b_3 q_7 + c_3 u_7 + g_3 v_7)
\end{aligned}
$$

s.t. $\forall\ n = 1, \dots, 7:$

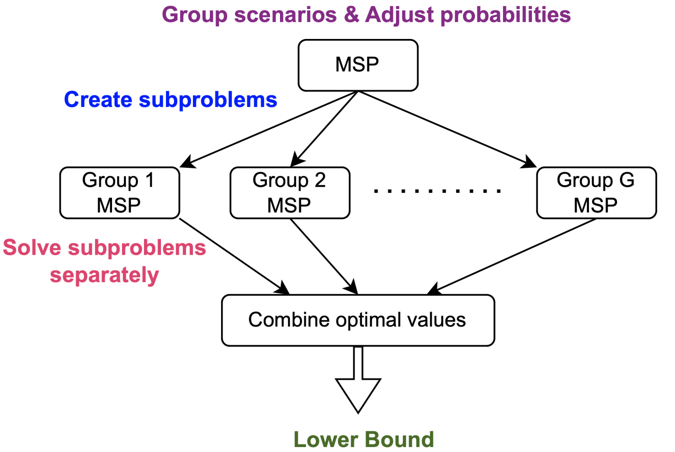
$$
\begin{aligned}
& h_n = h_{a(n)} + \xi_n - p_n + u_n - v_n \\
& \alpha p_n + q_n = d_n \\
& 0 \leq h_n \leq h^{\max} \\
& p_n, q_n, u_n, v_n \geq 0
\end{aligned}
$$

Tree Decomposition based Bounds

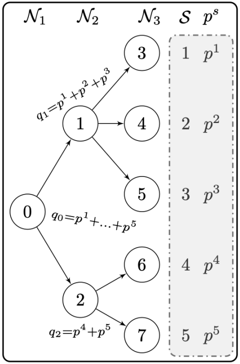
Bounds based on Scenario Tree Decomposition

Derive bounds on the MSP optimal value

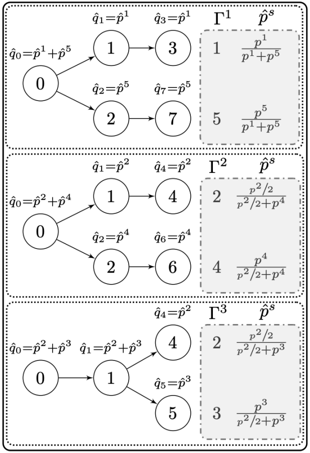
E.g.: [Maggioni et al., 2014, 2016], [Sandıkçı and Özaltın, 2017], [Bakır et al., 2019]



By Sandıkçı and Özaltın (2017)



(a) Scenario tree for the original problem



(b) Scenario trees for group subproblems *without* a reference scenario

Exact Approaches

Dynamic Programming Formulation

- For nodes at the last stage, define value functions:

$$Q_n\left(x_{a(n)}\right) := \min\left\{f_n(x_n, y_n) : (x_n, y_n) \in \mathcal{X}_n(x_{a(n)})\right\}$$

- For the others, recursively define (expected) value functions:

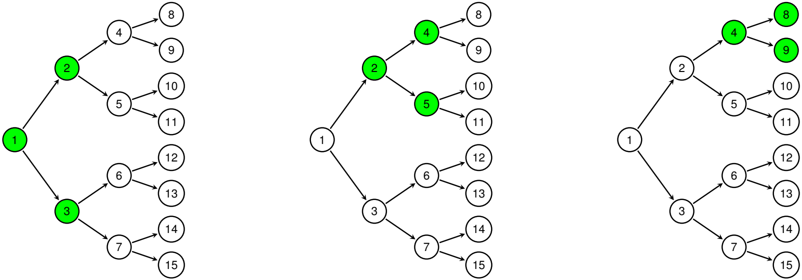
$$Q_n\left(x_{a(n)}\right) := \min\ f_n(x_n, y_n) + \overbrace{\sum_{m \in \mathcal{C}(n)} \bar{p}_{nm} Q_m(x_n)}^{\text{expected cost-to-go}}$$

$$\text{s.t. } (x_n, y_n) \in \mathcal{X}_n(x_{a(n)})$$

- MSP optimal value is given by $Q_1(x_0)$

Exact Approach: Purely continuous case

- General case: Nested Benders decomposition [Birge, 1985]

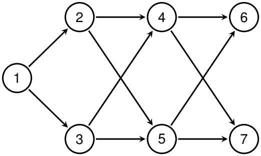


Via Benders cuts, approximate the expected cost-to-go functions:

$$Q_n(x_{a(n)}) = \min_{(x_n, y_n)} f_n(x_n, y_n) + \sum_{m \in \mathcal{C}(n)} \bar{p}_{nm} Q_m(x_n)$$

Exact Approach: Purely continuous case

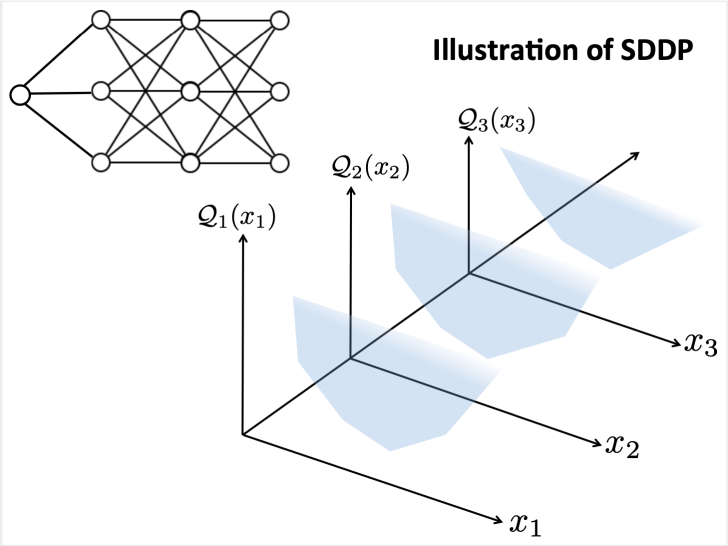
- Stage-wise independent case: SDDP [Pereira and Pinto, 1991]
  - Each stage has its own independent set of realizations
  - Can recombine the scenario tree:



  - One expected cost-to-go function per stage instead!
  - Many fewer nodes!

(SDDP Review Paper: [Füllner and Rebennack, 2023])

Shabbir's SDDP Illustration



Exact Methods

- Purely continuous:
  - General: Nested Benders [Birge, 1985]
  - Stage-wise independence: SDDP [Pereira and Pinto, 1991]
- Pure binary state variables: SDDiP [Zou et al., 2019]
  → Lagrangian cuts tight at binary points
- General integer state variables: Binarization + SDDiP
  → Large # of binary state variables
- Improved Lagrangian cut generation [Füllner, Sun, and Rebennack, 2024]
  → Lagrangian dual is normalized
  → Can get deep, facet-defining, or Pareto-optimal cuts
- Mixed-integer generalization [Deng and Xie, 2024]
  → ReLU Lagrangian cuts

Exact Methods

- Lipschitz continuous exp. cost-to-go functions: [Ahmed et al., 2020]
  $\rightarrow$ Nonlinear cuts + augmented Lagrangian

- General nonconvex mixed-integer nonlinear:
  - SDDP with generalized conjugacy cuts [Zhang and Sun, 2019]
    $\rightarrow$ Approximate *regularized* exp. cost-to-go functions
  - Nonconvex nested Benders [Füllner and Rebennack, 2022]
    $\rightarrow$ Extends binarization and regularization procedures
    $\rightarrow$ Successful implementation for deterministic multi-stage

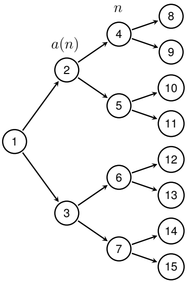
- Scaled-cut decomposition [Romeijnders and van der Laan, 2024]
  $\rightarrow$ Construct nonlinear cuts for the subproblems
  $\rightarrow$ Transform them into an affine cut for the master problem

Partially Extended Formulations

Continuous Recourse Case

[Castro, B., & Song, 2025]

Uncertainty model:



MSILP model:

$$\min \sum_{n \in \mathcal{N}} p_n f_n(x_n, z_n, y_n)$$

s.t. $\forall n \in \mathcal{N}$:

$$(x_n, z_n, y_n) \in \mathcal{X}_n(x_{a(n)}, z_{a(n)})$$

$y_n \in \mathbb{R}^m \rightarrow$ cont. local variables

$x_n \in \mathbb{R}^r \rightarrow$ cont. state variables

$z_n \in \mathbb{Z}^\ell \rightarrow$ int. state variables

- Partially extended DP formulation ($\rightarrow$ B&C + SDDP)
- Aggregation framework ($\rightarrow$ a range of policies)
  (leverage the stochastic process, e.g., Markov chain)

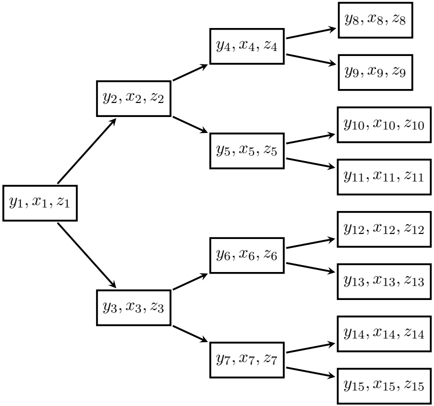
Our Idea

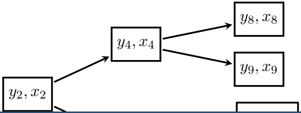
Challenge: Approximating nonconvex expected cost-to-go functions **(due to integer state variables)**

- Existing works: Develop exact lower-bounding techniques for the nonconvex expected cost-to-go functions
- Our work: Relocate all integer state variables to the first stage
  $\Rightarrow$ the resulting expected cost-to-go functions are **convex**
  $\Rightarrow$ can be approximated (exactly) by a decomposition scheme (e.g., nested Benders or SDDP)

Exact Method

Our idea: Relocate all integer state variables to the first stage

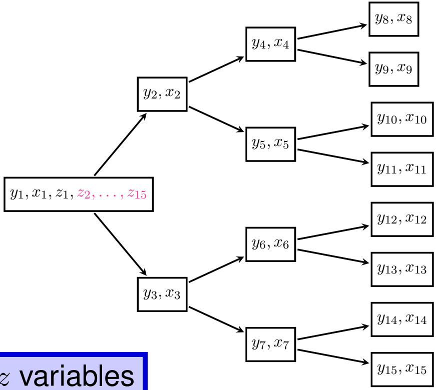




Develop High-quality Policies

Aggregate

Too many first-stage **(integer)** variables!



Impose additional structure to z variables

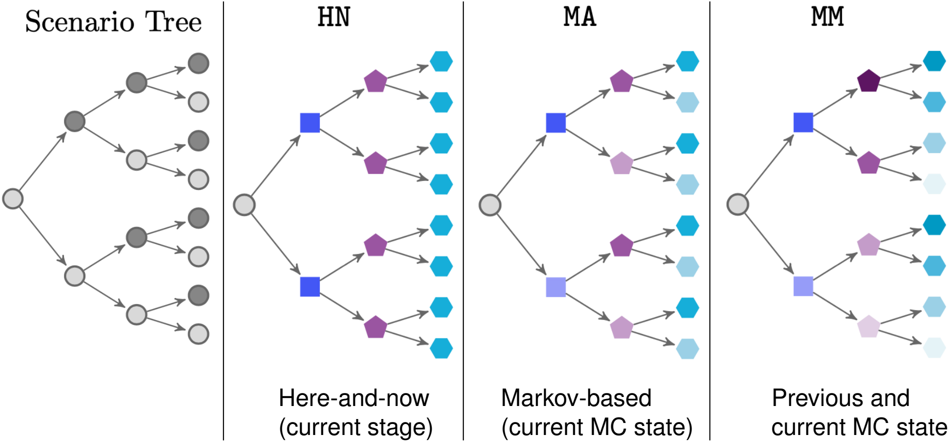
To obtain high-quality policies:

Leverage the structure of the underlying stochastic process

E.g.: Markov Chain

Aggregation Framework → Various Policies

Idea: Simply enforce $z_n = z_{n'}$ for some pairs of nodes based on MC



z^{A}'s: 15 4 7 11

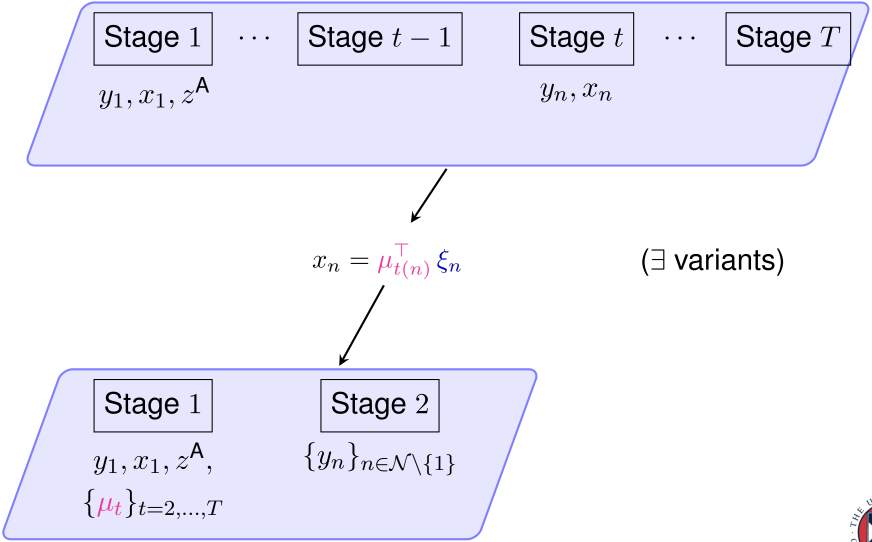
Solving the Aggregated Problem

- **Exactly:** B&C + SDDP
  (Employing the MC variant of SDDP)
- **Approximately:**
  - LB Exact method + an early stop in the SDDP sub-routine
  - ⋆ UB, i.e., policy Decision-rule restriction

Multi-stage ⟶ Two-stage

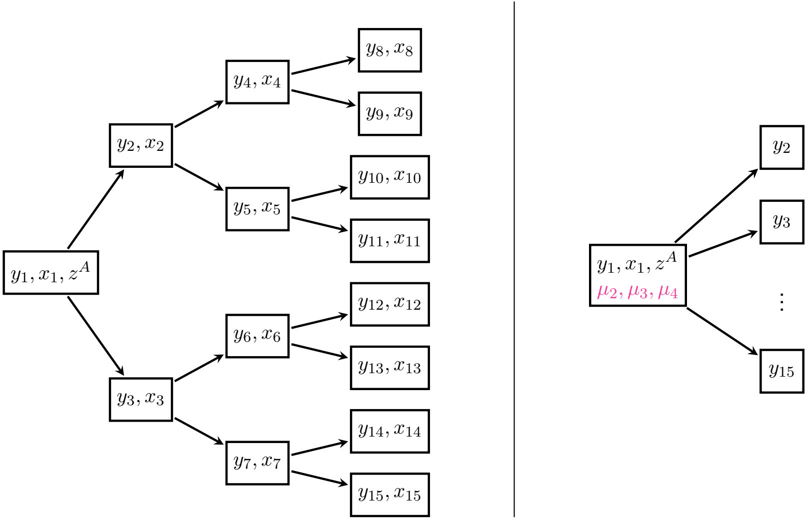
Two-stage Linear Decision Rules

[B. and Luedtke, 2022]



($\exists$ variants)

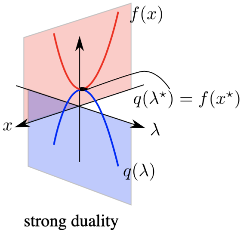
Our Scenario-tree Version

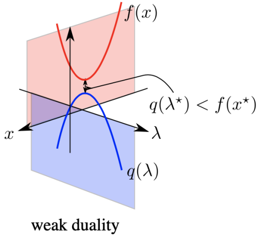


Candidate solution

More Decision Rules

Benefit from Duality





[From the website of Duc M. Nguyen]

[From the Trellis Society website]

MSLP: Consider its LP dual

[B. and Luedtke, 2022]

MSMIP: Consider a Lagrangian dual

[Daryalal, B., and Luedtke, 2024]

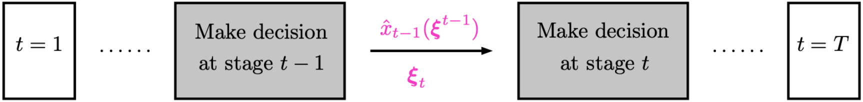
Apply LDRs on the dual variables ⇒ **two-stage** dual problem

Obtain LBs

Dual-driven Policies

General policy application scheme: Rolling the horizon

Solve an **optimization problem** at each stage



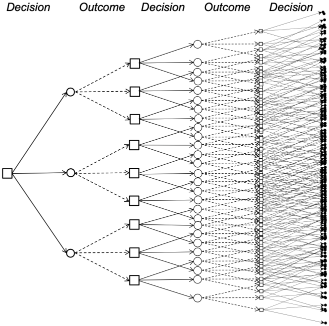
Dual-driven: Provide guidance by the dual LDR solution

Interpretable Policies

Desirable Properties

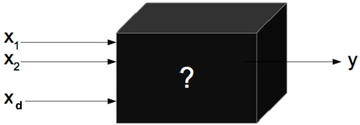
- Dynamic, complex decision-making under uncertainty ⇒ **MSP** ✓
- But, ∃ **major drawbacks**

Curses of dimensionality



© 2017 Warren B. Powell

Black box policies



Operators find it difficult to understand the connection between states and actions

⇒ mistrust in the policy

Goal: Efficiently generate interpretable policies

Interpretability Literature

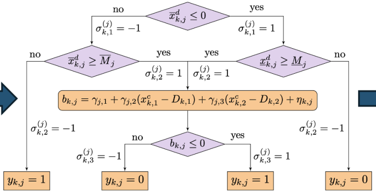
- Early attempts: Prove and use properties like monotonicity to generate simple policies
- Decision trees:
  - Highly interpretable structure (avoid black box policies)
  - Finding an optimal decision tree is difficult, building a tree in a top-down manner could result in a sub-optimal solution
- Rule lists:
  - Hard to find an optimal solution when the search space is large

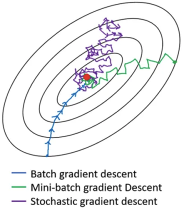
Our Approach: Smooth-in-expectation decision rules

$\rightarrow$ Policy structure: Flowchart

Overview: Smooth-in-expectation DRs [Hakizimana 2019]

$$
\begin{aligned}
\min_{\substack{\theta\in\Theta \\ x_k\in L^\infty(\Omega,\tilde{X}) \\ u_k\in L^\infty(\Omega,\tilde{U})}} \quad & C(\theta) + \mathbb{E}\left[\sum_{k=0}^{K} \ell_S(k, u_k(\omega), x_k(\omega), w_k, \theta)\right] \\
\text{s.t.} \quad & \mathbb{P}\left[g(k, u_k(\omega), x_k(\omega), w_k, \theta) \le 0\right] \ge 1-\epsilon \\
& x_0(\omega) = b_0, \quad \forall \omega \in \Omega \\
& x_{k+1}(\omega) = f(k, u_k(\omega), x_k(\omega), w_k, \theta), \quad \forall \omega \in \Omega \\
& u_k \quad \text{nonanticipative} \\
& \forall k \in \{0, \dots, K\}
\end{aligned}
$$





Multistage stochastic program with mixed-integer recourse

Flowchart for recourse decisions

- Interpretable
- Problem specific
- Introduce new decision variables

SGD algorithm

- Differentiability conditions
- Efficient

- **Key Observation:** Due to integer recourse, SAA objective is discontinuous. Nonetheless, the true expected value objective **might** be **smooth**!
- **Idea:** Apply certain DRs to all of the recourse decisions to ensure such smoothness ($\rightarrow$ simulation-optimization problem)

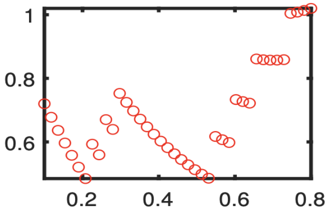
Smoothness Example

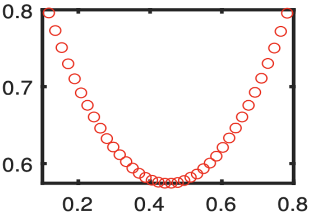
[Hakizimana 2019]

$$\min_{\theta \in [0.1, 0.8]} 2(\theta - 0.7)^2 + \mathbb{E}_\omega[\ell(\theta, \omega)]$$

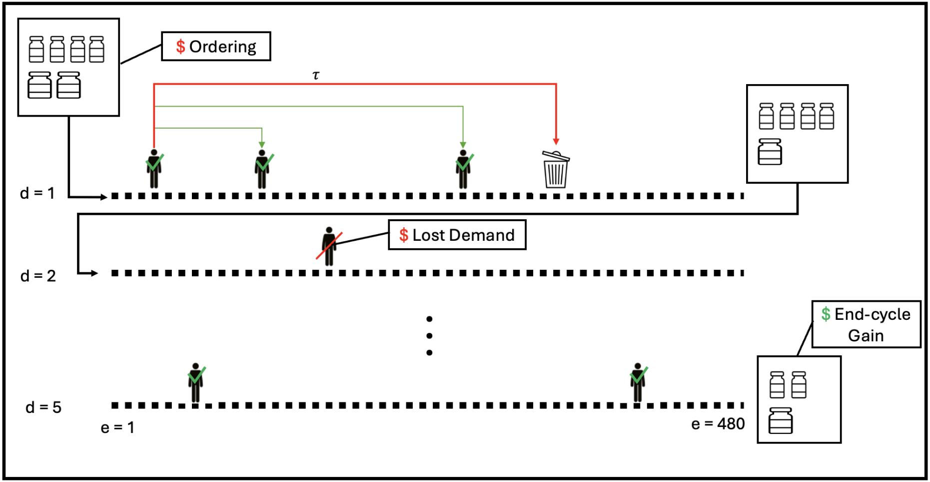
where $\ell(\theta, \omega) = \begin{cases} 0 & \text{if } \theta \leq \omega \\ 1 & \text{o.w.} \end{cases}$ and $\omega \sim U(0,1)$

- SAA with 7 scenarios vs True expected value objective





Vaccine Administration and Inventory Replenishment



Our Approach

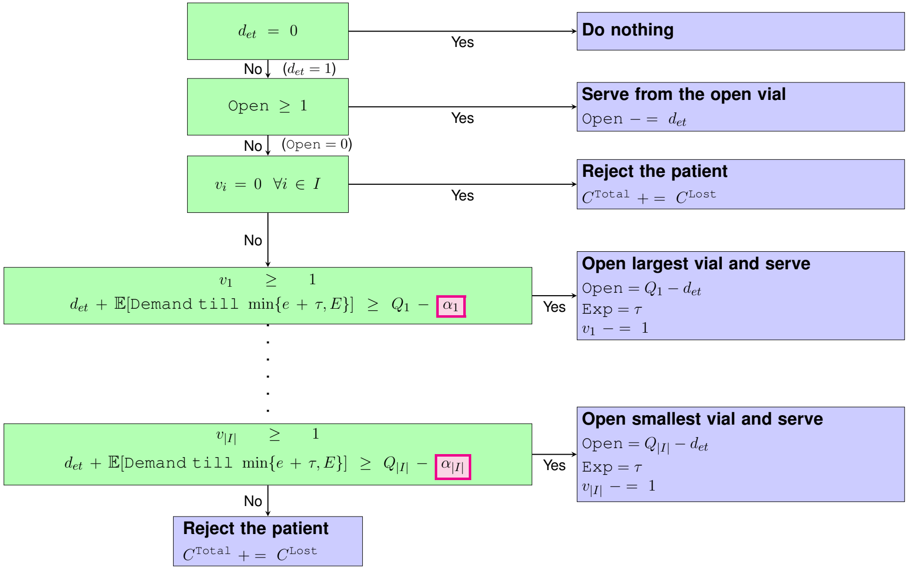
[Sherkat-Masoumi, B., and Luedtke, 2026+]

- A flexible multistage model
- Problem variations, e.g., incorporating queueing, early termination, unconditional service

- Joint optimization of ordering and administration decisions
- Policies: Efficient-to-compute, near-optimal, and interpretable

- Apply smooth-in-expectation decision rules for **vaccine administration decisions**
- Extent the framework to include integer first-stage **vaccine ordering decisions**

Vaccine Administration Policy Design (MSP → 1SP)



Summary

- Extensive form
- Bounds on the optimal value
- SDDP and its variants
- Partially extended formulations
- Aggregation-based policies
- Decision-rule based policies
- Interpretable policies

MSPs with integer decisions $\rightarrow$ computationally very challenging

Nice advancements

Still a lot of room for contributions