Methodological Advances in Two-stage Stochastic Programming

Merve Bodur
University of Toronto

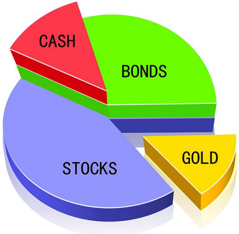
November 4, 2022

Optimization Under Uncertainty

"The only certainty is that nothing is certain."

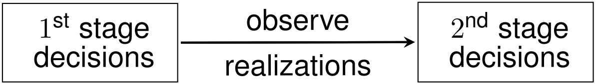
- Making decisions under uncertainty
- Integer/discrete decisions

$\Longrightarrow$ Stochastic programming



Two-Stage Stochastic Programs with Recourse

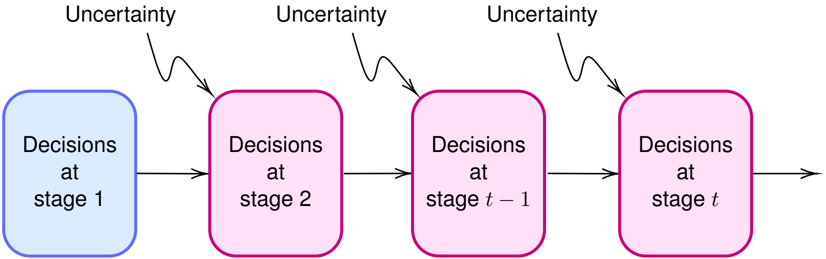
- There are uncertainties in some parameters
  (they are modeled as random variables)
- There are two decision stages



- Objective: min (1^{st} stage cost) + (Expected 2^{nd} stage cost)
- They have very wide range of applications

Computational Optimization Lab

Sequential Decision-making Under Uncertainty

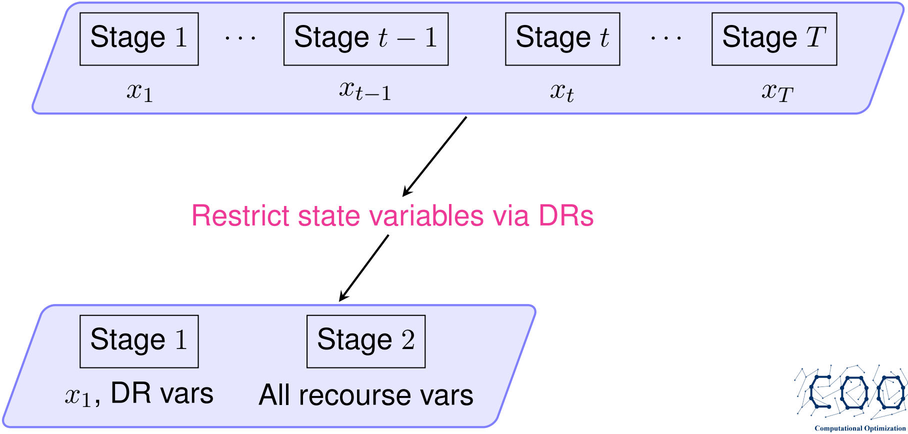


- Uncertainty is gradually observed
- Decisions are dynamically adapted
- Can be approximated via a **two-stage** model
  - Assumption: All the uncertainty is revealed after stage 1

Alternative Approximation

- Apply two-stage decision rules [B. & Luedtke, 2022]

  $\rightarrow$ Restrict state variables to be a specific formed (e.g., affine) function of observed uncertainty



Computational Optimization Lab

General Two-Stage Stochastic Program

$$\min_x \; c^\top x + \mathbb{E}_\omega[Q(x,\omega)]$$
$$\text{s.t. } x \in \mathcal{X}$$

Example recourse/value function:

$$Q(x,\omega) = \begin{array}{|ll|} \hline \min_y & q(\omega)^\top y \\ \text{s.t.} & T(\omega)x + W(\omega)y \geq h(\omega) \\ & y \geq 0 \\ \hline \end{array}$$

Challenges:

- Difficult to evaluate the expected value
  $\Rightarrow$ Use sample average approximation (SAA)
- **SAA problem** $\rightarrow$ Deterministic, but still difficult to solve

Assumption: Finitely many scenarios ($\mathcal{K}$)

Computational Optimization Lab

Extensive Form

A (very) large-scale (e.g., mixed-integer) program.

$$
\begin{aligned}
\min\ & c^\top x + \sum_{k \in \mathcal{K}} q_k^\top y_k \\
\text{s.t. } & T_k x + W_k y_k \geq h_k && \forall k \in \mathcal{K} \\
& x \in \mathcal{X} \\
& y_k \geq 0 && \forall k \in \mathcal{K}
\end{aligned}
$$

⇒ First classical approach: SCIP it! (Or, actually skip it!)

Extensive form does not scale well with $|\mathcal{K}|$

Call center staffing and scheduling instances: [B. & Luedtke, 2016]

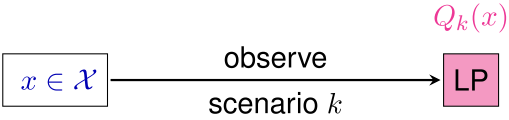
$I = 5,\ J = 5,\ T = 34,\ S = 333$, Time Limit = 3600 (sec)

$\lvert\mathcal{K}\rvert$	Gap(%)	Nodes
100	2.8	1840
500	10.6	28
1000	16.6	0
1500	28.9	0
2000	32.8	0

- Usually solved via decomposition

Computational Optimization Lab

Continuous Recourse



- Benders decomposition
- Dual decomposition

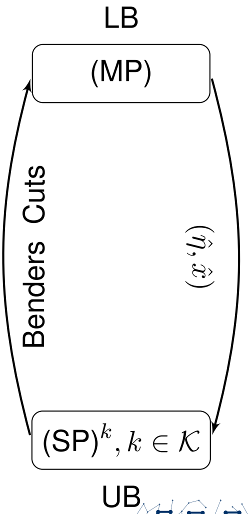
Benders Decomposition (L-Shaped Method)

$$(\text{MP}) : \min_{\eta, x} \; c^\top x + \sum_{k \in \mathcal{K}} \eta_k$$
$$\text{s.t. } x \in \mathcal{X}$$
$$\text{Benders Cuts for } \eta_k \geq Q_k(x) \quad \forall k \in \mathcal{K}$$
$$\eta \in \mathbb{R}^K$$

$$(\text{SP})^k : Q_k(\hat{x}) = \min_{y_k} \; q_k^\top y_k$$
$$\text{s.t. } W_k y_k \geq h_k - T_k \hat{x}$$
$$y_k \in \mathbb{R}^J_+$$

- Subproblem decomposes by scenario → LPs

Single-cut version: $\theta \geq \sum_{k \in \mathcal{K}} Q_k(x)$



Dual Decomposition

[Rockafellar and Wets, 1975]

Copy the first-stage decision-variables

$$\begin{aligned} \min\ & c^\top x + \sum_{k \in \mathcal{K}} q_k^\top y_k \\ \text{s.t. } & x_k = x && \forall k \in \mathcal{K} \\ & T_k x_k + W_k y_k \geq h_k && \forall k \in \mathcal{K} \\ & x_k \in \mathcal{X} && \forall k \in \mathcal{K} \\ & y_k \in \mathbb{R}^J_+ && \forall k \in \mathcal{K} \end{aligned}$$

Solve Lagrangian dual relaxing constraints $x_k = x$

- Decomposes problem by scenario
- Subproblems are mixed-integer programs

Computational Optimization Lab

Something in Between?

Benders decomposition

- Fast solution of LP relaxation (LP subproblems)
- Potentially weak bounds

Dual decomposition

- Expensive relaxation (many MIP subproblems)
- Potentially strong bounds

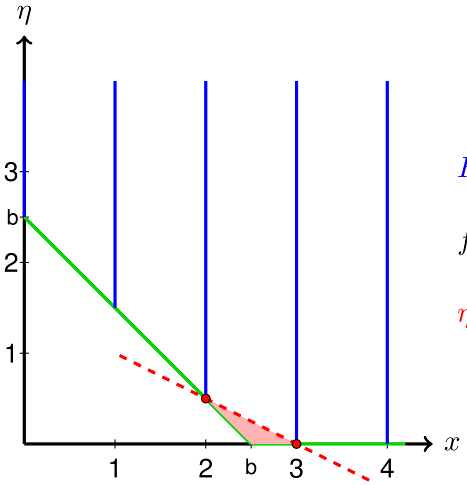
Idea

Strengthen Benders with first-stage integrality-based cuts

- Add MIR cuts to MP [B. & Luedtke, 2016]
- Also add cuts to SPs [B., Dash, Günlük & Luedtke, 2016]

Computational Optimization Lab

The basic mixed integer rounding inequality



$H = \{(\eta, x) \in \mathbb{R}_+ \times \mathbb{Z}_+ \mid \eta + x \geq b\}$

$f = b - \lfloor b \rfloor > 0$

$\eta \geq f(\lceil b \rceil - x)$ is valid for H

Mixed integer rounding (MIR)

Exact form of Benders cuts

$$H := \{(\eta, x) \in \mathbb{R}_+ \times \mathbb{Z}_+^I : \eta \geq d_0^1 - \sum_{i \in \mathcal{I}} d_i^1 x_i,\ \eta \geq d_0^2 - \sum_{i \in \mathcal{I}} d_i^2 x_i\}$$

Theorem

For any constant $\beta > 0$ with $\bar{f}_0 > 0$,

$$\eta \geq d_0^1 + \frac{\bar{f}_0 \lceil \beta(d_0^2 - d_0^1) \rceil}{\beta} - \sum_{i \in \mathcal{I}} \frac{\min\{\bar{f}_0 \lceil \beta(d_i^2 - d_i^1) \rceil, \bar{f}_i + \bar{f}_0 \lfloor \beta(d_i^2 - d_i^1) \rfloor\} + \beta d_i^1}{\beta} x_i$$

is valid for H where

$\bar{f}_0 := \beta(d_0^2 - d_0^1) - \lfloor \beta(d_0^2 - d_0^1) \rfloor$

$\bar{f}_i := \beta(d_i^2 - d_i^1) - \lfloor \beta(d_i^2 - d_i^1) \rfloor, \ \forall i \in \mathcal{I}.$

Applying MIR

How we obtain MIR inequalities:

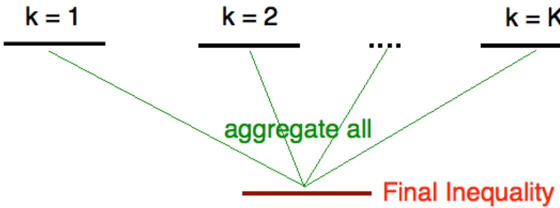
- Keep a pool of previously found Benders cuts
- Pair the current Benders cut with each previously found Benders cut and apply MIR

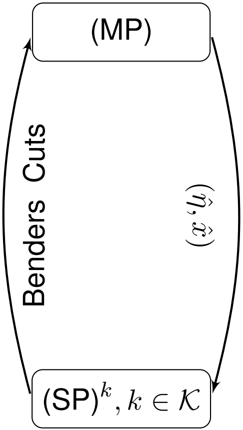
Can apply MIR in two different places:

1. Scenario level
2. Aggregated level

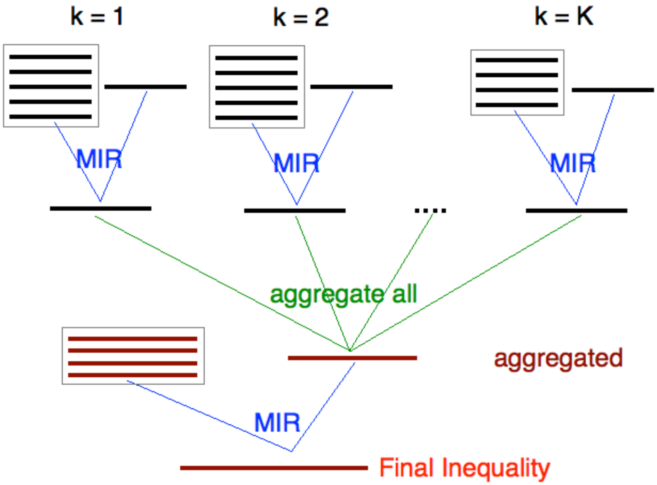
Computational Optimization Lab

Benders single cut generation

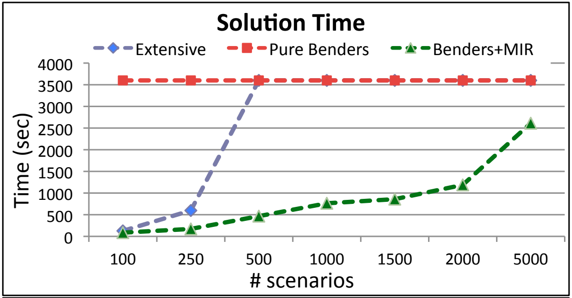


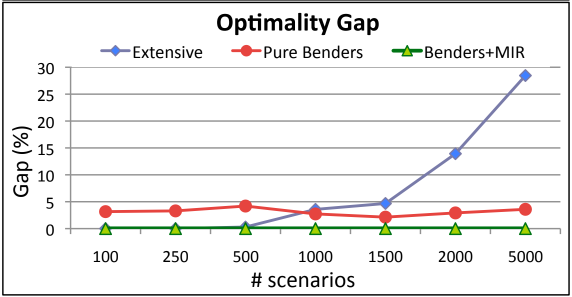


Cut generation using MIR



Call Center Staffing and Scheduling Instances





variables in Extensive Form:
534 + 42 · (# scenarios)

Average # nodes
in Benders+MIR = 750

Computational Optimization Lab

Two Options for Using Integrality-based Cuts

Strengthen Benders decomposition algorithm by:

- Project-and-cut: Add cuts to the master problem
- Cut-and-project: Add cuts to the subproblems

Project-and-cut

$$(\text{MP}) : \min_{\eta,x} c^\top x + \sum_{k\in\mathcal{K}} \eta_k$$

$$\text{s.t. } x \in \mathcal{X}$$

Benders cuts

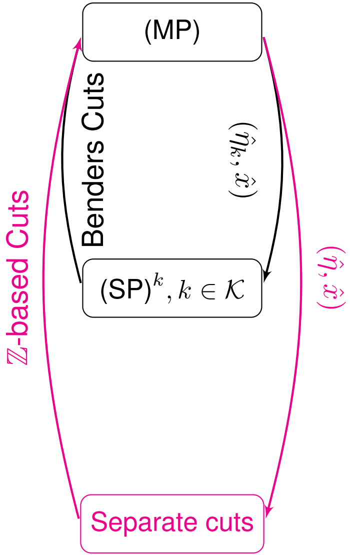
$$C\eta + Dx \geq g$$

$$\eta \in \mathbb{R}^K$$

$$(\text{SP})^k : Q_k(\hat{x}) := \min_{y_k} q_k^\top y_k$$

$$\text{s.t. } W_k y_k \geq h_k - T_k \hat{x}$$

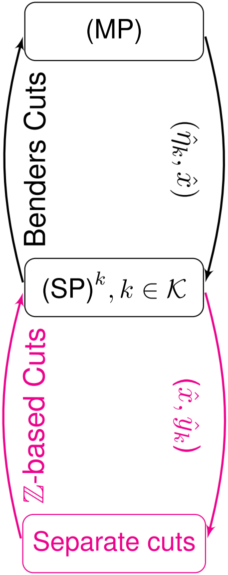
$$y_k \in \mathbb{R}_+^J$$



Cut-and-project

$$(\text{MP}) : \min_{\eta,x} c^\top x + \sum_{k\in\mathcal{K}} \eta_k$$
$$\text{s.t. } x \in \mathcal{X}$$
Benders cuts
$$\eta \in \mathbb{R}^K$$

$$(\text{SP})^k : Q_k(\hat{x}) := \min_{y_k} q_k^\top y_k$$
$$\text{s.t. } W_k y_k \geq h_k - T_k \hat{x}$$
$$C_k y_k \geq g_k - D_k \hat{x}$$
$$y_k \in \mathbb{R}_+^J$$



Add integrality-based cuts to $(\text{SP})^k$, even though it is an LP

Capacitated Facility Location Instances

- $K = 500$, Time limit = 4 hours

	Avg Time (# unsolved)			
CAP #	EXT	BEN	MP	SP
101-104	1171	- (4)	- (4)	149
111-114	10787(3)	- (4)	- (4)	957
121-124	10935(3)	- (4)	- (4)	4738(1)
131-134	9512(3)	- (4)	- (4)	1527
Mean Time	6020	-	-	1008
Avg Opt Gap	1.64%	14.87%	15.53%	0.02%

$\therefore$ Cut-and-project has far more impact

- Recent app on last-mile delivery with crowd-shipping and mobile depots: Also for the risk-averse (CVaR) setting [Mousavi, B., & Roorda, 2022]

Network Interdiction Instances

- $K = 456$, Time limit = 4 hours

Budget	EXT	BEN	SP	MP	MP+SP
30	- (5)	639	442	183	415
40	- (5)	7915(3)	2253	784	830
50	- (5)	8626(3)	2328	512	867
60	- (5)	10599(4)	2425(1)	906	1121
70	- (5)	- (5)	4435(1)	1402	1389
80	- (5)	- (5)	10096(4)	1938	1579
90	- (5)	- (5)	13283(4)	4794	4050
Mean Time	-	7536	3188	980	1169
Avg Opt Gap	25.7%	2.6%	0.4%	-	-

$\therefore$ Project-and-cut is very effective

Computational Optimization Lab

Aggregation Cuts

$Q_k(x)$

$x \in \mathcal{X}$ → observe scenario k → LP

Add Benders cuts to represent $\eta_k \geq Q_k(x)$

Aggregate second-stage constraints $\rightarrow$ change of variables: x to β

$Q_k^{\text{Agg}}(\beta)$

$x \in \mathcal{X}$, $\beta = \mathcal{A}(x)$ → observe scenario k → Agg-LP

First add Benders cuts to represent $\eta_k \geq Q_k^{\text{Agg}}(\beta)$

Telecommunications Application

Stochastic RWA and Lightpath Rerouting [Daryalal & B., 2022]

$x_{\ell,w}^{(s,d)}$ —— observe future ——→ $y_{\ell,w}^{(s,d)}$

Which wavelinks are used to serve existing requests

Which wave links are used to serve future requests

Aggregation over wavelengths: Substitute

$$\sum_{(s,d)\in\mathcal{SD}_k^2} y_{\ell,w}^{(s,d)} \leq 1 - \sum_{(s,d)\in\mathcal{SD}^1} x_{\ell,w}^{(s,d)} \quad \forall w \in \mathcal{W}, \ell \in \mathcal{L}$$

with

$$\sum_{w\in\mathcal{W}} \sum_{(s,d)\in\mathcal{SD}_k^2} y_{\ell,w}^{(s,d)} \leq (|\mathcal{W}| - \beta_\ell) \quad \forall \ell \in \mathcal{L}$$

Computational Optimization Lab

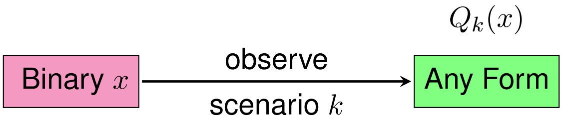
Optical Network Instances

Time limit = 600 seconds

K	EXTENSIVE		BENDERS-x			BENDERS-$x\beta$			
	time (s)	gap (%)	time (s)	gap (%)	$\#x-$cuts	time (s)	gap (%)	$\#\beta-$cuts	$\#x-$cuts
10	200	0	187	3	1124	17	0	8	15
20	TL	NA	465	2	1370	42	0	19	61
30	TL	NA	536	3	1538	157	0	31	204
40	TL	NA	TL	3	1785	164	0	34	203
50	TL	NA	TL	3	1980	206	0	58	209
100	TL	NA	TL	3	2164	305	0	99	400

Integer L-shaped Method

[Laporte & Louveaux, 1993]



- MP provides a candidate: $(\hat{x}, \hat{\eta})$
- (SP)^k evaluates $Q_k(\hat{x})$
- Integer L-shaped cut:

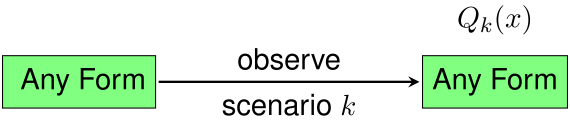
$$x = \hat{x} \Rightarrow \eta_k \geq Q_k(\hat{x})$$

$$x \neq \hat{x} \Rightarrow \eta_k \geq Q_k^{\text{LB}}$$ (i.e., redundant)

- Cuts might be strengthened based on problem-specific structure

Computational Optimization Lab

Logic-based Benders Decomposition



- LBBD cuts from the inference dual [Hooker & Ottosson, 2003]
- Very successful applications in IP
- Very few applications in SP

Distributed Operating Room Scheduling [Guo et al., 2021]

1^{st} stage	→	2^{nd} stage
Assign patients to (hospital, room, day)	observe surgery durations	Cancel patients if there is overtime

Objective: min (Operational cost) + (Expected cancellation cost)

$$
\begin{aligned}
Q^k_{hdr}(\hat{x}) = \min_z \quad & \sum_{p \in \mathcal{P}} c^{\text{cancel}}_p (\hat{x}_{hdpr} - z^k_{hdpr}) & \\
\text{s.t.} \quad & z^k_{hdpr} \leq \hat{x}_{hdpr} & p \in \mathcal{P} \\
& \sum_{p \in \mathcal{P}} T^k_p z^k_{hdpr} \leq B_{hd} & \\
& z^k_{hdpr} \in \{0, 1\} & p \in \mathcal{P}
\end{aligned}
$$

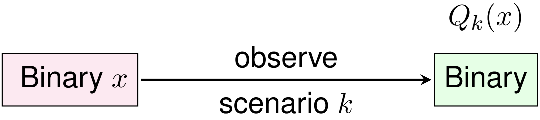
LBBD cut:

$$
\eta^k_{hdr} \geq Q^k_{hdr}(\hat{x}) - \sum_{p \in \hat{\mathcal{P}}_{hdr}} c^{\text{cancel}}_p \Big(1 - x_{hdpr}\Big)
$$

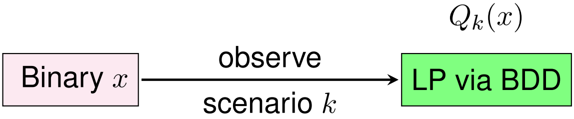
where $\hat{\mathcal{P}}_{hdr} = \{p \in \mathcal{P} | \hat{x}_{hdpr} = 1\}$

Computational Optimization Lab

Convexification via Binary Decision Diagrams



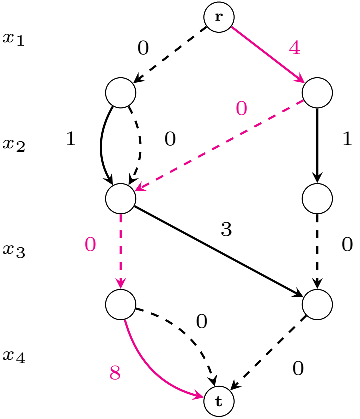
- Represent the second-stage problem via BDDs



$\Rightarrow$ amenable to Benders decomposition

Knapsack BDD Example

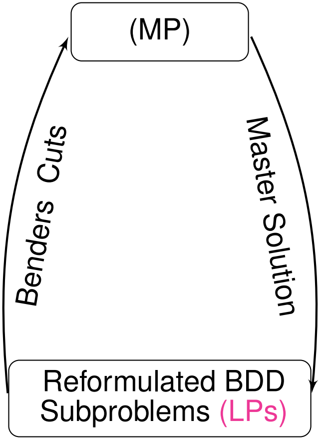
$$\max_x \; 4x_1 + x_2 + 3x_3 + 8x_4$$
$$\text{s.t. } 2x_1 + x_2 + 3x_3 + 3x_4 \le 5$$
$$x \in \{0,1\}^4$$



$$\max_f \sum_{a \in \mathcal{A}} w_a f_a$$
$$\text{s.t.} \sum_{a | s(a)=r} f_a = 1$$
$$\sum_{a|s(a)=i} f_a - \sum_{a|d(a)=i} f_a = 0 \quad \forall i \in \mathcal{N} \setminus \{r,t\}$$
$$\sum_{a|d(a)=t} f_a = -1$$
$$f_a \ge 0 \qquad \forall a \in \mathcal{A}$$

Approach by [Lozano & Smith, 2018]

- Assume special structure: Each recourse constraint is impacted by at most one first-stage binary variable
- Transform recourse problem to a capacitated shortest path problem
- Derive strengthened classical Benders cuts



The Transformed BDD Subproblem

$$\min \sum_{p\in\mathcal{P}} c_p^{\text{cancel}}(\hat{x}_{hdpr} - z_{hdpr}^k)$$
$$\text{s.t.} \sum_{p\in P} T_p^k z_{hdpr}^k \le B_{hd}$$
$$z_{hdpr}^k \le \hat{x}_{hdpr} \qquad p\in\mathcal{P}$$
$$z_{hdpr}^k \in \{0,1\} \qquad p\in\mathcal{P}$$

→ A knapsack problem

$$\min \sum_{a\in\mathcal{A}^k} g_a^k f_a$$
$$\text{s.t.} \sum_{a|s(a)=r} f_a = 1$$
$$\sum_{a|s(a)=i} f_a - \sum_{a|d(a)=i} f_a = 0 \quad i\in\mathcal{N}^k\setminus\{r,t\}$$
$$\sum_{a|d(a)=t} f_a = -1$$
$$f_a \le \hat{x}_{hdpr} \qquad a\in\mathcal{A}_1^{sp}$$
$$f_a \ge 0 \qquad a\in\mathcal{A}^k$$

→ A shortest path problem

Computational Optimization Lab

Further Leveraging Binary Decision Diagrams

- Previously in [Lozano & Smith, 2018]:

$$Q_k(x) = \min \; q_k^\top y$$

$$\text{s.t. } y \in \mathcal{Y}_k \subseteq \{0,1\}^{n_\mathrm{y}}, \quad \boxed{x_i^\mathrm{B} = 0 \implies y \in \mathcal{Y}_i^{\mathtt{logical},k} \quad \forall i = 1, \dots, n_\mathrm{x}^\mathrm{B}}$$

- Recently in [MacNeil & B., 2022]:

$$Q_k(x) = \min q_k^\top y$$

$$y \in \mathcal{Y}_k \subseteq \{0,1\}^{n_\mathrm{y}}, \quad \boxed{\mathbb{I}(L_j^k(x)) = 1 \implies y \in \mathcal{Y}_j^{\mathtt{logical},k} \quad \forall j = 1, \dots, m}$$

and

$$Q_k(x) = \min \; \boxed{(q_k^1 + q_k^2)^\top y}$$

$$y \in \mathcal{Y}_k \subseteq \{0,1\}^{n_\mathrm{y}}, \quad \boxed{\mathbb{I}(L_j^k(x)) = 1 \implies q^1_{k,\sigma(j)=0} \quad \forall j = 1, \dots, m}$$

New BDD-based Approaches

Model 1:

- Generalizes the existing BDD-based decomposition approach
- **Arc capacities** in the BDDs are parametrized by x

Model 2:

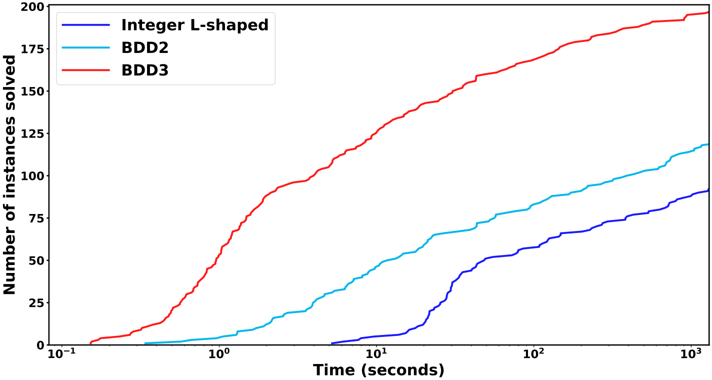
- Novel; might be more natural for certain applications
- **Arc costs** in the BDDs are parametrized by x

They are extended to a risk-averse setting as well

Computational Optimization Lab

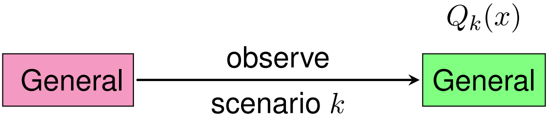
Dominating Set Instances

- Up to 50 vertices, varying edge densities, 850 scenarios
- Solution time limit = 1 hour (i.e., after BDD generation)



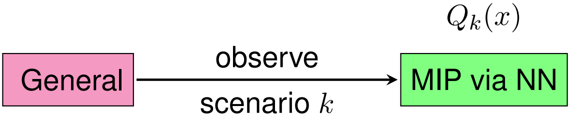
Neur2SP

[Dumouchelle, Patel, Khalil, & B., 2022]

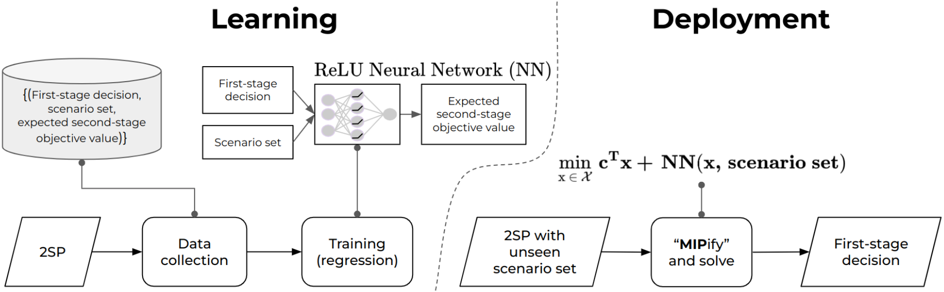


The idea: Just SCIP it!

- Learn $Q_k(x)$ or even better $\mathbb{E}_k[Q_k(x)]$ via supervised learning
- MIPify the obtained neural network (NN)
- Solve the combined surrogate model



Neur2SP Overview



Variety of Problem Classes

Problem	First stage	Second stage	Objective	Constraints
CFLP	Binary	Binary	Linear	Linear
INVP	Continuous	Binary	Linear	Linear
SSLP	Binary	Binary	Linear	Linear
PP	Binary	Continuous	Bilinear	Bilinear

Computational Optimization Lab

Snapshot of Results

SSLP_(**n** servers)_(**m** clients)_(**K** scenarios)

500k and 1m variables in EF

Problem	Gap to Optimal (%)		Solving Time	
	Neur2SP	EF	Neur2SP	EF
SSLP_10_50_50	**0.00**	**0.00**	**0.11**	10,801.27
SSLP_10_50_100	**0.00**	**0.00**	**0.11**	10,800.04
SSLP_10_50_500	**0.00**	0.00	**0.11**	10,818.23
SSLP_10_50_1000	**0.00**	28.64	**0.12**	10,800.26
SSLP_10_50_2000	**0.00**	51.24	**0.13**	10,800.20
SSLP_15_45_5	0.46	**0.00**	**0.32**	4.17
SSLP_15_45_10	1.57	**0.00**	**0.25**	3.71
SSLP_15_45_15	0.53	**0.00**	**0.41**	4.74
SSLP_5_25_50	**0.00**	**0.00**	**0.26**	2.35
SSLP_5_25_100	**0.00**	**0.00**	**0.18**	8.87

EF times out after 3 hours with huge gaps

Findings are similar for the other problem classes[1]

[1] Progressive hedging ∼10,000 slower based on [Torres et al., 2022]

Computational Optimization Lab

Summary

- Bring IP technology to stochastic IP
  - Integrality-based cuts
  - Logic-based cuts
- Incorporate new tools:
  - Decision diagrams
  - Machine learning

Computational Optimization Lab