A Fast Matheuristic for Two-Stage Stochastic Programs through Supervised Learning

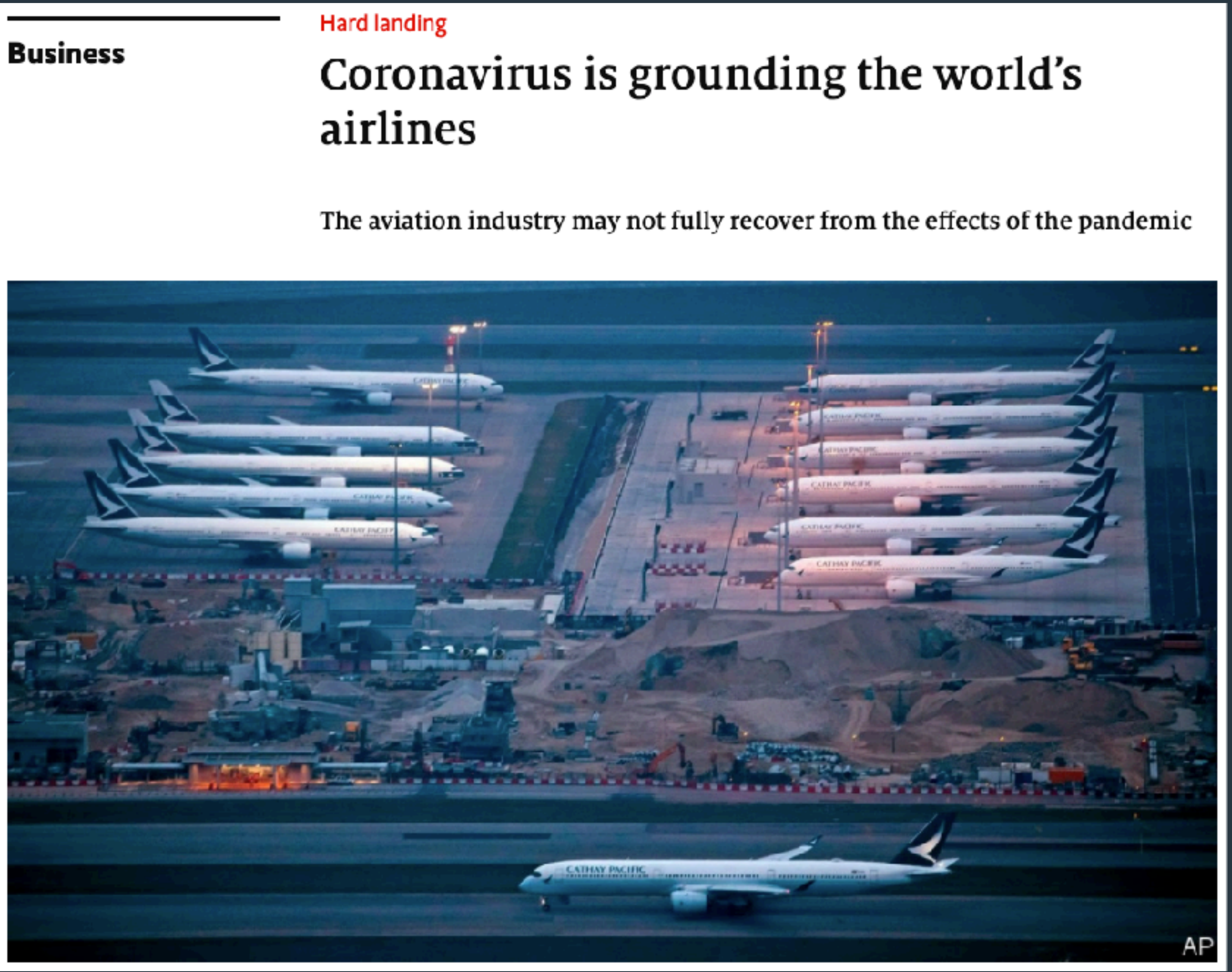
ERIC LARSEN, EMMA FREJINGER, BERNARD GENDRON AND ANDREA LODI

ELLIIT focus period workshop:

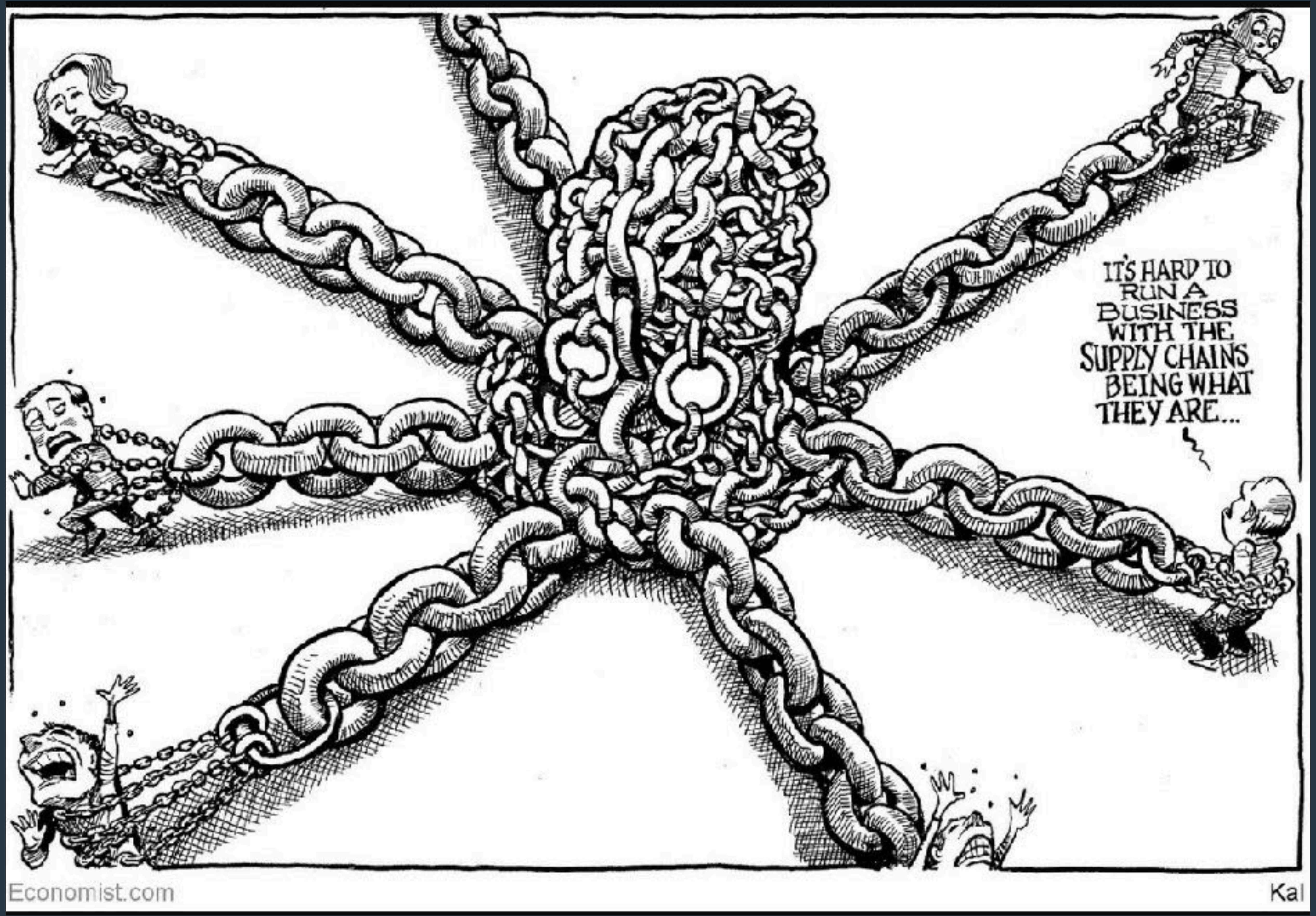
Hybrid AI – Where data-driven and model-based methods meet

November 1, 2022

arXiv:2205.00897



The Economist, March 15th 2020



The Economist, January 14th 2022

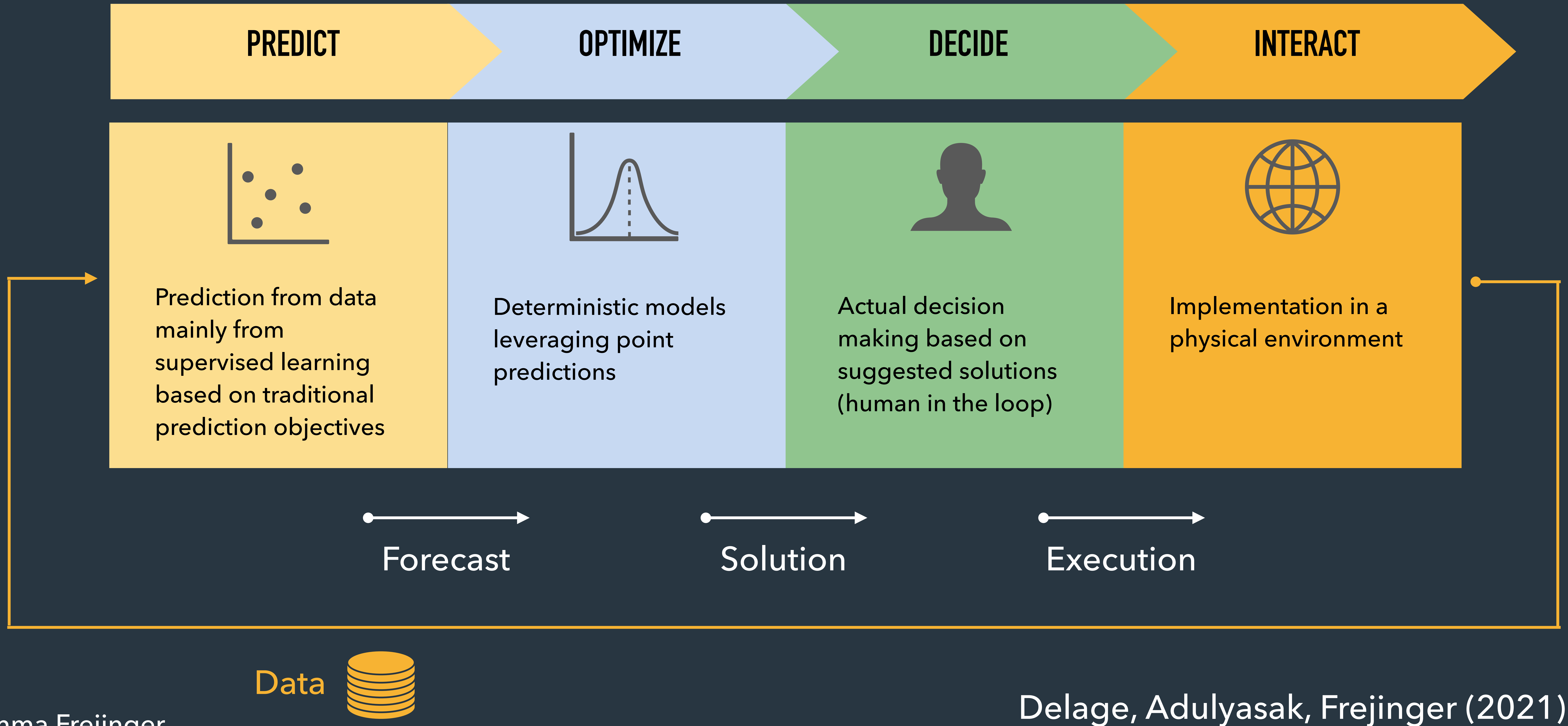
Characteristics of many decision-making problems related to freight transportation:

Large-scale discrete optimization problems

Subject to uncertainty

Similar problem instances solved repeatedly over time

Decision-making process, traditionally deterministic models and done in silos



Delage, Adulyasak, Frejinger (2021)

Pervasive issues arise when facing uncertainty

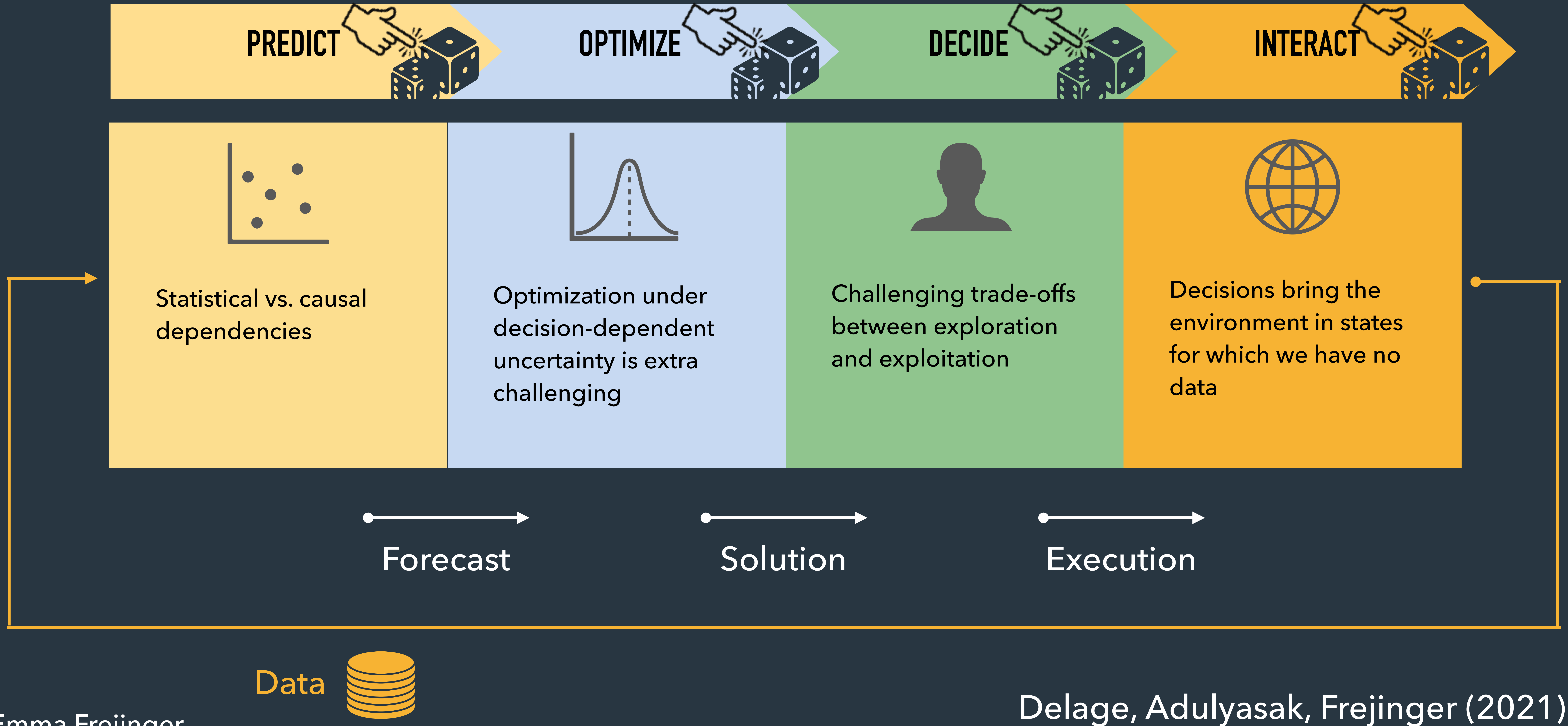
PREDICT	OPTIMIZE	DECIDE	INTERACT
Point predictions lead to unexplained variations Nature rarely follows a “nice” distribution	Deterministic models only control one scenario of the future Solving stochastic models is challenging	Decision maker is not risk neutral Stochastic models are over simplified or too obscure	The past does not predict the future Post-decision disappointment

Forecast → Solution → Execution

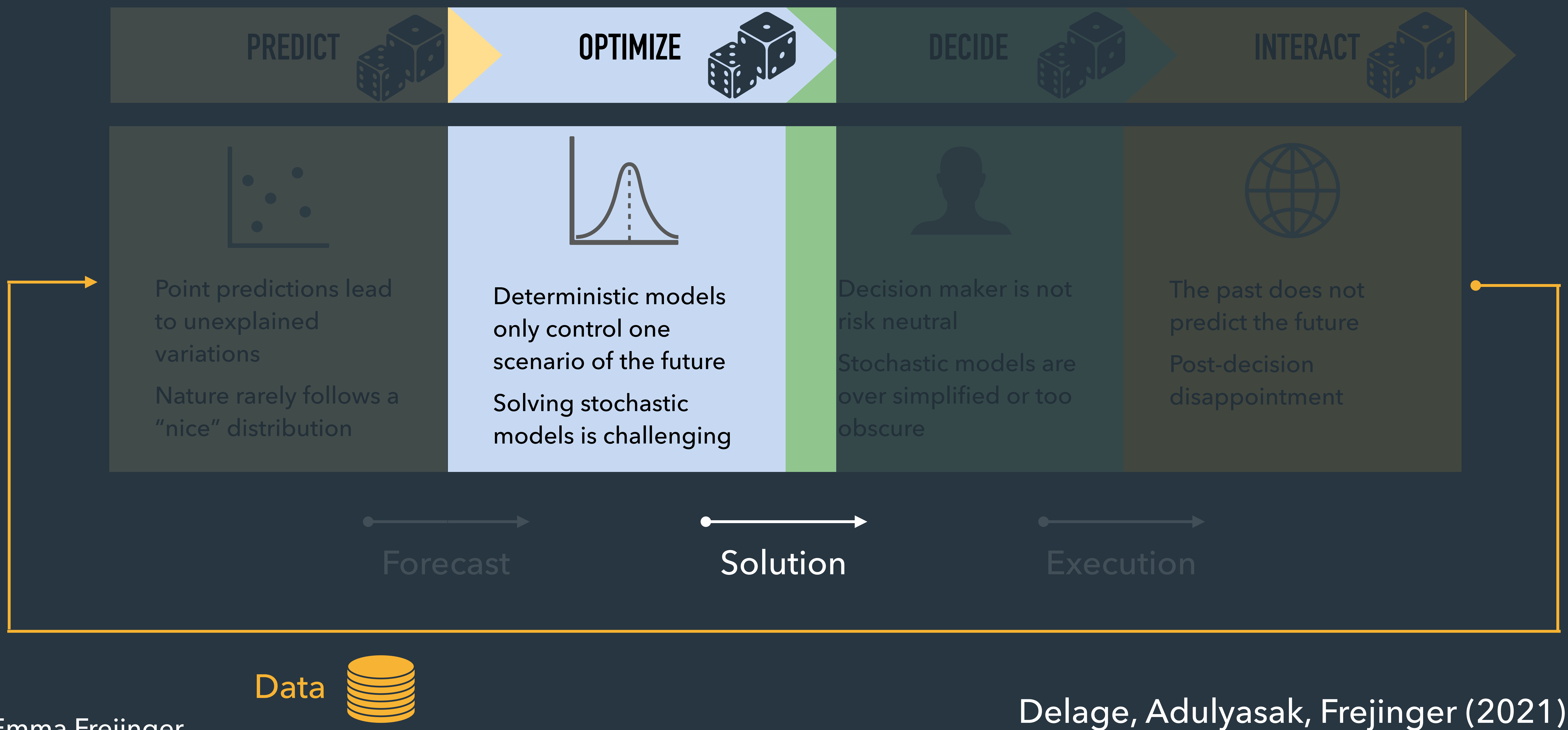
Data

Delage, Adulyasak, Frejinger (2021)

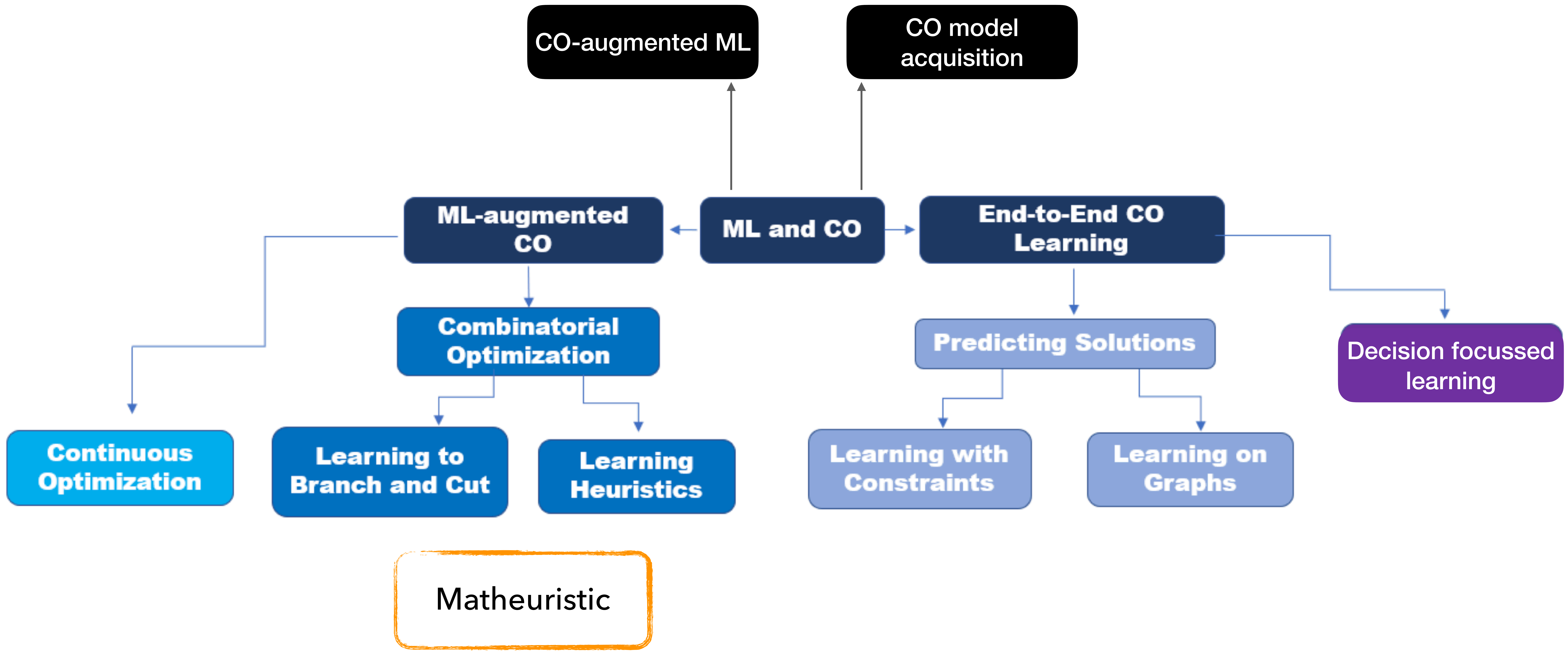
Endogeneity perturbs the entire decision-making process



Delage, Adulyasak, Frejinger (2021)



Delage, Adulyasak, Frejinger (2021)



James Kotary, Ferdinando Fioretto, Pascal Van Hentenryck, Bryan Wilder:
End-to-End Constrained Optimization Learning: A Survey. IJCAI 2021: 4475-4482

OUTLINE

1 Background

Can we predict useful (expected) information about second-stage problems in two-stage formulations?

2 Motivation and scope

Integer linear two-stage stochastic programs with hard second-stage problems

3 ML-L-Shaped: replace costly computations in the L-shaped method by ML predictions

4 Extensive numerical study: ML-L-Shaped gives a speedup of x6 to x167 compared to the best performing exact method

Optimality gaps close to zero

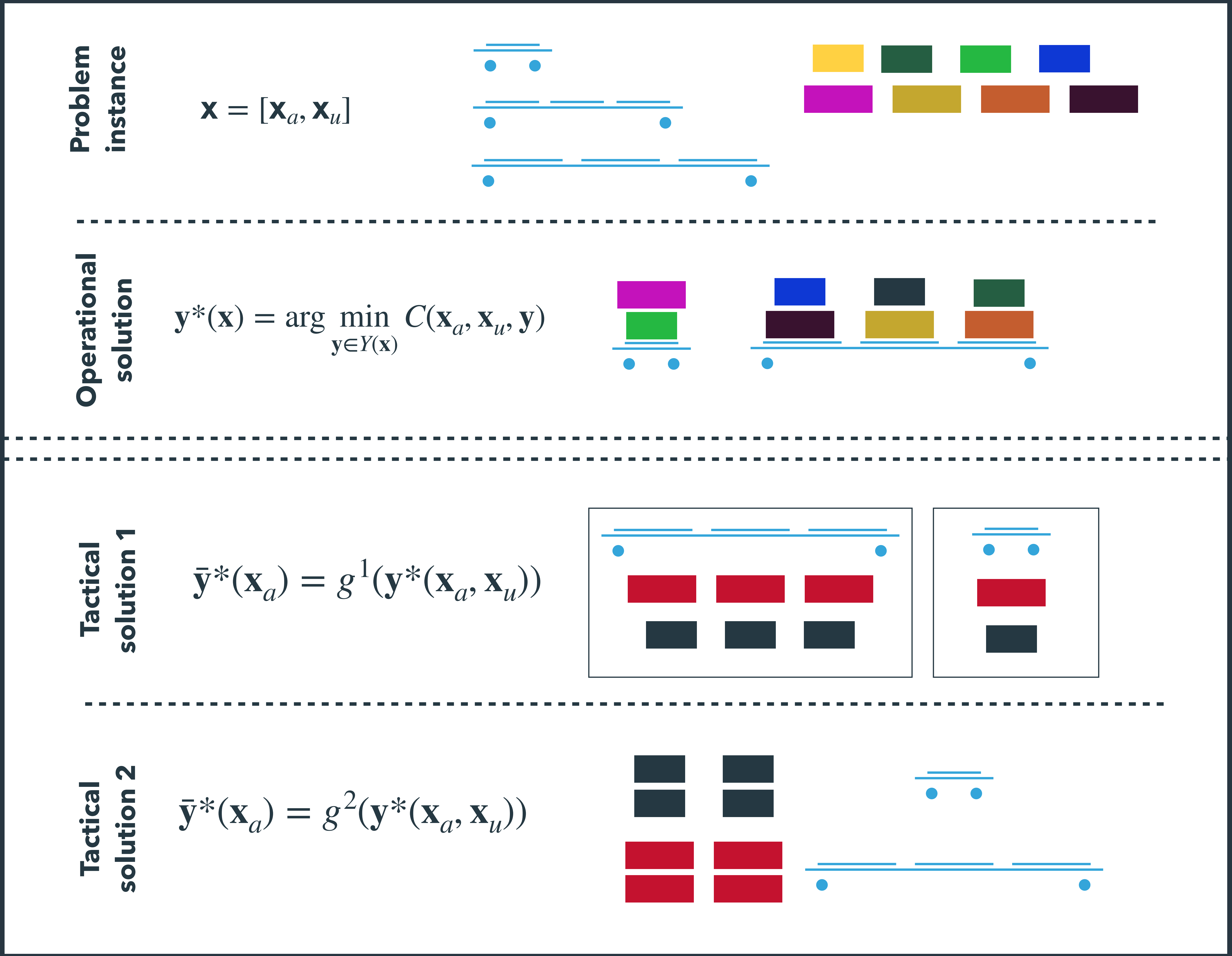
1

Can we predict useful (expected) information about second-stage problems in two-stage formulations?

- Problem: quickly predict expected descriptions of second-stage problem solutions (synthesis of a solution) conditional on first-stage variables
- Data: large number of (sampled) deterministic second-stage problems solved independently and offline
- Supervised learning: examples – available information on instance and synthesized solution

1

Can we predict useful (expected) information about second-stage problems in two-stage formulations?



1

Can we predict useful (expected) information about second-stage problems in two-stage formulations?

YES

- Prediction accuracy close lower bounds computed using sample average approximation
- Predictions generated in milliseconds

1

Can we predict useful (expected) information about second-stage problems in two-stage formulations? Yes.

WHY DO WE CARE?

- Predictions useful for real-time applications or as part of another algorithm (solve two/multi-stage problem)
  - Avoids online generation of multiple second-stage scenarios and solutions
- Easy to implement in practice (standard supervised learning and a general purpose solver)

CURIOUS?

ERIC LARSEN, SÉBASTIEN LACHAPELLE, YOSHUA BENGIO, EMMA FREJINGER, SIMON LACOSTE-JULIEN AND ANDREA LODI

Predicting Tactical Solutions to Operational Planning Problems Under Imperfect Information, IJOC 34(1):227-242, 2021.

2

Motivation and scope

Integer linear two-stage stochastic programs with hard second-stage problems

General two-stage linear stochastic program (notation follows e.g. Angulo et al., 2016)

$$
\begin{aligned}
\min_{x,z,\theta} \quad & \{cx + dz + \theta\} \\
\text{s.t.} \quad & Ax + Cz \leq b, && (1) \\
& Q(x) - \theta \leq 0, && (2) \\
& x \in \{0,1\}^n, && (3) \\
& z \geq 0, \quad z \in \mathcal{Z}, && (4)
\end{aligned}
$$

Second-stage cost of x with respect to random data $\xi = (q_\xi, W_\xi, T_\xi, h_\xi)$ with **finite support**

$$\boxed{Q(x)} :\equiv \mathbb{E}_\xi[\min_y\{q_\xi y : W_\xi y \geq h_\xi - T_\xi x, y \in \mathcal{Y}\}]$$

Integrality constraints on y

$$= \sum_\xi p_\xi Q_\xi(x)$$

SCOPE

- Two-stage stochastic programming
  - E.g., tactical planning with second-stage operational planning problem (relatively complete recourse)
- Costly integer second-stage problems
- High level of uncertainty – large number of scenarios

Master (first-stage) problem

$$\min_{x,z,\theta} \{cx + dz + \theta\}$$
$$\text{s.t. } (1), (3), (4),$$
$$\boxed{\Pi x - \mathbf{1}\theta \leq \pi_0,}$$
$$\theta \geq L$$

Integer L-shaped optimality cut at x^* where $S(x^*) := \{i : x_i^* = 1\}$ and L lower bound on $Q(x^*)$

$$(Q(x^*) - L)\Big(\sum_{i \in S(x^*)} x_i - \sum_{i \notin S(x^*)} x_i - |S(x^*)|\Big) + Q(x^*) \leq \theta$$

Subgradient cut given by continuous relaxation $\tilde{Q}(x)$ of $Q(x)$

$$s(x - x^*) + \tilde{Q}(x^*) \leq \theta$$

Subgradient s of $\tilde{Q}(x)$ at x^*

RELATED WORK

- Exact methods
  - Integer L-shaped method (Laporte and Louveaux, 1993)
  - L-shaped method with alternating cut strategy (Angulo et al., 2016) Avoid costly computations of $Q(x)$ by first checking feasibility, if infeasible add subgradient cut
- Heuristics, e.g.,
  - Progressive hedging (Rockafellar and Wets, 1991, Watson and Woodruff, 2011)
  - Dual decomposition (Carøe and Schultz, 1999)
  - Neur2SP (Dumouchelle et al., 2022)

3

ML-L-Shaped:

replace costly computations in the L-shaped method by ML predictions

Integer L-shaped optimality cut at candidate solution x^* where $S(x^*) := \{i : x_i^* = 1\}$ and L lower bound on $Q(x^*)$

$$(Q(x^*) - L)\Big(\sum_{i \in S(x^*)} x_i - \sum_{i \notin S(x^*)} x_i - |S(x^*)|\Big) + Q(x^*) \le \theta$$

Continuous L-shaped (subgradient) optimality mono-cut (Birge and Louveaux, 2011)

$$\mathbb{E}_\xi[\phi(h_\xi - T_\xi x) - \mathbf{1}'\psi] \le \theta$$

ϕ and ψ are solutions to the dual of the continuous relaxation of subproblem at x^*

Predictions

$Q^{ML}(x^*)$	Subproblem value $Q(x^*)$
$\tilde{Q}^{ML}(x^*)$	Relaxed subproblem value $\tilde{Q}(x^*)$
ϕ^{ML}, ψ^{ML}	Solutions ϕ and ψ, respectively

IDEA

- Solve problem instances stemming from a distribution of instances sharing similar characteristics
- Matheuristic
  - L-shaped method with or without alternating cuts
  - Costly computations replaced by fast machine learning predictions

Algorithm 2 Benders decomposition: Heuristic callback

procedure HeuristicCallback($\boldsymbol{isAlt}, \boldsymbol{\mu}, \boldsymbol{\nu}$)

    end if

    Compute prediction $Q^{ML}(x^*)$ ▷ Integer L-shaped method

    if $\nu Q^{ML}(x^*) \leq \theta^*$ **then**

        if $cx^* + dz^* + \theta^* < UB$ **then**

            $UB \leftarrow cx^* + cx^* + \theta^*$ ▷ Update upper bound

            $(x^{**}, z^{**}) \leftarrow (x^*, z^*)$ ▷ Update incumbent solution

            return

        end if

    else

        Add a heuristic integer L-shaped cut

        return

    end if

    return

end procedure

Implementation

C-language bindings launch GPU computations returning ML predictions

ML-L-SHAPED

ML-Standard-L-Shaped

- Compute prediction of $Q(x^*)$
- Shift coefficient: Control bias against rejection of valid first-stage integral candidate solutions

$$\textbf{if } \mu Q^{ML}(x^*) \leq \theta^*$$

Algorithm 2 Benders decomposition: Heuristic callback

1: **procedure** HeuristicCallback($\boldsymbol{isAlt}, \boldsymbol{\mu}, \boldsymbol{\nu}$)
2: **if** $!\boldsymbol{isAlt}$ **then**
3: **go to** 10
4: **end if**
5: Compute predictions $\widetilde{Q}^{ML}(x^*)$, ϕ^{ML}, ψ^{ML} ▷ Alternating cut strategy
6: **if** $\mu\widetilde{Q}^{ML}(x^*) > \theta^*$ **then**
7: Add a heuristic continuous L-shaped mono-cut
8: **return**
9: **end if**
10: Compute prediction $Q^{ML}(x^*)$ ▷ Integer L-shaped method
11: **if** $\nu Q^{ML}(x^*) \leq \theta^*$ **then**
12: **if** $cx^* + dz^* + \theta^* < UB$ **then**
13: $UB \leftarrow cx^* + cx^* + \theta^*$ ▷ Update upper bound
14: $(x^{**}, z^{**}) \leftarrow (x^*, z^*)$ ▷ Update incumbent solution
15: **return**
16: **end if**
17: **else**
18: Add a heuristic integer L-shaped cut
19: **return**
20: **end if**
21: **return**
22: **end procedure**

Implementation

C-language bindings launch GPU computations returning ML predictions

ML-L-SHAPED

ML-AlternatingCut-L-Shaped

Compute prediction of $\tilde{Q}(x^*), \phi, \psi$

Shift coefficient: Control bias against rejection of valid first-stage integral candidate solutions

$$\textbf{if } \nu\widetilde{Q}^{ML}(x^*) > \theta^*$$

Algorithm 2 Benders decomposition: Heuristic callback

1: **procedure** HeuristicCallback($\boldsymbol{isAlt}, \boldsymbol{\mu}, \boldsymbol{\nu}$)
2:     **if** $!\boldsymbol{isAlt}$ **then**
3:         **go to** 10
4:     **end if**
5:     Compute predictions $\widetilde{Q}^{ML}(x^*)$, ϕ^{ML}, ψ^{ML} ▷ Alternating cut strategy
6:     **if** $\mu\widetilde{Q}^{ML}(x^*) > \theta^*$ **then**
7:         Add a heuristic continuous L-shaped mono-cut
8:         **return**
9:     **end if**
10:     Compute prediction $Q^{ML}(x^*)$ ▷ Integer L-shaped method
11:     **if** $\nu Q^{ML}(x^*) \leq \theta^*$ **then**
12:         **if** $cx^* + dz^* + \theta^* < UB$ **then**
13:             $UB \leftarrow cx^* + cx^* + \theta^*$ ▷ Update upper bound
14:             $(x^{**}, z^{**}) \leftarrow (x^*, z^*)$ ▷ Update incumbent solution
15:             **return**
16:         **end if**
17:     **else**
18:         Add a heuristic integer L-shaped cut
19:         **return**
20:     **end if**
21:     **return**
22: **end procedure**

Implementation

C-language bindings launch GPU computations returning ML predictions

ML-L-SHAPED

- Feasible solution guarantee: in (unlikely) event of failure, resolve using decreasing values of μ and ν
- Two-phase variants
  - Exact: warm start with heuristic solution
  - Warm start with heuristic solution and a probabilistic lower bound (10% one-sided Chebyshev lower confidence bound based on the distribution of exact first-stage values in distinct dataset)

Algorithm 2 Benders decomposition: Heuristic callback

1: **procedure** HEURISTICCALLBACK($\boldsymbol{isAlt}, \boldsymbol{\mu}, \boldsymbol{\nu}$)
2:     **if** $!\boldsymbol{isAlt}$ **then**
3:         **go to** 10
4:     **end if**
5:     Compute predictions $\widetilde{Q}^{ML}(x^*)$, ϕ^{ML}, ψ^{ML} ▷ Alternating cut strategy
6:     **if** $\mu\widetilde{Q}^{ML}(x^*) > \theta^*$ **then**
7:         Add a heuristic continuous L-shaped mono-cut
8:         **return**
9:     **end if**
10:     Compute prediction $Q^{ML}(x^*)$ ▷ Integer L-shaped method
11:     **if** $\nu Q^{ML}(x^*) \leq \theta^*$ **then**
12:         **if** $cx^* + dz^* + \theta^* < UB$ **then**
13:             $UB \leftarrow cx^* + cx^* + \theta^*$ ▷ Update upper bound
14:             $(x^{**}, z^{**}) \leftarrow (x^*, z^*)$ ▷ Update incumbent solution
15:             **return**
16:         **end if**
17:     **else**
18:         Add a heuristic integer L-shaped cut
19:         **return**
20:     **end if**
21:     **return**
22: **end procedure**

In exact version, alternating cuts designed to avoid costly computations of $Q(x^*)$

ALGORITHMS — REMARKS

- Learning to predict $Q(x^*)$ is easier than learning to predict $\tilde{Q}(x^*)$, ϕ and ψ
- Predictions are very fast to compute (a few milliseconds)
  - Invariant with respect to number of scenarios
  - Nearly constant across instances and these tasks
- A priori favours the ML-based matheuristic version of the standard integer L-shaped method over alternating cut strategy (except when first-stage problem is hard)

Generation of training/validation data for supervised learning

{instance, solution} examples

Instances

1. Parametrize (deterministic and stochastic problem data)
2. Pseudo-random sampling

Solutions

\$\$\$ a) Solve (expectation over all scenarios)

$Q(x)$

$\tilde{Q}(x), \phi, \psi$

\$ b) Solve for each scenario independently

(Larsen et al., 2021)

GENERAL REMARKS ON ML

- Training/validation data distribution should cover problem instances that are relevant to the application at hand (simulated and/or historical data)
- Input structure
  - Instance description
  - Size reduction and normalization of values
- Output structure
  - Integer L-shaped cuts (output in $\mathbb{R}$)
  - Continuous L-shaped cuts
    - Size reduction (naive – potentially large size)

4

Extensive numerical study:

Large speedups when there is a large number of scenarios

Optimality gaps close to zero

PROBLEM CLASSES

- Benchmark instances from Angulo et al. (2016) available in SIPLIB (Ahmed et al., 2015)
- Stochastic Server Location Problem SSLP (n,m,k)
  - Locate *n* servers to satisfy *m* customers, k scenarios
  - Good candidate for the proposed methodology
- Stochastic Multiple Binary Knapsack Problem

Stochastic Server Location Problem – SSLP(n,m,k)

Relatively easy 1st stage

All second-stage coefficients are deterministic except right-hand side of some constraints

Relatively **hard 2nd stage**

Stochastic Multiple Binary Knapsack Problem – SMKP

Relatively hard 1st stage

All second-stage coefficients are deterministic except those in the objective function

Relatively **easy 2nd stage**

Hardest instances
* from the literature
(and make 2nd
stage harder)

→

Parametrize and
generate data for
training / validation

→

Supervised
learning

Basic deep learning
(feed forward nets)

Prediction time: few
milliseconds

→

Results

Test performance on
100 instances

SSLP(15,45,15)* SSLPF(15,45,150) SSLPF(15,80,150) **SSLP(10,50,2000)***	Input: server capacities and coupling binaries ($\mathbb{N}^{20}$ or $\mathbb{N}^{30}$) Output: $\mathbb{R}$	Std MAE < 1%
SMKP(29)* **SMKP(30)*** (No solution in Angulo et al. (2016) 20 scenarios	Naive Input: $\mathbb{N}^{600}$, reduced to $\mathbb{R}^5$ Output: reduced to $\mathbb{R}^7$	Alt MAE < 7.5% Std MAE < 1%

RESULTS — KEY TAKEAWAYS

- SSLP (ML-Standard-L-Shaped)
  - Average speed up: x11 - x167 compared to exact
  - First-stage solution quality: average optimality gaps < 2% (median < 0.000)

Scenarios

- SMKP (ML-AlternatingCuts-L-Shaped)
  - x6-x7 compared to exact, but PH is x8-x14 compared to ML-L-Shaped
  - First-stage solution quality: average optimality gaps < 0.08%, PH slightly worse (~0.2%)
    - Speed of our method is invariant wrt the number of scenarios, while PH is not

RESULTS — SSLP

- ML-based matheuristic version of the standard integer L-shaped method
- Average speed up
  - x11 - x167 compared to exact
- First-stage solution quality
  - Excluding index, average optimality gaps < 2% (median < 0.000)

Computing time (s)

Problem family	Our		Exact		PH	
	avg	std. err	avg	std. err	avg	std. err
(10,50,2000)	0.93	(0.08)	156.06	(2.89)	224.44	(12.29)
(10,50,2000)index	0.85	(0.08)	–	–	–	–
(15,45,15)	0.45	(0.01)	5.25	(0.11)	24.80	(2.26)
(15,45,150)	0.55	(0.01)	34.95	(0.57)	36.69	(2.08)
(15,80,15)	4.12	(0.29)	58.55	(3.15)	48.88	(4.59)

Optimality gaps (%)

Problem family	Our		PH	
	avg	std. err	avg	std. err
(10,50,2000)	0.006	(0.003)	0.078	(0.011)
(10,50,2000)index	2.609	(0.242)	–	–
(15,45,15)	0.064	(0.019)	0.005	(0.001)
(15,45,150)	1.943	(1.150)	0.053	(0.006)
(15,80,15)	0.075	(0.018)	0.119	(0.052)

RESULTS — SSLP

Number of integral second-stage problems

Problem family	Alt-L Quantiles 0.05	0.5	0.95	Avg	ML-L-Shaped Quantiles 0.05	0.5	0.95	Avg	ML-L-Shaped/Alt-L ratio Quantiles 0.05	0.5	0.95	Avg
SSLPF(10,50,2000)	56.00	65.00	82.90	66.73 (0.88)	381.00	417.00	455.95	419.17 (2.73)	514.06%	643.20%	756.01%	636.07% (7.14%)
SSLPF-indx(10,50,2000)	56.00	65.00	82.90	66.73 (0.88)	327.10	402.00	467.00	401.16 (3.98)	475.16%	621.40%	733.87%	608.65% (8.06%)
SSLPF(15,45,15)	52.00	63.00	79.95	64.06 (0.96)	1075.05	1243.50	1601.55	1289.88 (17.59)	1563.66%	1973.50%	2749.83%	2052.07% (38.41%)
SSLPF(15,45,150)	57.05	70.00	86.00	70.29 (0.84)	1006.25	1259.00	1582.85	1266.55 (24.91)	1289.20%	1791.57%	2556.04%	1828.42% (42.47%)
SSLPF(15,80,15)	37.10	72.00	89.95	70.26 (1.48)	4952.95	6052.00	13922.15	6641.81 (221.40)	6587.86%	8504.23%	18138.67%	9875.65% (362.88%)

Number of relaxed second-stage problems

Problem family	Alt-L Quantiles 0.05	0.5	0.95	Avg	ML-L-Shaped Quantiles 0.05	0.5	0.95	Avg	ML-L-Shaped/Alt-L ratio Quantiles 0.05	0.5	0.95	Avg
SSLPF(10,50,2000)	364.00	406.00	460.95	408.13 (2.97)	0.00	0.00	0.00	0.00 (0.00)	0.00%	0.00%	0.00%	0.00% (0.00%)
SSLPF-indx(10,50,2000)	364.00	406.00	460.95	408.13 (2.97)	0.00	0.00	0.00	0.00 (0.00)	0.00%	0.00%	0.00%	0.00% (0.00%)
SSLPF(15,45,15)	933.15	1047.50	1347.20	1084.63 (14.04)	0.00	0.00	0.00	0.00 (0.00)	0.00%	0.00%	0.00%	0.00% (0.00%)
SSLPF(15,45,150)	924.70	1089.00	1392.25	1115.20 (15.40)	0.00	0.00	0.00	0.00 (0.00)	0.00%	0.00%	0.00%	0.00% (0.00%)
SSLPF(15,80,15)	5432.45	6234.00	7711.50	6308.44 (65.66)	0.00	0.00	0.00	0.00 (0.00)	0.00%	0.00%	0.00%	0.00% (0.00%)

Standard L-Shaped version of ML-L-Shaped

Number of integral second-stage problems comparable to number of relaxed second-stage problems of exact method with alternating cuts

RESULTS — SSLP

Total time spent in integral second-stage problems (ms)

Problem family	Alt-L Quantiles 0.05	0.5	0.95	Alt-L Avg	ML-L-Shaped Quantiles 0.05	0.5	0.95	ML-L-Shaped Avg	ML-L-Shaped/Alt-L ratio Quantiles 0.05	0.5	0.95	ML-L-Shaped/Alt-L ratio Avg
SSLPF(10,50,2000)	26439.20	36704.00	63863.85	42256.11 (2867.43)	568.05	622.00	709.55	632.82 (6.44)	1.02%	1.77%	2.32%	1.69% (0.04%)
SSLPF-indx(10,50,2000)	26439.20	36704.00	63863.85	42256.11 (2867.43)	530.25	643.50	772.35	647.42 (7.89)	1.09%	1.73%	2.34%	1.73% (0.04%)
SSLPF(15,45,15)	269.50	511.50	1434.75	633.63 (34.56)	1606.10	1845.50	2268.75	1876.13 (21.33)	141.95%	357.32%	684.90%	370.39% (16.43%)
SSLPF(15,45,150)	2140.10	3851.00	7380.75	4097.17 (166.85)	1539.75	1891.50	2339.10	1886.55 (33.91)	27.87%	48.97%	87.26%	51.92% (1.82%)
SSLPF(15,80,15)	2492.65	9765.00	84545.40	20430.97 (2999.29)	4882.90	5599.50	10086.90	5921.83 (131.59)	8.76%	57.08%	245.03%	90.41% (10.39%)

High speed offsets the larger number of integral second-stage problems

Total time spent in relaxed second-stage problems (ms)

Problem family	Alt-L Quantiles 0.05	0.5	0.95	Alt-L Avg	ML-L-Shaped Quantiles 0.05	0.5	0.95	ML-L-Shaped Avg	ML-L-Shaped/Alt-L ratio Quantiles 0.05	0.5	0.95	ML-L-Shaped/Alt-L ratio Avg
SSLPF(10,50,2000)	591539.95	667071.00	749049.35	664969.61 (5065.41)	0.00	0.00	0.00	0.00 (0.00)	0.00%	0.00%	0.00%	0.00% (0.00%)
SSLPF-indx(10,50,2000)	591539.95	667071.00	749049.35	664969.61 (5065.41)	0.00	0.00	0.00	0.00 (0.00)	0.00%	0.00%	0.00%	0.00% (0.00%)
SSLPF(15,45,15)	22277.30	25686.50	34244.55	26816.72 (480.28)	0.00	0.00	0.00	0.00 (0.00)	0.00%	0.00%	0.00%	0.00% (0.00%)
SSLPF(15,45,150)	142335.25	172721.50	223728.10	177046.35 (2736.33)	0.00	0.00	0.00	0.00 (0.00)	0.00%	0.00%	0.00%	0.00% (0.00%)
SSLPF(15,80,15)	180629.90	209324.00	250452.65	210419.66 (2140.48)	0.00	0.00	0.00	0.00 (0.00)	0.00%	0.00%	0.00%	0.00% (0.00%)

RESULTS — SMKP

- ML-based matheuristic version of the L-shaped method with alternating cuts
- Average speed up
  - x6-x7 compared to exact, but PH is x8-x14 compared to ours
- First-stage solution quality
  - Average optimality gaps < 0.08%, PH slightly worse (~0.2%)
  - Speed of our method is invariant wrt the number of scenarios, while PH is not

Computing time (s)

Problem family	Our		Exact		PH	
	avg	std. err	avg	std. err	avg	std. err
(29) 20 scenarios	175.56	(18.20)	1237.13	(124.28)	21.17	(0.33)
(30) 20 scenarios	328.41	(56.51)	2137.45	(257.23)	22.49	(1.00)
(29) 2000 scenarios	–	–	–	–	484.57	(98.27)
(30) 2000 scenarios	–	–	–	–	372.60	(80.58)

Optimality gaps (%)

Problem family	Our		PH	
	avg	std. err	avg	std. err
(29)	0.008	(0.002)	0.223	(0.009)
(30)	0.005	(0.001)	0.224	(0.008)

CONCLUSIONS

- Replacing costly computations by fast ML predictions
  - Large reductions in computing time compared to best performing exact method, especially when second-stage problems are hard / large number of scenarios
  - Online prediction time invariant to the number of scenarios (but not offline data generation)
  - High-quality solution
- First version of the paper – arXiv:2205.00897
- Future work: sample efficiency, account for prediction errors, real-world problems

Thank you!

emma.frejinger@umontreal.ca

emmafrejinger.org

- Ahmed, S., Garcia, R., Kong, N., Ntaimo, L., Parija, G., Qiu, F., and Sen, S. SIPLIB: A stochastic integer programming test problem library, 2015. URL: https://www2.isye.gatech.edu/~sahmed/siplib
- Angulo, G., Ahmed, S., and Dey, S. S. Improving the integer L-shaped method. INFORMS Journal on Computing, 28(3):483–499, 2016.
- Carøe, C. C. and Schultz, R. Dual decomposition in stochastic integer programming. Operations Research Letters, 24(1):37–45, 1999.
- Rockafellar, R. T. and Wets, R. J.-B. Scenarios and policy aggregation in optimization under uncertainty. Mathematics of Operations Research, 16(1): 119–147, 1991.
- Watson, J.-P. and Woodruff, D. L. Progressive hedging innovations for a class of stochastic mixed-integer resource allocation problems. Computational Management Science, 8(4):355–370, 2011.