Recent Trends in Combinatorial Optimization Augmented Machine Learning

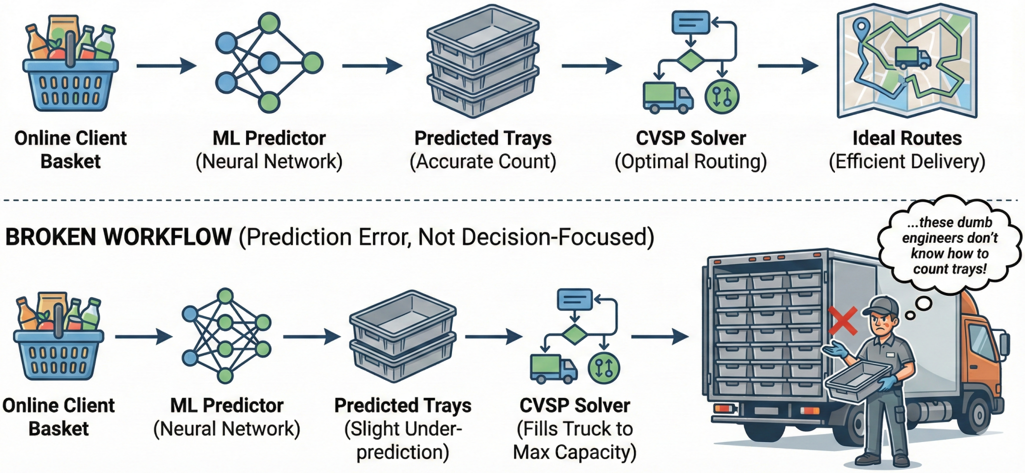
Axel Parmentier

Cermics École Nationale des Ponts et Chaussées

February 2, 2026

Why Non-Decision-Focused Learning Breaks Workflows

IDEAL WORKFLOW (Decision-Focused)



Online Client Basket → **ML Predictor** (Neural Network) → **Predicted Trays** (Accurate Count) → **CVSP Solver** (Optimal Routing) → **Ideal Routes** (Efficient Delivery)

BROKEN WORKFLOW (Prediction Error, Not Decision-Focused)

Online Client Basket → **ML Predictor** (Neural Network) → **Predicted Trays** (Slight Under-prediction) → **CVSP Solver** (Fills Truck to Max Capacity) →

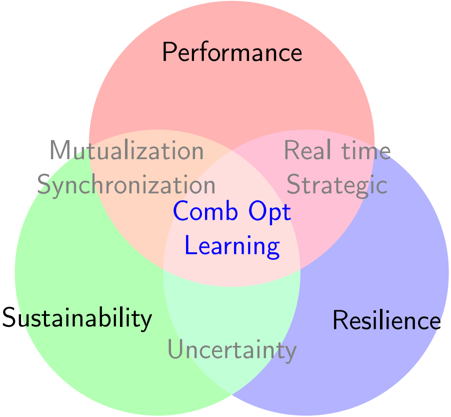
The Data-Driven Revolution in Operations Research

OR algorithms are **embedded in data-driven workflows**

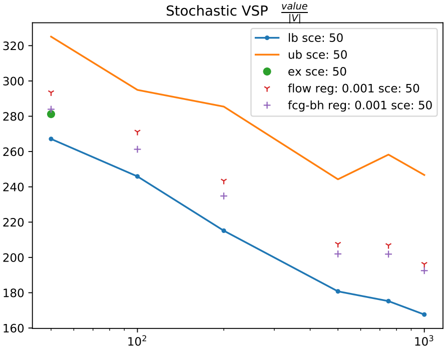
Exploit data to tame uncertainty

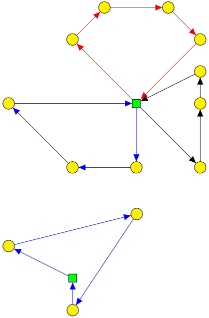
- **More Efficient:** Optimize resource allocation
- **More Robust:** Handle unexpected disruptions
- **More Sustainable:** Reduce waste and empty miles / handle Sustainable Energies

Separating learning from decision can break worklows



Value in OR comes from decreasing marginal costs

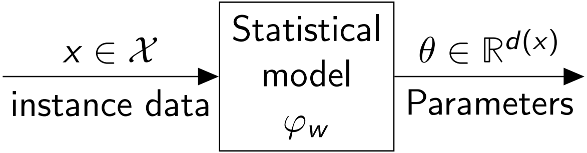




- Pure ML fails on Combinatorial Optimization
- OR researchers tend to focus on CO to make algorithms scale

The Trap: Predict-then-Optimize

First estimate the statistical model



$x \in \mathcal{X}$

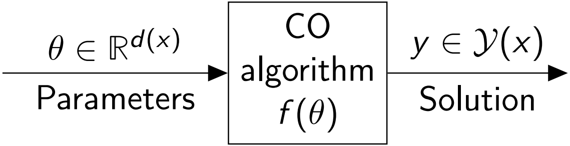
instance data

Statistical model φ_w

$\theta \in \mathbb{R}^{d(x)}$

Parameters

Then solve the (stochastic) CO problem



$\theta \in \mathbb{R}^{d(x)}$

Parameters

CO algorithm $f(\theta)$

$y \in \mathcal{Y}(x)$

Solution

Learning algorithms ignore application

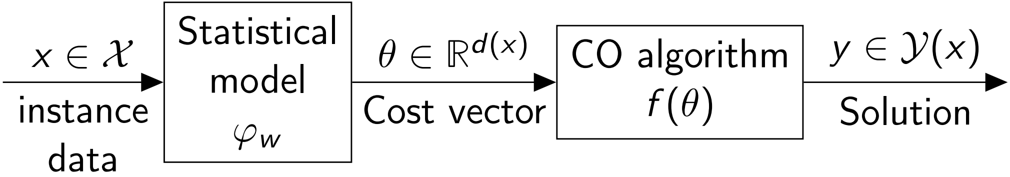
Training set $(x_1, \bar{\theta}_1), \dots, (x_n, \bar{\theta}_n)$
Loss $\mathcal{L}(\theta, \bar{\theta})$

Learning problem

$$\min_w \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}(\varphi_w(x_i), \bar{\theta}_i)$$

Small prediction errors can lead to catastrophic decisions

Combinatorial Optimization Augmented Machine Learning



$x \in \mathcal{X}$

instance data

Statistical model φ_w

$\theta \in \mathbb{R}^{d(x)}$

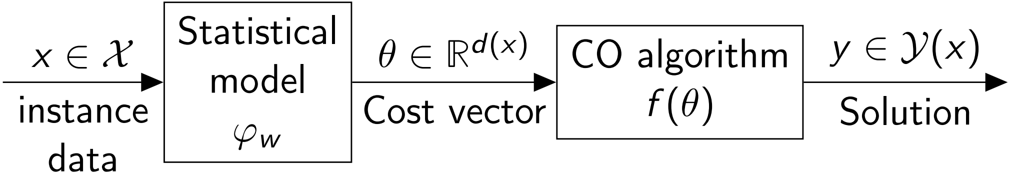
Cost vector

CO algorithm $f(\theta)$

$y \in \mathcal{Y}(x)$

Solution

Combinatorial Optimization Augmented Machine Learning



$x \in \mathcal{X}$

instance data

Statistical model φ_w

$\theta \in \mathbb{R}^{d(x)}$

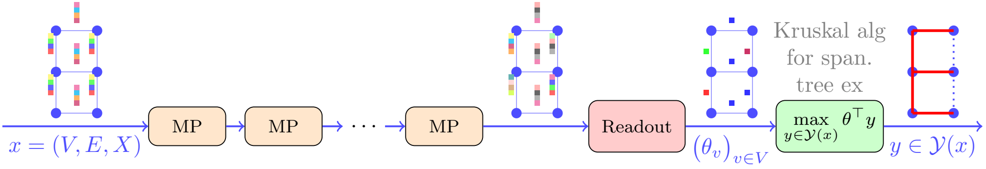
Cost vector

CO algorithm $f(\theta)$

$y \in \mathcal{Y}(x)$

Solution

Trained by decision focused learning $\min\limits_{w} \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}(\varphi_w(x_i), \bar{y}_i)$.

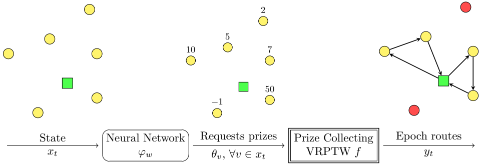


Plan

1. Applications in OR and architectures
   - Contextual stochastic optimization
   - Dynamic problems
2. Supervised learning for static problems
3. Empirical risk minimization for contextual stochastic optimization
4. Learning for dynamic problems

Settings and architectures

Multistage stochastic optimization



State x_t

Neural Network φ_w

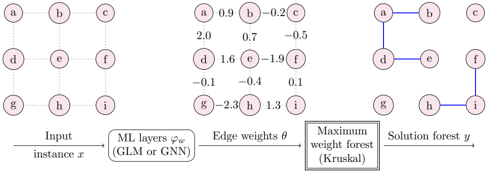
Requests prizes $\theta_v, \forall v \in x_t$

Prize Collecting VRPTW f

Epoch routes y_t

EURO NeurIPS challenge 2022. Baty et al. 2024; Greif et al. 2024

Contextual stochastic optimization



Input instance x

ML layers φ_w (GLM or GNN)

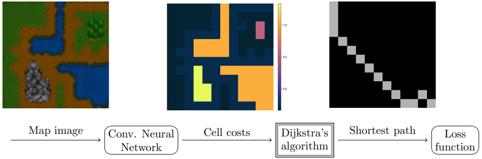
Edge weights θ

Maximum weight forest (Kruskal)

Solution forest y

Donti, Amos, and Kolter 2017; Dalle et al. 2022

Data-driven optimization



Map image

Conv. Neural Network

Cell costs

Dijkstra's algorithm

Shortest path

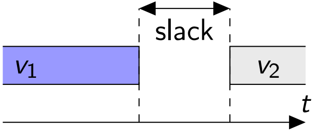
Loss function

Pogančić et al. 2019; Berthet et al. 2020

Plan

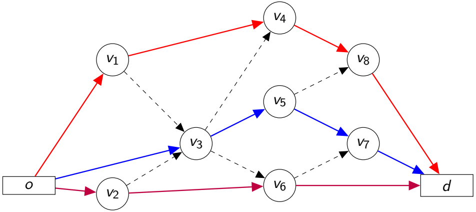
1. Applications in OR and architectures
   - Contextual stochastic optimization
   - Dynamic problems
2. Supervised learning for static problems
3. Empirical risk minimization for contextual stochastic optimization
4. Learning for dynamic problems
   - Supervised learning for dynamic problems
   - Structured Reinforcement Learning

Resilience: Stochastic Vehicle Scheduling Problem



$$c_p = \text{vehicle cost} + \mathbb{E}\big(\text{propagated delay cost}\big)$$
$$= c^{\text{veh}} + \frac{1}{|\Omega|}\sum_{\omega\in\Omega}\sum_{v\in P}\xi_v^P(\omega)$$

Reduce costs dues to delay propagation along rotations



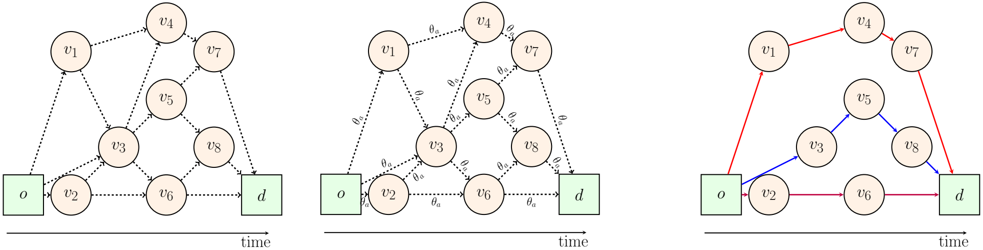
$$\min \sum_{P\in\mathcal{P}} c_P z_y$$
$$\sum_{P\ni v} y_P = 1 \quad \forall v$$
$$y_P \in \{0,1\}$$

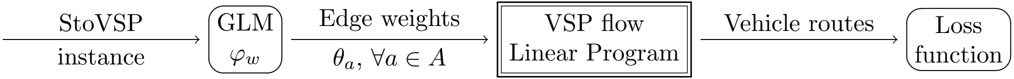
Challenges

Even with simplest delay models

- No tractable moment formulation
- SAA does not scale (exact $|V| \leq 80$, heuristics $|V| \leq 400$)
- Cannot afford more than a single deterministic resolution

Decision aware learning for Contextual Stochastic VSP





StoVSP instance → GLM φ_w → Edge weights $\theta_a, \forall a \in A$ → VSP flow Linear Program → Vehicle routes → Loss function

Excellent performance on large scale instances[1]

Enables being contextual

[1] A. P. (Apr. 2021). "Learning to Approximate Industrial Problems by Operations Research Classic Problems". In: *Operations Research*; Guillaume Dalle et al. (July 2022). *Learning with Combinatorial Optimization Layers: A Probabilistic Approach*. eprint: 2207.13513.

Contextual stochastic combinatorial optimization

Contextual stochastic optimization problem[2]

$$\min_{\pi\in\mathcal{H}} \mathcal{R}(\pi) \quad \text{where} \quad \mathcal{R}(\pi) = \mathbb{E}_{(\boldsymbol{x},\,\xi),\, \boldsymbol{y}\sim\pi(\cdot|\boldsymbol{x})}\Big[\boldsymbol{c}\big(\underset{\text{context in } \mathcal{X}}{\boldsymbol{x}},\, \underset{\text{decision in } \mathcal{Y}(x)}{\boldsymbol{y}},\, \overset{\text{noise correlated with } \boldsymbol{x}}{\xi}\big)\Big]$$

Assumptions:

- we have an efficient algorithm to solve

$$\min_{y\in\mathcal{Y}(x)} c\big(x(\omega), y, \xi(\omega)\big) + \langle\theta|y\rangle$$

- $\mathcal{Y}(x)$ is finite (but exponentially large)
- we have access to a dataset $\mathcal{D} = (x_i, \xi_i)_{i\in[N]}$

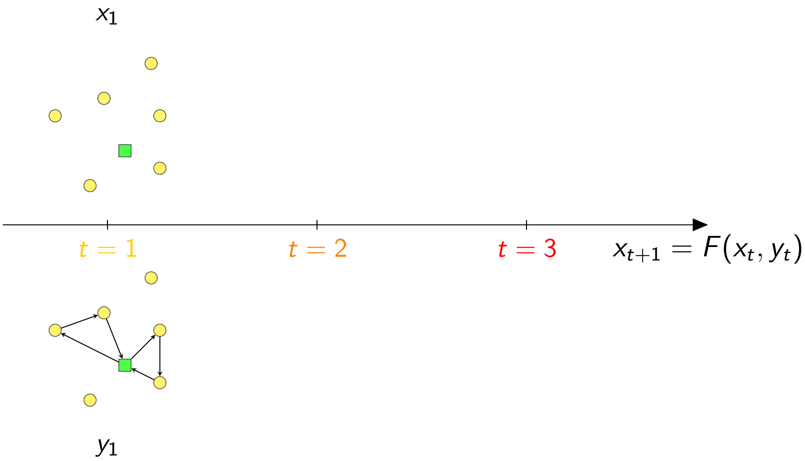
[2] Utsav Sadana et al. (Mar. 2024). "A Survey of Contextual Optimization Methods for Decision-Making under Uncertainty". In: *European Journal of Operational Research*. issn: 0377-2217. doi: 10.1016/j.ejor.2024.03.020. (Visited on 07/12/2024).

Plan

1. Applications in OR and architectures
   - Contextual stochastic optimization
   - Dynamic problems
2. Supervised learning for static problems
3. Empirical risk minimization for contextual stochastic optimization
4. Learning for dynamic problems
   - Supervised learning for dynamic problems
   - Structured Reinforcement Learning

Dynamic Vehicle Routing Problem

State
$x_t \in \mathcal{X}$
set of customers



Decision
$y_t \in \mathcal{Y}(x_t)$
set of routes

Dynamic Vehicle Routing Problem

State

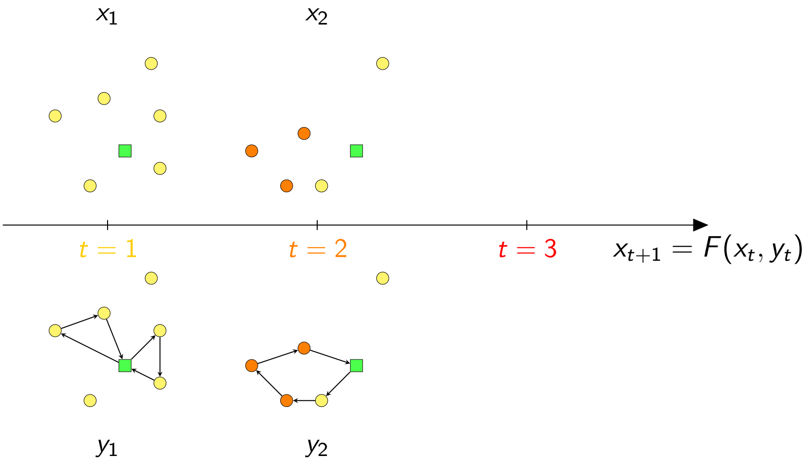
$x_t \in \mathcal{X}$

set of customers

Decision

$y_t \in \mathcal{Y}(x_t)$

set of routes



Dynamic Vehicle Routing Problem

State

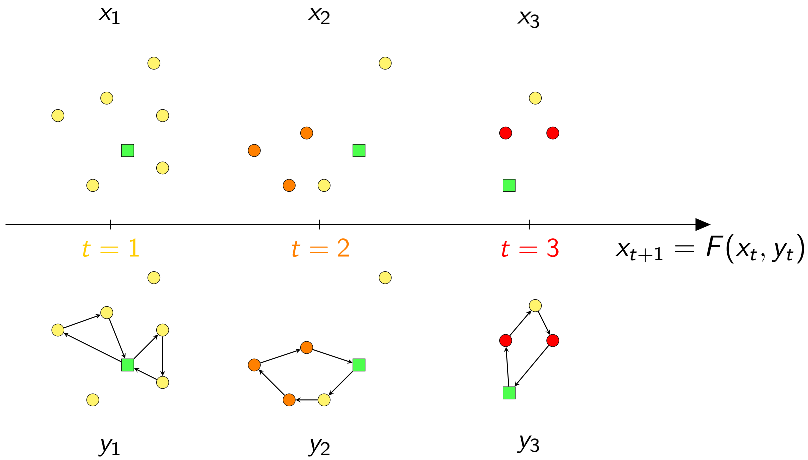
$x_t \in \mathcal{X}$

set of customers

Decision

$y_t \in \mathcal{Y}(x_t)$

set of routes



Dynamic VRPTW

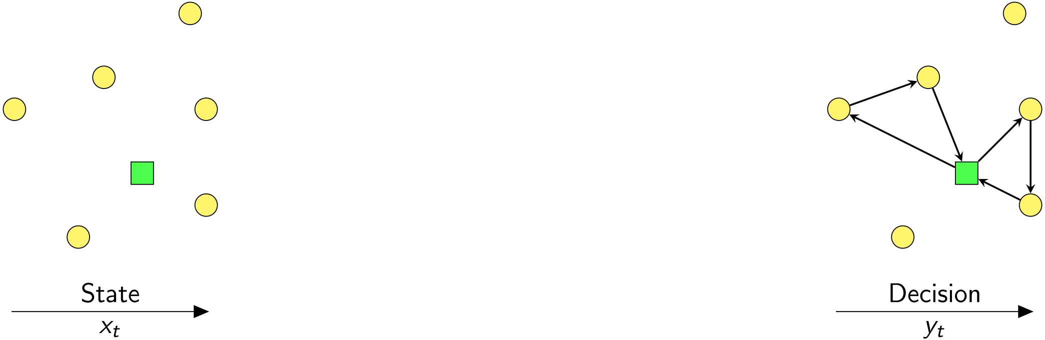
A solution of this problem is a **policy**:

$$\begin{aligned}\pi\colon \mathcal{X} &\to \mathcal{Y}\\ x_t &\mapsto y_t\end{aligned}$$

Objective: find $\pi^\star$, serving all customers before end of horizon, and minimizing total cost

$$\pi^\star = \arg\min_{\pi} \mathbb{E}\left[\sum_{\text{epochs } t} \text{total cost of routes in decision } y_t = \pi(x_t)\right]$$

Policy that won the EURO-NeurIPS challenge[3]



State

x_t

Decision

y_t

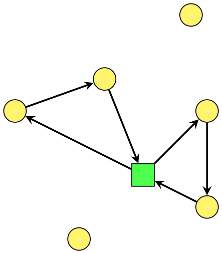
[3] Léo Baty et al. (Feb. 2024). “Combinatorial Optimization-Enriched Machine Learning to Solve the Dynamic Vehicle Routing Problem with Time Windows”. In: *Transportation Science*. issn: 0041-1655. doi: `10.1287/trsc.2023.0107`. (Visited on 07/18/2024).

Policy that won the EURO-NeurIPS challenge[3]

Epoch decisions can be seen as the solution of a Prize Collecting VRPTW:

- Serving customers is optional
- Serving customer v gives prize θ_v
- **Objective**: $\max \text{profit} = \text{prize} - \text{route}\,cost$

$$\max_{y\in\mathcal{Y}(x_t)} \underbrace{\sum_{(u,v)\in x_t^2} \theta_v y_{u,v}}_{\text{total prize}} - \underbrace{\sum_{(u,v)\in x_t^2} c_{u,v} y_{u,v}}_{\text{total routes cost}} .$$



$$\xrightarrow[y_t]{\text{Decision}}$$

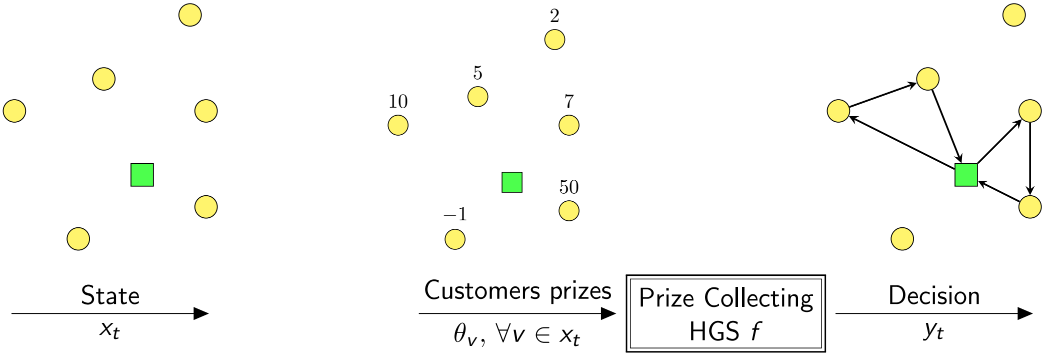
- **Algorithm**: Prize Collecting Hybrid Genetic Search

$\Rightarrow$ Combinatorial Optimization layer f

[3] Léo Baty et al. (Feb. 2024). "Combinatorial Optimization-Enriched Machine Learning to Solve the Dynamic Vehicle Routing Problem with Time Windows". In: *Transportation Science*. issn: 0041-1655. doi: `10.1287/trsc.2023.0107`. (Visited on 07/18/2024).

Policy that won the EURO-NeurIPS challenge[3]

Difficulty: no natural way of computing meaningful prizes



State
x_t

Customers prizes
$\theta_v, \forall v \in x_t$

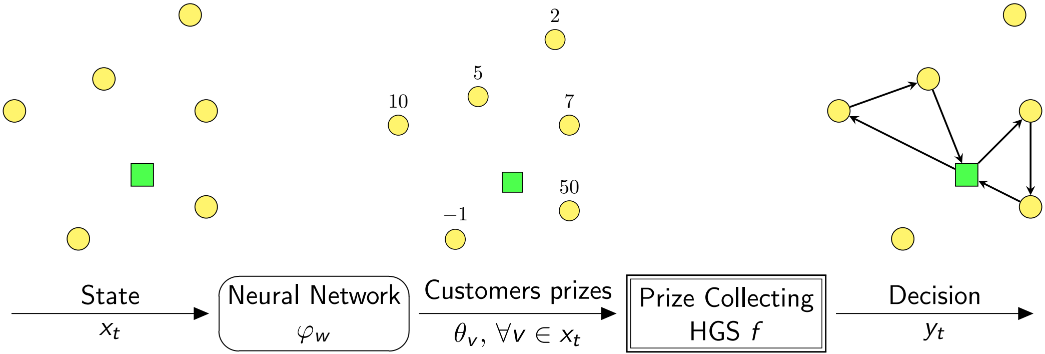
$\boxed{\text{Prize Collecting HGS } f}$

Decision
y_t

[3] Léo Baty et al. (Feb. 2024). "Combinatorial Optimization-Enriched Machine Learning to Solve the Dynamic Vehicle Routing Problem with Time Windows". In: *Transportation Science*. issn: 0041-1655. doi: 10.1287/trsc.2023.0107. (Visited on 07/18/2024).

Policy that won the EURO-NeurIPS challenge[3]

Solution: use a neural network to predict request prizes $\theta = \varphi_w(x_t)$



State x_t

Neural Network φ_w

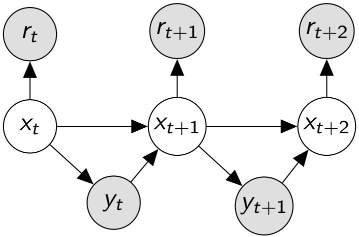
Customers prizes $\theta_v, \forall v \in x_t$

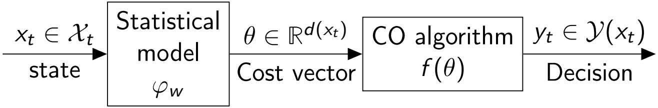
Prize Collecting HGS f

Decision y_t

[3] Léo Baty et al. (Feb. 2024). "Combinatorial Optimization-Enriched Machine Learning to Solve the Dynamic Vehicle Routing Problem with Time Windows". In: *Transportation Science*. issn: 0041-1655. doi: 10.1287/trsc.2023.0107. (Visited on 07/18/2024).

Policy for multistage stochastic optimization





$x_t \in \mathcal{X}_t$

state

Statistical model φ_w

$\theta \in \mathbb{R}^{d(x_t)}$

Cost vector

CO algorithm $f(\theta)$

$y_t \in \mathcal{Y}(x_t)$

Decision

Neural network with a CO layer: policy for MDPs with large state *and decision* spaces.

$$\min_w \mathbb{E}_\pi \sum_t r_t \qquad \text{with} \qquad \pi_{w,t} : \mathcal{X}_t \to \mathcal{Y}_t$$

Plan

1. Applications in OR and architectures
2. Supervised learning for static problems
3. Empirical risk minimization for contextual stochastic optimization
4. Learning for dynamic problems

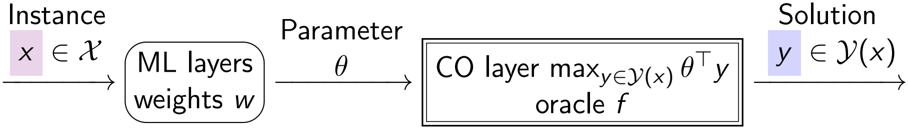
Policy encoded by neural networks with CO layers

Goal: find a policy π that minimizes

$$\min_{\pi\in\mathcal{H}} \mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}_{\boldsymbol{x}},\, \boldsymbol{y}\sim\pi(\cdot|\boldsymbol{x})}\Big[\overset{\text{cost function}}{c^0}\big(\underset{\text{instance in } \mathcal{X}}{\boldsymbol{x}};\underset{\text{decision in } \mathcal{Y}(x)}{\boldsymbol{y}}\big)\Big]$$

$\mathbb{P}_{\boldsymbol{x}}$ unknown but access to $x_1,\ldots,x_n$.

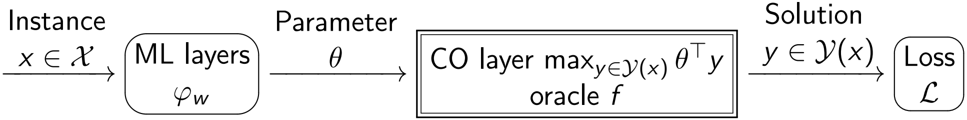
Model choice: we restrict ourselves to policies π_w based on



We thus seek weights w that minimize the risk

$$\min_{w} \mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}_{\boldsymbol{x}},\, \boldsymbol{y}\sim\pi_w(\cdot|\boldsymbol{x})}\Big[c^0\big(\boldsymbol{x};\boldsymbol{y}\big)\Big]$$

End-to-end learning: two paradigms



Instance $x \in \mathcal{X}$

ML layers φ_w

Parameter θ

CO layer $\max_{y \in \mathcal{Y}(x)} \theta^\top y$ oracle f

Solution $y \in \mathcal{Y}(x)$

Loss $\mathcal{L}$

Empirical risk minimization

Dataset: $\mathcal{D} = (x_i)_{i \in [N]}$

Learning problem:

$$\min_w \frac{1}{N} \sum_{i=1}^N c^0\Big(x_i; f\left(\varphi_w(x_i)\right)\Big)$$

Supervised learning

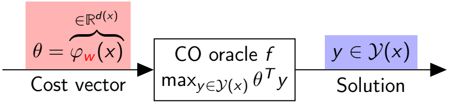
Dataset: $\mathcal{D} = (x_i, \bar{y}_i)_{i \in [N]}$

Learning problem:

$$\min_w \frac{1}{N} \sum_{i=1}^N \mathcal{L}\Big(x_i; f\left(\varphi_w(x_i)\right), \bar{y}_i\Big)$$

→ We would like both to rely on stochastic gradient descent (SGD)

Lack of informative derivatives



$\theta = \overbrace{\varphi_w(x)}^{\in\mathbb{R}^{d(x)}}$

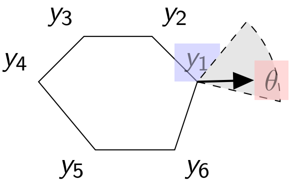
Cost vector

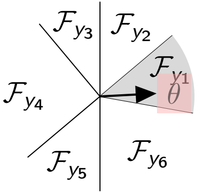
CO oracle f

$\max_{y\in\mathcal{Y}(x)} \theta^T y$

$y \in \mathcal{Y}(x)$

Solution

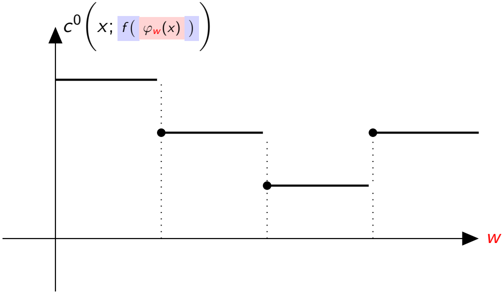




Piecewise-constant learning problem

$$\frac{1}{N}\sum_{i=1}^{N} c^0\Big(x_i; f\big(\varphi_w(x_i)\big)\Big)$$

$$\frac{1}{N}\sum_{i=1}^{N} \mathcal{L}\Big(x_i; f\big(\varphi_w(x_i)\big), \bar{y}_i\Big)$$

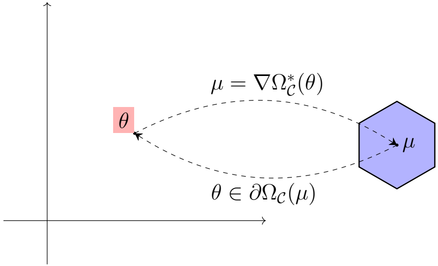


Smoothing by regularization or perturbation

$$\max_{\mu\in\mathcal{C}(x)} \theta^T\mu - \Omega(\mu), \quad \mathcal{C}(x) = \mathrm{conv}\left(\mathcal{Y}(x)\right)$$

Ex. 1: $\Omega(\mu) = ||\mu||_2^2 + \mathbb{I}_{\mathcal{C}(x)}(\mu)$

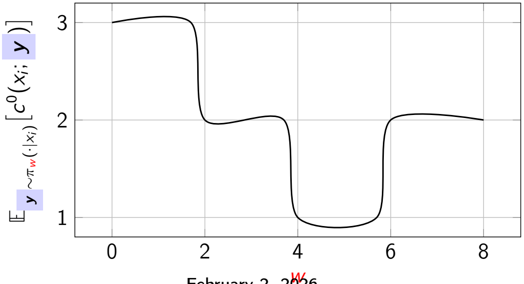
Ex. 2: $\Omega^*(\theta) = \mathbb{E}_{\boldsymbol{Z}}[\max_{\mu\in\mathcal{C}(x)}(\theta + \varepsilon \boldsymbol{Z})^\top \mu]$


$\mu = \nabla\Omega^*_{\mathcal{C}}(\theta)$
θ
μ
$\theta \in \partial\Omega_{\mathcal{C}}(\mu)$

Smoothed learning problem

$$\frac{1}{N}\sum_{i=1}^{N}\mathbb{E}_{\boldsymbol{y}\sim\pi_w(\cdot|x_i)}\left[c^0(x_i;\boldsymbol{y})\right]$$

$$\frac{1}{N}\sum_{i=1}^{N}\mathbb{E}_{\boldsymbol{y}\sim\pi_w(\cdot|x_i)}\left[\mathcal{L}(x_i;\boldsymbol{y},\bar{y}_i)\right]$$



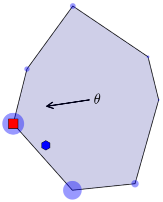
Supervised learning: Fenchel-Young losses[4]

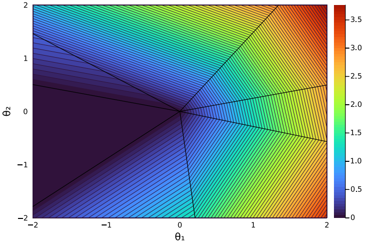
Properties that make SGD tractable

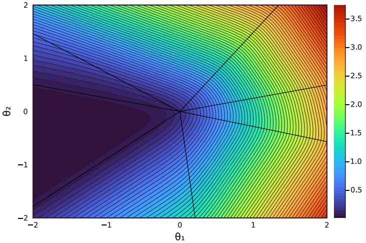
$$\mathcal{L}_\Omega(\theta;\bar{y}) = \overbrace{\max_{y\in\mathcal{C}(x)} \left(\langle\theta|y\rangle - \Omega(y)\right) - \left(\langle\theta|\bar{y}\rangle - \Omega(\bar{y})\right)}^{\text{Non-optimality of } \bar{y} \text{ as a solution of the regularized prediction problem}}$$

$$\mathcal{L}_\Omega(\theta;\bar{y}) = \Omega^*(\theta) + \Omega(\bar{y}) - \langle\theta|\bar{y}\rangle$$

- $\mathcal{L}_\Omega(\theta;\bar{y}) \geq 0$
- $\mathcal{L}_\Omega(\theta;\bar{y}) = 0 \Leftrightarrow \bar{y} = \nabla\Omega^*(\theta)$
- Convex in θ
- $\nabla_\theta\mathcal{L}_\Omega(\theta;\bar{y}) = \widehat{f}_\Omega(\theta) - \bar{y}$

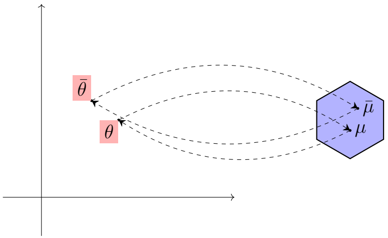


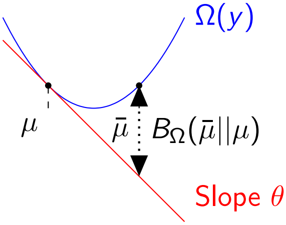




[4] Blondel, Martins, and Niculae 2020.

Fenchel-Young loss as a primal-dual Bregman divergence





$$B_\Omega(\bar{\mu}||\mu) = \Omega(\bar{\mu}) - \Omega(\mu) - \langle \nabla\Omega(\mu)|\bar{\mu}-\mu\rangle \quad \text{and} \quad B_\Omega(\bar{\mu}||\mu) = \mathcal{L}_\Omega(\theta;\bar{\mu}) = B_{\Omega^*}(\theta||\bar{\theta})$$

$$\min_{\mu\in\mathcal{C}} \frac{1}{N}\sum_{i=1}^N B_\Omega(\bar{\mu}_i||\mu) \Leftrightarrow \min_{\theta\in\mathbb{R}^d} \frac{1}{N}\sum_{i=1}^N \mathcal{L}_\Omega(\theta;\bar{\mu}_i)$$

Choice of the regularization: State of the art

$$\nabla_\theta \ell_\Omega(\theta, \bar{y}) = \nabla\Omega^*(\theta) - \bar{y} = \operatorname*{arg\,max}_{\mu\in\mathcal{C}} \theta^\top \mu - \Omega(\mu)$$

Perturbation (Berthet et al. 2020)

$$\Omega^*(\theta) = \mathbb{E}_{\boldsymbol{z}}\Big[\max_{y\in\mathcal{Y}}(\theta + \boldsymbol{z})^\top y\Big]$$

$$\nabla\Omega^*(\theta) = \mathbb{E}_{\boldsymbol{z}}\Big[\operatorname*{arg\,max}_{y\in\mathcal{Y}}(\theta + \boldsymbol{z})^\top y\Big]$$

MonteCarlo estimate of $\nabla\Omega^*(\theta)$:
Sample $z_1, \ldots, z_k$ and solve *exactly*

$$\max_{y\in\mathcal{Y}}(\theta + z_i)^\top y$$

Negentropy (e.g., Wainwright, Jordan, et al. 2008)

$$\Omega(\mu) = \min_{q\in\Delta^{\mathcal{Y}}}\Big\{-\mathrm{H}(q) \colon \mathbb{E}_{\boldsymbol{y}\sim q}[\boldsymbol{y}] = \mu\Big\}$$

$$\nabla\Omega^*(\theta) = \mathbb{E}_{\boldsymbol{y}\sim p(\cdot|\theta)}[\boldsymbol{y}]$$

Exact $\nabla\Omega^*(\theta)$ if $\max_{y\in\mathcal{Y}} \theta^\top y$ tractable by dynamic programming (Mensch and Blondel 2018)

$$H(q) = -\sum_{y\in\mathcal{Y}} q(y)\log q(y)$$

$$p(y|\theta) = \frac{e^{\theta^\top y}}{Z(\theta)} \text{ where } Z(\theta) = \sum_{y\in\mathcal{Y}} e^{\theta^\top y}$$

From Simulated Annealing to Metropolis Hasting

Simulated annealing (SA) with neigh. $\mathcal{N}$

$$\max_{y\in\mathcal{Y}} \theta^\top y$$

```
Inputs: $\theta\in\mathbb{R}^d$, $(0)\in\mathcal{Y}$, $(t_k)$, $K\in\mathbb{N}$, $\mathcal{N}$, $q$
for $k = 0 : K$ do
    Sample a neighbor in $\mathcal{N}(y^{(k)})$:
    $y' \sim q\left(y^{(k)}, \cdot\right)$
    $U \sim \mathcal{U}([0,1])$
    $\Delta^{(k)} \leftarrow \langle\theta, y'\rangle + \varphi(y') - \langle\theta, y^{(k)}\rangle - \varphi(y^{(k)})$
    $p^{(k)} \leftarrow \qquad \exp\left(\Delta^{(k)}/t_k\right)$
    If $U \leq p^{(k)}$, accept move: $y^{(k+1)} \leftarrow y'$
    If $U > p^{(k)}$, reject move: $y^{(k+1)} \leftarrow y^{(k)}$
end for
Output: $\widehat{y}(\theta) \approx y^{(K)}$
```

From Simulated Annealing to Metropolis Hasting

Simulated annealing (SA) with neigh. $\mathcal{N}$

$$\max_{y \in \mathcal{Y}} \theta^\top y$$

is Metropolis Hasting (MH) MCMC for

$$\mathbb{E}_{\boldsymbol{y} \sim p(\cdot|\theta)}[\boldsymbol{y}]$$

where p is the exponential family on $\mathcal{Y}$

$$p(y|\theta) = e^{\theta^\top y - A(\theta)} = \frac{e^{\theta^\top y}}{Z(\theta)}$$

where $Z(\theta) = \sum_{y \in \mathcal{Y}} e^{\theta^\top y}$ and $A(\theta) = \log Z(\theta)$

Used in the 1980s to study SA convergence (**faigle_convergence_1988**; Mitra, Romeo, and Sangiovanni-Vincentelli 1986)

```
Inputs: $\theta \in \mathbb{R}^d$, $(0) \in \mathcal{Y}$, $(t_k)$, $K \in \mathbb{N}$, $\mathcal{N}$, $q$
for $k = 0 : K$ do
    Sample a neighbor in $\mathcal{N}(y^{(k)})$:
    $y' \sim q\left(y^{(k)}, \cdot\right)$
    $\alpha(y^{(k)}, y') \leftarrow 1$ (SA) or
    $\alpha(y^{(k)}, y') \leftarrow \frac{q(y', y^{(k)})}{q(y^{(k)}, y')}$ (MH)
    $U \sim \mathcal{U}([0,1])$
    $\Delta^{(k)} \leftarrow \langle \theta, y' \rangle + \varphi(y') - \langle \theta, y^{(k)} \rangle - \varphi(y^{(k)})$
    $p^{(k)} \leftarrow \alpha(y^{(k)}, y') \exp\left(\Delta^{(k)} / t_k\right)$
    If $U \leq p^{(k)}$, accept move: $y^{(k+1)} \leftarrow y'$
    If $U > p^{(k)}$, reject move: $y^{(k+1)} \leftarrow y^{(k)}$
end for
Output: $\widehat{y}(\theta) \approx y^{(K)}$ (SA) or
$\bar{y}_t(\theta) = \mathbb{E}_{\pi_{\theta,t}}[Y] \approx \frac{1}{K} \sum_{k=1}^K y^{(k)}$ (MH)
```

MH gives stochastic gradient, solves regularized problem

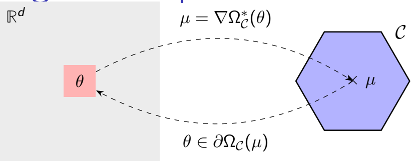
$$\underbrace{\mathbb{E}_{\boldsymbol{y}\sim p(\cdot|\theta)}[\boldsymbol{y}]}_{\textit{Expectation}} = \underbrace{\nabla A(\theta)}_{\substack{\text{Grad. of}\\ \text{logpartition}}}$$

MH gives stochastic gradient, solves regularized problem

$$\underbrace{\mathbb{E}_{\boldsymbol{y}\sim p(\cdot|\theta)}[\boldsymbol{y}]}_{\textit{Expectation}} = \underbrace{\nabla A(\theta)}_{\substack{\text{Grad. of}\\ \text{logpartition}}} = \nabla \overbrace{\log \underbrace{Z(\theta)}_{\sum_{y\in\mathcal{Y}} e^{\theta^\top y}}}^{A(\theta)} = \sum_{y\in\mathcal{Y}} y \underbrace{\frac{e^{\theta^\top y}}{Z(\theta)}}_{p(y|\theta)}$$

MH gives stochastic gradient, solves regularized problem

$$\underbrace{\mathbb{E}_{\boldsymbol{y}\sim p(\cdot|\theta)}[\boldsymbol{y}]}_{\textit{Expectation}} = \underbrace{\nabla A(\theta)}_{\text{Grad. of logpartition}}$$


$\mathbb{R}^d$
$\mu = \nabla\Omega_{\mathcal{C}}^*(\theta)$
$\mathcal{C}$
θ
μ
$\theta \in \partial\Omega_{\mathcal{C}}(\mu)$

Introducing the Fenchel conjugate of A

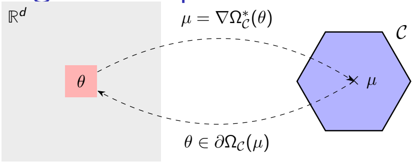
$$\Omega(\mu) \doteq A^*(\mu) = \max_{\theta} \theta^\top \mu - A(\theta)$$

as regularization, , denoting $\mathcal{C} = \operatorname{conv}\mathcal{Y}$, we get

$$\underbrace{\mathbb{E}_{\boldsymbol{y}\sim p(\cdot|\theta)}[\boldsymbol{y}]}_{\text{MH (i.e., SA) for this inference problem}} = \underbrace{\nabla\Omega^*(\theta)}_{\text{get stochastic gradients}} = \underbrace{\operatorname*{arg\,max}_{\mu\in\mathcal{C}} \theta^\top \mu - \Omega(\mu)}_{\text{which are near optimal solutions of regularized problem}}$$

MH gives stochastic gradient, solves regularized problem

$$\underbrace{\mathbb{E}_{\boldsymbol{y}\sim p(\cdot|\theta)}[\boldsymbol{y}]}_{\textit{Expectation}} = \underbrace{\nabla A(\theta)}_{\substack{\text{Grad. of}\\ \text{logpartition}}}$$


$\mathbb{R}^d$
$\mu = \nabla\Omega^*_{\mathcal{C}}(\theta)$
$\mathcal{C}$
θ
μ
$\theta \in \partial\Omega_{\mathcal{C}}(\mu)$

Introducing the Fenchel conjugate of A

$$\Omega(\mu) \doteq A^*(\mu) = \max_{\theta} \theta^\top \mu - A(\theta)$$

as regularization, , denoting $\mathcal{C} = \operatorname{conv} \mathcal{Y}$, we get

$$\underbrace{\mathbb{E}_{\boldsymbol{y}\sim p(\cdot|\theta)}[\boldsymbol{y}]}_{\substack{\text{MH (i.e., SA)}\\ \text{for this}\\ \text{inference problem}}} = \underbrace{\nabla\Omega^*(\theta)}_{\substack{\text{get}\\ \text{stochastic}\\ \text{gradients}}} = \underbrace{\operatorname*{arg\,max}_{\mu\in\mathcal{C}} \theta^\top \mu - \Omega(\mu)}_{\substack{\text{which are near}\\ \text{optimal solutions of}\\ \text{regularized problem}}}$$

Characterization of Ω

$$\begin{aligned}\Omega(\mu) &= -\mathrm{H}\big(p(\cdot|\theta)\big)\\ &= \min_{q\in\Delta^{\mathcal{Y}}} \Big\{ -\mathrm{H}(q) \colon \mathbb{E}_{\boldsymbol{y}\sim q}[\boldsymbol{y}] = \mu \Big\}\end{aligned}$$

where

$$\mathrm{H}(q) = -\sum_{y\in\mathcal{Y}} q(y)\log(q(y)).$$

Classic results on variational inference in exponential families Wainwright, Jordan, et al. 2008

SA as MH with Negentropy

$$\underbrace{\mathbb{E}_{\boldsymbol{y}\sim p(\cdot|\theta)}[\boldsymbol{y}]}_{\substack{\text{MH (i.e., SA)}\\ \text{for this}\\ \text{inference problem}}} = \underbrace{\nabla\Omega^*(\theta)}_{\substack{\text{get}\\ \text{stochastic}\\ \text{gradients}}} = \underbrace{\operatorname*{arg\,min}_{\mu\in\mathcal{C}} \theta^\top\mu - \Omega(\mu)}_{\substack{\text{which are near}\\ \text{optimal solutions of}\\ \text{regularized problem}}}$$

Parameter estimations with training set $\bar{y}_1, \ldots, \bar{y}_N$, and $\bar{Y}_N = \frac{1}{N}\sum_{i=1}^N y_i$

$$\hat{\boldsymbol{\theta}}_{n+1} = \hat{\boldsymbol{\theta}}_n + \gamma_{n+1}\left[\bar{Y}_N - \overbrace{\frac{1}{K_{n+1}}\sum_{k=1}^{K_{n+1}} \boldsymbol{y}^{(n+1,\,k)}}^{\text{MH estimate}}\right]$$

$\boldsymbol{y}^{(n+1,k)}$: k-th iterate of MH with temp t, direction $\hat{\boldsymbol{\theta}}_n$, initialized at $\boldsymbol{y}^{(n+1,1)} = \boldsymbol{y}^{(n,K_n)}$

Proposition SGD convergence with MH estimate (Vivier Ardisson, Blondel, P., 2025)

Under some classic assumptions for SGD, $\hat{\theta}_n \xrightarrow{a.s.} \boldsymbol{\theta}_N^\star$

Plan

1. Applications in OR and architectures
2. Supervised learning for static problems
3. Empirical risk minimization for contextual stochastic optimization
4. Learning for dynamic problems

Contextual stochastic combinatorial optimization[5]

Consider the risk

$$\min_{\pi\in\mathcal{H}} \mathcal{R}(\pi) \quad \text{where} \quad \mathcal{R}(\pi) = \mathbb{E}_{(\boldsymbol{x},\xi),\, \boldsymbol{y}\sim\pi(\cdot|\boldsymbol{x})}\Big[c\big(\underset{\text{context in } \mathcal{X}}{\boldsymbol{x}}, \underset{\text{decision in } \mathcal{Y}(x)}{\boldsymbol{y}}, \overset{\text{noise correlated with } \boldsymbol{x}}{\xi}\big)\Big]$$

Assumptions:

- we have an efficient algorithm to solve

$$\min_{y\in\mathcal{Y}(x)} c\big(x(\omega), y, \xi(\omega)\big) + \langle\theta|y\rangle$$

- $\mathcal{Y}(x)$ is finite (but exponentially large)
- we have access to a dataset $\mathcal{D} = (x_i, \xi_i)_{i\in[N]}$

Classic decomposition approaches from stochastic optimization (progressive hedging, L-shaped method) may not scale

[5] Sadana et al. 2024.

Contextual stochastic combinatorial optimization[5]

Consider the risk

$$\min_{\pi\in\mathcal{H}} \mathcal{R}(\pi) \quad \text{where} \quad \mathcal{R}(\pi) = \mathbb{E}_{(\boldsymbol{x},\xi),\,\boldsymbol{y}\sim\pi(\cdot|\boldsymbol{x})}\Big[c\big(\underbrace{\boldsymbol{x}}_{\text{context in } \mathcal{X}}, \underbrace{\boldsymbol{y}}_{\text{decision in } \mathcal{Y}(x)}, \overbrace{\xi}^{\text{noise correlated with } \boldsymbol{x}}\big)\Big]$$

Assumptions:

- we have an efficient algorithm to solve

$$\min_{y\in\mathcal{Y}(x)} c\big(x(\omega), y, \xi(\omega)\big) + \langle\theta|y\rangle$$

- $\mathcal{Y}(x)$ is finite (but exponentially large)
- we have access to a dataset $\mathcal{D} = (x_i, \xi_i)_{i\in[N]}$

Classic decomposition approaches from stochastic optimization (progressive hedging, L-shaped method) may not scale

Our Approach: Louis Bouvier et al. (2025). “Primal-dual algorithm for contextual stochastic combinatorial optimization”. In: *arXiv preprint arXiv:2505.04757*

[5] Sadana et al. 2024.

A coordination heuristic

Given a training set $(x_1, \xi_i), \dots, (x_n, \xi_n)$, start with imitation learning

$$\min_x \frac{1}{n} \sum_{i=1}^{n} \ell\Big(\varphi_w(x_i), \bar{y}_i\Big) \quad \text{where} \quad \bar{y}_i = \operatorname*{arg\,min}_{y \in \mathcal{Y}(x_i)} c(x_i, y, \xi_i)$$

Then minimize a linear combination of (anticipative) objective and prediction

$$\bar{y}_i = \operatorname*{arg\,min}_{y \in \mathcal{Y}(x_i)} c(x_i, y, \xi_i) + \kappa \underbrace{\left(-\varphi_w(x_i)^\top y\right)}_{\substack{\text{non regularized} \\ \ell(\varphi_w(x_i), y)\text{constant}}}$$

Then update w

$$\min_w \sum_i \ell\big(\varphi_w(x_i), y_i\big)$$

and iterate

A coordination heuristic

Given a training set $(x_1, \xi_i), \dots, (x_n, \xi_n)$, start with imitation learning

$$\min_x \frac{1}{n} \sum_{i=1}^n \ell\Big(\varphi_w(x_i), \bar{y}_i\Big) \quad \text{where} \quad \bar{y}_i = \operatorname*{arg\,min}_{y \in \mathcal{Y}(x_i)} c(x_i, y, \xi_i)$$

Then minimize a linear combination of (anticipative) objective and prediction

$$\bar{y}_i = \operatorname*{arg\,min}_{y \in \mathcal{Y}(x_i)} c(x_i, y, \xi_i) + \kappa \underbrace{\left(-\varphi_w(x_i)^\top y\right)}_{\substack{\text{non regularized} \\ \ell(\varphi_w(x_i), y)\text{constant}}}$$

Then update w

$$\min_w \sum_i \ell\big(\varphi_w(x_i), y_i\big)$$

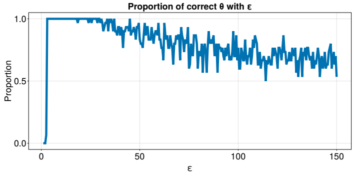
and iterate

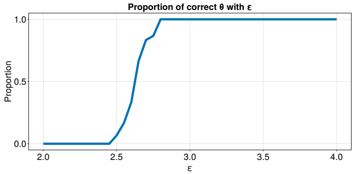
which happens to be an exact algorithm

Applications

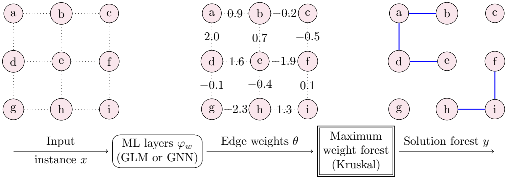
Toy problem

	Scenario ξ_1	Scenario ξ_2	Scenario ξ_3
Solution 0	4	-1	-2
Solution 1	0	0	0

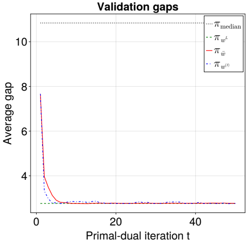


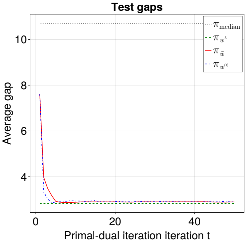


Two-stage minimum weight spanning tree



Input instance x → ML layers φ_w (GLM or GNN) → Edge weights θ → Maximum weight forest (Kruskal) → Solution forest y





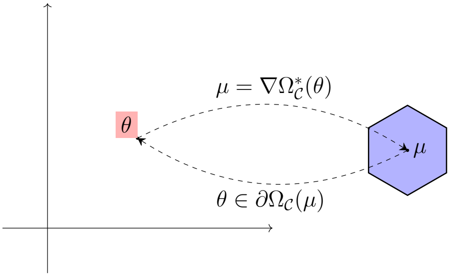
Surrogate problem on the distribution space

$$\min_{y \in \mathcal{Y}} \theta^\top y$$

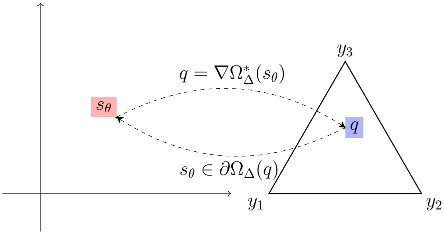
is equivalent to

$$\min_{q \in \Delta^{\mathcal{Y}}} \mathbb{E}(\theta^\top y | q) = \underbrace{\theta^\top Y}_{s_\theta^\top} q$$

$$Y = \Big(y_1 \Big| \dots \Big| y_{|\mathcal{Y}|} \Big)$$



$\mu = \nabla \Omega_{\mathcal{C}}^*(\theta)$
θ
μ
$\theta \in \partial \Omega_{\mathcal{C}}(\mu)$



y_3
$q = \nabla \Omega_{\Delta}^*(s_\theta)$
s_θ
q
$s_\theta \in \partial \Omega_{\Delta}(q)$
y_1
y_2

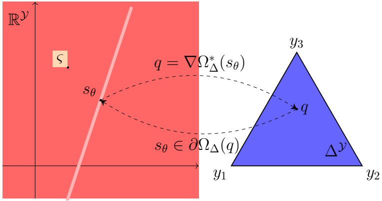
Empirical risk minimization and surrogate problem

Any cost function $c(x, \cdot, \xi)$

- vector γ in $\mathbb{R}^{|\mathcal{Y}|}$, the dual of $\Delta^{\mathcal{Y}}$

Surrogate problem minimizes:

- scenario decisions costs
- scenario decision divergence to policy



$\mathbb{R}^{\mathcal{Y}}$
ς
s_θ
$q = \nabla\Omega^*_{\Delta}(s_\theta)$
$s_\theta \in \partial\Omega_{\Delta}(q)$
q
y_3
y_1
y_2
$\Delta^{\mathcal{Y}}$

$$\min_{w\in\mathcal{W}} R_N(\pi_w) := \min_{w} \frac{1}{N}\sum_{i=1}^{N} \overbrace{\mathbb{E}_{\boldsymbol{y}\sim\pi_w(\cdot|\boldsymbol{x}_i)}\Big[c\,(x_i, \boldsymbol{y}, \xi_i)\Big]}^{\substack{\text{Scenario } i \text{ cost} \\ \text{under policy } \pi_w}} = \min_{w} \frac{1}{N}\sum_{i=1}^{N} \langle\, \gamma_i \,|\nabla\Omega^*_{\Delta(x_i)}\big(Y(x_i)^\top\varphi_w(x_i)\big)\rangle$$

cost vector $\big(c(x_i, y, \xi_i)\big)_{y\in\mathcal{Y}}$

$$\min_{w,q_\otimes} \mathcal{S}_N(s_w; q_\otimes) := \min_{w,q_\otimes} \frac{1}{N}\sum_{i=1}^{N} \underbrace{\mathbb{E}_{\boldsymbol{y}\sim q_i}\Big[c\,(x_i, \boldsymbol{y}, \xi_i)\Big]}_{\substack{\text{independent pb} \\ \text{per scenario } i}} + \kappa\, \underbrace{\mathcal{L}_{\Omega_{\Delta(x_i)}}\Big(Y(x_i)^\top\varphi_w(x_i); q_i\Big)}_{\substack{\text{coupled by FY} \\ \text{loss to policy}}}$$

Alternating minimization scheme

Surrogate problem

$$\min_{w,q_\otimes} \mathcal{S}_N(s_w; q_\otimes) := \min_{w,q_\otimes} \frac{1}{N}\sum_{i=1}^{N} \mathbb{E}_{\boldsymbol{y}\sim q_i}\Big[c(x_i,\boldsymbol{y},\xi_i)\Big] + \kappa \mathcal{L}_{\Omega_{\Delta(x_i)}}\Big(Y(x_i)^\top \varphi_w(x_i); q_i\Big)$$

Alternating minimization algorithm

$$q_i^{(t+1)} = \operatorname*{arg\,min}_{q_i\in\Delta(x_i)} \mathbb{E}_{\boldsymbol{y}\sim q_i}\Big[c(x_i,\boldsymbol{y},\xi_i)\Big] + \kappa \mathcal{L}_{\Omega_{\Delta(x_i)}}\Big(Y(x_i)^\top \varphi_{\bar{w}^{(t)}}(x_i); q_i\Big) \qquad \text{(decomposition)}$$

$$\bar{w}^{(t+1)} \in \operatorname*{arg\,min}_{w\in\mathcal{W}} \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}_{\Omega_{\mathcal{C}(x_i)}}\Big(\varphi_w(x_i); Y(x_i) q_i^{(t+1)}\Big) \qquad \text{(coordination)}$$

Proposition Bouvier, Prunet, Leclère, P., 2025

For well-chosen regularizations, we get tractable alternating minimization updates

High-level strategy: minimizing the surrogate function

Given some technical assumptions / settings restrictions

Theorem *Convergence to surrogate optimum* Bouvier, Prunet, Leclère, P., 2025

Provided some technical assumptions, the (average) iterates $q_\otimes^{(t)}$ concide with those of mirror descent and converge to $\min\limits_{q_\otimes} \min_{s_\otimes} \mathcal{S}_N(s_\otimes, q_\otimes)$

Proposition *Empirical risk bound*, Bouvier, Prunet, Leclère, P., 2025

$$\theta_{\mathcal{S},N} \in \arg\min_{\theta} \min_{q_\otimes} \mathcal{S}_N(s_\theta, q_\otimes) \quad \Longrightarrow \quad \mathcal{R}_N(\theta_{\mathcal{S},N}) - \min_{\theta} \mathcal{R}_N(\theta) \leq \dots$$

Theorem *Generalization bounds*, Aubin-Frankowski, De Castro, P., Rudi, 2024

In the large data regime, $\mathcal{R}(\theta_{\mathcal{S},N}) - \min_\theta \mathcal{R}(\theta) \leq \dots$

Introducing sparse perturbation over distributions

$$F_{\varepsilon,\mathcal{C}}(\theta) = \mathbb{E}[\max_{y\in\mathcal{Y}}(\theta + \varepsilon \boldsymbol{Z})^\top y]$$
$$= \mathbb{E}[\max_{y\in\mathcal{C}}(\theta + \varepsilon \boldsymbol{Z})^\top y]$$

$\rightarrow \Omega_{\varepsilon,\mathcal{C}} = F^*_{\varepsilon,\mathcal{C}}$

Proposition Berthet et al. 2020

- defined over $\mathbb{R}^d$
- strict convexity
- $\nabla_\theta F_{\varepsilon,\mathcal{C}}(\theta) = \mathbb{E}[\arg\max_{y\in\mathcal{C}}(\theta + \varepsilon \boldsymbol{Z})^\top y]$
- $\mathrm{dom}(F^*_{\varepsilon,\mathcal{C}}) = \mathcal{C}$
- $F^*_{\varepsilon,\mathcal{C}}$ Legendre-type

$$F_{\varepsilon,\Delta}(s) = \mathbb{E}[\max_{y\in\mathcal{Y}}(s(y) + \varepsilon \boldsymbol{Z})^\top y]$$
$$= \mathbb{E}[\max_{q\in\Delta}(s + \varepsilon Y^\top \boldsymbol{Z})^\top q]$$

$\rightarrow \Omega_{\varepsilon,\Delta} = F^*_{\varepsilon,\Delta}$

Proposition Bouvier et al. 2025

- defined over $\mathbb{R}^{\mathcal{Y}}$
- strict convexity
- $\nabla_s F_{\varepsilon,\Delta}(s) = \mathbb{E}[\arg\max_{q\in\Delta^{\mathcal{Y}}}(s + \varepsilon Y^\top \boldsymbol{Z})^\top q]$
- $\mathrm{dom}(F^*_{\varepsilon,\Delta}) = \Delta^{\mathcal{Y}}$
- $F^*_{\varepsilon,\Delta}$ Legendre-type

Tractable updates

Using $\Omega_{\varepsilon,\Delta(x)} = F_{\varepsilon,\Delta(x)}(s)^*$

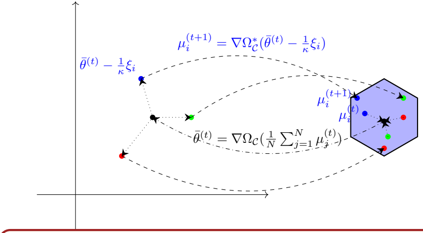
$$\begin{aligned}
\mu_i^{(t+1)} &= Y(x_i)q_i^{(t+1)} \\
&= Y(x_i)\nabla F_{\varepsilon,\Delta(x_i)}\Big(Y(x_i)^\top \varphi_{\bar{w}^{(t)}}(x_i) - \frac{1}{\kappa}\gamma_i\Big) \\
&= \mathbb{E}_{\boldsymbol{Z}}\Big[\operatorname*{arg\,min}_{y_i \in \mathcal{Y}(x_i)} c(x_i, y_i, \xi_i) - \kappa(\varphi_{\bar{w}^{(t)}}(x_i) + \varepsilon \boldsymbol{Z})^\top y_i\Big]
\end{aligned}$$

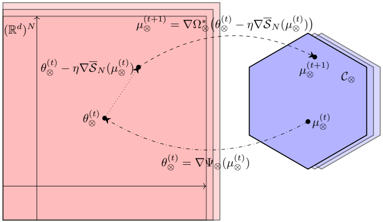
- Swap integration and derivation
- Danskin's theorem
- Dirac on a vertex

Proposition Bouvier, Prunet, Leclère, P., 2025

In the case $\Omega_{\mathcal{C}(x)} := F^*_{\varepsilon,\mathcal{C}(x)}$ and $\Omega_{\Delta(x)} := F^*_{\varepsilon,\Delta(x)}$ we get tractable approximate alternating minimization updates

Link with mirror descent (simplified)


$\mu_i^{(t+1)} = \nabla\Omega_{\mathcal{C}}^*(\bar\theta^{(t)} - \frac{1}{\kappa}\xi_i)$
$\bar\theta^{(t)} - \frac{1}{\kappa}\xi_i$
$\mu_i^{(t+1)}$
$\mu_i^{(t)}$
$\bar\theta^{(t)} = \nabla\Omega_{\mathcal{C}}(\frac{1}{N}\sum_{j=1}^N \mu_j^{(t)})$


$(\mathbb{R}^d)^N$
$\mu_\otimes^{(t+1)} = \nabla\Omega_\otimes^*(\theta_\otimes^{(t)} - \eta\nabla\overline{\mathcal{S}}_N(\mu_\otimes^{(t)}))$
$\theta_\otimes^{(t)} - \eta\nabla\overline{\mathcal{S}}_N(\mu_\otimes^{(t)})$
$\mu_\otimes^{(t+1)}$
$\mathcal{C}_\otimes$
$\theta_\otimes^{(t)}$
$\mu_\otimes^{(t)}$
$\theta_\otimes^{(t)} = \nabla\Psi_\otimes(\mu_\otimes^{(t)})$

Theorem Bouvier, Prunet, Leclère, P., 2025

Our iterates coincide with the ones of mirror descent applied to

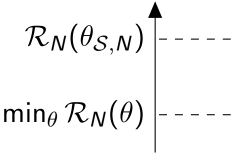
$$\bar{\mathcal{S}}_N(q_\otimes) := \min_{s_\otimes} \mathcal{S}_N(s_\otimes; q_\otimes) = \frac{1}{N}\sum_{i=1}^N \langle \gamma_i | q_i \rangle + \frac{\kappa}{N}\Big[\underbrace{\sum_{i=1}^N \Omega_\Delta(q_i) - N\Omega_\Delta(\frac{1}{N}\sum_{i=1}^N q_i)}_{\text{Jensen gap}}\Big]$$

with a mirror map $\Psi_\otimes$ such that $\Omega_\otimes = \Psi_\otimes + \mathbb{I}_{\Delta_\otimes}$

Bounded non-optimality (in a restricted setting)

$$\mathcal{R}_N(\theta) := R_N\Big(p_{\Omega_\Delta}(\cdot|\theta)\Big)$$

$$\underline{\mathcal{S}_N}(\theta) := \min_{q_\otimes \in \Delta_\otimes} \mathcal{S}_N\Big(s_\theta, q_\otimes\Big) \quad \text{and} \quad \theta_{\mathcal{S},N} \in \arg\min_\theta \underline{\mathcal{S}_N}(\theta)$$



$\mathcal{R}_N(\theta_{\mathcal{S},N})$

$\min_\theta \mathcal{R}_N(\theta)$

Theorem Bouvier, Prunet, Leclère, P., 2025

Let $\theta \in \mathbb{R}^d$, provided that $\nabla\Omega_\Delta^*$ is $\frac{1}{L}$-Lipschitz-continuous with respect to $||\cdot||$

$$|\underline{\mathcal{S}_N}(\theta) - \mathcal{R}_N(\theta)| \leq \frac{3}{2NL\kappa} \sum_{i=1}^{N} ||\gamma_i||^2$$

cost vector $\big(c(x_i, y, \xi_i)\big)_{y\in\mathcal{Y}}$

we deduce that

$\in \arg\min_\theta \underline{\mathcal{S}_N}(\theta)$

$$\mathcal{R}_N(\theta_{\mathcal{S},N}) - \mathcal{R}_N(\theta_{\mathcal{R},N}) \leq \frac{3}{L\kappa N} \sum_{i=1}^{N} ||\gamma_i||^2$$

$\in \arg\min_\theta \mathcal{R}_N(\theta)$

Which guarantees can we obtain for the policy returned by our learning algorithm ?[6]

Get back to the $c^0(x,y)$ setting.

Contextual stochastic optimization: given x, ξ, define

$$c^0(y,x) = c(x,y,\xi)$$

$$\min_{\boldsymbol{w}\in\mathcal{W}} \mathcal{R}_{n\lambda}(h_{\boldsymbol{w}}) \quad \text{with} \quad \mathcal{R}_{n,\lambda}(h_{\boldsymbol{w}}) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_Z\Big\{\big[c^0(\hat{\boldsymbol{y}}(\psi_{\boldsymbol{w}}(X_i) + \lambda Z(X_i)), X_i)\big]\Big\}$$

[6] Pierre-Cyril Aubin-Frankowski et al. (July 2024). *Generalization Bounds of Surrogate Policies for Combinatorial Optimization Problems*. doi: 10.48550/arXiv.2407.17200. arXiv: 2407.17200 [stat]. (Visited on 12/10/2024).

Risks and estimators

$$\bar{\mathcal{R}} = \mathbb{E}\Big[\min_{\boldsymbol{y}\in\mathcal{Y}(x)} c^0(\boldsymbol{y}, X)\Big]$$

$$\mathcal{R}_t(h_{\boldsymbol{w}}) = \mathbb{E}_{X,Z}\Big[c^0\big(\hat{\boldsymbol{y}}(\psi_{\boldsymbol{w}}(X) + tZ(X)), X\big)\Big]$$

$$\mathcal{R}_{n,t}(h_{\boldsymbol{w}}) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_Z\Big[c^0\big(\hat{\boldsymbol{y}}(\psi_{\boldsymbol{w}}(X_i) + tZ(X_i)), X_i\big)\Big]$$

$$\boldsymbol{w}^* = \operatorname*{arg\,min}_{\boldsymbol{w}\in\mathcal{W}} \mathcal{R}_0(h_{\boldsymbol{w}}) \qquad \text{opt. pol}$$

$$\boldsymbol{w}_{n,\lambda} = \operatorname*{arg\,min}_{\boldsymbol{w}\in\mathcal{W}} \mathcal{R}_{n\lambda}(h_{\boldsymbol{w}}) \qquad \text{learn. opt.}$$

$$\boldsymbol{w}_{n,\lambda}^{\mathrm{alg}} : \text{ learning algorithm} \qquad \text{result}$$

$$\begin{aligned}
0 \leq \mathcal{R}_0(h_{\boldsymbol{w}_{n,\lambda}^{\mathrm{alg}}}) - \bar{\mathcal{R}} = {} & \underbrace{\mathcal{R}_0(h_{\boldsymbol{w}_{M,n,\lambda}}) - \mathcal{R}_\lambda(h_{\boldsymbol{w}_{M,n,\lambda}})}_{\text{Pert. bias Theorem}} + \underbrace{\mathcal{R}_\lambda(h_{\boldsymbol{w}_{M,n,\lambda}}) - \mathcal{R}_{n,\lambda}(h_{\boldsymbol{w}_{M,n,\lambda}})}_{\text{Emp. process Theorem}} \\
& + \underbrace{\mathcal{R}_{n,\lambda}(h_{\boldsymbol{w}_{M,n,\lambda}}) - \mathcal{R}_{n,\lambda}(h_{\boldsymbol{w}_{n,\lambda}})}_{\text{Alt. min. alg.}} + \underbrace{\mathcal{R}_{n,\lambda}(h_{\boldsymbol{w}_{n,\lambda}}) - \mathcal{R}_{n,\lambda}(h_{\boldsymbol{w}^\star})}_{\leq 0} \\
& + \underbrace{\mathcal{R}_{n,\lambda}(h_{\boldsymbol{w}^\star}) - \mathcal{R}_\lambda(h_{\boldsymbol{w}^\star})}_{\text{Emp. process Theorem}} + \underbrace{\mathcal{R}_\lambda(h_{\boldsymbol{w}^\star}) - \mathcal{R}_0(h_{\boldsymbol{w}^\star})}_{\text{Pert. bias Theorem}} \\
& + \underbrace{\mathcal{R}_0(h_{\boldsymbol{w}^\star}) - \bar{\mathcal{R}}}_{\text{Model bias.}}
\end{aligned}$$

Theorem Aubin-Frankowski, De Castro, P., and Rudi, 2024

Let $0 \geq 0$ and $\lambda > 0$ be such that $\lambda \geq 0$. Let $\tau \in (0,1)$. Under conditions detailed later, there exists a constant $C > 0$ that depends only on ε, τ and c^0 such that for any $\boldsymbol{w} \in \mathcal{W}$ and $n \geq 1$, one has

$$|\mathcal{R}_0(h_{\boldsymbol{w}}) - \mathcal{R}_\lambda(h_{\boldsymbol{w}})| = C\lambda^\tau \mathrm{polylog}(\lambda) \qquad \text{(Perturbation bias Theorem)}$$

$$|\mathcal{R}_\lambda(h_{\boldsymbol{w}}) - \mathcal{R}_{n,\lambda}(h_{\boldsymbol{w}})| = \mathcal{O}_{\mathbb{P}}\left(\frac{1}{\lambda\sqrt{n}}\right) \qquad \text{(Empirical process Theorem)}$$

where $\mathrm{polylog}(\lambda)$ is a polynomial logarithm term.

Optimizing over λ, we get $\mathcal{R}_0(h_{\boldsymbol{w}^{\mathrm{alg}}_{n,\lambda}}) - \bar{\mathcal{R}} \underset{n\to\infty}{\longrightarrow} \mathcal{R}_0(h_{\boldsymbol{w}^\star}) - \bar{\mathcal{R}}$ in the large data regime.

Plan

1. Applications in OR and architectures

2. Supervised learning for static problems

3. Empirical risk minimization for contextual stochastic optimization

4. Learning for dynamic problems
   - Supervised learning for dynamic problems
   - Structured Reinforcement Learning

Plan

1. Applications in OR and architectures
   - Contextual stochastic optimization
   - Dynamic problems

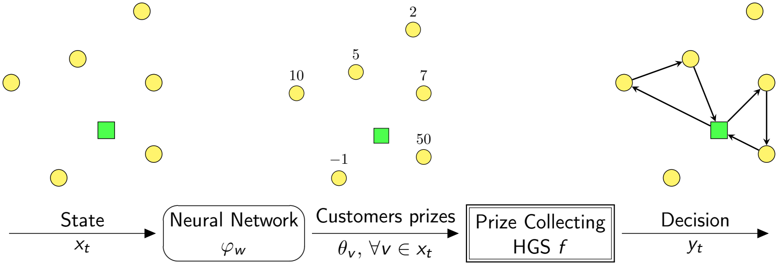
2. Supervised learning for static problems

3. Empirical risk minimization for contextual stochastic optimization

4. Learning for dynamic problems
   - Supervised learning for dynamic problems
   - Structured Reinforcement Learning

Learning dynamic problem policy

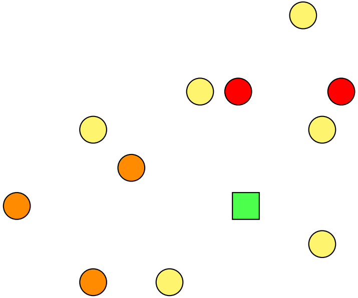
Goal: find parameters w such that our pipeline is a “good” policy.



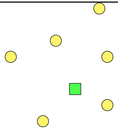
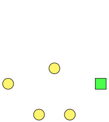
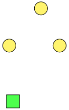
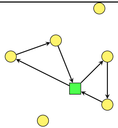
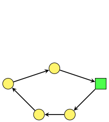
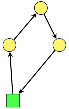
State x_t → Neural Network φ_w → Customers prizes $\theta_v, \forall v \in x_t$ → Prize Collecting HGS f → Decision y_t

$$\hat{w} = \arg\min_{w} \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}(\varphi_w(x^i), \bar{y}^i)$$

Learn to imitate anticipative decisions[7]



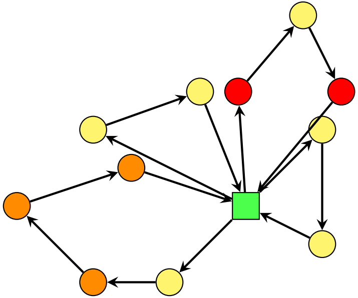
We rebuild the anticipative decisions a posteriori

i	1	2	3
x^i			
$\bar{y}^i$			

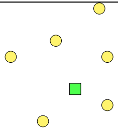
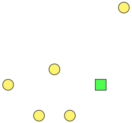
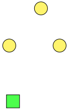
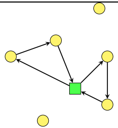
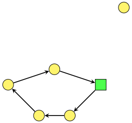
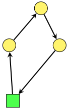
Gives a training set $x_1, y_1, \ldots, x_n, y_n$, and we can then formulate the learning problem as minimizing the Fenchel Young loss.

[7] Léo Baty et al. (Feb. 2024). "Combinatorial Optimization-Enriched Machine Learning to Solve the Dynamic Vehicle Routing Problem with Time Windows". In: *Transportation Science*. issn: 0041-1655. doi: 10.1287/trsc.2023.0107. (Visited on 07/18/2024).

Learn to imitate anticipative decisions[7]



We rebuild the anticipative decisions a posteriori

i	1	2	3
x^i			
$\bar{y}^i$			

Gives a training set $x_1, y_1, \ldots, x_n, y_n$, and we can then formulate the learning problem as minimizing the Fenchel Young loss.

[7] Léo Baty et al. (Feb. 2024). "Combinatorial Optimization-Enriched Machine Learning to Solve the Dynamic Vehicle Routing Problem with Time Windows". In: *Transportation Science*. issn: 0041-1655. doi: 10.1287/trsc.2023.0107. (Visited on 07/18/2024).

Additional ingredient needed on other problems[8]

We should solve (an empirical version of)

$$\min_w \mathbb{E}_{X\sim\delta_w}\Big[\mathcal{L}\big(\varphi_w(X),\delta^*(X)\big)\Big]$$

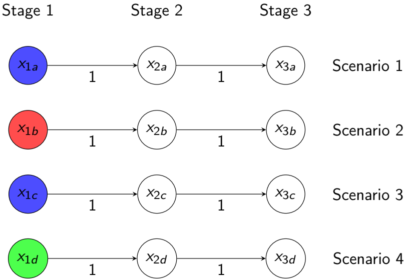
while we solve (an empirical version of)

$$\min_w \mathbb{E}_{X\sim\delta^*}\Big[\mathcal{L}\big(\varphi_w(X),\delta^*(X)\big)\Big]$$

How to build $\mathcal{D}$?

- Several epoch: DAgger $\alpha\delta^* + (1-\alpha)\delta_w$
- Single epoch: Add states from random policy

Why does it work ? Voting policy



- Average across states
- Learning conditional dist. via gen. MLE
- Take mode

[8] Toni Greif et al. (Feb. 2024). *Combinatorial Optimization and Machine Learning for Dynamic Inventory Routing*. arXiv: 2402.04463 [math]. (Visited on 03/04/2024).

Plan

1. Applications in OR and architectures
   - Contextual stochastic optimization
   - Dynamic problems
2. Supervised learning for static problems
3. Empirical risk minimization for contextual stochastic optimization
4. Learning for dynamic problems
   - Supervised learning for dynamic problems
   - Structured Reinforcement Learning

Single step reinforcement learning

Reinforcement learning setting

$$\min_{\pi\in\mathcal{H}} \mathcal{R}(\pi) \quad \text{where} \quad \mathcal{R}(\pi) = \mathbb{E}_{(\boldsymbol{x},\xi),\, \boldsymbol{y}\sim\pi(\cdot|\boldsymbol{x})}\Big[\boldsymbol{c}(\underbrace{\boldsymbol{x}}_{\text{context in } \mathcal{X}}, \underbrace{\boldsymbol{y}}_{\text{decision in } \mathcal{Y}(x)}, \overbrace{\xi}^{\text{noise (not observed)}})\Big]$$

Access to evaluation oracle for $c(x,y,\xi)$.
No optimization oracle for $\min_{y\in\mathcal{Y}(x)} c(x,y,\xi)$ or $\min_{y\in\mathcal{Y}(x)} \mathbb{E}_\xi\big[c(x,y,\xi)\big]$

Single step reinforcement learning

Reinforcement learning setting

noise (not observed)

$$\min_{\pi\in\mathcal{H}} \mathcal{R}(\pi) \quad \text{where} \quad \mathcal{R}(\pi) = \mathbb{E}_{(\boldsymbol{x},\xi),\,\boldsymbol{y}\sim\pi(\cdot|\boldsymbol{x})}\Big[c(\boldsymbol{x},\boldsymbol{y},\xi)\Big]$$

context in $\mathcal{X}$

decision in $\mathcal{Y}(x)$

Access to evaluation oracle for $c(x,y,\xi)$.

No optimization oracle for $\min_{y\in\mathcal{Y}(x)} c(x,y,\xi)$ or $\min_{y\in\mathcal{Y}(x)} \mathbb{E}_{\xi}\big[c(x,y,\xi)\big]$

Alternating minimization: decomposition step is not tractable anymore

$$\begin{aligned}\mu_i^{(t+1)} &= Y(x_i) \operatorname*{arg\,min}_{q_i\in\Delta(x_i)} \mathbb{E}_{\boldsymbol{y}\sim q_i}\Big[c(x_i,\boldsymbol{y},\xi_i)\Big] + \kappa \mathcal{L}_{\Omega_{\Delta^{\mathcal{Y}(x_i)}}}\Big(Y(x_i)^\top \varphi_{\bar{w}^{(t)}}(x_i); q_i\Big)\\ &= \mathbb{E}_{\boldsymbol{Z}}\Big[\operatorname*{arg\,min}_{y_i\in\mathcal{Y}(x_i)} c(x_i,y_i,\xi_i) - \kappa\big(\varphi_{\bar{w}^{(t)}}(x_i)+\varepsilon\boldsymbol{Z}\big)^\top y_i\Big]\end{aligned}$$

No oracle available

Structured reinforcement learning[9]

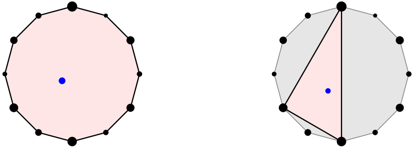
Replace $\mathcal{Y}(x_i)$ by $\hat{\mathcal{Y}}_k^{(t)}(x_i)$: k atoms sampled from $p(\boldsymbol{y}|x_i, w^{(t)})$

$$\mu_i^{(t+1)} = Y(x_i) \operatorname*{arg\,min}_{q_i \in \Delta(x_i)} \mathbb{E}_{\boldsymbol{y} \sim q_i}\Big[c(x_i, \boldsymbol{y}, \xi_i)\Big] + \kappa \mathcal{L}_{\Omega_{\Delta^{\hat{\mathcal{Y}}_k^{(t)}(x_i)}}}\Big(Y(x_i)^\top \varphi_{\bar{w}^{(t)}}(x_i); q_i\Big)$$

$$=_{\text{entr.}} \operatorname{soft\,max}_{y_i \in \hat{\mathcal{Y}}_k^{(t)}(x_i)} \Big[\kappa \varphi_{\bar{w}^{(t)}}(x_i)^\top y_i - c(x_i, y_i, \xi_i)\Big] \qquad \text{(Entropic regularization)}$$

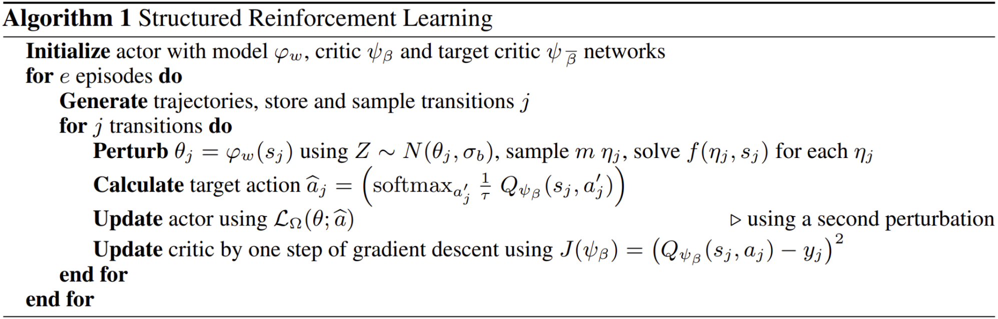
$$=_{\text{pert.}} \mathbb{E}_{\boldsymbol{Z}}\Big[\operatorname*{arg\,min}_{y_i \in \hat{\mathcal{Y}}_k^{(t)}(x_i)} c(x_i, y_i, \xi_i) - \kappa\big(\varphi_{\bar{w}^{(t)}}(x_i) + \varepsilon \boldsymbol{Z}\big)^\top y_i\Big] \qquad \text{(Perturbation)}$$

Tractable by evaluation of the k elements of $\hat{\mathcal{Y}}_k^{(t)}(x_i)$



[9] Heiko Hoppe et al. (2025). *Structured Reinforcement Learning for Combinatorial Decision-Making*. arXiv: 2505.19053 [cs.LG]. url: https://arxiv.org/abs/2505.19053.

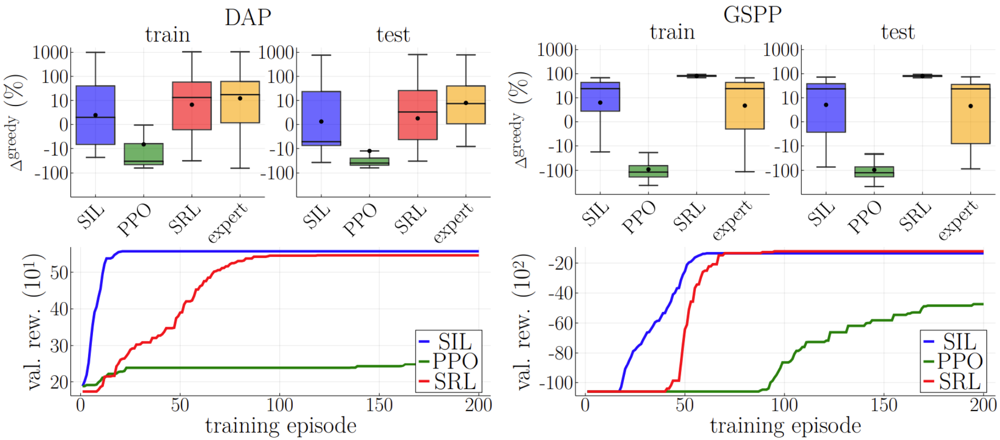
Embedding in an actor critic to go multistage



Algorithm 1 Structured Reinforcement Learning

```
Initialize actor with model $\varphi_w$, critic $\psi_\beta$ and target critic $\psi_{\overline{\beta}}$ networks
for $e$ episodes do
    Generate trajectories, store and sample transitions $j$
    for $j$ transitions do
        Perturb $\theta_j = \varphi_w(s_j)$ using $Z \sim N(\theta_j, \sigma_b)$, sample $m$ $\eta_j$, solve $f(\eta_j, s_j)$ for each $\eta_j$
        Calculate target action $\widehat{a}_j = \left(\text{softmax}_{a'_j} \frac{1}{\tau} Q_{\psi_\beta}(s_j, a'_j)\right)$
        Update actor using $\mathcal{L}_\Omega(\theta; \widehat{a})$        ▷ using a second perturbation
        Update critic by one step of gradient descent using $J(\psi_\beta) = \left(Q_{\psi_\beta}(s_j, a_j) - y_j\right)^2$
    end for
end for
```

Numerical results



Conclusion

Neural network with combinatorial optimization layers improve state of the art

- Contextual Stochastic Optimization (tactic, strategic)
- Dynamic problems (operations)

in combinatorial settings.

Alternating minimization for empirical risk minimization

- Deep learning compatible
- Leads to practically better policies
- Convergence to minimum of empirical risk minimization problem
- Generalization guarantees (approximation ratio in probability)
- Can be turned into an RL algorithm

`https://github.com/JuliaDecisionFocusedLearning`

Combinatorial, convex, stochastic optimization, statistical learning.

References I

Aubin-Frankowski, Pierre-Cyril et al. (July 2024). *Generalization Bounds of Surrogate Policies for Combinatorial Optimization Problems*. doi: `10.48550/arXiv.2407.17200`. arXiv: `2407.17200 [stat]`. (Visited on 12/10/2024).

Baty, Léo et al. (Feb. 2024). "Combinatorial Optimization-Enriched Machine Learning to Solve the Dynamic Vehicle Routing Problem with Time Windows". In: *Transportation Science*. issn: 0041-1655. doi: `10.1287/trsc.2023.0107`. (Visited on 07/18/2024).

Berthet, Quentin et al. (2020). "Learning with differentiable pertubed optimizers". In: *Advances in neural information processing systems* 33, pp. 9508–9519.

Blondel, Mathieu, André F. T. Martins, and Vlad Niculae (2020). "Learning with Fenchel-Young Losses". In: *Journal of Machine Learning Research* 21.35, pp. 1–69.

Bouvier, Louis et al. (2025). "Primal-dual algorithm for contextual stochastic combinatorial optimization". In: *arXiv preprint arXiv:2505.04757*.

References II

- Dalle, Guillaume et al. (July 2022). *Learning with Combinatorial Optimization Layers: A Probabilistic Approach*. eprint: 2207.13513.
- Donti, Priya, Brandon Amos, and J. Zico Kolter (2017). “Task-Based End-to-end Model Learning in Stochastic Optimization”. In: *Advances in Neural Information Processing Systems*. Vol. 30. Curran Associates, Inc. (Visited on 07/24/2024).
- Greif, Toni et al. (Feb. 2024). *Combinatorial Optimization and Machine Learning for Dynamic Inventory Routing*. arXiv: 2402.04463 [math]. (Visited on 03/04/2024).
- Hoppe, Heiko et al. (2025). *Structured Reinforcement Learning for Combinatorial Decision-Making*. arXiv: 2505.19053 [cs.LG]. url: https://arxiv.org/abs/2505.19053.
- Mensch, Arthur and Mathieu Blondel (2018). “Differentiable dynamic programming for structured prediction and attention”. In: *International Conference on Machine Learning*. PMLR, pp. 3462–3471.

References III

- Mitra, Debasis, Fabio Romeo, and Alberto Sangiovanni-Vincentelli (1986). "Convergence and Finite-Time Behavior of Simulated Annealing". In: *Advances in Applied Probability* 18.3, pp. 747–771. issn: 0001-8678. doi: 10.2307/1427186. url: https://www.jstor.org/stable/1427186.
- P., A. (Apr. 2021). "Learning to Approximate Industrial Problems by Operations Research Classic Problems". In: *Operations Research*.
- Pogančić, Marin Vlastelica et al. (Sept. 2019). "Differentiation of Blackbox Combinatorial Solvers". In: *International Conference on Learning Representations*. (Visited on 07/22/2024).
- Sadana, Utsav et al. (Mar. 2024). "A Survey of Contextual Optimization Methods for Decision-Making under Uncertainty". In: *European Journal of Operational Research*. issn: 0377-2217. doi: 10.1016/j.ejor.2024.03.020. (Visited on 07/12/2024).

References IV

Wainwright, Martin J, Michael I Jordan, et al. (2008). "Graphical models, exponential families, and variational inference". In: *Foundations and Trends® in Machine Learning* 1.1–2, pp. 1–305.