Primal-dual algorithm for multistage stochastic optimization

Solène Delannoy-Pavy (RTE, Ecole des Ponts ParisTech)
Axel Parmentier (Ecole des Ponts ParisTech)

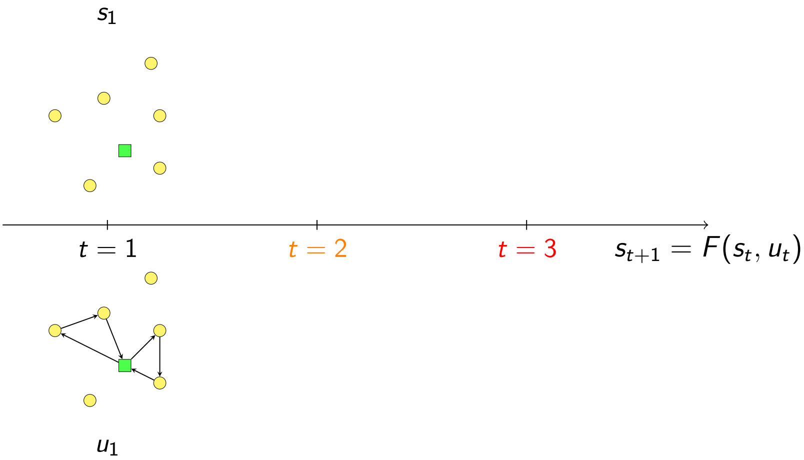
01/08/25

XVII[th] CONFERENCE ON STOCHASTIC PROGRAMMING
28 JULY – 01 AUGUST 2025, FRANCE

Dynamic Vehicle Routing Problem with Time Windows[1]

State
$s_t \in \mathcal{S}$
set of customers

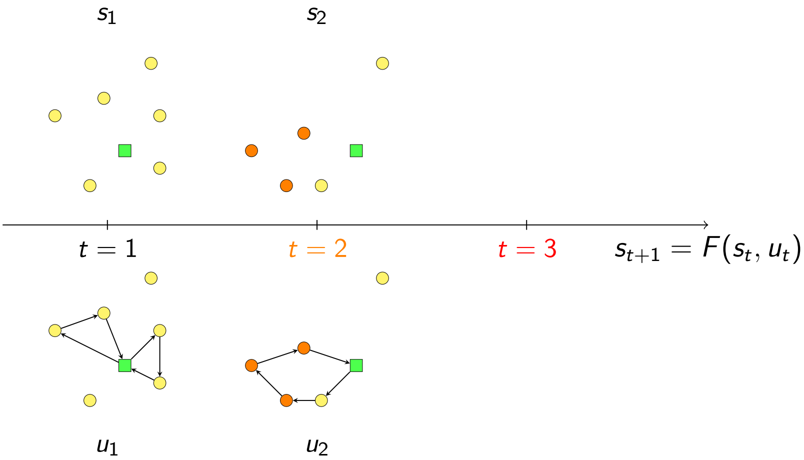
Decision
$u_t \in \mathcal{U}(u_t)$
set of routes



[1] Baty et al. 2024.

Dynamic Vehicle Routing Problem with Time Windows[1]

State
$s_t \in \mathcal{S}$
set of customers



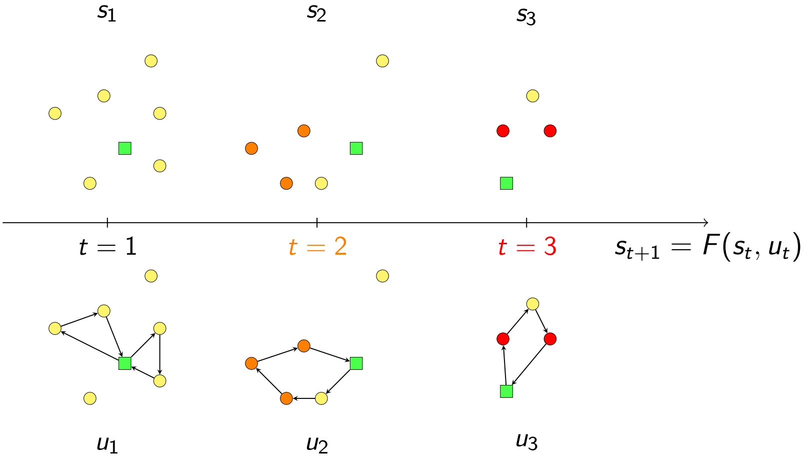
Decision
$u_t \in \mathcal{U}(u_t)$
set of routes

[1] Baty et al. 2024.

Dynamic Vehicle Routing Problem with Time Windows[1]

State
$s_t \in \mathcal{S}$
set of customers

Decision
$u_t \in \mathcal{U}(u_t)$
set of routes



[1] Baty et al. 2024.

Dynamic VRPTW

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

$$\pi : \quad \mathcal{X} \quad \to \quad \mathcal{Y}$$
$$\underbrace{s_t}_{\text{set of customers}} \quad \mapsto \quad \underbrace{u_t}_{\text{set of routes}}$$

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

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

Combinatorial Markov Decision Processes

Setting:

- High-dimensional set of states $\mathcal{S}$
- Finite but combinatorial set of decisions $\mathcal{U}(s) \subset \mathbb{R}^{d(s)}$
- Exogeneous independent random variables $\boldsymbol{\xi}$
- Dynamics $s' = F(s, u, \xi)$ and initial probability distribution on $\mathcal{S}$
- Cost function $c(s, u)$

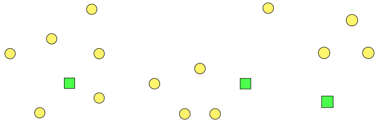
Goal: find a policy π^* (possibly random) minimizing the total cost

$$\pi^* \in \arg\min_{\pi} \mathbb{E}_{\boldsymbol{\xi}, \boldsymbol{u_t} \sim \pi(\cdot|\boldsymbol{s}_t)} \left[\sum_t c(\boldsymbol{s}_t, \boldsymbol{u}_t)) \right]$$

Full information on history

For a given T we have N samples

$$\xi_i = (\xi_{i,1}, \ldots, \xi_{i,T})$$



The following problem is hard to solve for combinatorial MDPs

$$\min_{(u_{i,t})_{i,t}} \frac{1}{N} \sum_{i=1}^{N} \sum_{t=0}^{T} c(s_{i,t}, u_{i,t})$$

s. a. $u_{i,t} \in \mathcal{U}(s_{i,t})$

$s_{i,t+1} = F(s_{i,t}, u_{i,t}, \xi_{i,t+1})$ Dynamics

$s_{i,0} = s$

$u_{i,t} = u_{i',t} \quad \forall i, i' \quad \text{such as} \quad \xi_{i,1} = \xi_{i',1}, \ldots, \xi_{i,t} = \xi_{i',t}$ Nonanticipativity constaints

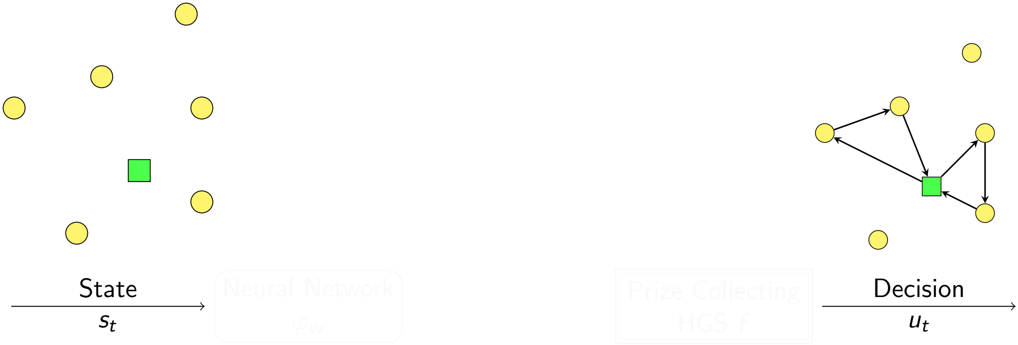
Classical assumptions in stochastic programming

We have an efficient algorithm to solve the determistic single scenario problem

$$
\begin{aligned}
& \min_{u_{[T]}} \sum_{t=0}^{T} c(s_t, u_t) - \theta_t \top u_t \\
\text{s. a.} \quad & u_t \in \mathcal{U}(s_t) \\
& s_{t+1} = F(s_t, u_t, \xi_{t+1}) \\
& s_0 = s
\end{aligned}
$$

where θ_t are dual vectors.

Policy that won the EURO-NeurIPS challenge[2]



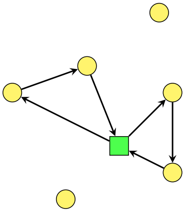
[2] Léo Baty et al. (Feb. 2024). "Combinatorial Optimization-Enriched Machine Learning to Solve the Dynamic Vehicle Routing Problem with Time Windows". In: *Transportation Science*. ISSN: 0041-1655. DOI: 10.1287/trsc.2023.0107. (Visited on 07/18/2024).

Policy that won the EURO-NeurIPS challenge[2]

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

- Serving customers is optional
- Serving customer n gives prize θ_n
- **Objective**: maximize total profit minus routes costs

$$\max_{u \in \mathcal{U}(s_t)} \underbrace{\sum_{(n,m)\in s_t^2} \theta_n u_{n,m}}_{\text{total profit}} - \underbrace{\sum_{(n,m)\in s_t^2} c_{n,m} u_{n,m}}_{\text{total routes cost}} .$$

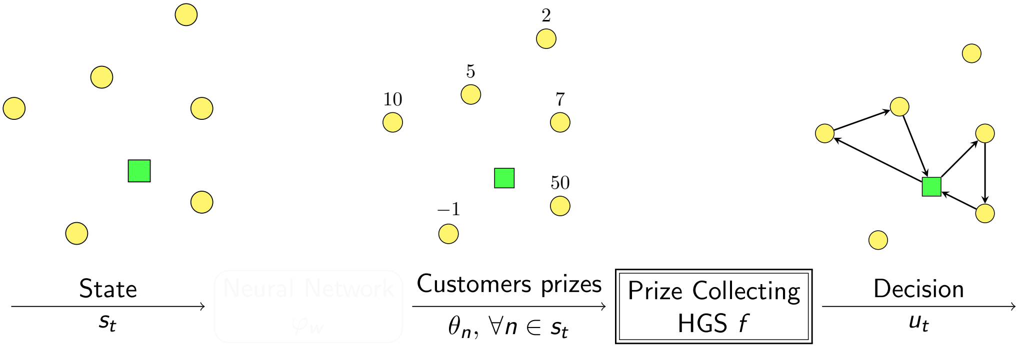


- **Algorithm**: Prize Collecting Hybrid Genetic Search

[2] 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[2]

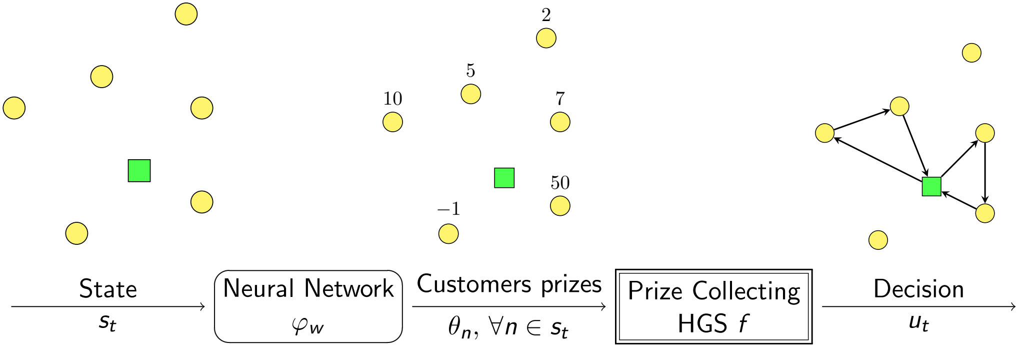
Difficulty: no natural way of computing meaningful prizes



[2] 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[2]

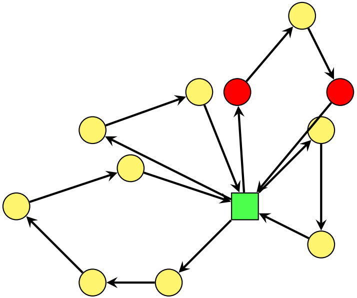
Solution: use a neural network to predict request prizes $\theta = \varphi_w(s_t)$



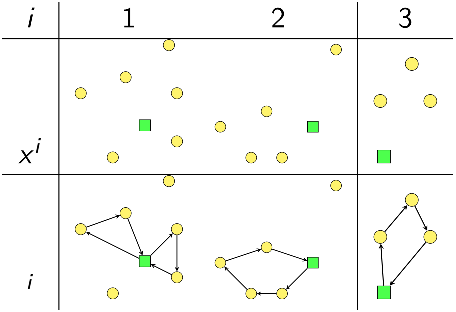
➥ Policy π_w

[2] 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).

State of the art: imitate anticipative decisions Baty et al. 2024

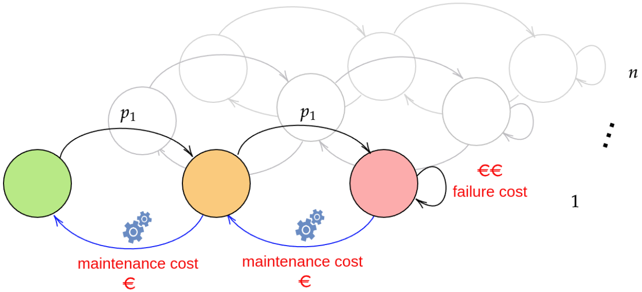


We rebuild the anticipative decisions a posteriori



- ➥ use COaML (Combinatorial Optimization augmented ML)
- ➥ train by imitating anticipative trajectories

Multi-components Ressource constrained Maintenance Problem (MRMP)



- n components
- maintain at most r at each stage

State
$s_t = s_1, \ldots, s_n \in \mathcal{S}_1 \times \cdots \times \mathcal{S}_n$
Decision $u_t = u_1, \ldots, u_n \in [0,1]^n$

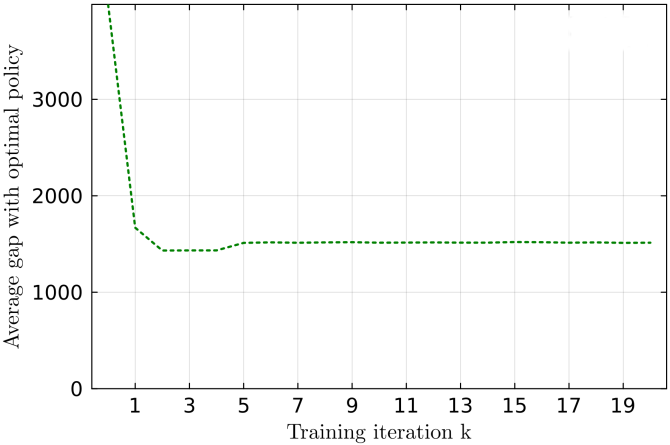
$\sum_{i=1}^n u_i \leq r$

CO layer: maintaining component n gives prize θ_n

$$\xrightarrow[s_t]{\text{State}} \boxed{\text{Neural Network } \varphi_w} \xrightarrow[\theta_i]{\text{Maintenance prizes}} \boxed{\max_{\sum_{i=1}^N u_i \leq 1} \sum_{i=1}^N \theta_i \top u_i} \xrightarrow[u_i]{\text{Decision}}$$

Anticipative solutions can be bad - we need coordination!

Imitate expert anticipative trajectories



Bad performance on the MRMP

The states in our training set $\mathcal{D}$ are poor

We should solve

$$\min_{w} \mathbb{E}_{s \sim \delta_w} \left[\mathcal{L}\big(\varphi_w(s), \delta^*(s)\big) \right]$$

while we solve

$$\min_{w} \mathbb{E}_{s \sim \delta^*} \left[\mathcal{L}\big(\varphi_w(s), \delta^*(s)\big) \right]$$

Building $\mathcal{D}$ is a classical problem in Reinforcement Learning. One solution is to update the dataset for expert demonstration, for example using DAgger[3] ($\alpha \in [0, 1]$)

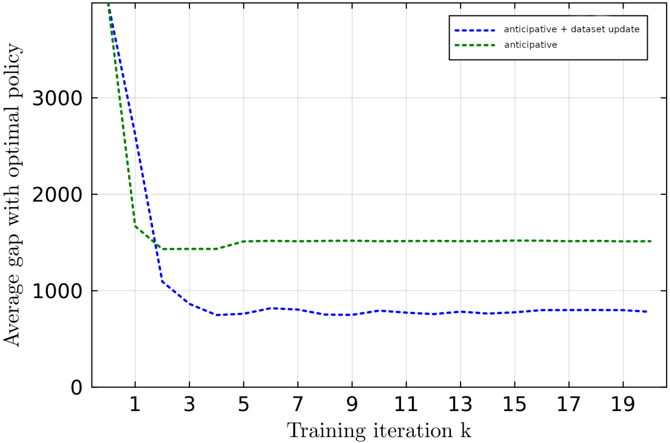
$$\alpha \delta^* + (1 - \alpha) \delta_w$$

[3] Ross, Gordon, and Bagnell 2010.

Anticipative solutions can be bad - we need coordination!

Imitate anticipative decisions

$+$ the learner updates the dataset for expert demonstration

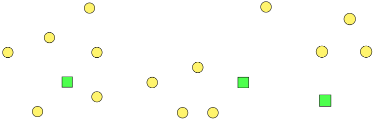


The gap with the optimal solution is still huge.

Coordinating decisions at the current time step

For a given T we have N samples

$$\xi_i = (\xi_{i,1}, \dots, \xi_{i,T})$$



$$\min_{(u_{i,t})_{i,t}} \frac{1}{N} \sum_{i=1}^{N} \sum_{t=0}^{T} c(s_{i,t}, u_{i,t})$$

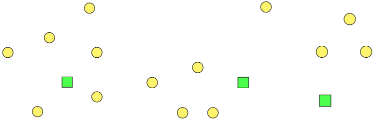
s. a. $u_{i,t} \in \mathcal{U}(s_{i,t})$

$s_{i,t+1} = F(s_{i,t}, u_{i,t}, \xi_{i,t+1})$ Dynamics

$s_{i,0} = s$

$u_{i,t} = u_{i',t} \quad \forall i, i'$ such as $\xi_{i,1} = \xi_{i',1}, \dots, \xi_{i,t} = \xi_{i',t}$ Nonanticipativity constaints

Coordinating decisions at the current time step



For a given T we have N samples

$$\xi_i = (\xi_{i,1}, \dots, \xi_{i,T})$$

$$\begin{aligned}
\min_{(u_{i,t})_{i,t}} \quad & \frac{1}{N} \sum_{i=1}^{N} \sum_{t=0}^{T} c(s_{i,t}, u_{i,t}) \\
\text{s. a.} \quad & u_{i,t} \in \mathcal{U}(s_{i,t}) \\
& s_{i,t+1} = F(s_{i,t}, u_{i,t}, \xi_{i,t+1}) \quad \text{Dynamics} \\
& s_{i,0} = s \\
& u_{i,1} = u_{i',1} \quad \forall i, i' \quad \text{First stage nonanticipativity constaints}
\end{aligned}$$

We try to learn the solutions of the two-stage approximation of the sampled problem

Corresponding empirical cost minimization problem

Cost in the two-stage approximation:

$$c^{2\mathrm{S}}(s,u,\xi) = c(s,u) + Q(s,u,\xi)$$

$$\text{Recourse cost:}\quad Q(s,u,\xi) = \min_{u_{[1:T]}} \sum_{t=1}^{T} c(s_t,u_t)$$

$$\text{s.t.}\quad s_1 = F(s,u,\xi_1)$$

$$s_{t+1} = F(s_t,u_t,\xi_{t+1}) \quad \forall t \in [1:T-1]$$

$$u_t \in \mathcal{U}(s_t) \quad \forall t \in [1:T]$$

The first stage solutions of the previous problem are solutions to

$$\min_{u\in\mathcal{U}(s)} \frac{1}{N} \sum_{i=1}^{N} c^{2\mathrm{S}}(s,u,\xi_i)$$

Learning coordinated policies

We want to learn policies minimizing the empirical cost

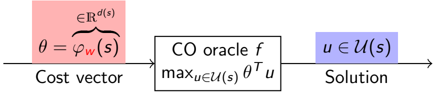
$$\min_{w} \mathbb{E}_{\boldsymbol{s}\sim d_w}\left[\frac{1}{N}\sum_{i=1}^{N}\mathbb{E}_{\boldsymbol{u}\sim\pi_w(\cdot|\boldsymbol{s})}\left[c^{2\mathrm{S}}(\boldsymbol{s},\boldsymbol{u},\xi_i)\right]\right]$$

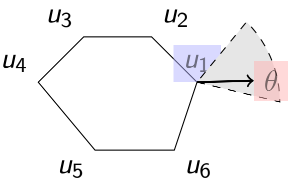
Assuming that we have sampled a dataset $\mathcal{D} = (s_i, \xi_i)_{i\in[N]}$

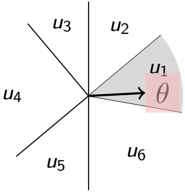
$$\min_{w}\left[\frac{1}{N}\sum_{i=1}^{N}\mathbb{E}_{\boldsymbol{u}\sim\pi_w(\cdot|s_i)}\left[c^{2\mathrm{S}}(s_i,\boldsymbol{u},\xi_i)\right]\right]$$

Challenges with CO-augmented Machine Learning (COaML)

Policies π_w based on







Supervised learning: Fenchel-Young Losses (FYL)[4]

$$\mathcal{L}_\Omega(\theta; \bar{u}) = \overbrace{\max_{u \in \mathcal{C}(s)} \big(\langle \theta | u \rangle - \Omega(u)\big) - \big(\langle \theta | \bar{u} \rangle - \Omega(\bar{u})\big)}^{\substack{\text{Non-optimality of } \bar{u} \\ \text{as a solution of the} \\ \text{regularized prediction problem}}} = \Omega^*(\theta) + \Omega(\bar{u}) - \langle \theta | \bar{u} \rangle$$

[4] Blondel, Martins, and Niculae 2020.

Learning coordinated policies

We want to learn policies minimizing the empirical cost

$$\min_w \; \mathbb{E}_{\boldsymbol{s}\sim d_w}\left[\frac{1}{N}\sum_{i=1}^{N}\mathbb{E}_{\boldsymbol{u}\sim\pi_w(\cdot|\boldsymbol{s})}\left[c^{\mathrm{2S}}(\boldsymbol{s},\boldsymbol{u},\xi_i)\right]\right]$$

Assuming that we have sampled a dataset $\mathcal{D} = (s_i, \xi_i)_{i\in[N]}$

$$\min_w \left[\frac{1}{N}\sum_{i=1}^{N}\mathbb{E}_{\boldsymbol{u}\sim\pi_w(\cdot|s_i)}\left[c^{\mathrm{2S}}(s_i,\boldsymbol{u},\xi_i)\right]\right]$$

Proposition

We can learn w such that π_w minimizes the empirical risk for two stage problems using an Alternating Minimization (AM) algorithm, see Bouvier et al.[5]

[5] Bouvier et al. 2025.

Coordinating decisions during learning[6]

Surrogate problem with dataset $\mathcal{D} = (s_i, \xi_i)_{i\in[N]}$

$$\min_{w, q_\otimes} \mathcal{S}_N(s_w; q_\otimes) := \min_{w, q_\otimes} \frac{1}{N}\sum_{i=1}^{N} \mathbb{E}_{\boldsymbol{u}\sim q_i}\left[c^{2\mathrm{S}}(s_i, \boldsymbol{u}, \xi_i)\right] + \kappa \mathcal{L}_{\Omega_{\Delta(s_i)}}\left(U(s_i)^\top \varphi_w(s_i); q_i\right)$$

Alternating minimization update:

$$q_i^{(k+1)} = \min_{q_i} \mathbb{E}_{\boldsymbol{u}\sim q_i}\left[c^{2\mathrm{S}}(s_i, \boldsymbol{u}, \xi_i)\right] + \kappa \mathcal{L}_{\Omega_{\Delta(s_i)}}\left(U(s_i)^\top \varphi_{\bar{w}^{(k)}}(s_i); q_i\right) \quad \text{(decomposition)}$$

$$\bar{w}^{(k+1)} \in \arg\min_{w\in\mathcal{W}} \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}_{\Omega_{\mathcal{C}(s_i^{(k)})}}\left(\varphi_w(s_i^{(k)}); U(s_i^{(k)}) q_i^{(k+1)}\right) \quad \text{(coordination)}$$

$$\mathcal{D}^{(k)} \to \mathcal{D}^{(k+1)} \quad \text{(dataset update)}$$

[6] Bouvier et al. 2025.

Tractable updates for well chosen $\Omega_{\Delta(s_i)}$

Decomposition:

$$q_i^{(k+1)} = \mathbb{E}_{\mathbf{Z}}\Big[\Big(\arg\min_{u_{i,0:T}} \sum_{t=0}^{T} c(s_{i,t}, u_{i,t}) - \kappa\left(\varphi_{\bar{w}^{(k)}}(s_i) + \epsilon\mathbf{Z}\right)^\top u_{i,0}\Big)_0\Big]$$

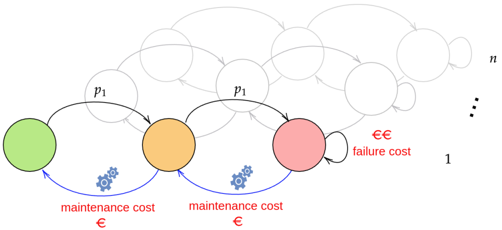
$$\begin{aligned} \text{s.t.} \quad & s_{i,0} = s_i^{(k)}, \\ & u_{i,t} \in \mathcal{U}(s_{i,t}) \quad \forall t \in [0:T], \\ & s_{i,t+1} = F(s_{i,t}, u_{i,t}, \xi_{i,t+1}^{(k)}) \quad \forall t \in [0:T-1]. \end{aligned}$$

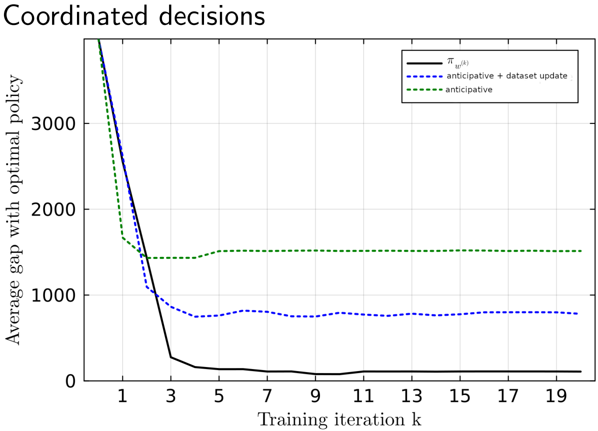
Coordination:

$$\bar{w}^{(k+1)} \in \arg\min_{w \in \mathcal{W}} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}_{\Omega_{\mathcal{C}(s_i^{(k)})}}\left(\varphi_w(s_i^{(k)}); U(s_i^{(k)}) q_i^{(k+1)}\right)$$

Dataset update: $\mathcal{D}^{(k)} \to \mathcal{D}^{(k+1)}$

Current stage coordination - MRMP





The learned policy outperforms the policy imitating anticipative decisions

Problem

- Imitating anticipative decisions can fail on problems where strong coordination is needed, typically on maintenance and pricing problems.

Takeaways

- We coordinate decisions during learning.
- Encouraging results on a simple problem, benchmark on large size problems coming soon.

Questions

- What are the best rules for updating the dataset ?
- Could we coordinate T decisions at the same learning step?