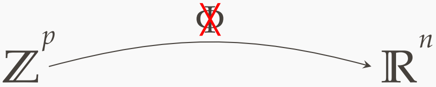
ADM & ML

FOR DISCRETE CONTROL WITH STORAGE

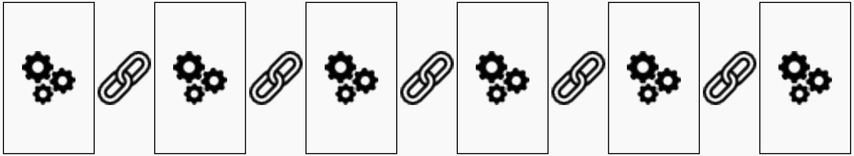
Sophie Demassey, Valentina Sessa, Amirhossein Tavakoli (Mines Paris – PSL)

ISCO 2024

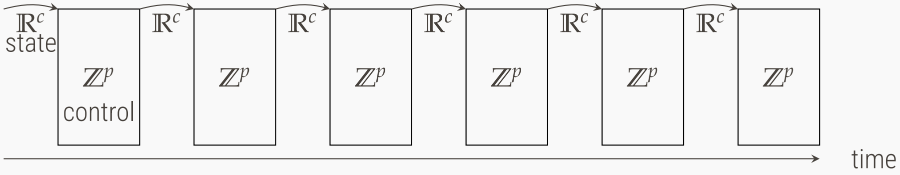
I for MIP



- metaheuristics, local search, B&B, Benders **search in the projected $\mathbb{Z}$-space** to take advantage of: a calculable optimal mapping Φ, finite neighborhoods, finite search space
- **large nonconvex MIP**:
  what if the feasible integer solutions are sparse and scarce and if computing Φ is not that easy ?
- **dualizing** the complicating and/or coupling constraints: no clear trade-off



M for MIP (ex: discrete control)



- **dualizing** the time-coupling/state constraints: static control without initial condition
- **fixing** the time-coupling/state variables: static control with a **known** initial condition

$$\mathbb{Z}^{pT} \xleftarrow{\;\Psi\;} \mathbb{R}^{cT} \times \mathbb{R}^{nT}$$

Black-box Ψ is nonconvex (vs. lagrangian dual) but *somewhat* smooth + denser state space: optimize Ψ **locally** on $\mathbb{R}^{cT}$ (0-order information)

Hybrid decomposition and convergence

$$\mathbb{Z}^{pT} \underset{\Psi}{\overset{\Phi_\mu}{\rightleftarrows}} \mathbb{R}^{cT} \times \mathbb{R}^{nT}$$

0. dualize the coupling constraints with multipliers μ
   1. start with an approximate state profile $S \in \mathbb{R}^{cT}$
   2. **alternate** solving Ψ (separated control) and Φ_μ (resulting states)
   3. stop at fixed point $\Phi_\mu(\Psi(S)) \approx S$.

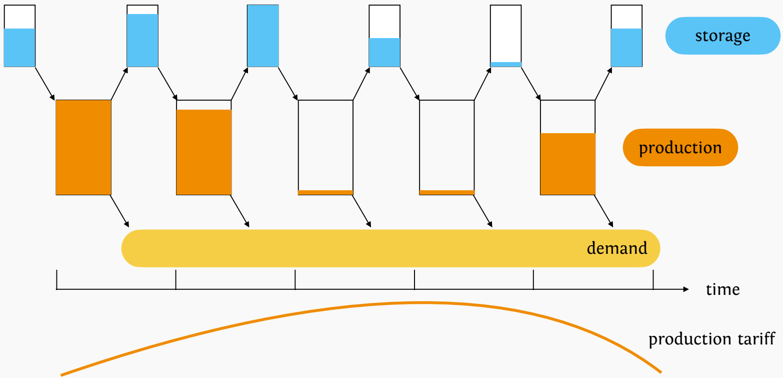
policies for updating μ:

- under conditions: get a global optimum (ADMM), a stationary point (biconvex), or nothing
- fixed high μ: local search around one (or more) approximate candidate(s) S and repair feasibility

hybrid ML/MP decomposition method: learn the starting points S

bilevel method: implied continuous variables (outer level) / discrete decisions (inner level)

SCHEDULING WITH STORAGE



- find operation and storage levels to meet demand, storage conservation and capacity on each period and minimize the total operation cost
- feasible solutions are rare when storage capacities are tight and operations are stepped

SCHEDULING WITH STORAGE

$$
\begin{aligned}
(P):\ &\min_{x,y,s} \sum_{t\in\mathcal{T}} f_t(x_t, y_t, s_t, C_t) && && \text{operation cost} \\
s.t.:\ &g_t(x_t, y_t, s_t, L_t) = 0 && \forall\, t \in \mathcal{T} && \text{static operation} \\
&s_{t+1} = s_t + y_t^I && \forall\, t \in \mathcal{T} && \text{storage conservation} \\
&s_t \in \mathcal{S}_t = [\underline{S}_t, \overline{S}_t] \subseteq \mathbb{R}^I && \forall\, t \in \overline{\mathcal{T}} && \text{storage capacity} \\
&x_t \in \mathcal{X}_t \subseteq \{0,1\}^N, y_t \in \mathcal{Y}_t \subseteq \mathbb{R}^M && \forall\, t \in \mathcal{T}. &&
\end{aligned}
$$

find operation (x_t, y_t) and storage level s_t to meet demand, storage conservation and capacity on each period t and minimize the total operation cost.

ASSUMPTION ON THE STEADY STATE

steady state operation (x_t, y_t) for given storage level s_t and demand L_t

$$(x_t, y_t) \ : \ g_t(x_t, y_t, s_t, L_t) = 0$$

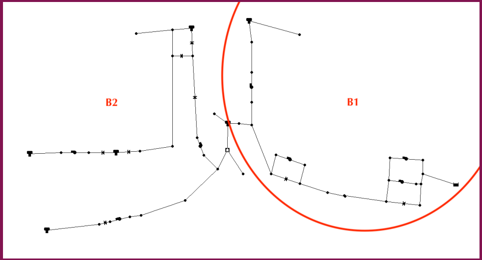
- possibly a nonconvex system
- but we assume that it is easy to solve and optimize on **when s_t is known**

EX: SCHEDULING OF POTENTIAL-FLOW NETWORKS

sequence of potential-flow equilibria on a dynamic graph

$$g_t \equiv \left(x_t^\top (y_t^H - \phi(y_t^Q)),\ y_{tJ}^Q - L_t,\ y_{tR}^H - s_t\right)$$

x_t: arc activity on/off,
(y_t^Q, y_t^H): active arc flows, nodal potentials
s_t: potential at storage nodes,
L_t: demand at service nodes



- nonlinear potential-flow relation ϕ_a on each arc
- for x_t and s_t fixed: (y_t^Q, y_t^H) unique KKT solution of a linearly-constrained strictly convex problem
- for s_t fixed: $\min_{x_t \in \{0,1\}^A} f_t(x_t)$ is enumerable with graph partition along the storage nodes

OPTION 0: DUALIZE THE TIME-COUPLING CONSTRAINTS

ex: lagrangian subproblem

$$(P): \min_{x,y,s} \sum_t f_t(x_t, y_t, s_t, C_t) + \mu_t(s_{t+1} - s_t - y_t^I)$$

$$s.t.: g_t(x_t, y_t, s_t, L_t) = 0 \qquad \forall t \in \mathcal{T}$$

- the model becomes separable in time
- but each static component remains hard (and poor) as the initial state s_t **is unknown**

Option 1: full variable-split and ADMM

ADMM: variant of the augmented lagrangian $p_t(z,\mu) = \mu_d^\top z + \mu_p||z||_2$ with partial update

1: fix storage s, then compute (x,y)

$$P(s) : \min_{(x,y)} \sum_t f_t(x_t, y_t, s_t, C_t) + p_t(s_{t+1} - s_t - y_t, \mu_t) + p_t(g_t(x_t, y_t, s_t, L_t), \rho_t).$$

$\downarrow$ $\uparrow$ update μ, ρ

2: fix command (x,y), then compute s

$$P(x,y) : \min_{s} \sum_t f_t(x_t, y_t, s_t, C_t) + p_t(s_{t+1} - s_t - y_t, \mu_t) + p_t(g_t(x_t, y_t, s_t, L_t), \rho_t)$$

- no theoretical convergence with nonconvex coupling constraints
- $P(s)$ is too poor, $P(x,y)$ too hard (inverse problem)

OPTION 2: PARTIAL SPLIT AND ADM-LIKE

no theory ? be practical: keep $g_t(x_t, y_t, s_t, L_t) = 0$ in $P(s)$, but **drop it** from $P(x, y)$

1: fix storage s, then compute (x, y)

$$P(s) : \min_{(x,y)} \sum_t f_t(x_t, y_t, s_t, C_t) + p_t(s_{t+1} - s_t - y_t, \mu_t)$$

$$s.t. : g_t(x_t, y_t, s_t, L_t) = 0 \qquad \forall t \in \mathcal{T}$$

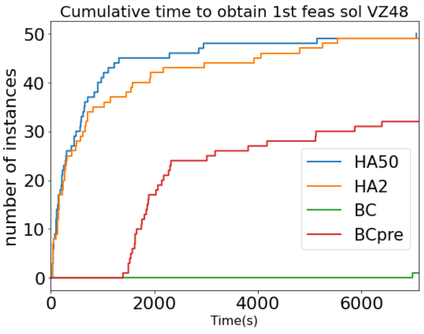
$\downarrow$ $\uparrow$ stop when $\|s_{t+1} - s_t - y_t\| < \epsilon$

2: fix command (x, y), then compute s

$$P(x, y) : \min_{s} \sum_t f_t(x_t, y_t, s_t, C_t) + p_t(s_{t+1} - s_t - y_t, \mu_t)$$

Experiments: pump scheduling in water networks

- **HA**: partial split $\mu \in \{50, 2\}$ from multiple learned storage profiles
- **BC**: SOA Branch-and-Check [Opt&Eng 2021] + **BCpre** advanced preprocessing [ICAE 2022]
- run algorithms on 50 instances within 2 hours; stop at the first feasible solution
- hard to just compute a feasible solution when storage limits are tight



Learning the starting storage profiles

We implemented:

- a standard deep learning architecture to capture temporal dependencies and local trends
- data set: historical data and corresponding optimal solutions
- Monte-Carlo dropout to **get multiple profiles** $\longrightarrow$ multi-start ADM $\longrightarrow$ diversification
- **a scaling mechanism**: train on coarse-grained data ($|\mathcal{T}| = 12$) then apply to fine time-discretization ($|\mathcal{T}| = 48$) by resizing input (load and tariff) and output (profile) linearly.
- learning continuous states vs discrete command: regression vs classification, smoother moves, up-scaling... more chance to end up with a feasible solution

Reconsider full-split on the bilevel model

- the static potential/flow equilibrium for given x and s (at time t) is unique:

$$y = (y^Q, y^H) : g(x, y, s, L) = 0$$

- as the **KKT solution** of a min-strictly-convex-cost flow problem on $G(N, A(x))$:

$$y^Q \in \arg\min_q \{\Phi(q) + s^\top q_R : q_J = L\} \qquad \text{(primal flow)}$$

$$\equiv \quad y^H \in \arg\max_h \{\Phi^*(h) + L^\top h : h_R = s\} \qquad \text{(dual potential)}$$

$$\equiv \quad \Phi(y^Q) + s^\top y^Q_R = \Phi^*(y^H) + L^\top y^H,\ y^Q_J = L,\ y^H_R = s \qquad \text{(strong duality condition).}$$

- with a suited variable change, x does not appear in the SD condition
- but nonconvexity remains in the bilinear term $s^\top y^Q_R$

DUALIZE THE SD CONDITION + ADMM

1: fix storage s, then compute (x, y)

$$P(s): \min_{(x,y)\in\mathcal{X}\times\mathcal{Y}} \sum_{t\in\mathcal{T}} f_t(x_t, y_t, s_t, C_t) + p_t(s_{t+1} - s_t - y^Q_{tR}, \mu_t) + p_t(SD_t(y_t, s_t), \rho_t)$$

$$s.t.: (1 - x_t)y^Q_t = 0,\ y^Q_{tJ} = L_t,\ y^H_{tR} = s_t \qquad \forall t \in \mathcal{T}.$$

with $SD_t(y_t, s_t) = \Phi(y^Q_t) + s_t^\top y^Q_{tR} - \Phi^*(y^H_t) - L_t^\top y^H_t$ (and f_t, p_t linear), then for each t, $P_t(s_t)$ is **separable in primal/dual parts**, i.e. (y^Q, y^H)-split, corresponding to two equilibrium problems with perturbed costs and penalties

primal: perturbed potentials s_t and resistance ϕ

$$P_t(x_t, s_t): \min_{y^Q_t} \mu_t\Phi(y^Q_t) + l(s_t, C_t, \mu_t, \rho_t)^\top y^Q_t$$

$$s.t.: y^Q_{tJ} = L_t,\ (1 - x_t)y^Q_t = 0.$$

dual: perturbed load L_t and resistance ϕ

$$D_t(x_t, s_t): \max_{y^H_t} \mu_t\Phi^*(y^H_t) + \rho_t L^\top y^H_t$$

$$s.t.: y^H_{tJ} = s_t.$$

CONCLUSION

- solving MIPs by optimizing Ψ

$$\mathbb{Z}^p \xleftarrow[\Psi]{} \mathbb{R}^c$$

- hybrid ML/MIP decomposition approach: ML for optimality, MIP for feasibility

APPLICATIONS

temporal decomposition for control/scheduling/planning:

- **load shifting** or scheduling with storage: storage state → static control at each time
- **capacity expansion planning**: periodic investment → operation on each period

spatial decomposition of networks:

- **traffic network design**: inflow in hubs → design and flows in each component

decomposition by level/stage/scenario:

- **stochastic programming**: first-stage decision → second-stage decision for each scenario

References

- ADMM: **Boyd** on proximal algorithms; **Rockafellar**, **Eckstein** on monotone operators; apps in ML and OPF
- potential networks: cf **Rockafellar** on nonlinear flows and monotropic programming
- [Opt&Eng 2021] **G. Bonvin, S. Demassey, A. Lodi** Pump scheduling in drinking water distribution networks with an LP/NLP-based branch and bound. Optimization and Engineering 2021.
- [ICAE 2022] **A. Tavakoli, V. Sessa, S. Demassey** Strengthening mathematical models for pump scheduling in water distribution. In 14th International Conference on Applied Energy 2022.
- papers available at `https://sofdem.github.io/`
- code available at `https://github.com/sofdem/gopslpnlpbb`