Interface Course 2019
Stochastic Optimization for Large-Scale Systems
♢
Spatial Decomposition Methods I

P. Carpentier, J.-Ph. Chancelier, M. De Lara , V. Leclère

November 8, 2019

École des Ponts
ParisTech

ENSTA
IP PARIS

Ultimate goal of the lecture

How to to obtain "good" strategies for a large scale stochastic optimal control problem, for example a problem corresponding to the optimal management over a given time horizon of a system involving a large amount of dynamical production units.

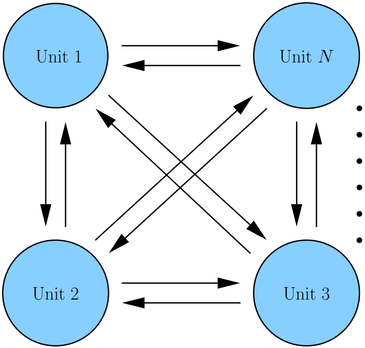
- In order to obtain decision strategies (closed-loop controls), we have to use Dynamic Programming or related methods.
  - **Assumption**: Markovian case,
  - **Difficulty**: curse of dimensionality.
- In order to to take into account the size of the system, we have to use decomposition/coordination techniques.
  - **Assumption**: convexity,
  - **Difficulty**: information pattern of the problem.

Mixture of spatial and temporal decompositions

Lecture outline

1. Examples and background
   - Examples of interconnected systems
   - Convex optimization background
2. Decomposition in the deterministic case
   - Additive model: 3 decomposition methods
   - General model: Auxiliary Problem Principle
3. About decomposition in the stochastic case
   - Dynamic Programming and decomposition
   - Couplings in stochastic optimization

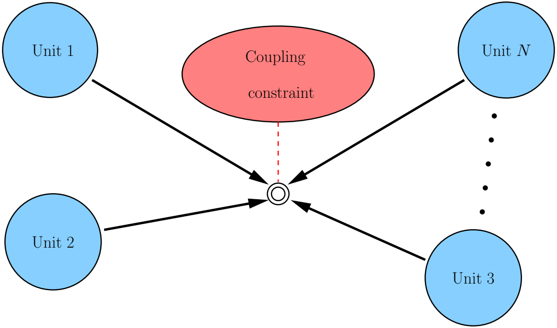
Decomposition and coordination



Interconnected units

- The (large) system to be optimized consists of interconnected subsystems: we want to use this structure in order to formulate optimization subproblems of reasonable complexity.
- But the presence of interactions requires a level of coordination.
- Coordination must provide a local model of the interactions to each subproblem: it is an iterative process.
- The ultimate goal is to obtain the solution of the overall problem by concatenation of the solutions of the subproblems.

Example: the “flower model”

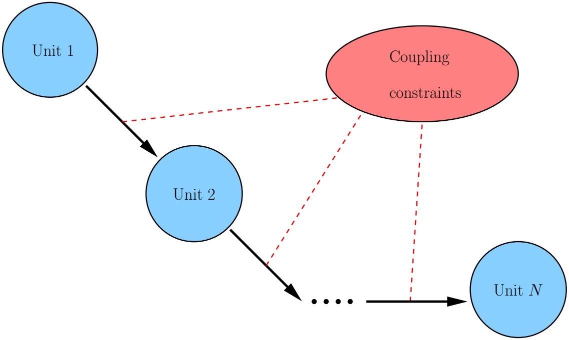


$$\min_{u} \sum_{i=1}^{N} J_i(u_i) \ ,$$

$$\text{s.t.} \quad \sum_{i=1}^{N} \Theta_i(u_i) = \theta \ .$$

Unit Commitment Problem

Example: the "cascade model"

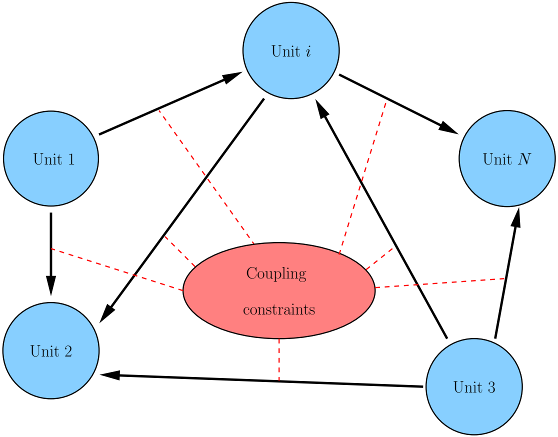


$$\min_{u,v} \sum_{i=1}^{N} J_i(u_i, v_i) ,$$
$$\text{s.t.} \quad H_i(u_i, v_i) = v_{i+1} \quad \forall i .$$

Dams Management Problem

Link with the flower model: $\Theta_i \rightsquigarrow \big(0, \ldots, -v_i, H_i(u_i, v_i), \ldots, 0\big)^\top$.

A general model



$$\min_{u,v} \sum_{i=1}^{N} J_i\Big(u_i, \sum_{j \neq i} v_{j,i}\Big) ,$$

$$\text{s.t.} \quad H_i\Big(u_i, \sum_{j \neq i} v_{j,i}\Big) = v_i .$$

Microgrid Management Problem

Optimization without explicit constraint

$$\min_{u \in \mathcal{U}^{\mathrm{ad}}} J(u) \ . \qquad (\mathcal{P}_{\mathrm{S}})$$

- $\mathcal{U}$: Hilbert space with scalar product $\langle \cdot \, , \cdot \rangle$.
  Examples: $\mathcal{U} = \mathbb{R}^n$ (vectors) or $\mathcal{U} = \mathrm{L}^2\big(\Omega, \mathcal{A}, \mathbb{P}; \mathbb{R}^n\big)$ (random variables).
- $\mathcal{U}^{\mathrm{ad}}$: closed convex subset of $\mathcal{U}$.
- $J : \mathcal{U} \rightarrow \mathbb{R}$: function satisfying some properties
  (convexity, continuity, differentiability, coercivity).

Characterization of a solution $u^\sharp$ (**optimality conditions**):

$$\big\langle \nabla J(u^\sharp) \, , u - u^\sharp \big\rangle \geq 0 \quad \forall u \in \mathcal{U}^{\mathrm{ad}} \ .$$

Computation of the solution $u^\sharp$ (**projected gradient algorithm**):

$$u^{(k+1)} = \mathrm{proj}_{\mathcal{U}^{\mathrm{ad}}}\Big(u^{(k)} - \rho \nabla J\big(u^{(k)}\big)\Big) \ .$$

Optimization with explicit constraints I

$$\min_{u \in \mathcal{U}^{\mathrm{ad}}} J(u) \quad \text{subject to} \quad \Theta(u) \in -C \; . \qquad (\mathcal{P}_{\mathrm{C}})$$

- $\mathcal{U}$: Hilbert space.
- $\mathcal{U}^{\mathrm{ad}}$: closed convex subset of $\mathcal{U}$.
- $\mathcal{V}$: another Hilbert space.
- C: cone of $\mathcal{V}$ (examples: $C = \{0\}$, $C = \{v \geq 0\}$).
- $J : \mathcal{U} \to \mathbb{R}$: cost function.
- $\Theta : \mathcal{U} \to \mathcal{V}$: constraint function satisfying some properties (convexity w.r.t. C, continuity, differentiability).

Constraint Qualification Condition, e.g. $0 \in \mathrm{int}\big(\Theta(\mathcal{U}^{\mathrm{ad}}) + C\big)$.

The dual cone of C is defined by: $C^\star = \big\{\lambda \in \mathcal{V},\ \langle \lambda , v \rangle \geq 0 \ \ \forall v \in C\big\}$.

Optimization with explicit constraints II

Karush-Kuhn-Tucker Conditions

In addition to standard conditions on J and Θ, we assume that the constraints are qualified.

Then a necessary and sufficient condition for $u^\sharp \in \mathcal{U}^{\mathrm{ad}}$ to be a solution of Problem $(\mathcal{P}_{\mathrm{C}})$ is that there exists $\lambda^\sharp \in \mathcal{V}$ such that:

1. $\left\langle \nabla J(u^\sharp) + [\Theta'(u^\sharp)]^\star \lambda^\sharp \,, u - u^\sharp \right\rangle \geq 0 \ \ \forall u \in \mathcal{U}^{\mathrm{ad}}$,
2. $\Theta(u^\sharp) \in -C$,
3. $\lambda^\sharp \in C^\star$,
4. $\left\langle \lambda^\sharp \,, \Theta(u^\sharp) \right\rangle = 0$ (*Complementary Slackness*).

Optimization with explicit constraints III

Let $L : \mathcal{U}^{\mathrm{ad}} \times C^\star \to \mathbb{R}$ be the Lagrangian associated to $(\mathcal{P}_{\mathrm{C}})$:

$$L(u, \lambda) = J(u) + \left\langle \lambda \, , \Theta(u) \right\rangle \; .$$

A point $(u^\sharp, \lambda^\sharp) \in \mathcal{U}^{\mathrm{ad}} \times C^\star$ is a saddle point of L if, for all $(u, \lambda) \in \mathcal{U}^{\mathrm{ad}} \times C^\star$

$$L(u^\sharp, \lambda) \leq L(u^\sharp, \lambda^\sharp) \leq L(u, \lambda^\sharp) \; .$$

- If $(u^\sharp, \lambda^\sharp)$ is a saddle point of L, then $u^\sharp$ is a solution of $(\mathcal{P}_{\mathrm{C}})$.
- If $u^\sharp$ is a solution of $(\mathcal{P}_{\mathrm{C}})$ and if the **KKT** conditions are met for some $\lambda^\sharp$, then $(u^\sharp, \lambda^\sharp)$ is a saddle point of L.

Moreover we have that

$$J(u^\sharp) = \min_{u \in \mathcal{U}^{\mathrm{ad}}} \max_{\lambda \in C^\star} L(u, \lambda) = \max_{\lambda \in C^\star} \min_{u \in \mathcal{U}^{\mathrm{ad}}} L(u, \lambda) = L(u^\sharp, \lambda^\sharp) \; .$$

Optimization with explicit constraints IV

Define the dual function associated to the Lagrangian L as

$$\Phi(\lambda) = \min_{u \in \mathcal{U}^{\mathrm{ad}}} L(u, \lambda) \ ,$$

and assume that $\arg\min L(\cdot, \lambda) = \left\{ \widehat{u}_\lambda \right\}$, so that $\nabla\Phi(\lambda) = \Theta(\widehat{u}_\lambda)$.

To compute the solution $u^\sharp$, use a gradient algorithm for Problem:

$$\max_{\lambda \in C^\star} \Phi(\lambda) \qquad \Big(\Leftrightarrow \max_{\lambda \in C^\star} \min_{u \in \mathcal{U}^{\mathrm{ad}}} L(u, \lambda) \Big) \ .$$

Uzawa's Algorithm

Choose $\lambda^{(0)} \in C^\star$. At each itération k,

- obtain the solution $u^{(k+1)} = \arg\min\limits_{u \in \mathcal{U}^{\mathrm{ad}}} J(u) + \left\langle \lambda^{(k)} , \Theta(u) \right\rangle$,
- update the multiplier $\lambda^{(k+1)} = \mathrm{proj}_{C^\star} \left(\lambda^{(k)} + \rho\Theta\big(u^{(k+1)}\big) \right)$.

Optimization with explicit constraints V

Uzawa's algorithm convergence theorem

H1 $\mathcal{U}^{\mathrm{ad}}$ is a closed convex subset of the Hilbert space $\mathcal{U}$,
C is a closed convex cone of the Hilbert space $\mathcal{V}$.

H2 J is a proper l.s.c. strongly convex function with modulus a,
Gâteaux différentiable.

H3 Θ is a C-convex, Lipschitz with constant τ.

H4 L admits a saddle point $(u^\sharp, \lambda^\sharp) \in \mathcal{U}^{\mathrm{ad}} \times C^\star$.

H5 ρ is such that $0 < \rho < 2a/\tau^2$.

R1 The sequence $\{u^{(k)}\}_{k\in\mathbb{N}}$ converges toward $u^\sharp$.

R2 The sequence $\{\lambda^{(k)}\}_{k\in\mathbb{N}}$ is bounded, and any of its cluster points $\overline{\lambda}$ is such that $(u^\sharp, \overline{\lambda})$ is a saddle point of L.

Uzawa's geometric interpretation I

For the sake of simplicity, we consider here equality constraints:

$$u^{(k+1)} \in \operatorname*{arg\,min}_{u \in \mathcal{U}^{\mathrm{ad}}} J(u) + \left\langle \lambda^{(k)} , \Theta(u) \right\rangle ,$$

$$\lambda^{(k+1)} = \lambda^{(k)} + \rho \Theta\big(u^{(k+1)}\big) .$$

The minimization step is equivalent to:

$$\min_{v \in \mathcal{V}} \min_{u \in \mathcal{U}^{\mathrm{ad}}} J(u) + \left\langle \lambda^{(k)} , v \right\rangle \quad \text{s.t.} \quad \Theta(u) - v = 0 .$$

Introducing the perturbation function G:

$$G(v) \; = \; \min_{u \in \mathcal{U}^{\mathrm{ad}}} J(u) \quad \text{s.t.} \quad \Theta(u) - v = 0 ,$$

this minimization step also writes:

$$\min_{v \in \mathcal{V}} G(v) + \left\langle \lambda^{(k)} , v \right\rangle .$$

Uzawa's geometric interpretation II

With the help of G, Uzawa's algorithm writes:

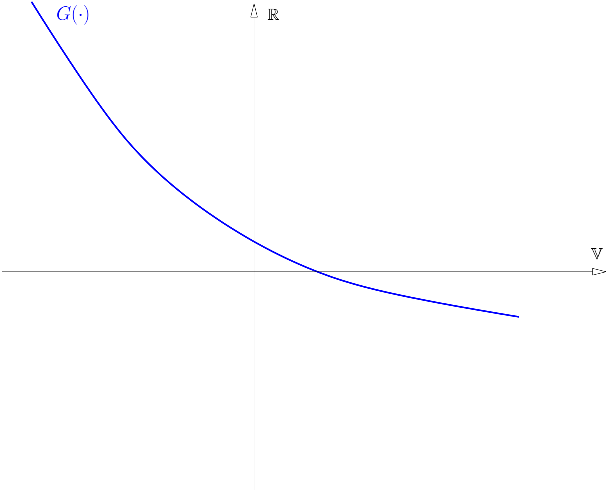
$$v^{(k+1)} \in \operatorname*{arg\,min}_{v \in \mathcal{V}} G(v) + \left\langle \lambda^{(k)} \, , v \right\rangle \, ,$$
$$\lambda^{(k+1)} = \lambda^{(k)} + \rho v^{(k+1)} \, .$$

From a (conceptual) geometric point of view, it amounts to:

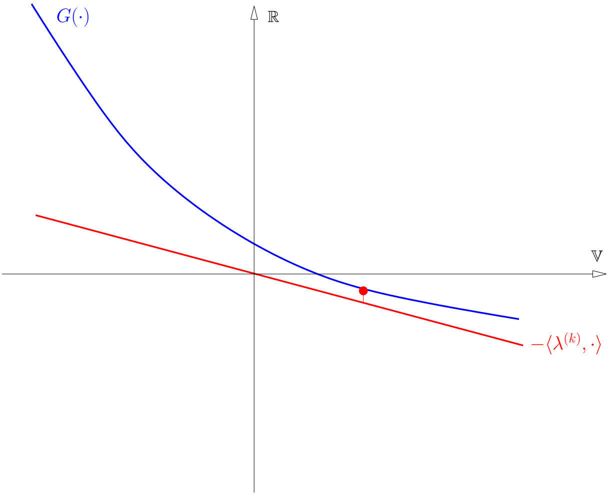
- **Step 1**: minimize the gap between $G(\cdot)$ et $\left\langle -\lambda^{(k)} \, , \cdot \right\rangle$.
- **Step 2**: adjust the slope $-\lambda^{(k)}$ if $v^{(k+1)} \neq 0$.

Recall that the initial problem consists in obtaining $G(0)$*...*

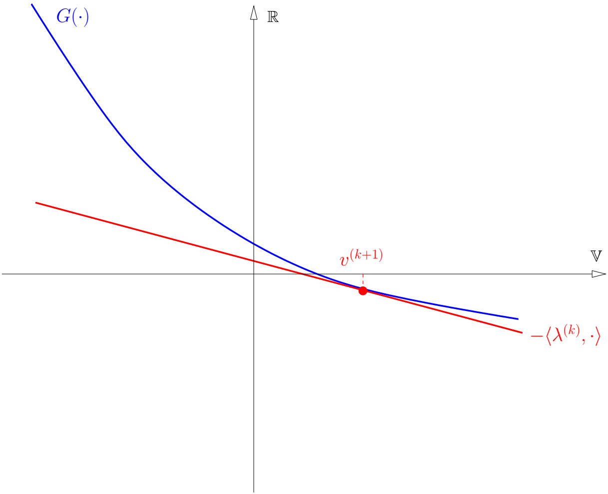
Uzawa's geometric interpretation III



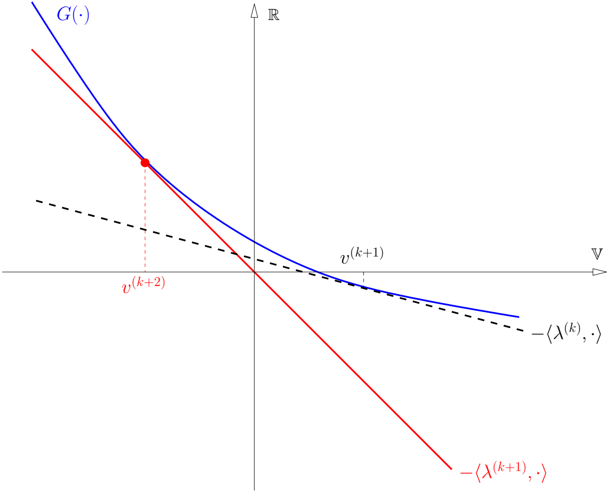
Uzawa's geometric interpretation III



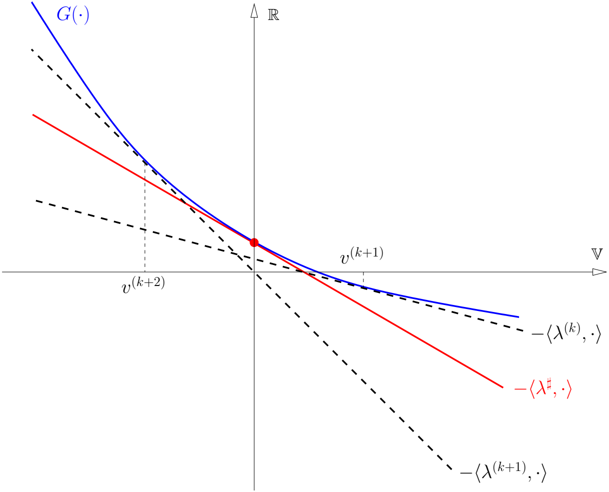
Uzawa's geometric interpretation III



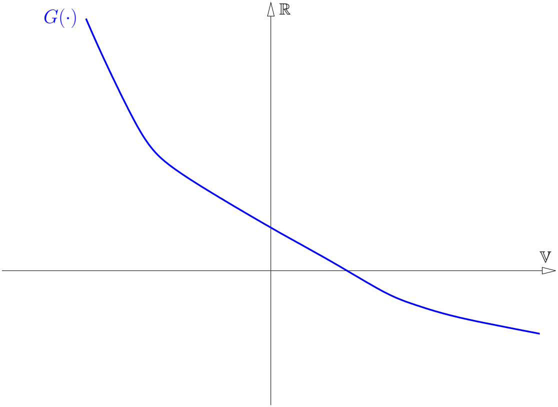
Uzawa's geometric interpretation III



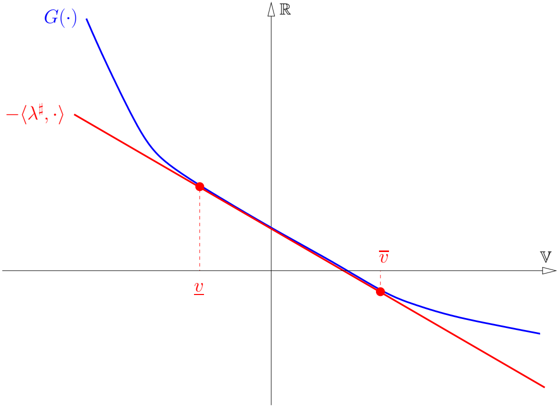
Uzawa's geometric interpretation III



Uzawa's geometric interpretation IV

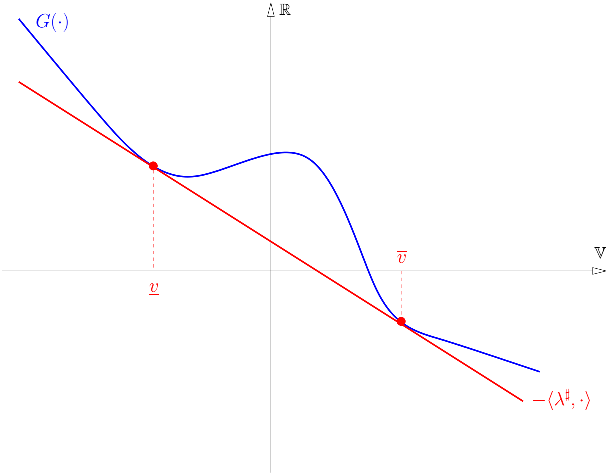


Uzawa's geometric interpretation IV

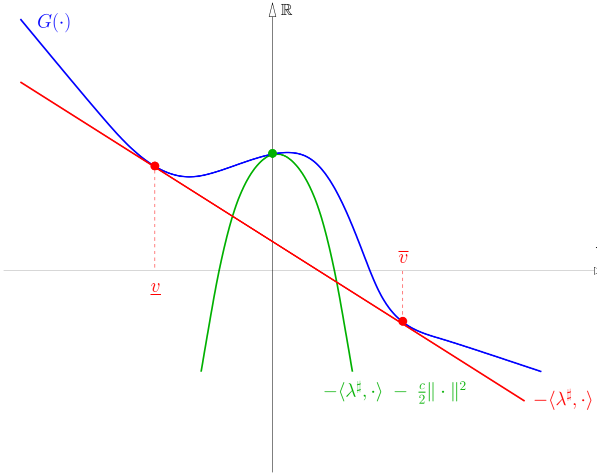


Even if $\{\lambda^{(k)}\}_{k\in\mathbb{N}}$ converges towards $\lambda^\sharp$, the constraint level $v^{(k)}$ oscillates between $\underline{v}$ and $\overline{v}$, but the value $v^\sharp = 0$ is never reached.

Uzawa's geometric interpretation V



Uzawa's geometric interpretation V



In the non convex case, use an augmented Lagrangian...

Additive model

Consider the following problem:

$$\min_{u \in \mathcal{U}^{\mathrm{ad}} \subset \mathcal{U}} J(u) \quad \text{subject to} \quad \Theta(u) - \theta = 0 \in \mathcal{V} ,$$

and consider a **decomposition** of the space $\mathcal{U} = \mathcal{U}_1 \times \ldots \times \mathcal{U}_N$, so that $u \in \mathcal{U}$ writes $u = \big(u_1, \ldots, u_N\big)$ with $u_i \in \mathcal{U}_i$. Assume that

- $\mathcal{U}^{\mathrm{ad}} = \mathcal{U}_1^{\mathrm{ad}} \times \ldots \times \mathcal{U}_N^{\mathrm{ad}} \qquad \mathcal{U}_i^{\mathrm{ad}} \subset \mathcal{U}_i$,
- $J(u) = J_1(u_1) + \ldots + J_N(u_N) \qquad u_i \in \mathcal{U}_i$,
- $\Theta(u) = \Theta_1(u_1) + \ldots + \Theta_N(u_N) \qquad u_i \in \mathcal{U}_i$.

Then the problem displays the following **additive structure**:

$$\min_{\substack{u_1 \in \mathcal{U}_1^{\mathrm{ad}} \\ \vdots \\ u_N \in \mathcal{U}_N^{\mathrm{ad}}}} \sum_{i=1}^{N} J_i(u_i) \quad \text{subject to} \quad \sum_{i=1}^{N} \Theta_i(u_i) - \theta = 0 .$$

Note that the coupling between the i's only arises from the constraint Θ.

Additive model — Price decomposition I

$$\min_{u \in \mathcal{U}^{\mathrm{ad}}} \sum_{i=1}^{N} J_i(u_i) \quad \text{subject to} \quad \sum_{i=1}^{N} \Theta_i(u_i) - \theta = 0 \; .$$

1. Form the Lagrangian of the problem. We assume that a saddle point exists, so that solving the initial problem is equivalent to:

$$\max_{\lambda \in \mathcal{V}} \min_{u \in \mathcal{U}^{\mathrm{ad}}} \sum_{i=1}^{N} \Big(J_i(u_i) + \langle \lambda \, , \Theta_i(u_i) \rangle \Big) - \langle \lambda \, , \theta \rangle \; ,$$

2. Solve this problem by the Uzawa algorithm:

$$u_i^{(k+1)} \in \arg\min_{u_i \in \mathcal{U}_i^{\mathrm{ad}}} J_i(u_i) + \langle \lambda^{(k)} \, , \Theta_i(u_i) \rangle \; , \quad i = 1, \ldots, N \; ,$$

$$\lambda^{(k+1)} = \lambda^{(k)} + \rho \Big(\sum_{i=1}^{N} \Theta_i\big(u_i^{(k+1)}\big) - \theta \Big) \; .$$

Additive model — Price decomposition I

$$\min_{u\in\mathcal{U}^{\mathrm{ad}}} \sum_{i=1}^{N} J_i(u_i) \quad \text{subject to} \quad \sum_{i=1}^{N} \Theta_i(u_i) - \theta = 0 \ .$$

1. Form the Lagrangian of the problem. We assume that a saddle point exists, so that solving the initial problem is equivalent to:

$$\max_{\lambda\in\mathcal{V}} \sum_{i=1}^{N} \min_{u_i\in\mathcal{U}_i^{\mathrm{ad}}} \Big(J_i(u_i) + \langle \lambda \,, \Theta_i(u_i) \rangle \Big) - \langle \lambda \,, \theta \rangle \ ,$$

2. Solve this problem by the Uzawa algorithm:

$$u_i^{(k+1)} \in \arg\min_{u_i\in\mathcal{U}_i^{\mathrm{ad}}} J_i(u_i) + \langle \lambda^{(k)} \,, \Theta_i(u_i) \rangle \ , \quad i = 1 \ldots, N \ ,$$

$$\lambda^{(k+1)} = \lambda^{(k)} + \rho \Big(\sum_{i=1}^{N} \Theta_i\big(u_i^{(k+1)}\big) - \theta \Big) \ .$$

Additive model — Price decomposition I

$$\min_{u \in \mathcal{U}^{\mathrm{ad}}} \sum_{i=1}^{N} J_i(u_i) \quad \text{subject to} \quad \sum_{i=1}^{N} \Theta_i(u_i) - \theta = 0 \; .$$

1. Form the Lagrangian of the problem. We assume that a saddle point exists, so that solving the initial problem is equivalent to:

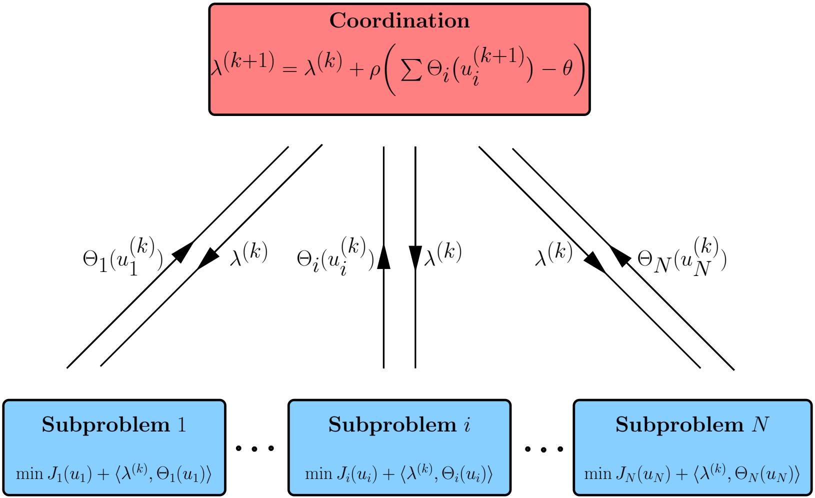
$$\max_{\lambda \in \mathcal{V}} \sum_{i=1}^{N} \min_{u_i \in \mathcal{U}_i^{\mathrm{ad}}} \Big(J_i(u_i) + \left\langle \lambda \, , \Theta_i(u_i) \right\rangle \Big) - \left\langle \lambda \, , \theta \right\rangle \, ,$$

2. Solve this problem by the Uzawa algorithm:

$$u_i^{(k+1)} \in \operatorname*{arg\,min}_{u_i \in \mathcal{U}_i^{\mathrm{ad}}} J_i(u_i) + \left\langle \lambda^{(k)} \, , \Theta_i(u_i) \right\rangle \, , \quad i = 1 \ldots , N \, ,$$

$$\lambda^{(k+1)} = \lambda^{(k)} + \rho \left(\sum_{i=1}^{N} \Theta_i \Big(u_i^{(k+1)} \Big) - \theta \right) \, .$$

Additive model — Price decomposition II



Additive model — Resource allocation I

$$\min_{u \in \mathcal{U}^{\mathrm{ad}}} \sum_{i=1}^{N} J_i(u_i) \quad \text{subject to} \quad \sum_{i=1}^{N} \Theta_i(u_i) - \theta = 0 \; .$$

1. Write the constraint in a equivalent manner by introducing new variables $v = (v_1, \dots, v_N)$ (the so-called "allocation"):

$$\sum_{i=1}^{N} \Theta_i(u_i) - \theta = 0 \quad \Leftrightarrow \quad \Theta_i(u_i) - v_i = 0 \;\text{ and }\; \sum_{i=1}^{N} v_i = \theta \; ,$$

and minimize the criterion w.r.t. u and v:

$$\min_{v \in \mathcal{V}^N} \sum_{i=1}^{N} \Big(\min_{u_i \in \mathcal{U}_i^{\mathrm{ad}}} J_i(u_i) \;\text{ s.t. }\; \Theta_i(u_i) - v_i = 0 \Big) \;\text{ s.t. }\; \sum_{i=1}^{N} v_i = \theta \; ,$$

Additive model — Resource allocation II

$$\min_{v\in\mathcal{V}^N} \sum_{i=1}^{N} \underbrace{\Big(\min_{u_i\in\mathcal{U}_i^{\mathrm{ad}}} J_i(u_i) \ \text{ s.t. } \ \Theta_i(u_i) - v_i = 0 \Big)}_{G_i(v_i)} \ \text{ s.t. } \ \sum_{i=1}^{N} v_i = \theta \ ,$$

$$\Updownarrow$$

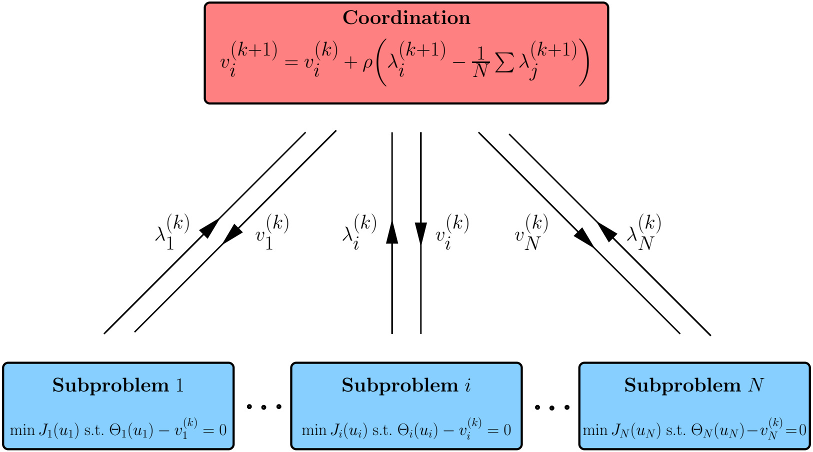
$$\min_{v\in\mathcal{V}^N} \sum_{i=1}^{N} G_i(v_i) \quad \text{s.t.} \quad \sum_{i=1}^{N} v_i = \theta \ .$$

2. Solve the last problem using a projected gradient method:

$$G_i(v_i^{(k)}) = \min_{u_i\in\mathcal{U}_i^{\mathrm{ad}}} J_i(u_i) \ \text{ s.t. } \ \Theta_i(u_i) - v_i^{(k)} = 0 \ \rightsquigarrow \ \lambda_i^{(k+1)} \ ,$$

$$v_i^{(k+1)} = v_i^{(k)} + \rho \Big(\lambda_i^{(k+1)} - \frac{1}{N} \sum_{j=1}^{N} \lambda_j^{(k+1)} \Big) \ .$$

Additive model — Resource allocation III



Additive model — Prediction

$$\min_{u \in \mathcal{U}^{\mathrm{ad}}} \sum_{i=1}^{N} J_i(u_i) \quad \text{subject to} \quad \sum_{i=1}^{N} \Theta_i(u_i) - \theta = 0 \; .$$

We assume for the moment that the constraint is scalar...

1. Choose the unit that will drive the constraint (e.g. unit 1) and split the constraint according to that choice:

$$\Theta_1(u_1) - v = 0 \qquad , \qquad \sum_{i \neq 1} \Theta_i(u_i) - \theta + v = 0 \; .$$

2. Formulate the problem obtained by dualizing only the second part of the constraint:

$$\max_{\lambda \in \mathbb{R}} \min_{v \in \mathcal{V}} \Big(\min_{u \in \mathcal{U}^{\mathrm{ad}}} \sum_{i=1}^{N} J_i(u_i) + \Big\langle \lambda \, , \sum_{i \neq 1} \Theta_i(u_i) - \theta + v \Big\rangle \Big)$$

$$\text{subject to} \;\; \Theta_1(u_1) - v = 0 \; .$$

Additive model — Prediction II

3. With $v = v^{(k)}$ and $\lambda = \lambda^{(k)}$ fixed, the problem decomposes:

$$\min_{u_1 \in \mathcal{U}_1^{\mathrm{ad}}} J_1(u_1) \text{ s.t. } \Theta_1(u_1) - v^{(k)} = 0 \quad \rightsquigarrow \quad \lambda_1^{(k+1)} \; ,$$

$$\min_{u_i \in \mathcal{U}_i^{\mathrm{ad}}} J_i(u_i) + \left\langle \lambda^{(k)} , \Theta_i(u_i) \right\rangle \; \forall i \neq 1 \quad \rightsquigarrow \quad \Theta_i(u_i^{(k+1)}) \; .$$

4. Update v and λ by solving the optimality conditions in λ and v of the global problem:

$$v^{(k+1)} = \theta - \sum_{i \neq 1} \Theta_i(u_i^{(k+1)}) \; ,$$

$$\lambda^{(k+1)} = \lambda_1^{(k+1)} \; .$$

In case of multiple constraints, incorporate them one by one. A choice has to be done for each constraint. The constraints are thus distributed among the units.

Additive model — Prediction II

3. With $v = v^{(k)}$ and $\lambda = \lambda^{(k)}$ fixed, the problem decomposes:

$$\min_{u_1 \in \mathcal{U}_1^{\mathrm{ad}}} J_1(u_1) \ \text{s.t.}\ \Theta_1(u_1) - v^{(k)} = 0 \quad \rightsquigarrow \quad \lambda_1^{(k+1)} ,$$

$$\min_{u_i \in \mathcal{U}_i^{\mathrm{ad}}} J_i(u_i) + \left\langle \lambda^{(k)} , \Theta_i(u_i) \right\rangle \ \forall i \neq 1 \quad \rightsquigarrow \quad \Theta_i(u_i^{(k+1)}) .$$

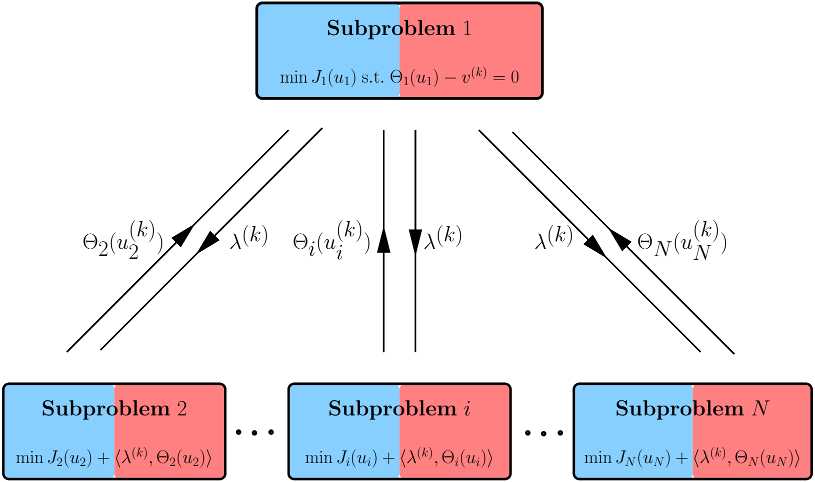
4. Update v and λ by solving the optimality conditions in λ and v of the global problem:

$$v^{(k+1)} = \theta - \sum_{i \neq 1} \Theta_i(u_i^{(k+1)}) ,$$

$$\lambda^{(k+1)} = \lambda_1^{(k+1)} .$$

In case of multiple constraints, incorporate them one by one. A choice has to be done for each constraint. The constraints are thus distributed among the units.

Additive model — Prediction III



Additive model: conclusions

1. **Price decomposition**
    - Pros: "non-destructive" method.
    - Cons: admissible solution once convergence achieved.
2. **Resource allocation**
    - Pros: admissible solution at each iteration.
    - Cons: potential existence of unfeasible subproblems.
3. **Prediction**
    - Pros and Cons: depending on the constraints distribution...

Straightforward extension to inequality constraints...

General model — Auxiliary Problem Principle

The 3 decomposition schemes we have presented seem to depend crucially on the additive structure of the underlying problems...
In fact they can be extended to general problems:

$$\min_{u \in \mathcal{U}^{\mathrm{ad}}} J(u_1, \ldots, u_N) \quad \text{s.t.} \quad \Theta(u_1, \ldots, u_N) - \theta = 0 \;.$$

This generalization is achieved by the Auxiliary Problem Principle (**APP**), whose aim is to recover additivity by replacing the two functions J and Θ by their first-order approximation around the current point $u^{(k)}$:

$$J(u) \rightsquigarrow \sum_{i=1}^{N} \left\langle \nabla_{u_i} J(u^{(k)}) \,,\, u_i \right\rangle \quad , \quad \Theta(u) \rightsquigarrow \sum_{i=1}^{N} \Theta'_{u_i}(u^{(k)}).u_i \;.$$

The solution $u^{(k+1)}$ of the auxiliary problem built around $u^{(k)}$ is used to formulate the next auxiliary problem (iterative process).

General model — Auxiliary Problem Principle

The 3 decomposition schemes we have presented seem to depend crucially on the additive structure of the underlying problems...
In fact they can be extended to general problems:

$$\min_{u \in \mathcal{U}^{\mathrm{ad}}} J(u_1, \dots, u_N) \quad \text{s.t.} \quad \Theta(u_1, \dots, u_N) - \theta = 0 \; .$$

This generalization is achieved by the Auxiliary Problem Principle (**APP**), whose aim is to recover additivity by replacing the two functions J and Θ by their first-order approximation around the current point $u^{(k)}$:

$$J(u) \rightsquigarrow \sum_{i=1}^{N} \left\langle \nabla_{u_i} J(u^{(k)}) \, , u_i \right\rangle \quad , \quad \Theta(u) \rightsquigarrow \sum_{i=1}^{N} \Theta'_{u_i}(u^{(k)}).u_i \; .$$

The solution $u^{(k+1)}$ of the auxiliary problem built around $u^{(k)}$ is used to formulate the next auxiliary problem (iterative process).

APP without explicit constraint I

$$\min_{u \in \mathcal{U}^{\mathrm{ad}}} J(u) .$$

1. Replace $J(u)$ by its first order approximation around $u^{(k)}$:

$$J(u^{(k)}) + \left\langle \nabla J(u^{(k)}) , u - u^{(k)} \right\rangle .$$

2. Choose a strongly convex function K, some $\epsilon > 0$ and form:

$$\frac{1}{\epsilon}\Big(K(u) - K(u^{(k)}) - \left\langle \nabla K(u^{(k)}) , u - u^{(k)} \right\rangle \Big) .$$

Add these two terms to obtain the auxiliary problem at iteration k:

$$\min_{u \in \mathcal{U}^{\mathrm{ad}}} K(u) + \left\langle \epsilon \nabla J(u^{(k)}) - \nabla K(u^{(k)}) , u \right\rangle ,$$

whose unique solution is denoted by $u^{(k+1)}$.

APP without explicit constraint II

Convergence theorem

H1 $\mathcal{U}^{\mathrm{ad}}$ is a closed convex subset of the Hilbert space $\mathcal{U}$.

H2 J is a proper l.s.c. convex function, coercive over $\mathcal{U}^{\mathrm{ad}}$, and its derivative J' is Lipschitz with constant A.

H3 K is a proper l.s.c. strongly convex function with modulus b, and its derivative K' is Lipschitz with constant B.

H4 ϵ is a coefficient such that $0 < \epsilon < 2b/A$.

R1 $\left\{J(u^{(k)})\right\}_{k\in\mathbb{N}}$ is a strictly decreasing real sequence which converges towards $J(u^\sharp)$.

R2 $\left\{u^{(k)}\right\}_{k\in\mathbb{N}}$ is a bounded sequence, and each of its cluster points is a solution of the initial problem.

Moreover, if J is strongly convex, then $\left\{u^{(k)}\right\}_{k\in\mathbb{N}}$ converges to $u^\sharp$.

APP without explicit constraint III

Consider the auxiliary problem obtained at iteration k:

$$\min_{u \in \mathcal{U}^{\mathrm{ad}}} \; K(u) + \left\langle \epsilon \nabla J(u^{(k)}) - \nabla K(u^{(k)}) \, , u \right\rangle .$$

Assume that there exists a decomposition $\mathcal{U}_1 \times \ldots \times \mathcal{U}_N$ of $\mathcal{U}$, that is, $u \in \mathcal{U}$ writes $u = \big(u_1, \ldots, u_N\big)$ with $u_i \in \mathcal{U}_i$, such that:

$$\mathcal{U}^{\mathrm{ad}} = \mathcal{U}_1^{\mathrm{ad}} \times \ldots \times \mathcal{U}_N^{\mathrm{ad}} \quad \text{with} \quad \mathcal{U}_i^{\mathrm{ad}} \subset \mathcal{U}_i \, .$$

A additive choice of K leads to decomposition. Indeed, using

$$K(u) = \sum_{i=1}^{N} K_i(u_i) \, ,$$

the k-th auxiliary problem can be decomposed in N subproblems:

$$\min_{u_i \in \mathcal{U}_i^{\mathrm{ad}}} \; K_i(u_i) + \left\langle \epsilon \nabla_{u_i} J(u^{(k)}) - \nabla_{u_i} K(u^{(k)}) \, , u_i \right\rangle \, , \quad i = 1, \ldots, N \, .$$

APP without explicit constraint IV

Variants of the algorithm

- Take into account an additional cost function J^{Σ}:

$$\min_{u \in \mathcal{U}^{\mathrm{ad}}} K(u) + \left\langle \epsilon \nabla J(u^{(k)}) - \nabla K(u^{(k)}) , u \right\rangle + \epsilon J^{\Sigma}(u) .$$

- K and ϵ may depend on the iteration index k:

$$\min_{u \in \mathcal{U}^{\mathrm{ad}}} K^{(k)}(u) + \left\langle \epsilon^{(k)} \nabla J(u^{(k)}) - \nabla K^{(k)}(u^{(k)}) , u \right\rangle .$$

- Use $\epsilon \equiv 1$ by adding an under-relaxation step in the algorithm:

$$u^{(k+\frac{1}{2})} = \operatorname*{arg\,min}_{u \in \mathcal{U}^{\mathrm{ad}}} K(u) + \left\langle \nabla J(u^{(k)}) - \nabla K(u^{(k)}) , u \right\rangle ,$$

$$u^{(k+1)} = \rho \, u^{(k+\frac{1}{2})} + (1 - \rho) \, u^{(k)} , \quad 0 < \rho < 1 .$$

APP with explicit constraints I

$$\min_{u \in \mathcal{U}^{\mathrm{ad}}} J(u) \quad \text{s.t.} \quad \Theta(u) \in -C \ ,$$

Denote by $L(u, \lambda) = J(u) + \langle \lambda \, , \Theta(u) \rangle$ the associated Lagrangian.

1. Replace L by its first order approximation around $(u^{(k)}, \lambda^{(k)})$:

$$L(u^{(k)}, \lambda^{(k)}) + \langle \nabla_u L(u^{(k)}, \lambda^{(k)}) \, , u - u^{(k)} \rangle + \langle \nabla_\lambda L(u^{(k)}, \lambda^{(k)}) \, , \lambda - \lambda^{(k)} \rangle \ .$$

2. Choose a convex-concave operator $M(u, \lambda)$ and some $\epsilon > 0$.

Use these elements to form the auxiliary Lagrangian at iteration k:

$$M(u, \lambda) + \langle (\epsilon \nabla_u L - \nabla_u M)(u^{(k)}, \lambda^{(k)}) \, , u \rangle + \langle (\epsilon \nabla_\lambda L - \nabla_\lambda M)(u^{(k)}, \lambda^{(k)}) \, , \lambda \rangle \ ,$$

and obtain a point $(u^{(k+1)}, \lambda^{(k+1)})$ satisfying optimality conditions.

APP with explicit constraints II

We denote by $\mathfrak{L}^{(k)}$ the auxiliary Lagrangian at iteration k:

$$\mathfrak{L}^{(k)}(u,\lambda) = M(u,\lambda) \; + \; \left\langle \epsilon \nabla_u L(u^{(k)},\lambda^{(k)}) - \nabla_u M(u^{(k)},\lambda^{(k)}) \, , u \right\rangle + \left\langle \epsilon \nabla_\lambda L(u^{(k)},\lambda^{(k)}) - \nabla_\lambda M(u^{(k)},\lambda^{(k)}) \, , \lambda \right\rangle \, .$$

We have two possible algorithms to solve the auxiliary problem.

1. SIM: solve simultaneously the optimality conditions:

$$u^{(k+1)} = \operatorname*{arg\,min}_{u \in \mathcal{U}^{\mathrm{ad}}} \mathfrak{L}^{(k)}(u,\lambda^{(k+1)}) \; ,$$

$$\lambda^{(k+1)} = \operatorname*{arg\,max}_{\lambda \in C^\star} \mathfrak{L}^{(k)}(u^{(k+1)},\lambda) \; .$$

2. SEQ: solve sequentially the optimality conditions:

$$u^{(k+1)} = \operatorname*{arg\,min}_{u \in \mathcal{U}^{\mathrm{ad}}} \mathfrak{L}^{(k)}(u,\lambda^{(k)}) \; ,$$

$$\lambda^{(k+1)} = \operatorname*{arg\,max}_{\lambda \in C^\star} \mathfrak{L}^{(k)}(u^{(k+1)},\lambda) \; .$$

APP with explicit constraints II

We denote by $\mathfrak{L}^{(k)}$ the auxiliary Lagrangian at iteration k:

$$\mathfrak{L}^{(k)}(u,\lambda) = M(u,\lambda) + \left\langle \epsilon \nabla_u L(u^{(k)},\lambda^{(k)}) - \nabla_u M(u^{(k)},\lambda^{(k)}) \, , u \right\rangle + \left\langle \epsilon \nabla_\lambda L(u^{(k)},\lambda^{(k)}) - \nabla_\lambda M(u^{(k)},\lambda^{(k)}) \, , \lambda \right\rangle .$$

We have two possible algorithms to solve the auxiliary problem.

1. SIM: solve simultaneously the optimality conditions:

$$u^{(k+1)} = \operatorname*{arg\,min}_{u \in \mathcal{U}^{\mathrm{ad}}} \mathfrak{L}^{(k)}(u, \lambda^{(k+1)}) \, ,$$

$$\lambda^{(k+1)} = \operatorname*{arg\,max}_{\lambda \in C^\star} \mathfrak{L}^{(k)}(u^{(k+1)}, \lambda) \, .$$

2. SEQ: solve sequentially the optimality conditions:

$$u^{(k+1)} = \operatorname*{arg\,min}_{u \in \mathcal{U}^{\mathrm{ad}}} \mathfrak{L}^{(k)}(u, \lambda^{(k)}) \, ,$$

$$\lambda^{(k+1)} = \operatorname*{arg\,max}_{\lambda \in C^\star} \mathfrak{L}^{(k)}(u^{(k+1)}, \lambda) \, .$$

APP with explicit constraints: one-level algorithm I

Possible choice: $M(u,\lambda) = K(u) + \langle \lambda \,, \Omega(u) \rangle$ and Algorithm SIM.

The expression of the auxiliary Lagrangian is as follows:

$$\begin{aligned}
\mathfrak{L}^{(k)}(u,\lambda) = M(u,\lambda) \; &+ \; \left\langle \epsilon \nabla_u L(u^{(k)},\lambda^{(k)}) - \nabla_u M(u^{(k)},\lambda^{(k)}) \,, u \right\rangle \\
&+ \; \left\langle \epsilon \nabla_\lambda L(u^{(k)},\lambda^{(k)}) - \nabla_\lambda M(u^{(k)},\lambda^{(k)}) \,, \lambda \right\rangle \\
= K(u) \; &+ \; \left\langle \epsilon \nabla J(u^{(k)}) - \nabla K(u^{(k)}) \,, u \right\rangle \\
&+ \; \left\langle \lambda^{(k)} \,, \left(\epsilon \Theta'(u^{(k)}) - \Omega'(u^{(k)})\right).u \right\rangle \\
&+ \; \left\langle \lambda \,, \Omega(u) + \epsilon \Theta(u^{(k)}) - \Omega(u^{(k)}) \right\rangle .
\end{aligned}$$

APP with explicit constraints: one-level algorithm I

Possible choice: $M(u,\lambda) = K(u) + \langle \lambda \, , \Omega(u) \rangle$ and Algorithm SIM.

The expression of the auxiliary Lagrangian is as follows:

$$
\begin{aligned}
\mathfrak{L}^{(k)}(u,\lambda) = M(u,\lambda) \; &+ \; \big\langle \epsilon \nabla_u L(u^{(k)},\lambda^{(k)}) - \nabla_u M(u^{(k)},\lambda^{(k)}) \, , u \big\rangle \\
&+ \; \big\langle \epsilon \nabla_\lambda L(u^{(k)},\lambda^{(k)}) - \nabla_\lambda M(u^{(k)},\lambda^{(k)}) \, , \lambda \big\rangle
\end{aligned}
$$

$$
\begin{aligned}
= K(u) \; &+ \; \big\langle \epsilon \nabla J(u^{(k)}) - \nabla K(u^{(k)}) \, , u \big\rangle \\
&+ \; \big\langle \lambda^{(k)} \, , \big(\epsilon \Theta'(u^{(k)}) - \Omega'(u^{(k)})\big).u \big\rangle \\
&+ \; \big\langle \lambda \, , \Omega(u) + \epsilon \Theta(u^{(k)}) - \Omega(u^{(k)}) \big\rangle \, .
\end{aligned}
$$

APP with explicit constraints: one-level algorithm II

The saddle point $\left(u^{(k+1)}, \lambda^{(k+1)}\right)$ of $\mathfrak{L}^{(k)}$ is obtained by solving the associated constrained optimization problem:

$$\min_{u \in \mathcal{U}^{\mathrm{ad}}} K(u) + \left\langle \epsilon \nabla J(u^{(k)}) - \nabla K(u^{(k)}) \, , u \right\rangle + \left\langle \lambda^{(k)} \, , \left(\epsilon \Theta'(u^{(k)}) - \Omega'(u^{(k)})\right) . u \right\rangle \, ,$$

$$\text{subject to} \quad \Omega(u) - \Omega(u^{(k)}) + \epsilon \Theta(u^{(k)}) \in -C \, .$$

The convergence proof of this algorithm is available for problems involving a quadratic cost function and linear equality constraints. Moreover, a geometric condition, namely $\Theta J^{-1} \Omega^{\star} + \Omega J^{-1} \Theta^{\star} > 0$ (weak coupling through the constraints) has to be met.

APP with explicit constraints: one-level algorithm III

With regard to decomposition, consider the following choices:

$$K(u) = \sum_{i=1}^{N} K_i(u_i) \quad , \quad \Omega(u) = \begin{pmatrix} \Omega_1(u_1) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \Omega_N(u_N) \end{pmatrix} ,$$

that is,

- an additive auxiliary cost function K,
- a block diagonal auxiliary constraint Ω,

and assume that $\mathcal{U}^{\mathrm{ad}} = \mathcal{U}_1^{\mathrm{ad}} \times \ldots \times \mathcal{U}_N^{\mathrm{ad}}$.

Then the auxiliary problem can be decomposed in N subproblems.

This algorithm is in fact a generalization of the decomposition by prediction that has been studied for additive models. The choice of Ω as a block-diagonal operator corresponds to the distribution of the constraints among the units.

APP with explicit constraints: two-level algorithm I

Alternative choice: $M(u,\lambda) = K(u) - \dfrac{\|\lambda\|^2}{2\alpha}$ and Algorithm SEQ.

The expression of the auxiliary Lagrangian is as follows:

$$\begin{aligned}\mathfrak{L}^{(k)}(u,\lambda) = M(u,\lambda) &+ \left\langle \epsilon\nabla_u L(u^{(k)},\lambda^{(k)}) - \nabla_u M(u^{(k)},\lambda^{(k)}) \, , u \right\rangle \\ &+ \left\langle \epsilon\nabla_\lambda L(u^{(k)},\lambda^{(k)}) - \nabla_\lambda M(u^{(k)},\lambda^{(k)}) \, , \lambda \right\rangle ,\end{aligned}$$

so that

$$\begin{aligned}\mathfrak{L}^{(k)}(u,\lambda^{(k)}) &\leftrightarrow K(u) + \left\langle \epsilon\nabla J(u^{(k)}) - \nabla K(u^{(k)}) \, , u \right\rangle \\ &\qquad + \epsilon\left\langle \lambda^{(k)} \, , \Theta'(u^{(k)}).u \right\rangle , \\ \mathfrak{L}^{(k)}(u^{(k+1)},\lambda) &\leftrightarrow -\frac{1}{2}\|\lambda\|^2 + \left\langle \alpha\epsilon\Theta(u^{(k+1)}) + \lambda^{(k)} \, , \lambda \right\rangle .\end{aligned}$$

APP with explicit constraints: two-level algorithm I

Alternative choice: $M(u,\lambda) = K(u) - \dfrac{\|\lambda\|^2}{2\alpha}$ and Algorithm SEQ.

The expression of the auxiliary Lagrangian is as follows:

$$\begin{aligned}\mathfrak{L}^{(k)}(u,\lambda) = M(u,\lambda) \ &+ \ \big\langle \epsilon\nabla_u L(u^{(k)},\lambda^{(k)}) - \nabla_u M(u^{(k)},\lambda^{(k)})\,,u \big\rangle \\ &+ \ \big\langle \epsilon\nabla_\lambda L(u^{(k)},\lambda^{(k)}) - \nabla_\lambda M(u^{(k)},\lambda^{(k)})\,,\lambda \big\rangle \ ,\end{aligned}$$

so that

$$\begin{aligned}\mathfrak{L}^{(k)}(u,\lambda^{(k)}) \quad &\leftrightarrow \ K(u) \ + \ \big\langle \epsilon\nabla J(u^{(k)}) - \nabla K(u^{(k)})\,,u \big\rangle \\ &\qquad\qquad + \ \epsilon\big\langle \lambda^{(k)}\,,\Theta'(u^{(k)}).u \big\rangle \ , \\ \mathfrak{L}^{(k)}(u^{(k+1)},\lambda) \ &\leftrightarrow \ -\frac{1}{2}\,\|\lambda\|^2 \ + \ \big\langle \alpha\epsilon\Theta(u^{(k+1)}) + \lambda^{(k)}\,,\lambda \big\rangle \ .\end{aligned}$$

APP with explicit constraints: two-level algorithm II

The optimization problems are solved sequentially. Solving the first problem $\min_{u \in \mathcal{U}^{\mathrm{ad}}} \mathfrak{L}^{(k)}(u, \lambda^{(k)})$ leads to

$$\min_{u \in \mathcal{U}^{\mathrm{ad}}} K(u) + \left\langle \epsilon \nabla J(u^{(k)}) - \nabla K(u^{(k)}) \, , u \right\rangle \; + \epsilon \left\langle \lambda^{(k)} \, , \Theta'(u^{(k)}).u \right\rangle \, ,$$

whose solution is denoted $u^{(k+1)}$, and solving the second problem $\max_{\lambda \in C^\star} \mathfrak{L}^{(k)}(u^{(k+1)}, \lambda)$ is equivalent to

$$\lambda^{(k+1)} = \mathrm{proj}_{C^\star}\big(\lambda^{(k)} + \underbrace{\alpha \, \epsilon}_{\rho} \Theta(u^{(k+1)})\big) \; ,$$

that is, an update of the multiplier λ.

The convergence proof of this algorithm can be established under standard assumptions in the convex (sub)differentiable framework.

APP with explicit constraints: two-level algorithm III

Convergence theorem

H1 $\mathcal{U}^{\mathrm{ad}}$ is a closed convex subset of the Hilbert space $\mathcal{U}$, and C is a closed convex cone of the Hilbert space $\mathcal{V}$.

H2 J is a proper l.s.c. strongly convex function with modulus a, and its derivative J' is Lipschitz with constant A.

H3 Θ is a C-convex, Lipschitz with constant τ, differentiable.

H4 A saddle point $(u^\sharp, \lambda^\sharp)$ of L exists.

H5 K is a proper l.s.c. strongly convex function with modulus b, and its derivative K' is Lipschitz with constant B.

H6 ϵ and ρ are such that $0 < \epsilon < b/A$ and $0 < \rho < a/\tau^2$.

R1 The sequence $\{u^{(k)}\}_{k\in\mathbb{N}}$ converges toward $u^\sharp$.

R2 The sequence $\{\lambda^{(k)}\}_{k\in\mathbb{N}}$ is bounded, and any of its cluster points $\overline{\lambda}$ is such that $(u^\sharp, \overline{\lambda})$ is a saddle point of L.

APP with explicit constraints: two-level algorithm IV

This algorithm corresponds to a generalization of both Uzawa and Arrow-Hurwicz algorithms. Roughly speaking,

- $K(u) = J(u)$ and $\epsilon = 1 \rightsquigarrow$ **Uzawa**.
- $K(u) = \frac{1}{2}\|u\|^2 \rightsquigarrow$ **Arrow-Hurwicz**.

Choosing an additive auxiliary function K:

$$K(u) = \sum_{i=1}^{N} K_i(u_i) ,$$

and assuming that $\mathcal{U}^{\mathrm{ad}} = \mathcal{U}_1^{\mathrm{ad}} \times \ldots \times \mathcal{U}_N^{\mathrm{ad}}$, the minimization step in the previous algorithm splits into N independent subproblems:

$$\min_{u_i \in \mathcal{U}_i^{\mathrm{ad}}} K_i(u_i) + \left\langle \epsilon \nabla_{u_i} J(u^{(k)}) - \nabla_{u_i} K(u^{(k)}), u_i \right\rangle + \epsilon \left\langle \lambda^{(k)}, \Theta'_{u_i}(u^{(k)}).u_i \right\rangle .$$

APP with explicit constraints: augmented Lagrangian I

For the sake of simplicity, consider an optimization problem under equality constraints:

$$\min_{u \in \mathcal{U}^{\mathrm{ad}}} J(u) \quad \text{s.t.} \quad \Theta(u) = 0 \ ,$$

The two-level APP algorithm writes in the following equivalent form:

$$u^{(k+1)} \in \operatorname*{arg\,min}_{u \in \mathcal{U}^{\mathrm{ad}}} K(u) + \left\langle \epsilon \nabla_u L(u^{(k)}, \lambda^{(k)}) - \nabla K(u^{(k)}) \, , u \right\rangle \ ,$$

$$\lambda^{(k+1)} = \lambda^{(k)} + \rho \nabla_\lambda L(u^{(k+1)}, \lambda^{(k)}) \ ,$$

L being the standard Lagrangian: $L(u, \lambda) = J(u) + \left\langle \lambda \, , \Theta(u) \right\rangle$.

APP with explicit constraints: augmented Lagrangian II

Introduce now the augmented Lagrangian L_c, whose expression in the case of equality constraints is given by

$$L_c(u, \lambda) = L(u, \lambda) + \frac{c}{2} \left\| \Theta(u) \right\|^2 \ .$$

Applying the APP methodology to this new Lagrangian leads to the following two-level algorithm:

$$u^{(k+1)} \in \mathop{\arg\min}_{u \in \mathcal{U}^{\mathrm{ad}}} K(u) + \left\langle \epsilon \nabla_u L_c(u^{(k)}, \lambda^{(k)}) - \nabla K(u^{(k)}) \ , u \right\rangle \ ,$$

$$\lambda^{(k+1)} = \lambda^{(k)} + \rho \nabla_\lambda L_c(u^{(k+1)}, \lambda^{(k)}) \ ,$$

that is, APP allows to decompose augmented Lagrangians!

APP with explicit constraints: augmented Lagrangian III

Convergence theorem

H1 $\mathcal{U}^{\mathrm{ad}}$ is a closed convex subset of the Hilbert space $\mathcal{U}$, and C is a closed convex cone of the Hilbert space $\mathcal{V}$.

H2 J is a proper l.s.c convex function, and its derivative J' is Lipschitz with constant A.

H3 Θ is a C-convex, Lipschitz with constant τ, differentiable.

H4 A saddle point $(u^\sharp, \lambda^\sharp)$ exists.

H5 K is a proper l.s.c strongly convex function with modulus b, and its derivative K' is Lipschitz with constant B.

H6 ϵ and ρ are such that $0 < \epsilon < b/(A + c\tau^2)$ and $0 < \rho < 2c$.

R1 The sequence $\left\{(u^{(k)}, \lambda^{(k)})\right\}_{k\in\mathbb{N}}$ is bounded, and any of its cluster points is a saddle point.

References on decomposition/coordination methods

P. Carpentier & G. Cohen, "Décomposition-coordination en optimisation déterministe et stochastique". *Springer Berlin Heidelberg*, Mathématiques et Applications **81**, 2017.

G. Cohen, "Auxiliary Problem Principle and Decomposition of Optimization Problems". *Journal of Optimization Theory and Applications*, **32**, 1980.

G. Cohen & D.L. Zhu, "Decomposition coordination methods in large scale optimization problems. The nondifferentiable case and the use of augmented Lagrangians". In J.B. Cruz (Ed.): "*Advances in Large Scale Systems*", **1**, 203-266, JAI Press, Greenwich, Connecticut, 1984.

G. Cohen & B. Miara, "Optimization with an Auxiliary Constraint and Decomposition". *SIAM Journal on Control and Optimization*, **28**, 137-157, 1990.

Final remarks on decomposition methods I

The theory is available for general (infinite dimensional) Hilbert spaces, and thus applies in the stochastic framework, that is, the case where $\mathcal{U}$ is a space of random variables.

The minimization algorithm used for solving the subproblems is not specified in the decomposition process and is left to the user! It is however assumed that the user is able to solve the subproblem, for example in the price decomposition case:

$$\min_{u_i \in \mathcal{U}_i^{\mathrm{ad}}} J_i(u_i) + \left\langle \lambda^{(k)} , \Theta_i(u_i) \right\rangle ,$$

and to send the requested information, namely $\Theta_i(u_i^{(k+1)})$, to the coordination level.

Question: *what methods are suitable in the stochastic case?*

Final remarks on decomposition methods II

Whatever the decomposition/coordination scheme used (price, allocation, prediction, APP), new variables (depending on $u^{(k)}$ and/or $\lambda^{(k)}$) appear in the subproblems arising at iteration k of the optimization process.

Example: subproblem i in price decomposition:

$$\min_{u_i \in \mathcal{U}_i^{\mathrm{ad}}} J_i(u_i) + \left\langle \lambda^{(k)} , \Theta_i(u_i) \right\rangle .$$

All these new variables are considered as fixed when solving the subproblems (they only depend on the iteration index k). They are nothing but constants, and therefore do not cause any trouble in the deterministic case.

Question: *what happens in the stochastic case?*

Reminder of our ultimate goal

How to to obtain "good" strategies for a large scale stochastic optimal control problem, for example a problem corresponding to the optimal management over a given time horizon of a system involving a large amount of dynamical production units.

- In order to obtain decision strategies (closed-loop controls), we have to use Dynamic Programming or related methods.
  - **Assumption**: Markovian case,
  - **Difficulty**: curse of dimensionality.
- In order to to take into account the size of the system, we have to use decomposition/coordination techniques.
  - **Assumption**: convexity,
  - **Difficulty**: information pattern of the problem.

Stochastic optimal control problems

We consider a SOC problem (in the *Decision-Hazard* setting):

$$\min_{\boldsymbol{U},\boldsymbol{X}} \ \mathbb{E}\bigg(\sum_{i=1}^{N} \Big(\sum_{t=0}^{T-1} L_t^i(\boldsymbol{X}_t^i, \boldsymbol{U}_t^i, \boldsymbol{W}_{t+1}) + K^i(\boldsymbol{X}_T^i) \Big) \bigg) ,$$

subject to the constraints:

$$\boldsymbol{X}_0^i = f_{-1}^i(\boldsymbol{W}_0) , \qquad i = 1 \dots N ,$$

$$\boldsymbol{X}_{t+1}^i = f_t^i(\boldsymbol{X}_t^i, \boldsymbol{U}_t^i, \boldsymbol{W}_{t+1}) , \qquad t = 0 \dots T-1 , \ i = 1 \dots N ,$$

$$\boldsymbol{U}_t^i \preceq \sigma(\boldsymbol{W}_0, \dots, \boldsymbol{W}_t) , \qquad t = 0 \dots T-1 , \ i = 1 \dots N ,$$

$$\sum_{i=1}^{N} \Theta_t^i(\boldsymbol{X}_t^i, \boldsymbol{U}_t^i) = 0 , \qquad t = 0 \dots T-1 .$$

Stochastic optimal control problems

We consider a SOC problem (in the *Decision-Hazard* setting):

$$\min_{\boldsymbol{U},\boldsymbol{X}} \sum_{i=1}^{N} \left(\mathbb{E}\Big(\sum_{t=0}^{T-1} L_t^i(\boldsymbol{X}_t^i, \boldsymbol{U}_t^i, \boldsymbol{W}_{t+1}) + K^i(\boldsymbol{X}_T^i) \Big) \right) ,$$

subject to the constraints:

$$\boldsymbol{X}_0^i = f_{-1}^i(\boldsymbol{W}_0) , \qquad i = 1 \dots N ,$$

$$\boldsymbol{X}_{t+1}^i = f_t^i(\boldsymbol{X}_t^i, \boldsymbol{U}_t^i, \boldsymbol{W}_{t+1}) , \qquad t = 0 \dots T-1 , \; i = 1 \dots N ,$$

$$\boldsymbol{U}_t^i \preceq \sigma(\boldsymbol{W}_0, \dots, \boldsymbol{W}_t) , \qquad t = 0 \dots T-1 , \; i = 1 \dots N ,$$

$$\sum_{i=1}^{N} \Theta_t^i(\boldsymbol{X}_t^i, \boldsymbol{U}_t^i) = 0 , \qquad t = 0 \dots T-1 .$$

Dynamic Programming yields centralized controls

Remember that we want to solve this SOC problem using Dynamic Programming (DP) or related methods (such as SDDP).

The system is made of N interconnected subsystems, and we have denoted the control and the state of subsystem i at time t by $\boldsymbol{U}_t^i$ and $\boldsymbol{X}_t^i$. Recall that the optimal control of subsystem i when using DP is a function of the whole system state:

$$\boldsymbol{U}_t^i = \gamma_t^i \big(\boldsymbol{X}_t^1, \ldots, \boldsymbol{X}_t^N \big) ,$$

but a straightforward use of DP is prohibited for N large...

Moreover, decomposition should lead to decentralized feedbacks:

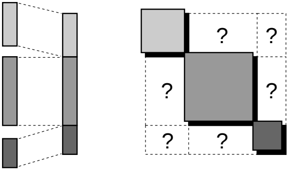
$$\boldsymbol{U}_t^i = \widehat{\gamma}_t^i (\boldsymbol{X}_t^i) ,$$

which are, in most cases, far from being optimal!

Straightforward decomposition of Dynamic Programming?

The crucial point is that the optimal feedback of a subsystem a priori depends on the state of all other subsystems, so that using a decomposition scheme by subsystems is far from being obvious...

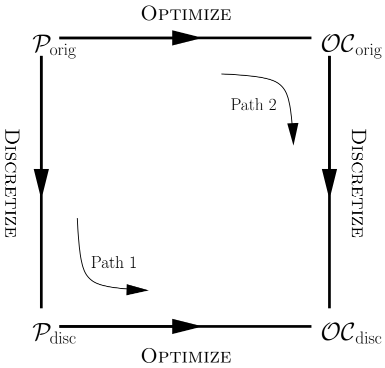
As far as we have to deal with Dynamic Programming, the central concern for decomposition/coordination purpose is resumed as:



- how to decompose a feedback γ_t w.r.t. its domain $\mathbb{X}_t$ rather than its range $\mathbb{U}_t$?

  And the answer is:
- impossible in the general case!

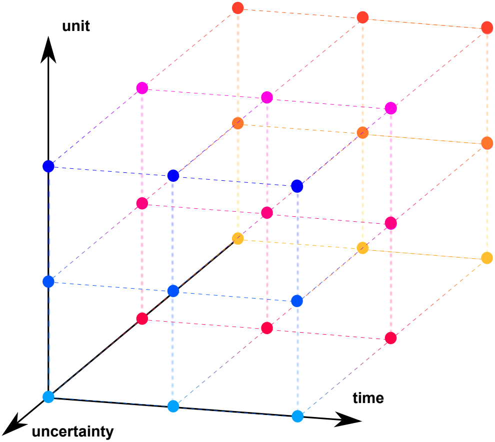
Remark on the approximation of a SOC problem



1. Following Path 1 (discretize, then optimize), we solve a deterministic approximation of the SOC problem.
   $\rightsquigarrow$ Scenario tree approximation.
   All the decomposition/coordination methods are available.
2. Following Path 2 (optimize, then discretize) we directly make use of a decomposition/coordination method on the SOC problem and then discretize the subproblems.
   $\rightsquigarrow$ Stochastic decomposition.

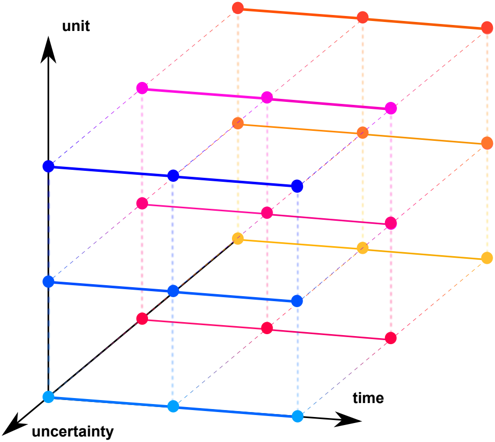
In this lecture, we are following path 2!

Couplings and decompositions for SOC problems I



$$\min \sum_{\omega} \sum_{i} \sum_{t} \pi_{\omega} L_t^i(\boldsymbol{X}_t^i, \boldsymbol{U}_t^i, \boldsymbol{W}_{t+1})$$

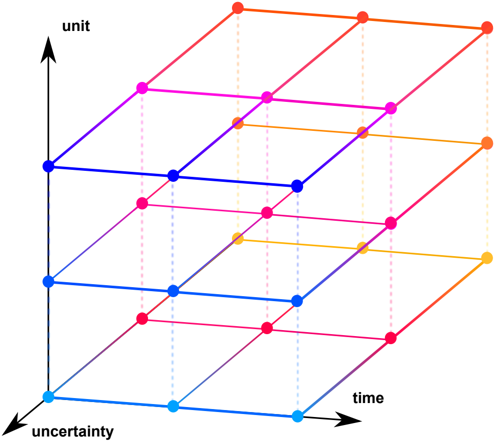
Couplings and decompositions for SOC problems II



$$\min \sum_{\omega} \sum_{i} \sum_{t} \pi_\omega L_t^i(\boldsymbol{X}_t^i, \boldsymbol{U}_t^i, \boldsymbol{W}_{t+1})$$

$$\text{s.t.} \quad \boldsymbol{X}_{t+1}^i = f_t^i(\boldsymbol{X}_t^i, \boldsymbol{U}_t^i, \boldsymbol{W}_{t+1})$$

Couplings and decompositions for SOC problems III

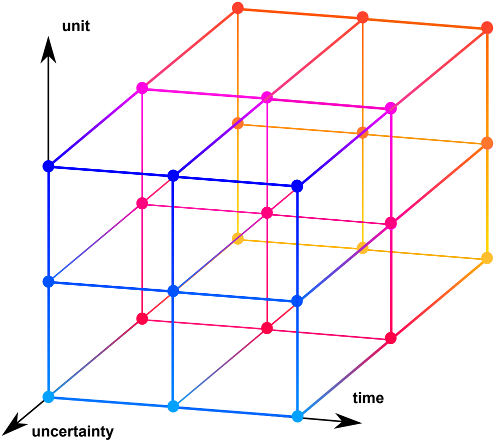


$$\min \sum_{\omega} \sum_{i} \sum_{t} \pi_\omega L_t^i(\boldsymbol{X}_t^i, \boldsymbol{U}_t^i, \boldsymbol{W}_{t+1})$$

$$\text{s.t. } \boldsymbol{X}_{t+1}^i = f_t^i(\boldsymbol{X}_t^i, \boldsymbol{U}_t^i, \boldsymbol{W}_{t+1})$$

$$\boldsymbol{U}_t^i \preceq \sigma(\boldsymbol{W}_0, \ldots, \boldsymbol{W}_t)$$

Couplings and decompositions for SOC problems IV



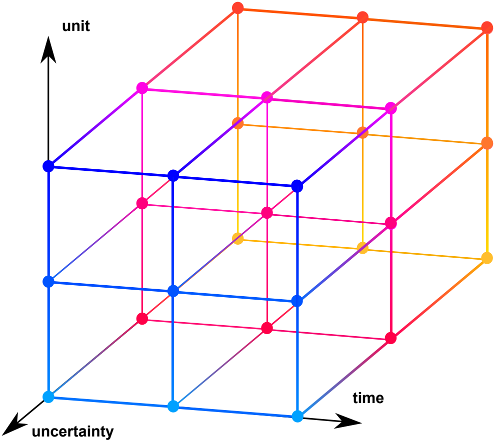
$$\min \sum_{\omega} \sum_{i} \sum_{t} \pi_\omega L_t^i(\boldsymbol{X}_t^i, \boldsymbol{U}_t^i, \boldsymbol{W}_{t+1})$$

$$\text{s.t.} \quad \boldsymbol{X}_{t+1}^i = f_t^i(\boldsymbol{X}_t^i, \boldsymbol{U}_t^i, \boldsymbol{W}_{t+1})$$

$$\boldsymbol{U}_t^i \preceq \sigma(\boldsymbol{W}_0, \ldots, \boldsymbol{W}_t)$$

$$\sum_{i} \Theta_t^i(\boldsymbol{X}_t^i, \boldsymbol{U}_t^i) = 0$$

Couplings and decompositions for SOC problems V



$$\min \sum_{\omega} \sum_{i} \sum_{t} \pi_\omega L_t^i(\boldsymbol{X}_t^i, \boldsymbol{U}_t^i, \boldsymbol{W}_{t+1})$$

s.t. $\boldsymbol{X}_{t+1}^i = f_t^i(\boldsymbol{X}_t^i, \boldsymbol{U}_t^i, \boldsymbol{W}_{t+1})$

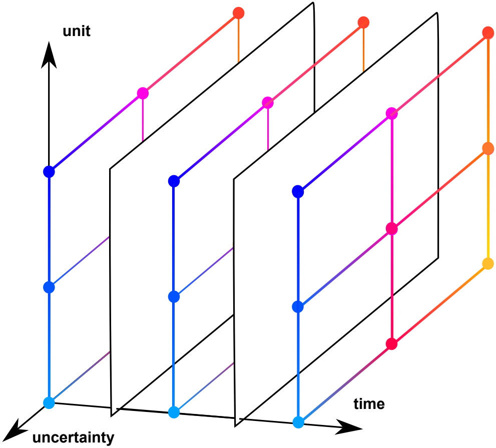
$$\boldsymbol{U}_t^i \preceq \sigma(\boldsymbol{W}_0, \ldots, \boldsymbol{W}_t)$$

$$\sum_i \Theta_t^i(\boldsymbol{X}_t^i, \boldsymbol{U}_t^i) = 0$$

3 additive structures!

Multiple decompositions...

Couplings and decompositions for SOC problems VI



$$\min \sum_{\omega} \sum_{i} \sum_{t} \pi_\omega L_t^i(\boldsymbol{X}_t^i, \boldsymbol{U}_t^i, \boldsymbol{W}_{t+1})$$

$$\text{s.t.} \quad \boldsymbol{X}_{t+1}^i = f_t^i(\boldsymbol{X}_t^i, \boldsymbol{U}_t^i, \boldsymbol{W}_{t+1})$$

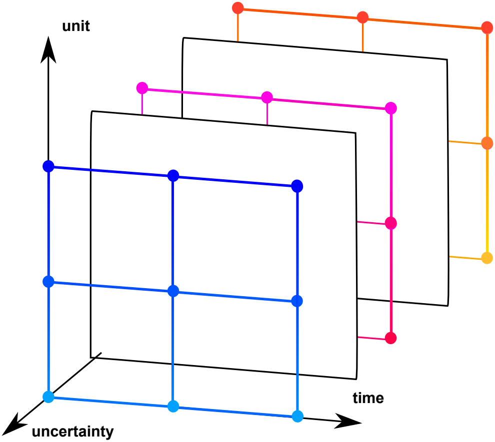
$$\boldsymbol{U}_t^i \preceq \sigma(\boldsymbol{W}_0, \ldots, \boldsymbol{W}_t)$$

$$\sum_i \Theta_t^i(\boldsymbol{X}_t^i, \boldsymbol{U}_t^i) = 0$$

Time decomposition

Dynamic Programming

Couplings and decompositions for SOC problems VII



$$\min \sum_{\omega} \sum_{i} \sum_{t} \pi_\omega L_t^i(\boldsymbol{X}_t^i, \boldsymbol{U}_t^i, \boldsymbol{W}_{t+1})$$

$$\text{s.t.} \quad \boldsymbol{X}_{t+1}^i = f_t^i(\boldsymbol{X}_t^i, \boldsymbol{U}_t^i, \boldsymbol{W}_{t+1})$$

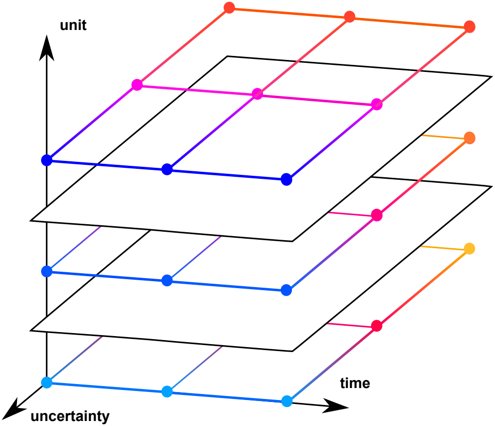
$$\boldsymbol{U}_t^i \preceq \sigma(\boldsymbol{W}_0, \ldots, \boldsymbol{W}_t)$$

$$\sum_i \Theta_t^i(\boldsymbol{X}_t^i, \boldsymbol{U}_t^i) = 0$$

Scenario decomposition

Progressive Hedging

Couplings and decompositions for SOC problems VIII



$$\min \sum_{\omega} \sum_{i} \sum_{t} \pi_\omega L_t^i(\boldsymbol{X}_t^i, \boldsymbol{U}_t^i, \boldsymbol{W}_{t+1})$$

$$\text{s.t.} \quad \boldsymbol{X}_{t+1}^i = f_t^i(\boldsymbol{X}_t^i, \boldsymbol{U}_t^i, \boldsymbol{W}_{t+1})$$

$$\boldsymbol{U}_t^i \preceq \sigma(\boldsymbol{W}_0, \ldots, \boldsymbol{W}_t)$$

$$\sum_{i} \Theta_t^i(\boldsymbol{X}_t^i, \boldsymbol{U}_t^i) = 0$$

Spatial decomposition

Purpose of the lecture

Price decomposition in the stochastic case I

Dualize the spatial coupling constraints in the SOC problem:

$$\min_{\boldsymbol{U},\boldsymbol{X}} \sum_{i=1}^{N} \left(\mathbb{E}\left(\sum_{t=0}^{T-1} L_t^i(\boldsymbol{X}_t^i, \boldsymbol{U}_t^i, \boldsymbol{W}_{t+1}) + K^i(\boldsymbol{X}_T^i) \right) \right) ,$$

subject to the constraints:

$$\boldsymbol{X}_0^i = f_{-1}^i(\boldsymbol{W}_0) , \qquad i = 1 \dots N ,$$

$$\boldsymbol{X}_{t+1}^i = f_t^i(\boldsymbol{X}_t^i, \boldsymbol{U}_t^i, \boldsymbol{W}_{t+1}) , \qquad t = 0 \dots T-1 , \quad i = 1 \dots N ,$$

$$\boldsymbol{U}_t^i \preceq \sigma(\boldsymbol{W}_0, \dots, \boldsymbol{W}_t) , \qquad t = 0 \dots T-1 , \quad i = 1 \dots N ,$$

$$\sum_{i=1}^{N} \Theta_t^i(\boldsymbol{X}_t^i, \boldsymbol{U}_t^i) = 0 , \qquad t = 0 \dots T-1 \quad \rightsquigarrow \quad \boldsymbol{\Lambda}_t .$$

Price decomposition in the stochastic case II

Applying price decomposition to the previous SOC problem leads to a collection of local stochastic optimal control subproblems indexed by $i \in [\![1, N]\!]$:

$$\min_{\boldsymbol{U}^i, \boldsymbol{X}^i} \mathbb{E}\bigg(\sum_{t=0}^{T-1} \big(L_t^i(\boldsymbol{X}_t^i, \boldsymbol{U}_t^i, \boldsymbol{W}_{t+1}) + \boldsymbol{\Lambda}_t^{(k)} \cdot \Theta_t^i(\boldsymbol{X}_t^i, \boldsymbol{U}_t^i) \big) + K^i(\boldsymbol{X}_T^i) \bigg) ,$$

subject to the constraints:

$$\begin{aligned} \boldsymbol{X}_0^i &= f_{-1}^i(\boldsymbol{W}_0) , \\ \boldsymbol{X}_{t+1}^i &= f_t^i(\boldsymbol{X}_t^i, \boldsymbol{U}_t^i, \boldsymbol{W}_{t+1}) , \qquad t = 0 \dots T-1 , \end{aligned}$$

$$\boldsymbol{U}_t^i \preceq \sigma(\boldsymbol{W}_0, \dots, \boldsymbol{W}_t) , \qquad t = 0 \dots T-1 .$$

Price decomposition in the stochastic case III

- As pointed out in the deterministic case, new variables, that is, dual multipliers $\boldsymbol{\Lambda}_t^{(k)}$, appear in the subproblems arising at iteration k: these variables, fixed at this stage of calculation, corresponds to random variables.

$$\min_{\boldsymbol{U}^i,\boldsymbol{X}^i} \mathbb{E}\left(\sum_t L_t^i(\boldsymbol{X}_t^i, \boldsymbol{U}_t^i, \boldsymbol{W}_{t+1}) + \boldsymbol{\Lambda}_t^{(k)} \cdot \Theta_t^i(\boldsymbol{X}_t^i, \boldsymbol{U}_t^i) \right) .$$

- The process $\boldsymbol{\Lambda}^{(k)}$ acts as an additional input (data) in the subproblems, but the structure of this process is a priori unknown: it may be correlated in time, so that the white noise assumption, crucial for the optimality of Dynamic Programming, has no reason to be fulfilled in that context!

Summary

- On the one hand, it seems that Dynamic Programming cannot be decomposed in a straightforward manner.
- On the other hand, applying a decomposition scheme to a SOC problem introduces coordination instruments in the subproblems, e.g. the multipliers $\boldsymbol{\Lambda}_t^{(k)}$ in the case of price decomposition, which correspond to additional fixed random variables whose time structure is unknown.

Question: how to handle the coordination instruments (random variables $\boldsymbol{\Lambda}_t^{(k)}$ in the case of price decomposition) in order to obtain an approximation of the overall optimum of the SOC problem?

BREAK