Combining decomposition and dynamic programming

IFORS'08

Kengy Barty[1], Pierre Girardeau[1,2,3], Peio Lahirigoyen[2]

[1]EDF R&D, France

[2]École Nationale Supérieure de Techniques Avancées, France

[3]École Nationale des Ponts et Chaussées, France

July 18, 2008

Outline

Framework (1)

- EDF is the main power producer in Europe.
- It owns a number of **power units** that have **various characteristics** (coal, nuclear, hydraulic, etc.) with which it has **to supply a power demand** at each time, while **minimizing the production cost**.

There are several difficulties:

- Large scale (number of units and of time periods);
- Stochasticity (demand and water inflows).

Framework (2)

Suppose we have N power units, either thermal or hydraulic.

- **Thermal units** do not have stock constraints, but they have a production cost L_t^i for time step t and unit i.
- On the contrary, **hydraulic plants** do not have production costs but stock constraints.

At each time step, the sum of the productions over the thermal and hydraulic plants must **equal the power demand**.

Notations

In the following, we denote random variables by **bold** letters. Let us note:

- $I_{hy} \cup I_{th} = \{1, \ldots, N\}$ the set of power units, either thermal or hydraulic;
- $\mathbf{X}_t^i$ the stock height at time step t for power unit i (only for hydraulic units);
- $\mathbf{U}_t^i$ the power production at time step t for unit i;
- $\mathbf{D}_t$ the power demand at time step t;
- $\boldsymbol{\xi}_t^i$ the inflows at time step t for unit i (only for hydraulic units);
- $\mathcal{F}_t = \sigma\left\{\mathbf{D}_1, \boldsymbol{\xi}_1, \ldots, \mathbf{D}_t, \boldsymbol{\xi}_t\right\}$ denotes the information available at time t.

Mathematical formulation

Our mid-term power management problem can be stated as follows:

$$
\begin{cases}
\min\limits_{\mathbf{U}} & \mathbb{E}\left(\sum\limits_{t=1}^{T-1}\sum\limits_{i\in I_{th}} L_t^i\left(\mathbf{U}_t^i\right) + \sum\limits_{i\in I_{hy}} V^i\left(\mathbf{X}_T^i\right)\right) \\
\text{s.t.} & \mathbf{X}_{t+1}^i = \mathbf{X}_t^i - \mathbf{U}_t^i + \boldsymbol{\xi}_t^i, \qquad \boxed{\text{independant dynamics}} \\
& \sum\limits_{i\in I_{th}\cup I_{hy}} \mathbf{U}_t^i = \mathbf{D}_t, \qquad \boxed{\text{demand constraint}} \\
& \left(\mathbf{U}_t^i\right)_{i\in I_{hy}} \text{ is } \mathcal{F}_{t-1}\text{-measurable}, \qquad \boxed{\text{measurability constraints}} \\
& \left(\mathbf{U}_t^i\right)_{i\in I_{th}} \text{ is } \mathcal{F}_t\text{-measurable}, \\
& + \text{ bound constraints on } \mathbf{X} \text{ and } \mathbf{U}.
\end{cases}
$$

Scenario trees

- When dealing with large-scale stochastic optimal control problems, a usual approach consists in **"discretising randomness"** by using a scenario tree [Higle and Sen, 1996].
- One can then use some **deterministic techniques to decompose** the problem into small-scale subproblems on each power unit.

Problem: **feedback synthesis** and randomness representation.

Dynamic programming

An other approach consists in introducing the value function V and to solve the **dynamic programming equation** [Bellman, 1957, Bertsekas, 2000]:

$$\begin{cases} V_T(x) &= \sum_{i \in I_{hy}} V^i\left(x^i\right), \\ V_t(x) &= \min_u \mathbb{E}\left(\sum_{i \in I_{th}} L_t^i\left(u^i\right) + V_{t+1}\left(\mathbf{X}_{t+1}\right) \middle| \mathbf{X}_t = x\right). \end{cases}$$

Problem: complexity is exponential with respect to the state space dimension (**curse of dimensionality**).

Duality (1)

One can always write:

$$\min_{\mathbf{U}} \max_{\boldsymbol{\lambda}} \quad \mathbb{E}\left(\sum_{t=0}^{T-1}\left(\sum_{i\in I_{th}} L_t^i\left(\mathbf{U}_t^i\right) + \boldsymbol{\lambda}_t\left(\mathbf{D}_t - \sum_{i=1}^{N}\mathbf{U}_t^i\right)\right) + \sum_{i\in I_{hy}} V^i\left(\mathbf{X}_T^i\right)\right),$$

$$\text{s.t.} \quad \mathbf{X}_{t+1}^i = \mathbf{X}_t^i - \mathbf{U}_t^i + \boldsymbol{\xi}_t^i, \quad \mathbb{P}\text{-p.s}, \forall t = 0,\ldots,T-1, \forall i \in I_{hy},$$

$$\mathbf{U}_t^i \text{ is } \mathcal{F}_{t-1}\text{-measurable}, \quad \forall i \in I_{hy},$$

$$\mathbf{U}_t^i \text{ is } \mathcal{F}_t\text{-measurable}, \quad \forall i \in I_{th}.$$

Duality (2)

When there is no duality gap:

$$\max_{\boldsymbol{\lambda}} \min_{\mathbf{U}} \quad \mathbb{E}\left(\sum_{t=0}^{T-1}\left(\sum_{i\in I_{th}} L_t^i\left(\mathbf{U}_t^i\right) + \boldsymbol{\lambda}_t\left(\mathbf{D}_t - \sum_{i=1}^{N}\mathbf{U}_t^i\right)\right) + \sum_{i\in I_{hy}} V^i\left(\mathbf{X}_T^i\right)\right),$$

$$\text{s.t.} \quad \mathbf{X}_{t+1}^i = \mathbf{X}_t^i - \mathbf{U}_t^i + \boldsymbol{\xi}_t^i, \quad \mathbb{P}\text{-p.s}, \forall t = 0, \dots, T-1, \forall i \in I_{hy},$$

$$\mathbf{U}_t^i \text{ is } \mathcal{F}_{t-1}\text{-measurable}, \quad \forall i \in I_{hy},$$

$$\mathbf{U}_t^i \text{ is } \mathcal{F}_t\text{-measurable}, \quad \forall i \in I_{th}.$$

Duality (3)

The problem is now decomposable:

$$\max_{\boldsymbol{\lambda}} \left[\sum_{i \in I_{th}} \min_{\mathbf{U}^i} \mathbb{E} \left(\sum_{t=0}^{T-1} \sum_{i \in I_{th}} \left(L_t^i \left(\mathbf{U}_t^i \right) - \boldsymbol{\lambda}_t \mathbf{U}_t^i \right) \right) + \sum_{i \in I_{hy}} \min_{\mathbf{U}^i} \mathbb{E} \left(\sum_{t=1}^{T-1} -\boldsymbol{\lambda}_t^i \mathbf{U}_t^i + V^i \left(\mathbf{X}_T^i \right) \right) + \mathbb{E} \left(\sum_{t=1}^{T-1} \boldsymbol{\lambda}_t \mathbf{D}_t \right) \right],$$

$$\text{s.t.} \quad \mathbf{X}_{t+1}^i = \mathbf{X}_t^i - \mathbf{U}_t^i + \boldsymbol{\xi}_t^i, \quad \mathbb{P}\text{-p.s}, \forall t = 0, \dots, T-1, \forall i \in I_{hy},$$

$$\mathbf{U}_t^i \text{ is } \mathcal{F}_{t-1}\text{-measurable}, \quad \forall i \in I_{hy},$$

$$\mathbf{U}_t^i \text{ is } \mathcal{F}_t\text{-measurable}, \quad \forall i \in I_{th}.$$

Subproblems (1)

The subproblem on the *i*-th thermal plant reads:

$$\min_{\mathbf{U}^i} \quad \mathbb{E}\left(\sum_{t=0}^{T-1}\left(L_t^i\left(\mathbf{U}_t^i\right) - \boldsymbol{\lambda}_t \mathbf{U}_t^i\right)\right),$$

$$\text{s.t.} \quad \mathbf{U}_t^i \text{ is } \mathcal{F}_t\text{-measurable.}$$

Subproblems (2)

The subproblem on the i-th hydraulic plant reads:

$$\min_{\mathbf{U}^i} \quad \mathbb{E}\left(\sum_{t=0}^{T-1}\left(-\boldsymbol{\lambda}_t \mathbf{U}_t^i\right) + V^i\left(\mathbf{X}_T^i\right)\right),$$

$$\text{s.t.} \quad \mathbf{X}_{t+1}^i = \mathbf{X}_t^i - \mathbf{U}_t^i + \boldsymbol{\xi}_t^i, \quad \mathbb{P}\text{-p.s}, \forall t = 0, \ldots, T-1,$$

$$\mathbf{U}_t^i \text{ is } \mathcal{F}_{t-1}\text{-measurable.}$$

Updating the Lagrange multipliers

We propose a decomposition and coordination approach that follows the work of [Cohen, 1978].

- For each subproblem, we obtain an optimal policy $\hat{\mathbf{U}}^i$ that depends on the price $\boldsymbol{\lambda}$.
- Then we update $\boldsymbol{\lambda}$ using a gradient step.
- Economically, when the production is low, one has to increase the price.

Prices update (Uzawa)

$$\boldsymbol{\lambda}_t^{k+1} = \boldsymbol{\lambda}_t^k + \gamma \left(\mathbf{D}_t - \sum_{i=1}^{N} \hat{\mathbf{U}}_t^i \right) .$$

Prices dynamics

In order to solve subproblems using dynamic programming, we need **dynamics for the prices $\boldsymbol{\lambda}$**. Our idea consists in deciding a prior the shape of these dynamics:

Dynamics we choose for λ

$$\boldsymbol{\lambda}_{t+1} = \alpha_t \boldsymbol{\lambda}_t + \beta_t \mathbf{D}_{t+1} + \gamma_t .$$

We will iterate on the parameters of these dynamics.

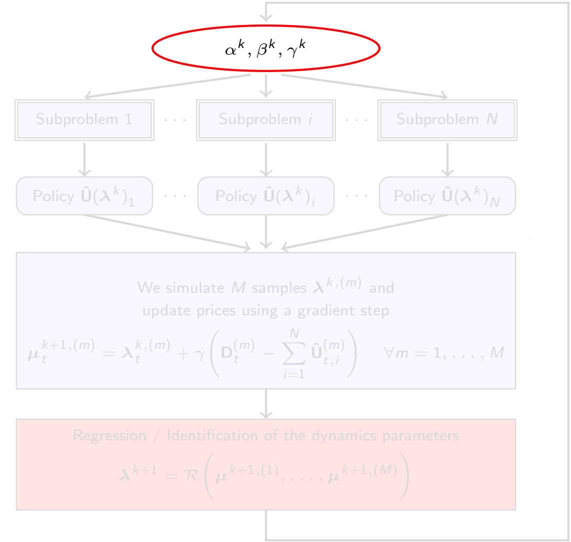
The algorithm

Parameters at iteration k

We solve subproblems using dynamic programming

We obtain policies

We update prices dynamics in order to satisfy the demand



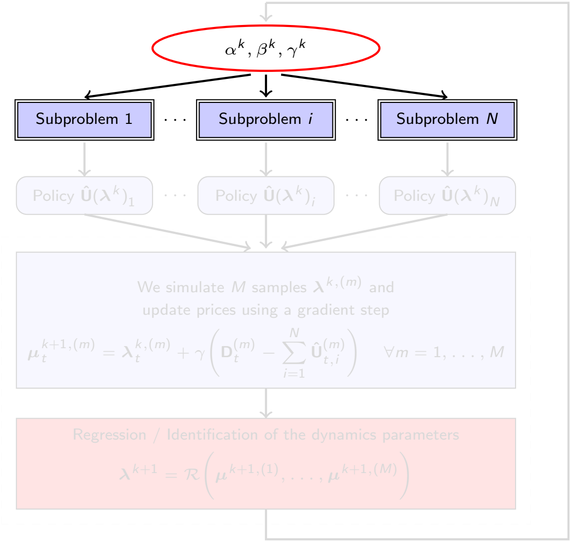
The algorithm

Parameters at iteration k

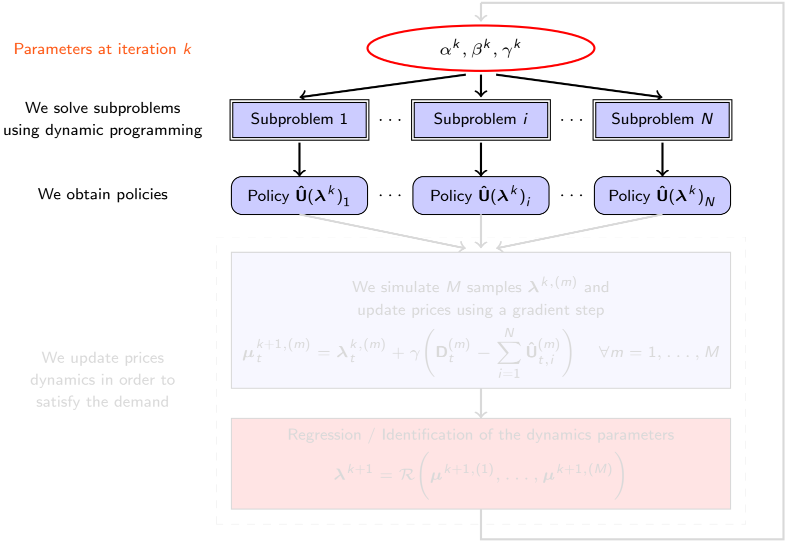
We solve subproblems using dynamic programming

We obtain policies

We update prices dynamics in order to satisfy the demand



The algorithm



The algorithm

Parameters at iteration k

$\alpha^k, \beta^k, \gamma^k$

We solve subproblems using dynamic programming

Subproblem 1 · · · Subproblem i · · · Subproblem N

We obtain policies

Policy $\hat{\mathbf{U}}(\boldsymbol{\lambda}^k)_1$ · · · Policy $\hat{\mathbf{U}}(\boldsymbol{\lambda}^k)_i$ · · · Policy $\hat{\mathbf{U}}(\boldsymbol{\lambda}^k)_N$

We update prices dynamics in order to satisfy the demand

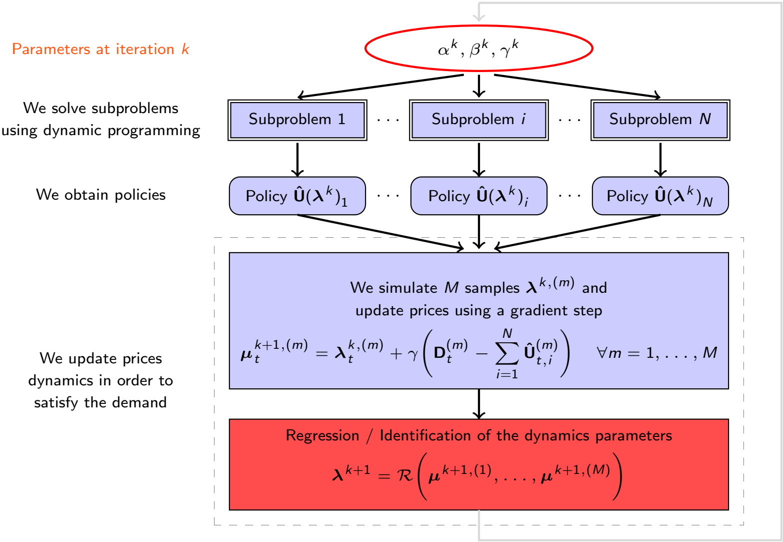
We simulate M samples $\boldsymbol{\lambda}^{k,(m)}$ and update prices using a gradient step

$$\boldsymbol{\mu}_t^{k+1,(m)} = \boldsymbol{\lambda}_t^{k,(m)} + \gamma \left(\mathbf{D}_t^{(m)} - \sum_{i=1}^{N} \hat{\mathbf{U}}_{t,i}^{(m)} \right) \qquad \forall m = 1, \ldots, M$$

Regression / Identification of the dynamics parameters

$$\boldsymbol{\lambda}^{k+1} = \mathcal{R}\left(\boldsymbol{\mu}^{k+1,(1)}, \ldots, \boldsymbol{\mu}^{k+1,(M)} \right)$$

The algorithm



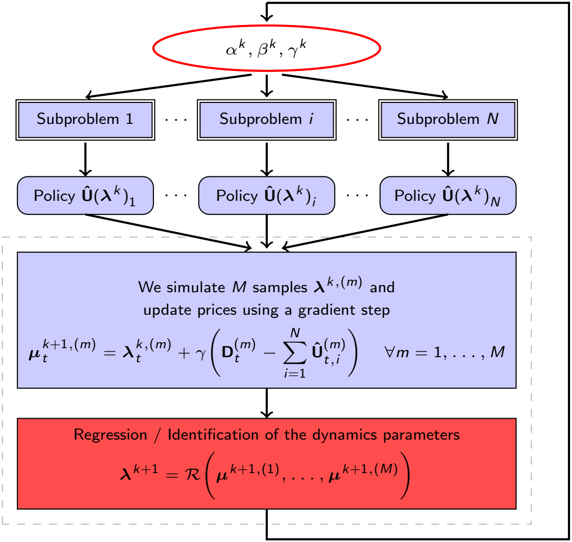
The algorithm

At iteration $k+1$

We solve subproblems using dynamic programming

We obtain policies

We update prices dynamics in order to satisfy the demand



The test case

We have applied this technique on a **two-stocks problem**, so that we can compute the optimal policy using dynamic programming.

Thermal power units

- We have several thermal power units that we model by a **quadratic production cost** in order to have strict convexity for the objective function.

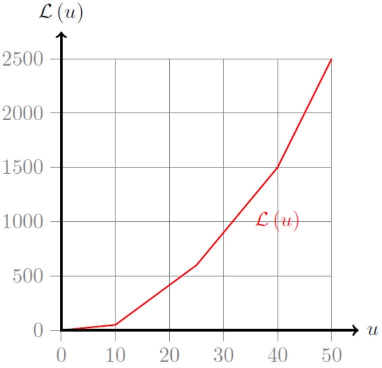


Fig.: Left: the real cost. Right: our approximation.

Hydraulic plants

- Two hydraulic plants:
  - $[\underline{x_1}, \overline{x_1}] = [0, 50]$
  - $[\underline{x_2}, \overline{x_2}] = [0, 40]$
  - $[\underline{v_1}, \overline{v_1}] = [0, 6]$
  - $[\underline{v_2}, \overline{v_2}] = [0, 6]$
- $V^1(x)$ and $V^2(x)$:

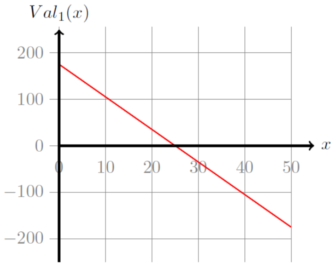


Fig.: End-period value functions

Random variables (1)

- We consider only white noises.

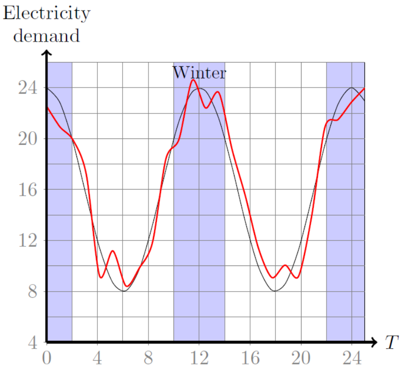


Fig.: A drawing of the demand.

Random variables (2)

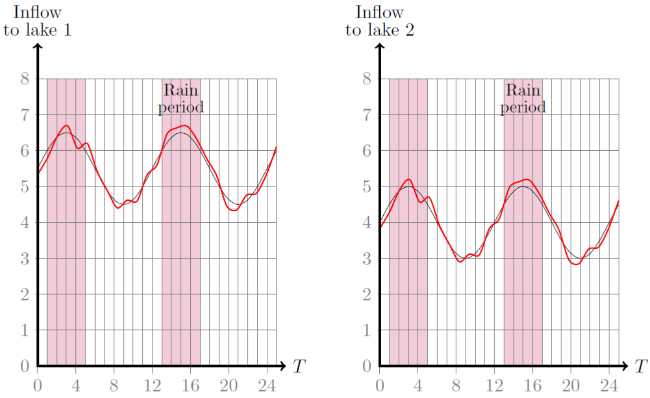
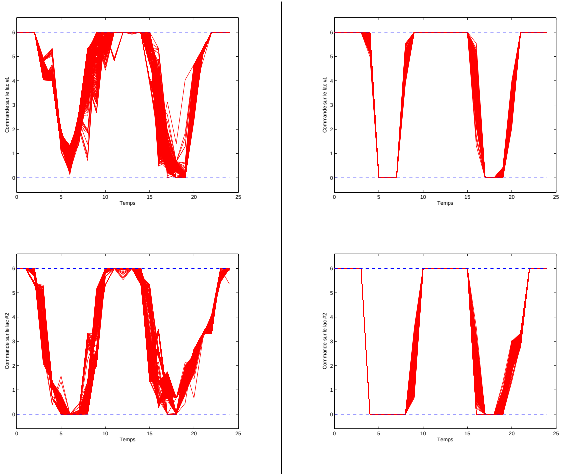


Fig.: A drawing of the inflows.

Results

- We compute policies by:
  - dynamic programming (reference),
  - by decomposition (250 iterations)

Commands



Tab.: 100 command samples for stock n˚1 (up) and n˚2 (down) using dynamic programming (left) and decomposition (right).

Costs

	Average cost	Confidence interval
Dynamic programming (optimum)	373.2	+/- 7.8
Decomposition	506,3	+/- 8.4
Dumb method (random commands)	1879,3	+/- 23.3

Cost along with iterations

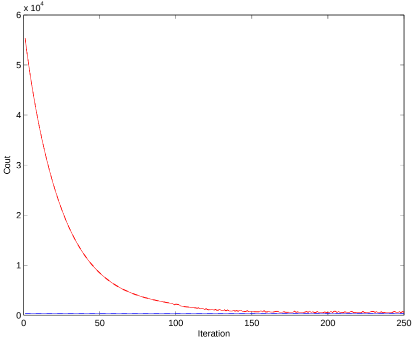
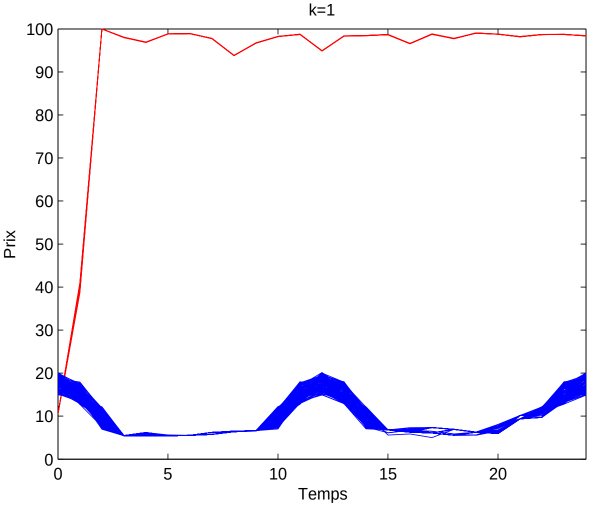
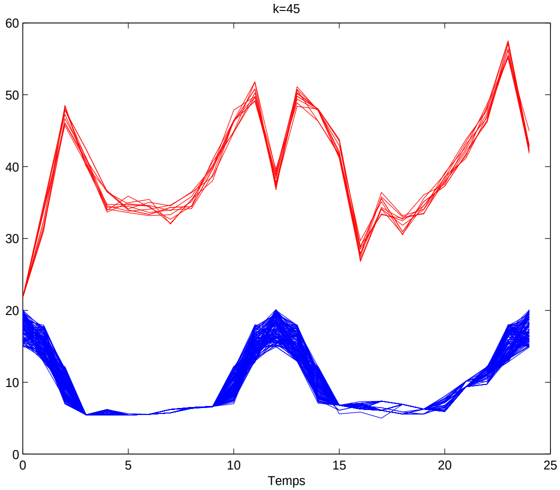


Fig.: We observe how average cost decreases along with the iterations in the decomposition method.

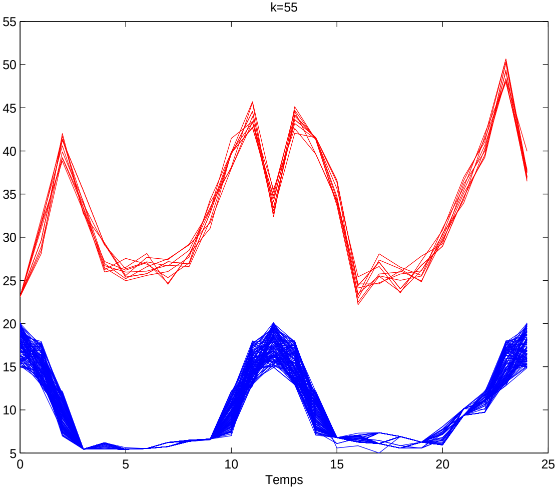
How prices converge to the optimal price



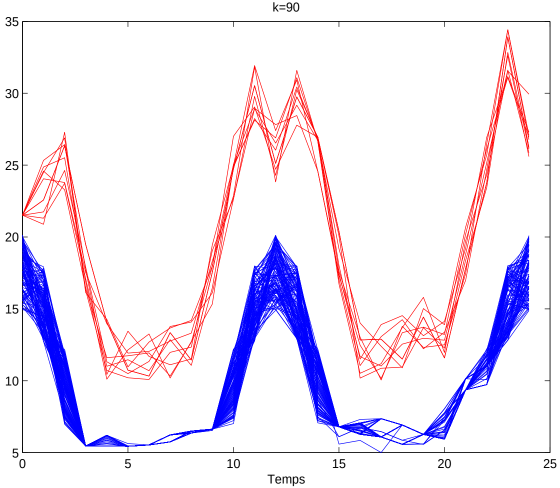
How prices converge to the optimal price



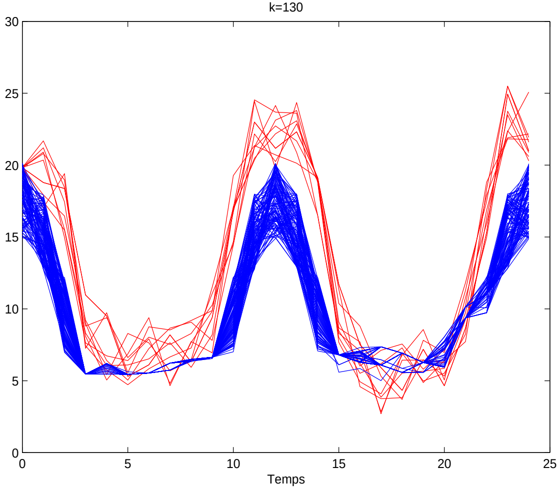
How prices converge to the optimal price



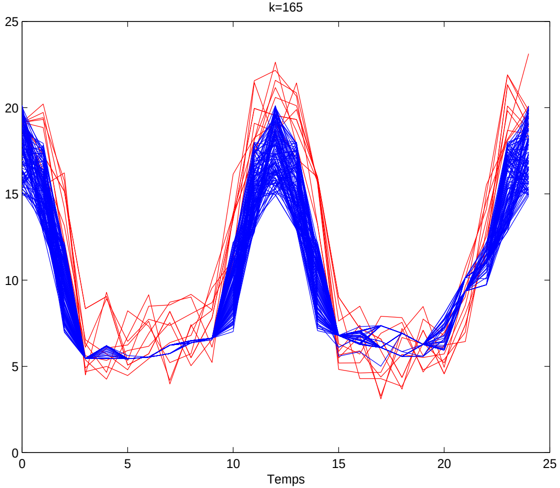
How prices converge to the optimal price



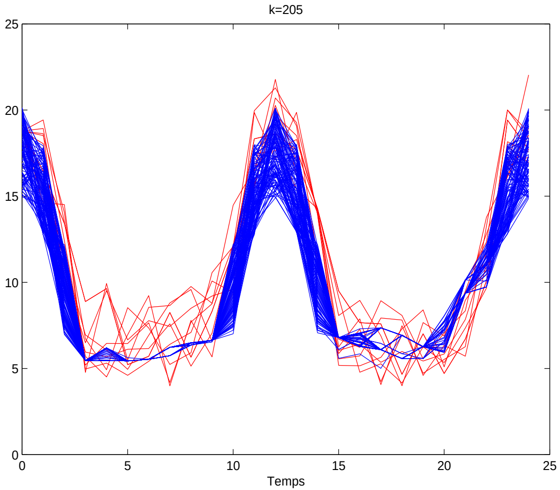
How prices converge to the optimal price



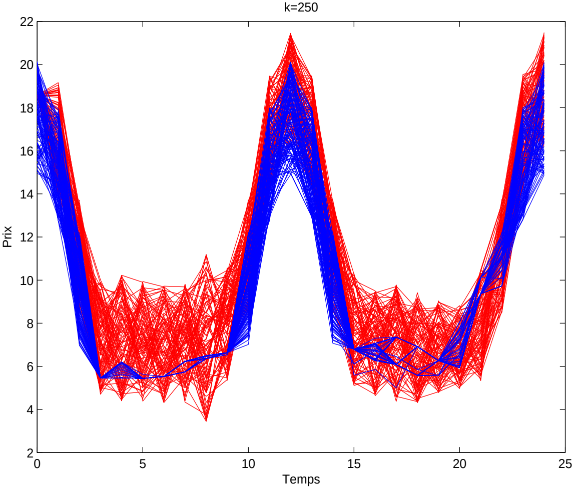
How prices converge to the optimal price



How prices converge to the optimal price



How prices converge to the optimal price



Conclusion (1)

We developped a decomposition method for large-scale stochastic optimal control problems where:

- units have independant dynamics;
- they are coupled by a static contraint, for example a demand satisfaction constraint.

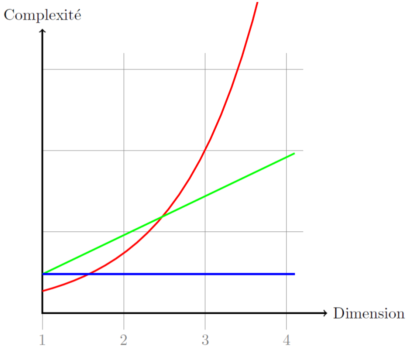
We obtain good results:

- on a two stocks and on a three stocks problem;
- on more complicated problems where the state appears in the coupling constraint.

Conclusion (2)

With this method, we are able to pass through the curse of dimensionality:

- dynamic programming
- decomposition without parallel computation
- decomposition with parallel computation



Future works

- Theoretical justification;
- How to identify the good shape for the dynamics?
- Higher dimension.

Bibliography

Bellman, R. (1957).
Dynamic Programming.
Princeton University Press, New Jersey.

Bertsekas, D. P. (2000).
Dynamic Programming and Optimal Control, volume 1 & 2.
Athena Scientific, 2 edition.

Cohen, G. (1978).
Optimization by decomposition and coordination : a unified approach.
IEEE Trans. Autom. Control, 23(2):222–232.

Higle, J. and Sen, S. (1996).
Stochastic Decomposition.
Kluwer, Dordrecht.