Nodal decomposition of stochastic Bellman functions

Application to the decentralized management of urban microgrids

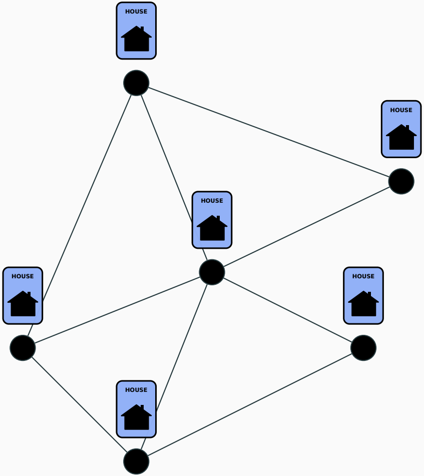
P. Carpentier — J.P. Chancelier — M. De Lara — F. Pacaud

SESO 2018

ENSTA ParisTech — ENPC ParisTech — Efficacity

Motivation

We consider a *peer-to-peer* community,
where different buildings exchange energy



- Decision centers at nodes
- Power flows through edges
- Multistage decisions
- Large-scale problem

Lecture outline

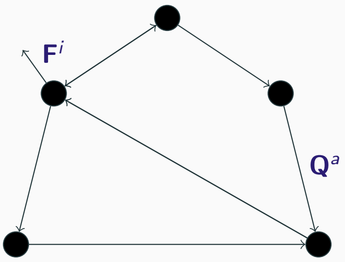
Problem statement

Price and resource decomposition algorithms

Application to the management of microgrids

Problem statement

Modeling energy exchanges between nodes



Grid is represented
by a graph $G = (\mathcal{N}, \mathcal{A})$

Let $T \in \mathbb{N}^\star$ be a horizon and

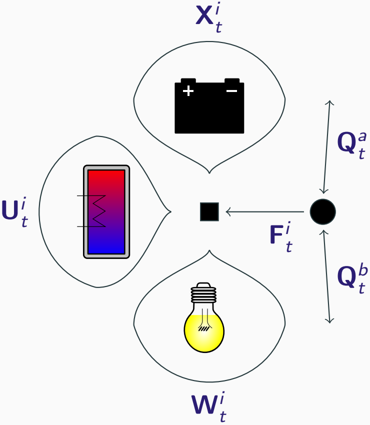
- $\mathbf{Q}_t^a$ energy exchanged through arc a,
- $\mathbf{F}_t^i$ energy imported at node i

At each time $t \in [\![0, T-1]\!]$ we consider a coupling between the nodal subproblems

$$\mathbf{F}_t^i = \sum_{a \in input(i)} \mathbf{Q}_t^a - \sum_{b \in output(i)} \mathbf{Q}_t^b$$

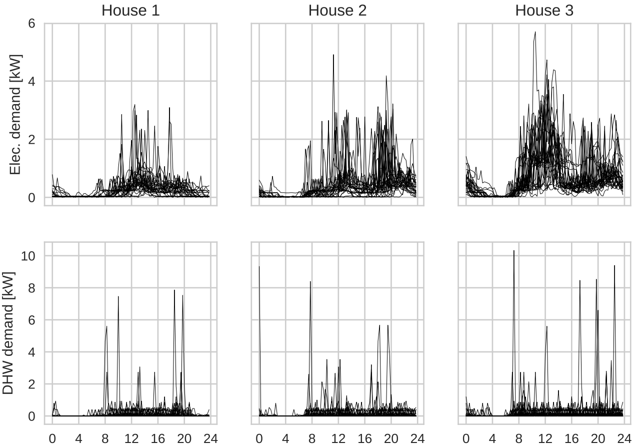
Each node manages its own devices

At each node i of the grid, at each time t, we have



- $\mathbf{X}_t^i \in \mathbb{X}_t^i$: state variable (battery, hot water tank)
- $\mathbf{U}_t^i \in \mathbb{U}_t^i$: control variable (energy production)
- $\mathbf{W}_t^i \in \mathbb{W}_t^i$: noise (consumption, renewable)

Electrical and thermal demands are uncertain



These scenarios are generated with `StRoBE`, a generator open-sourced by KU-Leuven

Writing down the nodal production problem

We aim at minimizing the operational costs over the nodes $i \in [\![1, N]\!]$

$$J_P^i(\mathbf{F}^i) = \min_{\mathbf{X}^i, \mathbf{U}^i} \mathbb{E}\Big[\sum_{t=0}^{T-1} \underbrace{L_t^i(\mathbf{X}_t^i, \mathbf{U}_t^i, \mathbf{W}_{t+1}^i)}_{\text{operational cost}} + K^i(\mathbf{X}_T^i) \Big]$$

subject to, for all $t \in [\![0, T-1]\!]$

i) The nodal dynamics constraint (for battery and hot water tank)

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

ii) The non-anticipativity constraint (future remains unknown)

$$\sigma(\mathbf{U}_t^i) \subset \sigma(\mathbf{W}_0^i, \cdots, \mathbf{W}_t^i)$$

iii) The load balance equation (production + import = demand)

$$\Delta_t^i(\mathbf{X}_t^i, \mathbf{U}_t^i, \mathbf{F}_t^i, \mathbf{W}_{t+1}^i) = 0$$

Transportation costs are decoupled in time

At each time step $t \in [\![0, T-1]\!]$, we define the transport cost as the sum of the costs of flows $\mathbf{Q}_t^a$ through the arcs a of the grid

$$J_{T,t}(\mathbf{Q}_t) = \mathbb{E}\Big(\sum_{a \in \mathcal{A}} l_t^a(\mathbf{Q}_t^a)\Big)$$

where the l_t^a's are easy to compute functions (say quadratic)

Kirchhoff's law

The balance equation stating the conservation between $\mathbf{Q}_t$ and $\mathbf{F}_t$ rewrites in a compact manner

$$A\mathbf{Q}_t + \mathbf{F}_t = 0$$

where A is the node-arc incidence matrix of the grid.

The overall production transport problem

The *production cost* J_P aggregates the costs at all nodes i

$$J_P(\mathbf{F}) = \sum_{i \in \mathcal{N}} J_P^i(\mathbf{F}^i)$$

and the *transport cost* J_T aggregates the edges costs at all time t

$$J_T(\mathbf{Q}) = \sum_{t=0}^{T-1} J_{T,t}(\mathbf{Q}_t)$$

The compact production transport problem formulation writes

$$V^\sharp = \min_{\mathbf{F},\mathbf{Q}} J_P(\mathbf{F}) + J_T(\mathbf{Q})$$
$$\text{s.t. } A\mathbf{Q} + \mathbf{F} = 0$$

What do we plan to do?

- We have formulated a stochastic optimization problem
- We will handle the coupling constraints by two methods:
  - Price decomposition
  - Resource decomposition
- We will show the scalability of decomposition algorithms!
  (We solve problem gathering up to 48 buildings)

Assumption

$J_P(\cdot)$ and $J_T(\cdot)$ are differentiables and strongly-convex w.r.t. $\mathbf{F}$ and $\mathbf{Q}$

Price and resource decomposition algorithms

Of decomposition and class struggle

Price decomposition formulates as a capitalistic world

Three levels of hierarchy

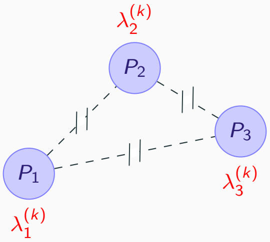
1. The *boss* fixes the price λ so as to optimize global cost
2. The *nodal managers* manage buildings to decrease local costs
3. The *workers* compute locally nodal value functions for each building

The boss basically just listens to the global oracle

- The boss aims to find the optimal deterministic price $\boldsymbol{\lambda}$

$$\max_{\boldsymbol{\lambda}} \underline{V}(\boldsymbol{\lambda}) := \min_{\mathbf{F},\mathbf{Q}} J_P(\mathbf{F}) + J_T(\mathbf{Q}) + \langle \boldsymbol{\lambda} , A\mathbf{Q} + \mathbf{F} \rangle$$

- Let $\boldsymbol{\lambda}^{(k)}$ be a given price
  The boss decomposes the global function $\underline{V}(\boldsymbol{\lambda}^{(k)})$ w.r.t. nodes and arcs



$$\min_{\mathbf{F}} J_P(\mathbf{F}) + \langle \boldsymbol{\lambda}^{(k)} , \mathbf{F} \rangle = \min_{\mathbf{F}^1,\cdots,\mathbf{F}^N} \sum_{i=1}^{N} J_P^i(\mathbf{F}^i) + \langle \boldsymbol{\lambda}^i , \mathbf{F}^i \rangle$$
$$= \sum_{i=1}^{N} \min_{\mathbf{F}^i} \left\{ J_P^i(\mathbf{F}^i) + \langle \boldsymbol{\lambda}^i , \mathbf{F}^i \rangle \right\}$$

- Once subproblems solved by each *nodal managers*,
  she updates the price with the oracle $\nabla \underline{V}(\boldsymbol{\lambda}^{(k)})$

$$\boldsymbol{\lambda}^{(k+1)} = \boldsymbol{\lambda}^{(k)} + \rho \nabla \underline{V}(\boldsymbol{\lambda}^{(k)})$$

Managing buildings in each node

At each building $i \in [\![1, N]\!]$, the nodal manager

- Receives a deterministic price $\boldsymbol{\lambda}^i$ from the boss and build the nodal problem

$$\underline{V}^i(\boldsymbol{\lambda}^i) = \min_{\mathbf{F}^i} J_P^i(\mathbf{F}^i) + \left\langle \boldsymbol{\lambda}^i \ , \mathbf{F}^i \right\rangle$$

  which rewrites as a Stochastic Optimal Control problem

$$\begin{aligned} \underline{V}^i(\boldsymbol{\lambda}^i) = \min_{\mathbf{X}^i, \mathbf{U}^i, \mathbf{F}^i} \ & \mathbb{E}\Big[\sum_{t=0}^{T-1} L_t^i(\mathbf{X}_t^i, \mathbf{U}_t^i, \mathbf{W}_{t+1}^i) + \left\langle \boldsymbol{\lambda}_t^i \ , \mathbf{F}_t^i \right\rangle + K^i(\mathbf{X}_T^i) \Big] \\ \text{s.t. } & \mathbf{X}_{t+1}^i = f_t^i(\mathbf{X}_t^i, \mathbf{U}_t^i, \mathbf{W}_{t+1}^i) \\ & \sigma(\mathbf{U}_t^i) \subset \sigma(\mathbf{W}_0^i, \cdots, \mathbf{W}_t^i) \\ & \Delta_t^i(\mathbf{X}_t^i, \mathbf{U}_t^i, \mathbf{F}_t^i) = 0 \end{aligned}$$

- Solves $\underline{V}^i$ by Dynamic Programming

- Estimates by Monte Carlo the local gradient with the optimal flow $(\mathbf{F}^i)^\sharp = (\mathbf{F}_0^i, \cdots, \mathbf{F}_{T-1}^i)^\sharp$

$$\nabla \underline{V}^i(\boldsymbol{\lambda}^i) = \mathbb{E}\big[(\mathbf{F}^i)^\sharp\big] \in \mathbb{R}^T$$

Workers compute value functions on the assembly line

$$\underline{V}_0^i \leftarrow \underline{V}_1^i \leftarrow \underline{V}_2^i \leftarrow \cdots \underline{V}_T^i$$

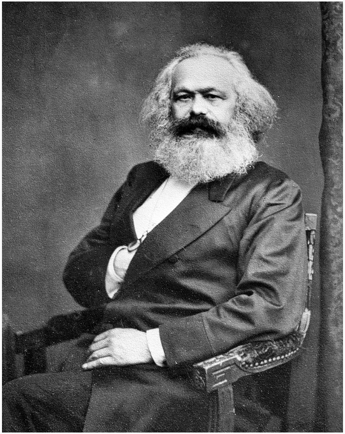
The price process is deterministic $\boldsymbol{\lambda} = (\lambda_0, \cdots, \lambda_{T-1})$. So

- We are able to compute value functions $\{\underline{V}_t^i\}$ by backward recursion
- Each worker has to solve the one-step DP problem

$$\underline{V}_t^i(x_t^i) = \min_{u_t^i, f_t^i} \mathbb{E}\big[L_t(x_t^i, u_t^i, \mathbf{W}_{t+1}^i) + \langle \lambda_t^i , f_t^i \rangle + \underline{V}_{t+1}^i(f_t^i(x_t^i, u_t^i, \mathbf{W}_{t+1}^i)\big]$$

- DP one-step problems formulate as LP or QP problems!

How about resource allocation?



- Same idea, but in a communistic world!
- We fix allocations $\mathbf{R}$ rather than prices $\boldsymbol{\lambda}$ and solve

$$\min_{\mathbf{R}} \overline{V}(\mathbf{R}) := \overline{V}_P(\mathbf{R}) + \overline{V}_T(\mathbf{R})$$

with

$$\overline{V}_P(\mathbf{R}) = \min_{\mathbf{F}} J_P(\mathbf{F}) \quad \text{s.t. } \mathbf{F} - \mathbf{R} = 0 \qquad \overline{V}_T(\mathbf{R}) = \min_{\mathbf{Q}} J_T(\mathbf{Q}) \quad \text{s.t. } A\mathbf{Q} + \mathbf{R} = 0$$

- We must ensure that $\mathbf{R}_t \in im(A)$, that is

$$\mathbf{R}_t^1 + \cdots + \mathbf{R}_t^N = 0$$

- The update step becomes

$$\mathbf{R}^{(k+1)} = \mathbf{R}^{(k)} - \rho \nabla \overline{V}(\mathbf{R}^{(k)})$$

We obtain lower and upper bounds

Theorem

- *For all multipliers* $\boldsymbol{\lambda} = (\boldsymbol{\lambda}_0, \cdots, \boldsymbol{\lambda}_{T-1})$
- *For all allocations* $\mathbf{R} = (\mathbf{R}_0, \cdots, \mathbf{R}_{T-1})$ *such that*

$$\mathbf{R}_t^1 + \cdots + \mathbf{R}_t^N = 0$$

we have

$$\underline{V}(\boldsymbol{\lambda}) \leq V^\sharp \leq \overline{V}(\mathbf{R})$$

Application to the management of microgrids

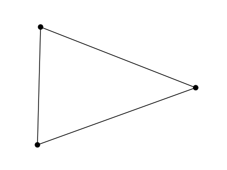
Problem settings

- One day horizon at 15mn time step: $T = 96$
- Weather corresponds to a sunny day in Paris *(June 28th, 2015)*
- We mix three kind of buildings
  1. Battery + Electrical Hot Water Tank
  2. Solar Panel + Electrical Hot Water Tank
  3. Electrical Hot Water Tank

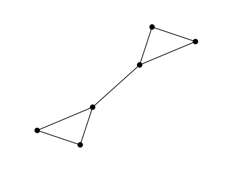
  and suppose that all consumers are commoners sharing their devices

We consider different configurations

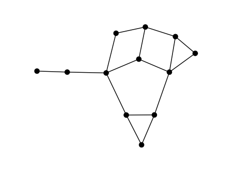
3-Nodes

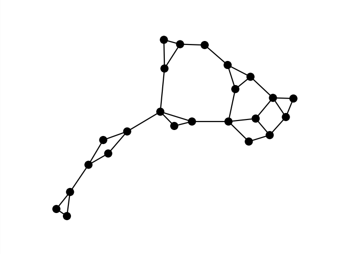


6-Nodes

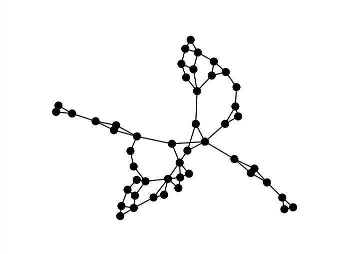


12-Nodes





24-Nodes



48-Nodes

Algorithms inventory

Nodal decomposition

- Encompass price and resource decomposition
- Resolution by Quasi-Newton (BFGS) gradient descent

$$\boldsymbol{\lambda}^{(k+1)} = \boldsymbol{\lambda}^{(k)} + \rho^{(k)} W^{(k)} \nabla \underline{V}(\boldsymbol{\lambda}^{(k)})$$

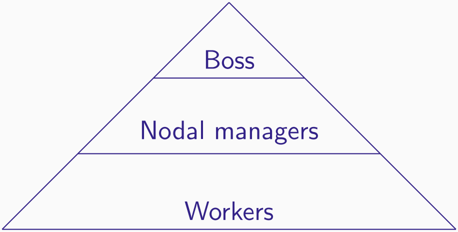
- BFGS iterates till no descent direction is found
- Each nodal subproblem solved by SDDP (quickly converge)
- Oracle $\nabla \underline{V}(\boldsymbol{\lambda})$ estimated by Monte Carlo ($N^{scen} = 1,000$)

SDDP

We use as a reference the good old SDDP algorithms

- Noises $\mathbf{W}_t^1, \cdots, \mathbf{W}_t^N$ are independent node by node (total support size is $|supp(\mathbf{W}_t^i)|^N$.) Need to resample the noise!
- Level-one cut selection algorithm
- Converged once gap between UB and LB is lower than 1%

Each level of hierarchy has its own algorithm



L-BFGS (IPOPT)

SDDP (StochDynamicProgramming)

QP (Gurobi)

All glue code is implemented in Julia 0.6 with JuMP

Results

Graph	**3-Nodes**	**6-Nodes**	**12-Nodes**	**24-Nodes**	**48-Nodes**
SDDP time	1'	3'	10'	79'	453'
SDDP LB	2.252	4.559	8.897	17.528	33.103
SDDP value	2.26 ± 0.006	4.71 ± 0.008	9.36 ± 0.011	18.59 ± 0.016	35.50 ± 0.023
Price time	6'	14'	29'	41'	128'
Price LB	2.137	4.473	8.967	17.870	33.964
Price value	2.28 ± 0.006	4.64 ± 0.008	9.23 ± 0.012	18.39 ± 0.016	34.90 ± 0.023
Resource time	13'	15'	36'	64'	
Resource UB	2.539	5.273	10.537	21.054	
Resource value	2.29 ± 0.006	4.71 ± 0.008	9.31 ± 0.011	18.56 ± 0.016	

Results

Graph	**3-Nodes**	**6-Nodes**	**12-Nodes**	**24-Nodes**	**48-Nodes**
SDDP time	1'	3'	10'	79'	453'
SDDP LB	2.252	4.559	8.897	17.528	33.103
SDDP value	2.26 ± 0.006	4.71 ± 0.008	9.36 ± 0.011	18.59 ± 0.016	35.50 ± 0.023
Price time	6'	14'	29'	41'	128'
Price LB	2.137	4.473	8.967	17.870	33.964
Price value	2.28 ± 0.006	4.64 ± 0.008	9.23 ± 0.012	18.39 ± 0.016	34.90 ± 0.023
Resource time	13'	15'	36'	64'	
Resource UB	2.539	5.273	10.537	21.054	
Resource value	2.29 ± 0.006	4.71 ± 0.008	9.31 ± 0.011	18.56 ± 0.016	

- For large problems ($N \geq 12$), Price Decomposition yields a better lower bound than SDDP (the larger the better)

Results

Graph	**3-Nodes**	**6-Nodes**	**12-Nodes**	**24-Nodes**	**48-Nodes**
SDDP time	1'	3'	10'	79'	453'
SDDP LB	2.252	4.559	8.897	17.528	33.103
SDDP value	2.26 ± 0.006	4.71 ± 0.008	9.36 ± 0.011	18.59 ± 0.016	35.50 ± 0.023
Price time	6'	14'	29'	41'	128'
Price LB	2.137	4.473	8.967	17.870	33.964
Price value	2.28 ± 0.006	4.64 ± 0.008	9.23 ± 0.012	18.39 ± 0.016	34.90 ± 0.023
Resource time	13'	15'	36'	64'	
Resource UB	2.539	5.273	10.537	21.054	
Resource value	2.29 ± 0.006	4.71 ± 0.008	9.31 ± 0.011	18.56 ± 0.016	

- For large problems ($N \geq 12$), Price Decomposition yields a better lower bound than SDDP (the larger the better)
- The upper bound is further from optimal than the lower bound

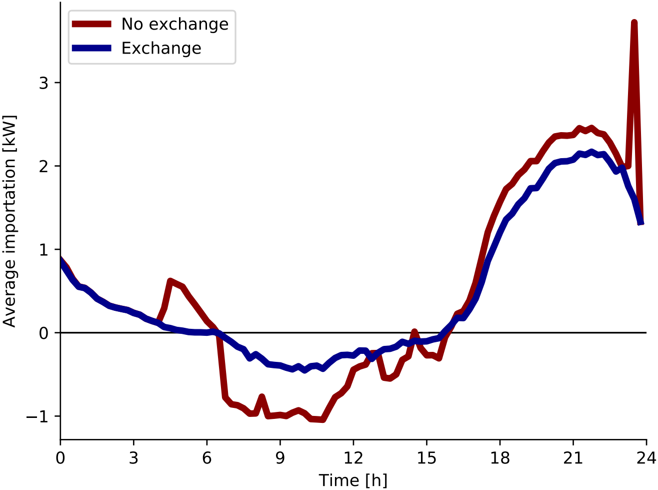
Results

Graph	3-Nodes	6-Nodes	12-Nodes	24-Nodes	48-Nodes
SDDP time	1'	3'	10'	79'	453'
SDDP LB	2.252	4.559	8.897	17.528	33.103
SDDP value	2.26 ± 0.006	4.71 ± 0.008	9.36 ± 0.011	18.59 ± 0.016	35.50 ± 0.023
Price time	6'	14'	29'	41'	128'
Price LB	2.137	4.473	8.967	17.870	33.964
Price value	2.28 ± 0.006	4.64 ± 0.008	9.23 ± 0.012	18.39 ± 0.016	34.90 ± 0.023
Resource time	13'	15'	36'	64'	
Resource UB	2.539	5.273	10.537	21.054	
Resource value	2.29 ± 0.006	4.71 ± 0.008	9.31 ± 0.011	18.56 ± 0.016	

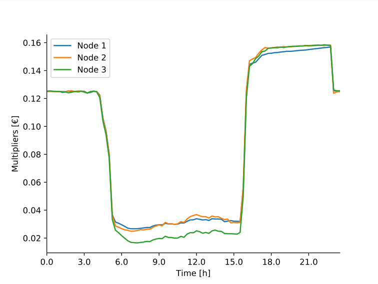
- For large problems ($N \geq 12$), Price Decomposition yields a better lower bound than SDDP (the larger the better)
- The upper bound is further from optimal than the lower bound
- For the biggest instance, Price Decomposition is 3.5x as fast as SDDP

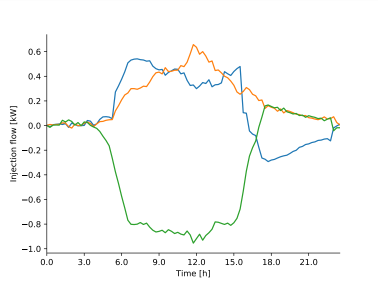
Hunting down the duck curve

Looking at the *average* global electricity import from EDF



Do the nodal units manage well their buildings?





- Node 1: 3kWh Battery
- Node 2: nothing
- Node 3: 16m^2 of solar panels

Looking at Node 3

- During day, Node 3
  - Produces energy with its solar panels
  - Exports energy to other nodes ($F_3 < 0$)
  - Has lowest marginal price λ_3
- During evening, Node 3
  - Imports energy from Node 1 (who has battery)
  - Has larger marginal price than Node 1

$$\lambda_1 < \lambda_3$$

Conclusion

Proletariat of the world, conclude!

- We design an algorithm that decompose spatially and temporally, in a decentralized manner
- Beat SDDP for large instances ($\geq$ 24 nodes)
- Can we obtain tighter bounds?
  If we select properly the stochastic processes $\mathbf{R}$ and $\boldsymbol{\lambda}$,
  we can obtain nodal value functions but with an extended local state