Optimal Management of Storage for Offsetting Solar Power Uncertainty using Multistage Stochastic Programming

Anthony Papavasiliou
Center for Operations Research and Econometrics
Université catholique de Louvain

Joint work with Taku Kaneda (UCL), Bruno Losseau (UCL), Damien Scieur (Ecole Normale Supérieure), Léopold Cambier (Stanford University), Pierre Henneaux (Tractebel), Niels Leemput (Tractebel)

June 15, 2018
20th Power Systems Computation Conference

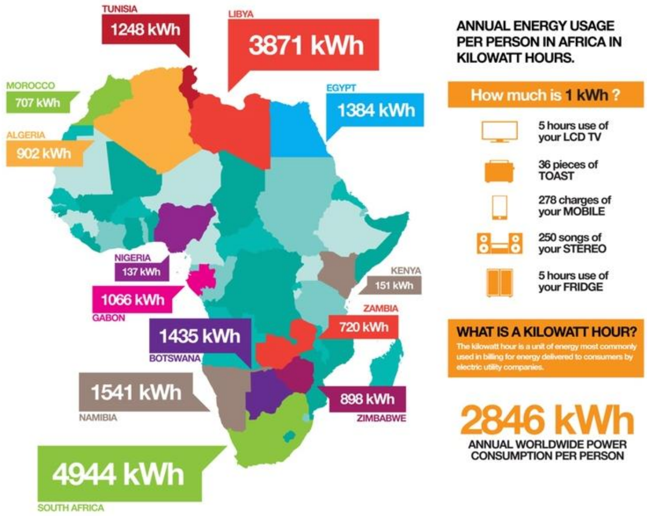
Outline

- Context
- Multistage Stochastic Linear Programming using the **FAST** Toolbox
- Muiltistage Storage Model
- Burkina Faso Case Study

Context

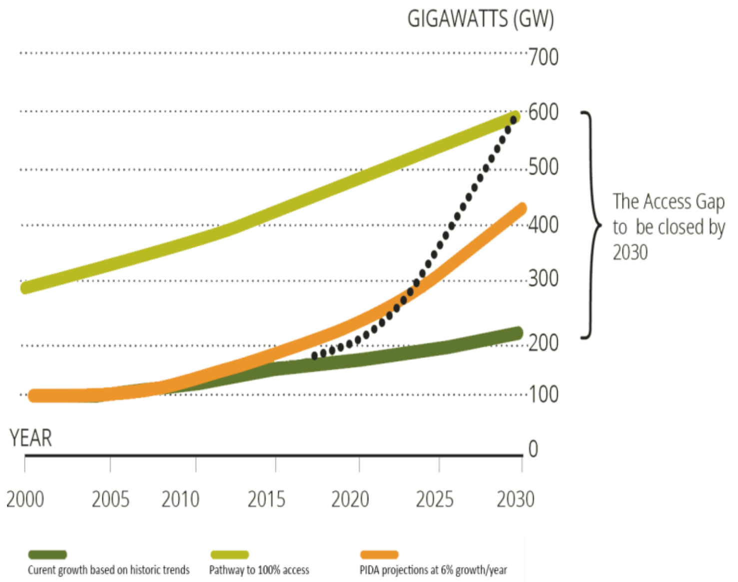
African Energy Poverty

- 600 million Africans with no access to electricity (50% of population)
- Population growth: from 1.2 billion to 2.5 billion in 2050
- **African Renewable Energy Initiative** (AREI): ratified by Europe, Canada, Japan, USA in December 2015
- AREI aims to gather at least **10 billion €** from 2015 to 2020
- Goal of AREI: *universal* access to sustainable energy by 2030

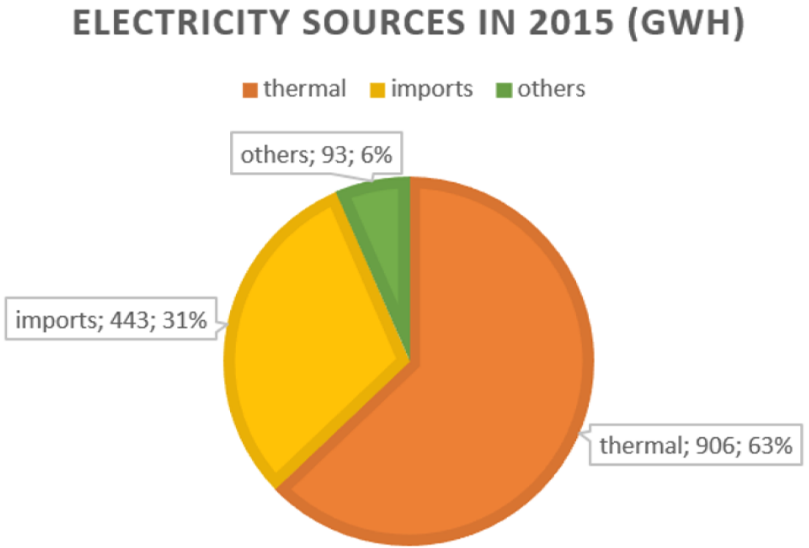


The Role of Renewable Resources

- Total capacity requirement for Africa by 2030 in order to cover energy access gap: 600 GW
- Current buildout plans up to 2030: 220 GW
- Renewable capacity deployment phases of AREI
  - Phase 1: at least 10 GW by 2020
  - Phase 2: at least 300 GW by 2030
- Opportunity for Africa to **leapfrog** to renewable energy



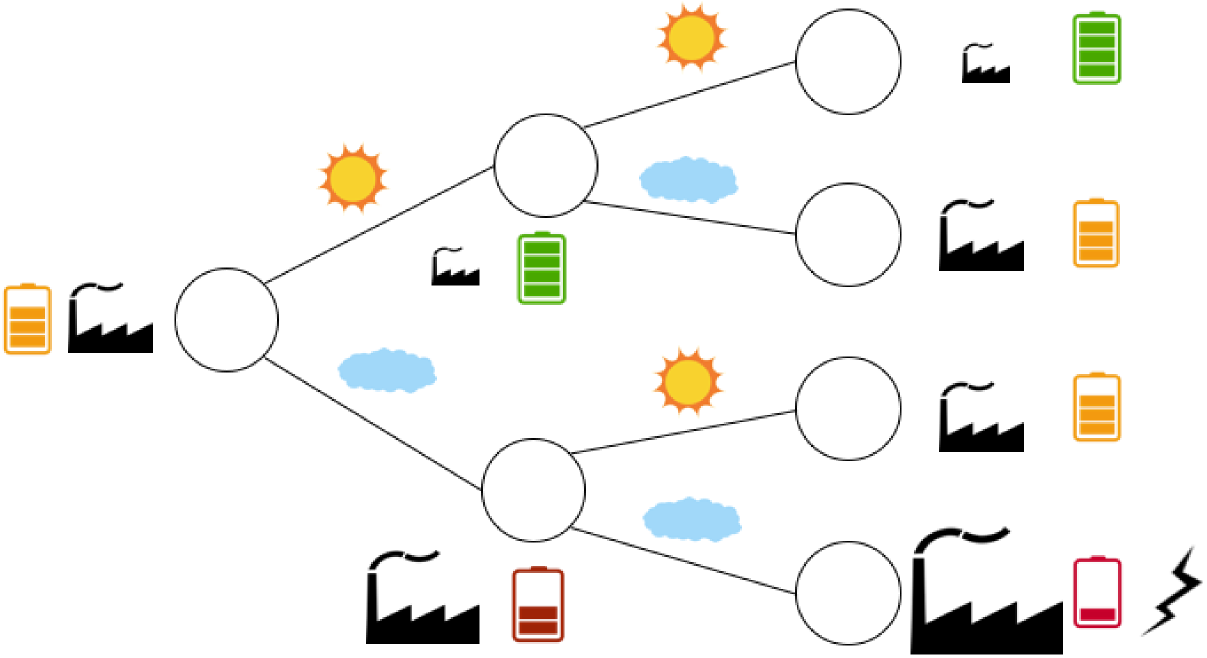
The Case Burkina Faso



- Heavily relies on
  - thermal resources
  - imports from Ivory Coast
- Low electrification
  - less than 5% in rural areas
  - 20% in urban areas
- Frequent load shedding

- Many projects and investments in PV since 2015
  - Zagtouli PV park produces 33 MW since November 2017
  - Government objective is to install 100 MW of solar energy in the national network by 2020, which represents around 30 % of total energy production

Offsetting Solar Power Uncertainty Using Storage



Multistage Stochastic Linear Programming on the **FAST** Toolbox

Multi Stage Stochastic Linear Programming (MSLP)

- Minimize expected cost over a finite time horizon
- The problem is **not scalable** due to the exponential growth of $\Omega_{[t]}$

$$\min_x \sum_{t=1}^{H} \sum_{\omega_{[t]} \in \Omega_{[t]}} \pi_{t,\omega_{[t]}} c_{t,\omega_t}^T x_{t,\omega_{[t]}}$$

$$W_{t,\omega_t} x_{t,\omega_{[t]}} = h_{t,\omega_t} - T_{t,\omega_t} x_{t-1,A(\omega_{[t]})}, t \in T, \omega_{[t]} \in \Omega_{[t]}$$

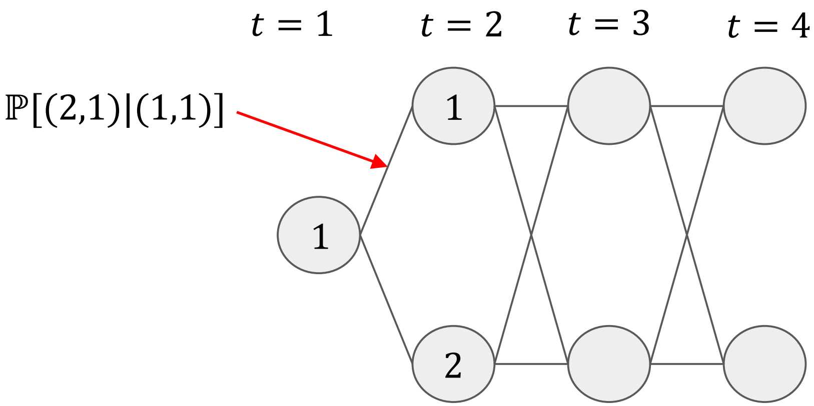
$$x_{t,\omega_{[t]}} \geq 0, t \in T, \omega_{[t]} \in \Omega_{[t]}$$

FAST Toolbox

- Open-source MATLAB implementation of *nested decomposition* for solving MSLP
- Developed by students at Université catholique de Louvain, currently maintained at https://web.stanford.edu/~lcambier/fast/
- Easy way to model MSLP (like CVX or YALMIP)
- All subproblems are compiled at the beginning so that forward and backward passes are performed quickly
- What the user needs to specify:
  - Description of Nested L-Shaped Decomposition Subproblem (NLDS)
  - Uncertainty lattice

Lattice Representation of Uncertainty

- Uncertainty follows a discrete-time, discrete value **Markov process**
  - Nodes: realization of uncertainty (e.g. amount of rainfall): ξ_{t,ω_t}
  - Edges: transition probability: $\mathbb{P}[(t,\omega_t)|(t-1,\omega_{t-1})]$
- **FAST** toolbox instruction: `L = Lattice.latticeEasyMarkovNonConst(H, P)`



Decomposition of the Problem

- Decompose the original problem to subproblems (**NLDS**)
  - Defined for each stage t and each node k
- Value function
  - Cost of remaining stages when the decision is *x*
- **FAST** toolbox uses easy syntax to describe $NLDS_{t,k}$

$$NLDS_{t,k} : \min_{x} c_{t,k}^{T} x + V_{t,k}(x)$$

$$W_{t,k} x = h_{t,k} - T_{t,k} \hat{x}_{t-1}$$

$$x \geq 0$$

Cost from stage $t + 1$ to end of horizon H

Trial decision of the previous stage

Multistage Storage Model

Objective Function and Power Balance

Goal : minimise the total cost

$$\min \left[\sum_{g \in G} C_g(p_g) + CI \cdot pi + VOLL \cdot ls \right]$$

thermal cost — imports — load shedding

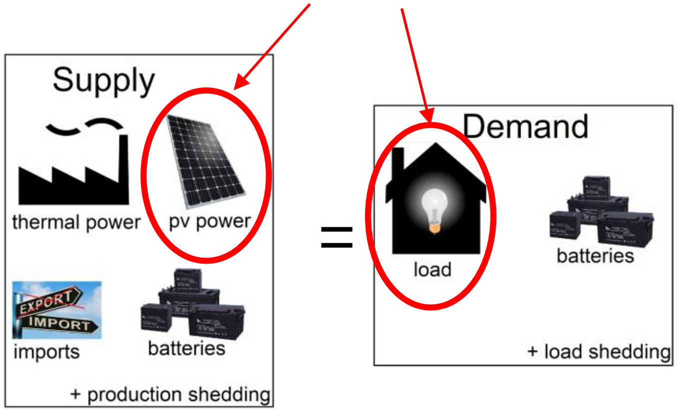
Power Balance

$$NL + \sum_{j \in J} pd_j + ps = \sum_{j \in J} pb_j + \sum_{g \in G} p_g + pi + ls$$

Demand — Supply

- pb_j: Battery power
- p_g: Thermal energy production
- pi: Imported power
- ls: Load shedding
- NL: Net load
- pd_j: Pumping demand of battery
- ps: Production shedding

Stochastic parameters



Other Constraints

- Storage balance in batteries

$$s_j = s_{j,t-1} + \left(\eta_j \cdot pd_j - \frac{pb_j}{\mu_j}\right), \ j \in J$$

- Capacity constraints

$$s_j \leq ST_j, pd_j \leq PD_j, pb_j \leq PB_j, \ j \in J$$
$$pi \leq PI$$
$$PMin_g \leq p_g \leq PMax_g, \ g \in G$$

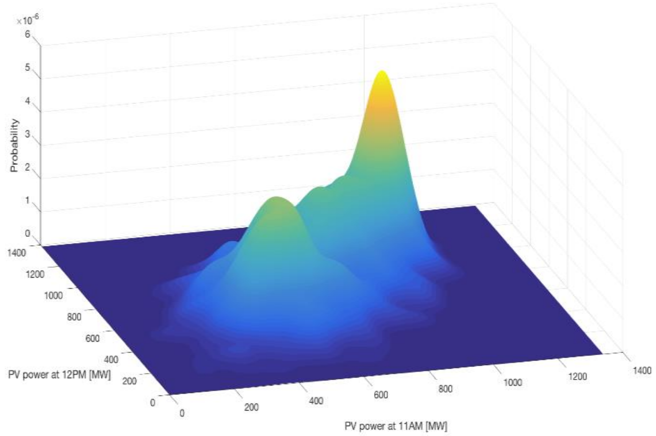
- Non-negativity

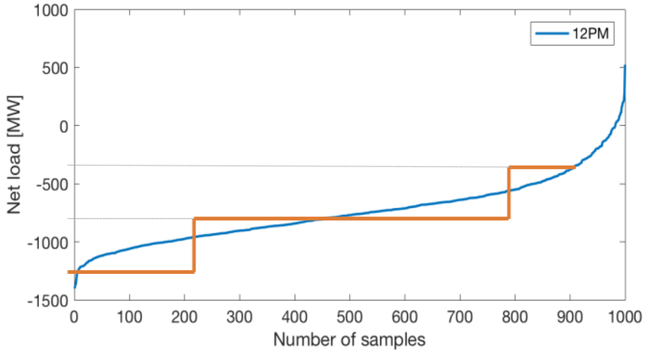
– pb_j: Battery power
– p_g: Thermal energy production
– pi: Imported power
– ls: Load shedding
– NL: Net load
– pd_j: Pumping demand of battery
– ps: Production shedding
– s_j: storage of batteries at stage t
– $\eta_j (< 1)$: efficiency of charging
– $\mu_j (< 1)$: efficiency of discharging

– ST : maximum storage capacity of batteries
– PD : capacity of pumping demand of batteries
– PB : capacity of battery power to extract
– PI : capacity of importation
– $PMax_g$: production capacity of generator g
– $PMin_g$: technical minimum of generator g

Procedure for Building a Lattice

1. Estimate joint distribution of start/end moments of PV output
2. Use kernel density estimation in order to estimate joint distribution of PV power at $t - 1$ and t
3. Populate FAST toolbox lattice by sampling from the continuous distirbution





Case Study

Problem setting

- Single thermal generator with a constant marginal cost
- 5 identical batteries, initially empty
- Lattice: 96 stages (4 days with hourly step), 10 nodes per stage

Physical Parameters

Battery capacity	1000 MWh
Battery charge/discharge rate	200 MW
Import capacity	200 MW
Generator capacity	300 MW

Cost parameters

Generator	200 $/MWh
Import	100 $/MWh
Value of lost load	1000 $/MWh

Cost Comparison

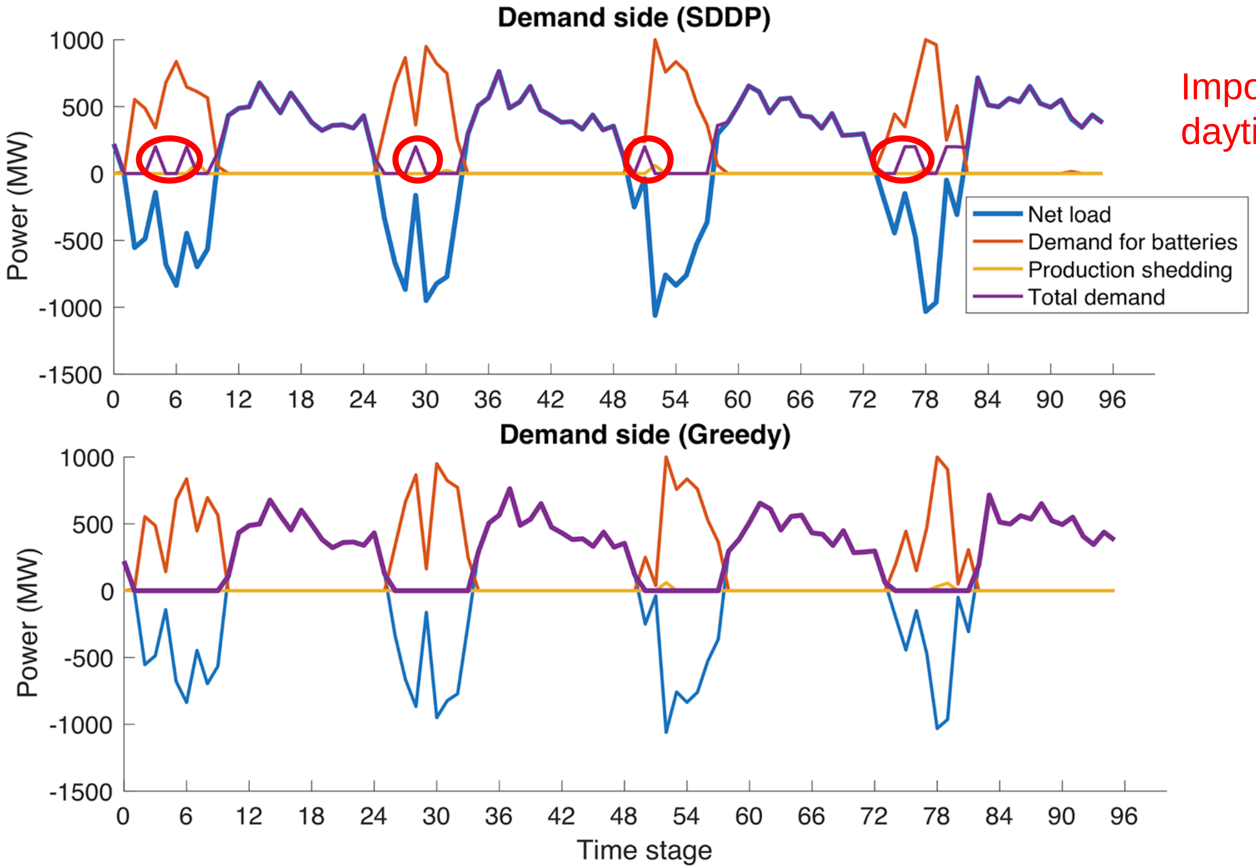
Performance on 1000 samples

	Stochastic programming		Greedy Policy	
	Cost (10^3 \$)	Percentage	Cost (10^3 \$)	Percentage
Total	1279		1850	
Generator	146	11%	1019	55%
Import	1133	89%	660	36%
Load shedding	0	0%	171	9%

Demand Side

Stochastic programming

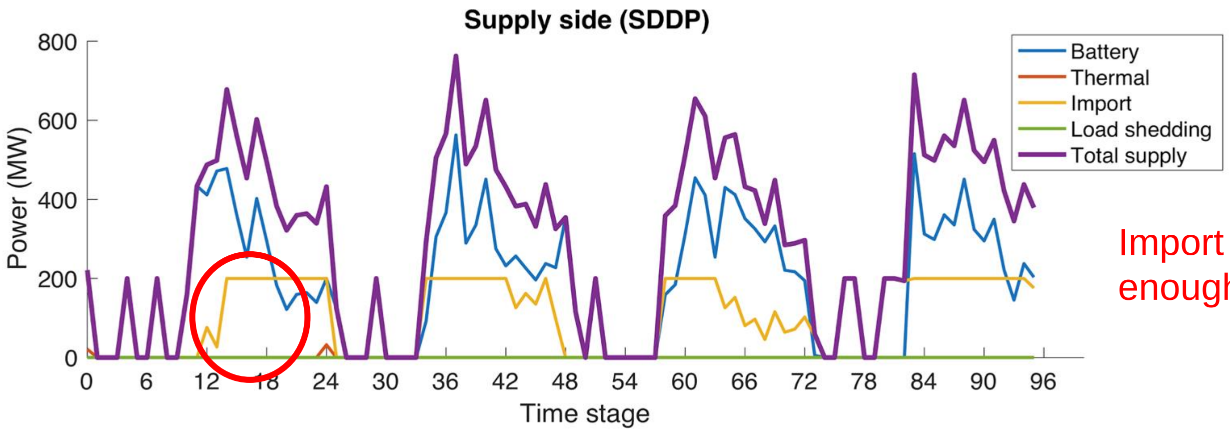
Greedy



Import during the daytime

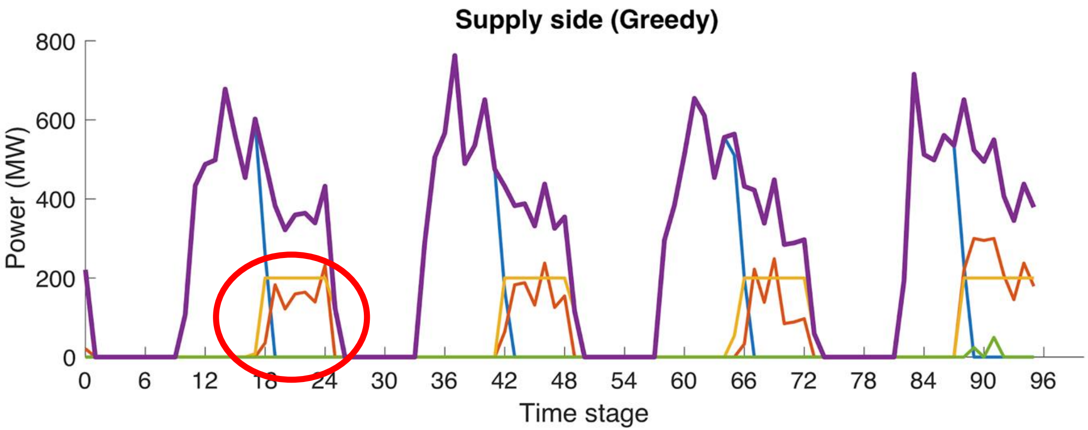
Supply Side

Stochastic programming



Import even if there is enough battery charge

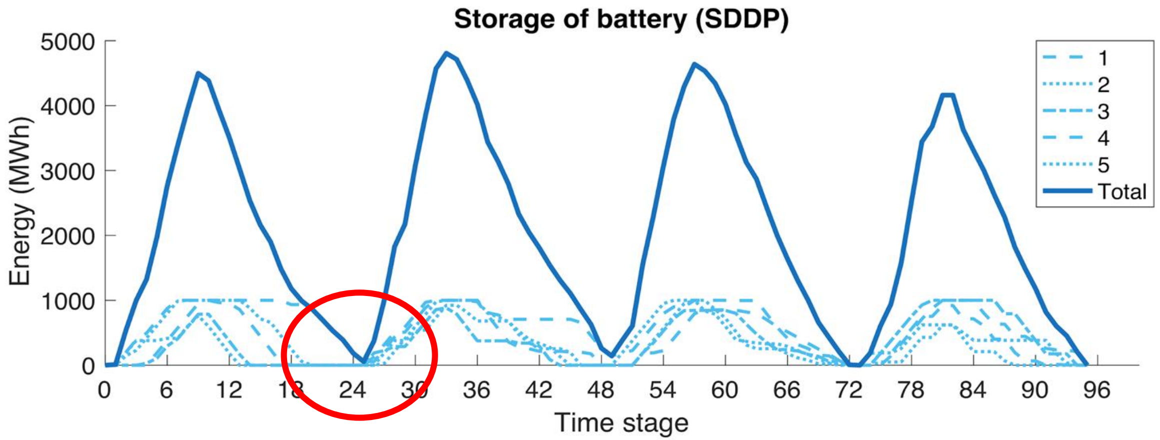
Greedy



Use expensive generator

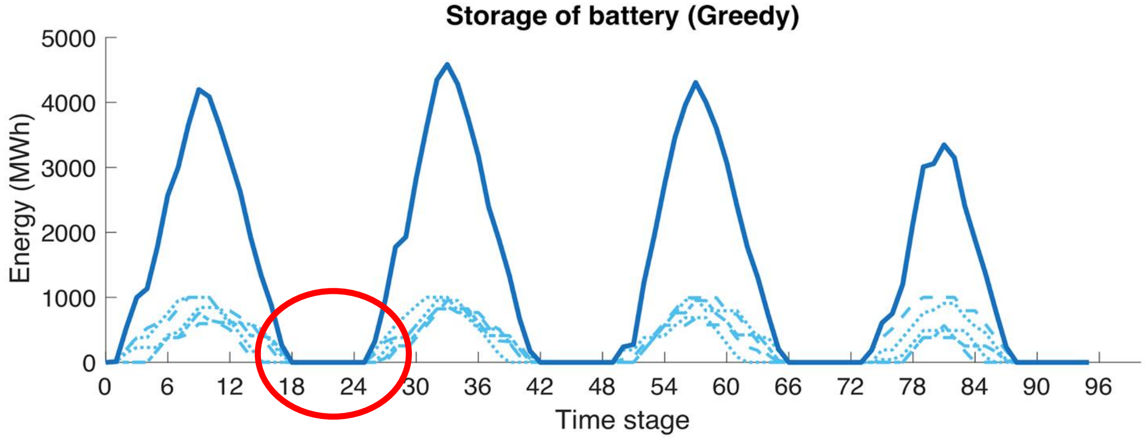
Battery Storage

Stochastic programming



Manage battery to be empty at the end of the day (i.e. at sunrise)

Greedy



Run out of battery

Conclusion

- Renewable energy integration can become an important means of addressing the African energy access gap
- Multistage stochastic programming can be valuable for short-term operation of storage under renewable supply (solar/wind) uncertainty
- We present **FAST**, an open-source MATLAB toolbox for nested decomposition
- Future extensions of **FAST**
  - New language: Julia or Python
  - Parallelization
  - Multistage stochastic nonlinear convex programming

Thank you

For more information

anthony.papavasiliou@uclouvain.be

http://perso.uclouvain.be/anthony.papavasiliou/public_html/home.html