The MIDAS touch:

solving hydro bidding problems using mixed integer programming

Faisal Wahid,
Cédric Gouvernet,
Prof. Andy Philpott,
Prof. Frédéric Bonnans

EPOC Winter Workshop
9th September 2015

Acknowledgements

- Anes Dallagi , EDF R&D France
- Andrew Kerr, Meridian Energy New Zealand
- PGMO
- OSIRIS EDF R&D
- FMJH

Problem Description

Problem Description

- How do you construct offers for hydro producing agents?

Problem Description

- How do you construct offers for hydro producing agents?
- Given that:

Problem Description

- How do you construct offers for hydro producing agents?
- Given that:

Problem Description

- How do you construct offers for hydro producing agents?
- Given that:
  - Price-taking agents

Problem Description

- How do you construct offers for hydro producing agents?
- Given that:
  - Price-taking agents
  - Discrete & finite offer stacks

Problem Description

- How do you construct offers for hydro producing agents?
- Given that:
  - Price-taking agents
  - Discrete & finite offer stacks
  - Cleared offers affect flexibility on future operations of the hydro-scheme

Problem Description

- How do you construct offers for hydro producing agents?
- Given that:
  - Price-taking agents
  - Discrete & finite offer stacks
  - Cleared offers affect flexibility on future operations of the hydro-scheme
- EDF & Meridian:

Problem Description

- How do you construct offers for hydro producing agents?
- Given that:
  - Price-taking agents
  - Discrete & finite offer stacks
  - Cleared offers affect flexibility on future operations of the hydro-scheme
- EDF & Meridian:
  - EDF participate in the French Balancing market using their hydro-schemes

Problem Description

- How do you construct offers for hydro producing agents?
- Given that:
  - Price-taking agents
  - Discrete & finite offer stacks
  - Cleared offers affect flexibility on future operations of the hydro-scheme
- EDF & Meridian:
  - EDF participate in the French Balancing market using their hydro-schemes
  - Meridian participate in the NZ Spot market using the Waitaki hydro-scheme

Problem description

- Produce optimal offers:

Problem description

- Produce optimal offers:
  - Each offer needs to be feasible across the hydro-scheme
  - Treat offer prices as exogenous random variable

Problem description

- Produce optimal offers:
  - Each offer needs to be feasible across the hydro-scheme
  - Treat offer prices as exogenous random variable
- **Hydro-bidding problem:** Combines offer optimization problem with hydro-scheduling problem

Problem description

- Produce optimal offers:
  - Each offer needs to be feasible across the hydro-scheme
  - Treat offer prices as exogenous random variable
- **Hydro-bidding problem:** Combines offer optimization problem with hydro-scheduling problem

Problem description

- Produce optimal offers:
  - Each offer needs to be feasible across the hydro-scheme
  - Treat offer prices as exogenous random variable
- **Hydro-bidding problem:** Combines offer optimization problem with hydro-scheduling problem
  1. Each period solve single stage hydro-bidding model;

Problem description

- Produce optimal offers:
  - Each offer needs to be feasible across the hydro-scheme
  - Treat offer prices as exogenous random variable
- **Hydro-bidding problem:** Combines offer optimization problem with hydro-scheduling problem
  1. Each period solve single stage hydro-bidding model;
  2. Then observe market clearing prices & dispatch;

Problem description

- Produce optimal offers:
  - Each offer needs to be feasible across the hydro-scheme
  - Treat offer prices as exogenous random variable
- **Hydro-bidding problem:** Combines offer optimization problem with hydro-scheduling problem
  1. Each period solve single stage hydro-bidding model;
  2. Then observe market clearing prices & dispatch;
  3. Solve for next period based on observed prices & dispatch

Hydro-bidding model

Hydro-bidding model

Objective

Maximize expected profit from offerstack, & expected future value of the remaining water, conditioned on observed prices.

Hydro-bidding model

Objective

Maximize expected profit from offerstack, & expected future value of the remaining water, conditioned on observed prices.

Constraints

Hydro-bidding model

Objective

Maximize expected profit from offerstack, & expected future value of the remaining water, conditioned on observed prices.

Constraints

- Respect physical constraints of the hydro-scheme (i.e. storage bounds and water flow balance)

Hydro-bidding model

Objective

Maximize expected profit from offerstack, & expected future value of the remaining water, conditioned on observed prices.

Constraints

- Respect physical constraints of the hydro-scheme (i.e. storage bounds and water flow balance)
- Respect the operational constraints of the turbines (i.e. plant capacity)

Hydro-bidding model

Objective

Maximize expected profit from offerstack, & expected future value of the remaining water, conditioned on observed prices.

Constraints

- Respect physical constraints of the hydro-scheme (i.e. storage bounds and water flow balance)
- Respect the operational constraints of the turbines (i.e. plant capacity)
- Respect the physical constraints of the hydro-scheme

Hydro-bidding model

Objective

Maximize expected profit from offerstack, & expected future value of the remaining water, conditioned on observed prices.

Constraints

- Respect physical constraints of the hydro-scheme (i.e. storage bounds and water flow balance)
- Respect the operational constraints of the turbines (i.e. plant capacity)
- Respect the physical constraints of the hydro-scheme
- Meet ancillary requirements:
  - Frequency containment reserve
  - Frequency restoration reserve

Description of HERBS

Description of HERBS

- Stands for **H**ydro **E**lectric **R**eservoir **B**idding **S**ystem

Description of HERBS

- Stands for **H**ydro **E**lectric **R**eservoir **B**idding **S**ystem
- Stochastic dynamic programming hydro-bidding model

Description of HERBS

- Stands for **H**ydro **E**lectric **R**eservoir **B**idding **S**ystem
- Stochastic dynamic programming hydro-bidding model
- Uncertainty is the market clearing price

Description of HERBS

- Stands for **H**ydro **E**lectric **R**eservoir **B**idding **S**ystem
- Stochastic dynamic programming hydro-bidding model
- Uncertainty is the market clearing price
- Market clearing price modelled as a Markov chain of price distributions

Description of HERBS

- Stands for **H**ydro **E**lectric **R**eservoir **B**idding **S**ystem
- Stochastic dynamic programming hydro-bidding model
- Uncertainty is the market clearing price
- Market clearing price modelled as a Markov chain of price distributions
  - Partition overall price distribution into segments

Description of HERBS

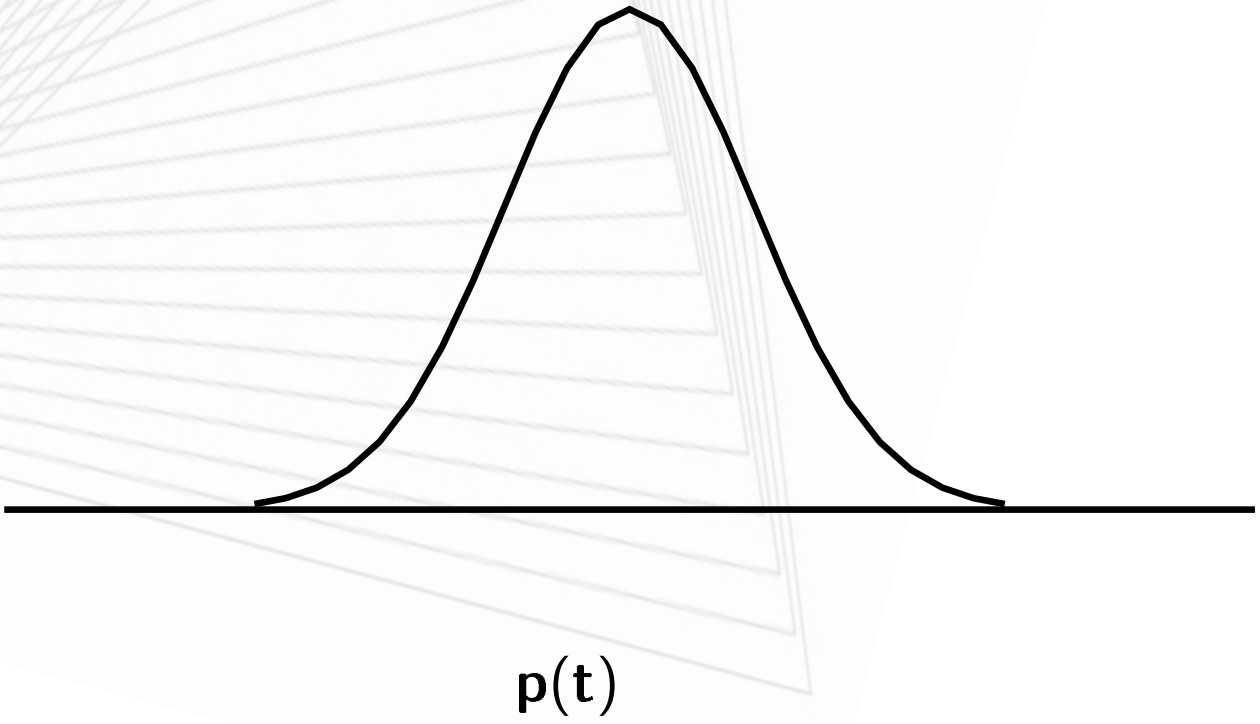
- Stands for **H**ydro **E**lectric **R**eservoir **B**idding **S**ystem
- Stochastic dynamic programming hydro-bidding model
- Uncertainty is the market clearing price
- Market clearing price modelled as a Markov chain of price distributions
  - Partition overall price distribution into segments
  - Price segment represents conditional distribution, based on the observed price of previous period

Description of HERBS

- Stands for **H**ydro **E**lectric **R**eservoir **B**idding **S**ystem
- Stochastic dynamic programming hydro-bidding model
- Uncertainty is the market clearing price
- Market clearing price modelled as a Markov chain of price distributions
  - Partition overall price distribution into segments
  - Price segment represents conditional distribution, based on the observed price of previous period
- Assumptions:
  - Price taking hydro power producers
  - Deterministic inflow
  - Discrete finite offers (i.e. an offer stack)

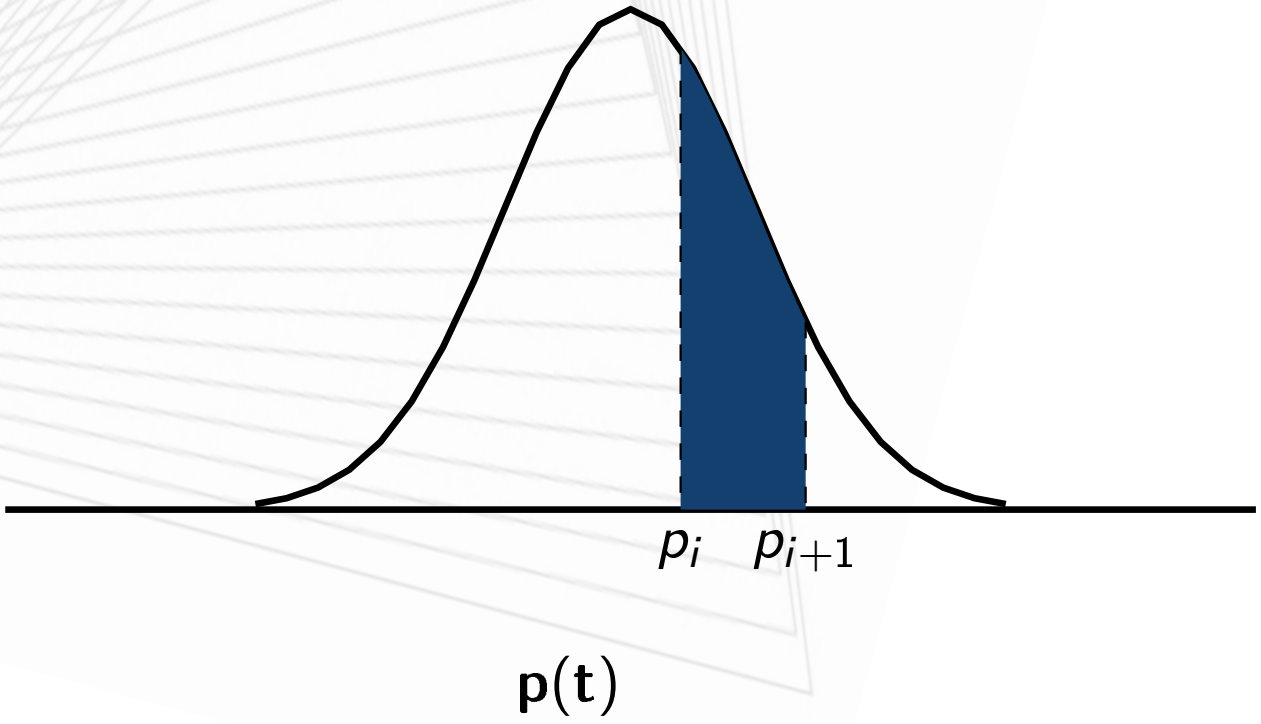
Formulation I: Modelling price

- Market clearing price $p(t)$ is random variable



Formulation I: Modelling price

- Market clearing price $p(t)$ is random variable
- Partition price $p(t)$ into M number of segments



Formulation I: Modelling price

- Price segments:

$$\{[p_1, p_2), [p_2, p_3), \ldots, [p_{M-1}, p_M), [p_M, p_{M+1})\}$$

Formulation I: Modelling price

- Price segments:

$$\{[p_1, p_2), [p_2, p_3), \ldots, [p_{M-1}, p_M), [p_M, p_{M+1})\}$$

- Each segment represents the price of each offer

Formulation I: Modelling price

- Price segments:

$$\{[p_1, p_2), [p_2, p_3), \ldots, [p_{M-1}, p_M), [p_M, p_{M+1})\}$$

- Each segment represents the price of each offer
- Transition probability of moving from segment i to segment j:

$$P_{i,j}(t) = \mathbb{P}[p(t) \in [p_j, p_{j+1}) \mid p(t-1) \in [p_i, p_{i+1})].$$

Formulation I: Modelling price

- Price segments:

$$\{[p_1, p_2), [p_2, p_3), \ldots, [p_{M-1}, p_M), [p_M, p_{M+1})\}$$

- Each segment represents the price of each offer
- Transition probability of moving from segment i to segment j:

$$P_{i,j}(t) = \mathbb{P}[p(t) \in [p_j, p_{j+1}) \mid p(t-1) \in [p_i, p_{i+1})].$$

- Conditionally expected price of segment j:

$$\pi_{t,j} = \mathbb{E}[p(t) \mid p(t) \in [p_j, p_{j+1})].$$

Formulation II: Modelling offerstack

Formulation II: Modelling offerstack

- **Decision variable:** Discrete offer quantities (i.e. offerstack)

$$q_t = (q_{t,1}, \ldots, q_{t,M})$$

Formulation II: Modelling offerstack

- **Decision variable:** Discrete offer quantities (i.e. offerstack)

$$q_t = (q_{t,1}, \ldots, q_{t,M})$$

- Expected profit from offer stack:

$$\sum_{j=1}^{M} P_{i,j}(t)\pi_{t,j}q_{t,j}.$$

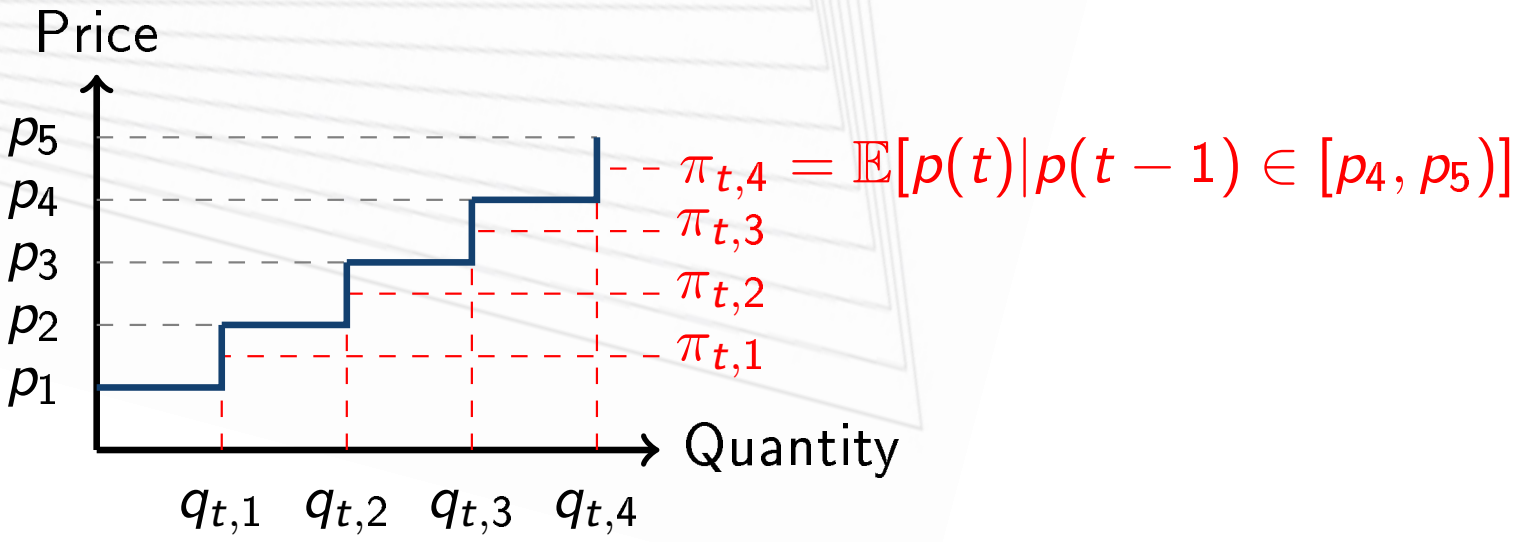
Formulation II: Modelling offerstack

- **Decision variable:** Discrete offer quantities (i.e. offerstack)

$$q_t = (q_{t,1}, \ldots, q_{t,M})$$

- Expected profit from offer stack:

$$\sum_{j=1}^{M} P_{i,j}(t)\pi_{t,j}q_{t,j}.$$



Formulation III: HERBS Objective function

Formulation III: HERBS Objective function

- **State variable**: Storage of reservoir r starting period $t+1$ for each offer j

Formulation III: HERBS Objective function

- **State variable**: Storage of reservoir r starting period $t+1$ for each offer j

$$x_{t+1,j,r} \in [\underline{x}_r, \overline{x}_r]$$

Formulation III: HERBS Objective function

- **State variable**: Storage of reservoir r starting period $t+1$ for each offer j

$$x_{t+1,j,r} \in [\underline{x}_r, \overline{x}_r]$$

- $x_{t,i}$: Observed storage level from cleared offer $q_{t-1,i}$ in previous period $t-1$

Formulation III: HERBS Objective function

- **State variable**: Storage of reservoir r starting period $t+1$ for each offer j

$$x_{t+1,j,r} \in [\underline{x}_r, \overline{x}_r]$$

- $x_{t,i}$: Observed storage level from cleared offer $q_{t-1,i}$ in previous period $t-1$
- Profit objective for period t (i.e. objective function):

$$V_{t,i}(x_{t,i}) =$$

Formulation III: HERBS Objective function

- **State variable**: Storage of reservoir r starting period $t+1$ for each offer j

$$x_{t+1,j,r} \in [\underline{x}_r, \overline{x}_r]$$

- $x_{t,i}$: Observed storage level from cleared offer $q_{t-1,i}$ in previous period $t-1$
- Profit objective for period t (i.e. objective function):

$$V_{t,i}(x_{t,i}) = \max \mathbb{E}[\text{Profit}(q_t)]$$
$$\sum_{j=1}^{M} P_{i,j}(t)\pi_{t,j}q_{t,j}$$

Formulation III: HERBS Objective function

- **State variable**: Storage of reservoir r starting period $t+1$ for each offer j

$$x_{t+1,j,r} \in [\underline{x}_r, \overline{x}_r]$$

- $x_{t,i}$: Observed storage level from cleared offer $q_{t-1,i}$ in previous period $t-1$
- Profit objective for period t (i.e. objective function):

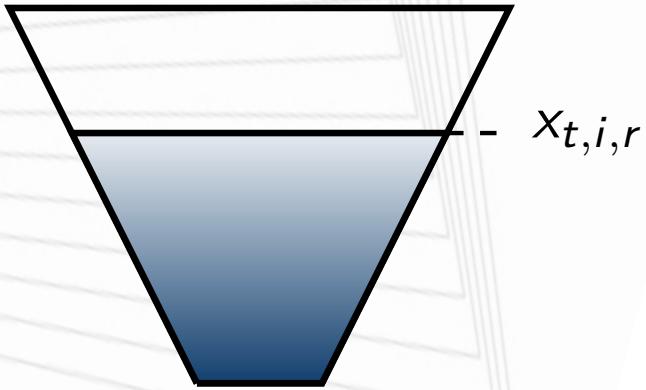
$$\begin{aligned} V_{t,i}(x_{t,i}) = \quad & \max \mathbb{E}[\text{Profit}(q_t)] \quad + \mathbb{E}[\text{FutureProfit}(x_{t+1,j})] \\ & \sum_{j=1}^{M} P_{i,j}(t)\pi_{t,j}q_{t,j} \quad + \sum_{j=1}^{M} P_{i,j}(t)V_{t+1,j}(x_{t+1,j}) \end{aligned}$$

Formulation IV: Water-balance constraint

Formulation IV: Water-balance constraint

- Enforce the water-balance constraint:

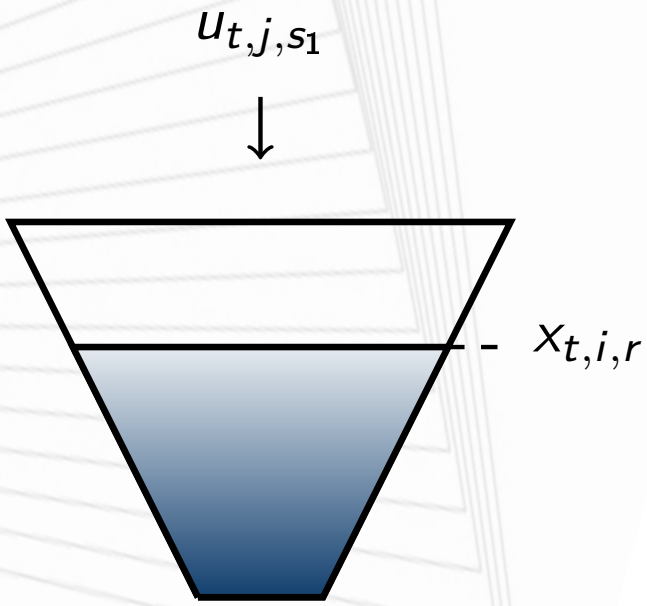
$x_{t,i,r}$



Formulation IV: Water-balance constraint

- Enforce the water-balance constraint:

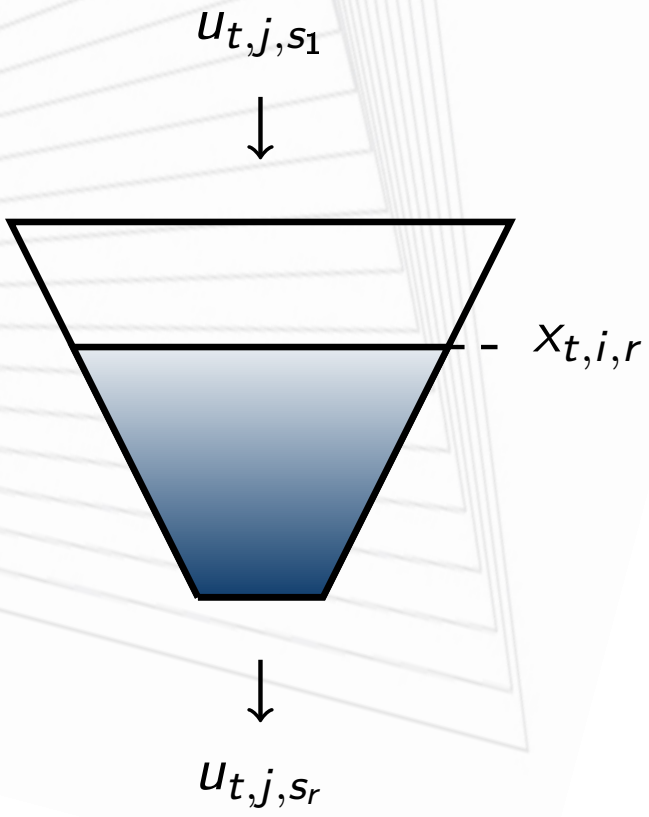
$x_{t,i,r}$ $+$upstream flow



Formulation IV: Water-balance constraint

- Enforce the water-balance constraint:

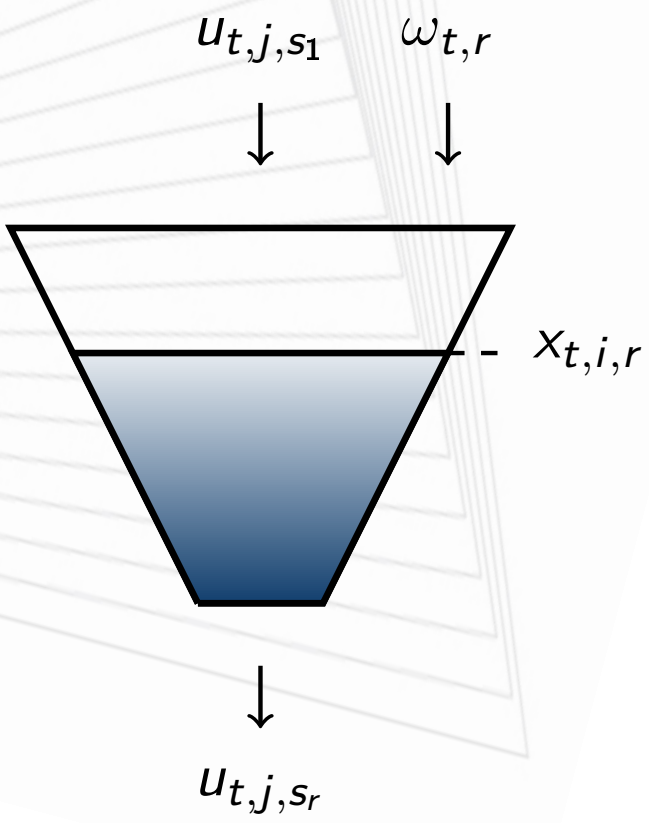
$$x_{t,i,r} \quad + \text{upstream flow} \; - \text{station discharge}$$



Formulation IV: Water-balance constraint

- Enforce the water-balance constraint:

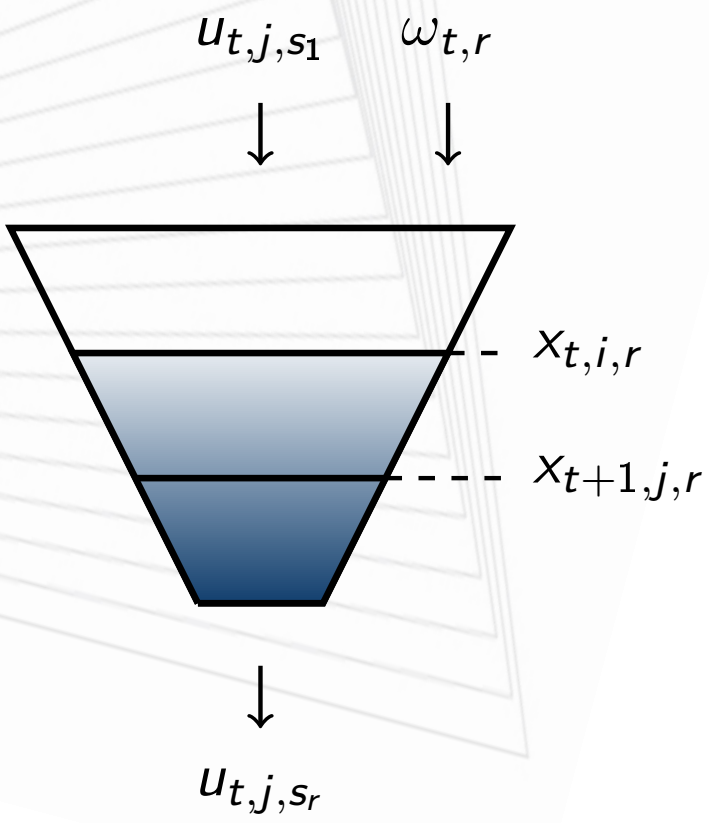
$$x_{t,i,r} \quad + \text{upstream flow} \; - \text{station discharge} \quad + \text{inflow}$$



Formulation IV: Water-balance constraint

- Enforce the water-balance constraint:

$$x_{t+1,j,r} = \quad x_{t,i,r} \quad + \text{upstream flow} \; - \text{station discharge} \quad + \text{inflow}$$



Formulation V: Discrete power production

Formulation V: Discrete power production

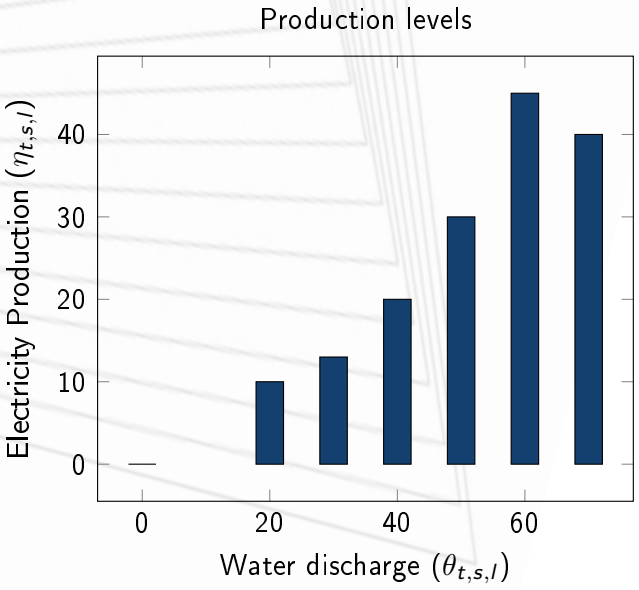
- $(\theta_{t,s,l}, \eta_{t,s,l})$: Discrete pairs of waterflow (cubic meters) and power generation quantities (MWhs)

Formulation V: Discrete power production

- $(\theta_{t,s,l}, \eta_{t,s,l})$: Discrete pairs of waterflow (cubic meters) and power generation quantities (MWhs)
- $z_{j,s,l}$: Binary variable used to select a single pair of discrete production

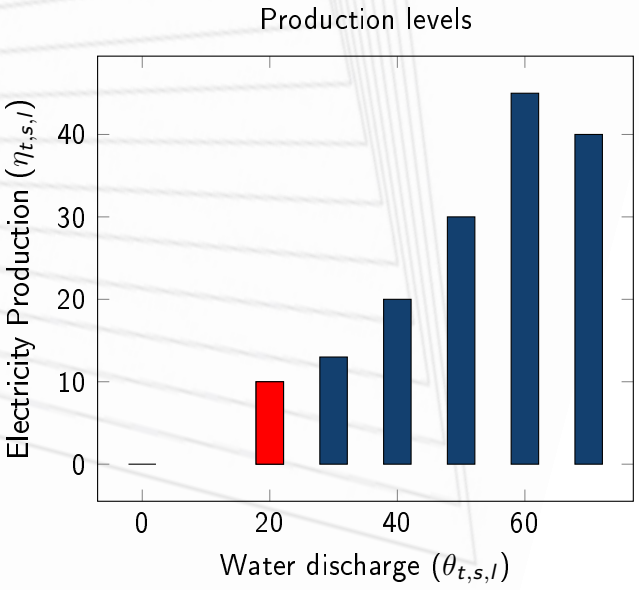
Formulation V: Discrete power production

- $(\theta_{t,s,l}, \eta_{t,s,l})$: Discrete pairs of waterflow (cubic meters) and power generation quantities (MWhs)
- $z_{j,s,l}$: Binary variable used to select a single pair of discrete production



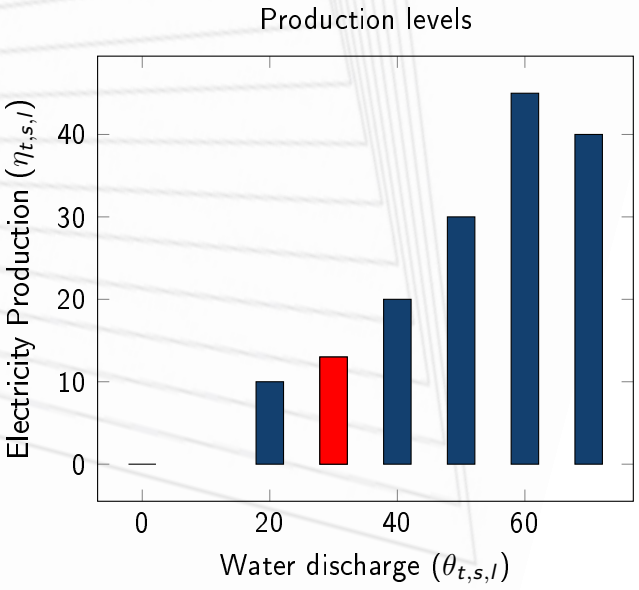
Formulation V: Discrete power production

- $(\theta_{t,s,l}, \eta_{t,s,l})$: Discrete pairs of waterflow (cubic meters) and power generation quantities (MWhs)
- $z_{j,s,l}$: Binary variable used to select a single pair of discrete production



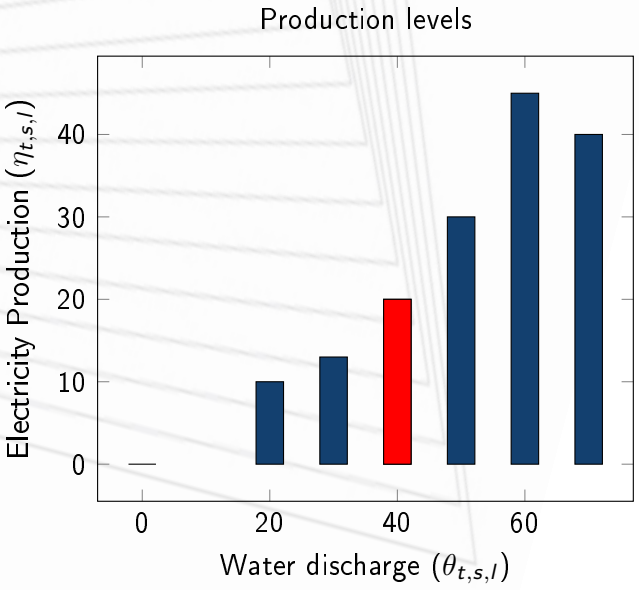
Formulation V: Discrete power production

- $(\theta_{t,s,l}, \eta_{t,s,l})$: Discrete pairs of waterflow (cubic meters) and power generation quantities (MWhs)
- $z_{j,s,l}$: Binary variable used to select a single pair of discrete production



Formulation V: Discrete power production

- $(\theta_{t,s,l}, \eta_{t,s,l})$: Discrete pairs of waterflow (cubic meters) and power generation quantities (MWhs)
- $z_{j,s,l}$: Binary variable used to select a single pair of discrete production



Formulation V: Discrete power production

Formulation V: Discrete power production

- Enforce discrete generation:

Formulation V: Discrete power production

- Enforce discrete generation:

$$u_{t,j,s} =$$

Formulation V: Discrete power production

- Enforce discrete generation:

$$u_{t,j,s} = \sum_{l=1}^{L_{t,s}} \theta_{t,s,l} z_{j,s,l}$$

Formulation V: Discrete power production

- Enforce discrete generation:

$$u_{t,j,s} = \sum_{l=1}^{L_{t,s}} \theta_{t,s,l} z_{j,s,l}$$

- Enforce the j'th offer quantity as total generation of the hydro-scheme:

Formulation V: Discrete power production

- Enforce discrete generation:

$$u_{t,j,s} = \sum_{l=1}^{L_{t,s}} \theta_{t,s,l} z_{j,s,l}$$

- Enforce the j'th offer quantity as total generation of the hydro-scheme:

$$q_{t,j} =$$

Formulation V: Discrete power production

- Enforce discrete generation:

$$u_{t,j,s} = \sum_{l=1}^{L_{t,s}} \theta_{t,s,l} z_{j,s,l}$$

- Enforce the j'th offer quantity as total generation of the hydro-scheme:

$$q_{t,j} = \sum_{s=1}^{S} \sum_{l=1}^{L_{t,s}} \eta_{t,s,l} z_{j,s,l}$$

Formulation V: Discrete power production

- Enforce discrete generation:

$$u_{t,j,s} = \sum_{l=1}^{L_{t,s}} \theta_{t,s,l} z_{j,s,l}$$

- Enforce the j'th offer quantity as total generation of the hydro-scheme:

$$q_{t,j} = \sum_{s=1}^{S} \sum_{l=1}^{L_{t,s}} \eta_{t,s,l} z_{j,s,l}$$

- Monotonicity of offers:

Formulation V: Discrete power production

- Enforce discrete generation:

$$u_{t,j,s} = \sum_{l=1}^{L_{t,s}} \theta_{t,s,l} z_{j,s,l}$$

- Enforce the j'th offer quantity as total generation of the hydro-scheme:

$$q_{t,j} = \sum_{s=1}^{S} \sum_{l=1}^{L_{t,s}} \eta_{t,s,l} z_{j,s,l}$$

- Monotonicity of offers:

$$q_{t,j}$$

Formulation V: Discrete power production

- Enforce discrete generation:

$$u_{t,j,s} = \sum_{l=1}^{L_{t,s}} \theta_{t,s,l} z_{j,s,l}$$

- Enforce the j'th offer quantity as total generation of the hydro-scheme:

$$q_{t,j} = \sum_{s=1}^{S} \sum_{l=1}^{L_{t,s}} \eta_{t,s,l} z_{j,s,l}$$

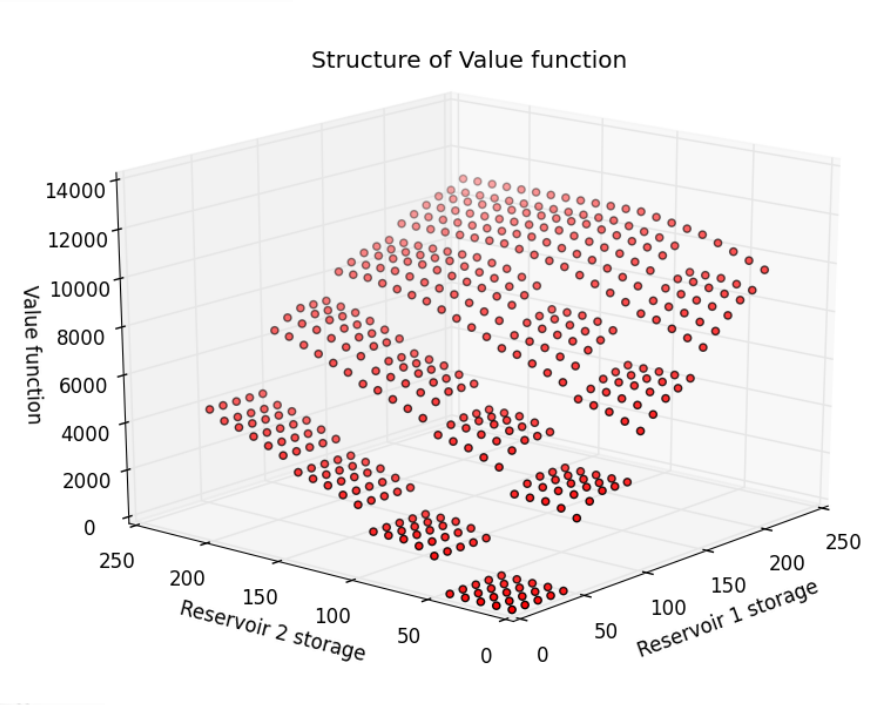
- Monotonicity of offers:

$$q_{t,j} \leq q_{t,j+1}$$

Value function

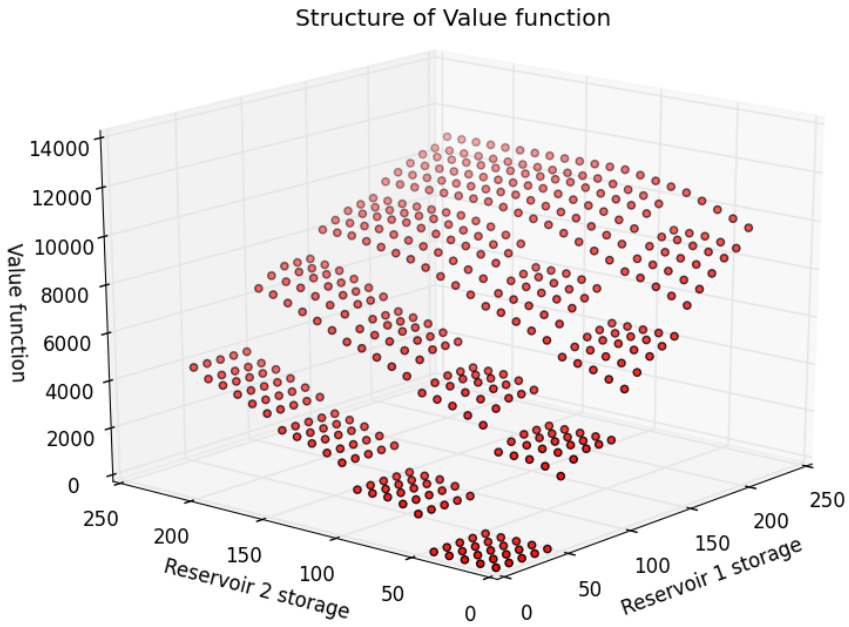
Value function

- $V_{t+1,j}(x_{t+1,j})$ is **monotonic & non-concave**



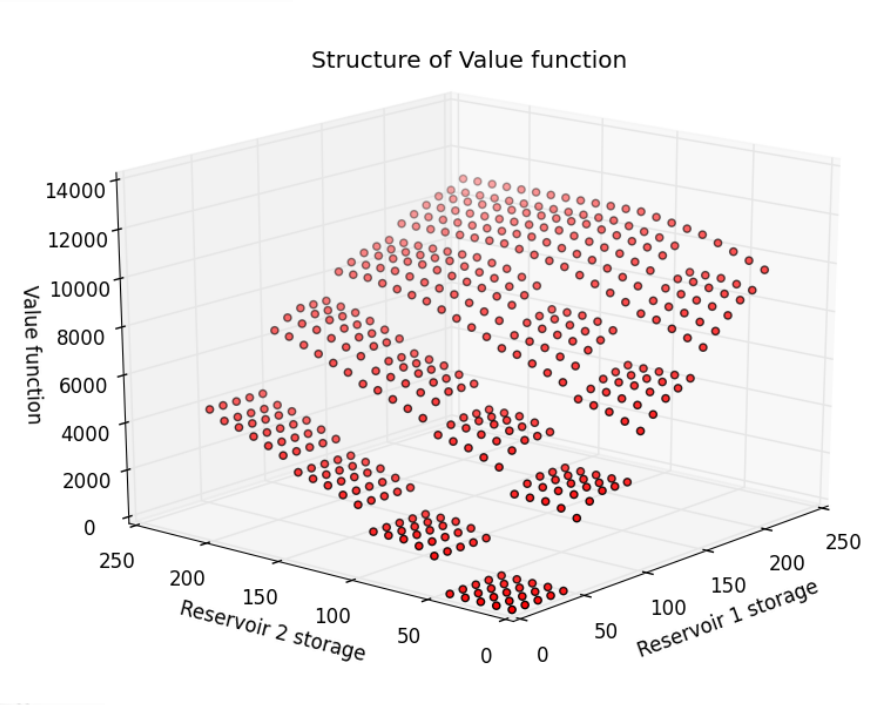
Value function

- $V_{t+1,j}(x_{t+1,j})$ is **monotonic & non-concave**
- Due to discrete production & offers = total block dispatch



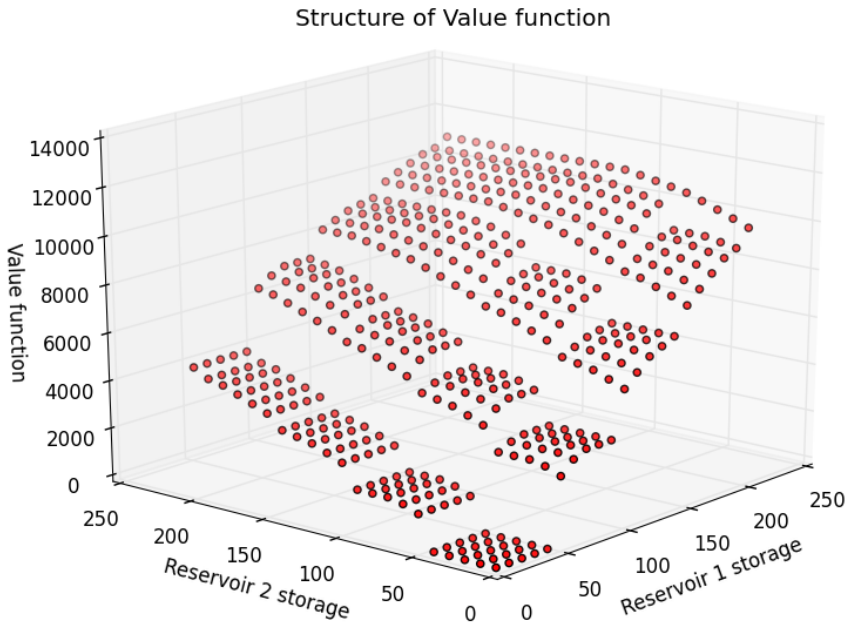
Value function

- $V_{t+1,j}(x_{t+1,j})$ is **monotonic & non-concave**
- Due to discrete production & offers = total block dispatch
- Creates plateaus of similar value function values



Value function

- $V_{t+1,j}(x_{t+1,j})$ is **monotonic & non-concave**
- Due to discrete production & offers = total block dispatch
- Creates plateaus of similar value function values
- Makes HERBS hard to solve



Solution approaches

Solution approaches

Solve the HERBS via existing algorithms:

Solution approaches

Solve the HERBS via existing algorithms:

- **Stochastic dynamic program (SDP)** (Pritchard & Zakeri [1])
  - Limited by the *'curse of dimensionality'*

Solution approaches

Solve the HERBS via existing algorithms:

- **Stochastic dynamic program (SDP)** (Pritchard & Zakeri [1])
  - Limited by the *'curse of dimensionality'*
- **Stochastic approximate dynamic program (SADP)** (Löhndorf et al. [2])
  - Difficult to find appropriate basis functions to represent the value function

Solution approaches

Solve the HERBS via existing algorithms:

- **Stochastic dynamic program (SDP)** (Pritchard & Zakeri [1])
  - Limited by the *'curse of dimensionality'*
- **Stochastic approximate dynamic program (SADP)** (Löhndorf et al. [2])
  - Difficult to find appropriate basis functions to represent the value function
- **Stochastic dual dynamic program (SDDP)** (Pereira & Pinto [3])
  - Assumes concave value functions

Solution approaches

Solve the HERBS via existing algorithms:

- **Stochastic dynamic program (SDP)** (Pritchard & Zakeri [1])
  - Limited by the *'curse of dimensionality'*
- **Stochastic approximate dynamic program (SADP)** (Löhndorf et al. [2])
  - Difficult to find appropriate basis functions to represent the value function
- **Stochastic dual dynamic program (SDDP)** (Pereira & Pinto [3])
  - Assumes concave value functions
- **Linear decision rules** (Braathen & Eriksrud [2])
  - Optimal policy not guaranteed, due to restriction of the decision space

Description of MIDAS

Description of MIDAS

- **M**ixed-**i**nteger **D**ynamic **A**pproximation **S**cheme

Description of MIDAS

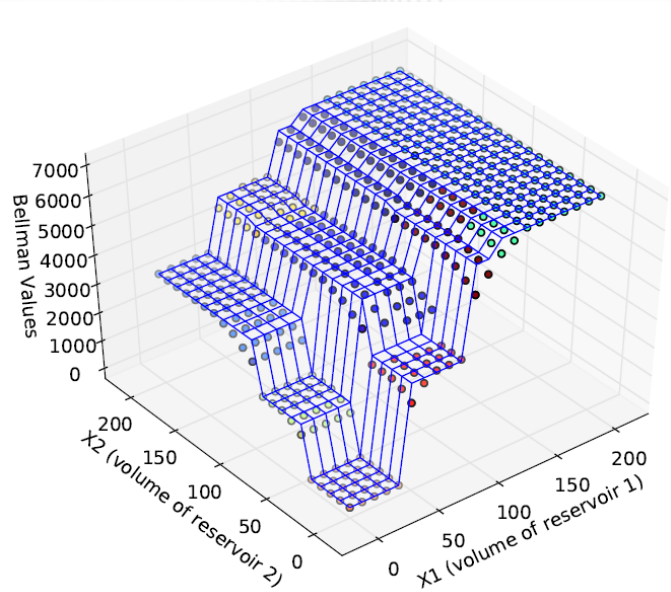
- **M**ixed-**i**nteger **D**ynamic **A**pproximation **S**cheme
- Stochastic optimization algorithm designed to solve **sequential decision problems under uncertainty**

Description of MIDAS

- **M**ixed-integer **D**ynamic **A**pproximation **Scheme**
- Stochastic optimization algorithm designed to solve **sequential decision problems under uncertainty**
- Creates an outer approximation of $V_{t+1,j}(x_{t+1,j})$

Description of MIDAS

- **M**ixed-**i**nteger **D**ynamic **A**pproximation **S**cheme
- Stochastic optimization algorithm designed to solve **sequential decision problems under uncertainty**
- Creates an outer approximation of $V_{t+1,j}(x_{t+1,j})$
- Uses piece-wise constant functions to represent the plateaus of $V_{t+1,j}(x_{t+1,j})$



Formulation I

Formulation I

- $\hat{v}_{t+1,j}$: Variable that approximates the value function $V_{t+1,j}(x_{t+1,j})$

Formulation I

- $\hat{v}_{t+1,j}$: Variable that approximates the value function $V_{t+1,j}(x_{t+1,j})$
- $\mathcal{H}_{t+1}$: Set of piece-wise constant functions

Formulation I

- $\hat{v}_{t+1,j}$: Variable that approximates the value function $V_{t+1,j}(x_{t+1,j})$
- $\mathcal{H}_{t+1}$: Set of piece-wise constant functions
- $a_{j,h,r}$: Points in the state space where piece-wise function h starts in dimension r

Formulation I

- $\hat{v}_{t+1,j}$: Variable that approximates the value function $V_{t+1,j}(x_{t+1,j})$
- $\mathcal{H}_{t+1}$: Set of piece-wise constant functions
- $a_{j,h,r}$: Points in the state space where piece-wise function h starts in dimension r
- $b_{j,h}$: Upper bounded estimate of $\hat{v}_{t+1,j}$

Formulation I

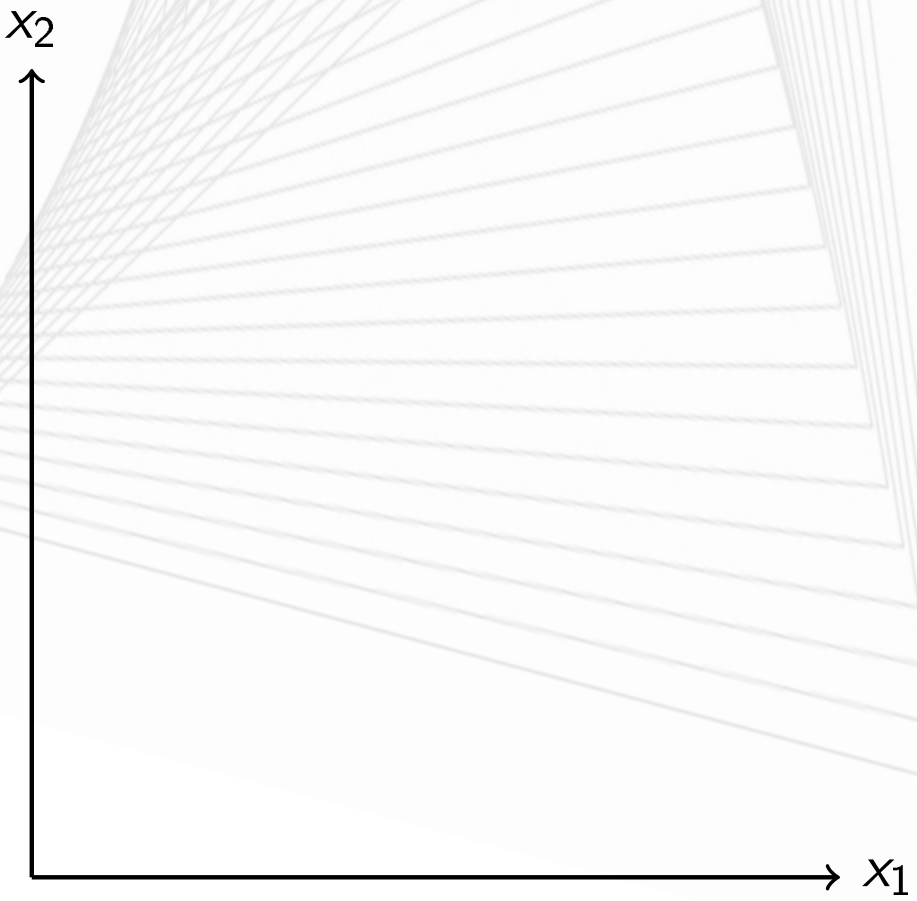
- $\hat{v}_{t+1,j}$: Variable that approximates the value function $V_{t+1,j}(x_{t+1,j})$
- $\mathcal{H}_{t+1}$: Set of piece-wise constant functions
- $a_{j,h,r}$: Points in the state space where piece-wise function h starts in dimension r
- $b_{j,h}$: Upper bounded estimate of $\hat{v}_{t+1,j}$
- Use series of binary variables & Big M constraints to couple $x_{t+1,j}$ to $\hat{v}_{t+1,j}$ using $a_{j,h,r}$ & $b_{j,h}$

Formulation II

- For period $t+1$ & price segment j:

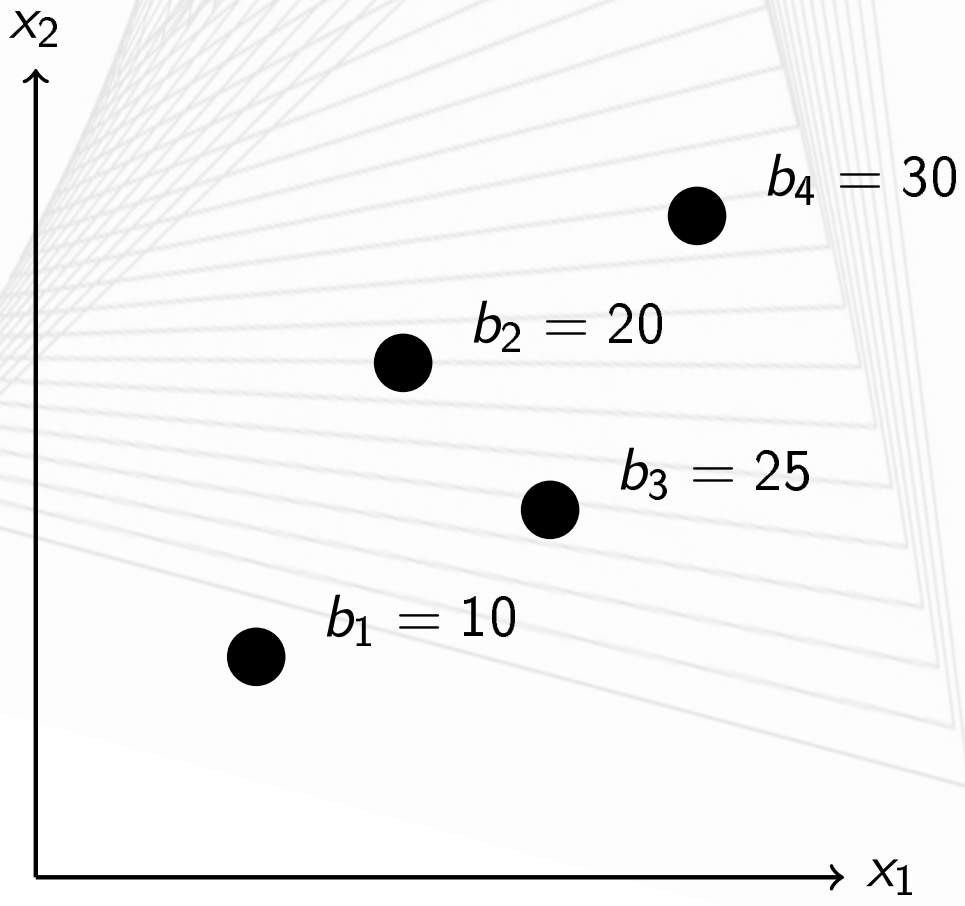
Formulation II

- For period $t+1$ & price segment j:



Formulation II

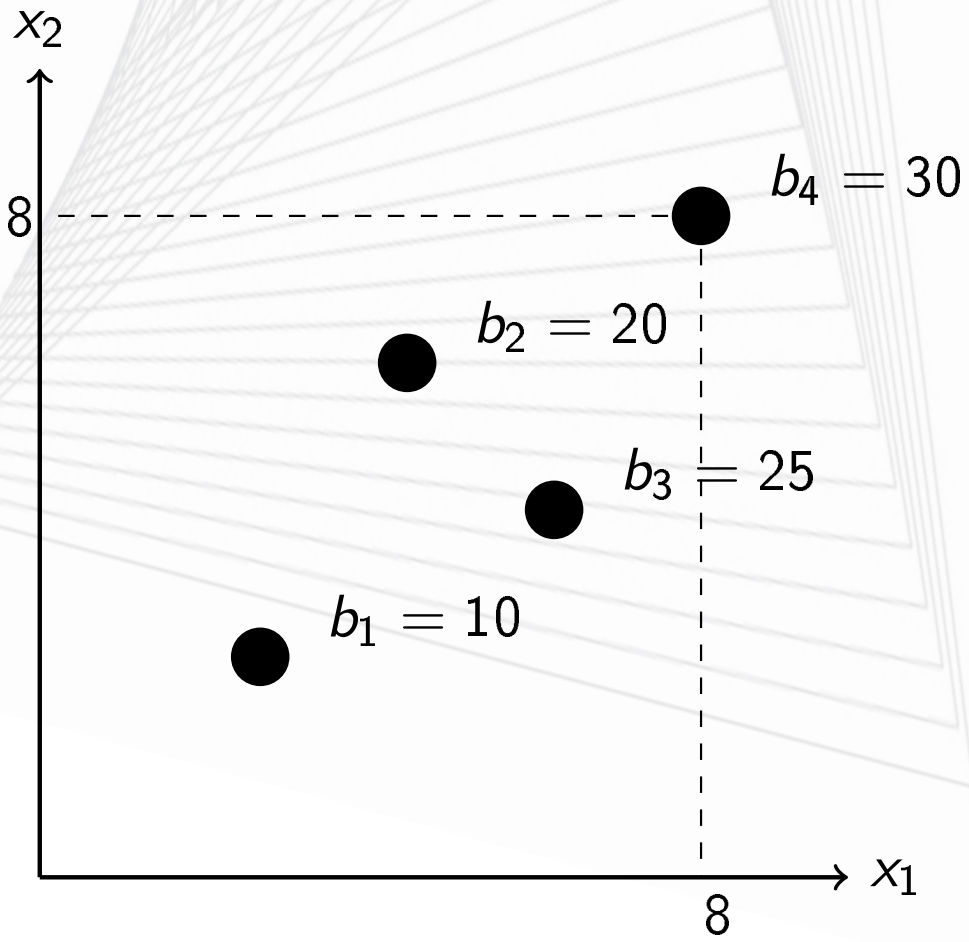
- For period $t+1$ & price segment j:



Formulation II

- For period $t+1$ & price segment j:

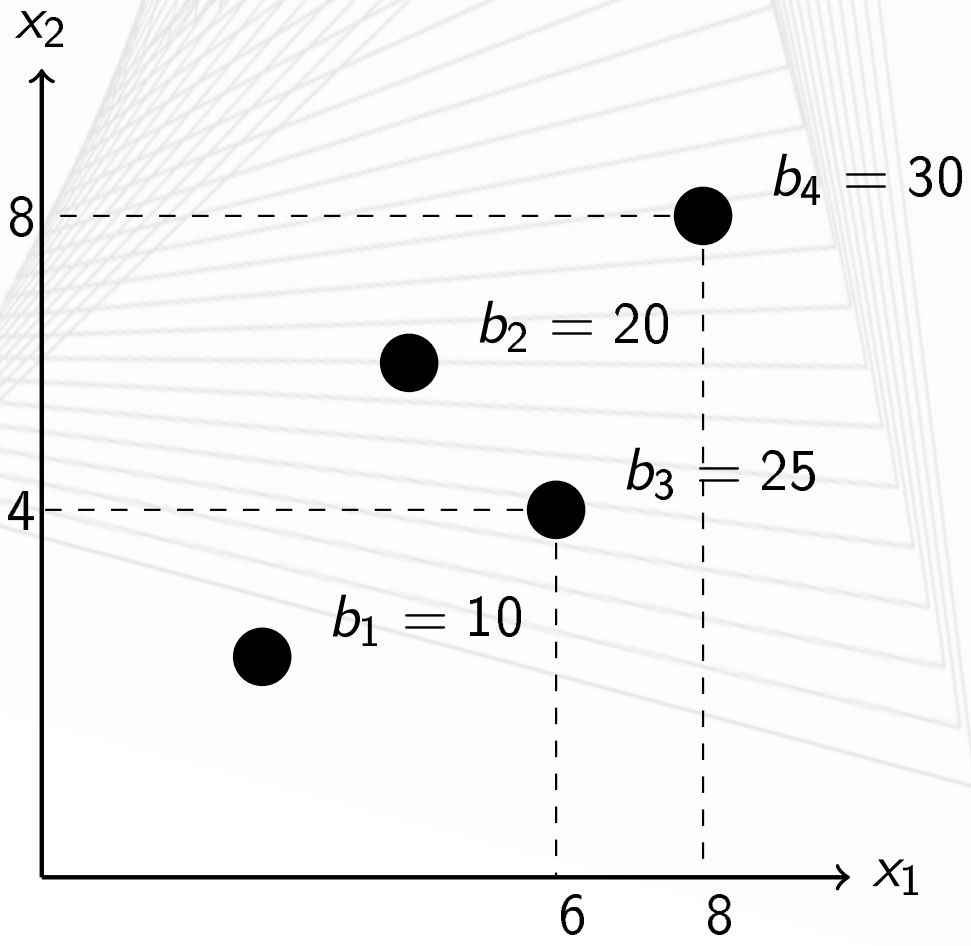
$$a_4 = (8, 8)$$



Formulation II

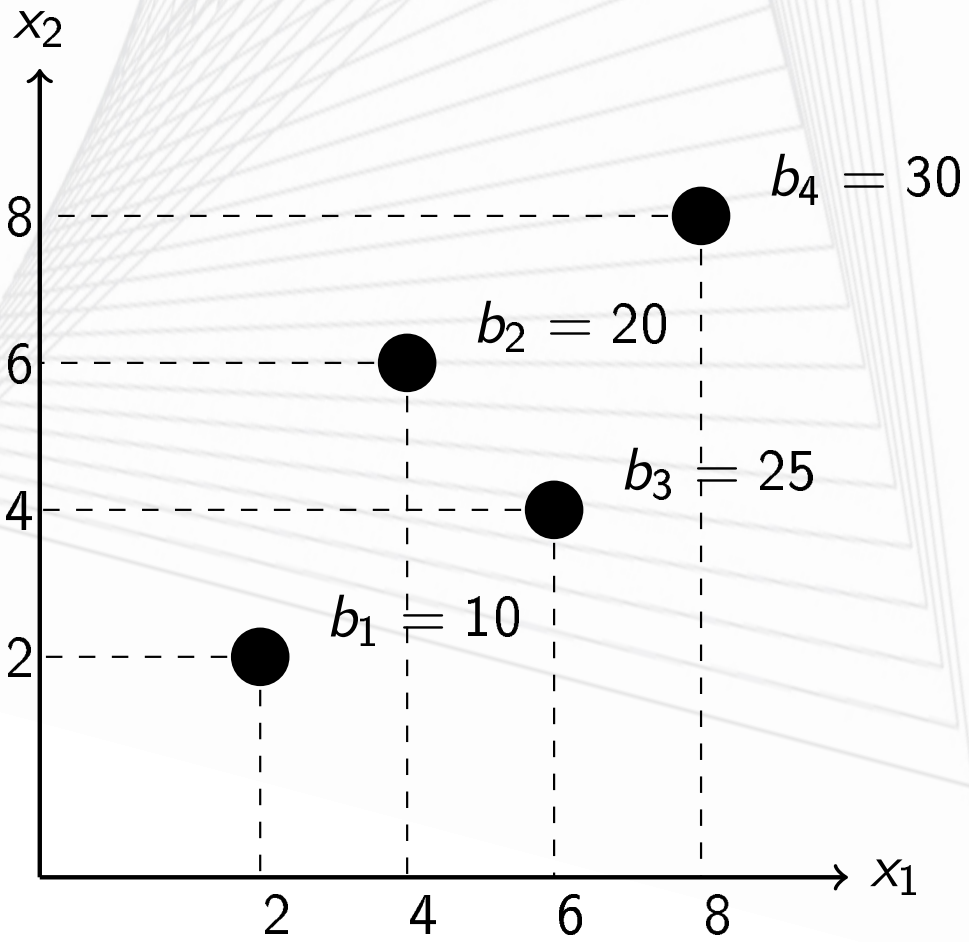
- For period $t+1$ & price segment j:

$$a_4 = (8, 8)$$
$$a_3 = (6, 4)$$



Formulation II

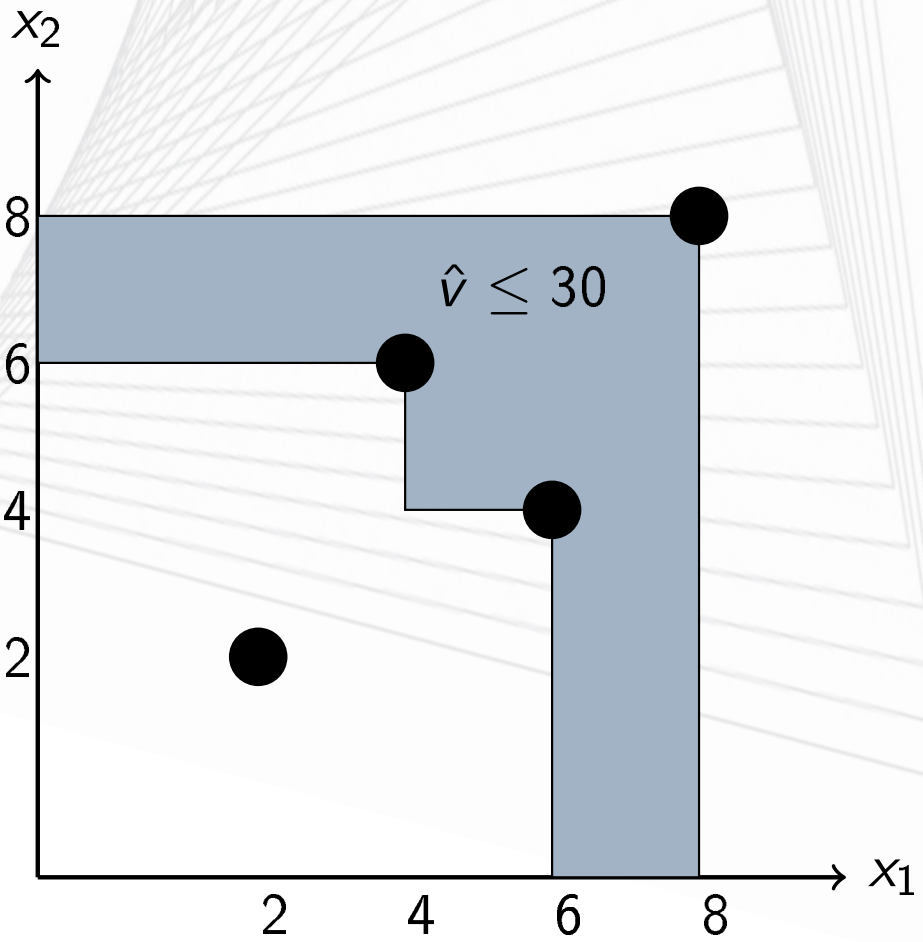
- For period $t+1$ & price segment j:



$$a_4 = (8, 8)$$
$$a_3 = (6, 4)$$
$$a_2 = (4, 6)$$
$$a_1 = (2, 2)$$

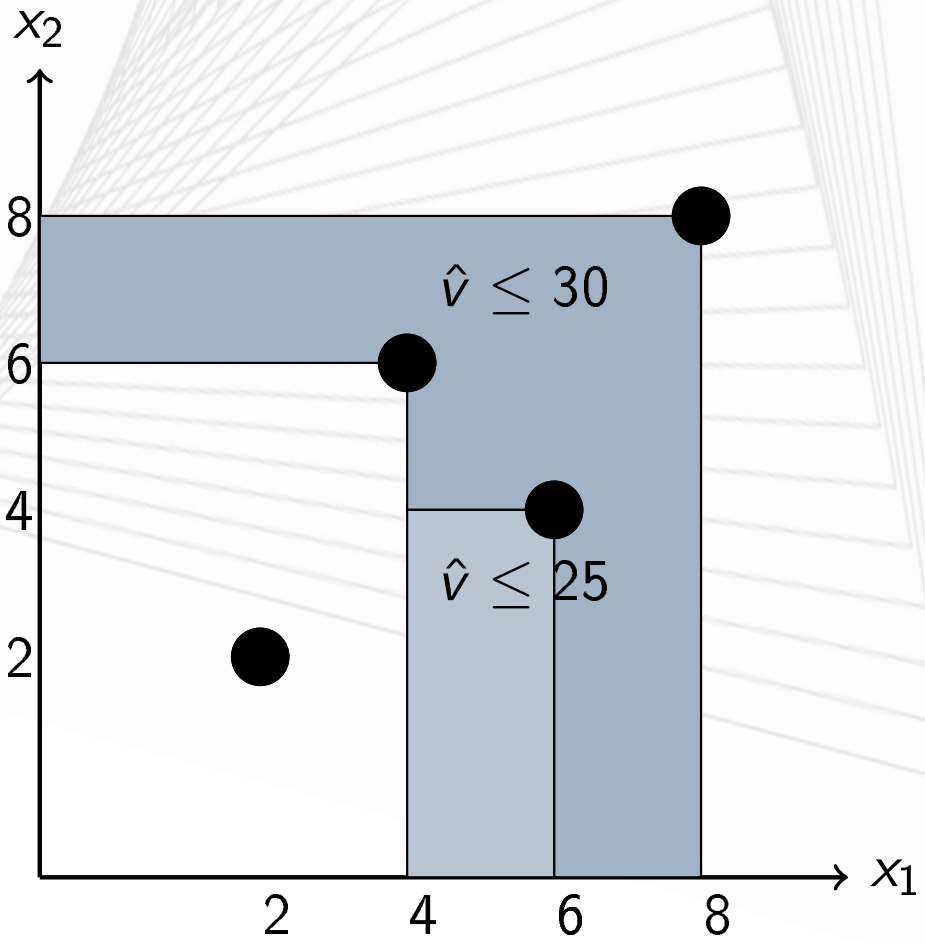
Formulation II

- For period $t+1$ & price segment j:



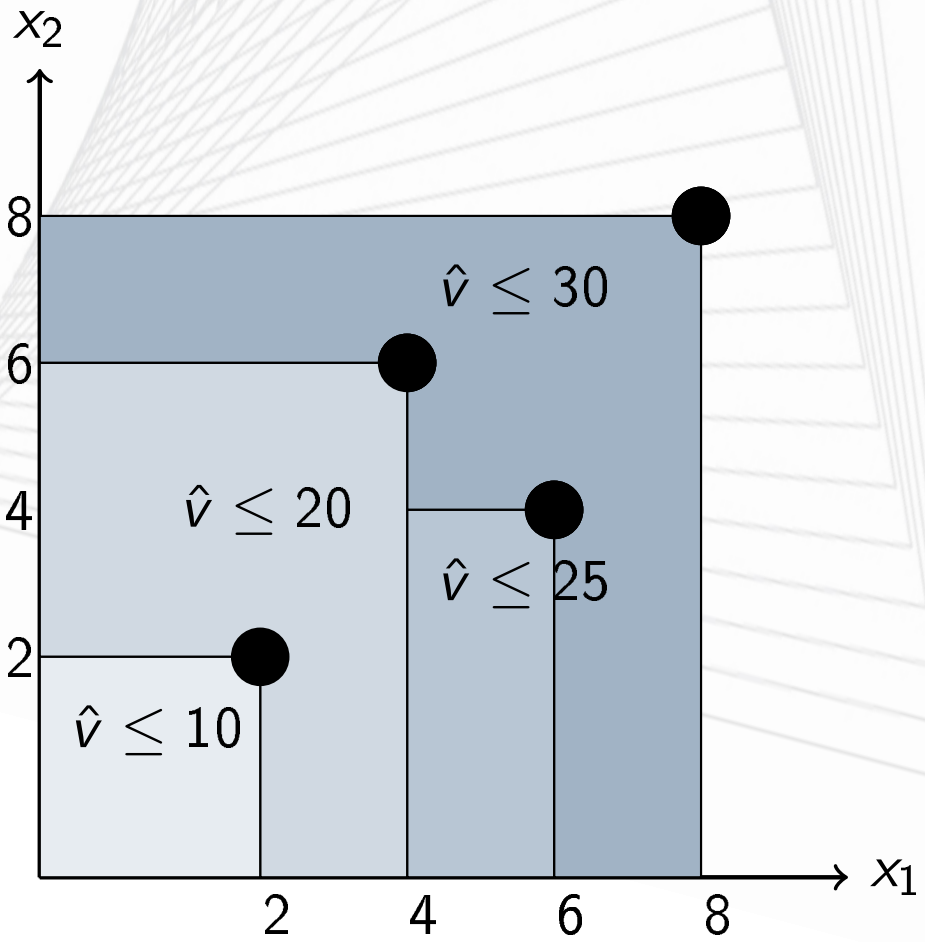
Formulation II

- For period $t+1$ & price segment j:



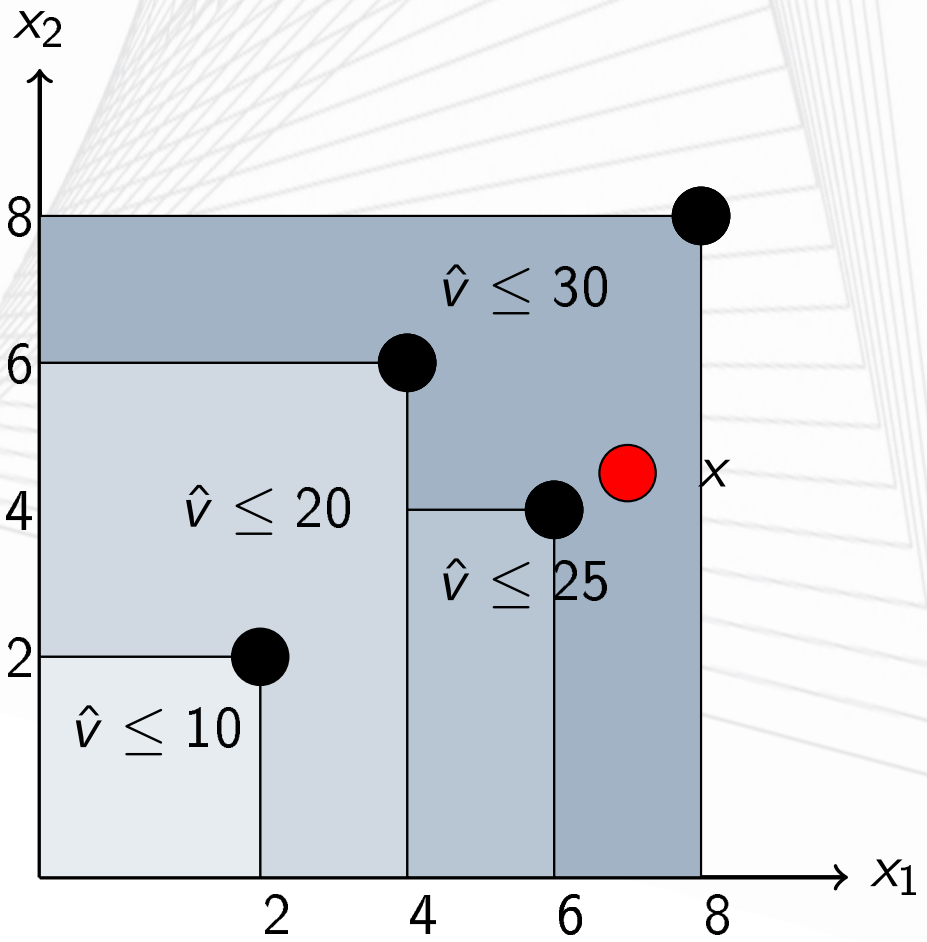
Formulation II

- For period $t+1$ & price segment j:



Formulation II

- For period $t+1$ & price segment j:



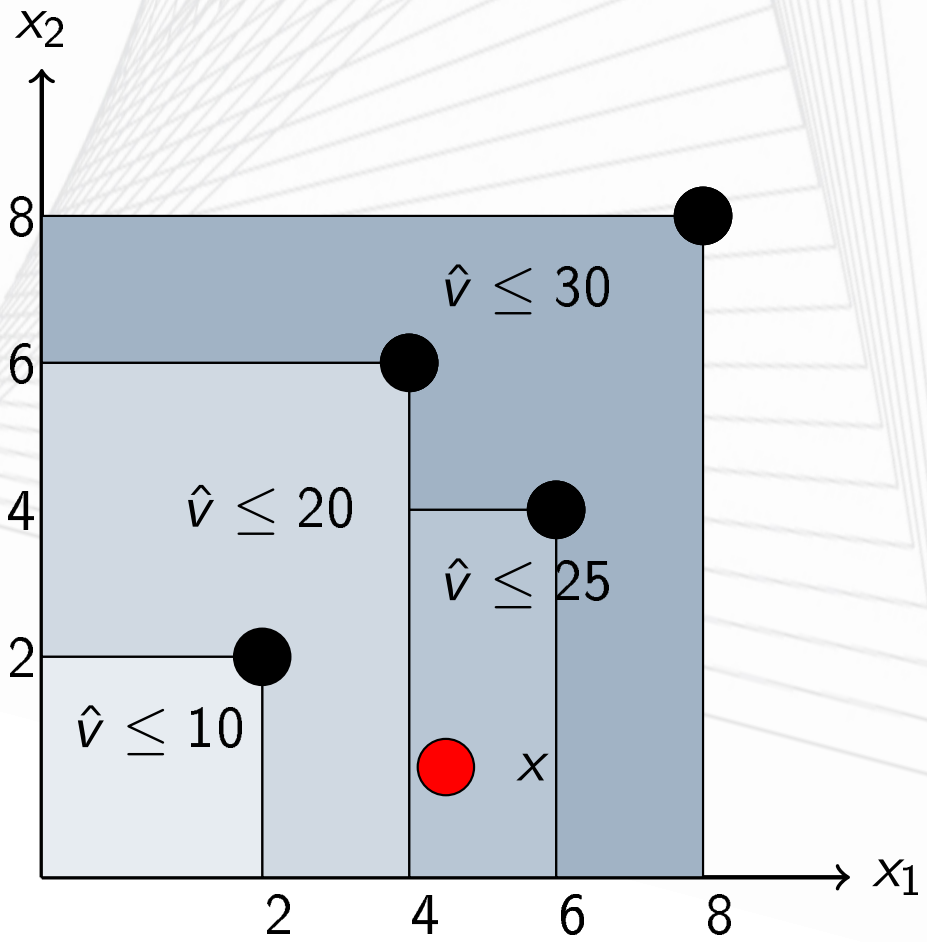
$\hat{v} \leq 30$

$x_1 > 6;$
Or $x_1 > 4$
Or $x_1 > 0$

$x_2 > 6;$
Or $x_2 > 4$
Or $x_2 > 0$

Formulation II

- For period $t+1$ & price segment j:



$\hat{v} \leq 30$;
And $\hat{v} \leq 25$

$x_1 > 4$

$x_2 > 0$

Description of the MIDAS algorithm

Description of the MIDAS algorithm

1. **Forward simulation:** for $t = 1, \ldots, T$

Description of the MIDAS algorithm

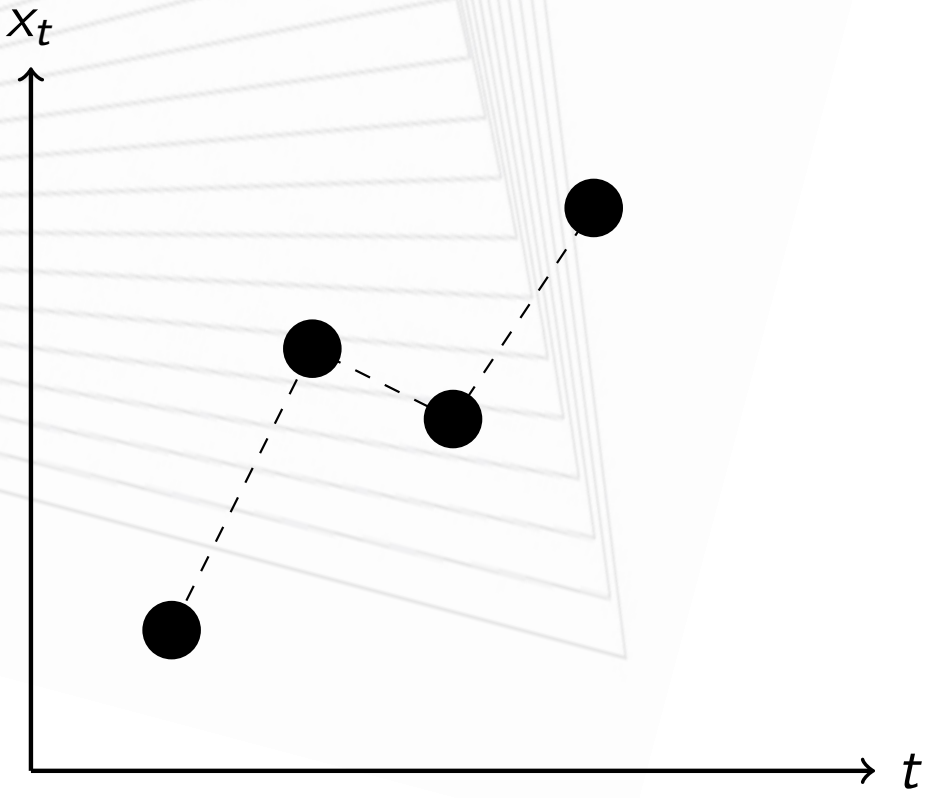
1. **Forward simulation:** for $t = 1, \ldots, T$
   1.1 Generates a scenario of market clearing prices

Description of the MIDAS algorithm

1. **Forward simulation:** for $t = 1, \ldots, T$
   1.1 Generates a scenario of market clearing prices
   1.2 Solves HERBS sub-problem & computes series of storage levels (state trajectory)

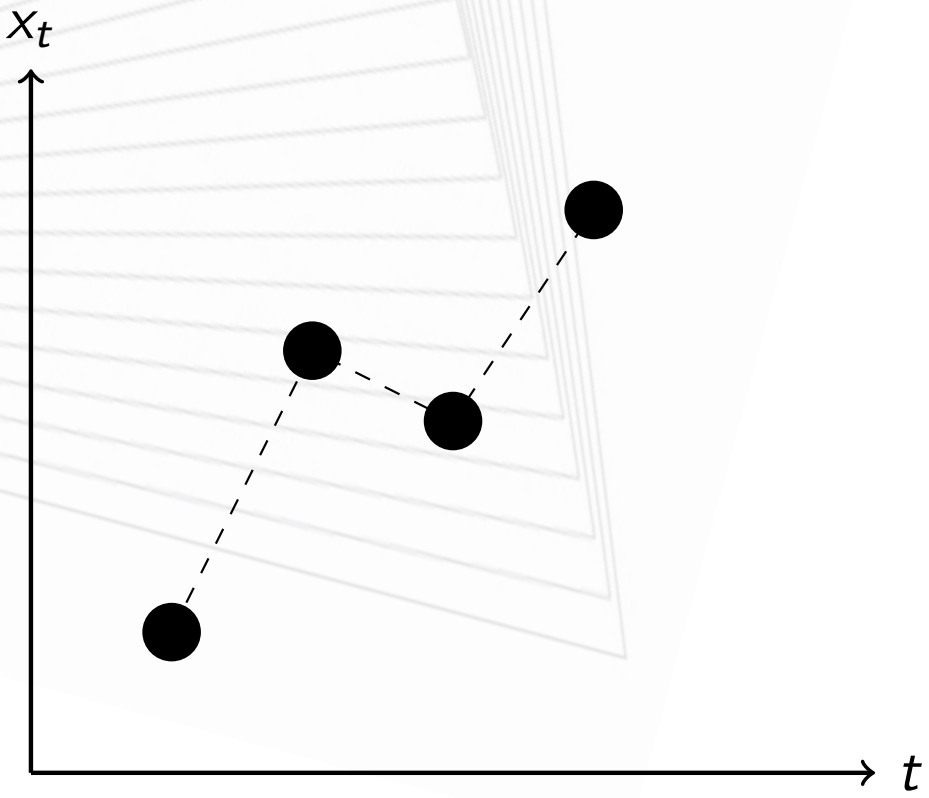
Description of the MIDAS algorithm

1. **Forward simulation:** for $t = 1, \ldots, T$
   1.1 Generates a scenario of market clearing prices
   1.2 Solves HERBS sub-problem & computes series of storage levels (state trajectory)



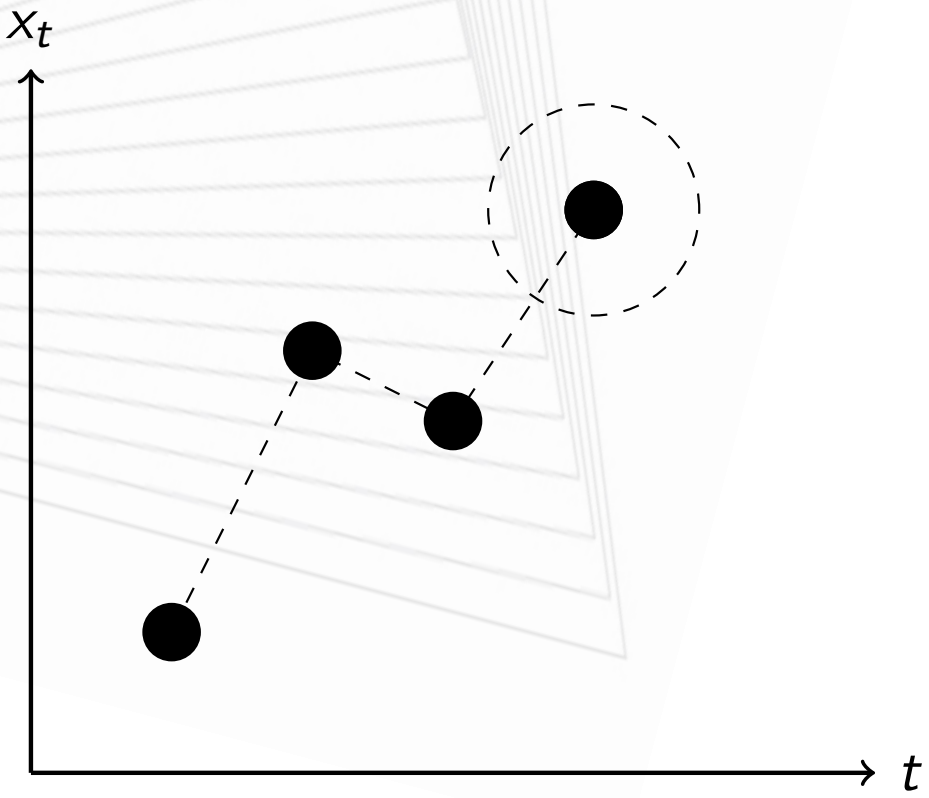
Description of the MIDAS algorithm

1. **Backward recursion:** for $t = T, \ldots, t$ & $m = 1, \ldots, M$



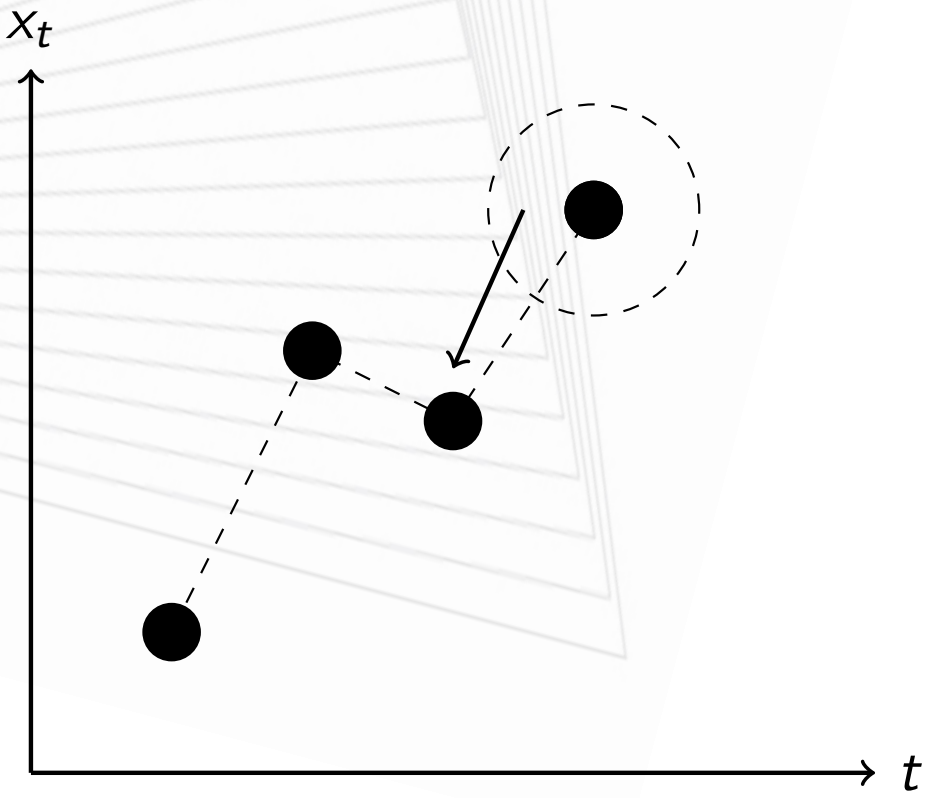
Description of the MIDAS algorithm

1. **Backward recursion:** for $t = T, \ldots, t$ & $m = 1, \ldots, M$
   1.1 Explores around the neighbourhood of state trajectory



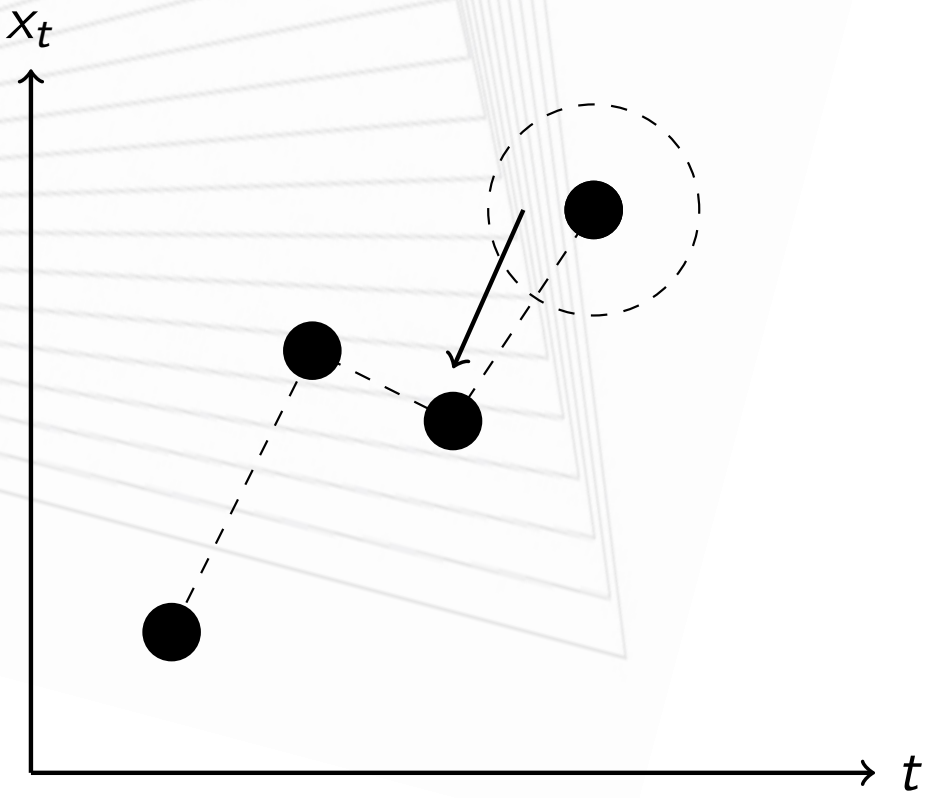
Description of the MIDAS algorithm

1. **Backward recursion:** for $t = T, \ldots, t$ & $m = 1, \ldots, M$
   1.1 Explores around the neighbourhood of state trajectory
   1.2 Computes $b_{j,h}$ & $a_{j,h}$ parameters



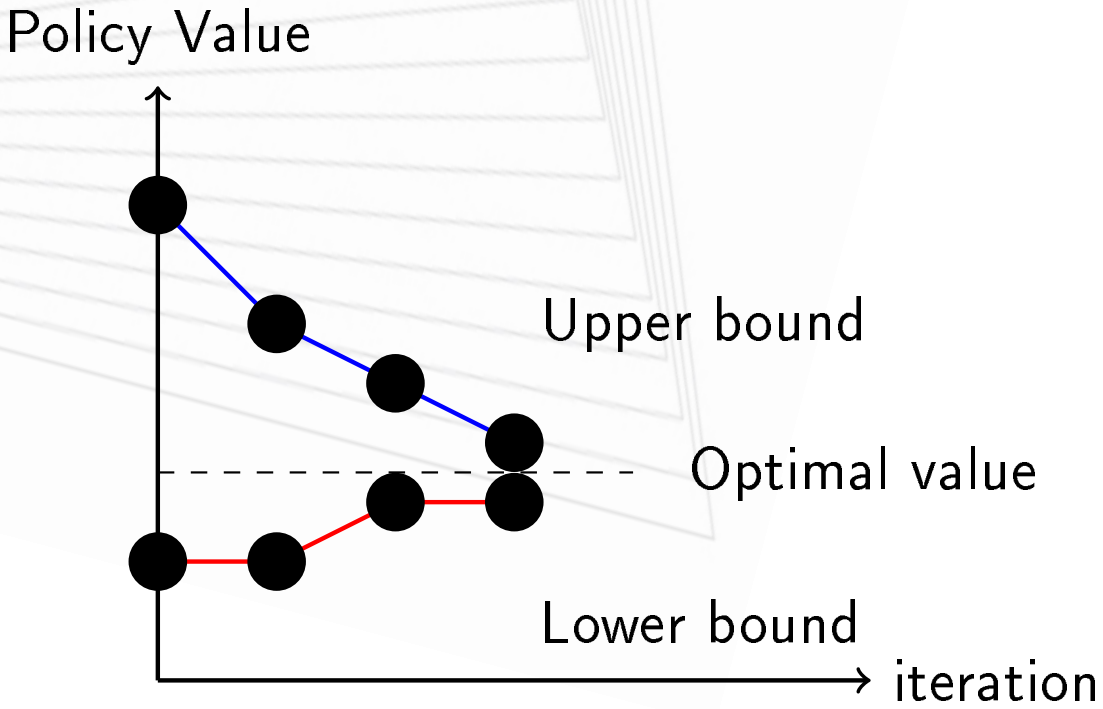
Description of the MIDAS algorithm

1. **Backward recursion:** for $t = T, \ldots, t$ & $m = 1, \ldots, M$
   1.1 Explores around the neighbourhood of state trajectory
   1.2 Computes $b_{j,h}$ & $a_{j,h}$ parameters
   1.3 Adds $(b_{j,h}, a_{j,h})$ to $\mathcal{H}_t$ to period $t-1$ sub-problem



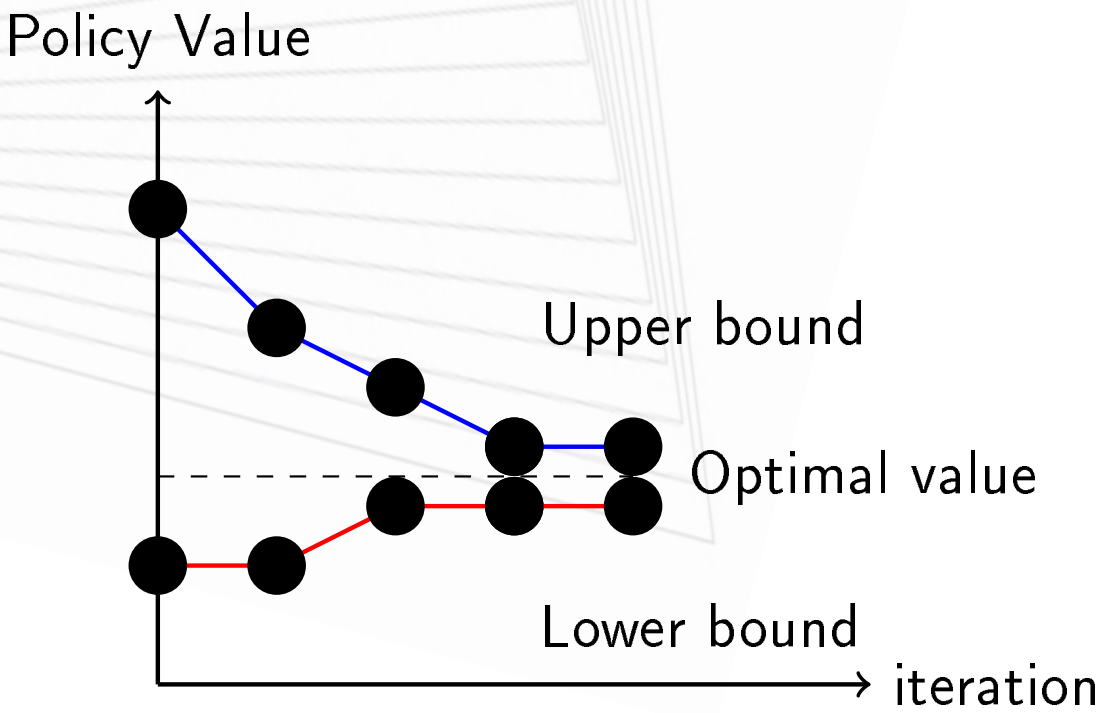
Description of the MIDAS algorithm

1. **Convergence testing:** After a certain number forward simulation & backward recursion



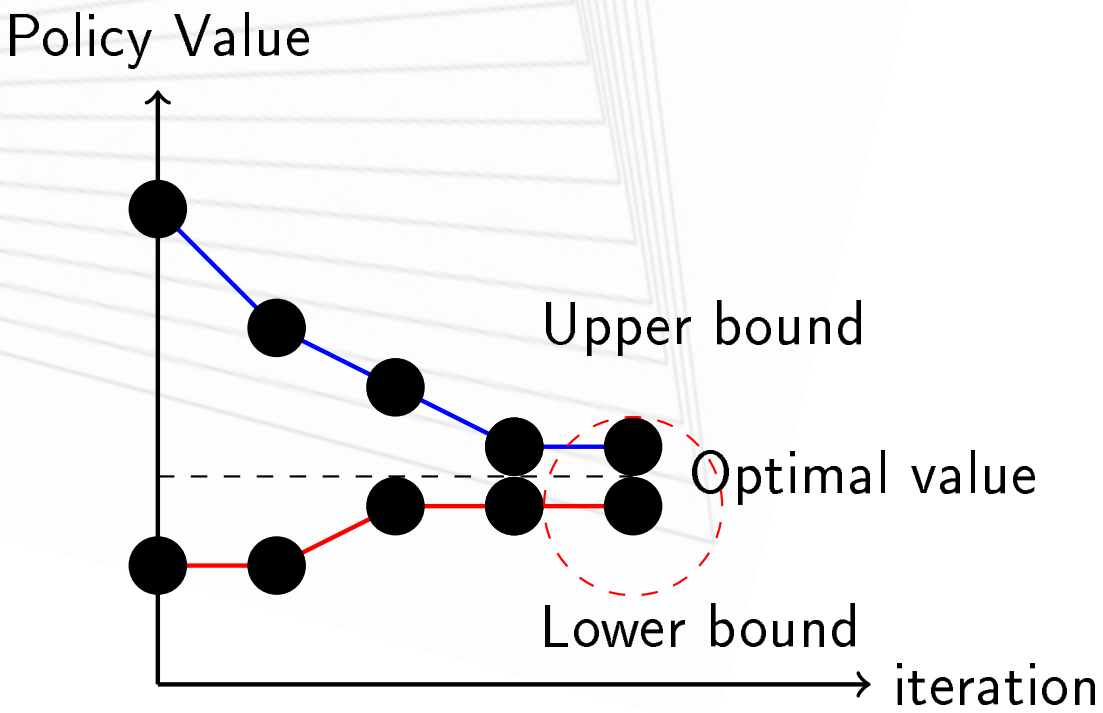
Description of the MIDAS algorithm

1. **Convergence testing:** After a certain number forward simulation & backward recursion
   1.1 Simulates policy over sample of price scenarios



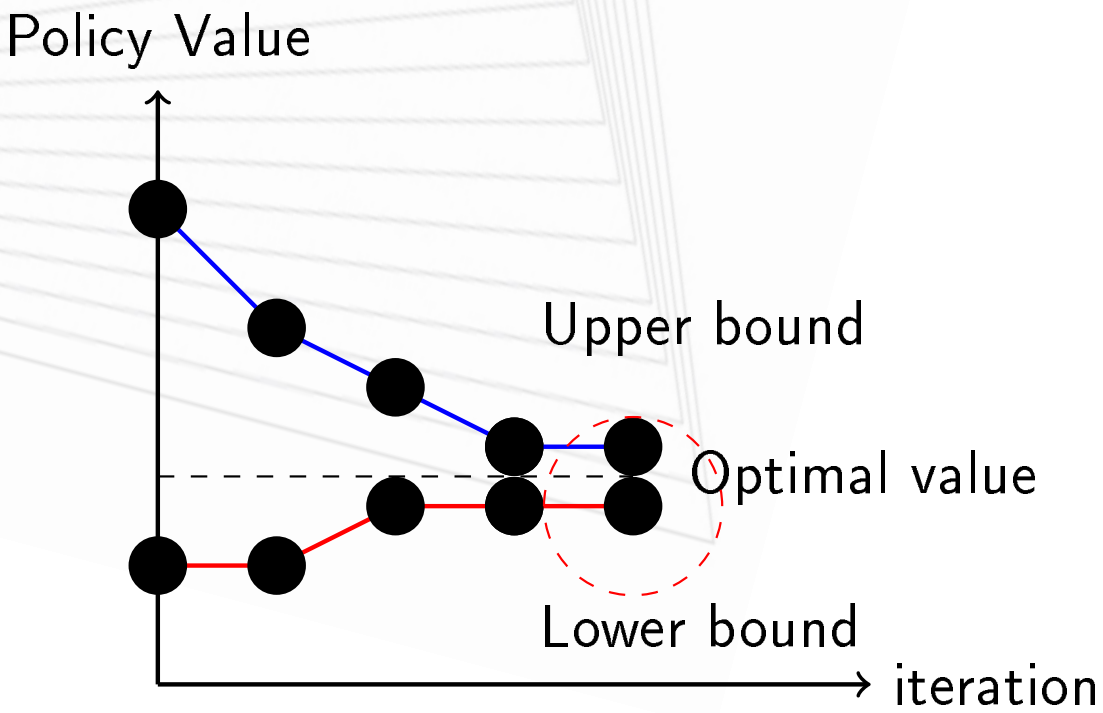
Description of the MIDAS algorithm

1. **Convergence testing:** After a certain number forward simulation & backward recursion
   1.1 Simulates policy over sample of price scenarios
   1.2 Compute confidence interval (i.e. Lower bound)



Description of the MIDAS algorithm

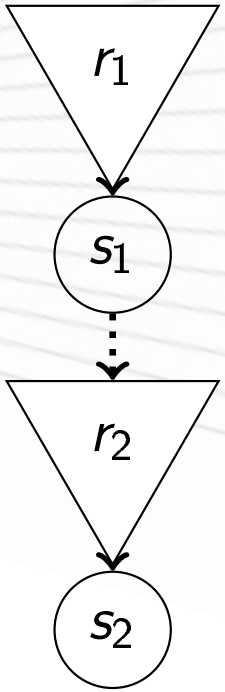
1. **Convergence testing:** After a certain number forward simulation & backward recursion
   1.1 Simulates policy over sample of price scenarios
   1.2 Compute confidence interval (i.e. Lower bound)
   1.3 Checks if first stage objective (i.e. Upper bound) is within confidence interval



Hydro-scheme data

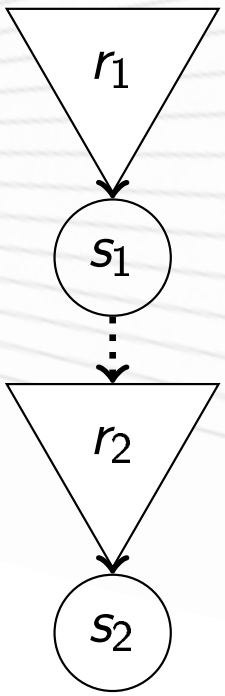
Hydro-scheme data

- Two identical, cascaded reservoirs



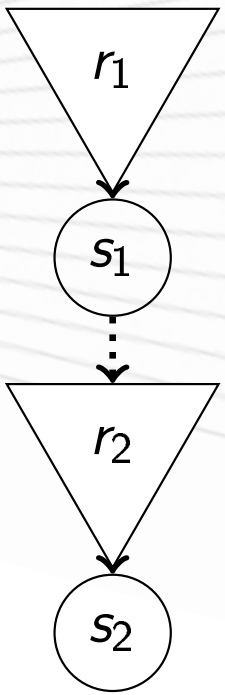
Hydro-scheme data

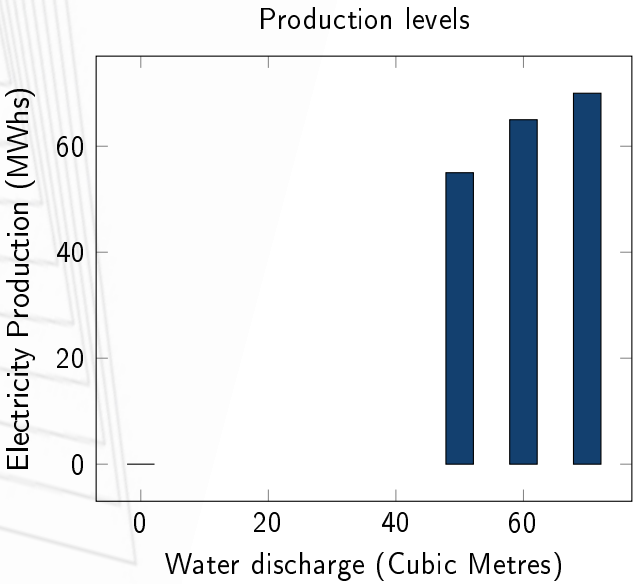
- Two identical, cascaded reservoirs
- $T = 4$ stages (boundary value function zero)
- $M = 3$ price segments OR 3 tranche offer stacks



Hydro-scheme data

- Two identical, cascaded reservoirs
- $T = 4$ stages (boundary value function zero)
- $M = 3$ price segments OR 3 tranche offer stacks





Comparison of policy value & approximation

Comparison of policy value & approximation

- Solved & simulated offer policy of MIDAS & SDDP for range of initial storage levels

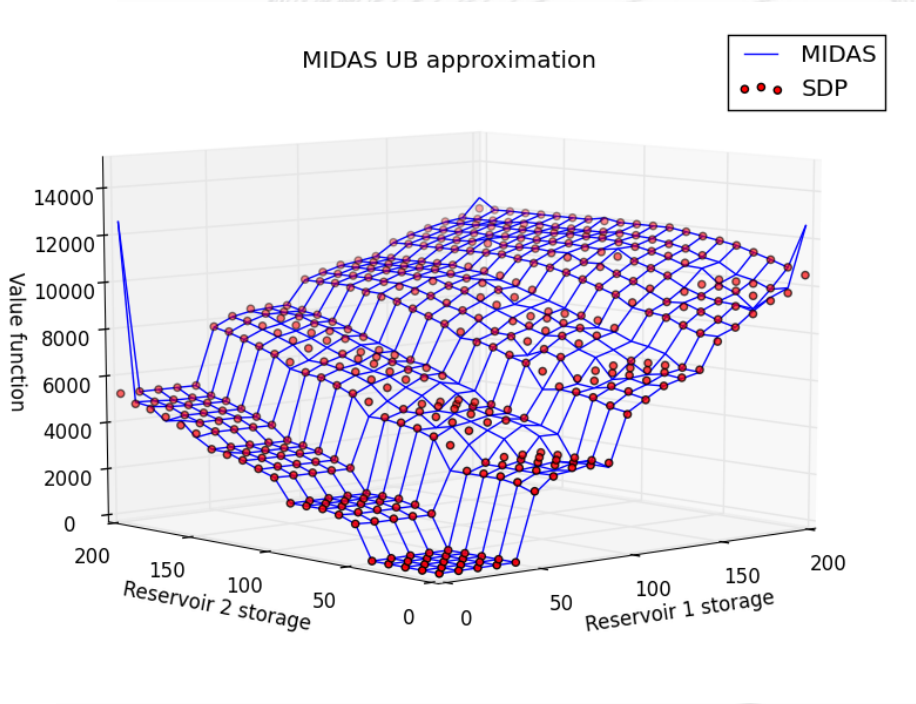
Comparison of policy value & approximation

- Solved & simulated offer policy of MIDAS & SDDP for range of initial storage levels
  - **MIDAS policy**: On average 98% of optimal value
  - **SDDP policy**: On average 93% of optimal value

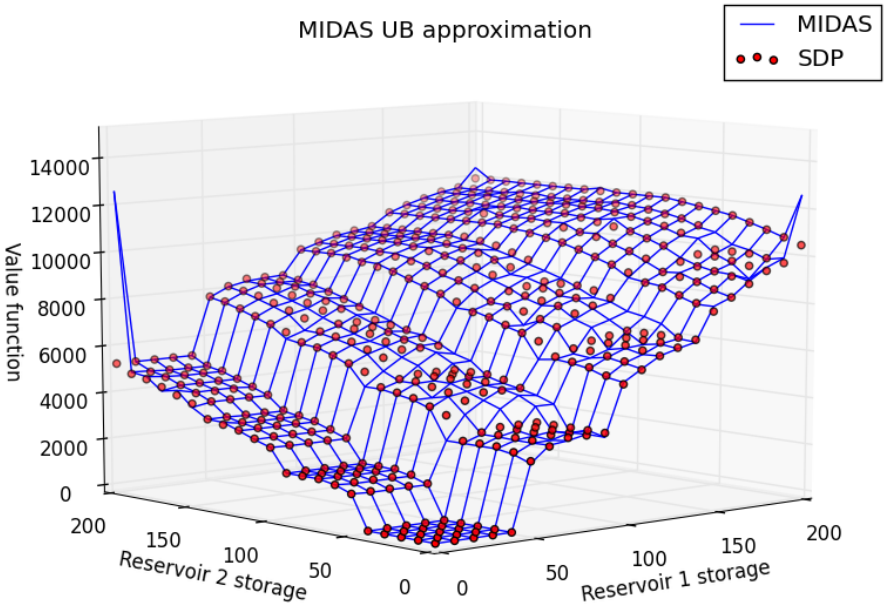
	Mean	**Median**	**Upper quartile**	**Lower quartile**
SDDP	92.92	88.22	81.18	96.35
MIDAS	97.93	99.06	97.18	99.67

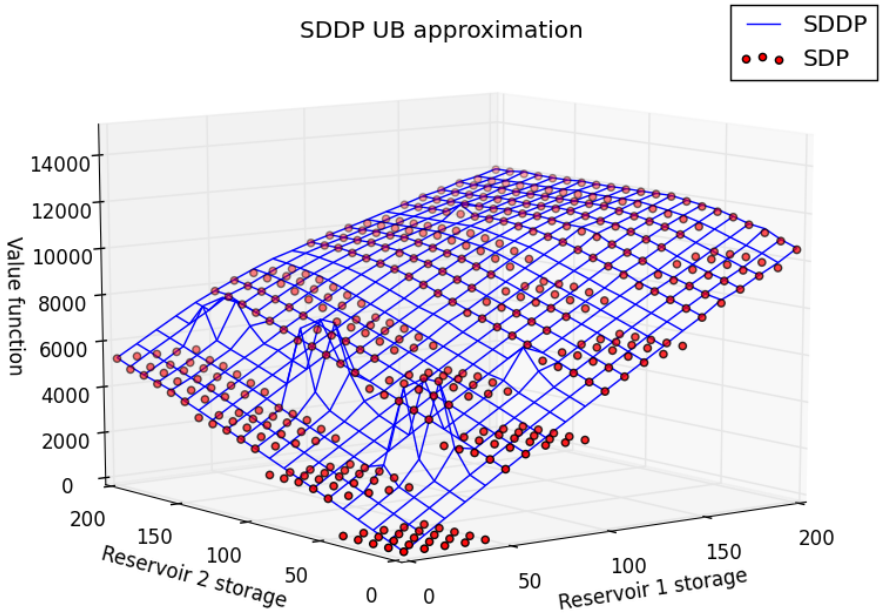
Approximation of value function

Approximation of value function



Approximation of value function





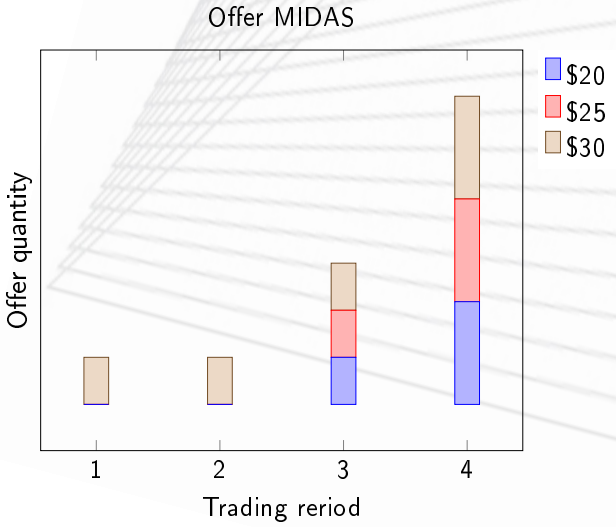
Comparison of an offer

Comparison of an offer

- **Price scenario:** $\$20 \rightarrow \$20 \rightarrow \$25 \rightarrow \$20$
- **Storage:** $x_{t,i} = (50, 60)$

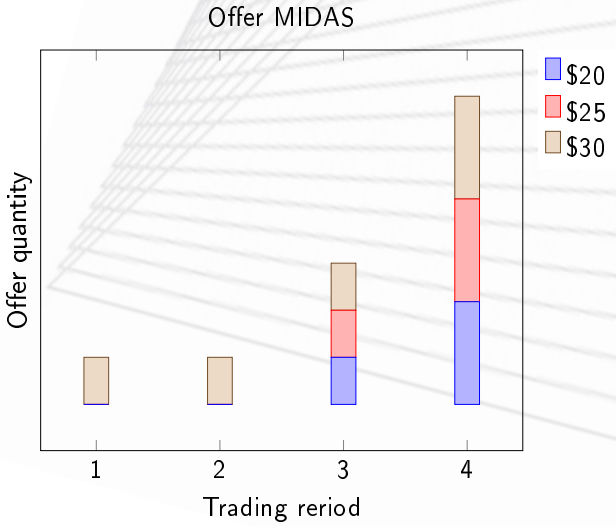
Comparison of an offer

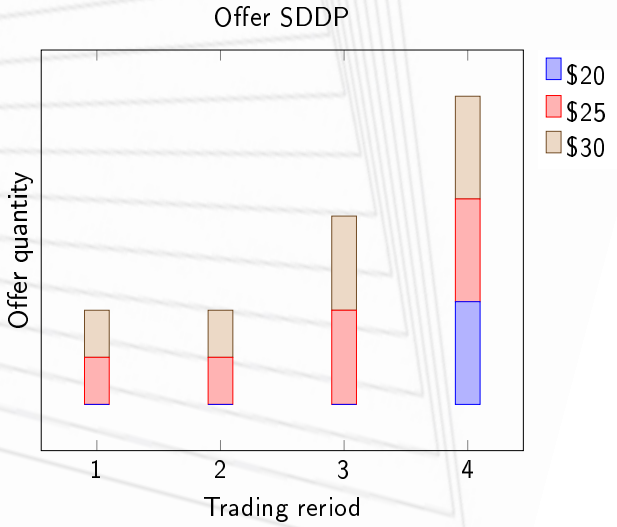
- **Price scenario:** $\$20 \rightarrow \$20 \rightarrow \$25 \rightarrow \$20$
- **Storage:** $x_{t,i} = (50, 60)$



Comparison of an offer

- **Price scenario:** $\$20 \rightarrow \$20 \rightarrow \$25 \rightarrow \$20$
- **Storage:** $x_{t,i} = (50, 60)$





Comparison of an offer

- **Price scenario:** $\$20 \rightarrow \$20 \rightarrow \$25 \rightarrow \$20$
- **Storage:** $x_{t,i} = (50, 60)$

Comparison of computation I

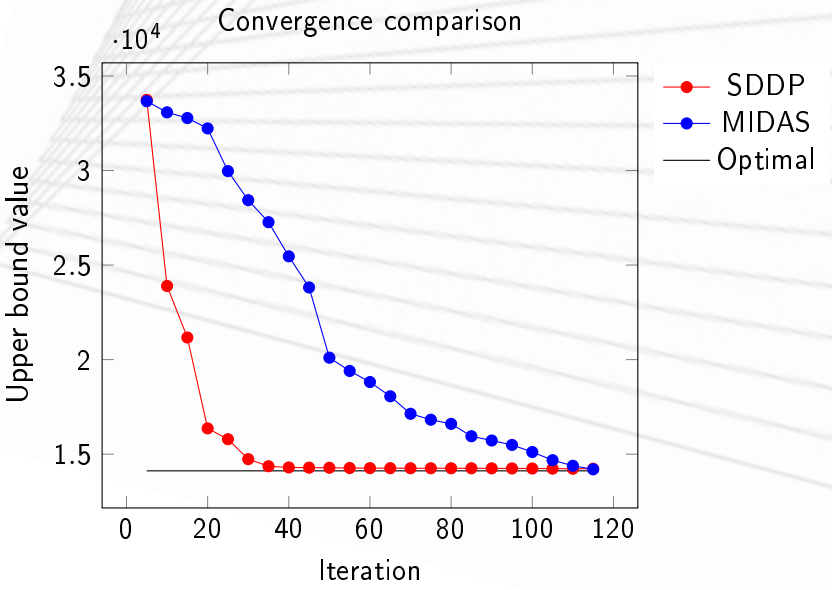
- Extend the number of periods to 10

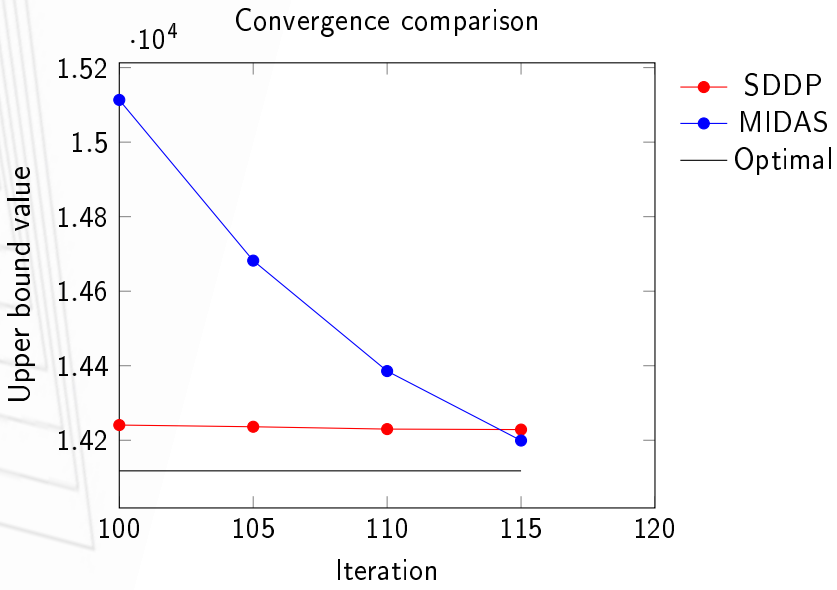
Comparison of computation I

- Extend the number of periods to 10
- **High storage:** $x_{t,i} = (200, 90)$

Comparison of computation I

- Extend the number of periods to 10
- **High storage:** $x_{t,i} = (200, 90)$
- MIDAS takes more iterations to get close to optimal value





Comparison of computation II

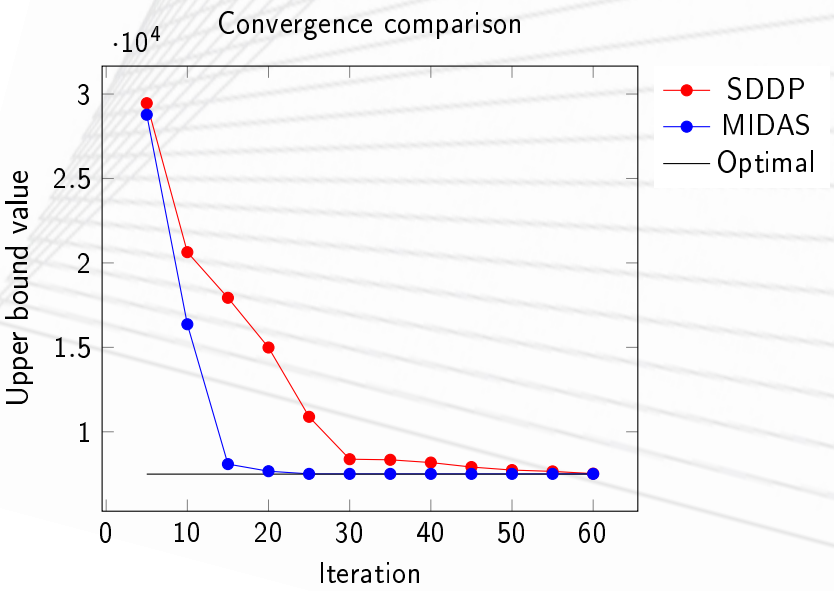
- **Low storage:** $x_{t,i} = (100, 50)$

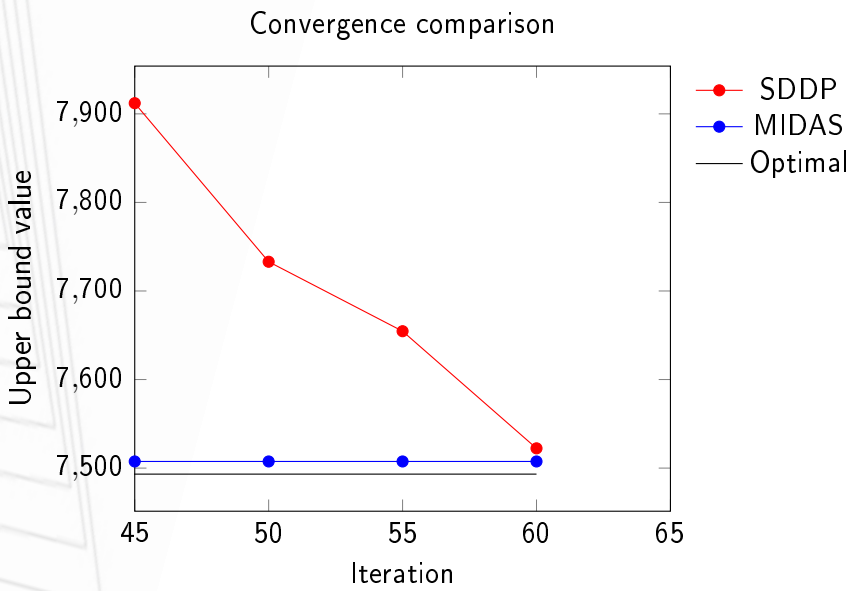
Comparison of computation II

- **Low storage:** $x_{t,i} = (100, 50)$
- SDDP takes more iterations to get close to optimal value

Comparison of computation II

- **Low storage:** $x_{t,i} = (100, 50)$
- SDDP takes more iterations to get close to optimal value





Comparison of computation III

Comparison of computation III

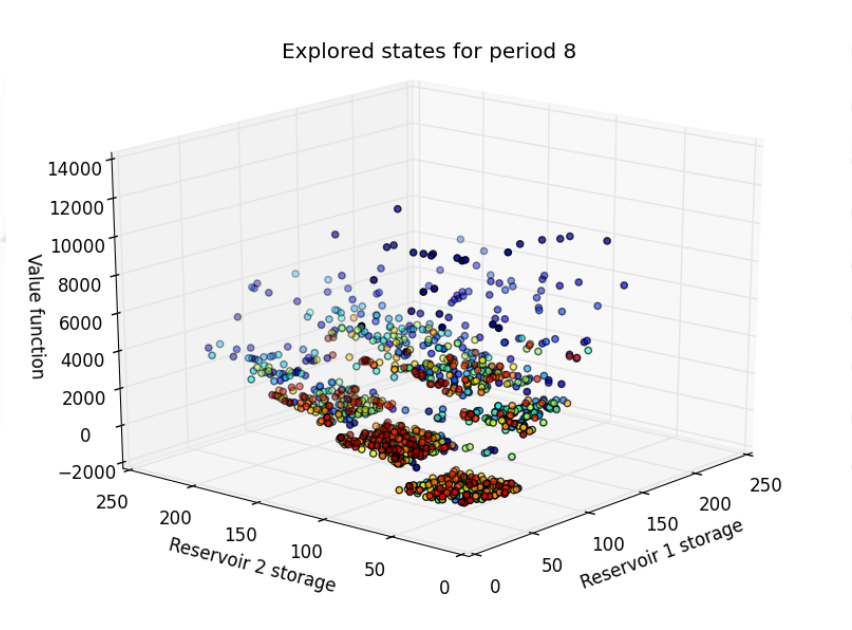
- Value function at high storage levels has convexity-like structure

Comparison of computation III

- Value function at high storage levels has convexity-like structure
- MIDAS not able to exploit the gradient of the value function (unlike SDDP)

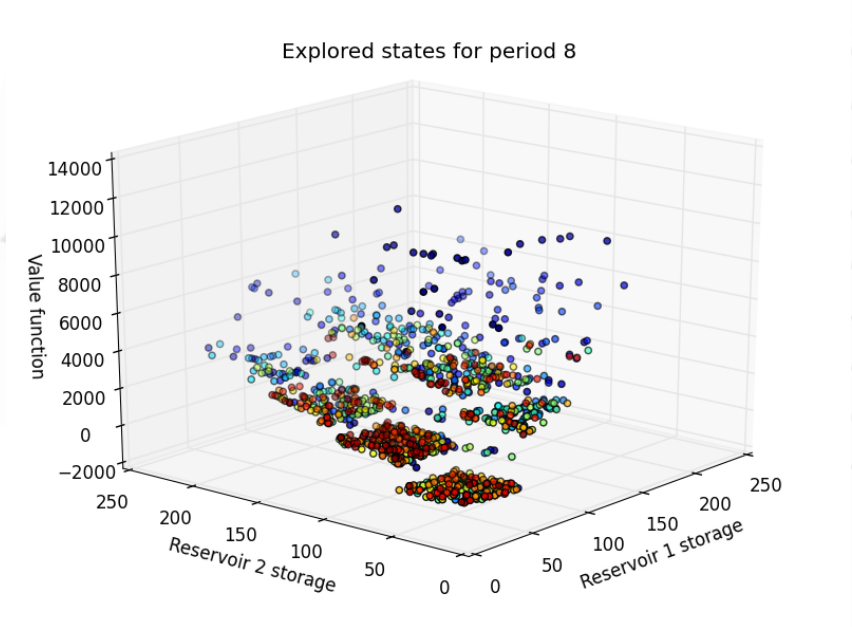
Comparison of computation III

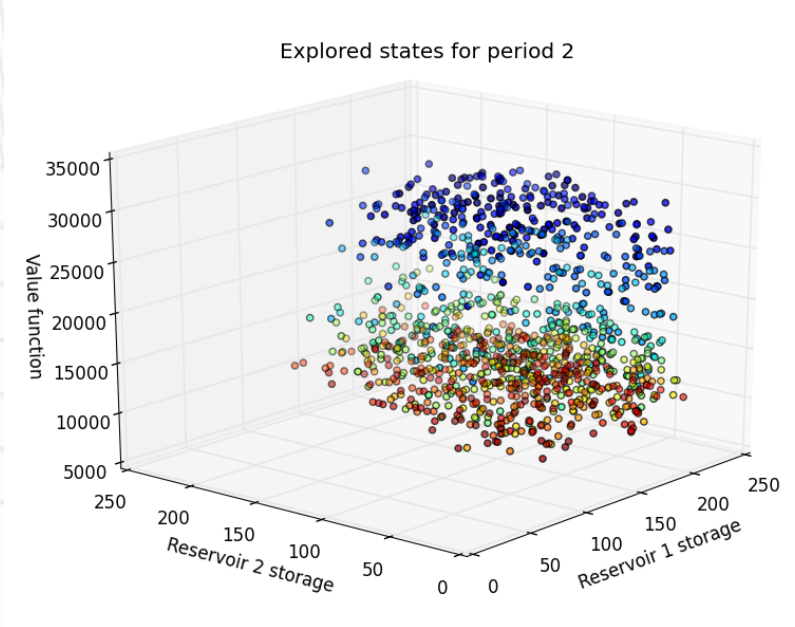
- Value function at high storage levels has convexity-like structure
- MIDAS not able to exploit the gradient of the value function (unlike SDDP)



Comparison of computation III

- Value function at high storage levels has convexity-like structure
- MIDAS not able to exploit the gradient of the value function (unlike SDDP)





Conclusion

Conclusion

- MIDAS has better representation of the structure of future value function compared to SDDP

Conclusion

- MIDAS has better representation of the structure of future value function compared to SDDP
- MIDAS produces good policy and accurate value function approximation
  - 98% of optimal value

Conclusion

- MIDAS has better representation of the structure of future value function compared to SDDP
- MIDAS produces good policy and accurate value function approximation
  - 98% of optimal value
- For high storage MIDAS takes many iterations to get close to optimal value :

Conclusion

- MIDAS has better representation of the structure of future value function compared to SDDP
- MIDAS produces good policy and accurate value function approximation
  - 98% of optimal value
- For high storage MIDAS takes many iterations to get close to optimal value :
  - Not able to exploit the gradient of the value function (unlike SDDP)

Conclusion

- MIDAS has better representation of the structure of future value function compared to SDDP
- MIDAS produces good policy and accurate value function approximation
  - 98% of optimal value
- For high storage MIDAS takes many iterations to get close to optimal value :
  - Not able to exploit the gradient of the value function (unlike SDDP)
  - Solving increasingly large MIPs with iterations

Conclusion

- MIDAS has better representation of the structure of future value function compared to SDDP
- MIDAS produces good policy and accurate value function approximation
  - 98% of optimal value
- For high storage MIDAS takes many iterations to get close to optimal value :
  - Not able to exploit the gradient of the value function (unlike SDDP)
  - Solving increasingly large MIPs with iterations
  - Solving MIPs with many Big M constraints & tight integer feasibility tolerance

Questions

Thank you for listening!

Questions?

References I

[1] Pritchard, G. and Zakeri, G. *Market offering strategies for hydroelectric generators. Operations Research,* 51(4):602–612, 2003.

[2] Löhndorf, N., Wozabal, D. and Minner, S. *Optimizing Trading Decisions for Hydro Storage Systems Using Approximate Dual Dynamic Programming. INFORMS,* 61(4):801-823, 2013.

[3] Pereira M.V.F. and Pinto L.M.V.G. *Multi-stage stochastic optimization applied to energy planning. Mathematical Programming,* 52(1):359–375, 1991.

[4] Boomsma, T.K., Juul N. and Fleten S.E. *Bidding in sequential electricity markets: The Nordic cas European Journal of of Operation Research,* 238(3):797–809, 2014.

References II

[1] Fleten, S.E. and Kristoffersen, T.K. *Stochastic programming for optimizing bidding strategies of a Nordic hydropower producer. European Journal of Operational Research,* 181(2):916–928, 2007.

[2] Braathen, J and Eriksrud, A.L. *Hydropower Bidding Using Linear Decision Rules. Institutt for industriell økonomi og teknologiledelse,* 2013.