Distributed Stochastic Programming

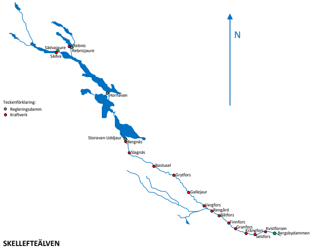
with Applications to Large-Scale Hydropower Operations

Martin Biel

KTH - Royal Institute of Technology

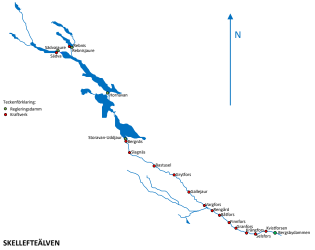
Doctoral thesis, December 3, 2021

Motivation - Hydropower operations

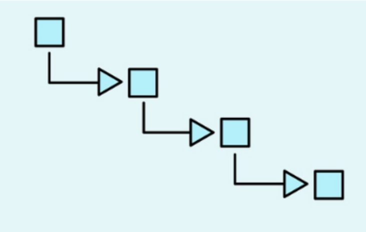


- Hydroelectric power production

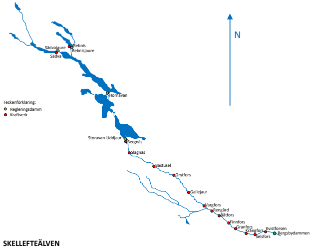
Motivation - Hydropower operations



- Hydroelectric power production
- Spatial dependence

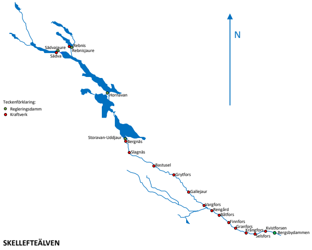


Motivation - Hydropower operations

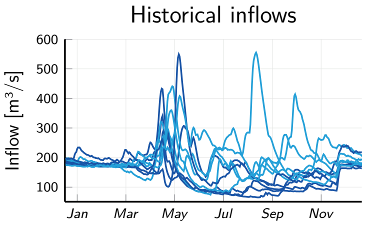


- Hydroelectric power production
- Spatial dependence
- Temporal dependence

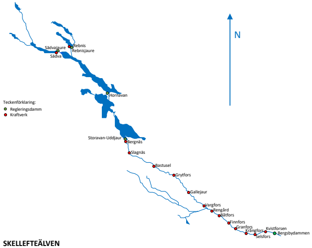
Motivation - Hydropower operations



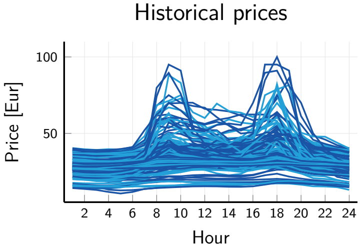
- Uncertain local inflow



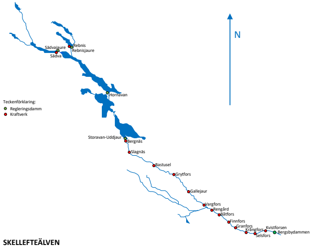
Motivation - Hydropower operations



- Uncertain local inflow
- Uncertain electricity price

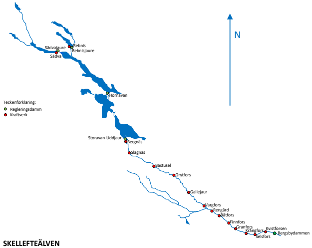


Motivation - Hydropower operations



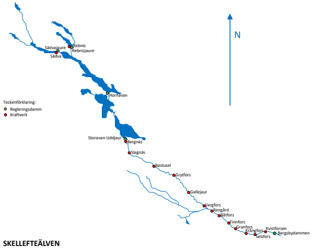
- Uncertain local inflow
- Uncertain electricity price
- Uncertain renewable production

Motivation - Hydropower operations

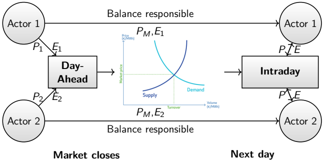


- Store energy in water reservoirs

Motivation - Hydropower operations



- Participate in electricity market



Motivation - Optimization models

- Decision support: formulate and solve optimization models

Motivation - Optimization models

- Decision support: formulate and solve optimization models
- Common: trade-off between accuracy and computation time

Motivation - Optimization models

- Decision support: formulate and solve optimization models
- Common: trade-off between accuracy and computation time
- Aim: **provide reliable decision-support in a short amount of time**

Motivation - Optimization models

- Decision support: formulate and solve optimization models
- Common: trade-off between accuracy and computation time
- Aim: **provide reliable decision-support in a short amount of time**
  - Accurate models: optimal model reductions
  - Fast computations: scalable algorithms on commodity hardware

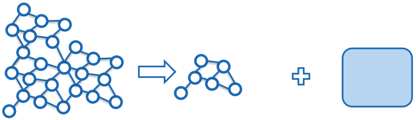


Figure: Manageable models

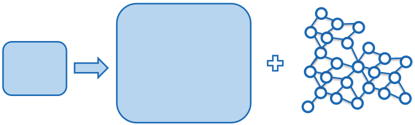


Figure: Scalable algorithms

Motivation - Stochastic programming

Mathematical framework for decision problems subjected to uncertainty

Motivation - Stochastic programming

Mathematical framework for decision problems subjected to uncertainty

Decision

Actions

- Investments
- Schedules
- Orders

Motivation - Stochastic programming

Mathematical framework for decision problems subjected to uncertainty

Decision $\longrightarrow$ **Observation**

Actions	**Uncertainties**
• Investments	• Demand
• Schedules	• Weather conditions
• Orders	• Market price

Motivation - Stochastic programming

Mathematical framework for decision problems subjected to uncertainty

Decision $\longrightarrow$ **Observation** $\longrightarrow$ **Recourse**

Actions

- Investments
- Schedules
- Orders

Uncertainties

- Demand
- Weather conditions
- Market price

Actions

- Restock
- Reschedule
- Settle imbalances

Motivation - Stochastic programming

Stochastic programming for hydropower operations

- Order strategies in deregulated electricity markets
- Maintenance scheduling
- Capacity expansion
- Coordination with renewable production
- Seasonal planning: reservoir contents before spring flood

Motivation - Stochastic programming

Stochastic programming for hydropower operations

- **Order strategies in deregulated electricity markets**
- **Maintenance scheduling**
- **Capacity expansion**
- Coordination with renewable production
- Seasonal planning: reservoir contents before spring flood

Contribution

- `StochasticPrograms.jl`: framework for stochastic programming

Contribution

- `StochasticPrograms.jl`: framework for stochastic programming
- Distributed stochastic programming for large-scale models

Contribution

- `StochasticPrograms.jl`: framework for stochastic programming
- Distributed stochastic programming for large-scale models
- Efficient implementations of structure-exploiting algorithms

Contribution

- `StochasticPrograms.jl`: framework for stochastic programming
- Distributed stochastic programming for large-scale models
- Efficient implementations of structure-exploiting algorithms
- Algorithmic innovations and software patterns

Contribution

- `StochasticPrograms.jl`: framework for stochastic programming
- Distributed stochastic programming for large-scale models
- Efficient implementations of structure-exploiting algorithms
- Algorithmic innovations and software patterns
- Detailed consideration of three hydropower problems
  - Large-scale models of power stations in Skellefteälven
  - Distributed and solved on 32 workers using `StochasticPrograms.jl`
  - Statistically significant gain from stochastic planning

Outline

Outline

Preliminaries - Stochastic programming

- First-stage decision: x

Preliminaries - Stochastic programming

- First-stage decision: x
- Recourse decision: y

Preliminaries - Stochastic programming

- First-stage decision: x
- Recourse decision: y
- Uncertainty: $\xi(\omega) : \Omega \to \mathbb{R}^M$ random variable on the set of events Ω

Preliminaries - Stochastic programming

- First-stage decision: x
- Recourse decision: y
- Uncertainty: $\xi(\omega) : \Omega \to \mathbb{R}^M$ random variable on the set of events Ω

First stage

$$\begin{aligned} \underset{x \in \mathbb{R}^n}{\text{minimize}} \quad & c^T x + \mathbb{E}_\xi[Q(x, \xi(\omega))] \\ \text{subject to} \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

Second stage

$$\begin{aligned} Q(x, \xi(\omega)) = \min_{y \in \mathbb{R}^m} \quad & q_\xi^T y \\ \text{s.t.} \quad & W y = h_\xi - T_\xi x \\ & y \geq 0 \end{aligned}$$

Preliminaries - Finite extensive form

- Ω finite (N scenarios)

Preliminaries - Finite extensive form

- Ω finite (N scenarios)
- ξ discrete random variable

Preliminaries - Finite extensive form

- Ω finite (N scenarios)
- ξ discrete random variable

$$
\begin{aligned}
\underset{x \in \mathbb{R}^n, y_s \in \mathbb{R}^m}{\text{minimize}} \quad & c^T x + \sum_{s=1}^{N} \pi_s q_s^T y_s \\
\text{subject to} \quad & Ax = b \\
& T_s x + W y_s = h_s, \qquad s = 1, \dots, N \\
& x \geq 0, \; y_s \geq 0, \qquad s = 1, \dots, N
\end{aligned}
$$

Preliminaries - Finite extensive form

- Ω finite (N scenarios)
- ξ discrete random variable

$$
\begin{aligned}
\underset{x \in \mathbb{R}^n, y_s \in \mathbb{R}^m}{\text{minimize}} \quad & c^T x + \sum_{s=1}^{N} \pi_s q_s^T y_s \\
\text{subject to} \quad & Ax = b \\
& T_s x + W y_s = h_s, \qquad s = 1, \dots, N \\
& x \geq 0, \; y_s \geq 0, \qquad s = 1, \dots, N
\end{aligned}
$$

Also commonly referred to as the *deterministic equivalent problem*.

Preliminaries - Sample average approximation

- Ω infinite

Preliminaries - Sample average approximation

- Ω infinite
- ξ continuous random variable

Preliminaries - Sample average approximation

- Ω infinite
- ξ continuous random variable
- Sample N scenarios $\omega_s, \ s = 1, \dots, N$ independently from Ω

Preliminaries - Sample average approximation

- Ω infinite
- ξ continuous random variable
- Sample N scenarios $\omega_s,\ s = 1, \dots, N$ independently from Ω

$$
\begin{aligned}
\underset{x \in \mathbb{R}^n, y_s \in \mathbb{R}^m}{\text{minimize}} \quad & c^T x + \frac{1}{N} \sum_{s=1}^{N} q_s^T y_s \\
\text{subject to} \quad & Ax = b \\
& T_s x + W y_s = h_s, \qquad s = 1, \dots, N \\
& x \geq 0,\ y_s \geq 0, \qquad s = 1, \dots, N
\end{aligned}
$$

Preliminaries - Sample average approximation

- Ω infinite
- ξ continuous random variable
- Sample N scenarios $\omega_s,\ s = 1,\dots,N$ independently from Ω

$$
\begin{aligned}
\underset{x\in\mathbb{R}^n, y_s\in\mathbb{R}^m}{\text{minimize}} \quad & c^T x + \frac{1}{N}\sum_{s=1}^{N} q_s^T y_s \\
\text{subject to} \quad & Ax = b \\
& T_s x + W y_s = h_s, \qquad s = 1,\dots,N \\
& x \geq 0,\ y_s \geq 0, \qquad s = 1,\dots,N
\end{aligned}
$$

- Asymptotic convergence as N goes to infinity

Preliminaries - Sample average approximation

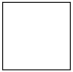
- Ω infinite
- ξ continuous random variable
- Sample N scenarios $\omega_s, \ s = 1, \dots, N$ independently from Ω

$$\begin{aligned} \underset{x \in \mathbb{R}^n, y_s \in \mathbb{R}^m}{\text{minimize}} \quad & c^T x + \frac{1}{N} \sum_{s=1}^{N} q_s^T y_s \\ \text{subject to} \quad & Ax = b \\ & T_s x + W y_s = h_s, \qquad s = 1, \dots, N \\ & x \geq 0, \ y_s \geq 0, \qquad s = 1, \dots, N \end{aligned}$$

- Asymptotic convergence as N goes to infinity
- Confidence intervals around optimal solution

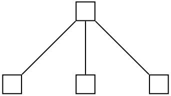
Preliminaries - Solution algorithms

Deterministic equivalent



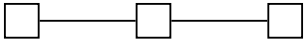
- GLPK
- Gurobi

Stage-decomposition



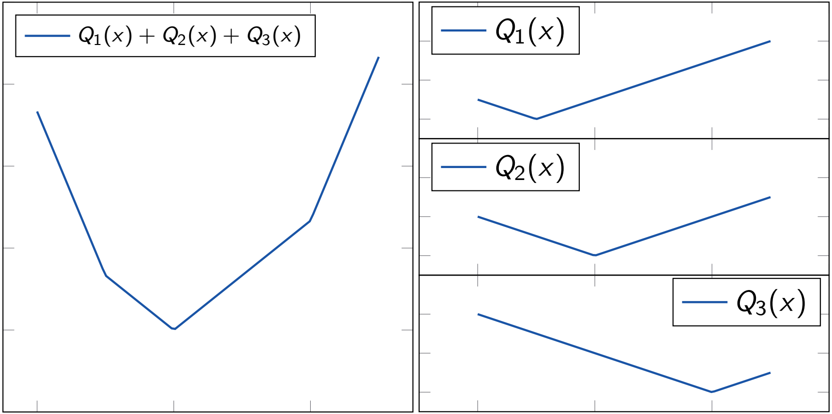
- L-shaped
- Quasi-gradient

Scenario-decomposition

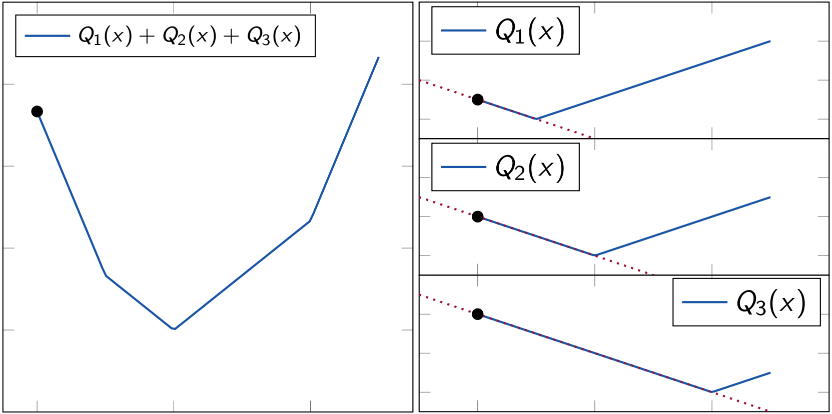


- Progressive-hedging

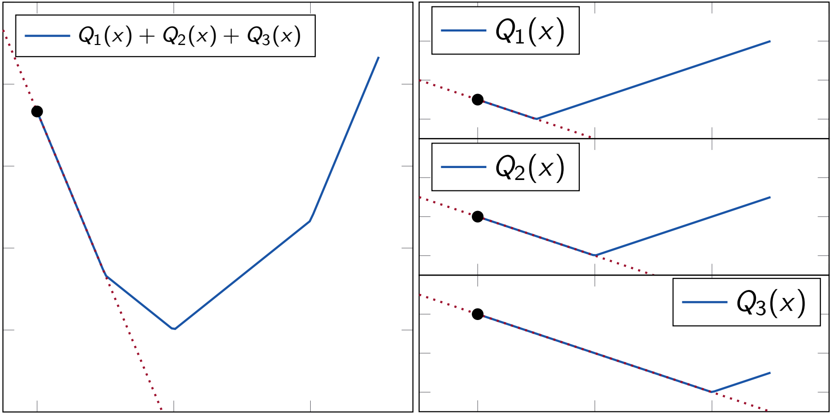
Preliminaries - The L-shaped algorithm



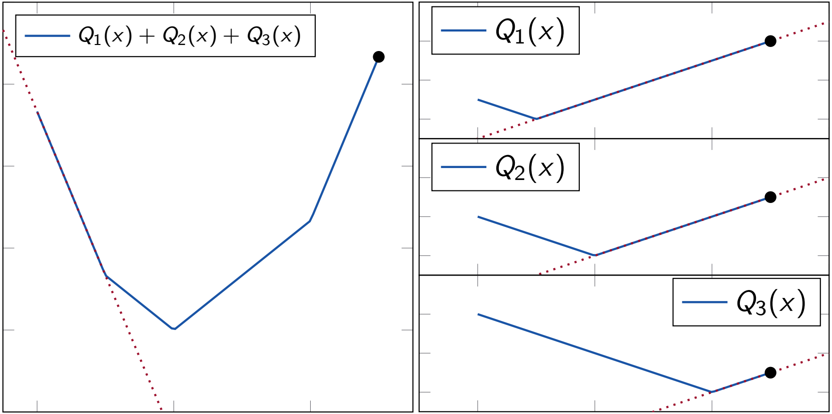
Preliminaries - The L-shaped algorithm



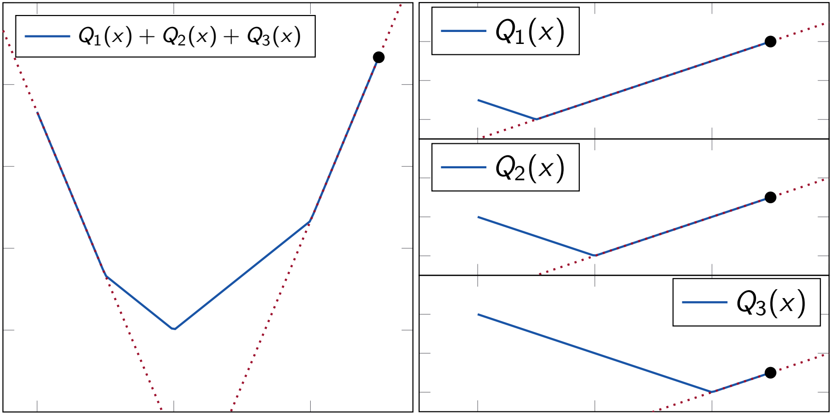
Preliminaries - The L-shaped algorithm



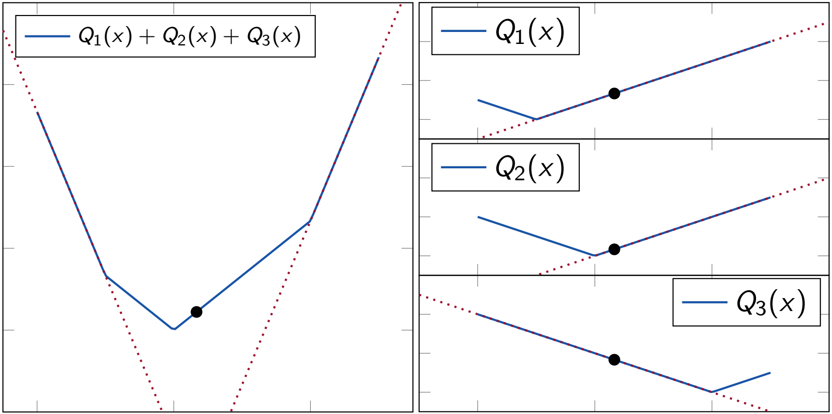
Preliminaries - The L-shaped algorithm



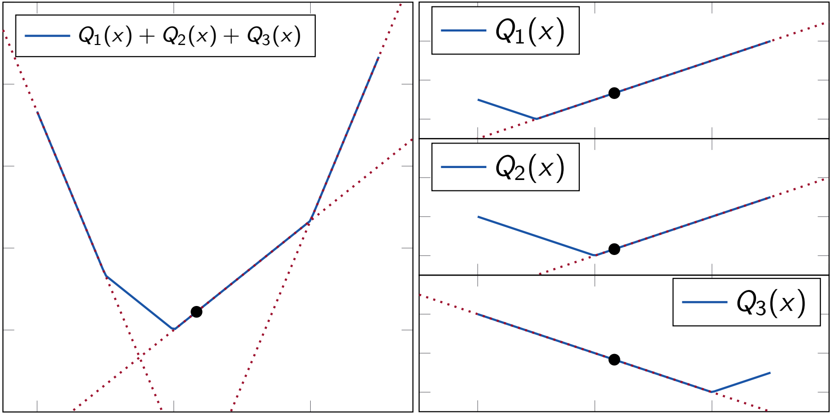
Preliminaries - The L-shaped algorithm



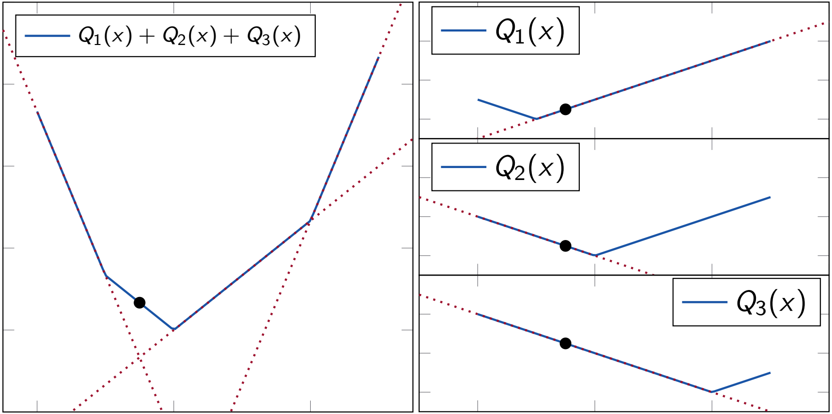
Preliminaries - The L-shaped algorithm



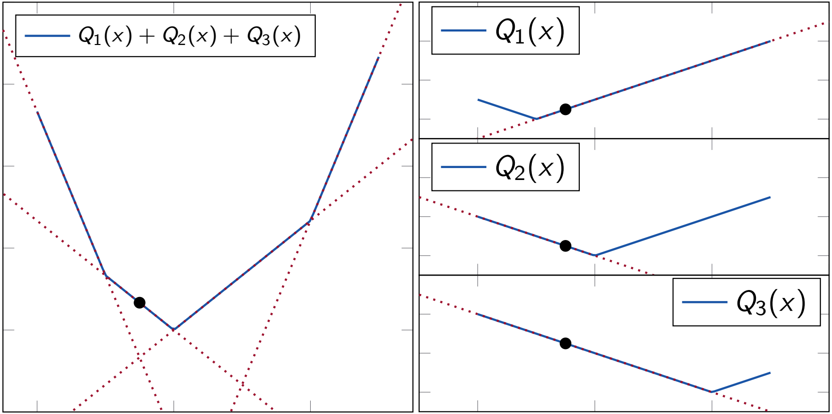
Preliminaries - The L-shaped algorithm



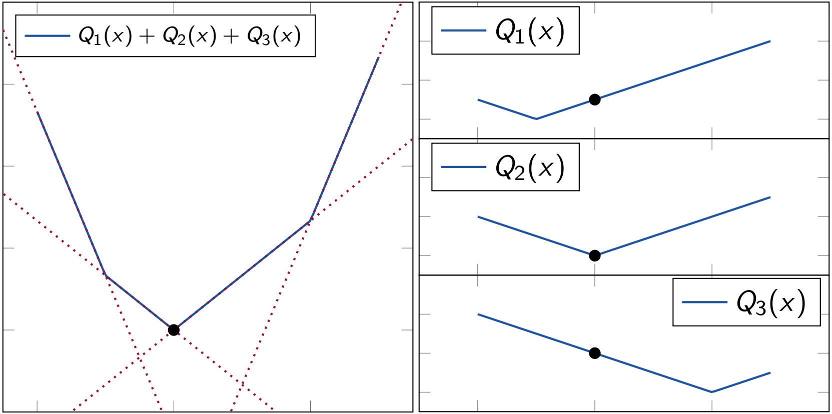
Preliminaries - The L-shaped algorithm



Preliminaries - The L-shaped algorithm



Preliminaries - The L-shaped algorithm



Outline

Contribution

- `StochasticPrograms.jl`: framework for stochastic programming

Publications

- Martin Biel and Mikael Johansson. Efficient stochastic programming in Julia. *INFORMS Journal on Computing*, 2021.
  in press

Contribution

- `StochasticPrograms.jl`: framework for stochastic programming
- Easy to formulate, solve, and analyze stochastic models

Publications

- Martin Biel and Mikael Johansson. Efficient stochastic programming in Julia. *INFORMS Journal on Computing*, 2021.
  in press

Contribution

- `StochasticPrograms.jl`: framework for stochastic programming
- Easy to formulate, solve, and analyze stochastic models
- Distributed-memory implementation for large-scale models

Publications

- Martin Biel and Mikael Johansson. Efficient stochastic programming in Julia. *INFORMS Journal on Computing*, 2021.
  in press

Contribution

- `StochasticPrograms.jl`: framework for stochastic programming
- Easy to formulate, solve, and analyze stochastic models
- Distributed-memory implementation for large-scale models
- Efficient implementations of structure-exploiting algorithms

Publications

- Martin Biel and Mikael Johansson. Efficient stochastic programming in Julia. *INFORMS Journal on Computing*, 2021.
  in press

StochasticPrograms.jl - Simple model

$$
\begin{aligned}
\underset{x_1, x_2 \in \mathbb{R}}{\text{minimize}} \quad & 100x_1 + 150x_2 + \mathbb{E}_\xi[Q(x_1, x_2, \xi)] \\
\text{subject to} \quad & x_1 + x_2 \leq 120 \\
& x_1 \geq 40 \\
& x_2 \geq 20
\end{aligned}
$$

where

$$
\begin{aligned}
Q(x_1, x_2, \xi) = \max_{y_1, y_2 \in \mathbb{R}} \quad & q_1(\xi)y_1 + q_2(\xi)y_2 \\
\text{s.t.} \quad & 6y_1 + 10y_2 \leq 60x_1 \\
& 8y_1 + 5y_2 \leq 80x_2 \\
& 0 \leq y_1 \leq d_1(\xi) \\
& 0 \leq y_2 \leq d_2(\xi)
\end{aligned}
$$

StochasticPrograms.jl - Simple model

```julia
@stochastic_model simple begin
    @stage 1 begin
        @decision(simple, x₁ >= 40)
        @decision(simple, x₂ >= 20)
        @objective(simple, Min, 100*x₁ + 150*x₂)
        @constraint(simple, x₁ + x₂ <= 120)
    end
    @stage 2 begin
        @uncertain q₁ q₂ d₁ d₂
        @recourse(simple, 0 <= y₁ <= d₁)
        @recourse(simple, 0 <= y₂ <= d₂)
        @objective(simple, Max, q₁*y₁ + q₂*y₂)
        @constraint(simple, 6*y₁ + 10*y₂ <= 60*x₁)
        @constraint(simple, 8*y₁ + 5*y₂ <= 80*x₂)
    end
end
```

StochasticPrograms.jl - Simple model

```
@stochastic_model simple begin
    @stage 1 begin
        @decision(simple, x₁ >= 40)
        @decision(simple, x₂ >= 20)
        @objective(simple, Min, 100*x₁ + 150*x₂)
        @constraint(simple, x₁ + x₂ <= 120)
    end
    @stage 2 begin
        @uncertain q₁ q₂ d₁ d₂
        @recourse(simple, 0 <= y₁ <= d₁)
        @recourse(simple, 0 <= y₂ <= d₂)
        @objective(simple, Max, q₁*y₁ + q₂*y₂)
        @constraint(simple, 6*y₁ + 10*y₂ <= 60*x₁)
        @constraint(simple, 8*y₁ + 5*y₂ <= 80*x₂)
    end
end
```

StochasticPrograms.jl - Simple model

```
@stochastic_model simple begin
    @stage 1 begin
        @decision(simple, x₁ >= 40)
        @decision(simple, x₂ >= 20)
        @objective(simple, Min, 100*x₁ + 150*x₂)
        @constraint(simple, x₁ + x₂ <= 120)
    end
    @stage 2 begin
        @uncertain q₁ q₂ d₁ d₂
        @recourse(simple, 0 <= y₁ <= d₁)
        @recourse(simple, 0 <= y₂ <= d₂)
        @objective(simple, Max, q₁*y₁ + q₂*y₂)
        @constraint(simple, 6*y₁ + 10*y₂ <= 60*x₁)
        @constraint(simple, 8*y₁ + 5*y₂ <= 80*x₂)
    end
end
```

StochasticPrograms.jl - Simple model

```julia
@stochastic_model simple begin
    @stage 1 begin
        @decision(simple, x₁ >= 40)
        @decision(simple, x₂ >= 20)
        @objective(simple, Min, 100*x₁ + 150*x₂)
        @constraint(simple, x₁ + x₂ <= 120)
    end
    @stage 2 begin
        @uncertain q₁ q₂ d₁ d₂
        @recourse(simple, 0 <= y₁ <= d₁)
        @recourse(simple, 0 <= y₂ <= d₂)
        @objective(simple, Max, q₁*y₁ + q₂*y₂)
        @constraint(simple, 6*y₁ + 10*y₂ <= 60*x₁)
        @constraint(simple, 8*y₁ + 5*y₂ <= 80*x₂)
    end
end
```

JuMP syntax

StochasticPrograms.jl - Simple model

```
@stochastic_model simple begin
    @stage 1 begin
        @decision(simple, x₁ >= 40)
        @decision(simple, x₂ >= 20)
        @objective(simple, Min, 100*x₁ + 150*x₂)
        @constraint(simple, x₁ + x₂ <= 120)
    end
    @stage 2 begin
        @uncertain q₁ q₂ d₁ d₂
        @recourse(simple, 0 <= y₁ <= d₁)
        @recourse(simple, 0 <= y₂ <= d₂)
        @objective(simple, Max, q₁*y₁ + q₂*y₂)
        @constraint(simple, 6*y₁ + 10*y₂ <= 60*x₁)
        @constraint(simple, 8*y₁ + 5*y₂ <= 80*x₂)
    end
end
```

$$\begin{aligned} \underset{x_1, x_2 \in \mathbb{R}}{\text{minimize}} \quad & 100x_1 + 150x_2 \\ \text{subject to} \quad & x_1 + x_2 \leq 120 \\ & x_1 \geq 40 \\ & x_2 \geq 20 \end{aligned}$$

StochasticPrograms.jl - Simple model

```
@stochastic_model simple begin
    @stage 1 begin
        @decision(simple, x₁ >= 40)
        @decision(simple, x₂ >= 20)
        @objective(simple, Min, 100*x₁ + 150*x₂)
        @constraint(simple, x₁ + x₂ <= 120)
    end
    @stage 2 begin
        @uncertain q₁ q₂ d₁ d₂
        @recourse(simple, 0 <= y₁ <= d₁)
        @recourse(simple, 0 <= y₂ <= d₂)
        @objective(simple, Max, q₁*y₁ + q₂*y₂)
        @constraint(simple, 6*y₁ + 10*y₂ <= 60*x₁)
        @constraint(simple, 8*y₁ + 5*y₂ <= 80*x₂)
    end
end
```

$$\begin{aligned} \max_{y_1, y_2 \in \mathbb{R}} \quad & q_1(\xi)\ y_1 + \ q_2(\xi)\ y_2 \\ \text{s.t.} \quad & 6y_1 + 10y_2 \leq 60\ x_1 \\ & 8y_1 + 5y_2 \leq 80\ x_2 \\ & 0 \leq y_1 \leq \ d_1(\xi) \\ & 0 \leq y_2 \leq \ d_2(\xi) \end{aligned}$$

StochasticPrograms.jl - Simple model

```julia
@stochastic_model simple begin
    @stage 1 begin
        @decision(simple, x₁ >= 40)
        @decision(simple, x₂ >= 20)
        @objective(simple, Min, 100*x₁ + 150*x₂)
        @constraint(simple, x₁ + x₂ <= 120)
    end
    @stage 2 begin
        @uncertain q₁ q₂ d₁ d₂
        @recourse(simple, 0 <= y₁ <= d₁)
        @recourse(simple, 0 <= y₂ <= d₂)
        @objective(simple, Max, q₁*y₁ + q₂*y₂)
        @constraint(simple, 6*y₁ + 10*y₂ <= 60*x₁)
        @constraint(simple, 8*y₁ + 5*y₂ <= 80*x₂)
    end
end
```

$$\begin{aligned} \max_{y_1, y_2 \in \mathbb{R}} \quad & q_1(\xi)\, y_1 + q_2(\xi)\, y_2 \\ \text{s.t.} \quad & 6y_1 + 10y_2 \leq 60\, x_1 \\ & 8y_1 + 5y_2 \leq 80\, x_2 \\ & 0 \leq y_1 \leq d_1(\xi) \\ & 0 \leq y_2 \leq d_2(\xi) \end{aligned}$$

StochasticPrograms.jl - Simple model

```
@stochastic_model simple begin
    @stage 1 begin
        @decision(simple, x₁ >= 40)
        @decision(simple, x₂ >= 20)
        @objective(simple, Min, 100*x₁ + 150*x₂)
        @constraint(simple, x₁ + x₂ <= 120)
    end
    @stage 2 begin
        @uncertain q₁ q₂ d₁ d₂
        @recourse(simple, 0 <= y₁ <= d₁)
        @recourse(simple, 0 <= y₂ <= d₂)
        @objective(simple, Max, q₁*y₁ + q₂*y₂)
        @constraint(simple, 6*y₁ + 10*y₂ <= 60*x₁)
        @constraint(simple, 8*y₁ + 5*y₂ <= 80*x₂)
    end
end
```

$$\begin{aligned} \max_{y_1,y_2\in\mathbb{R}} \quad & q_1(\xi)\, y_1 + q_2(\xi)\, y_2 \\ \text{s.t.} \quad & 6y_1 + 10y_2 \leq 60\, x_1 \\ & 8y_1 + 5y_2 \leq 80\, x_2 \\ & 0 \leq y_1 \leq d_1(\xi) \\ & 0 \leq y_2 \leq d_2(\xi) \end{aligned}$$

StochasticPrograms.jl - Simple model

```
ξ₁ = @scenario q₁ = 24.0 q₂ = 28.0 d₁ = 500.0 d₂ = 100.0 probability = 0.4;
ξ₂ = @scenario q₁ = 28.0 q₂ = 32.0 d₁ = 300.0 d₂ = 300.0 probability = 0.6;
sp = instantiate(simple, [ξ₁,ξ₂], optimizer = LShaped.Optimizer)

Stochastic program with:
 * 2 decision variables
 * 2 recourse variables
 * 2 scenarios of type Scenario
Structure: Stage-decomposition
Solver name: L-shaped
```

StochasticPrograms.jl - Simple model

```
ξ₁ = @scenario q₁ = 24.0 q₂ = 28.0 d₁ = 500.0 d₂ = 100.0 probability = 0.4;
ξ₂ = @scenario q₁ = 28.0 q₂ = 32.0 d₁ = 300.0 d₂ = 300.0 probability = 0.6;
sp = instantiate(simple, [ξ₁,ξ₂], optimizer = LShaped.Optimizer)

Stochastic program with:
 * 2 decision variables
 * 2 recourse variables
 * 2 scenarios of type Scenario
Structure: Stage-decomposition
Solver name: L-shaped
```

StochasticPrograms.jl - Simple model

```
ξ₁ = @scenario q₁ = 24.0 q₂ = 28.0 d₁ = 500.0 d₂ = 100.0 probability = 0.4;
ξ₂ = @scenario q₁ = 28.0 q₂ = 32.0 d₁ = 300.0 d₂ = 300.0 probability = 0.6;
sp = instantiate(simple, [ξ₁,ξ₂], optimizer = LShaped.Optimizer)

Stochastic program with:
 * 2 decision variables
 * 2 recourse variables
 * 2 scenarios of type Scenario
Structure: Stage-decomposition
Solver name: L-shaped
```

StochasticPrograms.jl - Simple model

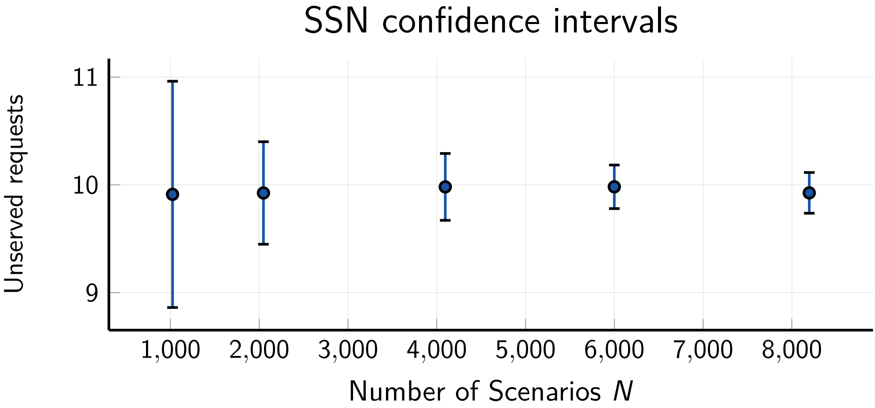
```
set_optimizer_attribute(sp, MasterOptimizer(), GLPK.Optimizer)
set_optimizer_attribute(sp, SubProblemOptimizer(), GLPK.Optimizer)
optimize!(sp)

L-Shaped Gap  Time: 0:00:00 (6 iterations)
  Objective:       -855.8333333333303
  Gap:             2.1254014452334763e-15
  Number of cuts:  7
  Iterations:      6
```

StochasticPrograms.jl - Features

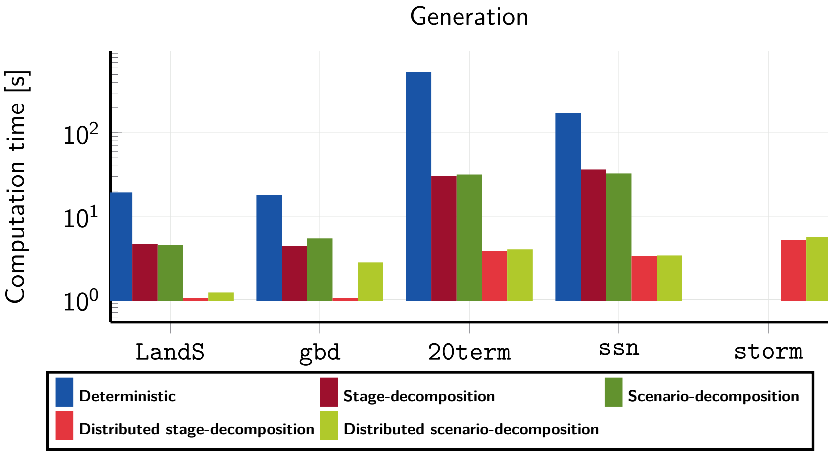
- Flexible and expressive problem definition
- Discrete models
- Continuous models (through sampling)
- Variety of tools for analyzing models
  - VSS
  - EVPI
  - Confidence intervals
- Distributed models
- Interface to structure-exploiting (parallel) solver algorithms
  - L-shaped variants
  - Progressive-hedging variants
  - Quasi-gradient variants

StochasticPrograms.jl - SSN



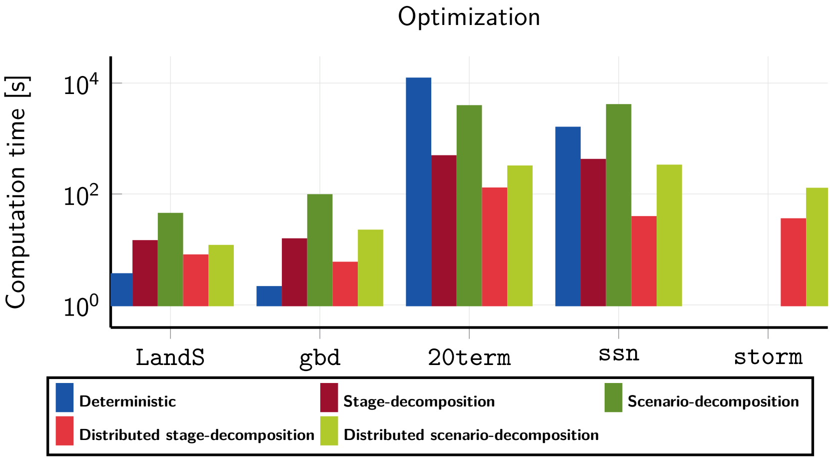
- 90% Confidence intervals around the optimal value
- Stable solution with 6000 scenarios

StochasticPrograms.jl - Numerical experiments



- Significant improvements with decomposition structures
- Largest problem `storm` requires distributed decomposition

StochasticPrograms.jl - Numerical experiments



Significant gains with decomposition structures on larger problems

StochasticPrograms.jl - Summary

- `StochasticPrograms.jl`
  - Easy to use
  - Comprehensive
  - Distributed capabilities

StochasticPrograms.jl - Summary

- `StochasticPrograms.jl`
  - Easy to use
  - Comprehensive
  - Distributed capabilities
- Useful for:
  - Researchers
  - Educators
  - Industrial practitioners

StochasticPrograms.jl - Summary

- `StochasticPrograms.jl`
  - Easy to use
  - Comprehensive
  - Distributed capabilities
- Useful for:
  - Researchers
  - Educators
  - Industrial practitioners

Open-source and available as an official Julia package:
`https://github.com/martinbiel/StochasticPrograms.jl`

Outline

Outline

Contribution

- Policy-based software design

Publications

- Martin Biel and Mikael Johansson.

  Efficient stochastic programming in Julia. *INFORMS Journal on Computing*, 2021. in press
- Martin Biel and Mikael Johansson.

  Distributed L-shaped algorithms in Julia. In *2018 IEEE/ACM Parallel Applications Workshop, Alternatives To MPI (PAW-ATM)*. IEEE, 2018
- Martin Biel, Arda Aytekin, and Mikael

  Johansson. POLO.jl: Policy-based optimization algorithms in Julia. *Advances in Engineering Software*, 136:102695, 2019

Contribution

- Policy-based software design
- The L-shaped algorithm
  - Variations
  - Distributed extension
  - Solver module

Publications

- Martin Biel and Mikael Johansson.

  Efficient stochastic programming in Julia. *INFORMS Journal on Computing*, 2021. in press
- Martin Biel and Mikael Johansson.

  Distributed L-shaped algorithms in Julia. In *2018 IEEE/ACM Parallel Applications Workshop, Alternatives To MPI (PAW-ATM)*. IEEE, 2018
- Martin Biel, Arda Aytekin, and Mikael

  Johansson. POLO.jl: Policy-based optimization algorithms in Julia. *Advances in Engineering Software*, 136:102695, 2019

Contribution

- Policy-based software design
- The L-shaped algorithm
  - Variations
  - Distributed extension
  - Solver module
- Progressive-hedging
- Quasi-gradient algorithm

Publications

- Martin Biel and Mikael Johansson.
  Efficient stochastic programming in Julia. *INFORMS Journal on Computing*, 2021. in press
- Martin Biel and Mikael Johansson.
  Distributed L-shaped algorithms in Julia. In *2018 IEEE/ACM Parallel Applications Workshop, Alternatives To MPI (PAW-ATM)*. IEEE, 2018
- Martin Biel, Arda Aytekin, and Mikael Johansson. POLO.jl: Policy-based optimization algorithms in Julia. *Advances in Engineering Software*, 136:102695, 2019

Contribution

- Policy-based software design
- The L-shaped algorithm
  - Variations
  - Distributed extension
  - Solver module
- Progressive-hedging
- Quasi-gradient algorithm
- Numerical experiments

Publications

- Martin Biel and Mikael Johansson.

  Efficient stochastic programming in Julia. *INFORMS Journal on Computing*, 2021. in press
- Martin Biel and Mikael Johansson.

  Distributed L-shaped algorithms in Julia. In *2018 IEEE/ACM Parallel Applications Workshop, Alternatives To MPI (PAW-ATM)*. IEEE, 2018
- Martin Biel, Arda Aytekin, and Mikael

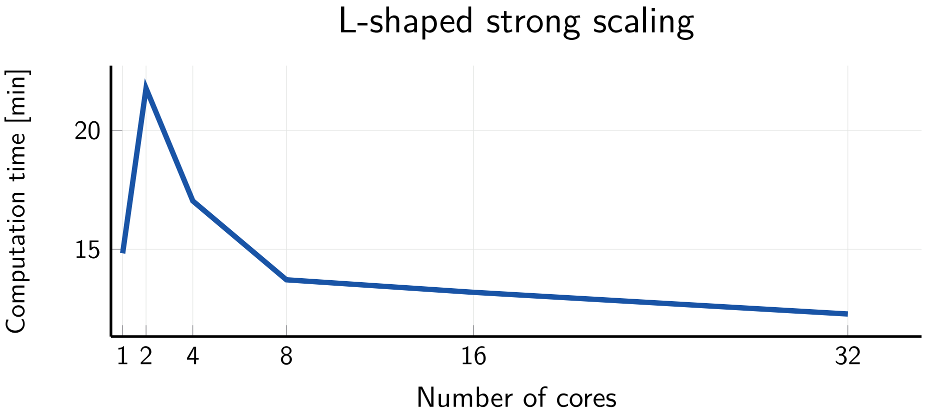
  Johansson. POLO.jl: Policy-based optimization algorithms in Julia. *Advances in Engineering Software*, 136:102695, 2019

Structured algorithm - Numerical experiments

The SSN problem

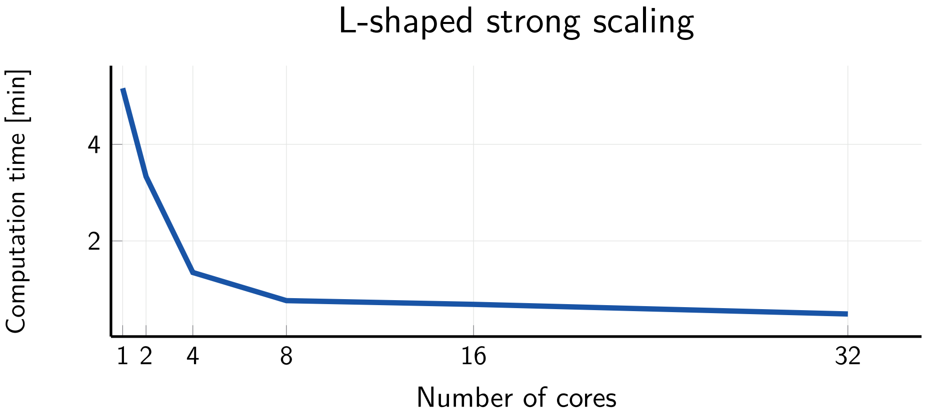
- Optimal bandwidth capacity expansion plan in a network
- Stable solution after sampling 6000 scenarios
- 30 minutes required to build and solve the deterministic equivalent
- Distribute over 32 remote worker nodes
- Laptop master node
- Non-neglible communication delay

Structured algorithms - Numerical experiments



- Initial parallel inefficiency with standard multi-cut L-shaped algorithm
- Cut communication considerable part of execution time
- Solving the master becomes bottleneck in final iterations

Structured algorithms - Numerical experiments



- Improved parallel efficiency with regularization and aggregation
- Improved load balance between master and workers
- Solved in 30 seconds on 32 cores (30 minutes in deterministic form)

Outline

Motivation

- Cut aggregation improves distributed performance

Motivation

- Cut aggregation improves distributed performance
  - Reduce communication latency
  - Reduce load imbalance

Motivation

- Cut aggregation improves distributed performance
  - Reduce communication latency
  - Reduce load imbalance
- Uniform cut aggregation has been applied in many recent works

Motivation

- Cut aggregation improves distributed performance
  - Reduce communication latency
  - Reduce load imbalance
- Uniform cut aggregation has been applied in many recent works
- Complexity analysis only covers single-cut and multi-cut L-shaped

Contribution

- Review of the use of cut aggregation in L-shaped algorithms

Publications

- Martin Biel and Mikael Johansson. Dynamic cut aggregation in L-shaped algorithms. *arXiv preprint arXiv:1910.13752*, 2019.
  Submitted for consideration to the European Journal of Operational Research

Contribution

- Review of the use of cut aggregation in L-shaped algorithms
- Novel dynamic cut aggregation procedure

Publications

- Martin Biel and Mikael Johansson. Dynamic cut aggregation in L-shaped algorithms. *arXiv preprint arXiv:1910.13752*, 2019.
  Submitted for consideration to the European Journal of Operational Research

Contribution

- Review of the use of cut aggregation in L-shaped algorithms
- Novel dynamic cut aggregation procedure
- Theoretical results
  - Static aggregation complexity
  - Dynamic aggregation convergence/complexity

Publications

- Martin Biel and Mikael Johansson. Dynamic cut aggregation in L-shaped algorithms. *arXiv preprint arXiv:1910.13752*, 2019.
  Submitted for consideration to the European Journal of Operational Research

Contribution

- Review of the use of cut aggregation in L-shaped algorithms
- Novel dynamic cut aggregation procedure
- Theoretical results
  - Static aggregation complexity
  - Dynamic aggregation convergence/complexity
- Dynamic aggregation schemes
  - On-line aggregation
  - Batch aggregation

Publications

- Martin Biel and Mikael Johansson. Dynamic cut aggregation in L-shaped algorithms. *arXiv preprint arXiv:1910.13752*, 2019.
  Submitted for consideration to the European Journal of Operational Research

Contribution

- Review of the use of cut aggregation in L-shaped algorithms
- Novel dynamic cut aggregation procedure
- Theoretical results
  - Static aggregation complexity
  - Dynamic aggregation convergence/complexity
- Dynamic aggregation schemes
  - On-line aggregation
  - Batch aggregation
- Granulated aggregation combines static and dynamic aggregation

Publications

- Martin Biel and Mikael Johansson. Dynamic cut aggregation in L-shaped algorithms. *arXiv preprint arXiv:1910.13752*, 2019.
  Submitted for consideration to the European Journal of Operational Research

Contribution

- Review of the use of cut aggregation in L-shaped algorithms
- Novel dynamic cut aggregation procedure
- Theoretical results
  - Static aggregation complexity
  - Dynamic aggregation convergence/complexity
- Dynamic aggregation schemes
  - On-line aggregation
  - Batch aggregation
- Granulated aggregation combines static and dynamic aggregation
- Performance improvements in large-scale examples

Publications

- Martin Biel and Mikael Johansson. Dynamic cut aggregation in L-shaped algorithms. *arXiv preprint arXiv:1910.13752*, 2019.
  Submitted for consideration to the European Journal of Operational Research

Cut aggregation - Numerical experiments

Algorithm \Problem	LandS	gbd	20term	ssn	storm	dayahead
Deterministic	-	-	11874.99	1544.95	-	1053.2
Multi-cut	719.08	307.24	118.12	81.38	236.12	36.31
Partial	129.21	29.47	101.95	58.61	117.58	19.93
SelectUniform	187.38	128.81	114.22	70.96	111.97	18.96
SelectClosest	256.28	561.38	648.43	109.63	68.89	22.07
Kmedoids	212.07	234.01	714.68	93.66	-	31.97
GranulatedSelectClosest	**129.16**	**24.39**	**55.99**	**43.26**	**68.15**	**15.87**
GranulatedKmedoids	157.73	29.39	91.34	70.69	81.13	20.69

Table: Median computation time, in seconds, required to solve the test problems using level-set regularized L-shaped with different aggregation schemes.

Cut aggregation - Numerical experiments

Algorithm \Problem	LandS	gbd	20term	ssn	storm	dayahead
Deterministic	-	-	11874.99	1544.95	-	1053.2
Multi-cut	719.08	307.24	118.12	81.38	236.12	36.31
Partial	129.21	29.47	101.95	58.61	117.58	19.93
SelectUniform	187.38	128.81	114.22	70.96	111.97	18.96
SelectClosest	256.28	561.38	648.43	109.63	68.89	22.07
Kmedoids	212.07	234.01	714.68	93.66	-	31.97
GranulatedSelectClosest	**129.16**	**24.39**	**55.99**	**43.26**	**68.15**	**15.87**
GranulatedKmedoids	157.73	29.39	91.34	70.69	81.13	20.69

Table: Median computation time, in seconds, required to solve the test problems using level-set regularized L-shaped with different aggregation schemes.

Outline

Motivation

- L-shaped has scalability issues as master problem grows in size

Motivation

- L-shaped has scalability issues as master problem grows in size
- Projected subgradient descent avoids scalability problem

Motivation

- L-shaped has scalability issues as master problem grows in size
- Projected subgradient descent avoids scalability problem
- ... but has poor convergence properties

Motivation

- L-shaped has scalability issues as master problem grows in size
- Projected subgradient descent avoids scalability problem
- ... but has poor convergence properties
- Recent advances to accelerate gradient descent

Contribution

- Smoothing procedure for linear two-stage stochastic programming

Publications

- Martin Biel, Vien Van Mai, and Mikael Johansson. A fast smoothing procedure for large-scale stochastic programming.
  In *2021 60th IEEE Conference on Decision and Control (CDC)*. IEEE, 2021.
  in press

Contribution

- Smoothing procedure for linear two-stage stochastic programming
- Enables the use of modern acceleration methods

Publications

- Martin Biel, Vien Van Mai, and Mikael Johansson. A fast smoothing procedure for large-scale stochastic programming.
  In *2021 60th IEEE Conference on Decision and Control (CDC)*. IEEE, 2021.
  in press

Contribution

- Smoothing procedure for linear two-stage stochastic programming
- Enables the use of modern acceleration methods
- Problem-dependent approximation bounds

Publications

- Martin Biel, Vien Van Mai, and Mikael Johansson. A fast smoothing procedure for large-scale stochastic programming.
  In *2021 60th IEEE Conference on Decision and Control (CDC)*. IEEE, 2021.
  in press

Contribution

- Smoothing procedure for linear two-stage stochastic programming
- Enables the use of modern acceleration methods
- Problem-dependent approximation bounds
- Convergence result

Publications

- Martin Biel, Vien Van Mai, and Mikael Johansson. A fast smoothing procedure for large-scale stochastic programming.
  In *2021 60th IEEE Conference on Decision and Control (CDC)*. IEEE, 2021.
  in press

Contribution

- Smoothing procedure for linear two-stage stochastic programming
- Enables the use of modern acceleration methods
- Problem-dependent approximation bounds
- Convergence result
- Theory suggest trade-off between speed and accuracy

Publications

- Martin Biel, Vien Van Mai, and Mikael Johansson. A fast smoothing procedure for large-scale stochastic programming.
  In *2021 60th IEEE Conference on Decision and Control (CDC)*. IEEE, 2021.
  in press

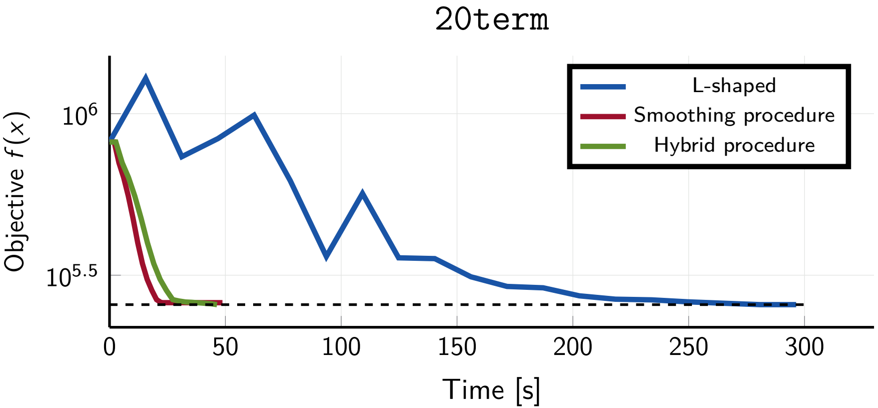
Contribution

- Smoothing procedure for linear two-stage stochastic programming
- Enables the use of modern acceleration methods
- Problem-dependent approximation bounds
- Convergence result
- Theory suggest trade-off between speed and accuracy
- Hybrid scheme that is fast and accurate

Publications

- Martin Biel, Vien Van Mai, and Mikael Johansson. A fast smoothing procedure for large-scale stochastic programming.
  In *2021 60th IEEE Conference on Decision and Control (CDC)*. IEEE, 2021.
  in press

Numerical experiments - 20term



- Smoothing procedure outperforms regularized L-shaped
- Accurate solution with hybrid procedure

Outline

Contribution

- Large-scale hydropower models

Publications

- Martin Biel. Optimal day-ahead orders using stochastic programming and noise-driven recurrent neural networks. In *2021 IEEE Madrid PowerTech*, pages 1–6, 2021
- Martin Biel. Large-scale hydropower models in StochasticPrograms.jl. *arXiv preprint arXiv:2111.02099*, 2021. Submitted for consideration to Computational Optimization and Applications. Under review

Contribution

- Large-scale hydropower models
- Three case studies:
  - Day-ahead planning
  - Maintenance scheduling
  - Capacity expansion

Publications

- Martin Biel. Optimal day-ahead orders using stochastic programming and noise-driven recurrent neural networks. In *2021 IEEE Madrid PowerTech*, pages 1–6, 2021
- Martin Biel. Large-scale hydropower models in StochasticPrograms.jl. *arXiv preprint arXiv:2111.02099*, 2021. Submitted for consideration to Computational Optimization and Applications. Under review

Contribution

- Large-scale hydropower models
- Three case studies:
  - Day-ahead planning
  - Maintenance scheduling
  - Capacity expansion
- Complete modeling procedure:
  - Data gathering
  - Forecast generation
  - Model formulation
  - Model implementation
  - Optimization
  - Result visualization

Publications

- Martin Biel. Optimal day-ahead orders using stochastic programming and noise-driven recurrent neural networks. In *2021 IEEE Madrid PowerTech*, pages 1–6, 2021
- Martin Biel. Large-scale hydropower models in StochasticPrograms.jl. *arXiv preprint arXiv:2111.02099*, 2021. Submitted for consideration to Computational Optimization and Applications. Under review

Contribution

- Day-ahead planning
  - Optimize orders submitted to market the next day
  - Statistically significant value of stochastic planning

Contribution

- Day-ahead planning
  - Optimize orders submitted to market the next day
  - Statistically significant value of stochastic planning
- Maintenance scheduling
  - Day-ahead problem with preventive maintenance
  - Value of stochastic planning increases

Contribution

- Day-ahead planning
  - Optimize orders submitted to market the next day
  - Statistically significant value of stochastic planning
- Maintenance scheduling
  - Day-ahead problem with preventive maintenance
  - Value of stochastic planning increases
- Capacity expansion
  - 1 year horizon (daily increment)
  - 20 year horizon (5-day increments)

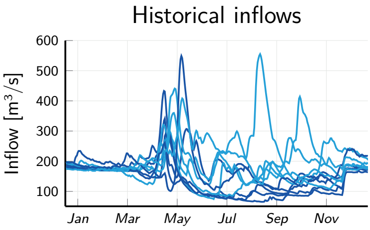
Contribution

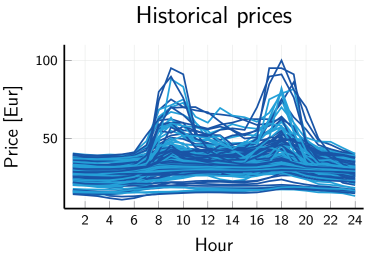
- Day-ahead planning
  - Optimize orders submitted to market the next day
  - Statistically significant value of stochastic planning
- Maintenance scheduling
  - Day-ahead problem with preventive maintenance
  - Value of stochastic planning increases
- Capacity expansion
  - 1 year horizon (daily increment)
  - 20 year horizon (5-day increments)

All models are formulated in `StochasticPrograms.jl`, distributed on 32 worker cores, and solved using parallel algorithms

Outline

Uncertain parameters





Forecast generation

- Noise-driven recurrent neural network
- Trained on price data and inflow data separately
- Seasonality modeled through separate inputs to the network

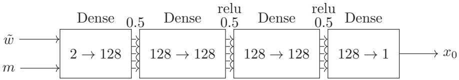


Figure: Initializer network in the price forecaster.

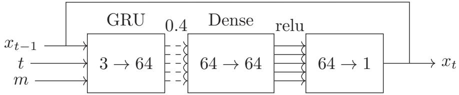


Figure: Sequence generation network in the price forecaster.

Forecast generation

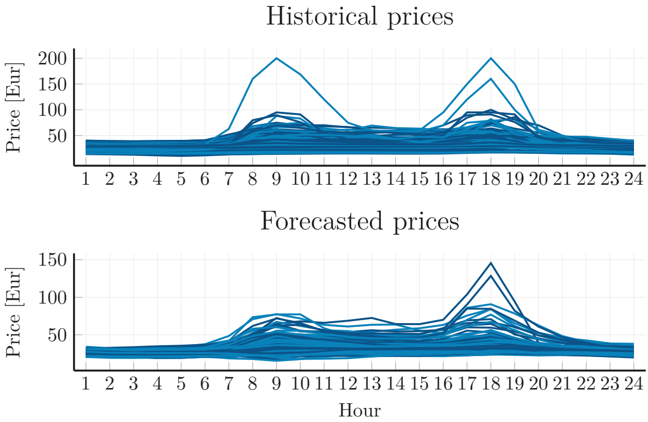


Figure: Historical electricity price curves in January and electricity price curves generated using the RNN forecaster in the same period.

Forecast generation

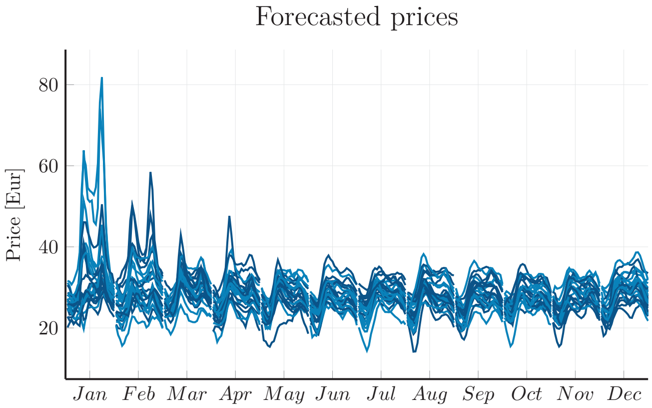


Figure: Daily electricity price curves predicted by the RNN forecaster in every month of the year.

Outline

Day-ahead - Electricity markets

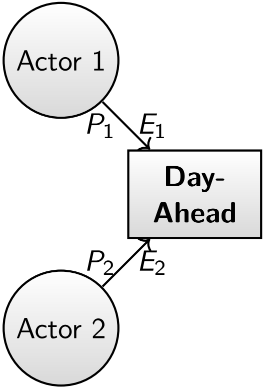
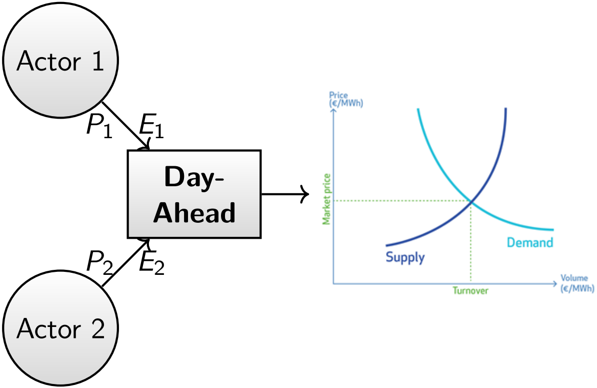


Figure: Deregulated electricity market.

Day-ahead - Electricity markets



Market closes

Figure: Deregulated electricity market.

Day-ahead - Electricity markets

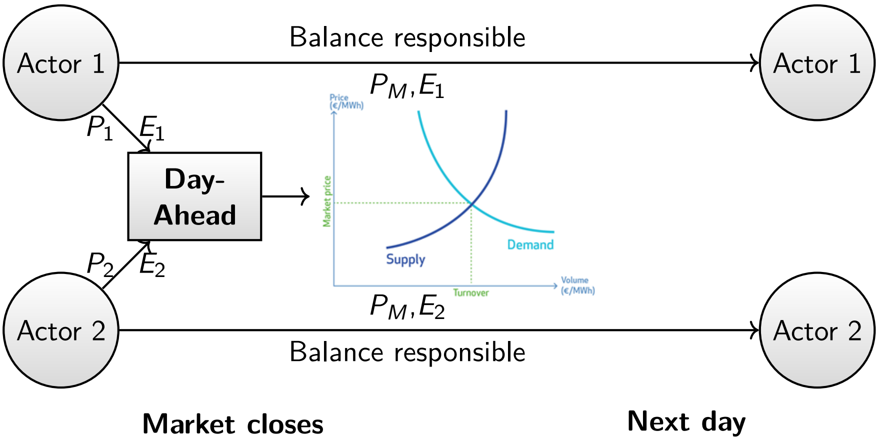


Figure: Deregulated electricity market.

Day-ahead - Electricity markets

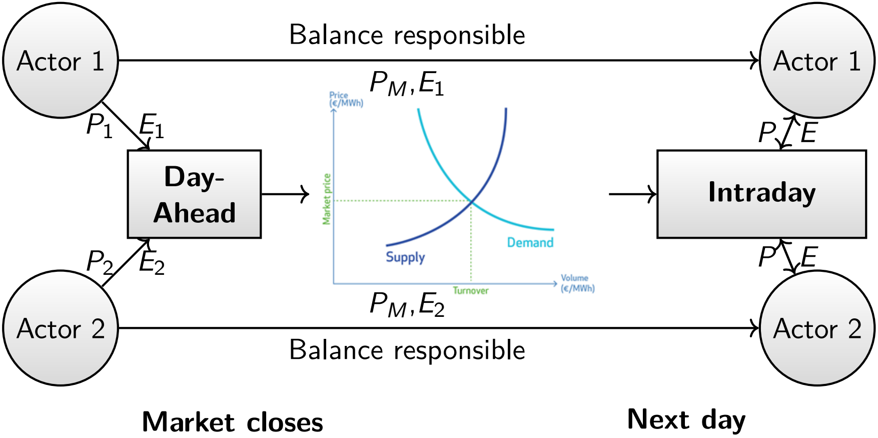


Figure: Deregulated electricity market.

Day-ahead - General description

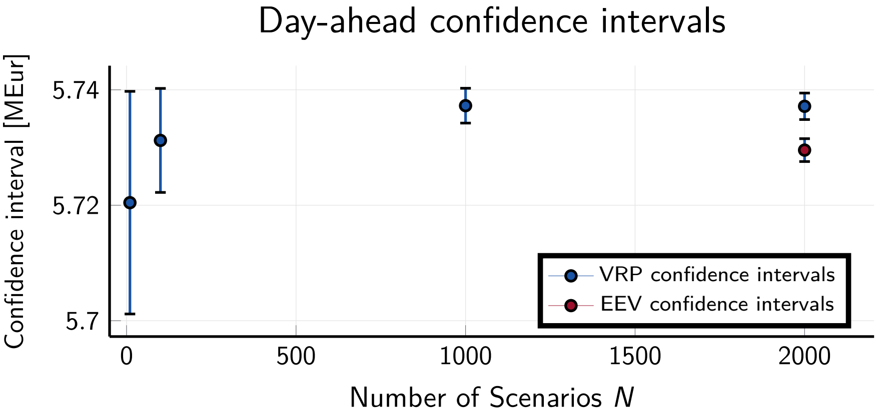
First stage

$$\underset{\text{Order strategy}}{\text{maximize}} \quad \mathbb{E}[\text{R}(\text{Order strategy}, \text{Price}, \text{Inflow})]$$
$$\text{subject to} \quad \text{Trade regulations}$$

Second stage

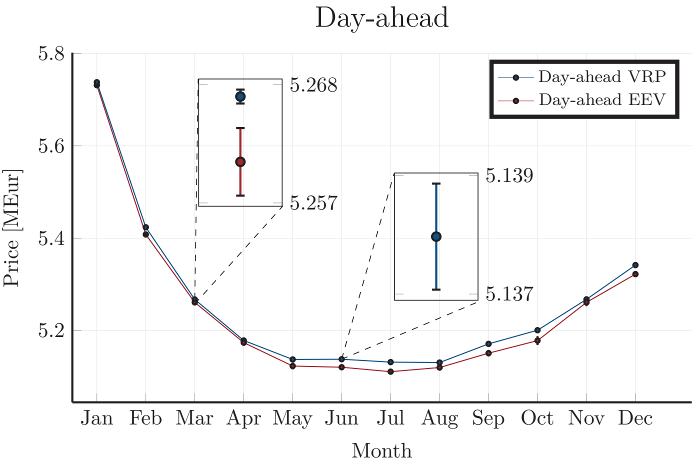
$$R = \max_{\text{Production}} \quad \text{Profit}(\text{Price}) + \text{Water value} - \text{Imbalance penalty}$$
$$\text{s.t.} \quad \text{Load balance}(\text{Order strategy}, \text{Balance trading}, \text{Price})$$
$$\text{Hydrological balance}(\text{Inflow})$$
$$\text{Electricity production}$$

Day-ahead - Convergence

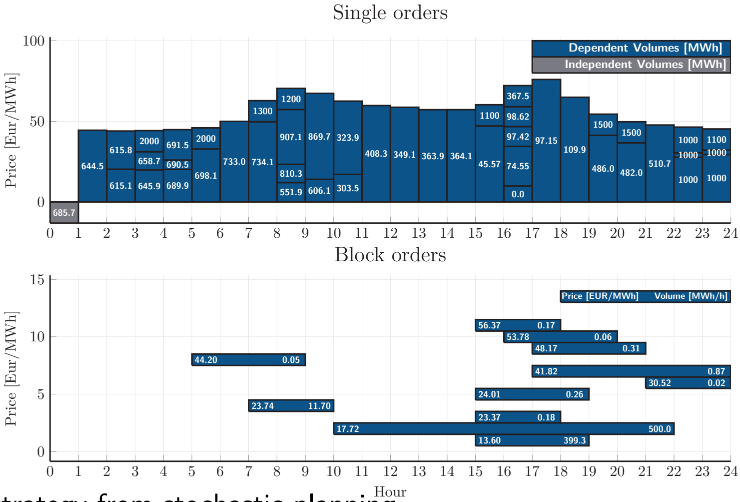


- Stable solution with 2000 scenarios
- Statistically significant gap to deterministic strategy

Day-ahead - Seasonal behavior

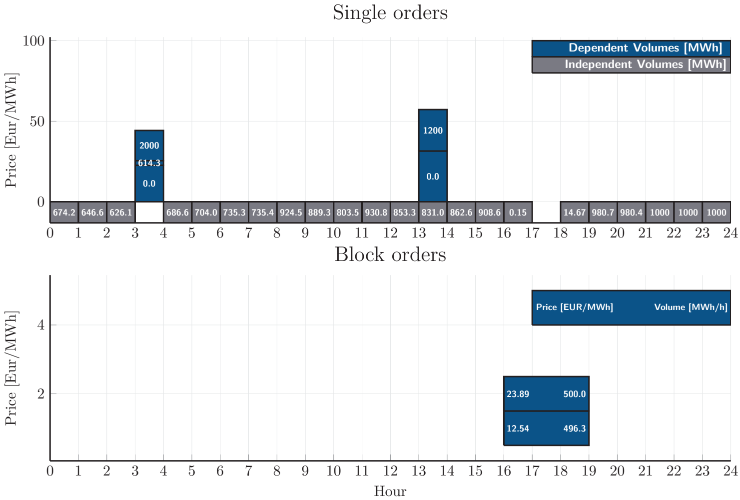


Day-ahead - Strategies



Order strategy from stochastic planning

Day-ahead - Strategies



Deterministic strategy from just considering the average market price

Outline

Maintenance scheduling - General description

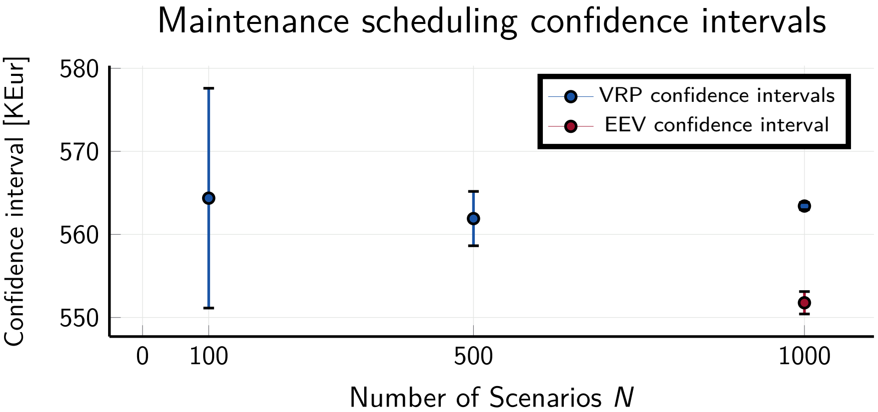
First stage

$$\begin{aligned} \underset{\text{Orders + schedule}}{\text{maximize}} \quad & \mathbb{E}[R(\text{Orders}, \text{Schedule}, \text{Price}, \text{Inflow})] \\ \text{subject to} \quad & \text{Trade regulations} \\ & \text{Schedule restrictions} \end{aligned}$$

Second stage

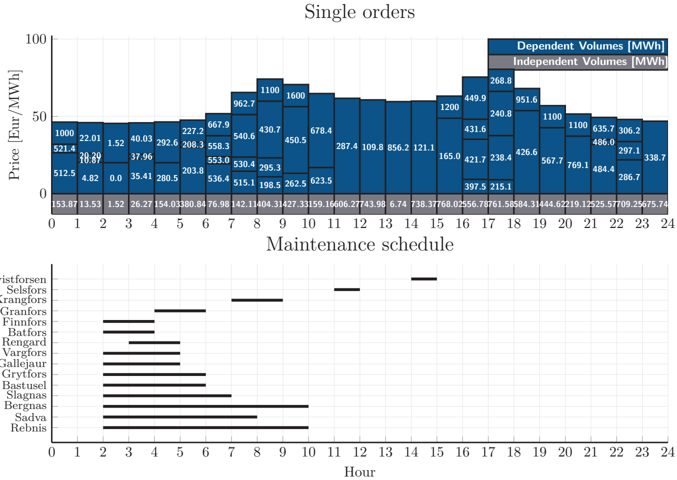
$$\begin{aligned} R = \max_{\text{Production}} \quad & \text{Profit}(\text{Price}) - \text{Imbalance penalty} \\ \text{s.t.} \quad & \text{Load balance}(\text{Order strategy}, \text{Balance trading}, \text{Price}) \\ & \text{Hydrological balance}(\text{Inflow}) \\ & \text{Electricity production}(\text{Maintenance schedule}) \end{aligned}$$

Maintenance scheduling - Convergence



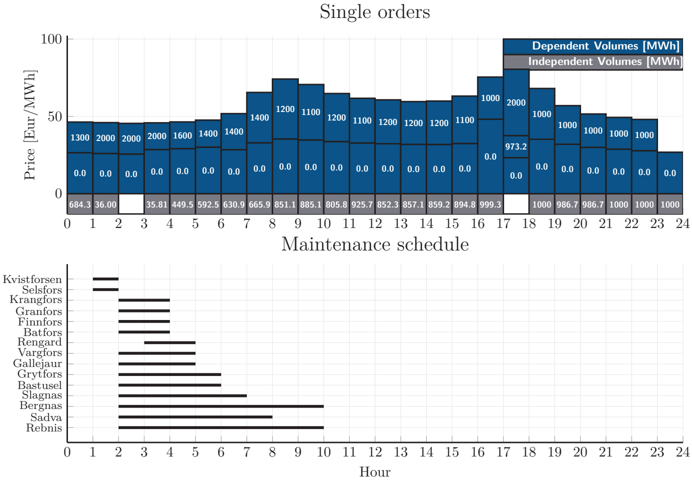
- Stable solution with 1000 scenarios
- Gap is larger compared to the day-ahead formulation

Maintenance scheduling - Strategies



Order strategy and maintenance schedule from stochastic planning

Maintenance scheduling - Strategies



Deterministic strategy from just considering the average market price

Outline

Capacity expansion - General description

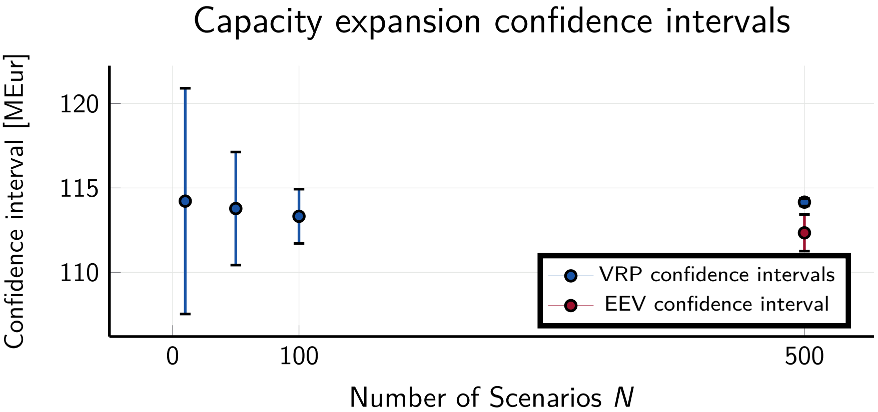
First stage

$$\begin{aligned} \underset{\text{Expansion}}{\text{maximize}} \quad & \mathbb{E}[R(\text{Expansion}, \text{Price}, \text{Inflow})] - \text{Cost}(\text{Expansion}) \\ \text{subject to} \quad & \text{Maximum expansion} \end{aligned}$$

Second stage

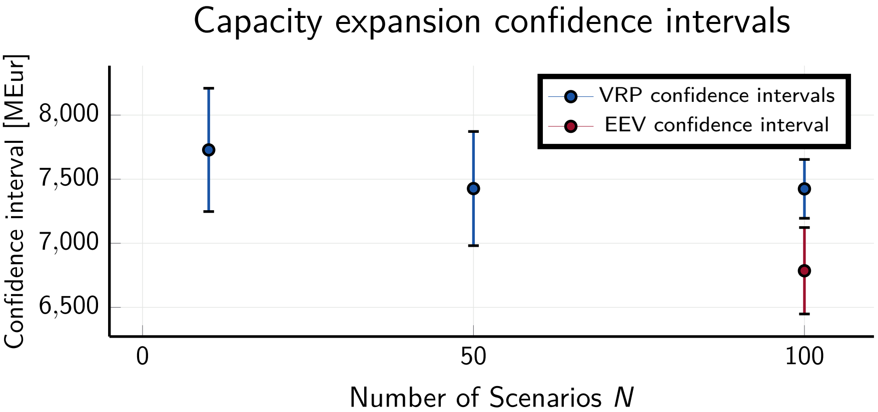
$$\begin{aligned} R = \max_{\text{Production}} \quad & \text{Profit}(\text{Price}) \\ \text{s.t.} \quad & \text{Hydrological balance}(\text{Inflow}) \\ & \text{Electricity production}(\text{Expansion}) \\ & \text{Load balance} \end{aligned}$$

Capacity expansion - Convergence



- 1 year horizon
- Stable solution with 500 scenarios

Capacity expansion - Convergence



- 20 year horizon
- Exceeds hardware capacity after 100 scenarios

Outline

Conclusion

Summary

- Efficient distributed stochastic programming methods

Conclusion

Summary

- Efficient distributed stochastic programming methods
- `StochasticPrograms.jl`: Julia software framework

Conclusion

Summary

- Efficient distributed stochastic programming methods
- `StochasticPrograms.jl`: Julia software framework
- Performance improvements of structure-exploiting algorithms

Conclusion

Summary

- Efficient distributed stochastic programming methods
- `StochasticPrograms.jl`: Julia software framework
- Performance improvements of structure-exploiting algorithms
- Effectiveness of the framework illustrated with three case studies