Stochastic Programming in Energy: Theory vs. Practical Application

Verbund

Andreas Eichhorn
VERBUND – Austrian Power Trading AG

6th Annual ÖGOR-IHS-Workshop "Mathematische Ökonomie und Optimierung in der Energiewirtschaft"

Institute for Advanced Studies (IHS), Vienna, Austria

September 23, 2010

Overview

- **Introduction:** medium-term optimization of power generation @ Verbund
- **Stochastic Programming**
  - (Mixed-integer) Linear Programming in power generation
  - 2-stage Stochastic Programming
  - Multi-stage Stochastic Programming
  - Scenario Trees
- *Application:* A scenario-tree based mean-risk model for optimal hedging
- **Dynamic Programming Approaches**
  - Dynamic Programming
  - Stochastic Dynamic Programming
  - Stochastic Dual Dynamic Programming (SDDP)
- *Application:* SDDP @ Verbund
- **Conclusion**

Introduction

Mathematical Optimization in the Power Industry

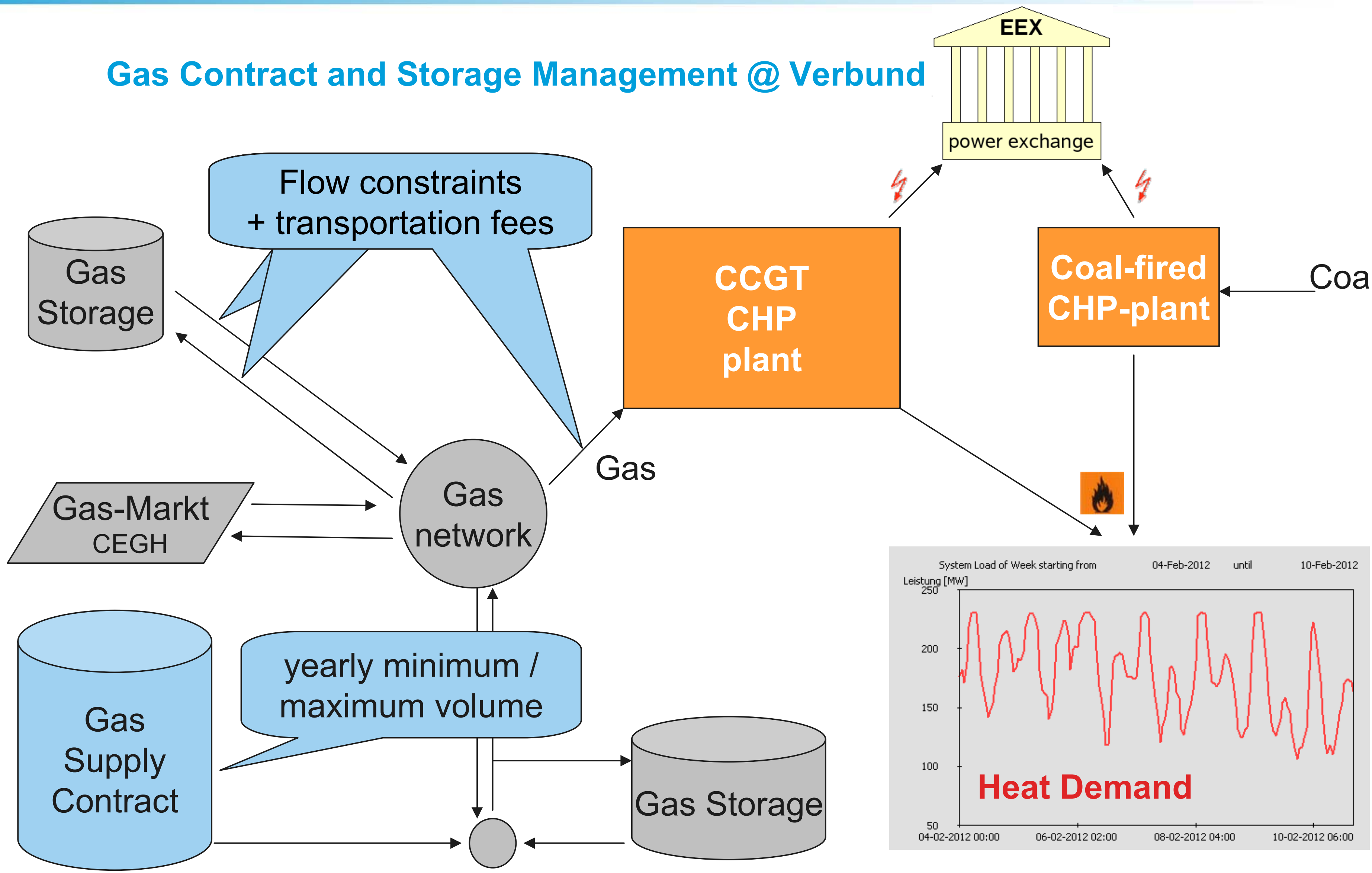
- Long-term planning (e.g. 30 years)
  - Investment decisions
  - Market models for price forecasting (→ marginal cost)
- Medium-term planning (few months up to few years)
  - **Rationing / managing of storable energy / limited recourses**
  - **Hedging** of power generation (i.e., future market sales)
  - **Hedging** of power sales contracts (i.e., future market purchases)
  - Financial planning
- Short-term planning
  - Actual power plant scheduling
  - Day-ahead bidding @ EPEX, intra-day redispatch

Introduction

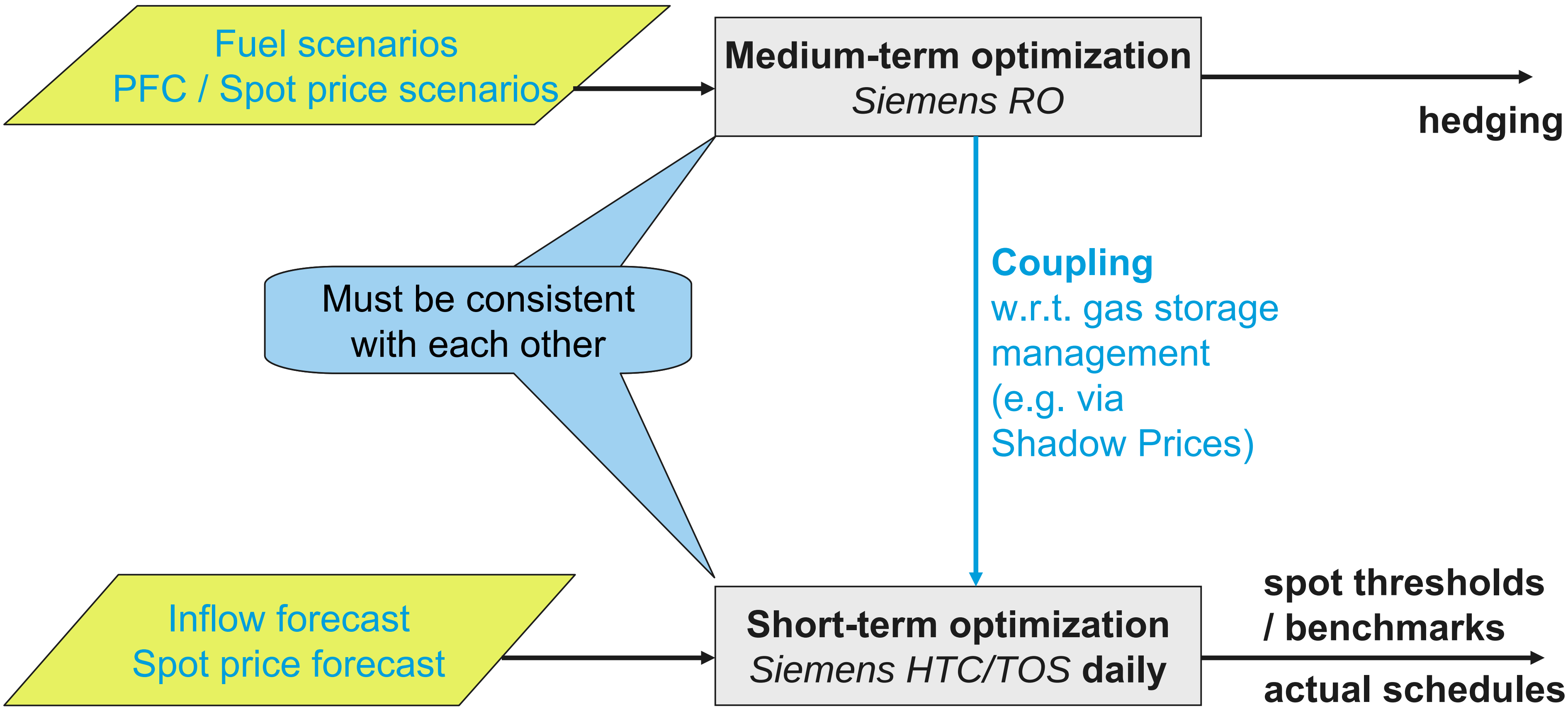
Power generation @ Verbund:

- Hydro power (≈ 90% in 2009)
  - Run-of-river plants
  - Hydro storage plants
- Other renewables
- Thermal

Gas Contract and Storage Management @ Verbund

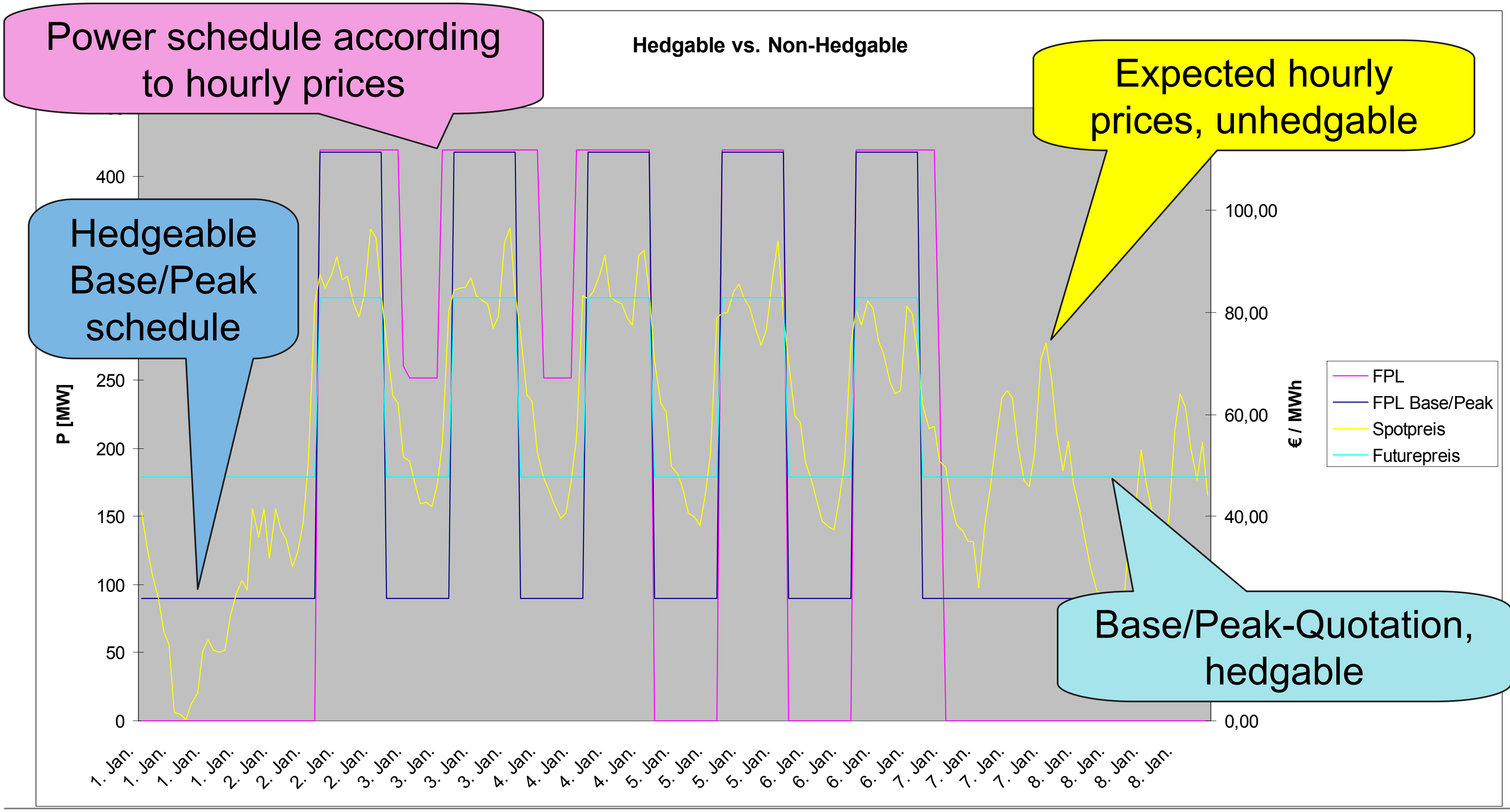


Thermal Generation Planning Process @ Verbund

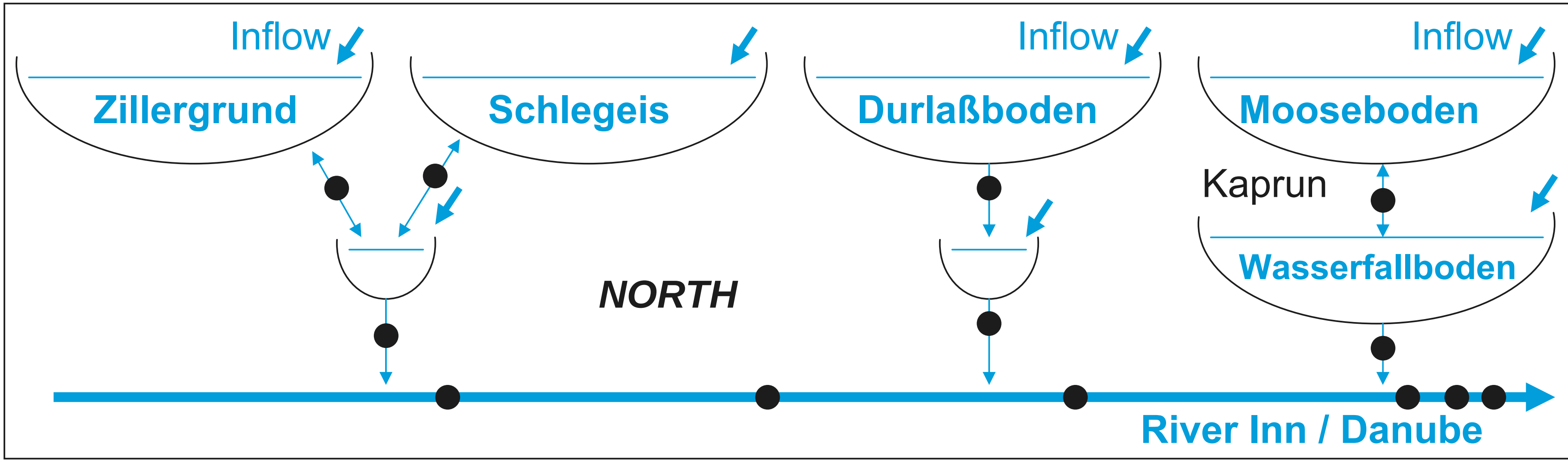


Hedging

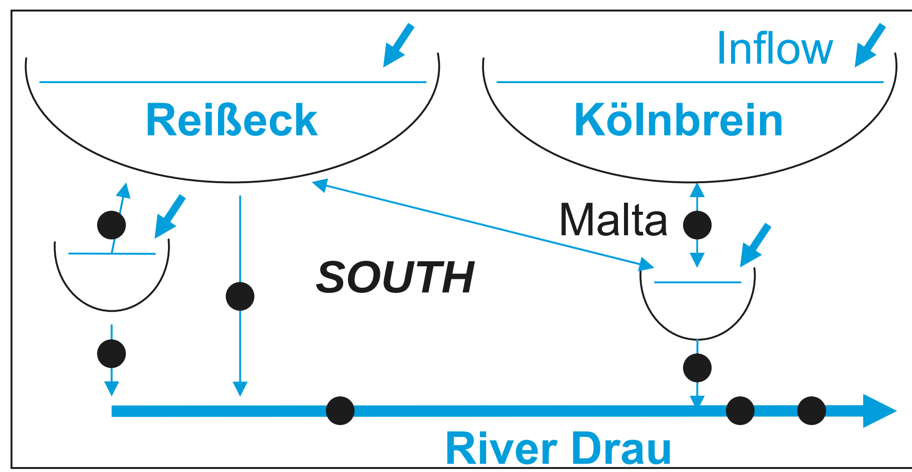
Spot prices are **uncertain**, but general price level can be “*locked in*”



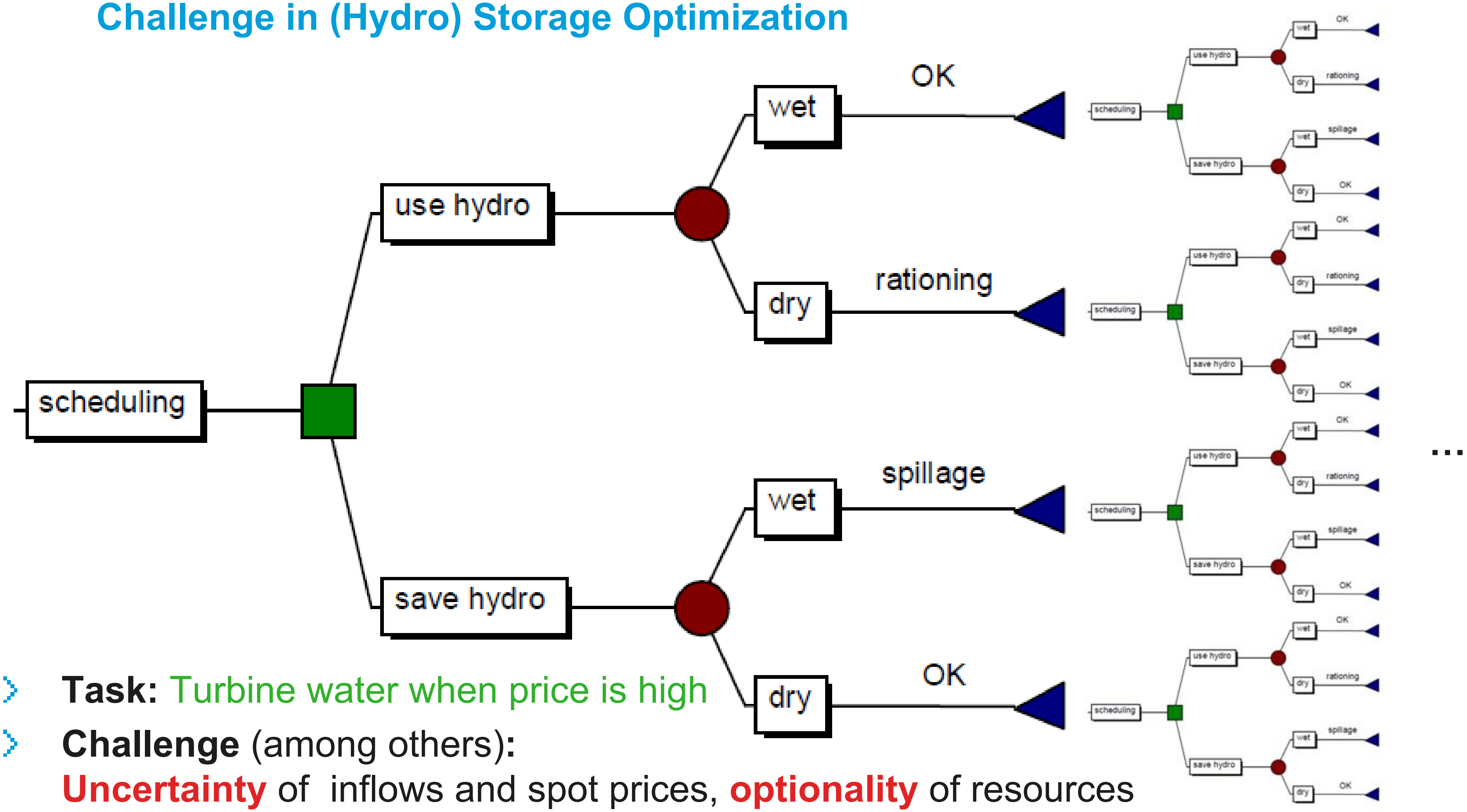
Hydro Storage Energy @ Verbund



- **7 big** (yearly) **storages**
  plus **many** other small (daily) storages, rivers, and run-of-river plants
- North and south system **coupled** via reserve responsibilities
- Seasonal hydrologic inflows,
  x TWh storage energy p.a.
  vs. ≈ x GW capacity (+ pumping)
  → **€€€**, worth to optimize

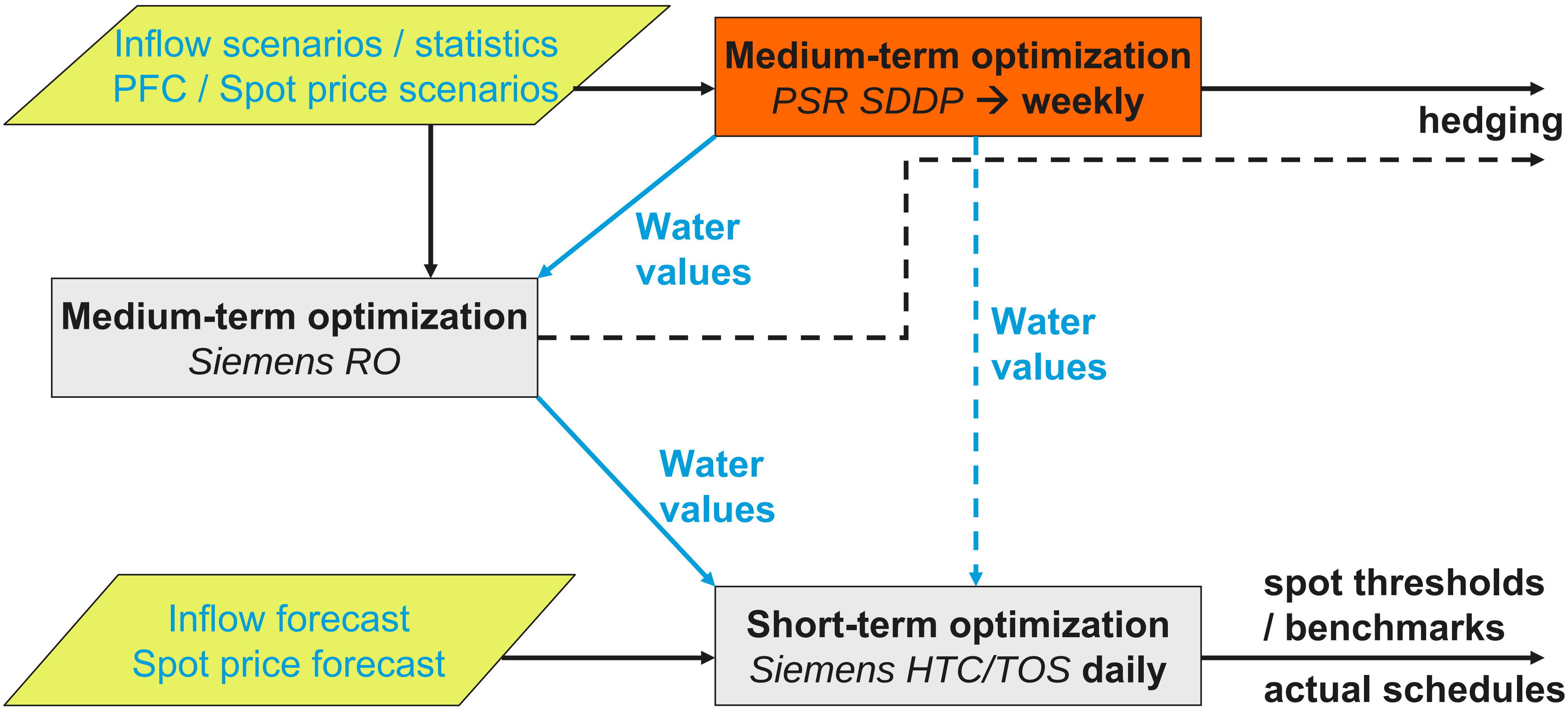


Challenge in (Hydro) Storage Optimization



- **Task:** Turbine water when price is high
- **Challenge** (among others):
  Uncertainty of inflows and spot prices, **optionality** of resources
  → [uncertainty tree] x [decision tree]
- **Hedging!** (= Selling power on forward market) but inflows cannot be hedged …

Hydro Storage Planning Process @ Verbund



Overview

- **Introduction:** medium-term optimization of power generation @ Verbund
- **Stochastic Programming**
  - (Mixed-integer) Linear Programming in power generation
  - 2-stage Stochastic Programming
  - Multi-stage Stochastic Programming
  - Scenario Trees
- *Application:* A scenario-tree based mean-risk model for optimal hedging
- **Dynamic Programming Approaches**
  - Dynamic Programming
  - Stochastic Dynamic Programming
  - Stochastic Dual Dynamic Programming (SDDP)
- *Application:* SDDP @ Verbund
- **Conclusion**

Mathematical Programming / Mathematical Optimization

(Mixed-integer) Linear Program:

Scalar product | Integer numbers | Componentwise

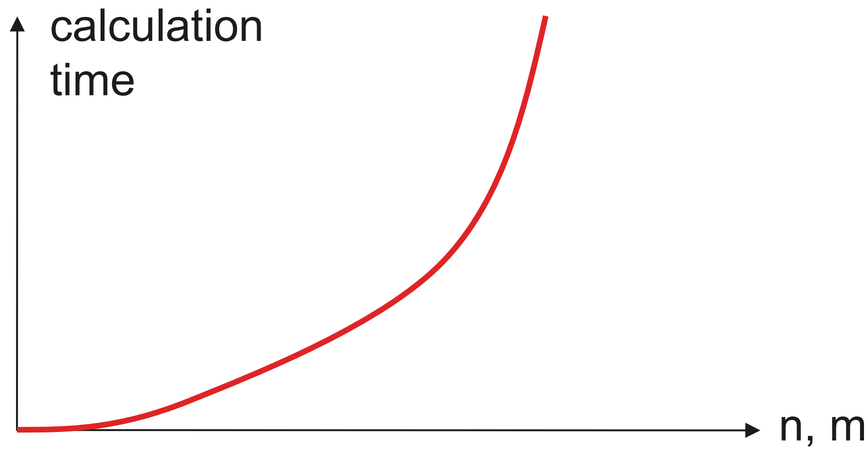
(LP) / (MIP)

$$\text{min. } c \cdot x \text{ over } x \in \mathbb{Z}^k \times \mathbb{R}^{n-k} \text{ subject to } Ax \geq b$$

$A \in R^{m \times n}$, $b \in R^{n}$ (right hand side vector), $c \in R^{m}$ (cost vector)

→ parameters, given!

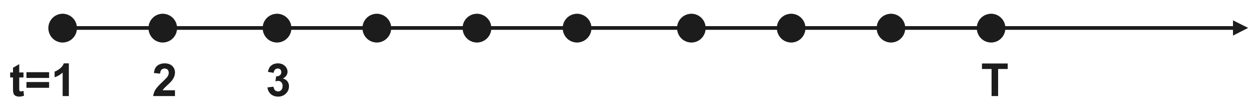
- Case k=0: **LP**,
  very efficient solution algorithms
- Case k>0: **MIP**, NP-hard,
  good heuristics for special problems



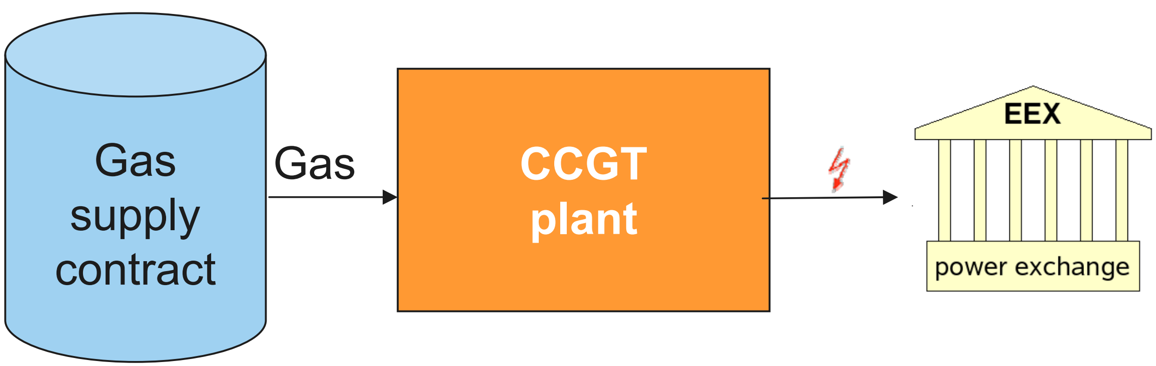
Mathematical Programming in Power Generation

- E.g., power plant scheduling, gas and hydro storage management
- Typical: **time structure**, e.g., hourly decisions (induced by hourly EPEX prices)
  → indexation $x_{t,c}$ instead of x_j where t=1,…,T and c refers to model component

$$\text{minimize} \quad \sum_{t=1}^{T} c_t' x_t \quad \text{over} \quad x_t \in \mathbb{R}^{n_t} \; (t=1,\ldots,T) \quad \text{subject to} \quad Ax \geq b$$



- E.g.:



Hourly price curve (forecast / PFC) → hourly schedule for power plant

Stochastic Programming:

- **Deterministic framework**
  Optimization / Solver has **perfect information** about all time steps
  (as if decision maker were clairvoyant)
  → somewhat unrealistic

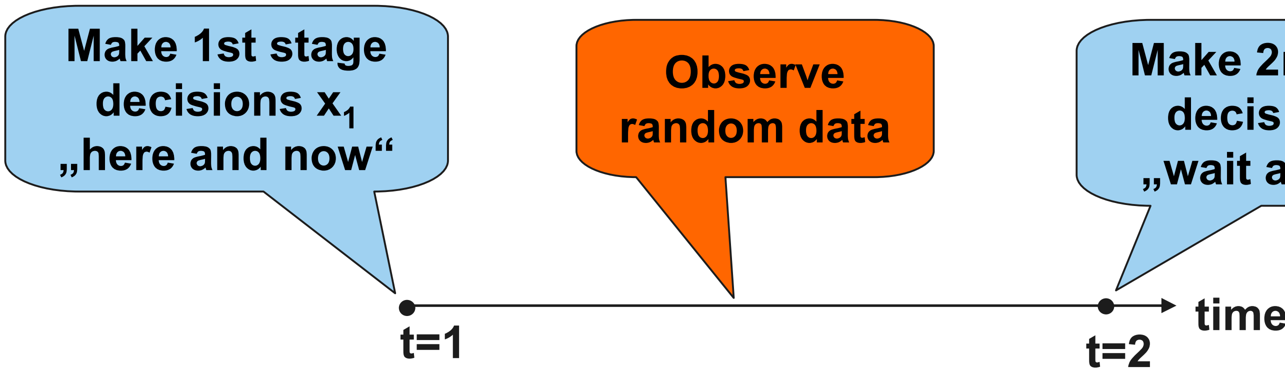
- „Scenario Analysis“ / „Monte Carlo“,
  i.e., repeated solution of LP / MIP with various c, A, b (scenarios)
  doesn‘t overcome this problem!

- Uncertainty **and** decision structure / information structure
  have to be modeled properly

→ **„Stochastic Programming“,** „Robust Optimization“, „Online Optimization“, **…**

Two-stage Stochastic Programming

Decision Structure:



LP / **MIP** →

- Some data (some components of c, A, b) may be uncertain (random)
- Some decisions may depend on outcome of randomness and some must not!
  → Segmentation of decision vector x into $x=(x_1,x_2)$
  - x_1: here-and-now decisions – one decision for all possible realizations (non-anticipative) but may depend on the probability distribution of the random data (prospective)
  - x_2: wait-and-see decisions – decision depending on outcome / realization
- Example: day-ahead spot auction @ EPEX

Verbund

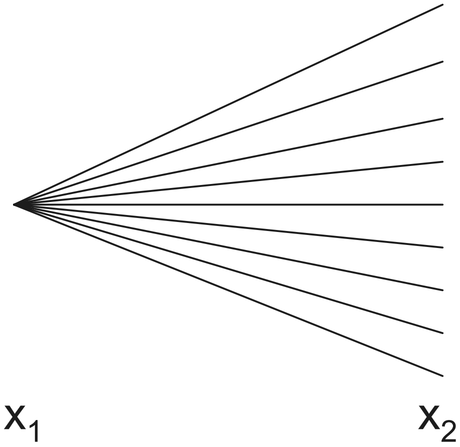
Two-stage Stochastic Programming:

(SP)

$$\text{minimize} \quad E\left[c'\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}\right] \quad \text{over} \quad x_1 \in \mathrm{R}^{n_1}, x_2 = x_2(c, A, b) \in \mathrm{R}^{n_2}$$

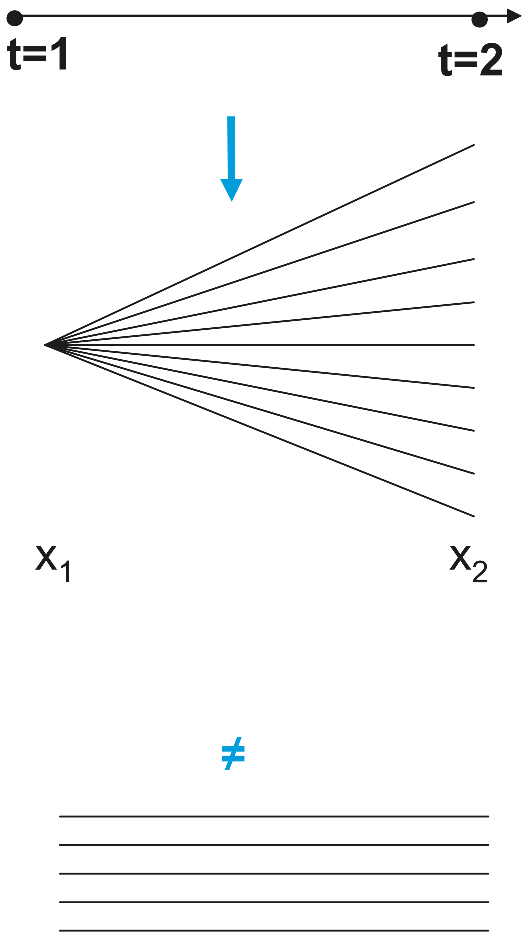
$$\text{subject to} \quad A\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \geq b \text{ for all possible realizations of } A, b$$

- E[.] = **expected value** functional (linear) as objective, alternatively: **risk measure**, e.g., VaR(.) / AVaR(.) → **minimizing risk** (nonlinear) → “Markowitz”
- x_2 is a **random vector** (to be chosen optimally)
- **In practice**: distribution of (c, A, b) must be finite (or be approximated by a finitely many **scenarios**)
  → distribution of x_2 is also finite
  → can be solved by ordinary **LP** / **MIP** solver



- **Limitation:** Decisions must not affect randomness

Two-stage Stochastic Programming:



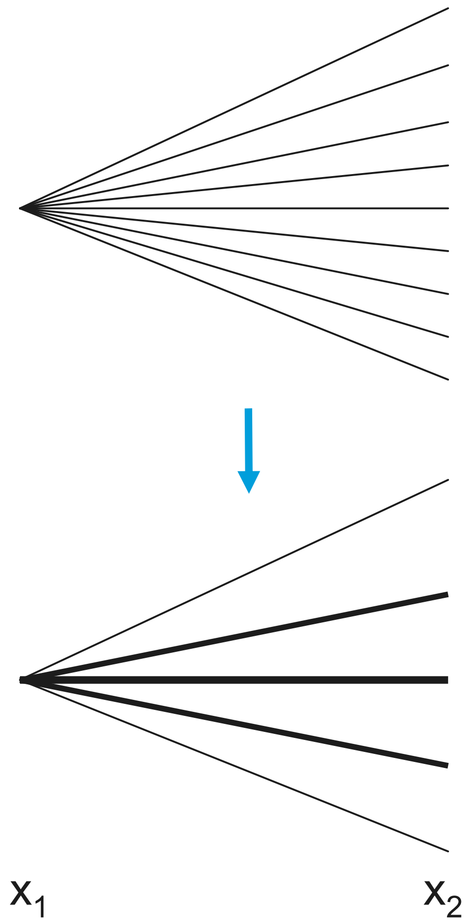
$$A = \begin{pmatrix} \blacksquare & \blacksquare \end{pmatrix} \qquad A \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \geq b$$

- Obtain **scenarios** from samples / quatization from stoch. Model
- For each scenario a “copy” of x_2 has to be introduced
- x_1 couples different scenarios

$$A = \begin{pmatrix} \blacksquare & \blacksquare & 0 & \cdots & 0 \\ \blacksquare & 0 & \blacksquare & & \vdots \\ \blacksquare & 0 & & \ddots \quad \blacksquare & 0 \\ \blacksquare & 0 & \ldots & 0 & \blacksquare \end{pmatrix} \qquad x = \begin{pmatrix} x_1 \\ x_2^1 \\ \vdots \\ x_2^S \end{pmatrix}$$

Verbund

Two-stage Stochastic Programming:



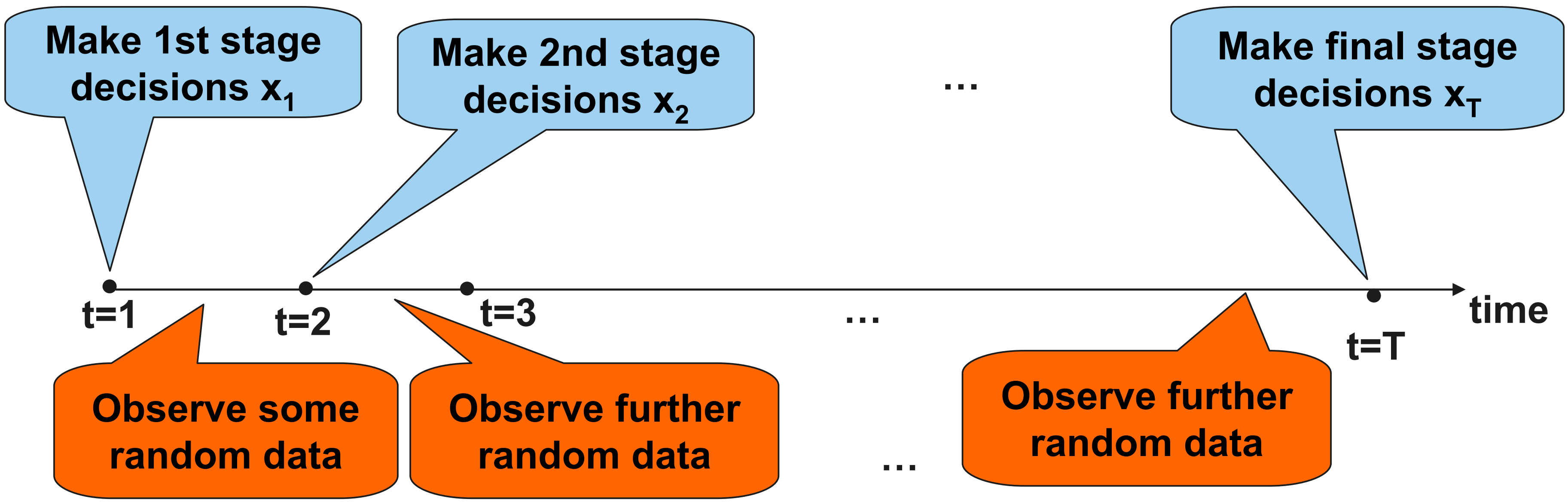
- Use “**scenario reduction**” techniques to select **representative scenarios** !
  E.g. methods by [Heitsch/Römisch]

  available in GAMS (“**scenred**”)

- Based on **Stability theory,** approximation such that solution of stochastic program changes only little

Multi-stage Stochastic Programming

Iteration / generalization of two-stage **SP** (2 → T) - dynamic decision process:



- Segmentation of decision vector x into $\mathbf{x=(x_1,x_2,x_3,\ldots,x_T)}$
- Each $\mathbf{x_t}$
  - may depend on random outcome observed until time t
  - must not depend on random outcome observed after time t (non-anticipativity)
  - may depend on (conditional) probability distribution of all random data (prospectivity)

Verbund

Multi-stage Stochastic Programming:

(SP) minimize $E\left[\sum_{t=1}^{T} c_t' x_t\right]$

over $x_t = x_t\left(c_1,\dots,c_t, A_{1,0},\dots,A_{t,s}\ (s=0,\dots,t-1), b_1,\dots,b_t\right) \in \mathrm{R}^{n_t}\quad (\mathrm{t}=1,\dots,\mathrm{T})$

subject to $\sum_{s=0}^{t-1} A_{t,s} x_{t-s} \geq b_t\quad (\mathrm{t}=1,\dots,\mathrm{T})$ for all possible $b_t, A_{t,s}\ (s=0,\dots,t-1)$

Theoretically:

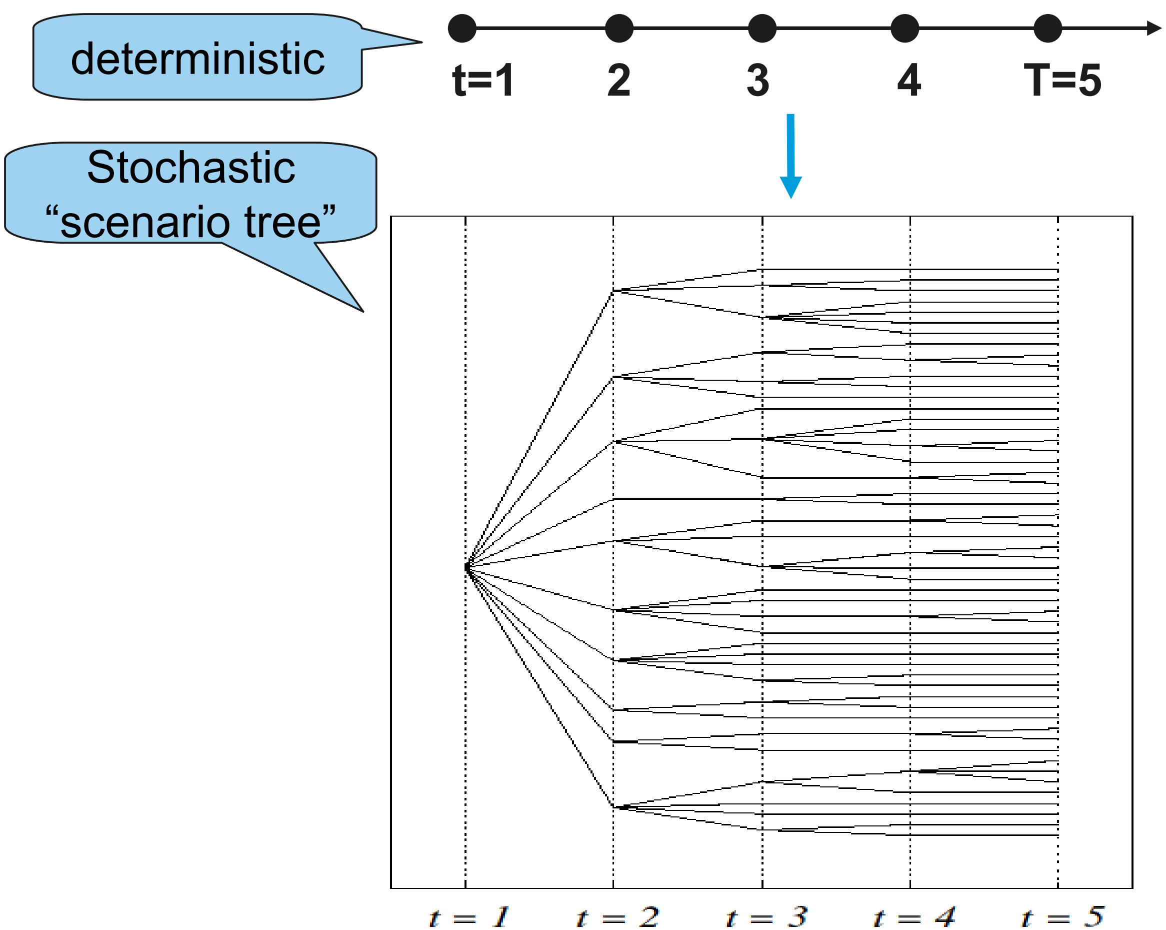
- Universal framework, copes with all kinds of stochastic processes and **optionalities**
- E[.] **expected value** functional (linear) as objective, alternatively:
  risk measure, e.g., VaR(.) / AVaR(.) → **minimizing risk** (nonlinear) → “Markowitz”
- stochastic input data (c_t, $A_{t,s}$, b_t) (t=1,…,T) is a **stochastic process**

In practice:

distribution of stochastic input process must be finite
(or be approximated by a finite one)

→ **Scenario tree**

Multi-stage Stochastic Programming:



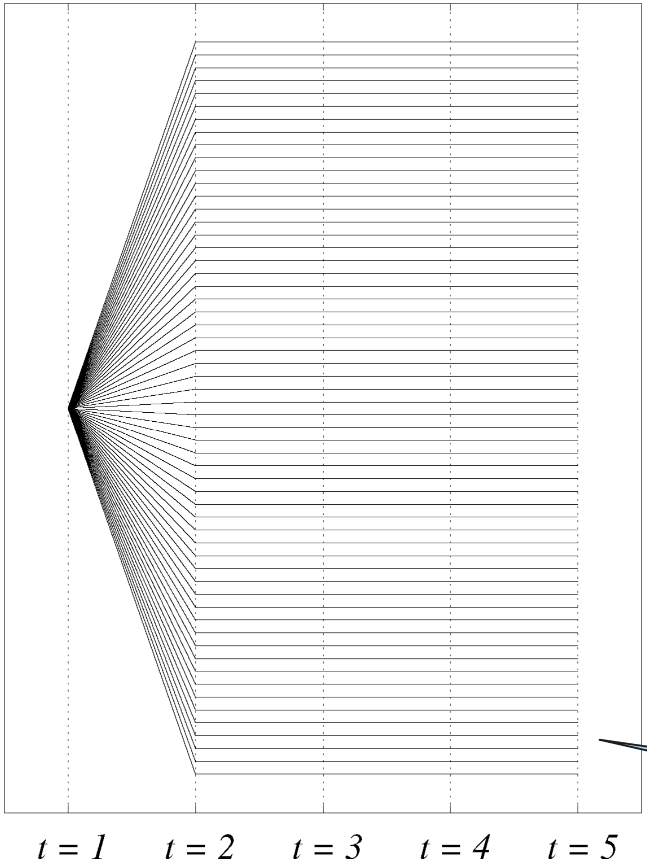
- Branching in tree represents **consecutive uncertainty**
- **Structure of input** data tree **= decision structure**;
  → introduce **copies of** $x_2, x_3, \ldots, x_T$
  → **ordinary LP / MIP**
- **scenario tree** gets big as T increases →
  "curse of dimensionality"

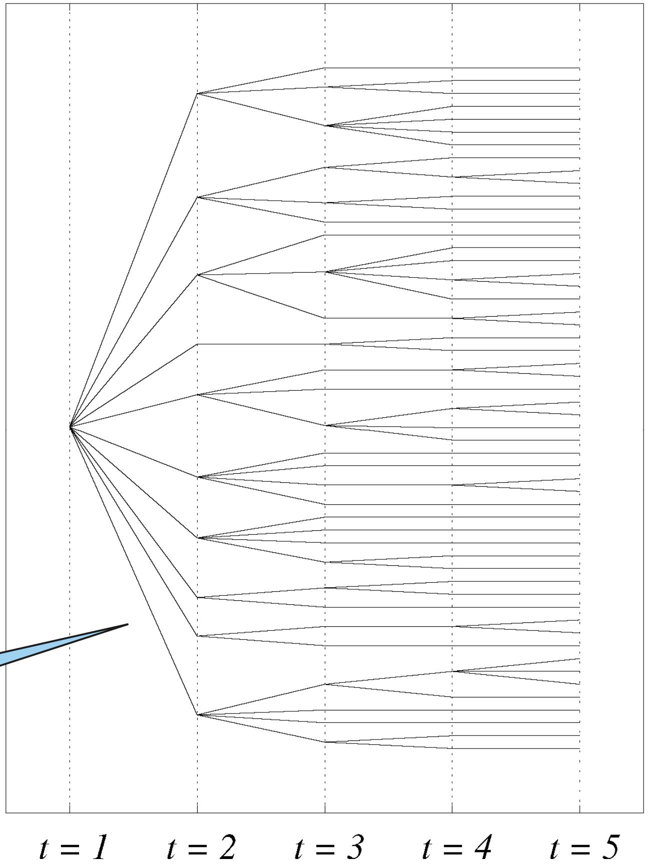
Verbund

Scenario Trees

available in GAMS (“**scenred**”)

Scenario tree generation via *Conditional Sampling* or ordinary sampling + **Scenario Tree Generation techniques** (e.g. [Heitsch/Römisch])



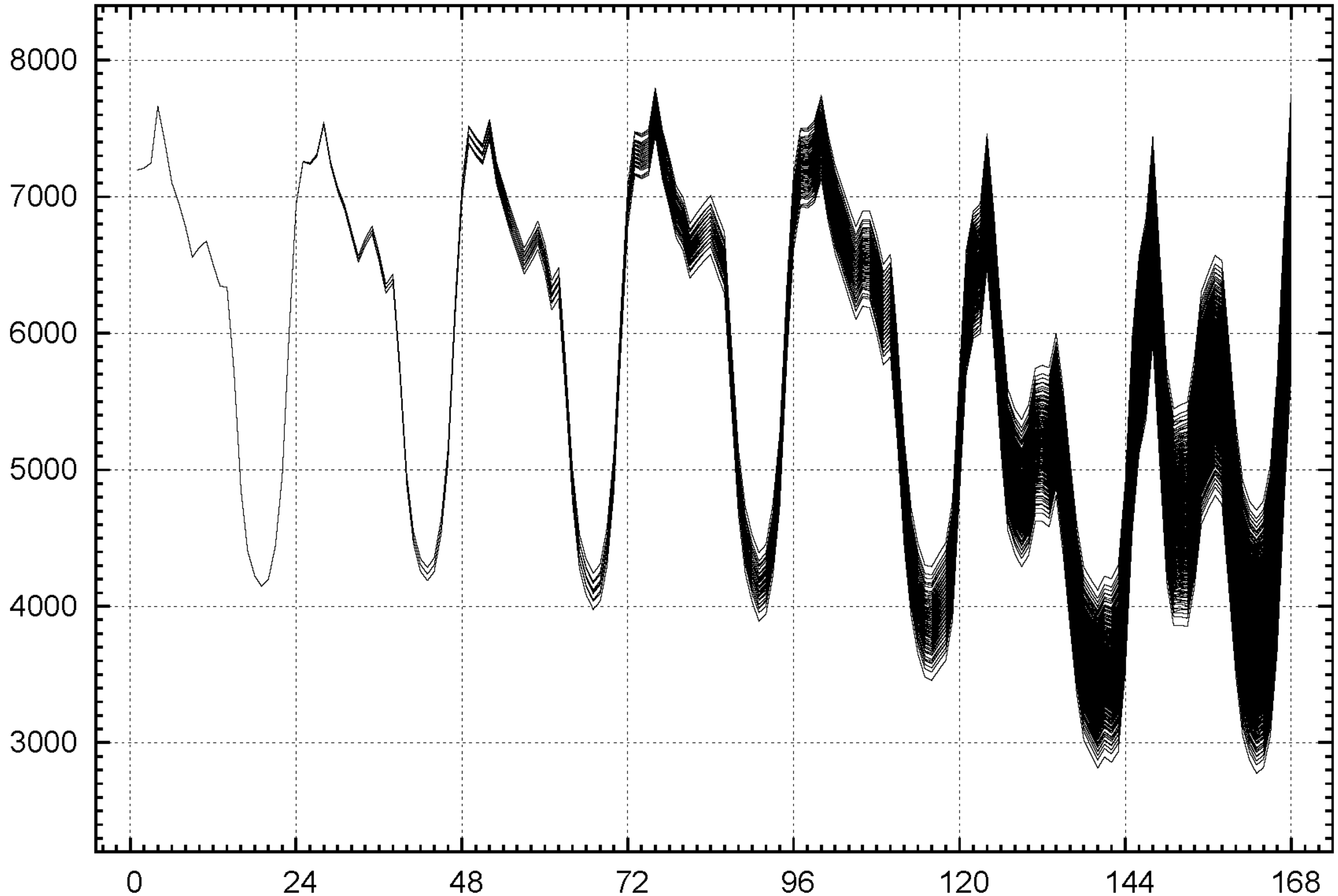


scenario tree

individual scenarios

Scenario Trees

Example: Some electricity demand, 1 week, branching every day



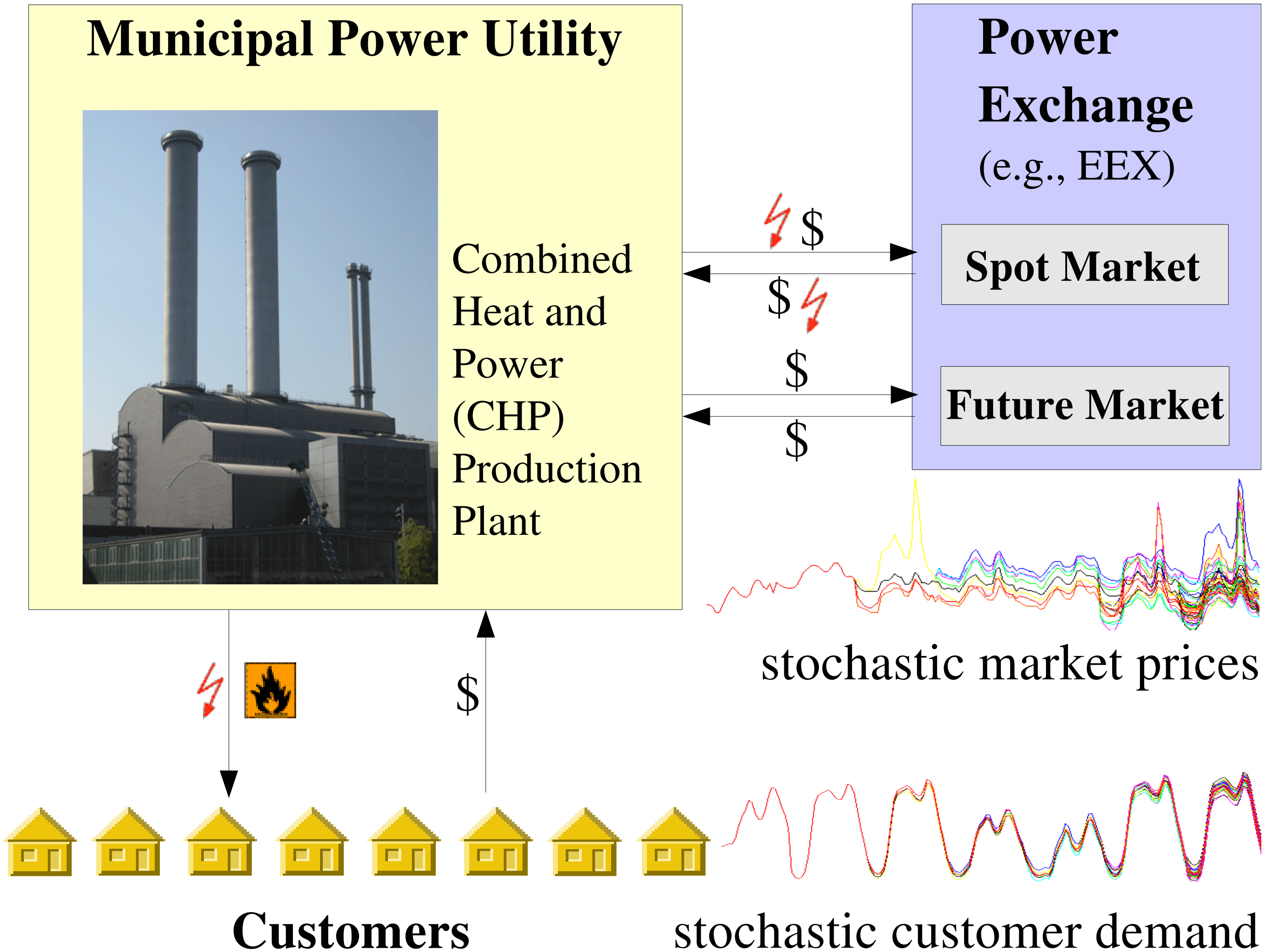
Overview

- **Introduction:** medium-term optimization of power generation @ Verbund
- **Stochastic Programming**
  - (Mixed-integer) Linear Programming in power generation
  - 2-stage Stochastic Programming
  - Multi-stage Stochastic Programming
  - Scenario Trees
- *Application:* A scenario-tree based mean-risk model for optimal hedging
- **Dynamic Programming Approaches**
  - Dynamic Programming
  - Stochastic Dynamic Programming
  - Stochastic Dual Dynamic Programming (SDDP)
- *Application:* SDDP @ Verbund
- **Conclusion**

A scenario-tree based mean-risk model for optimal hedging

Situation:

perspective of a small retailer with just one power plant



A scenario-tree based mean-risk model for optimal hedging

- **Purpose**: optimal Hedge-volumes to be sold here-and-now (rolling horizon for future actions)
- **Linear** Stochastic Program
- Optimization objective:

Accumulated revenue until t

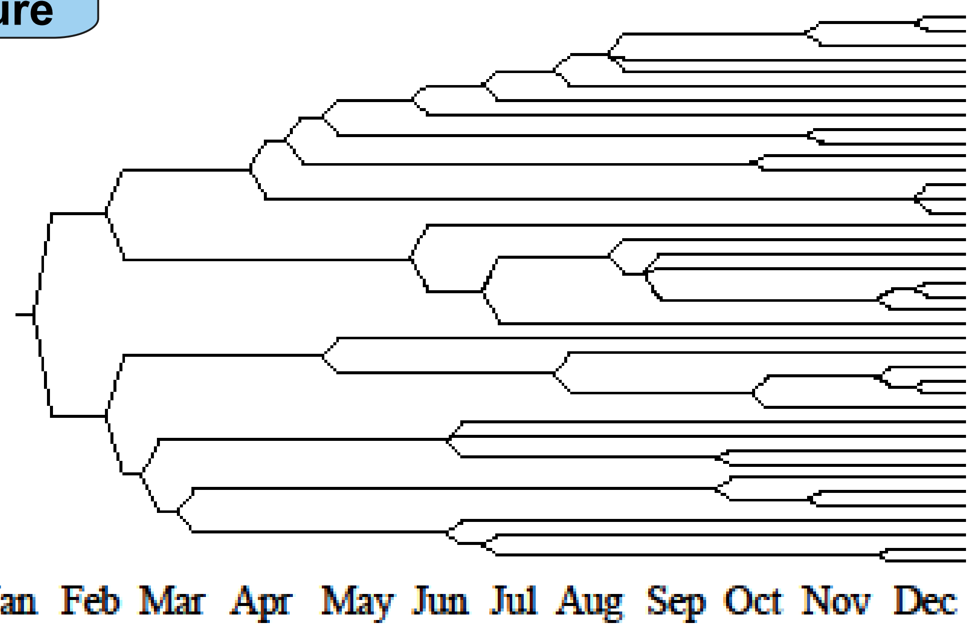
$$z_t := -\sum_{\tau=1}^{t} c_\tau ' x_\tau$$

$$\min 0.9 \cdot \rho(z_1, \ldots, z_t, \ldots, z_T) - 0.1 \cdot \mathbb{E}[z_T]$$

Risk Measure

Expected total revenue ("mean")

- Stochastic input data: given by **scenario tree** →
  (each node carries values for *heat* and *electr. demand*, *spot prices* and **arbitrage-free** *future prices*, respectively)
- t=1,…,T=8760
- Resulting **LP** solved with CPLEX



Verbund

A scenario-tree based mean-risk model for optimal hedging

Polyhedal risk measures

- Our optimization problem (abstract form):

$$\min_x \{\rho(z_1(x), \ldots, z_T(x)) \mid Ax \geq b\}$$

Risk Measure

Artificial Decision vector

- Definition of Polyhedral Risk Measures:

$$\rho(z_1, \ldots, z_T) = \min_y \left\{ \mathbb{E}\left[\sum_{t=1}^T \bar{c}_t \cdot y_t\right] \;\middle|\; Wy = \begin{pmatrix} z_1 \\ \vdots \\ z_T \end{pmatrix} \right\}$$

Artificial linear objective

- **Leads to →** i.e., a linear stochastic program

$$\min_{x,y} \left\{ \mathbb{E}\left[\sum_{t=1}^T \bar{c}_t \cdot y_t\right] \;\middle|\; \begin{array}{l} Ax \geq b \\ Wy = z(x) \end{array} \right\}$$

A scenario-tree based mean-risk model for optimal hedging

Polyhedal Risk Measures

- Example: **Average-Value-at-Risk** (AVaR = CVaR)

$$\mathrm{AVaR}_\alpha(z_T) = \frac{1}{\alpha}\int_0^\alpha \mathrm{VaR}_{\bar\alpha}(z_T)\,d\bar\alpha = \min_{r\in\mathbb{R}} \left\{ r + \frac{1}{\alpha}\mathbb{E}\left[(r+z_T)^-\right]\right\}$$

Evaluates only z_T (total profit)

- Example: **Multiperiod Extention of AVaR**

$$\rho(z_1,\ldots,z_T) = \min_{r\in\mathbb{R}} \left\{ r + \frac{1}{\alpha(T-1)}\sum_{t=2}^{T}\mathbb{E}\left[(z_t+r)^-\right]\right\}$$

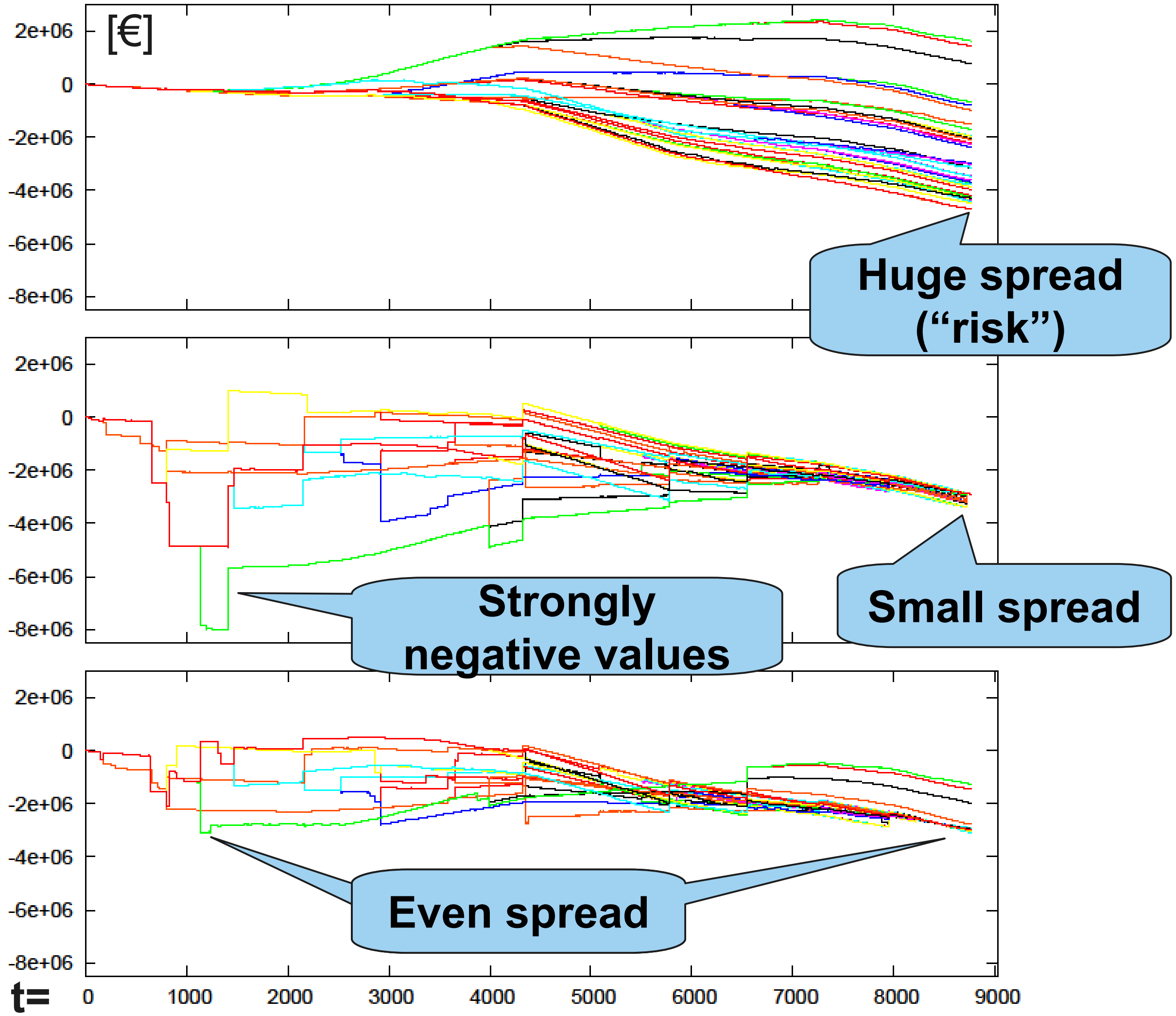
Evaluates entire sequence of revenues

A scenario-tree based mean-risk model for optimal hedging

Results:

Accumulated revenues

- No risk measure → (max. $E[z_T]$)
- $\text{AVaR}(z_T)$ →
- Multiperiod Polyhedral Risk Measure →



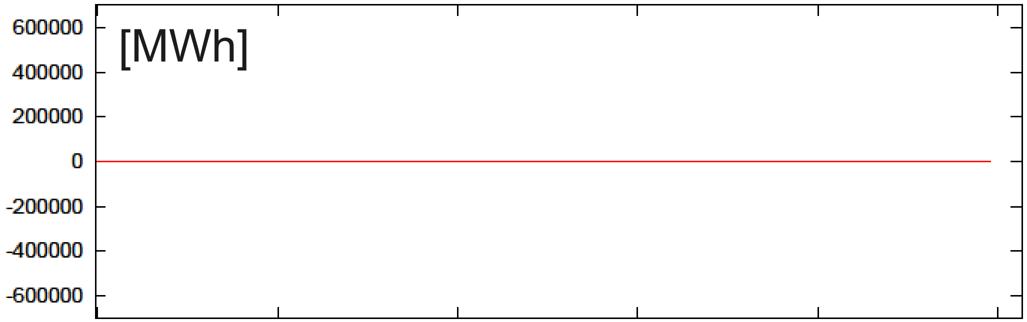
Verbund

A scenario-tree based mean-risk model for optimal hedging

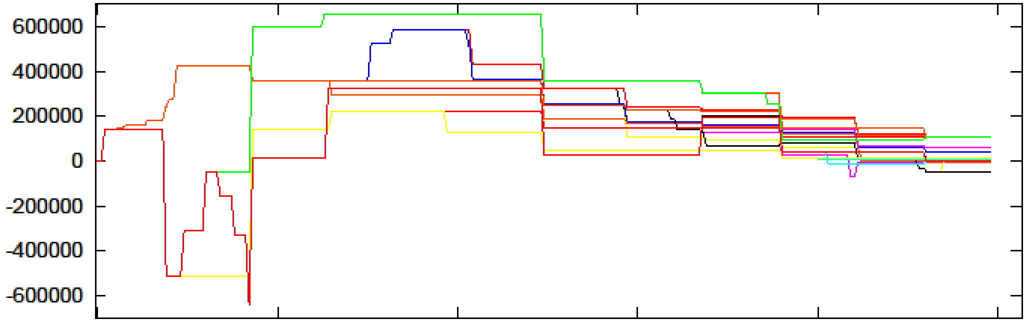
Results:

Total Future Stock

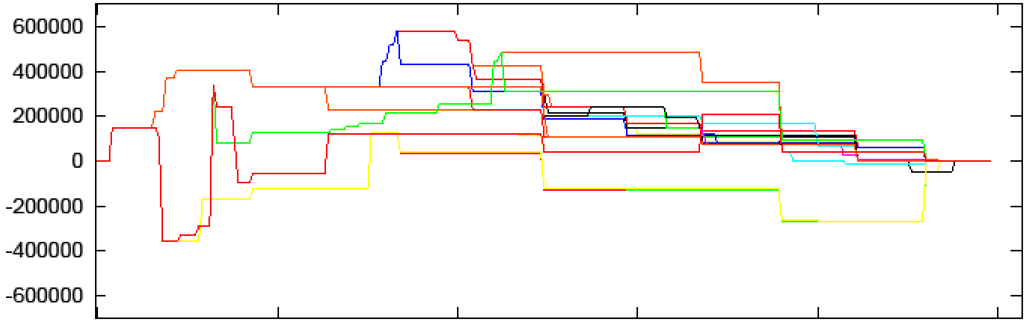
- No risk measure → (max. $E[z_T]$)



- $AVaR(z_T)$ →



- Multiperiod Polyhedral Risk Measure →



A scenario-tree based mean-risk model for optimal hedging

- Optimal Hedging according to risk measure

Limitation:

- Only LP
- Limited level of technical detail
- Limited number of components
- Only 40 scenarios (not much more is possible)
  → poor representation of uncertainty

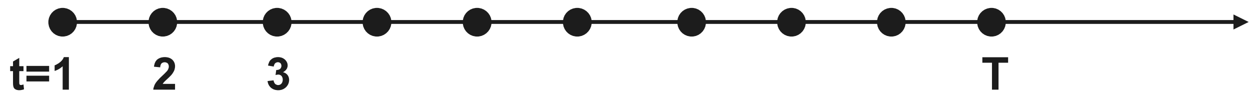
Overview

- **Introduction:** medium-term optimization of power generation @ Verbund
- **Stochastic Programming**
  - (Mixed-integer) Linear Programming in power generation
  - 2-stage Stochastic Programming
  - Multi-stage Stochastic Programming
  - Scenario Trees
- *Application:* A scenario-tree based mean-risk model for optimal hedging
- **Dynamic Programming Approaches**
  - Dynamic Programming
  - Stochastic Dynamic Programming
  - Stochastic Dual Dynamic Programming (SDDP)
- *Application:* SDDP @ Verbund
- **Conclusion**

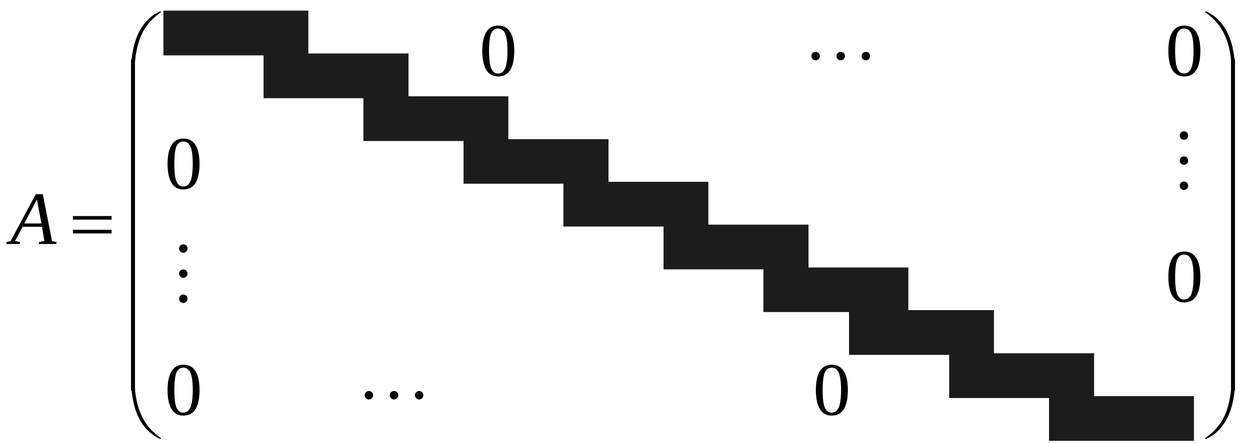
Mathematical Programming in Power Generation

- E.g., power plant scheduling, gas and hydro storage management
- Typical: **time structure**, e.g., hourly decisions (induced by hourly EPEX prices)
  → indexation $x_{t,c}$ instead of x_j where t=1,…,T and c refers to model component

$$\text{minimize} \quad \sum_{t=1}^{T} c_t' x_t \quad \text{over} \quad x_t \in \mathbb{R}^{n_t} \ (\mathrm{t}=1,\dots,\mathrm{T}) \quad \text{subject to} \ Ax \geq b$$



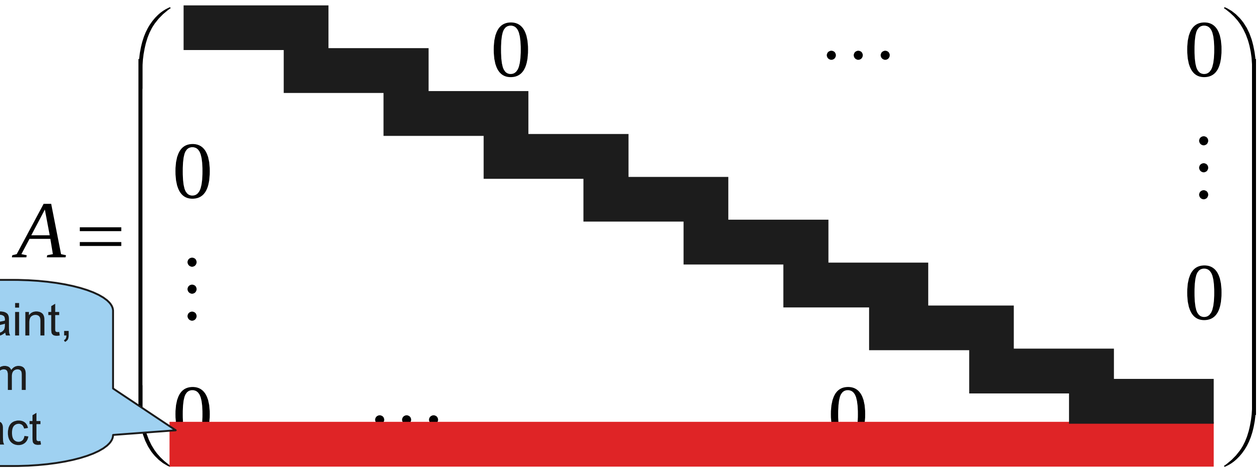
→ **sparse constraint matrix** [Ax ≥ b] with **staircase structure**:



$$x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_T \end{pmatrix} \qquad x_t = \begin{pmatrix} x_{t,1} \\ x_{t,2} \\ \vdots \\ x_{t,n} \end{pmatrix}$$

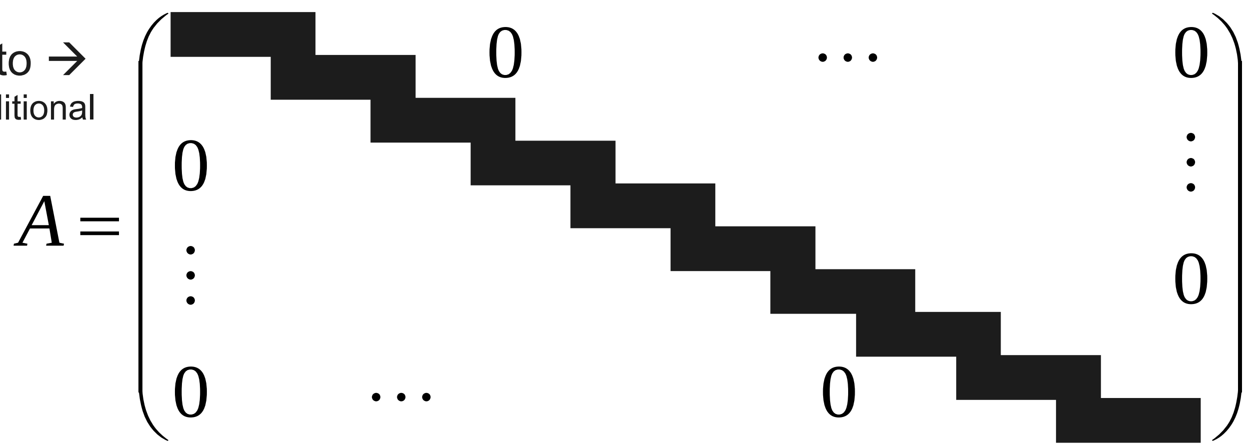
Mathematical Programming in Electricity

- Often also:



Time spanning constraint, e.g., yearly maximum volume of gas contract

- Can easily be transformed into → (by introducing additional decision variables)



Verbund

Dynamic Programming:

(LP) / (MIP)

$$\text{minimize} \quad \sum_{t=1}^{T} c_t' x_t \quad \text{over} \quad x_t \in \mathbb{R}^{n_t} \ (\text{t} = 1, \ldots, \text{T}) \quad \text{subject to} \ Ax \geq b$$

Staircase structure → possible to exploit the time structure → **recursive** solution of smaller sub-problems for each time step t:

$$C_t(x_{t-1}) = \min \left\{ c_t' x_t + C_{t+1}(x_t) \quad \text{over} \quad x_t \in \mathbb{R}^{n_t} \quad \text{subject to} \ A_t \begin{pmatrix} x_{t-1} \\ x_t \end{pmatrix} \geq b_t \right\}$$

Immidiate Cost ($c_t' x_t$)

Future Cost Function ($C_{t+1}(x_t)$)

block („stair“) of original matrix A (A_t)

where $C_{T+1} \equiv 0$

Bellman principle: Solving C_1 problem is equivalent to **(LP)** / **(MIP)**

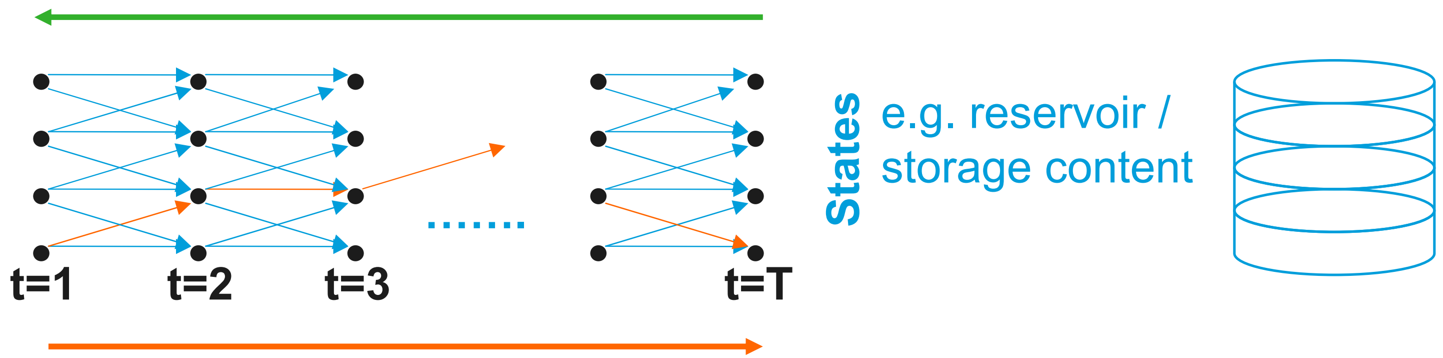
Dynamic Programming:

$$C_t(x_{t-1}) = \min\left\{ c_t' x_t + C_{t+1}(x_t) \quad \text{over} \quad x_t \in \mathbb{R}^{n_t} \quad \text{subject to} \;\; A_t \begin{pmatrix} x_{t-1} \\ x_t \end{pmatrix} \geq b_t \right\}$$

Algorithm:

start at t=T with given C_{T+1} (e.g. $C_{T+1} \equiv 0$) and proceed **backwards** t=T-1, T-2,…, 1

- Calculate (approximation of) **value function / future cost function $C_t(x_{t-1})$** for all possible (or some representative selection of) states



Finally, calculate solution x_t for t=1,…,T (**forward**) by using the calculated $C_t(.)$

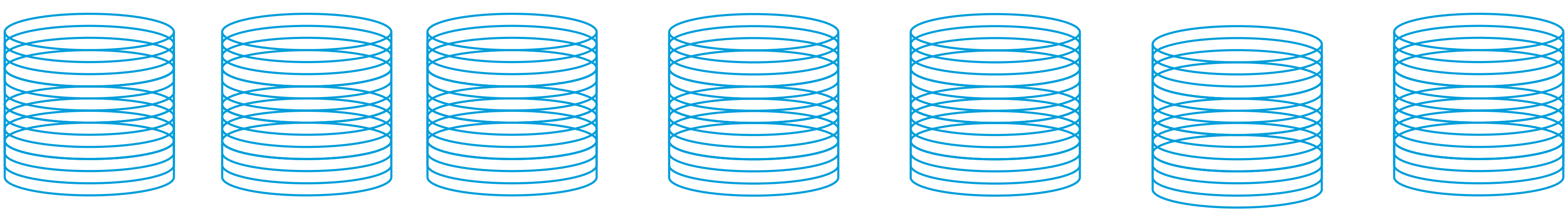
Dynamic Programming:

- C_t(.) can only be calculated for all possible states
  if [# possible states] resp. [# state variables] at time t is small

- **Numerical calculation effort:**

  > **Linear in T** (# time steps)
  > → major motivation since ordinary LP / MIP effort is superlinear in T

  > **Exponential in number of state variables („Curse of Dimensionality“)**
  > (→ often prohibitive, e.g. coupled multi-storage system)



Verbund

Stochastic Dynamic Programming

= Solving a **Multistage Stochastic Program** by **Dynamic Programming**

If **(SP)** has suitable staircase structure → equivalent to minimize C_1 where

$$C_t(x_{t-1}, \text{data}_{t-1}) = E_{\text{data}_t|\text{data}_{t-1}}\left[\min\left\{ c_t' x_t + C_{t+1}(x_t, \text{data}_t) \;\middle|\; \begin{array}{c} x_t \in \mathbb{R}^{n_t} \\ A_{t,0}x_t + A_{t,1}x_{t-1} \geq b_t \end{array} \right\}\right]$$

Conditional expectation

Immidiate Cost

Future Cost Function

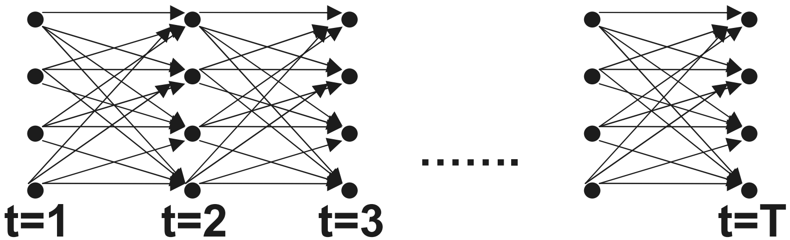
$\text{data}_t = (c_t, A_{t0}, A_{t1}, b_t)$

Can be applied only if

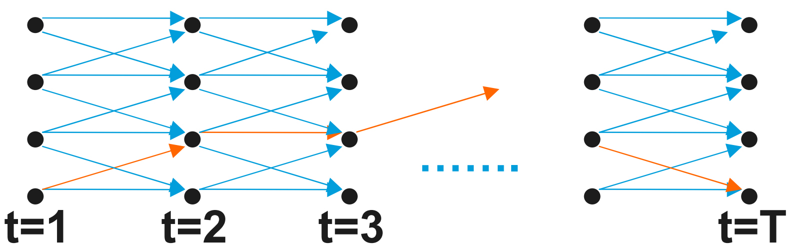
- Decision basis / system at time t can be described by few number of **states**
- Only few components of data is random and is **Markov** process (or AR_q)
  [respective past random data → **additional states**]

- **Example:** gas storage against spot

Stochastic Dynamic Programming

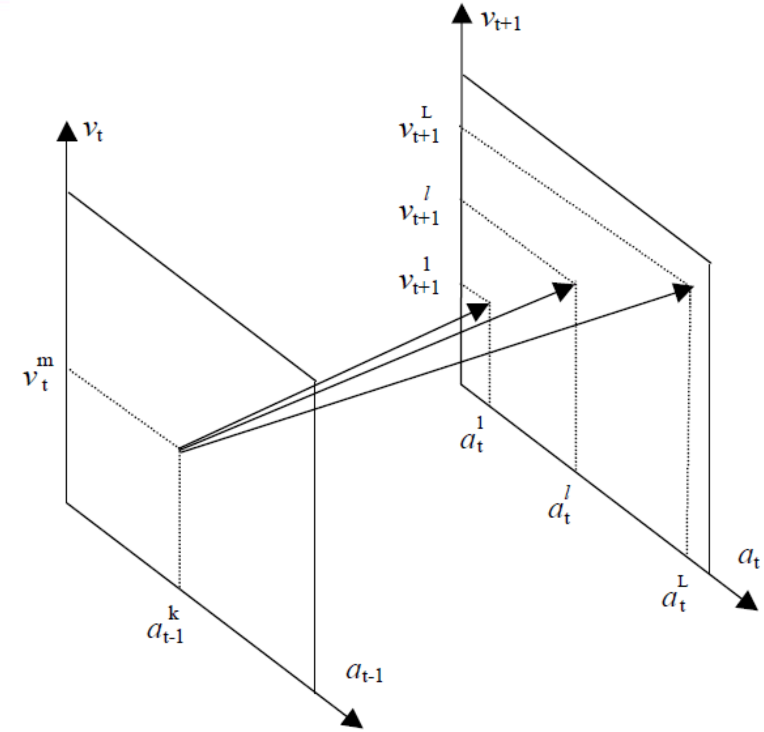


Markov process (e.g. spot prices) –
„**recombining scenario tree**“



States (e.g. storage contents)

Algorithm: Recursion
1 Backward pass to calculate C_t t=T,T-1,…,1
1 forward pass for optimal solution



Random components → states

- **Advantage:** Numerical calculation effort linear in T (# time steps)
- **Verbund Hydro system:** too many states and random components (prices and inflows for each reservoir)
  → **Curse of dimensionality …**

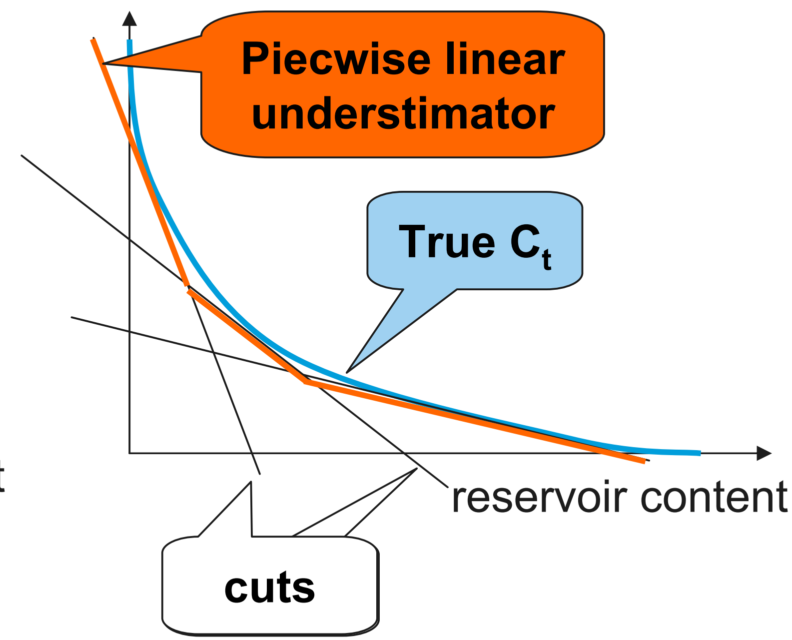
Verbund

SDDP – Stochastic Dual Dynamic Programming

Based on recursive formulation of multistage **stochastic programming**

$$C_t(x_{t-1}, \mathrm{data}_{t-1}) = E_{\mathrm{data}_t|\mathrm{data}_{t-1}}\left[\min\left\{c_t' x_t + C_{t+1}(x_t, \mathrm{data}_t) \;\middle|\; \begin{array}{c} x_t \in \mathbb{R}^{n_t} \\ A_{t,0}x_t + A_{t,1}x_{t-1} \ge b_t \end{array}\right\}\right]$$

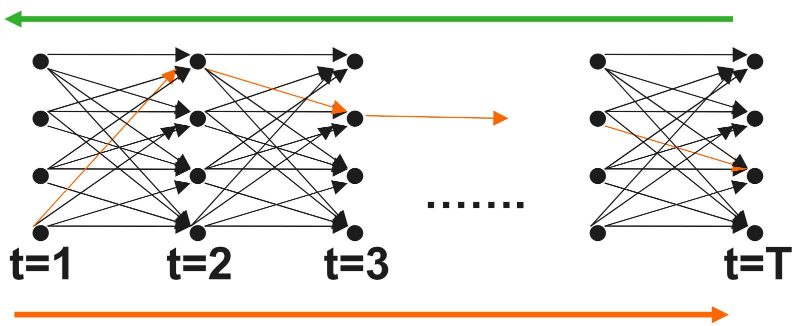
- **Key observation:** $C_t(.,.)$ is **convex** w.r.t. reservoir contents and inflows (b_t)
- **Idea:** Successive systematic approximation of $C_t(.,.)$ from below by **cut(ting plane)s** obtained via **shadow prices**
  Don‘t calculate $C_t(.,.)$ for **all** possible states!
- Replace C_{t+1} by x^C and add linear constraint [$x^C \ge \mathrm{cut}_n$] for each cut



SDDP Method

Algorithm:

- Fix recombinig scenario tree for spot prices and inflows as well as some sampled scenarios for the whole time horizon
- Set $C_t \equiv 0$ for t=1,…,T and initialize **reservoir contents associated to each scenario** and each t
- Loop n=1,2,…
  - start at t=T with and proceed **backwards** t=T-1, T-2,…, 1
    - **caluclate a further cut** at current content state for each scenario tree point by solving $\mathbf{C_t}$ problem with current cut approximation of $\mathbf{C_{t+1}(.,.)}$



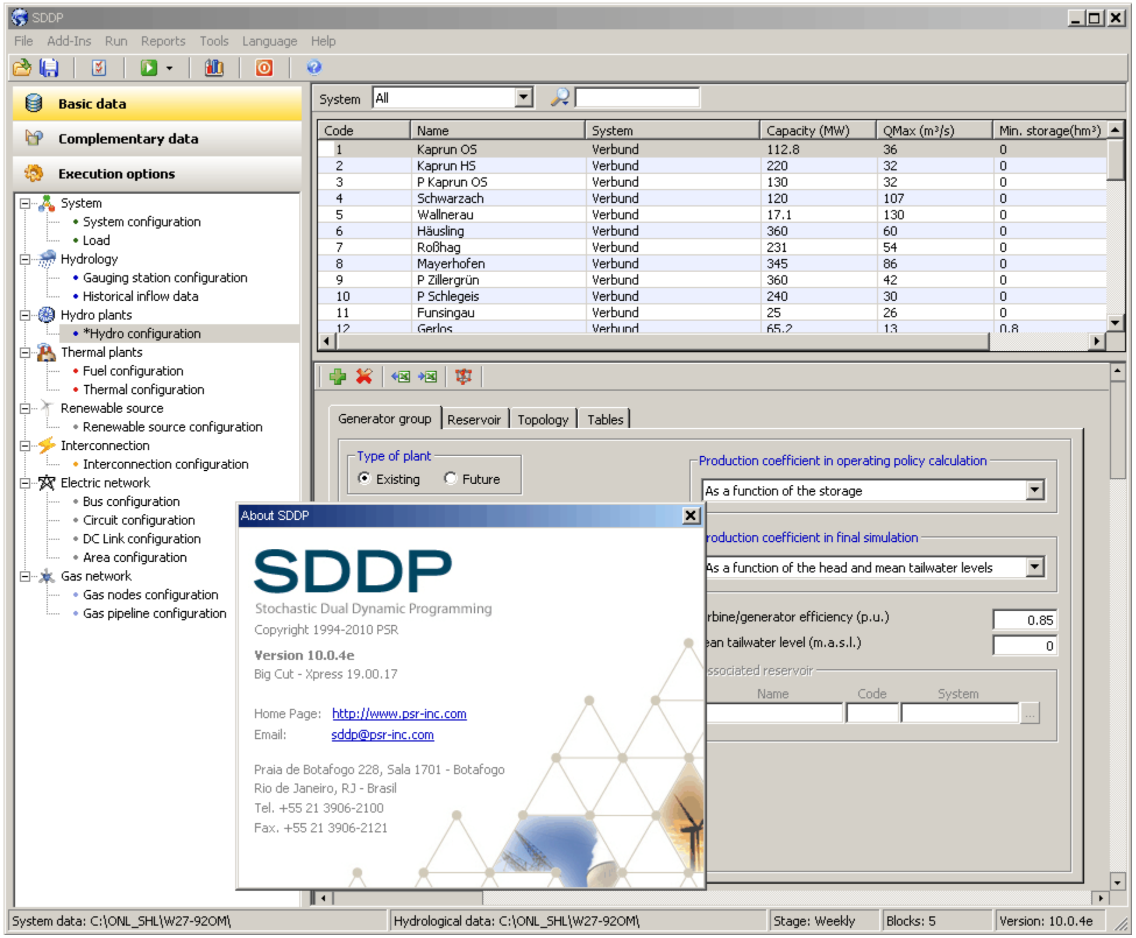
  - calculate solution for t=1,…,T (**forward**) by using the calculated $C_t(.)$ and update reservoir contents for each scenario and each t
- **Stop when reservoir contents don‘t change anymore**

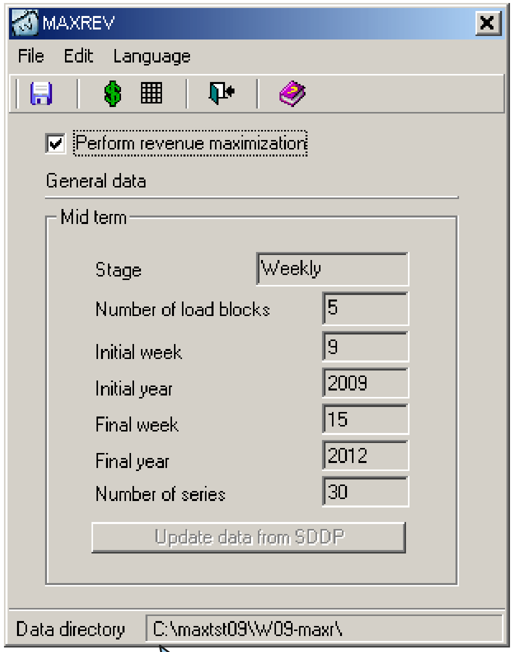
Overview

- **Introduction:** medium-term optimization of power generation @ Verbund
- **Stochastic Programming**
  - (Mixed-integer) Linear Programming in power generation
  - 2-stage Stochastic Programming
  - Multi-stage Stochastic Programming
  - Scenario Trees
- *Application:* A scenario-tree based mean-risk model for optimal hedging
- **Dynamic Programming Approaches**
  - Dynamic Programming
  - Stochastic Dynamic Programming
  - Stochastic Dual Dynamic Programming (SDDP)
- *Application:* SDDP @ Verbund
- **Conclusion**

SDDP Software

PSR SDDP / Maxrev (www.psr-inc.com) since 1986



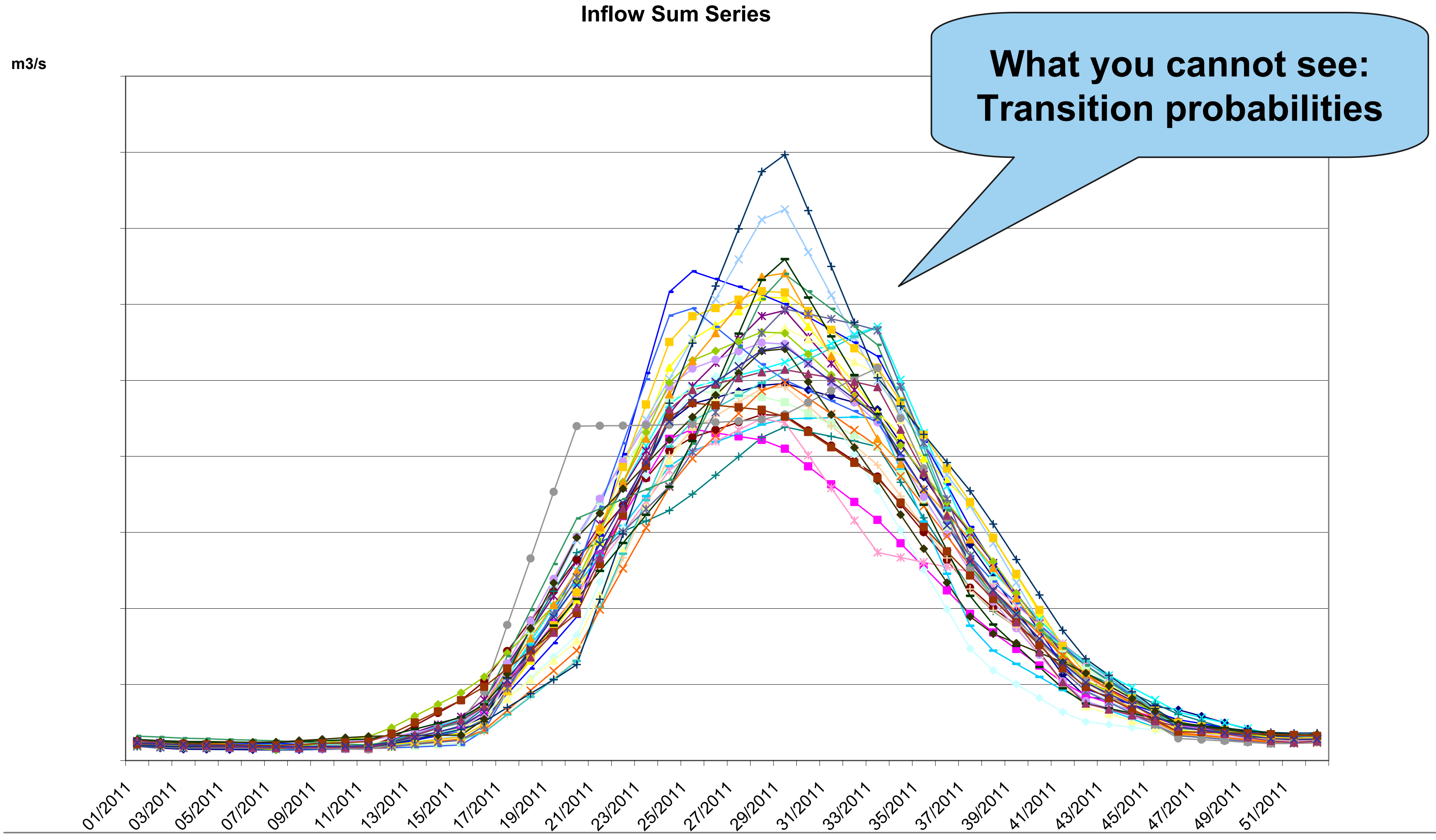


Add-in „Maxrev“ for handling stochastic prices

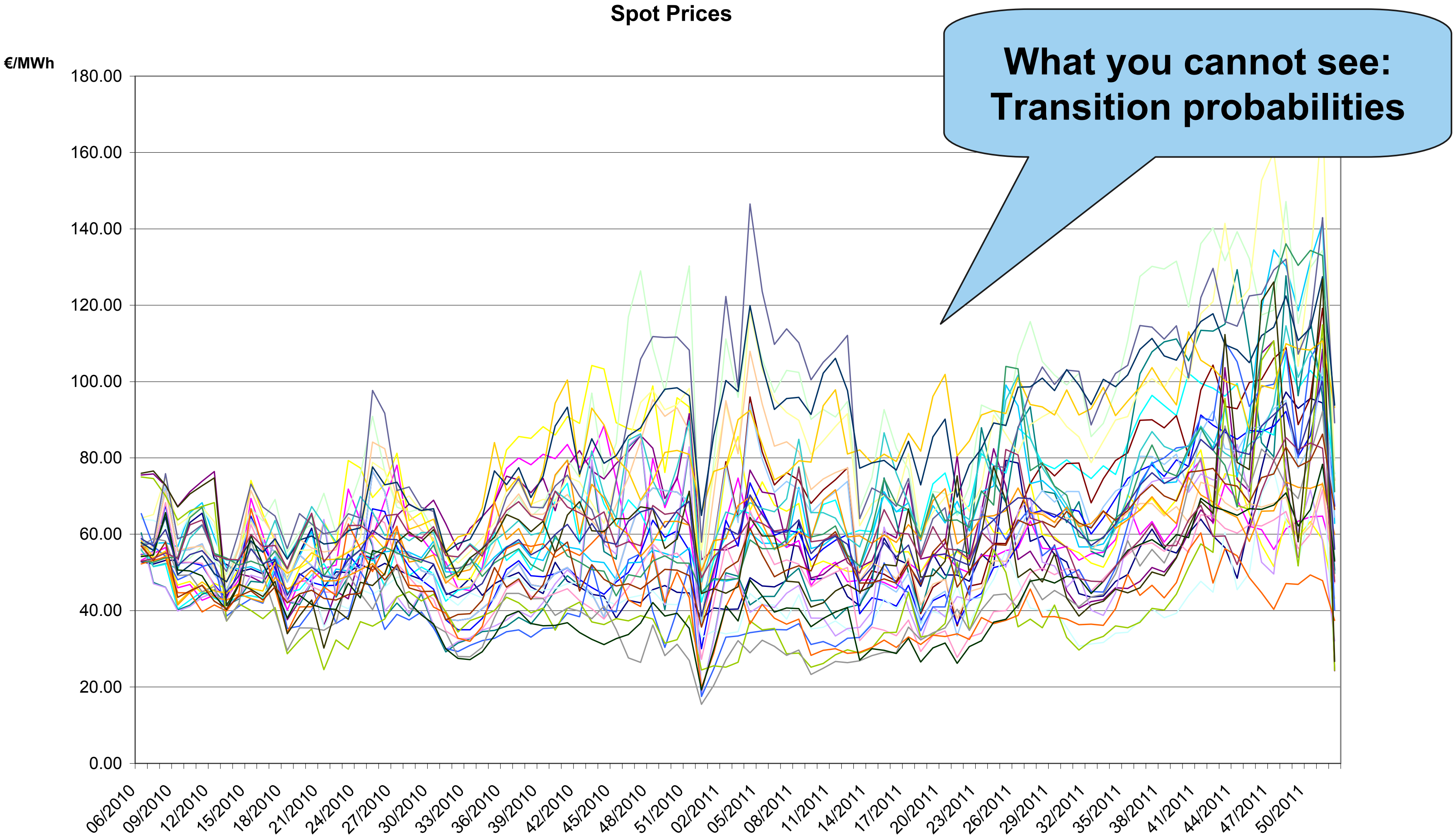
SDDP Software – Key characteristics:

- **Software for medium / long term energy planning** (hydro power, thermal power, power grid, gas network, … focus on hydro storage energy!)
- **Weekly time grid** (t → one week)
  intra-week structure considered by using 5 price segments („blocks“)
- Heuristic consideration of head level differences
- **Hydrologic Inflow stochastics**: AR_q (auto regressive) where $q \leq 6$
- **Price stochastics**: must be **Markov** (can be deduced from given scenarios by clustering method)
- **Parallel computing** possible (we use 8 CPU computer with Win XP 64bit)
  Cloud computing possible
- **Excel interface**

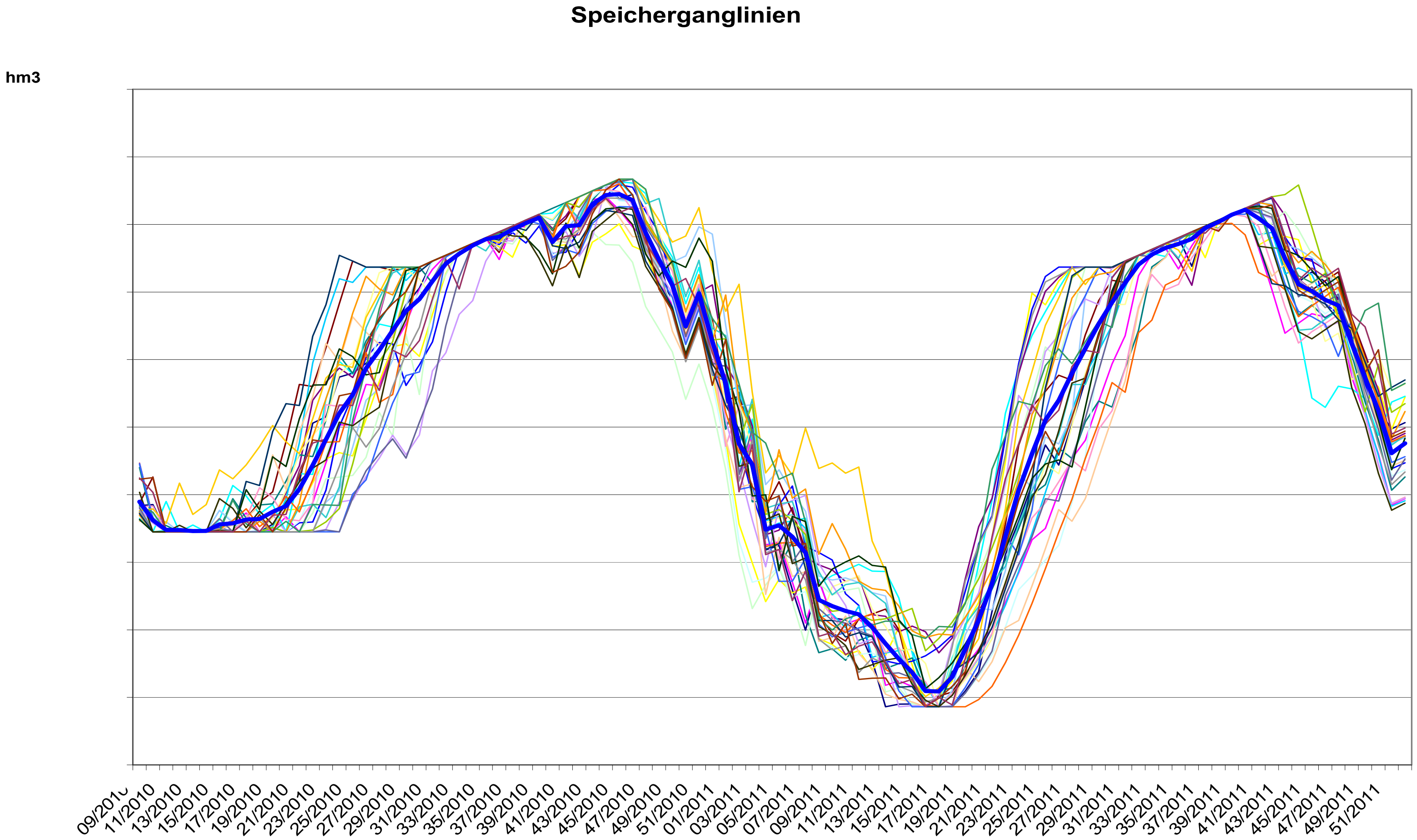
SDDP Software – Input



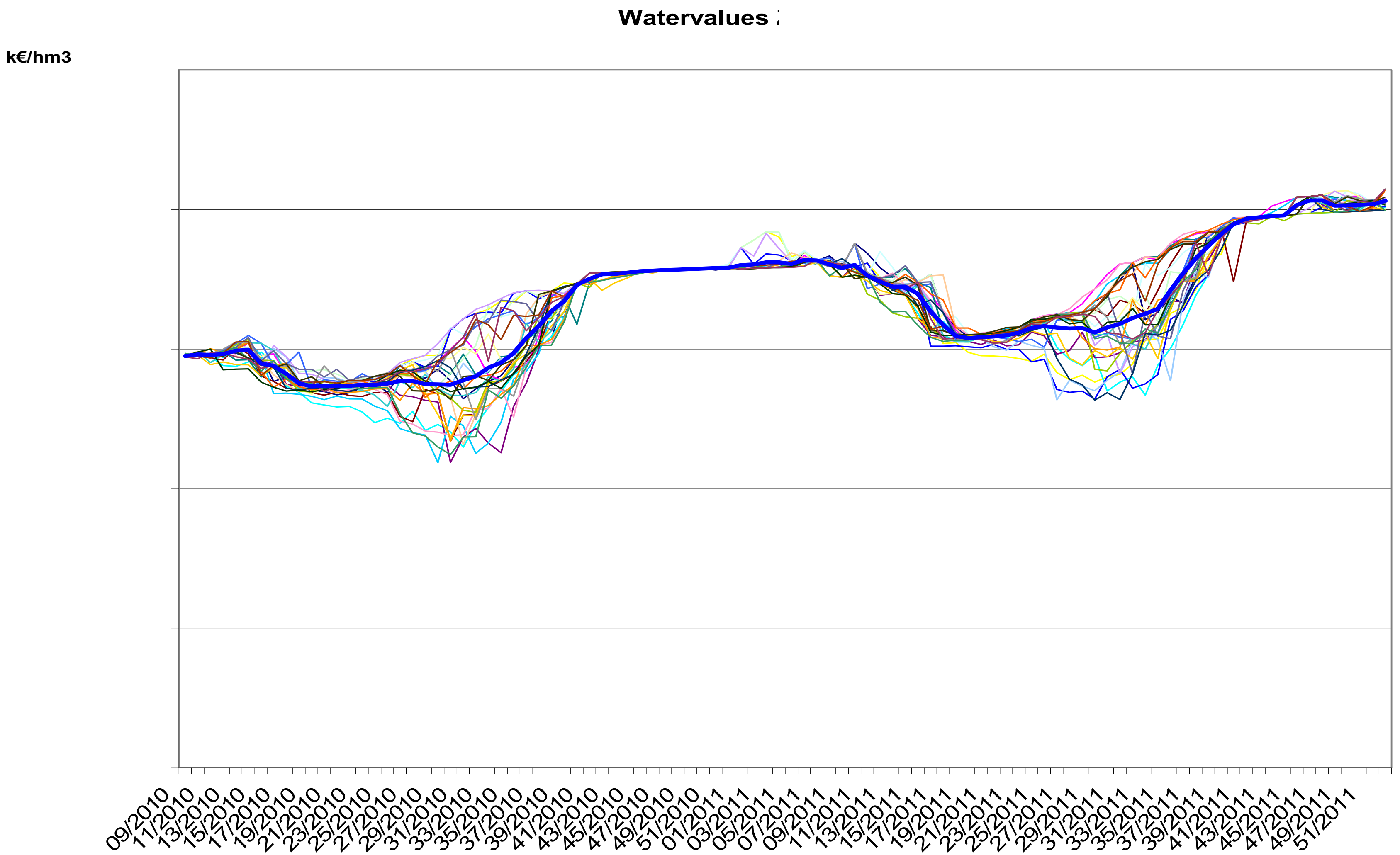
SDDP Software – Input



SDDP Software – Output content curve scenarios



SDDP Software – Output water values



SDDP approach

Pros:

- Excessive branching (good representation of uncertainty)

Limitation:

- No consideration of risk / hedging
- Only Markov price processes possible
- Approximation scheme
- SDDP / Maxrev software by *PSR Inc*.:
  - weekly time grid only
  - no integer variables

Conclusion

- **Multistage Stochastic Programming**
  - Most correct method to cope with uncertainty and optionality from a theoretical point of view
  - Risk management (Hedging) can be incorporated
  - Discretization of randomness necessary ("scenario trees")
  - „Curse of Dimensionality" in practice

  What can we expect?

- **For Hedging** power sales and generation and **management** of energy **storages**:
  - Mean-Risk models incorporating future trading

  vs.
  - Expectation-based stochastic optimization (e.g., SDDP) + ex-post calculation of hedge volumes

  vs.
  - Deterministic optimization + ex-post calculation of hedge volumes

  How much do we lose by using deterministic optimization + hedging?