1

Optimization of Dispersed Energy Supply - Stochastic Programming with Recombining Scenario Trees

Alexa Epe[1], Christian Küchler[2], Werner Römisch[2], Stefan Vigerske[2], Hermann-Josef Wagner[1], Christoph Weber[3], and Oliver Woll[3]

[1] Ruhr-Universität Bochum, Universitätsstraße 150, 44801 Bochum, Germany, {epe,lee}@lee.rub.de

[2] Humboldt–Universität zu Berlin, Unter den Linden 6, 10099 Berlin, Germany, {ckuechler,romisch,stefan}@math.hu-berlin.de

[3] Universität Duisburg-Essen, Universitätsstraße 2, 45141 Essen, Germany, {Christoph_Weber,Oliver.Woll}@uni-duisburg-essen.de

Summary. The steadily increasing share of wind energy within many power generating systems leads to strong and unpredictable fluctuations of the electricity supply and is thus a challenge with regard to power generation and transmission. We investigate the potential of energy storages to contribute to a cost optimal electricity supply by decoupling the supply and the demand. For this purpose we study a stochastic programming model of a regional power generating system consisting of thermal power units, wind energy, different energy storage systems, and the possibility for energy import. The identification of a cost optimal operation plan allows to evaluate the economical possibilities of the considered storage technologies.

On the one hand the optimization of energy storages requires the consideration of long-term planning horizons. On the other hand the highly fluctuating wind energy input requires a detailed temporal resolution. Consequently, the resulting optimization problem can, due to its dimension, not be tackled by standard solution approaches. We thus reduce the complexity by employing recombining scenario trees and apply a decomposition technique that exploits the special structure of those trees.

1.1 Introduction

Electric power, one of the most important fields within energy supply, has two main characteristics: on the one hand supply and demand have to be balanced at every time, on the other hand it is storable at only small rates. For these reasons, power plants have to regulate any imbalances between supply and demand, and, in particular, need to cope with unpredictable changes in the customer load. For that purpose regulating power plants are used, which mostly run in part load and with reduced efficiency. Alternatively fast power plants such as open cycle gas turbines may be used, which can start up within short time. Beyond the cover of the fluctuating load of the customer side, these power plants must also adjust to the increasing share of time-varying power production on the supply side, mostly caused from fluctuating renewables, notably wind.

Germany is the country with the highest installed wind power capacities worldwide. In the year 2006, there was approximately 20 GW installed (about 16.6% of the total installed power in Germany) and with the planned offshore development it could be up to 50 GW in 2030. Thereby, the sometimes strong and rapid fluctuations of the wind energy fed into the electrical network as well as the regional concentration in the north of the country increasingly pose problems to the network operators and power suppliers [7, 17]. Conventional fuel consumption may be saved by down-regulating conventional (back-up) power plants, but investments in the power plant park can hardly be saved.

In this context, electrical energy storages offer a possibility to decouple supply and demand and to achieve a better capacity utilization as well as a higher efficiency of existing power plants. The changing context has led to an increased interest in such possibilities over the last few years. Yet with the liberalization of the electricity markets, the economics of storages have to be valued against market prices as established at the energy exchanges. Also the operation of storages will mostly not follow local imbalances of demand and supply, but rather try to benefit from market price variations. In particular, the (partial) unpredictability of market prices as well as of wind energy supply have to be taken into account. Things are complicated further through daily, weekly, seasonal, and other cyclic patterns in demand, supply, and prices. This requires a valuation of storages (and other options) over periods as long as one year.

Cost optimal operation planning under uncertainty for such long time periods poses a huge challenge to conventional stochastic programming methods. In this paper we investigate a novel approach, reducing complexity by applying recombining scenario trees. The latter are used to analyze a regional energy system model that is described in Section 1.2. Section 1.3 presents the decomposition approach based on recombining trees, whereas Sections 1.4 and 1.5 are devoted to the results obtained so far.

1.2 Model Description

To study the economics of storages, a fundamental model is used. Combining technical and economical aspects, the model describes the energy supply of a large city, the available technologies for electricity generation, and the demand. An optimal load dispatch has to consider the marginal generation costs as well as the impact of other system restrictions such as start up costs, etc. Most important restriction of the model is the covering of the demand according to a given profile. For this purpose, energy can be produced by conventional

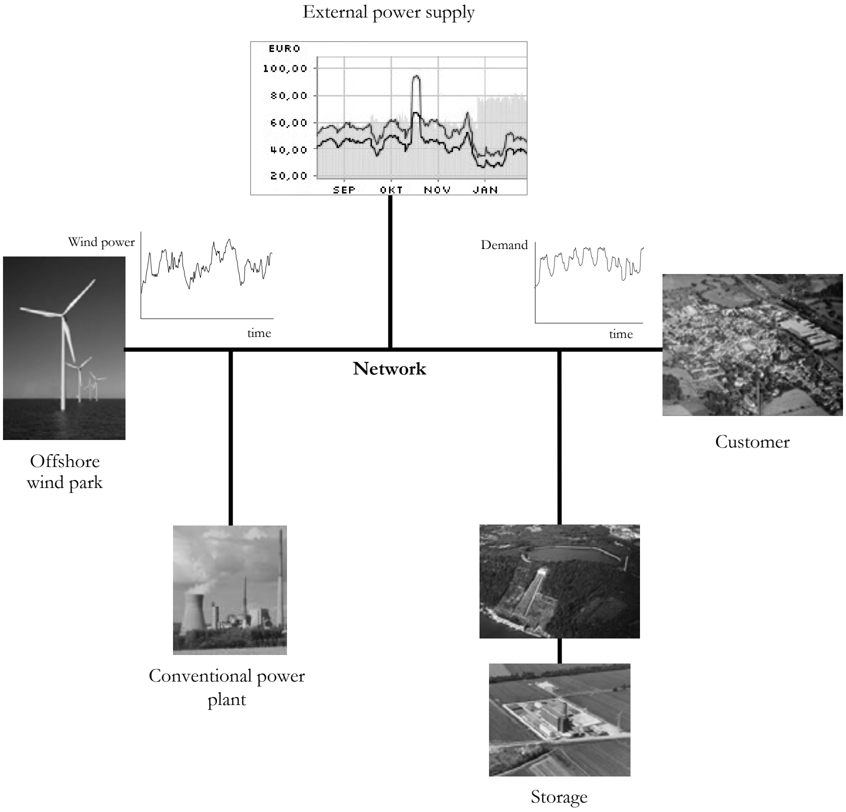

Fig. 1.1. Scheme of the fundamental model

Uncertainty in the amount of available wind energy and electricity prices is modelled by a multivariate stochastic process that can be represented by a recombining scenario tree. Thus, the proposed model combines many features of generation scheduling models (unit commitment and load dispatch) as found typically in energy system models [14, 15]. In the following, the model is discussed in detail. Table 1.1 gives an overview of the notation used.

Under the assumption of power markets with efficient information treatment and without market power, the market results correspond to the outcomes of an optimization carried out by a fully informed central planner. If electricity demand is assumed to be price inelastic, welfare maximization is equivalent to cost minimization within the considered power network. Thereby, the total costs TC are given as the sum of import costs IC_t, operating costs $OC_{t,i}$, and startup costs $SC_{t,i}$ over all time steps t and unit types i:

Variables			
Q	Production	IC	Import costs
H	Storage level	SC	Start-up costs
L	Capacity	OC	Operating costs
		TC	Total costs
Indices			
t	Time step	com	Compressing power
T	Final time	pum	Pumping power
i	Unit type	imp	Import power
stu	Start-up	wind	Wind power
Parameters			
D	Demand	c^{stu}	specific start-up costs
W	Wind power	c^{imp}	specific import costs
ℓ	Load factor	c^{oth}	other variable costs
η^0, η^m	Efficiency	c^{fuel}	fuel price

Table 1.1. Notation used by the model.

$$TC = \sum_{t=1}^{T} \left(IC_t + \sum_i OC_{t,i} + SC_{t,i} \right). \tag{1.1}$$

The costs for power import at time t are given by

$$IC_t = c_t^{\text{imp}} Q_t^{\text{imp}}. \tag{1.2}$$

For the operating costs $OC_{t,i}$, an affine function of the plant output $Q_{t,i}$ is assumed. An exact description of the plant operation costs requires a mixed-binary nonlinear formulation due to the dependency of the plant efficiency on the power output and the startup behaviour. This is hardly feasible due to the high level of time detail. An appopiate linearization can be done by defining an additional decision variable for each plant type, the *capacity currently online* $L_{t,i}^{onl}$ [18]. The capacity online forms an upper bound on the actual output. Multiplied with the minimum load factor, it is also a lower bound on the output for each power plant. Hence, operating costs can be decomposed in fuel costs for operation at minimum load, fuel costs for incremental output, and other variable costs:

$$OC_{t,i} = \frac{c_{i,t}^{\text{fuel}}}{\eta_i^0} \ell_i L_{t,i}^{\text{onl}} + \frac{c_{i,t}^{\text{fuel}}}{\eta_i^m} (Q_{t,i} - \ell_i L_{t,i}^{\text{onl}}) + c_i^{\text{oth}} Q_{t,i}. \tag{1.3}$$

Here, η_i^m denotes the marginal efficiency for an operating plant and η_i^0 the efficiency at the minimum load factor ℓ_i. With $\eta_i^m > \eta_i^0$, the operators have an incentive to reduce the capacity online (for details see [18]).

Besides operating costs, start-up costs may influence the power scheduling decisions considerably. The start-up costs of unit i at time t are given by

$$SC_{t,i} = c_i^{\text{stu}} L_{t,i}^{\text{stu}}, \tag{1.4}$$

where $L_{t,i}^{\mathrm{stu}}$ is the start-up capacity given by

$$L_{t,i}^{\mathrm{stu}} = \max(0, L_{t,i}^{\mathrm{onl}} - L_{t-1,i}^{\mathrm{onl}}). \tag{1.5}$$

Covering the demand at time step t is ensured by

$$\sum_i Q_{t,i} + Q_t^{\mathrm{wind}} + Q_t^{\mathrm{imp}} \geq D_t + \sum_i Q_{t,i}^{\mathrm{pum}} + \sum_i Q_{t,i}^{\mathrm{com}}, \tag{1.6}$$

i.e., the supply at time t is given by the sum of the power production $Q_{t,i}$, the imported energy Q_t^{imp}, and the wind energy supply Q_t^{wind}. The total demand equals the sum of the exogenously given domestic demand D_t and the pumping and compressing energies $Q_{t,i}^{pum}$ and $Q_{t,i}^{com}$ used to fill the pumped hydro storage and compressed-air storage, respectively.

The operation levels of the units, pumps, and air compressors are constrained by the available capacity,

$$Q_{t,i} \leq L_{t,i}, \qquad Q_{t,i}^{\mathrm{pum}} \leq L_{t,i}^{\mathrm{pum}}, \qquad Q_{t,i}^{\mathrm{com}} \leq L_{t,i}^{\mathrm{com}}, \tag{1.7}$$

whereas the wind energy supply is bounded by the available wind energy at time t,

$$Q_t^{\mathrm{wind}} \leq W_t. \tag{1.8}$$

For the storage plants, storage constraints need to be considered and the filling and discharging has to be described. This leads to the following storage level equation, linking the storage level $H_{t,i}$ at time t with the level $H_{t-1,i}$ at time $t-1$, both expressed in energy units. For the pumped hydro units, this reads as

$$\begin{aligned} H_{t,i} = H_{t-1,i} &- \frac{1}{\eta_i^m} Q_{t,i} - \left(\frac{1}{\eta_i^0} - \frac{1}{\eta_i^m} \right) \ell_i L_{t,i}^{\mathrm{onl}} \\ &+ \eta_i^{m,\mathrm{pum}} Q_{t,i}^{\mathrm{pum}} + (\eta_i^{0,\mathrm{pum}} - \eta_i^{m,\mathrm{pum}}) \ell_i L_{t,i}^{\mathrm{onl,pum}} \end{aligned} \tag{1.9}$$

for $t = 1, \ldots, T$, where $H_{0,i}$ denotes the initial fill level. Additionally, as an adequate terminal condition we require the initial and terminal fill levels of the reservoirs to be fixed at the minimum fill level $H_i^{\min}$. Further, the storage level at time step t is also limited by the minimum and maximum storage levels,

$$H_i^{\min} \leq H_{t,i} \leq H_i^{\max}. \tag{1.10}$$

Similar capacity constraints are formulated for the compressed-air units. Additionally, all variables have to fulfill non-negativity conditions.

The objective of the optimization is to find a decision process satisfying the constraints (1.5)–(1.10), being nonanticipative with respect to the stochastic process $(W_t, c_t^{\mathrm{imp}})_t$, and minimizing the expected total costs $\mathbb{E}[TC]$.

1.3 Decomposition using Recombining Scenario Trees

In this section, we present the solution method based on recombining scenario trees that has been developed in [11], and sketch a method for generating recombining scenario trees.

1.3.1 Problem Formulation

The optimization problem presented in Section 1.2 can be written as a linear multistage stochastic program:

$$\min \mathbb{E}\left[\sum_{t=1}^{T} \langle b_t(\boldsymbol{\xi}_t), x_t \rangle\right] \tag{1.11}$$

$$\begin{aligned} \text{s.t. } & x_t \in X_t, \quad x_t \in \sigma(\boldsymbol{\xi}^t), && t = 1, \ldots, T, \\ & A_{t,0}x_t + A_{t,1}x_{t-1} = h_t(\boldsymbol{\xi}_t), && t = 2, \ldots, T. \end{aligned} \tag{1.12}$$

Thereby, the vector x_t contains all decision variables at time stage t. The sets X_t are closed and polyhedral and model deterministic, static linear constraints at time t, i.e., the conditions (1.6), (1.7), and (1.10). The identities (1.12) describe the random and time-coupling constraints (1.5), (1.8), and (1.9). The uncertainty concerning the future wind energy input and spot prices is modeled by the bivariate discrete time stochastic process $\boldsymbol{\xi} = (\boldsymbol{\xi}_t)_{t=1,\ldots,T}$, that enters into the optimization model through the costs $b_t(\cdot)$ and the right-hand sides $h_t(\cdot)$, which are assumed to depend affinely linear on $\boldsymbol{\xi}_t$ for $t = 1, \ldots, T$. Furthermore, $\boldsymbol{\xi}$ defines the nonanticipativity constraints, i.e., a decision x_t at time t must depend exclusively on observations made until t. This is formalized by the condition $x_t \in \sigma(\boldsymbol{\xi}^t)$, where $\boldsymbol{\xi}^t$ denotes the vector $(\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_t)$.

To render possible a numerical solution of (1.11), every $\boldsymbol{\xi}^t$ is assumed to take values in a finite set $\Xi^t = \{\xi^t_{(1)}, \ldots, \xi^t_{(n_t)}\}$. Consequently, the process $\boldsymbol{\xi}$ can be represented by a scenario tree, cf., e.g., [2]. Then, (1.11) can be formulated as a (large scale) deterministic linear optimization problem that can be solved, in principle, by means of available solvers. However, with growing time horizon T, problem (1.11) becomes too large to be solved as a whole and one has to resort to decomposition techniques, e.g., temporal decomposition. To this end, one considers certain time stages $0 = R_0 < R_1 < \ldots < R_n < R_{n+1} = T$, and defines the cost-to-go function at time R_j and state $(x_{R_j}, \xi^{R_j}_{(i)}) \in X_{R_j} \times \Xi^{R_j}$ recursively by $\mathcal{Q}_{R_{n+1}}(\cdot, \cdot) := 0$ and the Bellman Equation

$$\begin{aligned}
&\mathcal{Q}_{R_j}(x_{R_j}, \xi^{R_j}_{(i)}) := && (\mathcal{Q}_{R_j})\\
&\qquad \min \mathbb{E}\left[\sum_{t=R_j+1}^{R_{j+1}} \langle b_t(\boldsymbol{\xi}_t), x_t\rangle + \mathcal{Q}_{R_{j+1}}(x_{R_{j+1}}, \boldsymbol{\xi}^{R_{j+1}}) \,\middle|\, \boldsymbol{\xi}^{R_j} = \xi^{R_j}_{(i)}\right]\\
&\qquad \text{s.t. } x_t \in X_t, \quad x_t \in \sigma(\boldsymbol{\xi}^t), \qquad t = R_j+1, \ldots, R_{j+1},\\
&\qquad\qquad A_{t,0}x_t + A_{t,1}x_{t-1} = h_t(\boldsymbol{\xi}_t), \qquad t = R_j+1, \ldots, R_{j+1},
\end{aligned}$$

for $j = 1, \ldots, n$. Using this notation, problem (1.11) can be reformulated in terms of Dynamic Programming:

$$\begin{aligned}
\min \quad & \mathbb{E}\left[\sum_{t=1}^{R_1} \langle b_t(\boldsymbol{\xi}_t), x_t\rangle + \mathcal{Q}_{R_1}(x_{R_1}, \boldsymbol{\xi}^{R_1})\right] && (\mathcal{Q}_0)\\
\text{s.t.} \quad & x_t \in X_t, \quad x_t \in \sigma(\boldsymbol{\xi}^t), && t = 1, \ldots, R_1,\\
& A_{t,0}x_t + A_{t,1}x_{t-1} = h_t(\boldsymbol{\xi}_t), && t = 2, \ldots, R_1,
\end{aligned}$$

and solved by, e.g., the *Nested Benders Decomposition method* [1, 12, 16]. Furthermore, a modification of this algorithm as proposed in [11] allows to exploit the structure of *recombining scenario trees* for simultaneous cutting plane approximations. This approach indeed enables to solve problem $(\mathcal{Q}_0)$ for longer time horizons T and large number of scenarios.

1.3.2 Recombining Scenario Trees

At time t, the scenario tree representing $\boldsymbol{\xi}$ has $n_t = |\varXi^t|$ nodes, that are denoted by $u = 1, \ldots, n_t$. The node u corresponds to the event $\{\boldsymbol{\xi}^t = \xi^t_{(u)}\}$. A special situation is given whenever the subtrees associated at some nodes u and k at time R_j coincide, i.e., the corresponding conditional distributions of $(\boldsymbol{\xi}_t)_{t=R_j+1,\ldots,T}$ are equal:

$$\mathbb{P}\left[(\boldsymbol{\xi}_t)_{t=R_j+1,\ldots,T} \in \cdot \;|\boldsymbol{\xi}^{R_j} = \xi^{R_j}_{(u)}\right] = \mathbb{P}\left[(\boldsymbol{\xi}_t)_{t=R_j+1,\ldots,T} \in \cdot \;|\boldsymbol{\xi}^{R_j} = \xi^{R_j}_{(k)}\right]. \tag{1.13}$$

As far as it concerns the tree representation of the process $\boldsymbol{\xi}$, property (1.13) would allow to recombine the nodes u and k, and recombining at several time stages R_j may prevent the node number to grow exponentially with the number of time stages. Unfortunately, recombining is not allowed under time coupling constraints (1.12) since the scenario-dependent control $x_{R_j}(\boldsymbol{\xi}^{R_j})$ will not be equal on $\{\boldsymbol{\xi}^{R_j} = \xi^{R_j}_{(u)}\}$ and $\{\boldsymbol{\xi}^{R_j} = \xi^{R_j}_{(k)}\}$, in general. However, (1.13) can be useful since it entails equality of the cost-to-go functions $\mathcal{Q}_{R_j}(\cdot, \xi^{R_j}_{(u)})$ and $\mathcal{Q}_{R_j}(\cdot, \xi^{R_j}_{(k)})$. This is exploited by the solution algorithm presented in Section 1.3.3.

In the remaining part of this section we sketch a method for generating scenario trees with property (1.13) for some nodes and several time stages

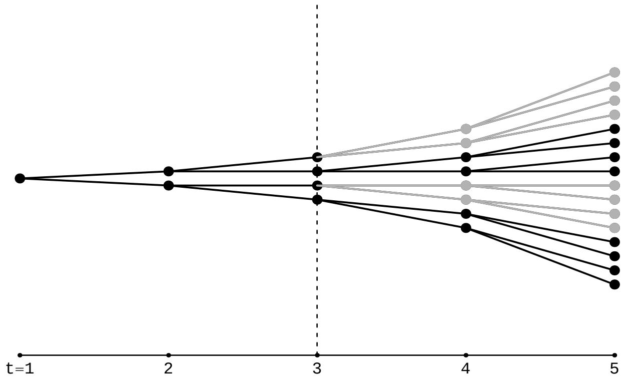

Fig. 1.2. Scenario tree with property (1.13), $R_1 = 3$, and $m_{R_1} = 2$, i.e., two different subtrees are associated at time stage 3. (The black and the gray subtrees coincide, respectively.)

R_j, $j = 1, \dots, n$. It is a modification of the *forward tree construction* [9], also based on successive stagewise clustering of a set of sampled trajectories $\zeta^i = (\zeta_1^i, \dots, \zeta_T^i), i = 1, \dots, N$, that coincide in $t = 1$. Basically, it consists of constructing non-recombining subtrees for every time period $[R_j + 1, R_{j+1}]$ (Step 2), and assigning to several nodes at time R_{j+1} the same subtree for the subsequent time period (Step 1). Thereby, two nodes at time R_{j+1} obtain the same subtree h, whenever the values of $\boldsymbol{\xi}$ in these nodes are close for some time $\underline{t}$ before R_{j+1}. Table 1.2 explains the notation used. The values of $m_{R_j}, n_{R_j}(h)$, and $s_{t+1}(u)$, determining the structure of the scenario tree, may be predefined or, as proposed in [9], determined within the algorithm to not exceed certain local error levels. Whenever the sampled trajectories come from a time series model, the parameter $\underline{t}$ may be chosen according to the latter.

Algorithm 1 (Generation of a recombining scenario tree).
Initialization: Set $C_1^{(1,1)} := \underline{C}_{R_0}^{(1)} := \{1, \dots, N\}, \xi_1^{(1)} := \zeta_1^1$ and $\underline{C}_{R_j}^{(h)} := \varnothing$ for all h and $j \geq 1$.
For $j = 0, \dots, n$ (*j-th recombination time stage*):

1. If $j > 0$: *Short-term history clustering for subtree assignment.*
   Find an index set $A = \{a_1, \dots, a_{m_{R_j}}\} \subset \{1, \dots, N\}$ with minimal

$$\sum_{h=1,\dots,m_{R_{j-1}}} \sum_{u=1,\dots,n_{R_j}(h)} \min_{a_l \in A} \sum_{i \in C_{R_j}^{(h,u)}} \|(\zeta^i_{R_j-\underline{t}}, \dots, \zeta^i_{R_j}) - (\zeta^{a_l}_{R_j-\underline{t}}, \dots, \zeta^{a_l}_{R_j})\|.$$

   Pass through all nodes (h, u) at time R_j:
   a) Consider an $a_{h'} \in A$ that is close to node (h, u) in the sense that

ζ_t^i	value of trajectory i at time t
$\xi_{t,(u)}$	value of the random variable $\boldsymbol{\xi}_t$ in node u
m_{R_j}	number of subtrees with root node at time R_j ($m_{R_0} := 1$)
(h,u)	node u of some subtree h
$n_{R_j}(h)$	number of nodes at time R_j of some subtree h
$s_{t+1}(u)$	number of nodes at time $t+1$ descending from some node u at time t
$\underline{t}$	time parameter for short-term history clustering
$C_t^{(h,u)}$	subset of $\{1,\ldots,N\}$, indicating trajectories ζ^i going at time t through node (h,u)
$\underline{C}_{R_j}^{(h)}$	subset of $\{1,\ldots,N\}$, indicating trajectories ζ^i lying in subtree h with root node at time R_j

Table 1.2. Notation used by Algorithm 1.

$$a_{h'} \in \operatorname*{arg\,min}_{a_l \in A} \sum_{i \in C_{R_j}^{(h,u)}} \|(\zeta_{R_j-\underline{t}}^i, \ldots, \zeta_{R_j}^i) - (\zeta_{R_j-\underline{t}}^{a_l}, \ldots, \zeta_{R_j}^{a_l})\|.$$

b) Node (h,u) obtains subtree h', i.e., one has to update

$$\underline{C}_{R_j}^{(h')} := \underline{C}_{R_j}^{(h')} \cup C_{R_j}^{(h,u)}.$$

In particular, all trajectories ζ^i belonging to node (h,u) will be used for the construction of subtree h' of the subsequent timeperiod.

2. *Subtree generation.*
For every subtree $h = 1, \ldots, m_{R_j}$ of period $[R_j+1, R_{j+1}]$:
a) Find an index set $A = \{a_1, \ldots, a_{n_{R_j}(h)}\} \subset \underline{C}_{R_j}^{(h)}$ with minimal

$$\sum_{i \in \underline{C}_{R_j}^{(h)}} \min_{a_l \in A} \|\zeta_{R_j+1}^i - \zeta_{R_j+1}^{a_l}\|.$$

Find a partition $C_{R_j+1}^{(h,u)}$, $u = 1, \ldots, n_{R_j+1}(h)$, of $\underline{C}_{R_j}^{(h)}$ with

$$C_{R_j+1}^{(h,u)} \subset \{i \in \underline{C}_{R_j}^{(h)} : a_u \in \operatorname*{arg\,min}_{a_l \in A} \|\zeta_{R_j+1}^i - \zeta_{R_j+1}^{a_l}\|\}.$$

Define the value of $\boldsymbol{\xi}_{R_j+1}$ on node (h,u) by $\xi_{R_j+1,(h,u)} := \zeta_{t+1}^{a_u}$.
For every subtree $\tilde{h}$ of the preceding period $[R_{j-1}+1, R_j]$ and every node $(\tilde{h}, \tilde{u})$ at time R_j: Define the *transition probability* from node $(\tilde{h}, \tilde{u})$ to node (h,u) by

$$\mathbb{P}_{R_j+1|R_j}[(h,u)|(\tilde{h},\tilde{u})] := \begin{cases} \frac{|C_{R_j+1}^{(h,u)}|}{|C_{R_j}^{(h)}|} & \text{if } C_{R_j}^{(\tilde{h},u)} \subset \underline{C}_{R_j}^{(h)}, \\ 0 & \text{else.} \end{cases}$$

b) For $t = R_j + 1, \dots, R_j - 1$:
For every node $\tilde{u}$ of subtree h at time t:
Find an index set $A = \{a_1, \dots, a_{s_{t+1}(\tilde{u})}\} \subset C_t^{(h,\tilde{u})}$ with minimal

$$\sum_{i \in C_t^{(h,\tilde{u})}} \min_{a_l \in A} \|\zeta_{t+1}^i - \zeta_{t+1}^{a_l}\|.$$

Find a partition $C_{t+1}^{(h,u)}, u = 1, \dots, s_{t+1}(\tilde{u})$, of $C_t^{(h,\tilde{u})}$ with

$$C_{t+1}^{(h,u)} \subset \{i \in C_t^{(h,\tilde{u})} : a_u \in \arg\min_{a_l \in A} \|\zeta_{t+1}^i - \zeta_{t+1}^{a_l}\|\}.$$

Set $\xi_{t+1,(h,u)} := \zeta_{t+1}^{a_u}$ and $\mathbb{P}_{t+1|t}[(h,u)|(h,\tilde{u})] := \frac{|C_{t+1}^{(h,u)}|}{|C_t^{(h,\tilde{u})}|}$.

The determination of the index sets A is a $k-$mean problem and, thus, an NP-hard combinatorial optimization problem [5]. While it is possible for small values $m_{R_j}, n_{R_j}(h)$, and $s_{t+1}(u)$ to find optimal sets A by enumeration, larger values demand for heuristics, e.g., the *forward selection* proposed in [3, 8].

In Step 1 of Algorithm 1, several nodes $(\tilde{h}, \tilde{u})$ at time R_j obtain the same subtree h. For notational convenience, we pick out a representative amongst them and denote the associated value of $\boldsymbol{\xi}^{R_j}$ by $\lambda_h^{R_j}$. The following function will be used in Section 1.3.3 and maps a node $(\tilde{h}, \tilde{u})$ with subtree h to the corresponding representative node:

$$\begin{aligned} \boldsymbol{\lambda}^{R_j} &: \Xi^{R_j} \to \{\lambda_1^{R_j}, \dots, \lambda_{m_{R_j}}^{R_j}\} =: \Lambda^{R_j}, \\ \boldsymbol{\lambda}^{R_j}(\xi_{(\tilde{h},\tilde{u})}^{R_j}) &:= \lambda_h^{R_j} \text{ whenever } C_{R_j}^{(\tilde{h},\tilde{u})} \subset \underline{C}_{R_j}^{(h)}. \end{aligned}$$

1.3.3 Solution Algorithm

In [11], it was shown how to modify a Nested Benders Decomposition [1, 12, 16] of problem (1.11) to exploit the recombining property (1.13) of the process $\boldsymbol{\xi}$. In the following, we sketch this modified algorithm.

Let us consider the formulation $(\mathcal{Q}_0)$ of problem (1.11). A Nested Benders Decomposition successively approximates the piecewise-linear convex functions $x_{R_j} \mapsto \mathcal{Q}_{R_j}(x_{R_j}, \xi_{(u)}^{R_j})$ by a set of supporting hyperplanes and evaluates them in an adaptively chosen sequence of points x_{R_j}. Whenever two nodes u and k at time R_j fulfill (1.13), the functions $\mathcal{Q}_{R_j}(\cdot, \xi_{(u)}^{R_j})$ and $\mathcal{Q}_{R_j}(\cdot, \xi_{(k)}^{R_j})$ coincide, and, thus, they may be approximated simultaneously.

To this end, we define the following underestimating functions: We set $\mathcal{Q}_{R_{n+1}}^{LC}(\cdot, \cdot) := 0$ and for $j = n, \dots, 0$, $\bar{x}_{R_j} \in X_{R_j}$, and $\lambda_i^{R_j} \in \Lambda^{R_j}$ let

$$
\begin{aligned}
&\mathcal{Q}^L_{R_j}(\bar{x}_{R_j}, \lambda_i^{R_j}) := && (\mathcal{Q}^L_{R_j})\\
&\min \mathbb{E}\left[\sum_{t=R_j+1}^{R_{j+1}} \langle b_t(\boldsymbol{\xi_t}), x_t\rangle + \mathcal{Q}^{LC}_{R_{j+1}}(x_{R_{j+1}}, \boldsymbol{\lambda}^{R_{j+1}}(\boldsymbol{\xi}^{R_{j+1}})) \middle| \boldsymbol{\xi}^{R_j} = \lambda_i^{R_j} \right] &&\\
&\text{s.t.}\, x_t \in X_t, \quad x_t \in \sigma(\boldsymbol{\xi}^t), \qquad t = R_j+1, \ldots, R_{j+1}, &&\\
&\quad A_{t,0}x_t + A_{t,1}x_{t-1} = h_t(\boldsymbol{\xi_t}), \quad t = R_j+1, \ldots, R_{j+1}, &&\\
&\quad x_{R_j} = \bar{x}_{R_j}. && (1.14)
\end{aligned}
$$

Thereby, $\mathcal{Q}^{LC}_{R_{j+1}}(\cdot, \lambda_i^{R_{j+1}})$ is an approximation of $\mathcal{Q}^{L}_{R_{j+1}}(\cdot, \lambda_i^{R_{j+1}})$ by supporting hyperplanes that is easy to evaluate and that will be properly defined in equation (1.15) below. Problem $(\mathcal{Q}^L_{R_j})$ is often referred to as the *master problem.* Note, that in contrast to the classical Nested Benders Decomposition, the *same* approximation $\mathcal{Q}^{LC}_{R_{j+1}}(\cdot, \lambda_i^{R_{j+1}})$ can be used in the objective function of $(\mathcal{Q}^L_{R_j})$ for all nodes with the same subtree i, i.e., whenever $\boldsymbol{\lambda}^{R_{j+1}}(\boldsymbol{\xi}^{R_{j+1}}) = \lambda_i^{R_{j+1}}$. Thus, this decomposition into subproblems for each timeperiod allows one to exploit the recombining nature of the process ξ. This had not been possible with other decomposition algorithms like scenario decomposition [12].

The function $\mathcal{Q}^{LC}_{R_j}(\cdot, \lambda_i^{R_j})$ is used to induce a feasible solution at stage R_j and to approximate the value of $\mathcal{Q}^{L}_{R_j}(\cdot, \lambda_i^{R_j})$ on its domain. For the latter purpose, given a point $\bar{x} \in X_{R_j}$ with $\mathcal{Q}^{L}_{R_j}(\bar{x}, \lambda_i^{R_j}) < \infty$, an *optimality cut* supporting $\mathcal{Q}^{L}_{R_j}(\cdot, \lambda_i^{R_j})$ is given by $\mathcal{Q}^{L}_{R_j}(\bar{x}, \lambda_i^{R_j}) + \langle \pi, x_{R_j} - \bar{x}\rangle \leq 0$, where π denotes the dual variables corresponding to the constraint (1.14) in an optimal solution of problem $(\mathcal{Q}^L_{R_j})$. To induce feasibility at time stage R_j, a point $\bar{x} \in X_{R_j}$ that is infeasible for $(\mathcal{Q}^L_{R_j})$ is cut off using a *feasibility cut* $\langle d, x\rangle + e \leq 0$. This cut is computed by solving an auxiliary problem, cf. [11], and has the property $\langle d, \bar{x}\rangle + e > 0$ and $\langle d, x\rangle + e \leq 0$ for all $x \in X_{R_j}$ with $\mathcal{Q}^{L}_{R_j}(x, \lambda_i^{R_j}) < \infty$.

Hence, an approximation of $\mathcal{Q}^{L}_{R_j}(\cdot, \lambda_i^{R_j})$ by means of optimality cuts $C_{\text{opt}}(\lambda_i^{R_j})$ and feasibility cuts $C_{\text{feas}}(\lambda_i^{R_j})$ is given by

$$
\begin{aligned}
\mathcal{Q}^{LC}_{R_j}(x_{R_j}, \lambda_i^{R_j}) := \max_{(\bar{x},\bar{\pi}) \in C_{\text{opt}}(\lambda_i^{R_j})} \quad & \mathcal{Q}^{L}_{R_j}(\bar{x}, \lambda_i^{R_j}) + \left\langle \bar{\pi}, x_{R_j} - \bar{x}\right\rangle \qquad (1.15)\\
\text{s.t.} \quad & \left\langle d, x_{R_j}\right\rangle + e \leq 0, \quad (d, e) \in C_{\text{feas}}(\lambda_i^{R_j}).
\end{aligned}
$$

The solution algorithm processes the master problems $(\mathcal{Q}^L_{R_j})$, $j = 0, \ldots, n$, of the decomposed scenario tree in a forward or backward manner. At each time stage R_j, each master problem $\mathcal{Q}^{L}_{R_j}(\cdot, \lambda_i^{R_j})$, $\lambda_i^{R_j} \in \Lambda^{R_j}$, is evaluated for a set $Z_j(\lambda_i^{R_j})$ of controls x_{R_j}. If $\mathcal{Q}^{LC}_{R_j}(x_{R_j}, \lambda_i^{R_j}) < \mathcal{Q}^{L}_{R_j}(x_{R_j}, \lambda_i^{R_j})$, the approximation $\mathcal{Q}^{LC}_{R_j}(\cdot, \lambda_i^{R_j})$ (and all master problems that use $\mathcal{Q}^{LC}_{R_j}(\cdot, \lambda_i^{R_j})$) is updated by generating new optimality or feasibility cuts. Further, in the forward mode, new control points $x_{R_{j+1}}$ are generated from the solution of the master problem $(\mathcal{Q}^L_{R_j})$ to form the sets $Z_{j+1}(\lambda_i^{R_{j+1}})$ for $\lambda_i^{R_{j+1}} \in \Lambda^{R_{j+1}}$. Since each such evaluation contributes several new controls $x_{R_{j+1}}$ to the sets $Z_{j+1}(\lambda_i^{R_{j+1}})$, the latter can grow exponentially with increasing j. It was shown in [11] how the problem structure allows to deal with this difficulty.

The algorithm stops when either the first timeperiod master problem $(\mathcal{Q}^L_0)$ is infeasible, or all master problems could be solved to optimality and the generation of cuts has stopped. In the former case, also problem $(\mathcal{Q}_0)$ is infeasible, in the latter, the problem has been solved to optimality. Another stopping criteria which allows to stop the algorithm when the error falls below a given tolerance, is also discussed in [11]. A more detailed description of the Nested Benders Decomposition Algorithm can be found in [1, 6, 12].

1.4 Case study

We study a power generating system, consisting of a hard coal power plant to cover the minimum and medium load, and two fast gas turbines on different power levels to cover the peaks. The operating parameters of these units rely on real data. Furthermore, the model contains an offshore wind park, a pump-storage power plant (PSW) with the basic data of the PSW Geesthacht, Germany, and a compressed-air energy storage (CAES) with the operating parameters of the CAES Huntorf, Germany. Further source of power supply is the EEX spot market. The time horizon considered for the optimization is one year and a hourly discretization is used, i.e., the model contains $T = 8760$ time stages.

The stochastic wind power process is represented by a time series model fitted to historical data and scaled to the size of the offshore wind park regarded. To take into account the interdependency between wind power and spot price behaviour, the *expected* spot market prices are calculated from a fundamental model that is based on the existing power plants in Germany and their reliability, prices for fuels and CO_2, the German load, and the wind power process above. Fluctuation of the spot prices around their expected value are modeled by a further time series model. This hybrid approach was used to generate 1 000 trajectories, containing hourly values of wind power and spot prices in the course of one year. These trajectories were used to generate a recombining tree by Algorithm 1 of Section 1.3.2. The resulting scenario tree branches three times per day in a binary way. Recombination into $m_{R_j} = 3$ different subtrees took place once a day, i.e., $R_j = j \cdot 24$, $j = 1, \ldots, 364$.

1.5 Numerical Results

The optimization problem was solved with varying model parameters. To this end, a *base setting* was defined, with wind power of approximately 50% of the

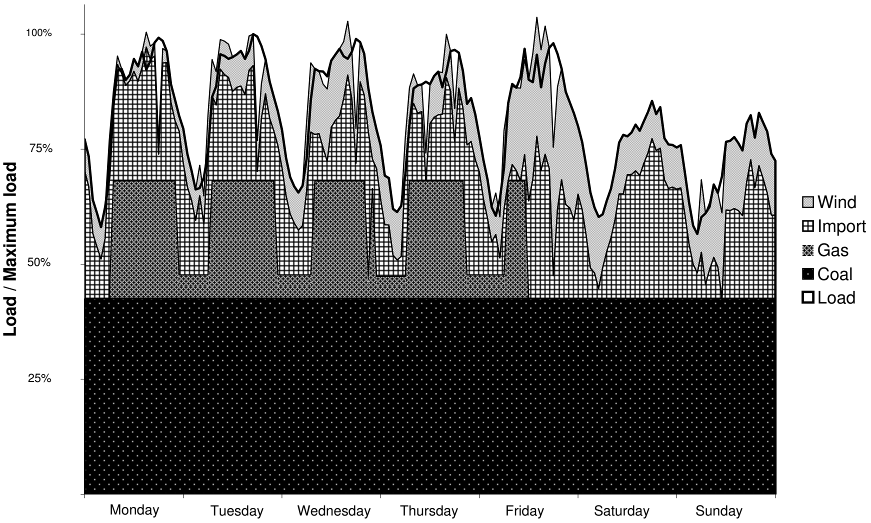

Fig. 1.3. Optimal power scheduling in a winter week.

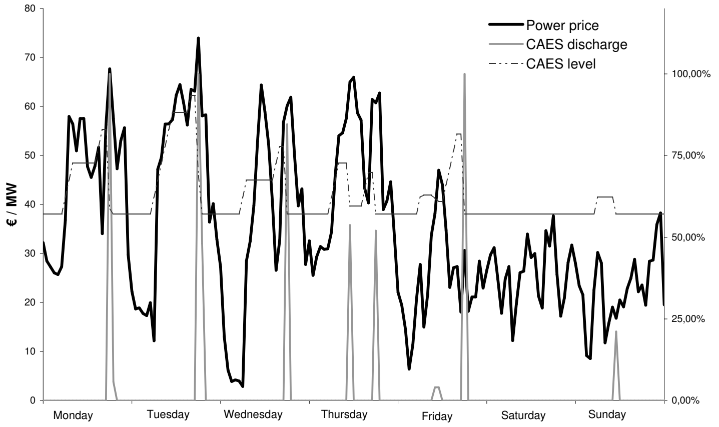

Fig. 1.4. Spot market price and CAES output in a winter week.

totally installed plant power and storage sizes corresponding to the aforementioned CAES and PSW units. Coming from this setting, variations with higher and lower levels of installed wind power and different storage dimensions were calculated. In the following some results are presented.

The optimal operation levels along a randomly chosen scenario from the base setting during a winter week are depicted in Figure 1.3. Whenever the power production exceeds the demand curve, energy is put into the storages, whereas the white spaces under the demand curve represent the output of the storage plants. The operation levels of the thermal units show the usual characteristics and availability of wind power obviously reduces imports from the spot market. The storage units are mainly used to cover the peaks and are only marginally used during the weekend. In this model, the contribution of the operating costs to the power supply costs amount to 2.08 Eurocents/kWh with using storage plants and 2.10 Eurocents/kWh without using storage plants. Figure 1.4 shows the optimal output and fill level of the CAES (as a fraction of maximum discharge power and maximum fill level, respectively) in comparison to the actual power price. The minimum fill level of the CAES is 60%. Obviously, the storage plant discharges in times of high spot prices on weekdays. The aforementioned marginal usage of storage plants during the weekend coincides with lower power prices over this period.

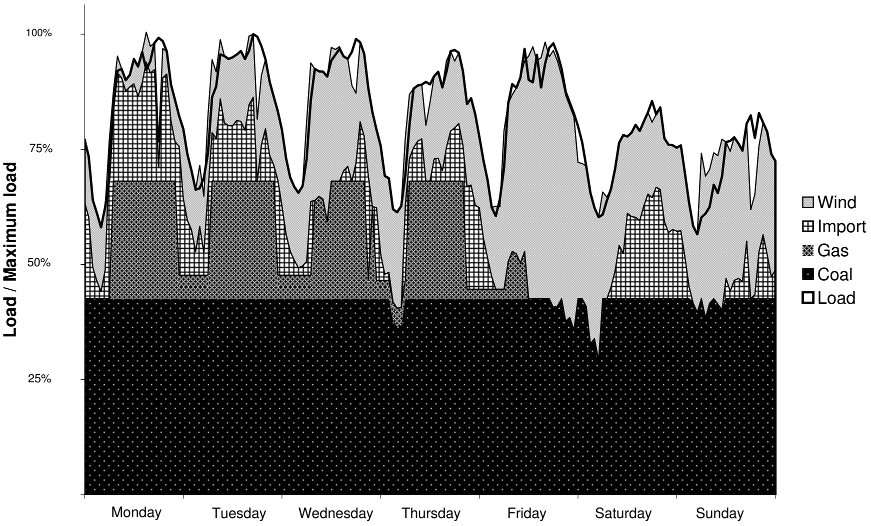

Fig. 1.5. Optimal power scheduling in a winter week with doubled wind power capacity.

To study the impact of the share of wind power on the system, the optimization problem was solved again with doubled wind power capacity. The results along the same scenario and for the same winter week are depicted in Figure 1.5. While this extension does not lead to significant changes of the thermal units, it enables to largely reduce the amount of energy bought at the spot market. Figure 1.6 shows the operation of the CAES in the course of the week. Again, the CAES is mainly used at peak times to avoid expensive

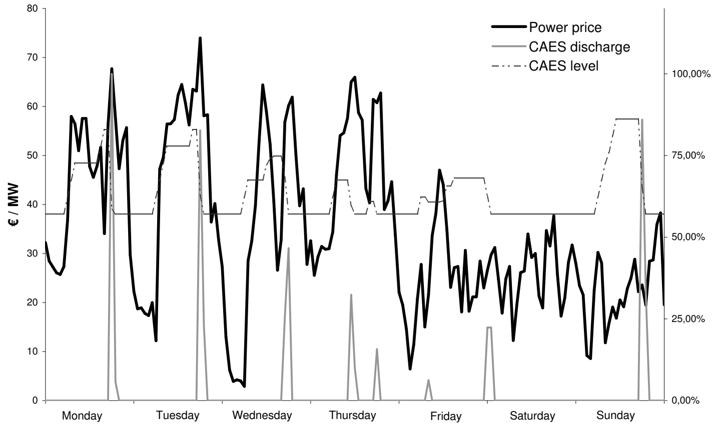

Fig. 1.6. Spot market price and CAES output in a winter week with doubled wind power capacity.

imports from the spot market. It can be seen that the availability of more wind power in the system can lead both to more and to less extraction of stored energy. This is due to the fact that, on the one hand, with more wind power more energy may be stored and therefore extracted (Sunday). On the other hand, less power has to be generated in times with high wind power (from Wednesday to Friday).

The optimization problem was solved further times with varying quantities of installed wind power and storage capacities. Figure 1.7 shows the minimal expected costs depending on the wind power capacity for different storage capacities. Thereby, a wind factor of y stands for an amount of wind power being y times the wind power of the *base setting*. In relation to the wind power capacity, the impact of an extension of the storage system on the costs appears to be rather marginal and the individual curves are almost superposed. Thus, to analyze the latter, Figure 1.8 shows the relative reduction of costs that can be achieved by the use of storage systems of different dimensions, where a model without storages generates operating costs of 100%. Again, a storage system dimension of y corresponds to y times the dimension of the base setting. The results clearly show, that the relative cost reduction due to storage use is the highest in the twice-wind-setting, and in all settings the most prevailing gradient is between no use and the use of the half dimension of storage sizes. Hence, it seems promising to study expansion models for cost-optimal storage sizes, taking into account operational as well as investment costs.

The optimization algorithm was implemented in C++ and the master problems were solved with CPLEX 10.0 [10]. Running time on a PC with 2.4 GHz CPU and 2 GB RAM was 10 minutes, approximatively.

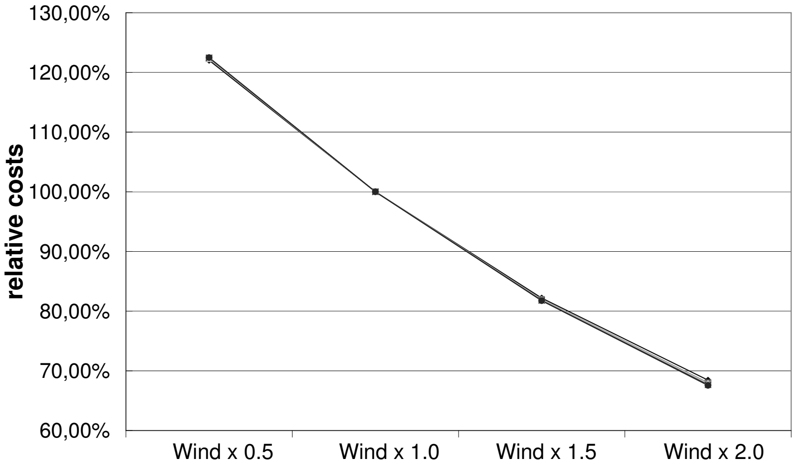

Fig. 1.7. Minimal expected costs depending on the wind power capacity installed for different storage capacities installed.

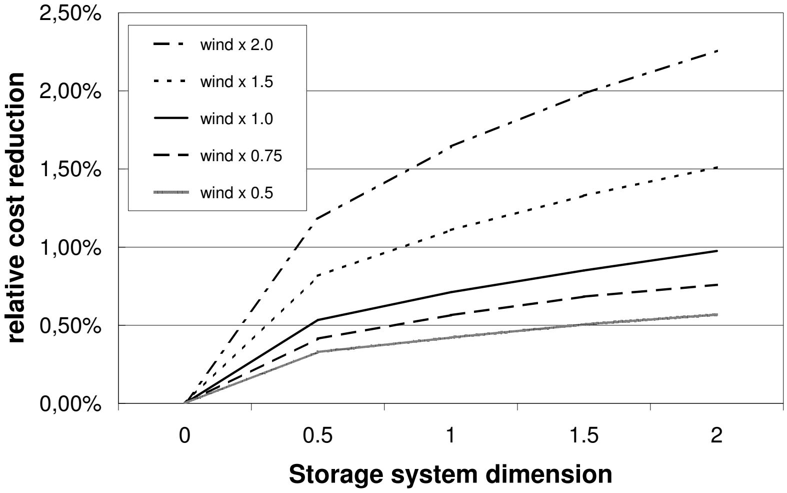

Fig. 1.8. Reduction of minimal expected costs depending on the storage capacity installed for different wind power capacities installed.

1.6 Conclusions and Outlook

We applied a decomposition method for linear multistage stochastic optimization problems proposed by [11] to optimal scheduling within a regional energy system including wind power and energy storages. It has been shown that this approach relying on recombining scenario trees allows to handle multistage problems with large numbers of scenarios and including time-coupling constraints, and, therefore, it is suitable for optimizing and analyzing energy systems.

In principle, the recombining tree decomposition approach allows for discrete decision variables in the first time stage and, hence, this method seems to be also appropriate to find optimal first-stage investment decisions within expansion models. This could be one aspect of future studies. However, further research is needed to extend the decomposition approach to more general optimization models, in particular those including discrete variables in later time stages. The latter would allow to adapt numerous aspects of the energy system model to achieve a more detailed picture of the system and its constraints.

Another aspect of future research could be the extension of the decomposition approach to optimization problems including multiperiod risk measures. For this purpose, especially the class of polyhedral risk measures [4] seems to be suitable, since the linear structure of the optimization problem is maintained.

Acknowledgement

This work was supported by the German Ministry of Education and Research (BMBF) within the topic “Dezentrale regenerative Energieversorgung: Innovative Modellierung und Optimierung” and the “Wiener Wissenschafts-, Forschungs- und Technologiefonds” in Vienna (`http://www.univie.ac.at/crm/simopt`).

References

1. J.R. Birge. Decomposition and partitioning methods for multistage stochastic programming. *Operations Research*, 33(5):989–1007, 1985.
2. J. Dupačová, G. Consigli, and S.W. Wallace. Scenarios for multistage stochastic programming. *Annals of Operations Research*, 100:25–53, 2000.
3. J. Dupačová, N. Gröwe-Kuska, and W. Römisch. Scenarios reduction in stochastic programming: An approach using probability metrics. *Mathematical Programming*, 95(A):493–511, 2003.
4. A. Eichhorn and W. Römisch. Polyhedral risk measures in stochastic programming. *SIAM Journal on Optimization*, 16:69–95, 2005.
5. M.R. Garey and D.S. Johnson. *Computers and Intractability - A Guide to the Theory of NP-Completeness.* W.H. Freeman, 1979.
6. H.I. Gassmann. MSLiP: a computer code for the multistage stochastic linear programming problem. *Mathematical Programming*, 47:407–423, 1990.
7. E. Handschin, F. Neise, H. Neumann, and R. Schultz. Optimal operation of dispersed generation under uncertainty using mathematical programming. *International Journal of Electrical Power & Energy Systems*, 28:618–626, 2006.

8. H. Heitsch and W. Römisch. Scenario reduction algorithms in stochastic programming. *Computational Optimization and Applications*, 24:187–206, 2003.
9. H. Heitsch and W. Römisch. Scenario tree modeling for multistage stochastic programs. *Mathematical Programming,* to appear, 2008.
10. ILOG, Inc. CPLEX 10.0. `http://www.ilog.com/products/cplex`.
11. C. Küchler and S. Vigerske. Decomposition of multistage stochastic programs with recombining scenario trees. *Stochastic Programming E-Print Series*, 9, 2007. `http://www.speps.org`.
12. A. Ruszczyński. *Decomposition Methods*, chapter 3, pages 141–221. In Ruszczyński and Shapiro [13], 2003.
13. A. Ruszczyński and A. Shapiro, editors. *Stochastic Programming.* Handbooks in Operations Research and Management Science. Elsevier, Amsterdam, 2003.
14. D. Swider, P. Vogel, and C. Weber. Stochastic model for the european electricity market and the integration costs for wind power. Technical report, GreenNet Report on WP 6, 2004.
15. D. Swider and C. Weber. The costs of wind's intermittency in Germany: Application of a stochastic electricity market model. *European Transactions on Electrical Power*, 17(2): 151–172.
16. R.M. Van Slyke and R. Wets. L-shaped linear programs with applications to optimal control and stochastic programming. *SIAM Journal of Applied Mathematics*, 17(4):638–663, 1969.
17. H.-J. Wagner. *Wind Energy and Present Status in Germany. In: N. Bansal and J. Mathur (Editors), Wind Energy Utilization.* Anamaya Publishers, New Delhi, 2002.
18. C. Weber. *Uncertainty in the Electric Power Industry: Methods and Models for Decision Support.* New York et al., 2005.