POWER MANAGEMENT IN A HYDRO-THERMAL SYSTEM UNDER UNCERTAINTY BY LAGRANGIAN RELAXATION

NICOLE GRÖWE-KUSKA[*], KRZYSZTOF C. KIWIEL[†],
MATTHIAS P. NOWAK[‡], WERNER RÖMISCH[§], AND ISABEL WEGNER[¶]

Abstract. We present a dynamic multistage stochastic programming model for the cost-optimal generation of electric power in a hydro-thermal system under uncertainty in load, inflow to reservoirs and prices for fuel and delivery contracts. The stochastic load process is approximated by a scenario tree obtained by adapting a SARIMA model to historical data, using empirical means and variances of simulated scenarios to construct an initial tree, and reducing it by a scenario deletion procedure based on a suitable probability distance. Our model involves many mixed-integer variables and individual power unit constraints, but relatively few coupling constraints. Hence we employ stochastic Lagrangian relaxation that assigns stochastic multipliers to the coupling constraints. Solving the Lagrangian dual by a proximal bundle method leads to successive decomposition into single thermal and hydro unit subproblems that are solved by dynamic programming and a specialized descent algorithm, respectively. The optimal stochastic multipliers are used in Lagrangian heuristics to construct approximately optimal first stage decisions. Numerical results are presented for realistic data from a German power utility, with a time horizon of one week and scenario numbers ranging from 5 to 100. The corresponding optimization problems have up to 200,000 binary and 350,000 continuous variables, and more than 500,000 constraints.

Key words. Stochastic programming, Lagrangian relaxation, unit commitment, bundle methods, scenario generation.

AMS(MOS) subject classifications. 90C15, 90C90, 90C11, 90C25, 65K05

1. Introduction.

Many issues motivate a growing interest in mathematical modeling and optimization techniques for operating power systems and trading electricity. Some of them are related to the ongoing liberalization of electricity markets: electric utilities generate power in a competitive environment, generating and trading activities must be coordinated, electricity portfolios for spot and option markets become important, and the electrical load as well as electricity prices become increasingly unpredictable. Further issues are related to the complex nature of mathematical models for the efficient generation, transmission and distribution of electric power. They often lead to optimization problems characterized by combinations of challenges such as mixed-integer decisions, nonlinear

[*] Institute of Mathematics, Humboldt University Berlin, Berlin, Germany. E-mail: nicole@mathematik.hu-berlin.de.

[†] Systems Research Institute, Warsaw, Poland. E-mail: kiwiel@ibspan.waw.pl.

[‡] Institute of Mathematics, Humboldt University Berlin, Berlin, Germany. E-mail: mefju@mathematik.hu-berlin.de.

[§] Institute of Mathematics, Humboldt University Berlin, Berlin, Germany. E-mail: romisch@mathematik.hu-berlin.de.

[¶] Institute of Mathematics, Humboldt University Berlin, Berlin, Germany. E-mail: wegner@mathematik.hu-berlin.de.

costs and constraints, huge dimensions and data uncertainty. The latter aspect mostly concerns uncertainty in electric load forecasts, generator failures, stream flows to hydro reservoirs, and fuel and electricity prices (see [20, 22, 24, 31, 44] for relevant earlier work).

The present paper aims at optimizing generation and trading of an electric hydro-thermal based utility under data uncertainty. More specifically, we consider a power system comprising thermal units, pumped hydro storage plants and contracts for delivery and purchase. The relevant uncertain data comprise electric load, stream flows to hydro units, and fuel and electricity prices.

We develop a dynamic stochastic programming model where the expected production costs are minimized subject to operational constraints. Since the model contains stochastic mixed-integer decisions and is large-scale, new questions are raised on designing solution algorithms and generating approximate scenario-based data processes. Our model and solution techniques are validated on the system of the German utility Vereinigte Energiewerke AG (VEAG). The VEAG generation system consists of 25 (coal-fired or gas-burning) thermal units and 7 pumped hydro units. Its total capacity is about 13,000 megawatts (MW) including a hydro capacity of 1,700 MW; the system peak loads are about 8,600 MW.

Nowadays, solution methods are well developed for linear dynamic (multistage) stochastic programs without integrality constraints (see the monographs [4, 26, 57] and the surveys [3, 52]). Most of them are based on discrete approximations of the stochastic data process in the form of scenario trees. Recently, some algorithmic progress has also been achieved in mixed-integer stochastic programming models and applications to power optimization. The following algorithmic approaches to mixed-integer stochastic programs appear in the literature: (a) stochastic branch and bound methods [40], (b) scenario decomposition by splitting methods combined with suitable heuristics [50, 38, 54], (c) scenario decomposition combined with branch and bound [7, 6], (d) stochastic (augmented) Lagrangian relaxation of coupling constraints [1, 8, 9, 48, 11, 51, 55]. The approaches in (b) and (c) are based on a successive decomposition of the stochastic program into finitely many deterministic (or scenario) programs that may be solved by available conventional techniques. The approach of (d) hinges on a successive decomposition into finitely many smaller stochastic subproblems for which (efficient) solution techniques must be developed eventually. Due to the nonconvexity of the underlying stochastic program, the successive decompositions in (b)–(d) have to be combined with certain global optimization techniques (branch-and-bound, heuristics, etc.).

The solution approach pursued in the present paper consists in a *stochastic version* of classical Lagrangian relaxation [36], which is very popular in power optimization [2, 18, 23, 37, 53, 59, 61]. Since the coupling constraints contain random variables, stochastic multipliers are needed for their dualization, and the dual problem is a nondifferentiable stochastic program. The algorithm employed in [8, 9] is based on an augmented Lagrangian containing a quadratic penalization of constraint violation which is similar to a variance expression. The specific algorithm proposed in [9] linearizes the nonseparable penalty term and is no longer formally justified in nonconvex situations. In [1, 48] disaggregated bundle methods are used to solve a nondifferentiable dual which is obtained from a Lagrangian that is not based on expected constraint violations. The authors report on ill-conditioning effects when solving the quadratic subproblems and present a preconditioner leading to improved convergence results. The solution method employed in [55] is close to the approach in the present paper. It it is based on a quadratic approximation strategy of the Lagrangian for solving the dual and benefits from treating a purely thermal power system. The approach of the present paper is based on the Rockafellar-Wets dualization scheme (see also [11, 51]) and on the same, but *stochastic*, ingredients as in the classical case: a solver for the nondifferentiable dual, subproblem solvers, and a Lagrangian heuristic. With a state-of-the-art bundle method for solving the dual, specialized subproblem solvers and Lagrangian heuristics, this *stochastic Lagrangian relaxation* algorithm becomes rather efficient. Compared to our earlier work [43], the algorithm is developed for a general stochastic data process (i.e., containing stochastic prices and inflows, too). Furthermore, it ends with a Lagrangian heuristic providing nearly optimal primal solutions at each time period. Our numerical results indicate that the algorithm bears potential for solving complex real-life power scheduling models under uncertainty in reasonable time.

Generation of representative scenario trees is presently an active field of research; see the survey [14]. Known scenario generation methods may essentially be classified into two categories: (a) approaches that are embedded in the solution procedure of stochastic programs [10, 30, 27, 21, 17], and (b) approaches that generate optimal scenario trees for classes of stochastic optimization problems [45, 29, 60, 39]. For power management under uncertainty discrete time stochastic models are calibrated from historical time series for the load and stream flows [20, 55]. The calibrated models can be used to simulate or select a large number of sample paths. These independently generated data trajectories are combined into scenario trees. The algorithmic approaches in (a) allow possible updates of the scenario tree structure as part of the solution procedure in the case of linear or convex stochastic programs without integrality constraints. Since a sequence of stochastic programs corresponding to subsequent approximations have to be solved, the computational effort of all these methods is high. The tree building procedures in (b) control the goodness-of-fit of the approximation by certain distances. An optimal scenario tree is defined as the tree-structured discrete distribution that minimizes the selected distance. The resulting scenario trees can be tested within postoptimality analysis [12, 13]. The iterative procedure in [45] is based on the Wasserstein distance of probability measures. A weighted least-squares criterion is used in [29] to obtain a scenario tree that preserves certain moments or other statistical properties of the true multivariate distribution; the scenario tree is obtained by solving highly nonlinear nonconvex programs. [60] proposes a scenario reduction technique (nonrandom sampling) for the expectation of path-dependent discount functions.

In our approach to load scenario tree generation, simulation scenarios are drawn from a SARIMA model for the load. Their empirical means and standard deviations enter a tree building scheme for the initial (binary) load scenario tree. In a final step the number of load scenarios is reduced by a scenario deletion procedure based on a suitable probability distance.

The paper is organized as follows. In §2 we give a description of a hydro-thermal generation system and develop our stochastic programming model. In §3 we describe the stochastic Lagrangian relaxation approach together with its components and report on numerical results for the VEAG system with uncertain load. In §4 we present our procedure for generating scenario trees of the electrical load process and report on numerical tests.

2. Power system modeling. We consider a power generation system comprising thermal units, pumped storage plants and contracts for delivery and purchase, and describe a model for its cost-optimal operation under uncertainty in electrical load (i.e., demand), stream flows in hydro units and prices for fuel or electricity.

The scheduling horizon for unit commitment is typically discretized into uniform (e.g., hourly) intervals. Accordingly, the load, stream flows and prices are assumed to be constant within each time period. The scheduling decisions for thermal units are: which units to commit in each period, and at what generating capacity. The decision variables for hydro plants are the generation and pumping levels for each period. Contracts for delivery and purchase are regarded as special thermal units. The schedule should minimize the total generation costs, subject to the operational requirements.

We use the following notation. There are T time periods. I and J are the numbers of thermal and hydro units, respectively. For a thermal unit i in period t, $u_{it} \in \{0, 1\}$ is its *commitment* (1 if on, 0 if off), and p_{it} its *production*, with $p_{it} = 0$ if $u_{it} = 0$, $p_{it} \in [p_{it}^{\min}, p_{it}^{\max}]$ if $u_{it} = 1$, where $p_{it}^{\min}$ and $p_{it}^{\max}$ are the minimum and maximum capacities. Additionally, there are *minimum up/down-time requirements*: when unit i is switched on (off), it must remain on (off) for at least $\bar{\tau}_i$ ($\underline{\tau}_i$, resp.) periods. For a hydro plant j, v_{jt} and w_{jt} are its *generation* and *pumping* levels in period t, with upper bounds $v_{jt}^{\max}$ and $w_{jt}^{\max}$ respectively, and l_{jt} is the *storage* volume in the upper dam at the end of period t, with upper bound $l_{jt}^{\max}$. The water balance relates l_{jt} with $l_{j,t-1}$, v_{jt}, w_{jt} and the *water inflow* γ_{jt}, using the *pumping efficiency* η_j. The initial and final volumes are specified by l_j^{in} and l_j^{end}.

The basic system requirement is to meet the electric load. Another

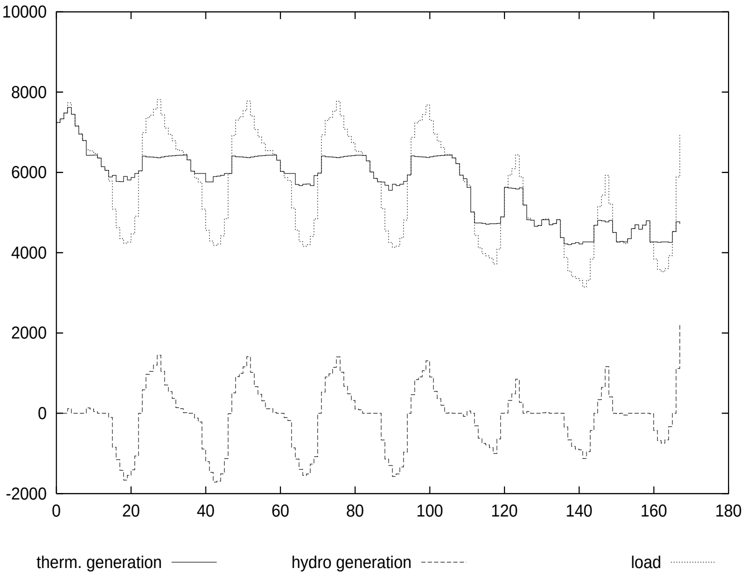

FIG. 1. *Typical load curve and hydro-thermal schedule*

important requirement is the spinning reserve constraint. To maintain reliability (compensate sudden load peaks or unforeseen outages of units) the total commited capacity should exceed the load in every period by a certain amount (e.g., a fraction of the demand). The *load* and the *spinning reserve* during period t are denoted by d_t and r_t, respectively.

Figure 1 shows a typical load curve and a corresponding cost-optimal hydro-thermal schedule. The load curve exhibits a daily cycle; also weekly cycles may occur (see, e.g., Fig. 5 in §4.1). Efficient operation of pumped storage hydro plants exploits such cycles by generating during peak load periods and pumping during off-peak periods.

Since the operating costs of hydro plants are usually negligible, the total system cost is given by the sum of startup and operating costs of all thermal units over the whole scheduling horizon. The *fuel cost* C_{it} for operating thermal unit i during period t has the form

$$C_{it}(p_{it}, u_{it}) := \max_{l=1:\bar{l}} \{\, a_{ilt} p_{it} + b_{ilt} u_{it} \,\}\,, \tag{2.1}$$

with coefficients a_{ilt}, b_{ilt} such that $C_{it}(\cdot, 1)$ is convex and increasing on $\mathbb{R}_+$; note that $C_{it}(0,0) = 0$. The startup cost of unit i depends on its downtime; it may vary from a maximum cold-start value to a much smaller value when the unit is still relatively close to its operating temperature.

This is modeled by the *startup cost*

$$S_{it}(u_i) := \max_{\tau=0:\tau_i^c} c_{i\tau}\left(u_{it} - \sum_{\kappa=1}^{\tau} u_{i,t-\kappa}\right), \tag{2.2}$$

where $0 = c_{i0} < \ldots < c_{i\tau_i^c}$ are fixed cost coefficients, τ_i^c is the cool-down time of unit i, $c_{i\tau_i^c}$ is its maximum cold-start cost, $u_i := (u_{it})_{t=1}^T$, and $u_{i\tau} \in \{0,1\}$ for $\tau = 1-\tau_i^c\colon 0$ are given initial values.

2.1. Stochastic model. In electric utilities, schedulers forecast the electric load for the required time span. Since the load is mainly driven by meteorological parameters (temperature, cloud cover, etc.), the actual load deviates from its prediction. Of course, the load uncertainty increases with the length of the planning horizon. Other sources of uncertainty are generator outages, stream flows in hydro units, and prices of fuel and electricity.

To formulate a power generation model that incorporates fluctuations in stream inflows in hydro plants, and fuel and electricity prices in addition to the load uncertainty, we use a probabilistic description of uncertainty. Thus

$$\{\,\boldsymbol{\rho}_t := (\boldsymbol{d}_t, \boldsymbol{r}_t, \boldsymbol{\gamma}_t, \boldsymbol{a}_t, \boldsymbol{b}_t, \boldsymbol{c}_t)\,\}_{t=1}^T \tag{2.3}$$

is assumed to be a discrete-time stochastic process on some probability space $(\Omega, \mathcal{F}, \mathcal{P})$, where $\boldsymbol{d}_t$, $\boldsymbol{r}_t$ and $\boldsymbol{\gamma}_t$ represent the load, the spinning reserve and the water inflows in period t, while $\boldsymbol{a}_t$, $\boldsymbol{b}_t$ and $\boldsymbol{c}_t$ collect the cost coefficients of (2.1) and (2.2) (we use bold characters to emphasize random elements).

The scheduling decisions for period t are made *after* learning the realization of the stochastic variables for that period. Denote by $\mathcal{F}_t \subseteq \mathcal{F}$ the σ-field generated by $\{\boldsymbol{\rho}_\tau\}_{\tau=1}^t$, i.e., the events observable till period t. Since the information on $\boldsymbol{\rho}_1$ is complete, $\mathcal{F}_1 = \{\emptyset, \Omega\}$, i.e., $\boldsymbol{\rho}_1$ is deterministic. By assuming $\mathcal{F}_T = \mathcal{F}$ we require that full information be available at the end of the planning horizon. The sequence of scheduling decisions $\{\boldsymbol{u}_t, \boldsymbol{p}_t, \boldsymbol{v}_t, \boldsymbol{w}_t\}$ also forms a stochastic process on $(\Omega, \mathcal{F}, \mathcal{P})$, which is assumed to be adapted to the filtration of σ-fields, i.e., *nonanticipative.* Nonanticipativity means that the decisions $(\boldsymbol{u}_t, \boldsymbol{p}_t, \boldsymbol{v}_t, \boldsymbol{w}_t)$ may depend only on the data observable till period t, or equivalently that $(\boldsymbol{u}_t, \boldsymbol{p}_t, \boldsymbol{v}_t, \boldsymbol{w}_t)$ is $\mathcal{F}_t$-measurable.

In a stochastic programming framework, an optimal schedule is obtained by minimizing the expectation of the costs caused by all nonanticipative decisions while meeting the operational constraints. Formally, our *stochastic problem* is stated as:

$$\min \mathbb{E}\left\{\sum_{t=1}^T \sum_{i=1}^I \left[\,\boldsymbol{C}_{it}(\boldsymbol{p}_{it}, \boldsymbol{u}_{it}) + \boldsymbol{S}_{it}(\boldsymbol{u}_i)\,\right]\right\} \quad \text{s.t.} \tag{2.4}$$

$$\text{(2.5a)}\quad p_{it}^{\min}\boldsymbol{u}_{it} \le \boldsymbol{p}_{it} \le p_{it}^{\max}\boldsymbol{u}_{it}, \quad \boldsymbol{u}_{it} \in \{0,1\}, \quad t = 1\colon T,\ i = 1\colon I,$$

$$\text{(2.5b)}\quad \boldsymbol{u}_{i\tau} - \boldsymbol{u}_{i,\tau-1} \le \boldsymbol{u}_{it}, \quad \tau = t - \bar{\tau}_i + 1\colon t-1,\ t = 1\colon T,\ i = 1\colon I,$$

$$\text{(2.5c)}\quad \boldsymbol{u}_{i,\tau-1} - \boldsymbol{u}_{i\tau} \le 1 - \boldsymbol{u}_{it}, \quad \tau = t - \underline{\tau}_i + 1\colon t-1,\ t = 1\colon T,\ i = 1\colon I,$$

$$\text{(2.6a)}\ 0 \le \boldsymbol{v}_{jt} \le v_{jt}^{\max},\ 0 \le \boldsymbol{w}_{jt} \le w_{jt}^{\max},\ 0 \le \boldsymbol{l}_{jt} \le l_{jt}^{\max},\ t = 1\colon T, j = 1\colon J,$$

$$\text{(2.6b)}\qquad \boldsymbol{l}_{jt} = \boldsymbol{l}_{j,t-1} - \boldsymbol{v}_{jt} + \eta_j \boldsymbol{w}_{jt} + \boldsymbol{\gamma}_{jt}, \quad t = 1\colon T,\ j = 1\colon J,$$

$$\text{(2.6c)}\qquad \boldsymbol{l}_{j0} = l_j^{\text{in}}, \quad \boldsymbol{l}_{jT} = l_j^{\text{end}}, \quad j = 1\colon J,$$

$$\text{(2.7a)}\qquad \sum_{i=1}^{I} \boldsymbol{p}_{it} + \sum_{j=1}^{J} (\boldsymbol{v}_{jt} - \boldsymbol{w}_{jt}) \ge \boldsymbol{d}_t, \quad t = 1\colon T,$$

$$\text{(2.7b)}\qquad \sum_{i=1}^{I} (\boldsymbol{u}_{it} p_{it}^{\max} - \boldsymbol{p}_{it}) \ge \boldsymbol{r}_t, \quad t = 1\colon T,$$

$$\text{(2.8)}\qquad (\boldsymbol{u}, \boldsymbol{p}, \boldsymbol{v}, \boldsymbol{w}) \in \mathop{\times}_{t=1}^{T} L^\infty \left(\Omega, \mathcal{F}_t, \mathcal{P}; \mathbb{R}^{2(I+J)}\right),$$

where (2.4) is the expected cost (cf. (2.1))–(2.2)), (2.5) describes the operating ranges and minimum up/down-time requirements of thermal units, (2.6) models the operating ranges and dynamics of hydro units (with $\boldsymbol{l}_t$ treated as state variables), (2.7) imposes the load and reserve requirements, (2.8) expresses the nonanticipativity constraint (since all decision variables are uniformly bounded, we may restrict attention to decisions in $L^\infty(\Omega, \mathcal{F}, P; \mathbb{R}^{2(I+J)})$), and for

$$\text{(2.9)}\qquad \tau_{\text{ini}} := 1 - \max_{i=1:I} \{\, \tau_i^c, \bar{\tau}_i - 1, \underline{\tau}_i - 1 \,\}$$

and $\tau = \tau_{\text{ini}}\colon 0$, $\boldsymbol{u}_{i\tau}$ in (2.4) (cf. (2.2)) and (2.5b)–(2.5c) are replaced by fixed initial values $u_{i\tau} \in \{0,1\}$, $i = 1\colon I$.

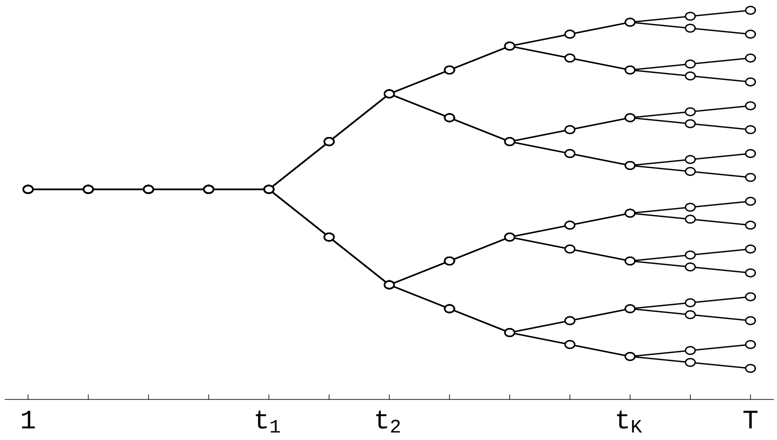

FIG. 2. *Example of a scenario tree*

2.2. Scenario tree model. To develop algorithms for problem (2.4)–(2.8), we now assume that we have a *discrete* distribution of the data process $\{\boldsymbol{\rho}_t\}_{t=1}^T$ (cf. (2.3)). Its support consists of *scenarios* (i.e., realizations of $\{\boldsymbol{\rho}_t\}_{t=1}^T$) that form a *scenario tree* based on a finite set of nodes $\mathcal{N}$ (cf. Fig. 2). The *root* node $n = 1$ stands for period $t = 1$. Every other node n has a unique *predecessor* node n_- and a *transition* probability $\pi_{n/n_-} > 0$, which is the probability of n being the successor of n_-. The successors to node n form the set $\mathcal{N}_+(n)$; their transition probabilities add to 1. The probability π_n of each node n is generated recursively by

$$\pi_1 = 1, \quad \pi_n = \pi_{n/n_-}\pi_{n_-} \quad \text{for } n \neq 1.$$

Nodes n with $\mathcal{N}_+(n) = \emptyset$ are called *leaves*; they constitute the *terminal* set $\mathcal{N}_T$. A scenario corresponds to a path from the root node to a leaf. The probabilities $\{\pi_n\}_{n\in\mathcal{N}_T}$ provide a distribution for the set of all scenarios. Conversely, given such scenario probabilities, the remaining node and transition probabilities are generated recursively by

$$\pi_n = \sum_{n_+\in\mathcal{N}_+(n)} \pi_{n_+}, \quad \pi_{n_+/n} = \pi_{n_+}/\pi_n \quad \text{for } n_+ \in \mathcal{N}_+(n).$$

Let path(n) denote the path from the root to node n. Then node n corresponds to a set of realizations of $\{\boldsymbol{\rho}_t\}_{t=1}^T$ that coincide until the period $t(n) := |\,\text{path}(n)|$ associated with node n; their common value $\boldsymbol{\rho}_{t(n)}$ is denoted by $\rho^n := (d^n, r^n, \gamma^n, a^n, b^n, c^n)$. Let the decisions for period t be made after learning the realization of $\{\boldsymbol{\rho}_t\}_{\tau=1}^t$. The scheduling decisions (u^n, p^n, v^n, w^n) assigned to nodes n in $\mathcal{N}_t := \{n : t(n) = t\}$ are realizations of the stochastic decisions $(\boldsymbol{u}_t, \boldsymbol{p}_t, \boldsymbol{v}_t, \boldsymbol{w}_t)$; note that $\sum_{n\in\mathcal{N}_t} \pi_n = 1$.

Let $u_i^{\mathrm{path}(n)} := (u_i^{\nu})_{\nu \in \mathrm{path}(n)}$. We use the following notation for the sequence of predecessors of any node $n \in \mathcal{N} \setminus \{1\}$: $n_{-1} := n_-$, $n_{-(\kappa+1)} := (n_{-\kappa})_-$ if $t(\kappa) > 1$; note that $t(n_{-\kappa}) = t(n) - \kappa$ for $\kappa = 1{:}\, t(n)-1$. To handle the initial values $u_i^{\tau} = u_{i\tau}$ with $\tau = \tau_{\mathrm{ini}}{:}\, 0$ (cf. (2.9)), we let $n_{\kappa} := \kappa - t(n)$ for $\kappa = t(n) + \tau_{\mathrm{ini}}{:}\, t(n)$ (as if the original tree were augmented with nodes $\tau = \tau_{\mathrm{ini}}{:}\, 0$ with associated periods $t(\tau) = \tau$). Then (cf. (2.1) and (2.2))

$$C_i^n(p_i^n, u_i^n) := \max_{l=1:\bar{l}} \{\, a_{il}^n p_i^n + b_{il}^n u_i^n \,\}$$

and

$$S_i^n\left(u_i^{\mathrm{path}(n)}\right) := \max_{\tau=0:\tau_i^c} c_{i\tau}^n \left(u_i^n - \sum_{\kappa=1}^{\tau} u_i^{n_{-\kappa}} \right) \tag{2.10}$$

are the fuel and startup costs of unit i at node n.

The *scenario-tree form* of the stochastic problem (2.4)–(2.8) reads:

$$\min \sum_{n \in \mathcal{N}} \pi_n \sum_{i=1}^{I} \left[C_i^n(p_i^n, u_i^n) + S_i^n\left(u_i^{\mathrm{path}(n)}\right) \right] \quad \text{s.t.} \tag{2.11}$$

$$p_{it(n)}^{\min} u_i^n \le p_i^n \le p_{it(n)}^{\max} u_i^n, \quad u_i^n \in \{0,1\}, \quad n \in \mathcal{N},\ i = 1{:}\, I, \tag{2.12a}$$

$$u_i^{n_{-\kappa}} - u_i^{n_{-(\kappa+1)}} \le u_i^n, \quad \kappa = 1{:}\, \bar{\tau}_i - 1,\ n \in \mathcal{N},\ i = 1{:}\, I, \tag{2.12b}$$

$$u_i^{n_{-(\kappa+1)}} - u_i^{n_{-\kappa}} \le 1 - u_i^n, \quad \kappa = 1{:}\, \underline{\tau}_i - 1,\ n \in \mathcal{N},\ i = 1{:}\, I, \tag{2.12c}$$

$$0 \le v_j^n \le v_{jt(n)}^{\max},\ 0 \le w_j^n \le w_{jt(n)}^{\max},\ 0 \le l_j^n \le l_{jt(n)}^{\max},\ n \in \mathcal{N},\ j = 1{:}\, J, \tag{2.13a}$$

$$l_j^n = l_j^{n_-} - v_j^n + \eta_j w_j^n + \gamma_j^n, \quad n \in \mathcal{N},\ j = 1{:}\, J, \tag{2.13b}$$

$$l_j^0 = l_j^{\mathrm{in}}, \quad l_j^n = l_j^{\mathrm{end}}, \quad n \in \mathcal{N}_T,\ j = 1{:}\, J, \tag{2.13c}$$

$$\sum_{i=1}^{I} p_i^n + \sum_{j=1}^{J} (v_j^n - w_j^n) \ge d^n, \quad n \in \mathcal{N}, \tag{2.14a}$$

$$\sum_{i=1}^{I} (u_i^n p_{it(n)}^{\max} - p_i^n) \ge r^n, \quad n \in \mathcal{N}, \tag{2.14b}$$

Table 1

Size of the scenario-tree model (2.11)–(2.14) *depending on the numbers of scenarios and nodes for* $T = 168$, $I = 25$ *and* $J = 7$

S	N	Variables		Constraints	Nonzeros
		binary	continuous		
1	168	4200	6652	13441	19657
20	1176	29400	45864	94100	137612
50	2478	61950	96642	198290	289976
100	4200	105000	163800	336100	491500

Note that the objective and constraints of (2.11)–(2.14) correspond directly to (2.4)–(2.7), whereas the nonanticipativity constraint (2.8) is handled *implicitly* (i.e., it is ensured automatically) by the tree-based model.

The tree-based form (2.11)–(2.14) for $N := |\mathcal{N}|$ nodes involves IN binary and $(I + 2J)N$ continuous decision variables. In contrast, the stochastic program (2.4)–(2.8) for $S := |\mathcal{N}_T|$ scenarios has ITS binary and $(I + 2J)TS$ continuous decision variables; note that typically $N \ll TS$.

Table 1 shows how the size of a mixed-integer LP formulation of the scenario-tree model (2.11)–(2.14) increases with the number of nodes (without taking into account the constraints of type (2.12b)–(2.12c) and the objective function).

3. Stochastic Lagrangian relaxation. In this section we develop Lagrangian duals of the stochastic program (2.4)–(2.8) and its tree-based version (2.11)–(2.14). We also describe the structure of Lagrangian relaxation, the bundle method used for solving the dual problem, the algorithms for solving subproblems and two Lagrangian heuristics for recovering primal solutions. Finally, we give numerical results.

3.1. Dual stochastic problem. Problem (2.4)–(2.8) is almost separable with respect to units, since only constraints (2.7) couple different units. This structure allows us to apply a stochastic version of Lagrangian relaxation by associating a stochastic Lagrange multiplier $\boldsymbol{\lambda}$ with the coupling constraints (2.7). For convex multistage stochastic programs, this approach is justified by the general duality theory of [49]. Hence suppose momentarily the constraint $\boldsymbol{u}_{it} \in \{0, 1\}$ of (2.5a) is relaxed to $\boldsymbol{u}_{it} \in [0, 1]$, so that problem (2.4)–(2.8) becomes convex. Then (cf. [11, §4]) with multipliers $\boldsymbol{\lambda} = (\boldsymbol{\lambda}^1, \boldsymbol{\lambda}^2)$ belonging to $\times_{t=1}^{T} L^1(\Omega, \mathcal{F}_t, \mathcal{P}; \mathbb{R}^2_+)$, the *Lagrangian*

$$(3.1)\quad L(\boldsymbol{u}, \boldsymbol{p}, \boldsymbol{v}, \boldsymbol{w}; \boldsymbol{\lambda}) := \mathbb{E} \sum_{t=1}^{T} \Bigg\{ \sum_{i=1}^{I} [\, \boldsymbol{C}_{it}(\boldsymbol{p}_{it}, \boldsymbol{u}_{it}) + \boldsymbol{S}_{it}(\boldsymbol{u}_i) \,]$$

$$+ \boldsymbol{\lambda}_t^1 \Big[\boldsymbol{d}_t - \sum_{i=1}^{I} \boldsymbol{p}_{it} - \sum_{j=1}^{J} (\boldsymbol{v}_{jt} - \boldsymbol{w}_{jt}) \Big] + \boldsymbol{\lambda}_t^2 \Big[\boldsymbol{r}_t - \sum_{i=1}^{I} (\boldsymbol{u}_{it} p_{it}^{\max} - \boldsymbol{p}_{it}) \Big] \Bigg\},$$

and the *dual function*

$$D(\boldsymbol{\lambda}) := \min_{(\boldsymbol{u},\boldsymbol{p},\boldsymbol{v},\boldsymbol{w})} L(\boldsymbol{u},\boldsymbol{p},\boldsymbol{v},\boldsymbol{w};\boldsymbol{\lambda}) \quad \text{s.t. constraints (2.5)–(2.6),} \tag{3.2}$$

the *dual problem* reads

$$\max\left\{ D(\boldsymbol{\lambda}) : \boldsymbol{\lambda} \in \mathop{\times}_{t=1}^{T} L^1(\Omega,\mathcal{F}_t,\mathcal{P};\mathbb{R}^2_+)\right\}. \tag{3.3}$$

In particular, this means that the stochastic multiplier process $\{\boldsymbol{\lambda}_t\}_{t=1}^T$ is nonnegative $\mathcal{P}$-almost surely and adapted to the filtration $\{\mathcal{F}_t\}_{t=1}^T$. In the general case of integrality constraints in (2.5a), the optimal value of the dual problem (3.3) only provides a lower bound for the optimal cost of the nonconvex primal problem (the duality gap is discussed in [11, §4]).

The minimization in (3.2) decomposes into stochastic single unit subproblems. Specifically, the dual function

$$D(\boldsymbol{\lambda}) = \sum_{i=1}^{I} D_i(\boldsymbol{\lambda}) + \sum_{j=1}^{J} \hat{D}_j(\boldsymbol{\lambda}^1) + \mathbb{E}\sum_{t=1}^{T}(\boldsymbol{\lambda}_t^1 \boldsymbol{d}_t + \boldsymbol{\lambda}_t^2 \boldsymbol{r}_t) \tag{3.4}$$

may be evaluated by solving the *thermal subproblems*

$$\begin{aligned} D_i(\boldsymbol{\lambda}) := \min_{\boldsymbol{u}_i} \Bigg\{ \mathbb{E}\sum_{t=1}^{T}[\min_{\boldsymbol{p}_{it}}\{\boldsymbol{C}_{it}(\boldsymbol{p}_{it},\boldsymbol{u}_{it}) - (\boldsymbol{\lambda}_t^1 - \boldsymbol{\lambda}_t^2)\boldsymbol{p}_{it}\} \\ - \boldsymbol{\lambda}_t^2 \boldsymbol{u}_{it} p_{it}^{\max} + \boldsymbol{S}_{it}(\boldsymbol{u}_i)] \quad \text{s.t. } (\boldsymbol{u}_i,\boldsymbol{p}_i) \in \mathop{\times}_{t=1}^{T} L^\infty(\Omega,\mathcal{F}_t,\mathcal{P};\mathbb{R}^2) \text{ and (2.5)}\Bigg\} \end{aligned} \tag{3.5}$$

(where we used separability and exchanged expectation with minimization over $\boldsymbol{p}_i$) and the *hydro subproblems*

$$\begin{aligned} \hat{D}_j(\boldsymbol{\lambda}^1) := \min_{(\boldsymbol{v}_j,\boldsymbol{w}_j)} \Bigg\{ \mathbb{E}\sum_{t=1}^{T} \boldsymbol{\lambda}_t^1(\boldsymbol{w}_{jt} - \boldsymbol{v}_{jt}) \text{ s.t.} \\ (\boldsymbol{v}_j,\boldsymbol{w}_j) \in \mathop{\times}_{t=1}^{T} L^\infty(\Omega,\mathcal{F}_t,\mathcal{P};\mathbb{R}^2) \text{ and (2.6)}\Bigg\}. \end{aligned} \tag{3.6}$$

Both subproblems represent multistage stochastic programming models for the operation of a single unit. While the thermal subproblem (3.5) is a combinatorial multistage program involving stochastic costs, the hydro subproblem (3.6) is a linear multistage model with stochastic costs and stochastic right-hand sides.

3.2. Dual scenario-based problem. Let us now assume that a discrete distribution of the data process $\{\boldsymbol{\rho}_t\}_{t=1}^T$ is given in the scenario tree form discussed in §2.2. Then $\{\boldsymbol{\lambda}_t\}_{t=1}^T$, being adapted to the filtration $\{\mathcal{F}_t\}_{t=1}^T$ generated by $\{\boldsymbol{\rho}_t\}_{t=1}^T$, has the tree structure of $\{\boldsymbol{\rho}_t\}_{t=1}^T$, and is nonnegative $\mathcal{P}$-almost surely. Accordingly, the multipliers $\lambda^n \in \mathbb{R}^2_+$ assigned to nodes n in $\mathcal{N}_t := \{n : t(n) = t\}$ are realizations of the stochastic multipliers $\boldsymbol{\lambda}_t$, for $t = 1{:}T$. Letting $\lambda := (\lambda^n)_{n\in\mathcal{N}} =: (\lambda_1, \lambda_2) \in \mathbb{R}^N_+ \times \mathbb{R}^N_+$, where $N := |\mathcal{N}|$, we may rewrite the dual problem (3.3), the decomposed dual objective (3.4) and the Lagrangian subproblems (3.5)–(3.6) as follows:

$$\max\left\{D(\lambda) : \lambda \in \mathbb{R}^{2N}_+\right\}, \tag{3.7}$$

$$D(\lambda) = \sum_{i=1}^{I} D_i(\lambda) + \sum_{j=1}^{J} \hat{D}_j(\lambda_1) + \sum_{n\in\mathcal{N}} \pi_n \left(\lambda^n_1 d^n + \lambda^n_2 r^n\right), \tag{3.8}$$

$$\begin{aligned} D_i(\lambda) = \min_{u_i} \Bigg\{ \sum_{n\in\mathcal{N}} \pi_n \Bigg[\min_{p^n_i} \{C^n_i(p^n_i, u^n_i) - (\lambda^n_1 - \lambda^n_2) p^n_i\} \\ - \lambda^n_2 u^n_i p^{\max}_{it(n)} + S^n_i\left(u^{\mathrm{path}(n)}_i\right)\Bigg] \text{ s.t. } (2.12)\Bigg\}, \end{aligned} \tag{3.9}$$

$$\hat{D}_j(\lambda_1) = \min_{(v_j, w_j)} \left\{ \sum_{n\in\mathcal{N}} \pi_n \lambda^n_1 (w^n_j - v^n_j) \text{ s.t. } (2.13)\right\}. \tag{3.10}$$

Alternatively, these expressions may be derived from the Lagrangian

$$\begin{aligned} L(u,p,v,w;\lambda) := \sum_{n\in\mathcal{N}} \pi_n \Bigg\{ \sum_{i=1}^{I} C^n_i(p^n_i, u^n_i) + \sum_{i=1}^{I} S^n_i\left(u^{\mathrm{path}(n)}_i\right) \\ + \lambda^n_1 \Bigg[d^n - \sum_{i=1}^{I} p^n_i - \sum_{j=1}^{J}(v^n_j - w^n_j)\Bigg] + \lambda^n_2 \Bigg[r^n - \sum_{i=1}^{I}(u^n_i p^{\max}_{it(n)} - p^n_i)\Bigg]\Bigg\}, \end{aligned} \tag{3.11}$$

and the definition of the dual function

$$D(\lambda) := \min_{(u,p,v,w)} L(u,p,v,w;\lambda) \quad \text{s.t. constraints (2.12)–(2.13).} \tag{3.12}$$

The dual function D is concave and polyhedral, since the fuel costs (2.1) are polyhedral.

3.3. Structure of the solution method. Extending Lagrangian relaxation approaches for deterministic power management models, our method for solving the tree-based model (2.11)–(2.14) consists of the following ingredients:

(a) Solving the dual problem (3.7) by a proximal bundle method using function and subgradient information;

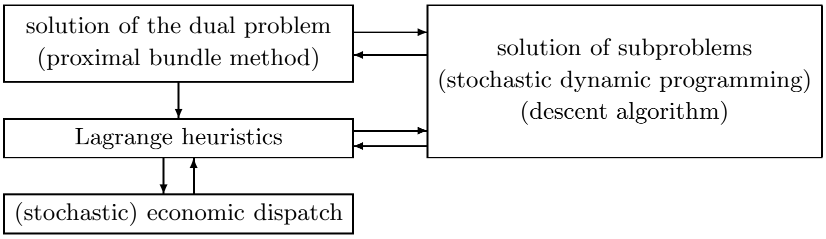
solution of the dual problem
(proximal bundle method)
solution of subproblems
(stochastic dynamic programming)
(descent algorithm)
Lagrange heuristics
(stochastic) economic dispatch

FIG. 3. *Structure of the stochastic Lagrangian relaxation method*

(b) Efficient solvers for the single unit subproblems: dynamic programming for (3.9) and a special descent algorithm for (3.10);

(c) Lagrangian heuristics for determining a nearly optimal first-stage decision that employ economic dispatch.

These components are discussed in the following subsections; their interaction is illustrated in Fig. 3.

3.4. Proximal bundle method. The tree-based problem (2.11)–(2.14) has the following form:

$$\psi_0^{\min} := \min\ \psi_0(z) \quad \text{s.t.} \quad \psi_l(z) \le 0,\ l = 1\colon L,\ z \in Z \tag{3.13}$$

with $z := (z_1, \dots, z_{I+J})$ and $Z := Z_1 \times \cdots \times Z_{I+J}$, where Z_i is the set of points $z_i := (u_i^n, p_i^n)_{n \in \mathcal{N}}$ satisfying (2.12) for $i = 1\colon I$, Z_{I+j} is the set of points $z_{I+j} := (v_j^n, w_j^n)_{n \in \mathcal{N}}$ satisfying (2.13) for $j = 1\colon J$, $L := 2N$, and

$$\psi_0(z) := \sum_{i=1}^{I} \sum_{n \in \mathcal{N}} \pi_n \left\{ C_i^n(p_i^n, u_i^n) + S_i^n \left(u_i^{\mathrm{path}(n)} \right) \right\}, \tag{3.14a}$$

$$\psi_n(z) := d^n - \sum_{i=1}^{I} p_i^n - \sum_{j=1}^{J} (v_j^n - w_j^n), \quad n = 1\colon N, \tag{3.14b}$$

$$\psi_{N+n}(z) := r^n - \sum_{i=1}^{I} (u_i^n p_{it(n)}^{\max} - p_i^n), \quad n = N+1\colon 2N. \tag{3.14c}$$

Note that each function ψ_l, $l = 0\colon L$, is continuous on the compact set Z.

Let Λ denote the dual space $\mathbb{R}^L$ of multipliers $\lambda = (\lambda_1, \lambda_2) \in \mathbb{R}^N \times \mathbb{R}^N$ equipped with the *probabilistic* inner product

$$\langle \lambda, \mu \rangle_\Pi := \sum_{n=1}^{N} \pi_n \left(\lambda_1^n \mu_1^n + \lambda_2^n \mu_2^n \right) = \langle \Pi \lambda, \mu \rangle, \tag{3.15}$$

where $\Pi \in \mathbb{R}^{L\times L}$ is a diagonal matrix with entries $\Pi_{nn} := \Pi_{N+n,N+n} := \pi_n$, $n = 1{:}\,N$, and $\langle\cdot,\cdot\rangle$ is the standard inner product on $\mathbb{R}^L$. Then, with the constraint function $\psi := (\psi_1, \dots, \psi_L)$, the Lagrangian (3.11) becomes

$$L(z;\lambda) := \psi_0(z) + \langle\lambda, \psi(z)\rangle_\Pi \tag{3.16}$$

(cf. (3.14)). Thus the dual function (3.12) of problem (3.13)

$$D(\lambda) := \min_{z\in Z} L(z;\lambda) = \min_{z\in Z} \{\psi_0(z) + \langle\lambda, \psi(z)\rangle_\Pi\}$$

may be evaluated at λ by finding a *partial Lagrangian solution*

$$z(\lambda) \in Z(\lambda) := \operatorname{Arg}\min_{z\in Z} L(z;\lambda) = \operatorname{Arg}\min_{z\in Z} \{\psi_0(z) + \langle\lambda, \psi(z)\rangle_\Pi\}, \tag{3.17}$$

which provides a *subgradient* $g_D(\lambda) := \psi(z(\lambda))$ of D at λ, i.e.,

$$D(\mu) \le L(z(\lambda);\mu) = D(\lambda) + \langle\mu - \lambda, g_D(\lambda)\rangle_\Pi \quad \forall\mu. \tag{3.18}$$

Clearly, $g_D(\cdot)$ is bounded, since ψ is continuous on the compact Z.

Suppose the primal problem (3.13) ($\equiv$(2.11)–(2.14)) is feasible. Then it has a nonempty solution set Z_* (by Weierstrass). Further, the lower bound $D_* := \sup_{\mathbb{R}^L_+} D \le \psi_0^{\min}$ (weak duality) yields $D_* < \infty$, so the *dual optimal set* $\Lambda_* := \max_{\mathbb{R}^L_+} D$ is nonempty (since D is polyhedral).

In effect, the proximal bundle method [32], [28, §XV.3] may be used for solving the dual problem [18]. This method generates a sequence $\{\lambda_c^k\}_{k=1}^\infty \subset \mathbb{R}^L_+$ converging to some $\lambda^* \in \Lambda_*$, and *trial points* $\lambda^k \in \mathbb{R}^L_+$ for evaluating the Lagrangian solutions $z^k := z(\lambda^k)$ (cf. (3.17)), the subgradients $g_D^k := \psi(z^k)$ of D and its *linearizations* (cf. (3.18))

$$D^k(\cdot) := D(\lambda^k) + \langle\cdot - \lambda^k, g_D^k\rangle_\Pi \ge D(\cdot),$$

starting from an arbitrary point $\lambda_c^1 = \lambda^1 \in \mathbb{R}^L_+$. Iteration k uses the *polyhedral model* of D

$$D_k(\cdot) := \min_{l\in L^k} D^l(\cdot) \quad \text{with} \quad k \in L^k \subset \{1{:}\,k\} \tag{3.19}$$

for finding the next trial point

$$\lambda^{k+1} := \arg\max\left\{ D_k(\lambda) - \tfrac{1}{2}u_k|\lambda - \lambda_c^k|_\Pi^2 : \lambda \in \mathbb{R}^L_+ \right\}, \tag{3.20}$$

where the *proximity weight* $u_k > 0$ and the penalty term $|\cdot|_\Pi^2 := \langle\cdot,\cdot\rangle_\Pi$ should keep λ^{k+1} close to the prox-center λ_c^k. An *ascent* step to $\lambda_c^{k+1} = \lambda^{k+1}$ occurs if λ^{k+1} is significantly better than λ_c^k as measured by

$$D(\lambda^{k+1}) \ge D(\lambda_c^k) + \kappa\delta_k, \tag{3.21}$$

where $\kappa \in (0,1)$ is a fixed Armijo-like parameter and

$$\delta_k := D_k(\lambda^{k+1}) - D(\lambda_c^k) \ge 0$$

is the *predicted ascent* (if $\delta_k = 0$ then $\lambda_c^k \in D_*$ and the method may stop). Otherwise, a *null* step $\lambda_c^{k+1} = \lambda_c^k$ improves the next model D_{k+1} with the new linearization D^{k+1} (cf. (3.19)).

The choice of weights u_k is discussed in [18, 32]. For choosing L^{k+1}, *subgradient selection* exploits the fact that the QP method of [34] for solving subproblem (3.20) produces multipliers $\nu_l^k \geq 0$ of the linear pieces D^l in (3.19) such that $\sum_{l\in L^k} \nu_l^k = 1$ and the set $\hat{L}^k := \{l \in L^k : \nu_l^k > 0\}$ satisfies $|\hat{L}^k| \leq L + 1$. To save storage without impairing convergence, it suffices to choose $L^{k+1} \supset \hat{L}^k \cup \{k+1\}$, i.e., we may drop inactive linearizations D^l with $\nu_l^k = 0$. (The multipliers ν_l^k could be used for constructing a generalized solution to a relaxed version of problem (3.13), and for recovering good primal feasible solutions; this idea is exploited for deterministic unit commitment in [18], but its stochastic extension requires further work.) Since subgradient selection may require too much storage (up to $L + 2$ linearizations), alternatively one may employ *subgradient aggregation* [32], in which groups of past linearizations are replaced by their convex combinations so that at most NGRAD ≥ 2 linearizations are stored.

The proximal bundle method has very strong convergence properties. First, because D is polyhedral, for subgradient selection the convergence is finite [33] (i.e., $\delta_k = 0$ and $\lambda_c^k \in \Lambda_*$ for some k) if the dual problem (3.7) satisfies a mild technical condition, or "sufficiently many" iterations require an exact ascent step, i.e., (3.21) with $\kappa = 1$. For subgradient aggregation, finite convergence need not occur, but $\lambda_c^k \to \lambda^* \in \Lambda_*$ and $\{z^k\}$ converges to $Z(\lambda^*)$ (cf. (3.17)). In particular, the thermal unit schedules $u_i^{(k)}$ of $z_i^k = (u_i^{(k)}, p_i^{(k)})$ converge to "dual optimal" schedules; this may be exploited in Lagrangian heuristics for recovering a good primal feasible solution. Further, $\delta_k \to 0$, so that for any *optimality tolerance* opt_tol > 0, the method eventually meets the stopping criterion

$$\delta_k \leq \text{opt\_tol}\left(1 + |D(\lambda_c^k)|\right). \tag{3.22}$$

Usually, when opt_tol $= 10^{-m}$ is used, upon termination the dual objective value $D(\lambda_c^k)$ has m correct digits [18].

We may add that using the probabilistic inner product (3.15) and norm $|\cdot|_\Pi := \langle \cdot, \cdot \rangle_\Pi^{1/2}$ in the Lagrangian (3.16) and the bundle subproblem (3.20) is natural in the stochastic setting. It may also enable faster convergence. Namely, in a similar context [1] reports poor bundle performance for Π replaced by the identity matrix in (3.16) and (3.20), and much better performance for Π replaced by $\Pi^{1/2}$ in (3.16) and by the identity matrix in (3.20); the latter version corresponds to ours (expressed in variables $\bar{\lambda} = \Pi^{1/2}\lambda$).

3.5. Descent algorithm for stochastic hydro units and economic dispatch. The hydro subproblem (3.10) for unit j is solved by a specialized descent method that generates a finite sequence of feasible hydro decisions (v_j, w_j) with decreasing objective values

$$\Psi(v_j, w_j) := \sum_{n \in \mathcal{N}} \pi_n \lambda_1^n (v_j^n - w_j^n)$$

and terminates with an optimal solution. The method begins by finding a feasible hydro decision (v_j, w_j) that satisfies (2.13). The next feasible iterate $(\tilde{v}_j, \tilde{w}_j)$ with $\Psi(\tilde{v}_j, \tilde{w}_j) < \Psi(v_j, w_j)$ is chosen so that the difference $(\tilde{v}_j^n, \tilde{w}_j^n) - (v_j^n, w_j^n)$ is nonzero only for n belonging to a rather small subset $\mathcal{N}_G$ of $\mathcal{N}$. Here the subscript G refers to a subset of $\mathcal{N}$ with the following properties: There exist $n_G \in G$ and $L_G \subseteq G$ such that $n_G \in \mathrm{path}(n)$ for each $n \in G$, $\mathcal{N}_+(n) \cap G = \emptyset$ for each $n \in L_G$, and $\mathcal{N}_+(n) \subseteq G$ for each $n \in G \setminus L_G$. Since such a subset G corresponds to a subtree with root node n_G and leaves in L_G, it is called a *descent subtree* in what follows.

It is shown in [42] that for each nonoptimal feasible hydro decision (v_j, w_j) there exist a descent subtree G and a hydro decision $(\tilde{v}_j, \tilde{w}_j)$ such that $\tilde{v}_j^n = v_j^n$ and $\tilde{w}_j^n = w_j^n$ for each node $n \in \mathcal{N} \setminus \mathcal{N}_G$ with $\mathcal{N}_G = \{n_G\} \cup L_G$, and

$$\sum_{n \in \mathcal{N}_G} \pi_n \lambda_1^n (\tilde{v}_j^n - v_j^n - (\tilde{w}_j^n - w_j^n)) < 0,$$

which implies $\Psi(\tilde{v}_j, \tilde{w}_j) < \Psi(v_j, w_j)$. Moreover, there exists a constant $\delta_G \neq 0$ such that $\tilde{l}_j^n = l_j^n + \delta_G$ for $n \in G \setminus L_G$ and $\tilde{l}_j^n = l_j^n$ for $n \in \mathcal{N} \setminus (G \setminus L_G)$, where $\tilde{l}_j$ and l_j are the corresponding storage volumes. If $\delta_G > 0$ then $\tilde{v}_j^{n_G} < v_j^{n_G}$ or $\tilde{w}_j^{n_G} > w_j^{n_G}$, and $\tilde{w}_j^n < w_j^n$ or $\tilde{v}_j^n > v_j^n$ for each $n \in L_G$, and similarly for $\delta_G < 0$. For a precise description of the iterative scheme we refer to [42]. It is also shown there that for each nonoptimal feasible hydro decision, a descent subtree leading to steepest descent of Ψ can be determined with complexity that grows linearly with N. Implementation issues and numerical results of the descent algorithm are given in [41, 42].

We now turn to the next item of Fig. 3. When the binary decisions u_i^n are fixed, the tree-based model (2.11)–(2.14) becomes an *economic dispatch problem.* This problem can be reformulated as

$$\min \sum_{n \in \mathcal{N}} \pi_n \Phi^n \left(\sum_{j=1}^{J} (v_j^n - w_j^n) \right) \quad \text{s.t. (2.13)}, \tag{3.23}$$

where Φ^n are the optimal value functions of the following one-parametric thermal subproblems

$$\Phi^n(\theta) := \min_p \left\{ \sum_{i=1}^{I} C_i^n(p_i, u_i^n) : p_{it(n)}^{\min} u_i^n \le p_i \le p_{it(n)}^{\max} u_i^n,\ i = 1\colon I, \right.$$
$$\left. d^n - \theta \le \sum_{i=1}^{I} p_i \le \sum_{i=1}^{I} p_{it(n)}^{\max} u_i^n - r^n \right\}.$$

Such piecewise linear functions may be evaluated via efficient algorithms (e.g., [56]). If the functions Φ^n were differentiable, successive linearization combined with the above descent technique could be used to solve (3.23). This suggests replacing each Φ^n by a differentiable function $\tilde{\Phi}^n$ that is obtained from Φ^n by smoothing its kinks with quadratic functions on small intervals. Then successive linearization and descent may be combined with progressive reduction of the smoothing intervals. More information on this economic dispatch algorithm and its numerical performance may be found in [42, 43].

3.6. Dynamic programming for stochastic thermal units. To solve the thermal subproblem (3.9) for unit i by dynamic programming, the startup costs (2.2) and the minimum up/down-times (2.12b)–(2.12c) are incorporated in its state space $\mathcal{S}_i := \{-\hat{\tau}_i\colon -1\} \cup \{1\colon \bar{\tau}_i\}$ with $\hat{\tau}_i := \max\{\tau_i^c, \underline{\tau}_i\}$. Unit i is in state $s > 0$ ($s < 0$) if it has been up (down) for at least s ($-s$, resp.) time periods. The set $\mathcal{T}_i \subseteq \mathcal{S}_i \times \mathcal{S}_i$ of feasible state transitions of unit i is given by

$$\begin{aligned}\mathcal{T}_i := \{&(s, s+1) \text{ for } s = 1\colon \bar{\tau}_i - 1, (\bar{\tau}_i, \bar{\tau}_i), (\bar{\tau}_i, -1), (-\hat{\tau}_i, -\hat{\tau}_i),\\ &(s, s-1) \text{ for } s = -\hat{\tau}_i - 1\colon -1,\ (s, 1) \text{ for } s = -\hat{\tau}_i\colon -\underline{\tau}_i\}\,.\end{aligned}$$

To formulate the dynamic programming recursion, we set for all nodes $n \in \mathcal{N}$ and integers $s, \tilde{s}$

$$\alpha_i^n(s) := \begin{cases} 0 & \text{if } s < 1,\\ \displaystyle\min_{p_{it(n)}^{\min} \le p \le p_{it(n)}^{\max}} [C_i^n(p, 1) - (\lambda_1^n - \lambda_2^n)p] - \lambda_2^n p_{it(n)}^{\max} & \text{else,}\end{cases}$$

$$\beta_i^n(s, \tilde{s}) := \begin{cases} c_{i,-s}^n & \text{if } s \in \{-\tau_i^c\colon -1\} \text{ and } \tilde{s} > 0,\\ 0 & \text{otherwise,}\end{cases}$$

where $c_{i\tau}^n$ are the startup cost coefficients of (2.10). Thus $\alpha_i^n(s)$ is the weight of node n in state s, and $\beta_i^n(s, \tilde{s})$ is the weight for the arc from state s to state $\tilde{s}$ at node n in the dynamic programming graph. Then we have

$$\begin{aligned} D_i(\lambda) &= \min_{u_i} \sum_{n \in \mathcal{N}} \pi_n \left[\alpha_i^n(u_i^n) + \max_{\tau = 0\colon \tau_i^c} c_{i\tau}^n \left(u_i^n - \sum_{\kappa=1}^{\tau} u_i^{n-\kappa} \right) \right] \\ &= \min \left\{ \sum_{n \in \mathcal{N}} \pi_n \left[\alpha_i^n(s^n) + \beta_i^n(s^{n-}, s^n) \right] : (s^{n-}, s^n) \in \mathcal{T}_i,\ n \in \mathcal{N} \right\} \\ &= \eta_i^0(s_i^0), \end{aligned}$$

where s_i^0 is the initial state determined by the given $\{u_{i\tau}\}_{\tau=\tau_{\text{ini}}}^0$, and $\eta_i^0(s)$ is determined by the backward recursion

$$\eta_i^n(s) = \alpha_i^n(s) + \sum_{n_+ \in \mathcal{N}_+(n)} \pi_{n_+/n} \min_{(s,\tilde{s}) \in \mathcal{T}_i} \left\{ \beta_i^{n_+}(s, \tilde{s}) + \eta_i^{n_+}(\tilde{s}) \right\}, \quad s \in \mathcal{S}_i,$$

for $n \in \mathcal{N} \cup \{0\}$ with $\alpha_i^0(s) \equiv 0$, $\mathcal{N}_+(0) = \{1\}$, $\pi_{1/0} = 1$. Now, the dynamic programming algorithm works as follows. First the cost-to-go $\eta_i^n(s)$ is computed for all states $s \in \mathcal{S}_i$ and nodes $n \in \mathcal{N}$ via the backward recursion, which also yields $\eta_i^0(s_i^0)$. Then the optimal scheduling decisions $\{(u_i^n(\lambda), p_i^n(\lambda))\}_{n \in \mathcal{N}}$ are obtained by forward tracing the tree. Implementation issues are discussed in more detail in [42].

3.7. Lagrangian heuristics. When the bundle method delivers an optimal multiplier λ^*, the optimal value $D(\lambda^*)$ provides a lower bound for the optimal cost of the model (2.11)–(2.14). In general, however, the "dual optimal" scheduling decisions $z(\lambda^*) = (u(\lambda^*), p(\lambda^*), v(\lambda^*), w(\lambda^*))$ (cf. (3.17)) violate the load and reserve constraints (2.14).

In practice the data forecast may be reliable until some period $t_1 \in \{1\colon T-1\}$, so that the data process $\{\boldsymbol{\rho}_t\}_{t=1}^{t_1}$ is deterministic. Thus it is useful to distinguish the *deterministic first stage* comprising periods $t = 1\colon t_1$. The nodes of the first stage form the set $\mathcal{N}_{\text{first}} := \cup_{t=1}^{t_1} \mathcal{N}_t$ (see also Fig. 2).

In the following, we describe two Lagrangian heuristics that determine nearly optimal first stage decisions $\{(u^n, p^n, v^n, w^n)\}_{n \in \mathcal{N}_{\text{first}}}$ starting from the optimal multiplier λ^* and $z(\lambda^*)$. While the first heuristic provides a nearly optimal decision only at nodes $n \in \mathcal{N}_{\text{first}}$, the result of the second one is a nearly optimal solution at every node in $\mathcal{N}$.

Our first heuristic LH1 starts by computing mean values of the scenario-based stochastic processes ρ, λ^* and $l_j = l_j(\lambda^*)$, $j = 1\colon J$, i.e., we determine $\bar{\rho} = \mathbb{E}[\rho]$, $\bar{\lambda}^* = \mathbb{E}[\lambda^*]$ and $\bar{l}_j = \mathbb{E}[l_j]$. For instance, we have

$$(\bar{d}_t, \bar{r}_t, \bar{\gamma}_t, \bar{a}_t, \bar{b}_t, \bar{c}_t) = \bar{\rho}_t = \sum_{n \in \mathcal{N}_t} \pi_n \rho^n = \sum_{n \in \mathcal{N}_t} \pi_n (d^n, r^n, \gamma^n, a^n, b^n, c^n).$$

Next, replacing $\mathcal{N}$ by $\{1\colon T\}$ and ρ by $\bar{\rho}$, we consider deterministic single-scenario versions of the model (2.11)–(2.14) and the thermal subproblems (3.9). Then we find deterministic generation and pumping decisions v_j and w_j that satisfy the constraints (2.13) with l_j and γ_j replaced by $\bar{l}_j$ and $\bar{\gamma}_j$, respectively. Furthermore, deterministic on/off decisions u_i are computed by dynamic programming as solutions of the thermal subproblems (3.9) with the multiplier λ and the cost coefficients a, b and c replaced by $\bar{\lambda}^*$, $\bar{a}$, $\bar{b}$ and $\bar{c}$. In the next step, the hydro decisions v_j and w_j are rescheduled in order to meet, as much as possible, the modified reserve constraint

$$\sum_{i=1}^{I} u_{it} p_{it}^{\max} \geq \bar{d}_t + \bar{r}_t + \sum_{j=1}^{J} (w_{jt} - v_{jt}) \quad t = 1\colon T, \tag{3.24}$$

i.e., the sum of the load and reserve constraints (2.14a) and (2.14b) with d and r replaced by $\bar{d}$ and $\bar{r}$. To this end our procedure reduces the right-hand side of (3.24) by modifying the hydro schedules at those t where the constraint is violated and its right-hand side is largest in a certain set of neighboring time periods. This procedure is repeated several times (see also

[23]). In the next step the hydro variables are fixed, and following [61] we search for binary variables u_i that satisfy the constraint (3.24). The main idea is to select the period t where (3.24) is most violated and to increase $\bar{\lambda}_t^*$ as much as necessary to switch on in the thermal subproblems just as many units as needed to satisfy (3.24) at t. This is repeated until the constraint (3.24) is satisfied in all periods. Since this technique does not distinguish between identical units, which appear quite often in practice, the startup costs of such units are slightly modified. Once the binary decisions u_i are fixed, the economic dispatch algorithm (see §3.5 and [43]) completes LH1 by providing (deterministic) scheduling decisions $\{p_t, v_t, w_t\}$ for the whole planning horizon $t = 1{:}T$.

The second Lagrangian heuristic LH2 is based on the observation that usually the binary decisions in $u(\lambda^* + \varepsilon\underline{1})$ change significantly relative to $u(\lambda^*)$ even for small $\varepsilon > 0$, and ensure feasibility for ε large enough. (Here $\underline{1}$ denotes the L-vector with unit components.) Hence, LH2 starts by finding some $\varepsilon > 0$ such that $z(\lambda^* + \varepsilon\underline{1})$ satisfies all constraints (2.12)–(2.14). Then taking $u(\lambda^* + \varepsilon\underline{1})$ as a starting point, a finite sequence of binary decisions is constructed such that their components are decreasing. This is done by selecting a node $n \in \mathcal{N}$ where the available reserve capacity $\sum_{i=1}^{I}(u_i^n p_{it(n)}^{\max} - p_i^n) - r^n$ is maximal, and switching some unit i off at n and some predecessor and successor nodes. This unit i and the neighboring nodes of n are detected by stochastic dynamic programming. Next, a stochastic economic dispatch problem is solved by the descent method described in §3.5 and [43]. This procedure, which generates a sequence of scheduling decisions at all nodes, is continued until infeasibility is detected during economic dispatch, at which point the procedure returns the scheduling decision having minimal cost (2.11) from those calculated so far. The LH2 heuristic is described in more detail in [42, §4.4].

3.8. Numerical results. The stochastic Lagrangian relaxation algorithm was implemented in C++ except for the proximal bundle method, for which the Fortran package NOA 3.0 [35] was used as a callable library. For numerical tests we considered the hydro-thermal power system of VEAG (with $T = 168$, $I = 25$ and $J = 7$) under uncertain load (i.e., the remaining data were deterministic). A bunch of load scenario trees was constructed as follows. Starting with a reference load scenario obtained from real-life data, $S - 1$ random branching points were selected successively to produce a scenario tree with S identical scenarios. Then a (discretized) Brownian motion was added to each node of the scenario tree. The test runs were performed on an HP 9000 (780/J280) computer with 180 MHz frequency and 768 MByte main memory under HP-UX 10.20.

First we consider the Lagrangian relaxation algorithm based on LH1. Table 2 shows computing times and gaps for different numbers of scenarios (S) and four randomly generated scenario trees, each having a different

Table 2
Computing times and gaps with LH1 (NOA 3.0: $\text{opt\_tol} = 10^{-3}$, $\text{NGRAD} = 50$*)*

S	N	time[s]	gap[%]	N	time[s]	gap[%]
20	1982	89	0.15	1627	94	0.10
20	1651	68	0.37	1805	85	0.07
50	4530	475	0.18	4060	274	0.10
50	4041	313	0.10	4457	288	0.43
100	9230	1183	0.11	9224	1072	0.13
100	7727	930	0.09	8867	1234	0.30

Table 3
Computing times and gaps with LH2 (NOA 3.0: $\text{opt\_tol} = 10^{-5}$, $\text{NGRAD} = 200$*)*

S	N	NOA time[s]	total time[s]	gap[%]
1	168	10	16	0.20
5	542	65	101	0.19
10	983	128	230	0.71
21	2098	351	531	0.39
24	2175	374	695	0.83
27	2208	380	8349	0.73
32	2173	359	3337	0.66
34	3043	497	1499	0.95
39	3848	874	4092	0.82

number of nodes (N). The gap refers to the relative difference

$$\frac{1}{D_*}\left(\sum_{t=1}^{T}\sum_{i=1}^{I}\left[C_{it}(p_{it},u_{it})+S_{it}(u_i)\right]-D_*\right)$$

of the cost of the scheduling decision (u,p,v,w) and the optimal value D_* of the dual problem. We note that, in general, this gap does not provide a quality measure for the approximate first stage solution (it may even become nonpositive). When reading the computing times in Table 2, it is worth recalling that $N = 4000$ and $N = 8000$ correspond to $100,000$ and $200,000$ binary variables in the model (2.11)–(2.14), respectively.

Table 3 reports computing times and gaps for the Lagrangian relaxation algorithm based on LH2 applied to test problems with different numbers S and N of scenarios and nodes of randomly generated load scenario trees. Here the gap refers to the following bound of the relative duality gap

$$\frac{1}{D_*}\left(\sum_{n\in\mathcal{N}}\pi_n\sum_{i=1}^{I}\left[C_i^n(p_i^n,u_i^n)+S_i^n\left(u_i^{\text{path}(n)}\right)\right]-D_*\right).$$

Clearly, this bound provides an accuracy certificate for the approximate primal-feasible solution $\{(u^n,p^n,v^n,w^n)\}_{n\in\mathcal{N}}$.

While the "deterministic" heuristic LH1 requires only short computing times, this becomes quite different for the "stochastic" heuristic LH2. Table 3 gives more insight into the (total) computing times of different test runs.

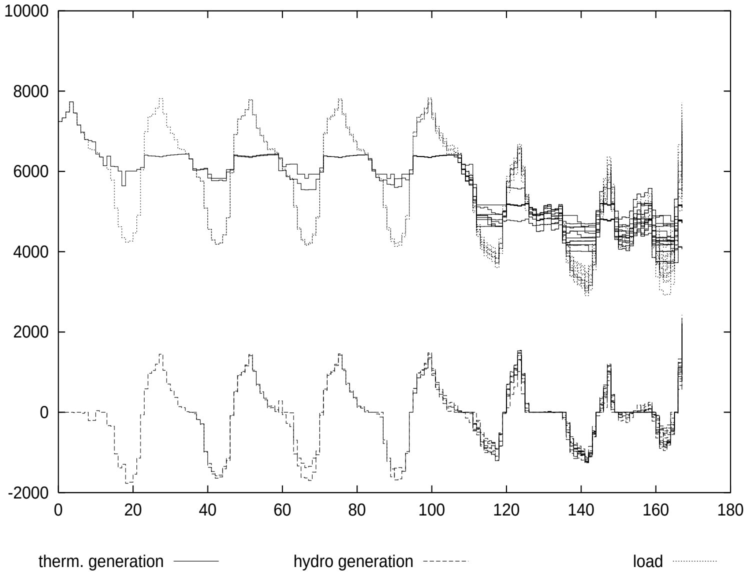

FIG. 4. *Optimal stochastic solution for one week*

Higher computing times are always due to very many economic dispatches required by LH2. It is worth mentioning here that LH2 is quite sensitive to the accuracy of the dual solution, i.e., to the optimality tolerance of the proximal bundle method. The advantage of using LH1 consists in low running times even for mid-size scenario trees, while its drawbacks are that only first-stage solutions are provided with no accuracy bounds. The advantage of LH2 is that it produces a "stochastic" solution together with a guaranteed accuracy bound, but at the expense of higher computing times even for scenario trees of smaller size. For further information the interested reader is referred to [42].

Another test employed a load scenario tree with sixteen scenarios and 912 nodes that was generated from real-life VEAG data by the technique described in §4.4. As before, we had $T = 168$, $I = 25$, $J = 7$. In effect, the scenario tree formulation of our optimization model had 22,800 binary and 41,952 continuous variables, 92,224 constraints and 242,704 nonzeros. Figure 4 provides the final output of the Lagrangian relaxation algorithm using LH2. It presents 16 realizations of load and generation levels.

4. Generation of load scenario trees. Our generation of load scenario trees for the stochastic power generation model (2.11)–(2.14) proceeds according to the following steps:

1. Identify a statistical (time series or regression) model of the load, and use it for generating a large number of simulation scenarios.

2. Determine an initial structure of the load tree. Compute scenario

values, using the sample means and standard deviations of the simulated scenarios.

3. Reduce the number of scenarios in the tree optimally.

These steps are explained in the following subsections.

4.1. Identification of a time series for the electric load. For the identification of a statistical model we got from the VEAG utility an hourly load profile for one year. We could not fit regression models because of missing meteorological parameters.

To select a suitable class of models for the set of *observed load data* $\{d_t\}_{t\in\mathbb{I}}$ with $\mathbb{I} \subset \mathbb{Z} := \{0, \pm 1, \pm 2, \ldots\}$, $\{d_t\}_{t\in\mathbb{I}}$ is considered as part of a realization of the stochastic load process $\{\boldsymbol{d}_t\}_{t\in\mathbb{Z}}$. A *time series model* for $\{d_t\}_{t\in\mathbb{I}}$ is a specification of the joint distributions of $\{\boldsymbol{d}_t\}_{t\in\mathbb{Z}}$. We now recall some concepts of time series analysis.

A complete time series model for a stochastic process $\{X_t\}_{t\in\mathbb{Z}}$ should specify the distribution of any random vector $(X_{i_1}, \ldots, X_{i_l})$. Often the analysis focuses on the second-order properties of $\{X_t\}$: the expected values $\mathbb{E}X_t$ and the covariances $\operatorname{cov}(X_t, X_s) := \mathbb{E}[(X_t - \mathbb{E}X_t)(X_s - \mathbb{E}X_s)]$ for all t, s. In the particular case of *Gaussian time series* all random variables X_t are normally distributed. Therefore all the joint distributions are multivariate normal and completely characterized by the second-order properties of $\{X_t\}$. Classical time series analysis relies on the concept of stationarity. Recall that $\{X_t\}$ is *stationary* if $\mathbb{E}X_t^2 < \infty$, $\mathbb{E}X_t$ is constant and $\operatorname{cov}(X_r, X_s) = \operatorname{cov}(X_{r+t}, X_{s+t})$, $\forall r, s, t \in \mathbb{Z}$.

To select an appropriate model for observed data, their properties are analyzed first. In particular, the data graph is searched for any *seasonal* (periodic) or *trend* (nonconstant mean) components, outlying observations or sharp changes in behavior. Then suitable transformations are applied to the data to get a new stationary series (*residuals*) with zero mean and unit variance. The trend and seasonal components may be removed by estimating these components and subtracting them from the data; this is the classical decomposition model incorporating trend, a seasonal component and random noise. Another transformation is called *differencing*; it replaces $\{X_t\}$ by $\{Y_t := X_t - X_{t-s}\}$ for some *lag* $s \in \mathbb{N}$, thus eliminating a seasonal component of period s.

Figures 5 and 6 highlight the periodic components of our historical data. In the week and month load data there is clearly a recurring pattern with the seasonal period of 24 (one day). There are further periodic components of length 168 (one week) and change points in the year data due to the start/end of the daylight saving time.

Most approaches for fitting a time series to the deseasonalized data rely on linear models. *Autoregressive moving average* (ARMA) models are characterized by finite-order linear difference equations with constant coefficients. The process $\{X_t\}$ is called *ARMA*(p,q) if it is stationary and

$$X_t - \phi_1 X_{t-1} - \ldots - \phi_p X_{t-p} = Z_t + \theta_1 Z_{t-1} + \ldots + \theta_q Z_{t-q} \quad \forall t, \tag{4.1}$$

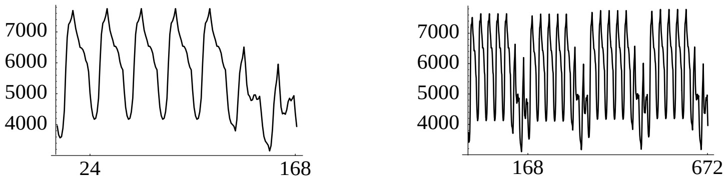

FIG. 5. *Time plot of the load profile for one week (left) and for one month (right)*

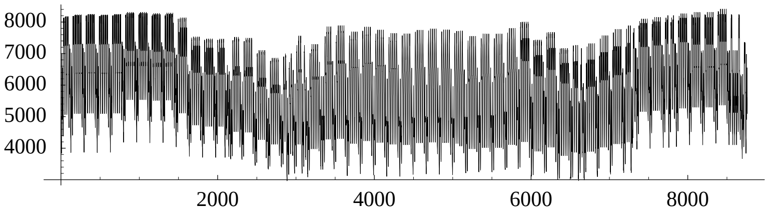

FIG. 6. *Time plot of the load profile for one year*

where $(\phi_k)_{k=1}^p$ and $(\theta_l)_{l=1}^q$ are real coefficients and $\{Z_t\}_{t\in\mathbb{Z}}$ is the *white noise* process $\mathrm{WN}(0,\sigma^2)$ with zero mean and variance σ^2, i.e., $\mathbb{E}Z_t = 0$, $\mathbb{E}Z_t^2 = \sigma^2$, $\forall t \in \mathbb{Z}$, and $\mathbb{E}Z_r Z_t = 0$ if $r \neq t$. Using the *backward shift operator* B defined by $B^\ell X_t := X_{t-\ell}$ for $t, \ell \in \mathbb{Z}$, the ARMA equations (4.1) can be rewritten as

$$\phi(B)X_t = \theta(B)Z_t, \quad \forall t \in \mathbb{Z}, \quad \{Z_t\} \sim \mathrm{WN}(0,\sigma^2),$$

where ϕ and θ denote the polynomials $\phi(z) = 1 - \phi_1 z - \ldots - \phi_p z^p$, $\theta(z) = 1 + \theta_1 z + \ldots + \theta_q z^q$. An ARMA$(p,q)$ process $\{X_t\}_{t\in\mathbb{Z}}$ is said to be *causal* (or *future-independent*) if there exists a real sequence $\{\psi_\ell\}$ such that $\sum_{\ell=0}^\infty \psi_\ell < \infty$ and

$$X_t = \sum_{\ell=0}^\infty \psi_\ell Z_{t-\ell}, \quad \forall t \in \mathbb{Z}.$$

If the differenced series $\{Y_t = (1 - B^s)X_t\}_{t\in\mathbb{Z}}$ is an ARMA(p,q) process then the model for the original series $\{X_t\}$ reads $\phi(B)(1-B^s)X_t = \theta(B)Z_t$; further, $\{X_t\}$ belongs to the class of *seasonal autoregressive integrated moving average* (SARIMA) processes if $\{Y_t\}$ is causal. General SARIMA processes are defined as follows. The process $\{X_t\}_{t\in\mathbb{Z}}$ is said to be a *SARIMA*$(p,d,q) \times (P,D,Q)_S$ *process with period* s if the differenced process $Y_t := (1-B)^d(1-B^S)^D X_t$ is the causal ARMA process

$$\phi(B)\Phi(B^s)Y_t = \theta(B)\Theta(B^s)Z_t, \quad \{Z_t\} \sim WN(0,\sigma^2),$$

where $\phi(z) = 1 - \ldots - \phi_p z^p$, $\Phi(z) = 1 - \ldots - \Phi_P z^P$, $\theta(z) = 1 + \ldots + \theta_q z^q$ and $\Theta(z) = 1 + \ldots + \Theta_Q z^Q$. Then the model for $\{X_t\}_{t\in\mathbb{Z}}$ reads $\phi(B)\Phi(B^S)(1-B)^d(1-B^S)^D X_t = \theta(B)\Theta(B^S)Z_t$.

There is no single systematic approach to identifying SARIMA models of higher order; see, e.g., [5]. To determine a suitable SARIMA model for a given time series, the differencing orders d, D and the length S of the seasonal component must be identified. Characteristics of the original time series like trend and substantial periodic components are reflected in the empirical autocorrelation function, the empirical counterpart of the *autocorrelation function* $\operatorname{cov}(X_\ell, X_0)/\operatorname{var}(X_0)$, $\ell \in \mathbb{Z}$. The length of the seasonal component S can be discovered by inspecting the periodicity of the empirical autocorrelation function, and the seasonal components are eliminated by differencing the data D times with lag S. Next, d is chosen so that differencing d times with lag 1 gives residuals $Y_t := (1-B)^d(1-B^S)^D X_t$ that are stationary in appearance. The behavior of the differenced (deseasonalized) series is described by two coupled ARMA models. The model orders P and Q should to be chosen so that the empirical autocorrelation function is consistent with that of an ARMA(P,Q) model for multiples of the period S. The orders p and q should be selected so that the empirical autocorrelation function within the period S shows the same behavior as the autocorrelation function of an ARMA(p,q) process. Finally, the model coefficients $(\phi_\ell)_{\ell=1}^p$, $(\Phi_\ell)_{\ell=1}^P$, $(\theta_\ell)_{\ell=1}^q$, $(\Theta_\ell)_{\ell=1}^Q$ and the white noise variance σ^2 can be estimated via parameter estimation procedures for ARMA processes. If the white noise process $\{Z_t\}$ is Gaussian, the most efficient estimates are produced by the maximum likelihood method. Since such estimates are found as optimal solutions to a highly nonlinear nonconvex optimization problem, good initial values for the model coefficients are needed. They can be obtained by the Hannan-Rissanen algorithm (cf. [5, §5]) that solves the problem of order selection and parameter estimation for ARMA processes simultaneously.

In our case, differencing the hourly load profile with lag 168 (one week) gave residuals that were stationary in appearance. The residuals were treated as part of a realization of the stochastic process $\{Y_t := \boldsymbol{d}_t - \boldsymbol{d}_{t-168}\}$. The Hannan-Rissanen algorithm from the *Mathematica Time Series Pack* [58] selected for $\{Y_t\}$ an ARMA(7,9) model that served as an initial model for the maximum likelihood method. For the resulting maximum likelihood estimates $(\hat{\phi}_\ell)_{\ell=1}^7$ and $(\hat{\theta}_\ell)_{\ell=1}^9$, the time series model for $\{Y_t\}$ reads

$$Y_t - \hat{\phi}_1 Y_{t-1} - \ldots - \hat{\phi}_7 Y_{t-7} = Z_t + \hat{\theta}_1 Z_{t-1} + \ldots + \hat{\theta}_9 Z_{t-9}, \quad t \in \mathbb{Z},$$

where the estimated model coefficients and random noise process are

$$\begin{aligned} (\hat{\phi}_1, \ldots, \hat{\phi}_7) &= (2.79, -4.35, 5.16, -4.88, 3.67, -1.92, 0.50), \\ (\hat{\theta}_1, \ldots, \hat{\theta}_9) &= (-1.27, 1.53, -1.35, 0.88, -0.31, -0.06, 0.18, 0.11, 0.07), \\ \{Z_t\} &\sim N(0, 11729.02), \quad t \in \mathbb{Z}. \end{aligned}$$

Accordingly, the time series model for the load process $\{\boldsymbol{d}_t\}_{t\in\mathbb{Z}}$ is the SARIMA$(7,0,9) \times (0,1,0)_{168}$ model

$$\boldsymbol{d}_t = \hat{\phi}_1 \boldsymbol{d}_{t-1} + \ldots + \hat{\phi}_7 \boldsymbol{d}_{t-7} - \boldsymbol{d}_{t-168} - \hat{\phi}_1 \boldsymbol{d}_{t-169} - \ldots - \hat{\phi}_7 \boldsymbol{d}_{t-175} \tag{4.2}$$

$$+ Z_t + \hat{\theta}_1 Z_{t-1} + \ldots + \hat{\theta}_9 Z_{t-9}, \quad t \in \mathbb{Z}.$$

Suppose there is a reliable load prediction $\{d_t\}_{t=1}^{t_1}$ for the first-stage (deterministic) time span $t = 1{:}\,t_1$, $t_1 < T$. A large number (M) of *simulated load scenarios* $\tilde{\boldsymbol{d}}^\ell = (\tilde{\boldsymbol{d}}_t^\ell)_{t=t_1+1}^T$, $\ell = 1{:}\,M$, may be generated using the SARIMA equation (4.2) with M i.i.d. realizations of $\{Z_t\}_{t=t_1-8}^T$, and starting values $\{d_t\}_{t=t_1-174}^{t_1}$ (supplied by the power utility). The *empirical means* $\bar{d}_t$ and *standard deviations* $\bar{\sigma}_t$ of the simulated load scenarios are defined by

$$(4.3) \quad \bar{d}_t = \frac{1}{M}\sum_{\ell=1}^M \tilde{\boldsymbol{d}}_t^\ell, \quad \bar{\sigma}_t^2 = \frac{1}{M-1}\sum_{\ell=1}^M (\tilde{\boldsymbol{d}}_t^\ell - \bar{d}_t)^2, \quad t = t_1 + 1{:}\,T.$$

4.2. The initial load scenario tree. An important initial decision is the choice of the number of stages and of the branching scheme for the scenario tree, i.e., the number and positions of branching levels and the branching degree in every node. We choose the following initial structure of the load scenario tree:

- A balanced tree with K branching periods t_k, $k = 1{:}\,K$. The branching periods t_k, $k = 2{:}\,K$, are equidistant within the time span $t = t_1{:}\,T$, i.e., $t_k := t_1 + (T - t_1)(k-1)/K$, $k = 2{:}\,K$.
- $|\mathcal{N}_+(n)| = \begin{cases} 2, & n \in \mathcal{N}_{t_k} = \{n : t(n) = t_k\},\ k = 1{:}\,K, \\ 1, & \text{otherwise.} \end{cases}$

Thus, the tree consists of $S := 2^K$ scenarios $d^s = (d_t^s)_{t=1}^T$, $s = 1{:}\,S$. The branching points t_k, $k = 2{:}\,K$, should correspond to the (normally fixed) times when already observable meteorological and load data provide the opportunity to re-adjust the unit commitment. For the planning horizon of one week with an hourly discretization, $t_k = 12 + 12k$ for $k = 1{:}\,12$ is a reasonable choice for the generation system of the utility VEAG. For longer scheduling periods, non-equidistant branching points would be preferable in order to restrict the number of scenarios. By assigning two successors to any node n in $\mathcal{N}_{t_k}$, $k = 1{:}\,K$, we may distinguish the events "low load" and "high load" for periods $t = t_k + 1{:}\,t_{k+1}$, where $t_{K+1} := T$. An additional event such as "medium load" could easily be included, but it would increase the scenario number to $S = 3^K$.

It remains to specify the scenario values and their probabilities. To this end, we first compute the empirical means $\{\bar{d}_t\}_{t=t_1+1}^T$ and the standard deviations $\{\bar{\sigma}_{t_k}\}_{k=2}^{K+1}$ (cf. (4.3)). The load predicted for the first-stage periods $t = 1{:}\,t_1$ yields the first t_1 components for all scenarios. (If no load prediction were available, one could use the empirical means.) To each scenario $s = 1{:}\,S$ we assign a vector $\omega^s = (\omega_k^s)_{k=1}^K$ with $\omega_k^s \in \{-1, 1\}$ that describes the path in the binary tree corresponding to scenario s. Specifically, we set $\omega_k^s = -1$ ($\omega_k^s = 1$) if the values of scenario s for $t = t_k + 1{:}\,t_{k+1}$ are realizations of the event "low load" ("high load"). The value of scenario

s for periods $t = t_1 + 1\colon T$ is defined as

$$d_t^s := \bar{d}_t + \sum_{i=1}^{k-1} \omega_i^s \frac{\bar{\sigma}_{t_{i+1}}}{2^{(K+1-i)/2}} + \omega_k^s \frac{\bar{\sigma}_{t_{k+1}}}{2^{(K+1-k)/2}} \frac{t-t_k}{t_{k+1}-t_k} \quad \text{for} \quad t = t_k + 1\colon t_{k+1},\ k = 1\colon K. \tag{4.4}$$

We let all scenarios have equal probabilities $S^{-1} = 2^{-K}$. (Alternative scenario probabilities might be computed from histograms of the simulated scenarios.)

A few comments on the tree construction formula (4.4) are in order. First, for $t = t_1 + 1\colon T$, the mean scenario value $\frac{1}{S}\sum_{s=1}^{S} d_t^s$ coincides with the empirical mean $\bar{d}_t$. Second, the symmetry of the load tree is consistent with the normality assumptions imposed on the time series model for the load process. Third, for $k = 1\colon K$, the events "low load" ("high load") for $t = t_k + 1\colon t_{k+1}$ are expressed in terms of scaled empirical standard deviations $\bar{\sigma}_{t_{k+1}}$. To model increasing load uncertainty, the variances $\operatorname{var}(d_t)$ of scenario values are strictly increasing with t. The extremal scenario s with $\omega_k^s = 1$ for all k has in the final period T the value

$$d_T^s = \bar{d}_T + 2^{-K/2}\bar{\sigma}_{t_2} + \cdots + 2^{-1/2}\bar{\sigma}_T.$$

Thus unrealistic ("too large") load values are avoided. Further,

$$\operatorname{var}(d_{t_{k+1}}) = 2^{-K}\bar{\sigma}_{t_2}^2 + \cdots + 2^{-(K+1-k)}\bar{\sigma}_{t_{k+1}}^2, \quad k = 1\colon K,$$

so for $\bar{\sigma}_{t_{k+1}} \approx \bar{\sigma}$, $k = 1\colon K$, we have $\operatorname{var}(d_T) \approx \bar{\sigma}^2(\frac{1}{2^K} + \cdots + \frac{1}{2}) \approx \bar{\sigma}^2$ (models with faster growth of variances are discussed in [15, 25]). Finally, we add that the scenario values between the t_k's are linearly interpolated so as to save work required for computing $\bar{\sigma}_t^2$ for all $t = t_1 + 1\colon T$.

Figure 7 shows ten scenarios (including the extremal paths corresponding to "low load" and "high load" for the time span $t = t_1 + 1 : T$) of a load scenario tree generated via the scheme (4.4) with $2^{12} = 4096$ scenarios for a planning horizon of one week with an hourly discretization and branching points $t_k = 12 + 12k$, $k = 1\colon 12$.

Calibration of scenario trees generated by the scheme (4.4) is studied in the forthcoming paper [25].

4.3. Optimal reduction of the scenario tree. As shown in §4.2, the probability distribution of the load may be approximated by a discrete probability distribution with a finite number of scenarios. Since the mixed-integer model (2.11)–(2.14) is large even for relatively few nodes, a compromise between acceptable computing times and the quality of the approximate scenario tree is unavoidable. Therefore, one often has to reduce the number of scenarios of the initial scenario tree.

Our reduction argument is based on certain probability metrics that measure the distance between the initial discrete approximation and the

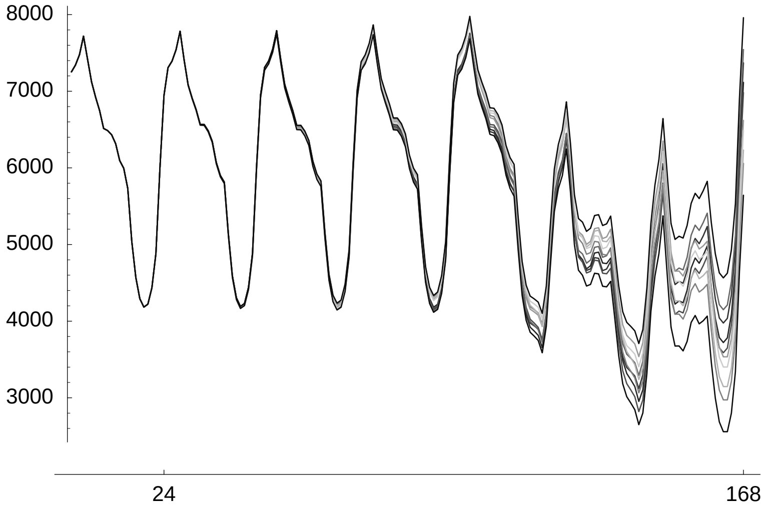

FIG. 7. *Ten selected scenarios of a load scenario tree for one week*

reduced one. Quantitative stability results for stochastic programs (cf. [16, 19, 47]) indicate which probability metric is canonically associated to a given model and/or to a specific type of approximation. In particular, the results in [15, 16, 47] suggest considering the Fortet-Mourier metrics ζ_h, $h \geq 1$, for a multistage stochastic program like (3.3).

For $h \geq 1$ we denote by $\mathcal{G}_h$ the class of functions $g : \mathbb{R}^T \to \mathbb{R}$ satisfying the Lipschitzian property

$$|g(\omega) - g(\omega')| \leq c_h(\omega, \omega') \quad \text{for all } \omega, \omega' \in \mathbb{R}^T,$$

where $c_h(\omega, \omega') := \max\{1, \|\omega\|^{h-1}, \|\omega'\|^{h-1}\}\|\omega - \omega'\|$ and $\|\cdot\|$ is the Euclidean norm on $\mathbb{R}^T$. Furthermore, we denote by $\mathcal{M}_h$ the set of all (Borel) probability measures μ such that $\int_{R^T} \|\omega\|^h \mu(d\omega) < \infty$. Then the *Fortet-Mourier metric* ζ_h of (Borel) probability measures $\mu, \nu \in \mathcal{M}_h$ is

$$\text{(4.5)} \quad \zeta_h(\mu, \nu) := \sup\left\{\Big|\int_{R^T} g(\omega)\mu(d\omega) - \int_{R^T} g(\omega)\nu(d\omega)\Big| : g \in \mathcal{G}_h\right\}.$$

For $h = 1$, ζ_1 is also known as the (L$_1$-) Wasserstein or Kantorovich metric. The metric ζ_h enjoys a well developed duality theory and convergence analysis (cf. [46, Chap. 5]).

Let δ_ω denote the probability measure on $\mathbb{R}^T$ having unit mass at $\omega \in \mathbb{R}^T$. Consider now two discrete probability measures

$$\mu := \sum_{s=1}^{S} \alpha_s \delta_{\omega_s} \quad \text{and} \quad \nu := \sum_{s=1}^{\tilde{S}} \tilde{\alpha}_s \delta_{\tilde{\omega}_s}$$

with supports $\{\omega_s\}_{s=1}^S$, $\{\tilde{\omega}_\sigma\}_{\sigma=1}^{\tilde{S}}$, and nonnegative weights α_s, $\tilde{\alpha}_\sigma$ such that $\sum_s \alpha_s = \sum_\sigma \tilde{\alpha}_\sigma = 1$. Then the dual transportation problem

$$\zeta_h(\mu, \nu) = \sup \left\{ \sum_{s=1}^S \alpha_s \xi_s + \sum_{\sigma=1}^{\tilde{S}} \tilde{\alpha}_\sigma \tilde{\xi}_\sigma : \xi_s + \tilde{\xi}_\sigma \le c_h(\omega_s, \tilde{\omega}_\sigma) \right\}$$

is the finite-dimensional analogue of (4.5). When the two measures μ and ν have the same support $\{\omega_s\}_{s=1}^S$, but different weights, upper and lower bounds for $\zeta_h(\mu, \nu)$ can be derived [15].

Now, let $\mu = \sum_{s=1}^S \alpha_s \delta_{\omega_s}$ be a discrete probability distribution on R^T that is regarded as a good initial approximation for the probability distribution entering a given stochastic program. For $\ell = 1{:}\,S$, let

$$\rho_\ell(\mu) := \min \left\{ \zeta_h \left(\mu, \sum_{s=1}^S \tilde{\alpha}_s \delta_{\omega_s} \right) : \tilde{\alpha}_s \ge 0, \sum_{s=1}^S \tilde{\alpha}_s = 1, \tilde{\alpha}_\ell = 0 \right\}.$$

Thus $\rho_\ell(\mu)$ is the distance of μ to a closest probability distribution having support $\{\omega_s : s = 1{:}\,S, s \ne \ell\}$, i.e., corresponding to deleting scenario ℓ of μ. Then we have (cf. [15])

$$\rho_\ell(\mu) \le \alpha_\ell \min_{s \ne \ell} c_h(\omega_\ell, \omega_s) \quad \text{for every } \ell \in \{1{:}\,S\}, \tag{4.6}$$

with the upper bounds attained if c_h satisfies the triangle inequality, i.e., for $h = 1$.

An *optimal rule* for deleting one scenario of μ may be stated as:

$$\text{Remove scenario } \omega_k \text{ with} \quad k \in \operatorname{Arg} \min_{\ell = 1:S} \rho_\ell(\mu).$$

Replacing $\rho_\ell(\mu)$ above by the upper bounds of (4.6) yields the more easily implementable deletion rule:

$$\text{Delete scenario } \omega_k \text{ with} \quad k \in \operatorname{Arg} \min_{\ell = 1:S} \left\{ \alpha_\ell \min_{s \ne \ell} c_h(\omega_\ell, \omega_s) \right\}.$$

Then, roughly speaking, deletion occurs where scenarios are close as measured by the distance c_h or where probabilities are small. The reduced discrete probability measure $\sum_{s=1, s \ne k}^S \beta_s \delta_{\omega_s}$ has $S - 1$ scenarios, where

$$\beta_{s_k} := \alpha_{s_k} + \alpha_k \quad \text{for some } s_k \in \operatorname{Arg} \min_{s \ne k} c_h(\omega_k, \omega_s),$$
$$\beta_s := \alpha_s \quad \text{for all } s \notin \{s_k, k\}.$$

This reduction procedure may be repeated until a prescribed number $\tilde{S}$ of scenarios in the reduced measure is attained.

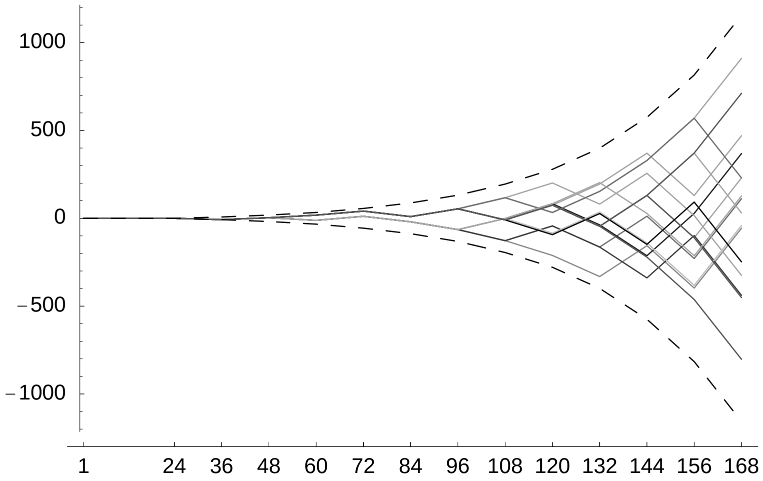

FIG. 8. *Shifted supports of the reduced scenario tree*

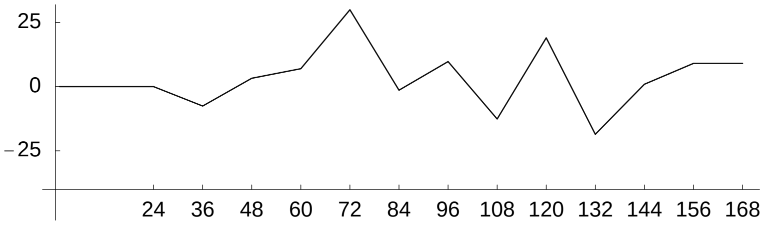

FIG. 9. *Difference of mean values of the reduced and initial scenario trees*

4.4. Example of scenario reduction. To test our approach, we generated a load scenario tree via the scheme (4.4) for an hourly discretized time horizon of one week ($T = 168$) with branching points $t_k = 12 + 12k$, $k = 1\colon 12$ (cf. Fig. 7). The initial number of scenarios $S = 4096$ was reduced to 16 by applying the scenario reduction rule of §4.3.

Figure 8 shows the position of the shifted supports $(d_t^s - \bar{d}_t)_{t=1}^{168}$, $s = 1\colon 16$, of the reduced scenario tree within the extremal paths of the initial scenario tree indicated by dashed lines, with grey levels proportional to scenario probabilities. The probabilities β_s, $s = 1\colon 16$, assigned to scenarios in the reduced tree vary between 0.04 and 0.11.

The reduction technique of §4.3 produces a discrete approximation whose moments differ in general from those of the initial approximation generated by (4.4). For example, Figure 9 shows the difference between the mean scenario value $\sum_{s=1}^{16} \beta_s d_t^s$ of the reduced tree and the empirical mean $\bar{d}_t$ of the simulation scenarios for $t = t_1 + 1\colon T$. Figure 10 compares the standard deviation at the branching periods t_k, $k = 1\colon 12$, for the initial approximation and the reduced scenario tree.

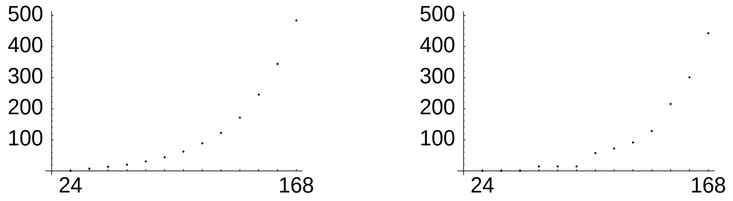

FIG. 10. *Standard deviations of the initial (left) and reduced (right) scenario trees*

Acknowledgment. The authors are grateful to Darinka Dentcheva (Lehigh University; formerly Humboldt-University Berlin), Svetlozar T. Rachev (University of California at Santa Barbara and University Karlsruhe) and Rüdiger Schultz (Gerhard-Mercator University Duisburg) for useful discussions on the material of this paper, and to G. Schwarzbach, J. Thomas and J. Krause (VEAG Vereinigte Energiewerke AG, Berlin) for providing us with the data used in our test runs. We extend our gratitude to an anonymous referee whose suggestions led to some improvements in presenting the material. This research was supported by the Schwerpunktprogramm *Echtzeit-Optimierung großer Systeme* of the Deutsche Forschungsgemeinschaft, the Alexander von Humboldt Foundation and the Graduiertenkolleg *Geometrie und Nichtlineare Analysis* at the Humboldt-University Berlin.

REFERENCES

[1] L. Bacaud, C. Lemaréchal, A. Renaud, and C. Sagastizábal, *Bundle methods in stochastic optimal power management: A disaggregated approach using preconditioners*, Comput. Optim. Appl. To appear.

[2] D. Bertsekas, G. S. Lauer, N. R. Sandell, and T. A. Posberg, *Optimal short-term scheduling of large-scale power systems*, IEEE Trans. Automat. Control, AC-28 (1983), pp. 1–11.

[3] J. R. Birge, *Stochastic programming computation and applications*, INFORMS J. Comput., 9 (1997), pp. 111–133.

[4] J. R. Birge and F. Louveaux, *Introduction to Stochastic Programming*, Springer, New York, 1997.

[5] P. J. Brockwell and R. A. Davies, *Introduction to Time Series and Forecasting*, Springer, New York, 1996.

[6] C. C. Carøe and R. Schultz, *A two-stage stochastic program for unit commitment under uncertainty in a hydro-thermal power system*, Preprint 98-13, Konrad-Zuse-Zentrum für Informationstechnik, Berlin, Germany, 1998.

[7] ———, *Dual decomposition in stochastic integer programming*, Oper. Res. Lett., 24 (1999), pp. 37–45.

[8] P. Carpentier, G. Cohen, and J.-C. Culioli, *Stochastic optimal control and decomposition-coordination methods, part II: Application*, in Recent Developments in Optimization, R. Durier and C. Michelot, eds., Lecture Notes in Economics and Mathematical Systems 429, Springer-Verlag, Berlin, 1995, pp. 88–103.

[9] P. Carpentier, G. Cohen, J.-C. Culioli, and A. Renaud, *Stochastic optimiza-*

tion of unit commitment: A new decomposition framework, IEEE Trans. Power Systems, 11 (1996), pp. 1067–1073.

[10] G. Consigli and M. A. H. Dempster, *Dynamic stochastic programming for asset-liability management*, Ann. Oper. Res., 81 (1998), pp. 131–161.

[11] D. Dentcheva and W. Römisch, *Optimal power generation under uncertainty via stochastic programming*, in Stochastic Programming Methods and Technical Applications, K. Marti and P. Kall, eds., Lecture Notes in Economics and Mathematical Systems 458, Springer-Verlag, Berlin, 1998, pp. 22–56.

[12] J. Dupačová, *Scenario-based stochastic programs: Resistance with respect to sample*, Ann. Oper. Res., 64 (1996), pp. 21–38.

[13] ———, *Portfolio optimization via stochastic programming: Methods of output analysis*, Math. Methods Oper. Res., 50 (1999), pp. 245–270.

[14] J. Dupačová, G. Consigli, and S. W. Wallace, *Scenarios for multistage stochastic programs*, Ann. Oper. Res. To appear.

[15] J. Dupačová, N. Gröwe-Kuska, and W. Römisch, *Scenario reduction in stochastic programming: An approach using probability metrics*, Preprint 00-09, Institut für Mathematik, Humboldt-Univ. Berlin, Berlin, Germany, 2000.

[16] J. Dupačová and W. Römisch, *Quantitative stability of scenario-based stochastic programs*, in Prague Stochastics '98, M. Hušková, P. Lachout, and J. Á. Višek, eds., JCMF, Prague, 1998, pp. 119–124.

[17] N. C. P. Edirisinghe, *Bound-based approximations in multistage stochastic programming: Nonanticipativity aggregation*, Ann. Oper. Res., 85 (1999), pp. 173–192.

[18] S. Feltenmark and K. C. Kiwiel, *Dual applications of proximal bundle methods, including Lagrangian relaxation of nonconvex problems*, SIAM J. Optim., 10 (2000), pp. 697–721.

[19] O. Fiedler and W. Römisch, *Stability in multistage stochastic programming*, Ann. Oper. Res., 56 (1995), pp. 79–93.

[20] S.-E. Fleten, S. W. Wallace, and W. T. Ziemba, *Portfolio management in a deregulated hydropower-based electricity market*, in Hydropower '97, E. Broch, D. K. Lysne, N. Flatabø, and E. Helland-Hansen, eds., Balkema, Rotterdam, 1997, pp. 197–204.

[21] K. Frauendorfer, *Barycentric scenario trees in convex multistage stochastic programming*, Math. Programming, 75 (1996), pp. 277–293.

[22] A. Gjelsvik, T. A. Røtting, and J. Røynstrand, *Long-term scheduling of hydro-thermal power systems*, in Hydropower '92, E. Broch and D. K. Lysne, eds., Balkema, Rotterdam, 1992, pp. 539–546.

[23] R. Gollmer, A. Möller, W. Römisch, R. Schultz, G. Schwarzbach, and J. Thomas, *Optimale Blockauswahl bei der Kraftwerkseinsatzplanung der VEAG*, in Optimierung in der Energieversorgung II, VDI-Berichte 1352, VDI, Düsseldorf, 1997, pp. 71–85.

[24] N. Gröwe, W. Römisch, and R. Schultz, *A simple recourse model for power dispatch under uncertain demand*, Ann. Oper. Res., 59 (1995), pp. 135–164.

[25] N. Gröwe-Kuska, M. P. Nowak, and I. Wegner, *Modelling of uncertainty for the real-time management of power systems*, tech. rep., Institut für Mathematik, Humboldt-Univ. Berlin, Berlin, Germany, 2001.

[26] J. L. Higle and S. Sen, *Stochastic Decomposition; A Statistical Method for Large Scale Stochastic Linear Programming*, Kluwer, Dordrecht, 1996.

[27] ———, *Statistical approximations for stochastic linear programming problems*, Ann. Oper. Res., 85 (1998), pp. 173–192.

[28] J.-B. Hiriart-Urruty and C. Lemaréchal, *Convex Analysis and Minimization Algorithms*, Springer, Berlin, 1993.

[29] K. Høyland and S. W. Wallace, *Generating scenario trees for multi-stage decision problems*, Management Sci. To appear.

[30] G. Infanger, *Monte Carlo (importance) sampling within a Bender's decomposition for stochastic linear programs*, Ann. Oper. Res., 39 (1992), pp. 69–95.

[31] J. Jacobs, G. Freeman, J. Grygier, D. Morton, G. Schultz, K. Staschus, and J. Stedinger, *SOCRATES: A system for scheduling hydroelectric generation under uncertainty*, Ann. Oper. Res., 59 (1995), pp. 99–133.

[32] K. C. Kiwiel, *Proximity control in bundle methods for convex nondifferentiable minimization*, Math. Programming, 46 (1990), pp. 105–122.

[33] ———, *Exact penalty functions in proximal bundle methods for constrained convex nondifferentiable minimization*, Math. Programming, 52 (1991), pp. 285–302.

[34] ———, *A Cholesky dual method for proximal piecewise linear programming*, Numer. Math., 68 (1994), pp. 325–340.

[35] ———, *User's guide for NOA 3.0: A Fortran package for convex nondifferentiable optimization*, tech. rep., Systems Research Institute, Warsaw, 1994.

[36] C. Lemaréchal, *Lagrangian decomposition and nonsmooth optimization: Bundle algorithm, prox iteration, augmented Lagrangian*, in Nonsmooth Optimization, Methods and Applications, F. Giannessi, ed., Gordon and Breach, Philadelphia, 1992, pp. 201–216.

[37] C. Lemaréchal and A. Renaud, *A geometric study of duality gaps, with applications*, Math. Programming. To appear.

[38] A. Løkketangen and D. L. Woodruff, *Progressive hedging and tabu search applied to mixed integer* $(0,1)$ *multi-stage stochastic programming*, J. Heuristics, 2 (1996), pp. 111–128.

[39] A. C. Miller and T. R. Rice, *Discrete approximations of probability distributions*, Management Sci., 29 (1983), pp. 352–362.

[40] V. I. Norkin, G. Ch. Pflug, and A. Ruszczyński, *A branch and bound method for stochastic global optimization*, Math. Programming, 83 (1998), pp. 425–450.

[41] M. P. Nowak, *A fast descent method for the hydro storage subproblem in power generation*, WP-96-109, International Institute for Applied Systems Analysis, Laxenburg, Austria, 1996.

[42] ———, *Stochastic Lagrangian Relaxation in Power Scheduling of a Hydro-Thermal System Under Uncertainty*, PhD thesis, Institut für Mathematik, Humboldt-Univ. Berlin, Berlin, Germany, 2000.

[43] M. P. Nowak and W. Römisch, *Stochastic Lagrangian relaxation applied to power scheduling in a hydro-thermal system under uncertainty*, Ann. Oper. Res. To appear.

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

[45] G. Ch. Pflug, *Scenario tree generation for multiperiod financial optimization by optimal discretization*, Math. Programming, 89 (2001), pp. 251–271.

[46] S. T. Rachev, *Probability Metrics and the Stability of Stochastic Models*, Wiley, Chichester, 1991.

[47] S. T. Rachev and W. Römisch, *Quantitative stability in stochastic programming: The method of probability metrics*, Preprint 00-22, Institut für Mathematik, Humboldt-Univ. Berlin, Berlin, Germany, 2000.

[48] A. Renaud, *Optimizing short-term operation of power generation: A new stochastic model*. Lecture presented at the VIII International Conference on Stochastic Programming, Vancouver (Canada), Aug. 1998.

[49] R. T. Rockafellar and R. J.-B. Wets, *The optimal recourse problem in discrete time: L^1-multipliers for inequality constraints*, SIAM J. Control Optim., 16 (1978), pp. 16–36.

[50] ———, *Scenarios and policy aggregation in optimization under uncertainty*, Math. Oper. Res., 16 (1991), pp. 119–147.

[51] W. Römisch and R. Schultz, *Decomposition of a multi-stage stochastic program for power dispatch*, Z. Angew. Math. Mech., 76 (1996), pp. 29–32. Suppl. 3.

[52] A. Ruszczyński, *Decomposition methods in stochastic programming*, Math. Programming, 79 (1997), pp. 333–353.

[53] G. B. Sheblé and G. N. Fahd, *Unit commitment literature synopsis*, IEEE Trans. Power Systems, 9 (1994), pp. 128–135.

[54] S. Takriti, J. R. Birge, and E. Long, *A stochastic model for the unit commitment problem*, IEEE Trans. Power Systems, 11 (1996), pp. 1497–1508.

[55] S. Takriti, B. Krasenbrink, and L. S.-Y. Wu, *Incorporating fuel constraints and electricity spot prices into the stochastic unit commitment problem*, Oper. Res., 48 (2000), pp. 268–280.

[56] P. P. J. van den Bosch and F. A. Lootsma, *Scheduling of power generation via large-scale nonlinear optimization*, J. Optim. Theory Appl., 55 (1987), pp. 313–326.

[57] R. J.-B. Wets, *Stochastic programming*, in Optimization, G. L. Nemhauser, A. H. G. Rinnooy-Kan, and M. J. Todd, eds., vol. 1 of Handbooks in Operations Research and Management Science, North-Holland, Amsterdam, 1989, pp. 573–629.

[58] Wolfram Research, *Mathematica Time Series Pack: Reference and User's Guide*, Champaign, IL, 1995.

[59] A. J. Wood and B. F. Wollenberg, *Power Generation, Operation, and Control*, Wiley, New York, second ed., 1996.

[60] S. A. Zenios and M. S. Shtilman, *Constructing optimal samples from a binomial lattice*, J. Inform. Optim. Sci., 14 (1993), pp. 125–147.

[61] G. Zhuang and F. D. Galiana, *Towards a more rigorous and practical unit commitment by Lagrangian relaxation*, IEEE Trans. Power Systems, 3 (1988), pp. 763–773.