Adaptive discretization of convex multistage stochastic programs

Stefan Vigerske* and Ivo Nowak†

May 9, 2006

Abstract

We propose a new scenario tree reduction algorithm for multistage stochastic programs, which integrates the reduction of a scenario tree into the solution process of the stochastic program. This allows to construct a scenario tree that is highly adapted on the optimization problem. The algorithm starts with a rough approximation of the original tree and locally refines this approximation as long as necessary. Promising numerical results for scenario tree reductions in the settings of portfolio management and power management with uncertain load are presented.

1 Introduction

Multistage stochastic programs (MSP) [21, 22] are a powerful tool to model a series of sequential optimization problems whose data underlies uncertainties. Therefore we consider a discrete-time stochastic data-process $\xi := \{\xi_t\}_{t=1}^T$ over some probability space $(\Omega, \mathcal{F}, \mathbb{P})$, $\xi_t : \Omega \to \mathbb{R}^{s_t}$, $s := \sum_{t=1}^T s_t$. Denote by $\zeta_t := (\xi_1, \ldots, \xi_t)$ the stochastic variable that represents the events that can be observed until time t, and by $\mathcal{F}_t := \sigma(\zeta_t)$ the σ-algebra generated by ζ_t. We assume $\mathcal{F}_1 = \{\emptyset, \Omega\}$, that is, the first-stage data is deterministic, and for convenience, $\mathcal{F}_T = \mathcal{F}$. The aim of the optimization is to find a decision process $x = \{x_t\}_{t=1}^T$ over $(\Omega, \mathcal{F}, \mathbb{P})$, $x_t : \Omega \to \mathbb{R}^{r_t}$, $\sum_{t=1}^T r_t = r$, which solves the multistage stochastic program

$$\begin{array}{lllll} \min\limits_x & \mathbb{E}\left[f(x,\xi)\right] & & & \\ \text{s.t.} & g_t(x_1,\ldots,x_t,\zeta_t) \le 0 & t=1,\ldots,T & \mathbb{P}\text{-a.s.} & \\ & x_t \in X_t & t=1,\ldots,T & \mathbb{P}\text{-a.s.} & \\ & x_t = \mathbb{E}\left[x_t \,|\mathcal{F}_t\right] & t=1,\ldots,T & & \end{array} \tag{MSP}$$

with $f : \mathbb{R}^r \times \mathbb{R}^s \to \mathbb{R}$, $g_t : \mathbb{R}^{\sum_{\tau=1}^t r_\tau} \times \mathbb{R}^{\sum_{\tau=1}^t s_\tau} \to \mathbb{R}^{d_t}$, and $X_t \subseteq \mathbb{R}^{r_t}$. The property $x_t = \mathbb{E}[x_t\,|\mathcal{F}_t]$ is called nonanticipativity and ensures that the decision x_t at time t depends on ζ_t only.

When the probability distribution has an infinite or very large support, the problem is numerically not tractable, so that the distribution is replaced by an approximation using a finite and smaller support Ω. Since the first-stage data is deterministic, the scenarios $\xi(\omega)$, $\omega \in \Omega$, can be represented in a scenario tree of depth T and with $|\Omega|$ leaves. The size of this scenario tree determines mainly the size of the optimization program. Hence, if it is too large, the scenario tree has to be reduced further, in order to make the MSP numerically tractable. This problem has attracted much interest over the last decades [1, 4, 5, 6, 7, 11, 12, 14, 15, 19]. In the following we mention some publications.

The algorithm of Casey and Sen for linear MSPs with random right-hand-side [1] constructs a sequence of scenario trees as an approximation of the original distribution. The optimal solution

*Humboldt-University Berlin, Institute for Mathematics, stefan@math.hu-berlin.de
†Niebuhrstraße 63, 10629 Berlin, Germany

for an approximated tree is prolongated to a policy for the original tree. The choice of the nodes for which the tree is refined is governed by two node-specific parameters. First, the probability of infeasibility of the prolongated policy, and second the gap between a lower and upper bound on a (nodal) salvage value function. The advantages of this algorithm are its asymptotic convergence and its measure of optimality and infeasibility. The computational challenge consists of the evaluation of bounds on the salvage value function and an upper bound on the infeasibility index in each node of the approximated scenario tree. Unfortunately, this algorithm seems to be limited to linear multistage programs with stochastic in the right-hand-side only, even though such problems are common in practical applications.

Another scenario tree generation algorithm which utilizes nodal information is the EVPI-based importance sampling algorithm from Dempster [4]. The expected value of perfect information (EVPI) process measures the impact of the nonanticipativity constraints on the optimal value of the MSP. For nodes of the scenario tree with a high EVPI process value the number of branches is increased, while for nodes with a small value the number of branches is decreased. The software package MSLiP [9] is an implementation of this algorithm for linear multistage stochastic programs.

Other approaches for solving MSPs are based on dual decomposition techniques. Edirisinghe's algorithm [7] for linear MSPs relaxes the nonanticipativity requirements of the decision-process up to a certain second-order aggregation and the computation of lower bounds. To form the aggregations, it is assumed that the support Ω of ξ can be enclosed in a compact simplex $\hat{\Omega}$, which is then partitioned into smaller subsimplices. Those which contribute most to the gap between a lower and an upper bound on the optimal value are then chosen for a refinement. The advantages of this approach are the computation of bounds on the value of the original problem, and the ability to work with continuous distributions. But currently it is unclear, whether this method provides any asymptotic guarantees.

Higle, Rayco, and Sen presented in [14] a stochastic scenario decomposition algorithm for MSPs. Here, the distribution is approximated by an empirical one obtained by scenario sampling, and dualization of the nonanticipativity constraints yields a dual problem, which is further approximated by a stochastic cutting plane method. To avoid a too extensive grow of the master problem due to the amount of sampled scenarios and generated cuts, columns referring to similar data in the problem are aggregated. Advantages of this approach are its mild assumptions on the stochastic program and the ability to work with a high number of scenarios. On the other hand, no estimate on the quality of a solution after a finite number of iterations is known, but almost sure asymptotical convergence can be shown.

Dupačová, Gröwe-Kuska, and Römisch have shown that the optimal value of a convex twostage stochastic program behaves (locally) Lipschitz-continuous with respect to the underlying probability distribution [6, 20]. Recently, Heitsch, Römisch, and Strugarek extended this result to linear multistage stochastic programs [13], where additionally the filtration $\{\mathcal{F}_t\}_{t=1}^T$ has to be taken into account. Hence, their scenario reduction algorithms aim to find a scenario tree whose associated probability distribution varies as less as possible from the original distribution with respect to the considered distances of probability measures and filtrations [11, 12]. We describe this approach more detailed in a later section, since we utilize it for our algorithm.

Pflug developed in [?] an algorithm related to methods from cluster analysis that considers the same distances of probability measures as in [6] to discretize a given probability distribution by a scenario tree of prescribed structure. For the case that the original distribution is not known, a stochastic approximation procedure, which is based on a sample from the original distribution, is proposed.

Also the approach of Høyland, Kaut, and Wallace for the discretization of a multivariate stochastic process ξ concentrates on properties of the probability distribution only, more precisely, the first four moments and the correlation matrix [15]. After generating s discrete independent univariate random variables, they are transformed to a multivariate random variable with given correlation matrix. The moments of the univariate variables are thereby chosen such that the moments of the multivariate variable match prescribed values. Since independency of the generated univariate variables is hard to achieve in general, an iterative algorithm is applied to obtain at least uncorrelated random variables.

In this work we develop an algorithm which integrates the construction of a reduced scenario tree into the solution process of a convex MSP. Hence, from now on we assume that the functions $f(x, \xi)$ and $g_t(x_1, \dots, x_t, \zeta_t)$ are convex in x and that X_t is a convex subset of $\mathbb{R}^{r_t}$. The convexity of the MSP might follow from the nature of the application, or might be given due to a relaxation of a nonconvex MSP. Indeed, we see a main application of our algorithm in the construction of scenario trees that are highly adapted to a convex relaxation of a nonconvex problem, and that can be used to discretize the nonconvex problem with a much smaller number of scenarios than by previous nonadaptive methods. We refer to the thesis of the second author [23] for such an extension. From a practical point of view, the assumption of convexity ensures that intermediate optimization problems remain numerically tractable.

The presented algorithm is based on the optimal scenario refinement algorithm of [17, chapter 9], which uses dual information to determine scenarios of an aggregated scenario tree, which have a high influence, and should therefore be refined. Similarly, out approach starts with a rough aggregation of a given scenario tree and refines this aggregation as long as the estimated gap to the original problem is too large. The gap is estimated by two error terms. To form the aggregations, the fast-forward selection algorithm of [11] is utilized.

The paper is organized as follows. Next, notations for scenario trees, aggregations of scenarios, and associated optimization problems are introduced. In the third section our approach to the adaptive discretization of scenario trees is described. The fourth section illustrates the algorithm on a product mix problem, a portfolio management problem, and the optimization of operation levels for a collection of coal fired thermal units and pumped hydro units under uncertain load. The paper finishes with concluding remarks.

2 Problem formulation

2.1 Scenario tree formulation

We let $\Omega = \{\omega_i\}_{i \in I}$ with finite index set I, and represent the information structure defined by the scenarios $\xi(\omega_i)$, $i \in I$, in a scenario tree: Nodes at depth t correspond to elements of a finite generator of $\mathcal{F}_t$ and two nodes have the same predecessor when their history is "the same", i.e., indistinguishable in terms of the filtration $\mathcal{F}_{t-1}$. We denote the nodes of the tree by $\bar{N}$ and introduce the following notation: For $n \in \bar{N}$ and $N \subseteq \bar{N}$ let

- n_- be the predecessor of n, and $n_{-k} := \left(n_{-(k-1)}\right)_-$ with $n_{-1} := n_-$
- $t(n)$ such that $n_{-t(n)+1}$ is the root node (the depth or timestage of n)
- $N_t := \{n \in N \,|\, t(n) = t\}$ all nodes with depth (or timestage) t in N
- $p(n) := \left(n_{-t(n)+1}, \dots, n_-, n\right)$ the path from the root node to node n
- $i(n) \in I$ the scenario index that belongs to $n \in \bar{N}_T$.

We define recursively node probabilities π_n: For $n \in \bar{N}_T$, $\pi_n := \mathbb{P}\left[\left\{\omega_{i(n)}\right\}\right]$, and for $n \in \bar{N} \setminus \bar{N}_T$, $\pi_n := \sum_{m \in \bar{N}: m_- = n} \pi_m$. Further, let $\xi^n := \xi_{t(n)}(\omega_i)$, and $\zeta^n := \xi^{p(n)} := (\dots, \xi^{n_-}, \xi^n)$. Then the scenario tree formulation of (MSP) reads

$$\begin{aligned} \min_x \quad & \sum_{n \in \bar{N}_T} \pi_n f\left(x^{p(n)}, \zeta^n\right) & \\ \text{s.t.} \quad & g_{t(n)}\left(x^{p(n)}, \zeta^n\right) \leq 0 & n \in \bar{N} \\ & x^n \in X_{t(n)} & n \in \bar{N} \end{aligned} \tag{MSP $[\bar{N}]$}$$

where $x = \{x^n\}_{n \in \bar{N}}$. (MSP $[\bar{N}]$) is a convex problem of size proportional to $|\bar{N}|$, i.e., the number of variables is $\sum_{t=1}^T |\bar{N}_t| r_t$ and the number of constraints is $\sum_{t=1}^T |\bar{N}_t| d_t$ plus the number of equations to describe the constraints "$x^n \in X_{t(n)}$" for each $n \in \bar{N}$. We denote the optimal value of (MSP $\left[\bar{N}\right]$) by $v\left(\bar{N}\right)$.

2.2 The approximation problem

In the adaptive discretization algorithm (next section), we make extensive use of aggregations of scenarios and corresponding scenario trees. We identify the scenarios $\xi(\omega_i)$, $i \in I$, with the leaves ξ^n, $n \in \bar{N}_T$, in the scenario tree, and obtain an aggregation $\mathcal{N}$ of $\bar{N}$ by partitioning $\bar{N}_T$ and selecting so-called representative nodes for each element $N \in \mathcal{N}$. To be more precise, let $\mathcal{N}$ be a partition of $\bar{N}_T$ and for $N \in \mathcal{N}$ (i.e., $N \subseteq \bar{N}_T$) let $n_T(N) \in N$ be a *representative* node for N. We let $n_t(N) := n_T(N)_{-T+t}$, $p(N) := p(n_T(N)) = (n_1(N), \ldots, n_T(N))$, $N \in \mathcal{N}$, and $\bar{N}|_{\mathcal{N}}$ be the scenario tree that is induced by the paths $p(N)$, $N \in \mathcal{N}$. The node probabilities for the leaves of $\bar{N}|_{\mathcal{N}}$ are $\pi_N := \sum_{n \in N} \pi_n$, $N \in \mathcal{N}$. We define the aggregated problem of (MSP $\left[\bar{N}\right]$) by restricting the decision-process and constraints to the aggregated scenario tree $\bar{N}|_{\mathcal{N}}$. For the objective function $F_{\mathcal{N}}(x, \xi)$, we average over aggregated scenarios, which will simplify the comparison with the objective function of a refinement of $\mathcal{N}$,

$$F_{\mathcal{N}}(x, \xi) := \sum_{N \in \mathcal{N}} \sum_{n \in N} \pi_n f\left(x^{p(N)}, \zeta^n\right).$$

Thus, the aggregated problem reads:

$$\begin{array}{lll} \min\limits_{x} & F_{\mathcal{N}}(x, \xi) & \\ \text{s.t.} & g_{t(n)}\left(x^{p(n)}, \zeta^n\right) \le 0 & n \in \bar{N}|_{\mathcal{N}} \\ & x^n \in X_{t(n)} & n \in \bar{N}|_{\mathcal{N}} \end{array} \qquad (\text{MSP}[\mathcal{N}])$$

We denote its optimal value by $v(\mathcal{N})$.

The aim of the adaptive scenario reduction algorithm is to find an aggregation $\mathcal{N}$ such that the optimal value of the aggregated problem (MSP $[\mathcal{N}]$) does not differ much from the optimal value of the original problem, $v(\mathcal{N}) \approx v\left(\bar{N}\right)$, and the number $|\mathcal{N}|$ of scenarios in the aggregated scenario tree is small.

We use a refinement of an aggregation $\mathcal{N}$ as a test-aggregation to find out which parts of $\mathcal{N}$ are aggregated too roughly, and have to be refined. Under a refinement of an aggregation $\mathcal{N}$ we understand another partition $\mathcal{N}'$ of $\bar{N}_T$ that is finer than $\mathcal{N}$, i.e., $\forall N' \in \mathcal{N}'\ \exists N \in \mathcal{N}$: $N' \subseteq N$. We require further that the representative nodes in $\mathcal{N}'$ are chosen such that $\bar{N}|_{\mathcal{N}} \subseteq \bar{N}|_{\mathcal{N}'}$. As last notation, for $N \in \mathcal{N}$ let $\mathcal{N}'|_N := \{N' \in \mathcal{N}' \,|\, N' \subseteq N\}$ be the sets N' in the refinement $\mathcal{N}'$ which union forms the set N.

3 Adaptive scenario reduction

The adaptive scenario reduction algorithm computes an aggregation $\mathcal{N}$ of the original scenario tree $\bar{N}$ such that the distance between the optimal values of the aggregated problem (MSP $[\mathcal{N}]$) and the original problem (MSP $\left[\bar{N}\right]$) is small. The main idea is the following. We estimate the approximation error by

$$\left|v(\mathcal{N}) - v\left(\bar{N}\right)\right| \le \underbrace{\left|v(\mathcal{N}) - v(\mathcal{N}')\right|}_{=:\Delta(\mathcal{N}, \mathcal{N}')} + \underbrace{\left|v(\mathcal{N}') - v\left(\bar{N}\right)\right|}_{=:\Delta\left(\mathcal{N}', \bar{N}\right)}, \qquad (3.1)$$

whereby $\mathcal{N}'$ is a refinement of $\mathcal{N}$. For the reduction of the former gap, $\Delta(\mathcal{N}, \mathcal{N}')$, we identify explicitely sets of aggregated scenarios $N \in \mathcal{N}$ which have a contribution to this gap. They are then replaced by a partition of N, based on $\mathcal{N}'|_N$. For the latter gap, $\Delta\left(\mathcal{N}', \bar{N}\right)$, we reduce the distance between the probability distributions associated with the aggregated tree $\bar{N}|_{\mathcal{N}'}$ and the original tree $\bar{N}$.

3.1 Reduction of the error $\Delta(\mathcal{N}, \mathcal{N}')$

Assume that the optimal values of (MSP $[\mathcal{N}]$) and (MSP $[\mathcal{N}']$) are known. Let σ_P be a (relative) tolerance on the gap $\Delta(\mathcal{N}, \mathcal{N}')$. If the gap exceeds the tolerance, we assemble a collection $\mathcal{M} \subseteq \mathcal{N}$ of at most $\bar{m}$ node sets which contribute most to this gap and replace each $N \in \mathcal{M}$ by an aggregation of $\mathcal{N}'|_N$, so that a refinement of N in $\mathcal{N}$ is achieved. Finally, to ensure that $\mathcal{N}'$ remains a nontrivial refinement of $\mathcal{N}$, sets $N' \in \mathcal{N} \cap \mathcal{N}'$, i.e., sets in the aggregation $\mathcal{N}$ that are associated with a trivial refinement in $\mathcal{N}'$, are refined in $\mathcal{N}'$ (if possible).

The parameter $\bar{m}$ determines the growth rate of the aggregated tree $\bar{N}|_{\mathcal{N}}$. If it is chosen too large, many $N \in \mathcal{N}$ are refined in one iteration, so that $\bar{N}|_{\mathcal{N}}$ grows fast and unnecessary many scenarios might be added. If $\bar{m}$ is small, many iterations might be necessary for a sufficient reduction of the gap $\Delta(\mathcal{N}', \bar{N})$.

We distinguish two cases for the identification of $\mathcal{M}$.

3.1.1 The case $v(\mathcal{N}) - v(\mathcal{N}') > \sigma_P\, |v(\mathcal{N}')|$

If $v(\mathcal{N})$ exceeds $v(\mathcal{N}')$ too much, we have to search for sets $N \in \mathcal{N}$ whose refinement allows a high diminishment in $v(\mathcal{N})$. Actually, we aim to reach $v(\mathcal{N}) - v(\mathcal{N}') \le \sigma_P\, |v(\mathcal{N})|$. Due to convexity, this relation is satisfied by a point that is feasible for (MSP $[\mathcal{N}]$) and stays close to the set of optimal points of (MSP $[\mathcal{N}']$). To find the sets $N \in \mathcal{N}$ which hamper this goal we compute a point $x = (x_{N'})_{N' \in \mathcal{N}'}$ that is feasible for (MSP $[\mathcal{N}']$), its objective value does not exceed $v(\mathcal{N}')$ by more than the fraction σ_P, and that is as "close" to the rougher aggregation $\mathcal{N}$ as possible. Then we select for $\mathcal{M}$ those $N \in \mathcal{N}$ where the variation of $x_{N'}$ over $\mathcal{N}'|_N$ is the highest.

More formal, we search for a point x feasible for (MSP $[\mathcal{N}']$) with $F_{\mathcal{N}'}(x, \xi) \le v(\mathcal{N}') + \sigma_P\, |v(\mathcal{N}')|$ that minimizes the variation of $x_{N'}$ from x_N for $N' \in \mathcal{N}'|_N$ and $N \in \mathcal{N}$. That is, we solve the convex problem

$$\begin{aligned}
\min_{x} \quad & \sum_{N \in \mathcal{N}} \pi_N \max_{t=1,\dots,T} \max_{N' \in \mathcal{N}'|_N} \left\| x^{n_t(N')} - x^{n_t(N)} \right\|_\infty & & \\
\text{s.t.} \quad & g_{t(n)}\left(x^{p(n)}, \zeta^n\right) \le 0 & n \in \bar{N}|_{\mathcal{N}'} & \\
& x^n \in X_{t(n)} & n \in \bar{N}|_{\mathcal{N}'} & \\
& F_{\mathcal{N}'}(x) \le v(\mathcal{N}') + \sigma_P\, |v(\mathcal{N}')| & &
\end{aligned} \tag{P}$$

Next, we select those $N \in \mathcal{N}$ with the highest impact on optimal value of (P), i.e., for an optimal point $\bar{x}$ of (P) let

$$\mathcal{M} := \operatorname{argmax} \left\{ \sum_{M \in \mathcal{M}} \pi_M \max_{t=1,\dots,T} \max_{M' \in \mathcal{N}'|_M} \left\| \bar{x}^{n_t(M')} - \bar{x}^{n_t(M)} \right\|_\infty \,\middle|\, \mathcal{M} \subseteq \mathcal{N},\ |\mathcal{M}| \le \bar{m} \right\}, \tag{3.2}$$

and such that for each $M \in \mathcal{M}$ the difference $\max_{t=1,\dots,T} \max_{M' \in \mathcal{N}'|_M} \left\| \bar{x}^{n_t(M')} - \bar{x}^{n_t(M)} \right\|_\infty$ is strictly positive.

$\mathcal{M}$ is not empty, since otherwise the optimal value of (P) would be zero. Then the solution point $\bar{x}$ of (P) could be reformulated to a feasible point $\tilde{x}$ for (MSP $[\mathcal{N}]$) with $F_{\mathcal{N}}(\tilde{x}) \le v(\mathcal{N}') + \sigma_P\, |v(\mathcal{N}')|$, which violates the assumption $v(\mathcal{N}) - v(\mathcal{N}') > \sigma_P\, |v(\mathcal{N}')|$.

3.1.2 The case $v(\mathcal{N}') - v(\mathcal{N}) > \sigma_P\, |v(\mathcal{N}')|$

Here, we cannot proceed as in the first case. Instead, we concentrate our attention directly to the objective functions of (MSP $[\mathcal{N}]$) and (MSP $[\mathcal{N}']$), and select those $N \in \mathcal{N}$ that contribute most to the difference of the objective functions evaluated at solution points of (MSP $[\mathcal{N}]$) and (MSP $[\mathcal{N}']$). That is, for $\bar{x} = (\bar{x}_N)_{N \in \mathcal{N}}$ optimal for (MSP $[\mathcal{N}]$) and $\bar{x}' = (\bar{x}'_{N'})_{N' \in \mathcal{N}'}$ optimal for (MSP $[\mathcal{N}']$) it is

$$F_{\mathcal{N}'}(\bar{x}') - F_{\mathcal{N}}(\bar{x}) = \sum_{N \in \mathcal{N}} \sum_{N' \in \mathcal{N}'|_N} \sum_{n \in N'} \pi_n \left(f\left(\bar{x}'^{p(N')}, \zeta^n\right) - f\left(\bar{x}^{p(N)}, \zeta^n\right) \right) > 0,$$

and we select $\mathcal{M}$ by

$$\mathcal{M} := \operatorname{argmax}\left\{ \sum_{M\in\mathcal{M}} \sum_{M'\in\mathcal{N}'|_M} \sum_{n\in M'} \pi_n \left(f\left(\bar{x}'^{p(M')}, \zeta^n\right) - f\left(\bar{x}^{p(M)}, \zeta^n\right)\right) \middle| \begin{array}{c} \mathcal{M}\subseteq\mathcal{N}, \\ |\mathcal{M}|\le\bar{m} \end{array} \right\}, \quad (3.3)$$

and such that for each $M \in \mathcal{M}$ the difference $\sum_{M'\in\mathcal{N}'|_M} \sum_{n\in M'} \pi_n \left(f\left(\bar{x}'^{p(M')}, \zeta^n\right) - f\left(\bar{x}^{p(M)}, \zeta^n\right)\right)$ is strictly positive.

3.2 Reduction of the error $\Delta\left(\mathcal{N}', \bar{N}\right)$

When the first error is small enough, we have to ensure that the approximation (MSP $[\mathcal{N}']$) gives a meaningful estimate for the optimal value of the original problem (MSP $[\bar{N}]$). For convex twostage ($T = 2$) stochastic programs, Dupačová, Gröwe-Kuska, and Römisch have shown that the gap $\Delta\left(\mathcal{N}', \bar{N}\right)$ can be bounded by a distance of the probability distributions associated with the trees $\bar{N}\,|_{\mathcal{N}'}$ and $\bar{N}$ [6]. For multistage stochastic programs, further the distance of the filtrations associated with the trees has to be taken into account [13].

In terms of the stochastic program (MSP) we summarize the main results as follows, cf. [6, 12, 13, 20]: Assume that the functions $f(x,\xi)$ and $g_t(x_1,\dots,x_t,\zeta_t)$ are affine linear in x and ξ with non-stochastic recourse matrices $A_{t,t}$, i.e., $f(x,\xi) = \sum_{t=1}^{T} \langle c_t(\zeta_t), x_t\rangle$ and $g_t(x_1,\dots,x_t,\zeta_t) = \sum_{\tau=1}^{t-1} A_{t,\tau}(\zeta_t)\, x_\tau + A_{t,t}x_t + b_t(\zeta_t)$ for suited vector- and matrix-functions c_t, b_t, and $A_{t,\tau}$, $\tau = 1,\dots,t-1$, $t = 1,\dots,T$, and assume X_t to be polyhedral for $t = 1,\dots,T$. Further on, we denote by $F(x,\xi) := \mathbb{E}[f(x,\xi)]$ the objective function of (MSP), by $\mathcal{X}_t(x_1,\dots,x_{t-1},\zeta_t) := \{x_t \in X_t\,|g_t(x_1,\dots,x_t,\zeta_t) \le 0\}$ the feasible set at timestage t, $t = 2,\dots,T$, and by

$$\mathcal{X}(\xi) := \left\{ x \in \bigtimes_{t=1}^{T} L_\infty(\Omega,\mathcal{F}_t,\mathbb{P};\mathbb{R}^r) \middle| x_1 \in X_1, x_t \in \mathcal{X}_t(x_1,\dots,x_{t-1},\zeta_t)\right\}$$

the feasible set of (MSP), where $\|x\|_\infty := \max_t \operatorname{ess\,sup}_\Omega \|x_t\|$. Hence, we can write (MSP) as $\min\{F(x,\xi)\,|x\in\mathcal{X}(\xi)\}$. We denote the optimal value of (MSP) by $v(\xi)$, and let, for any $\alpha \ge 0$,

$$l_\alpha(F(\cdot,\xi)) := \{x\in\mathcal{X}(\xi)\,|F(x,\xi)\le v(\xi)+\alpha\}$$

denote the α-level set of (MSP). Then the following statement about the stability of (MSP) with respect to the stochastic process ξ has been shown [12, 13]:

Theorem. *Let $\xi \in L_q(\Omega,\mathcal{F},\mathbb{P};\mathbb{R}^s)$ for some $q \ge 1$, where $\|\xi\|_q^q := \sum_{t=1}^{T}\mathbb{E}[\|\xi_t\|^q]$. Assume that there exists a $\delta > 0$ such that for $\tilde{\xi} \in L_q(\Omega,\mathcal{F},\mathbb{P};\mathbb{R}^s)$ with $\|\xi-\tilde{\xi}\|_q \le \delta$, the problem $\min\{F(x,\tilde{\xi})|x\in\mathcal{X}(\tilde{\xi})\}$ possesses relatively complete recourse. Assume further that the optimal value $v(\xi)$ is finite and there exists a constant $\alpha > 0$ such that the objective function F is level-bounded locally uniformly at ξ. Finally, assume that X_1 is bounded.*

Then there exists a positive constant L, such that the estimate

$$\left|v(\xi) - v(\tilde{\xi})\right| \le L\left(\|\xi-\tilde{\xi}\|_q + D_f(\xi,\tilde{\xi})\right) \quad (3.4)$$

holds for all random elements $\tilde{\xi}\in L_q(\Omega,\mathcal{F},\mathbb{P};\mathbb{R}^s)$ with $\|\xi-\tilde{\xi}\|_q\le\delta$ and $v(\tilde{\xi})$ finite. Here, $D_f(\xi,\tilde{\xi})$ denotes the filtration distance of ξ and $\tilde{\xi}$ defined by

$$D_f(\xi,\tilde{\xi}) := \sup_{\varepsilon\in(0,\alpha]} \inf_{\substack{x\in l_\varepsilon(F(\cdot,\xi)) \\ \tilde{x}\in l_\varepsilon(F(.,\tilde{\xi}))}} \sum_{t=2}^{T-1} \max\left\{\|x_t - \mathbb{E}[x_t|\,\tilde{\mathcal{F}}_t]\|_\infty, \|\tilde{x}_t - \mathbb{E}[\tilde{x}_t\,|\mathcal{F}_t]\|_\infty\right\},$$

where $\mathcal{F}_t$ and $\tilde{\mathcal{F}}_t$ denote the σ-algebras generated by ζ_t and $\tilde{\zeta}_t$, respectively.

Observe, that for twostage stochastic programs the filtration distance $D_f(\xi,\tilde{\xi})$ vanishes in (3.4). Starting from an estimate similar to (3.4), Dupačová et al. derived scenario reduction algorithms for convex twostage stochastic programs with finitely supported stochastic process ξ, which compute approximations $\tilde{\xi}$ on a subset of scenarios that minimize the distance $\|\xi-\tilde{\xi}\|_q$ [6, 12]. That is, for a finite set $\Omega=\{\omega_i\}_{i\in I}$ and a given stochastic process ξ on Ω (thus, having scenarios $\xi^i := \xi(\omega_i)$, $i\in I$), the principle of optimal scenario reduction can be written as: Determine an index set $J\subset I$ of a given cardinality k such that the process $\tilde{\xi}:\Omega\to\mathbb{R}^s$ with

$$\tilde{\xi}(\omega_i) := \begin{cases} \xi(\omega_i), & i\notin J, \\ \xi\left(\omega_{i(j)}\right), & j\in J, \end{cases} \tag{3.5}$$

minimizes $\|\xi-\tilde{\xi}\|_q$, where $i(j) := \operatorname{argmin}_{i\notin J}\left\|\xi^i-\xi^j\right\|$, $j\in I$ (*optimal redistribution rule*). Combining (3.5) and the definition of $\|\cdot\|_q$, we have

$$\left\|\xi-\tilde{\xi}\right\|_q^q = \sum_{j\in J} p_j \left\|\xi^j-\xi^{i(j)}\right\|^q = \sum_{j\in J} p_j \min_{i\notin J}\left\|\xi^j-\xi^i\right\|^q, \tag{3.6}$$

where $p_i := \mathbb{P}[\{\omega_i\}]$, $i\in I$. Hence, the selection of the set $J\subset I$ gives rise to the combinatorial optimization problem

$$\min_{\substack{J\subset I\\ |J|=k}} \sum_{j\in J} p_j \min_{i\notin J}\left\|\xi^i-\xi^j\right\|^q .$$

One heuristic to attack this problem is the (fast-)forward selection algorithm [6, 11] which chooses the set $I\setminus J$ one by one by selecting scenarios that achieve the best improvement in $\|\xi-\tilde{\xi}\|_q$: The set $I\setminus J := \left\{u_1,\dots,u_{|I|-k}\right\}$ is computed by

$$u_j := \operatorname*{argmin}_{u\in I\setminus\{u_1,\dots,u_{j-1}\}} \sum_{i\in I\setminus\{u_1,\dots,u_{j-1},u\}} p_i \min_{l\in\{u_1,\dots,u_{j-1},u\}} \left\|\xi^l-\xi^i\right\|^q, \quad j=1,\dots,|I|-k. \tag{FF}$$

If the stochastic program has only two stages, inequality (3.4) bounds the distance between the optimal values of the original problem and the problem with the reduced tree,

$$\left|v(\xi)-v(\tilde{\xi})\right| \le L\|\xi-\tilde{\xi}\|_q = L\left(\sum_{j\in J} p_j \min_{i\notin J}\left\|\xi^i-\xi^j\right\|^q\right)^{1/q} .$$

For multistage stochastic programs, a stagewise application of the fast-forward selection algorithm (FF) yields an algorithm for the reduction of a scenario tree that controls the distance $\|\xi-\tilde{\xi}\|_q$ [11]. But since it does not take care of the filtration distance $D_f(\xi,\tilde{\xi})$, it can be seen as a heuristic only. For the special case that ξ is a fan of scenarios, it is shown in [12] how the fast-forward selection algorithm can be adjusted to control the filtration distance too.

Measuring the distance of the original tree $\bar{N}$ to the aggregated tree $\bar{N}|_{\mathcal{N}'}$ by $\|\xi-\tilde{\xi}\|_q$ (with ξ the process associated with $\bar{N}$ and $\tilde{\xi}$ the process associated with $\bar{N}|_{\mathcal{N}'}$), we can write the distance (3.6) as

$$\mathcal{D}(\mathcal{N}') := \sum_{N'\in\mathcal{N}'} \mathcal{D}_{\mathcal{N}'}(N') \quad \text{with} \quad \mathcal{D}_{\mathcal{N}'}(N') := \sum_{n\in N':n\neq n_T(N')} p_{i(n)} \min_{N'\in\mathcal{N}'} \left\|\xi^{i\left(n_T(N')\right)}-\xi^{i(n)}\right\|.$$

For the case of a linear twostage problem, this definition results in $\Delta\left(\mathcal{N}',\bar{N}\right)\le L\,\mathcal{D}(\mathcal{N}')$ due to estimate (3.4).

To reduce $\mathcal{D}(\mathcal{N}')$, we select $N'\in\mathcal{N}'$ that maximizes $\mathcal{D}_{\mathcal{N}'}(N')$ and apply the fast-forward selection algorithm (FF) for N', i.e., we select nodes from N' other than the representative node $n_T(N')$ and add the corresponding paths to the refined tree $\bar{N}|_{\mathcal{N}'}$ (see [23] for details). This procedure is repeated until we reduced the distance $\mathcal{D}(\mathcal{N}')$ by at least 10 percent.

Since the distance of ξ and $\tilde{\xi}$ in (3.4) is multiplied by a problem-dependent factor L, it is even in the twostage case difficult to quantify the required accuracy of the approximation $\tilde{\xi}$ necessary to cause only a small perturbation of the optimal value. We proceed with the refinement of $\mathcal{N}'$ as long as the estimate $\mathcal{D}(\mathcal{N}')$ exceeds a given fraction on the gap between the tree consisting of only one scenario and the original tree, i.e., as long as

$$\mathcal{D}(\mathcal{N}') > \sigma_D \mathcal{D}_0,$$

where $\sigma_D \in [0,1]$ is a given parameter and $\mathcal{D}_0 := \left\|\xi - \xi^{i(\bar{n})}\right\|_q$ for a scenario $i(\bar{n})$ chosen by (FF) from the set of all scenarios. Note, that a very small choice for the tolerance σ_D on the distance between $\bar{N}|_{\mathcal{N}'}$ and $\bar{N}$ might indeed result in a very fine aggregation $\mathcal{N}'$, but has no direct impact on the final aggregation $\mathcal{N}$. This is the main difference to the scenario reduction algorithms in [6, 11, 12], where a too small choice for the tolerance on $\|\xi - \tilde{\xi}\|_q$ unaware of the problem-dependent parameter L can lead to unnecessary many scenarios and nodes in the reduced tree.

3.3 The algorithm

We can now formulate the adaptive scenario reduction algorithm, see Alg. 1. For the initial aggregation $\mathcal{N}$, we select one scenario $\xi^{i(\bar{n})}$ by fast-forward selection (FF) and set $\mathcal{N} := \left\{\bar{N}_T\right\}$ with $n_T\left(\bar{N}_T\right) := \bar{n}$. For the initial refinement $\mathcal{N}'$ of $\mathcal{N}$ we select some additional scenarios from $\bar{N}_T$ by fast-forward selection (FF).

Algorithm 1 adaptive scenario reduction

Input: initial tree $\bar{N}$, parameters $\bar{m} \in \mathbb{N}$ and $\sigma_P \in [0,1]$, $\sigma_D \in [0,1]$
Compute:
compute initial rough aggregation $\mathcal{N} := \left\{\bar{N}_T\right\}$ with $n_T\left(\bar{N}_T\right) := \bar{n}$ and $\bar{n}$ selected by (FF)
$\mathcal{D}_0 := \left\|\xi - \xi^{i(\bar{n})}\right\|_q$
compute initial refinement $\mathcal{N}'$ of $\bar{N}$
loop
  solve (MSP $[\mathcal{N}]$) and (MSP $[\mathcal{N}']$)
  if $\Delta(\mathcal{N},\mathcal{N}') > \sigma_P\, v(\mathcal{N}')$ **then**
    $\mathcal{M} := \emptyset$
    if $v(\mathcal{N}) - v(\mathcal{N}') > \sigma_P\, |v(\mathcal{N}')|$ **then**
      solve (P)
      compute $\mathcal{M}$ according to formula (3.2)
    else if $v(\mathcal{N}') - v(\mathcal{N}) > \sigma_P\, |v(\mathcal{N}')|$ **then**
      compute $\mathcal{M}$ according to formula (3.3)
    end if
    replace $M \in \mathcal{M}$ by an aggregation of $\mathcal{N}'|_M$, using fast-forward selection (FF)
    replace (in $\mathcal{N}'$) sets $N' \in \mathcal{N} \cap \mathcal{N}'$ by a refinement of N', using fast-forward selection (FF)
  else if $\mathcal{D}(\mathcal{N}') > \sigma_D\, \mathcal{D}_0$ **then**
    let $\mathcal{D}_1 := \mathcal{D}(\mathcal{N}')$
    while $\mathcal{D}(\mathcal{N}') \geq \frac{9}{10}\mathcal{D}_1$ **do**
      replace $N' := \operatorname{argmax}\{\mathcal{D}_{\mathcal{N}'}(N') \,|\, N' \in \mathcal{N}'\}$ by a refinement of N', using fast-forward selection (FF)
    end while
  else
    break
  end if
end loop
Output: the aggregated node set $\mathcal{N}$

Proposition. *Algorithm 1 computes a reduced scenario tree $\mathcal{N}$ with $\Delta(\mathcal{N},\mathcal{N}') \leq \sigma_P\, |v(\mathcal{N}')|$ and $\mathcal{D}(\mathcal{N}') \leq \sigma_D \mathcal{D}_0$ in a finite number of steps.*

Proof. The algorithm stops when no sets $\mathcal{M} \subseteq \mathcal{N}$ are chosen for a refinement (line 22). Hence, the size of the aggregation $\mathcal{N}$ or the refinement $\mathcal{N}'$ increases in every iteration (lines 14, 15, and 19), and, due to the finiteness of the original tree, $\mathcal{N}'$ can be equal to the original tree after a finite number of iterations, i.e., $\bar{N} = \bar{N}|_{\mathcal{N}'}$. In this (worst) case, the second error and its (heuristic) estimate, $\Delta\left(\mathcal{N}', \bar{N}\right)$ and $\mathcal{D}\left(\mathcal{N}'\right)$, respectively, are zero by definition. If the algorithm continues to select sets $\mathcal{M} \subseteq \mathcal{N}$ to reduce the first error, $\Delta\left(\mathcal{N}, \mathcal{N}'\right)$, the aggregation $\mathcal{N}$ is growing (lines 8 and 11) and can be equal to the original tree after a finite number of steps. However, in the general case, the algorithm stops because the first and second error are small, i.e., $\Delta\left(\mathcal{N}, \mathcal{N}'\right) \leq \sigma_P \left|v\left(\mathcal{N}'\right)\right|$ and $\mathcal{D}\left(\mathcal{N}'\right) \leq \sigma_D \mathcal{D}_0$. $\square$

To solve (MSP $[\mathcal{N}]$) and (MSP $[\mathcal{N}']$), warm starts of the solver are desirable, since the problems that are solved during the iteration process are similar. In fact, only new variables and related constraints are introduced and the objective function is changing. Furthermore, for an efficient algorithm, the problems (MSP $[\mathcal{N}']$) and (P) should be solved only approximately. Lower and upper bounds on $v\left(\mathcal{N}'\right)$ can be sufficient for an estimate of the first error, $\Delta\left(\mathcal{N}, \mathcal{N}'\right)$, and also an approximate solution of (P) can give reliable information for the determination of the sets $\mathcal{M} \subseteq \mathcal{N}$. For example, a bundle algorithm based on Lagrangian Decomposition (cf. [23]) might be suitable.

4 Numerical results

The described algorithm has been implemented for linear multistage stochastic programs. The model can be formulated in GAMS language [10], whereby information about timestages and the original scenario tree is given separately [23]. The arising optimization problems are solved by CPLEX [3].

We have applied the algorithm to three different linear stochastic programming problems. The first one is a twostage program with stochastic technology matrix. The second is a multistage program modeling a portfolio management problem with stochastic right-hand-side and stochastic objective function. The last example is a multistage program arising in power management where only the right-hand-side is stochastic.

All models were run for several combinations of the parameters σ_P and σ_D. The influence of these parameters on the number of scenarios and nodes of the final reduced tree, and the accuracy of the optimal value are presented in several tables. The accuracy is measured as relative distance between the optimal values with respect to the reduced tree and the original tree, i.e., $\frac{\left|v(\mathcal{N})-v(\bar{N})\right|}{1+\left|v(\bar{N})\right|}$. Also, the iteration progress is illustrated for some selected values of σ_P and σ_D. The left picture in these figures shows the optimal values for the reduced tree $\bar{N}|_{\mathcal{N}}$, the refined tree $\bar{N}|_{\mathcal{N}'}$, and the original tree $\bar{N}$ (indicated by a straight line). The right picture shows the quotients $v\left(\mathcal{N}\right)/v\left(\mathcal{N}'\right)$ and $\mathcal{D}\left(\mathcal{N}'\right)/\mathcal{D}_0$ and marks the interval $\left[\frac{1}{1+\sigma_P}, 1+\sigma_P\right]$ and the bound σ_D. Hence, we can track which of the cases discussed in the sections 3.1.1, 3.1.2, and 3.2 occur in which iteration. If the condition $v\left(\mathcal{N}'\right) - v\left(\mathcal{N}\right) < \sigma_P \left|v\left(\mathcal{N}'\right)\right|$ fails (when the curve falls below the marked interval), a set $\mathcal{M} \subseteq \mathcal{N}$ is selected according to (3.3) and $v\left(\mathcal{N}\right)$ increases. If the condition $v\left(\mathcal{N}\right) - v\left(\mathcal{N}'\right) < \sigma_P \left|v\left(\mathcal{N}'\right)\right|$ fails (when the curve exceeds the marked interval), problem (P) is solved and $\mathcal{M}$ is selected according to formula (3.2). Otherwise, if $\Delta\left(\mathcal{N}, \mathcal{N}'\right) \leq \sigma_P \left|v\left(\mathcal{N}'\right)\right|$, but $\mathcal{D}\left(\mathcal{N}'\right) > \sigma_D \mathcal{D}_0$, then $\mathcal{N}'$ is refined, i.e., sets $N' \in \mathcal{N}'$ that contribute most to $\mathcal{D}\left(\mathcal{N}'\right)$ are selected and refined.

The parameter $\bar{m}$, which determines the growth rate of the reduced tree, has been chosen relatively small ($\bar{m} = 2$) to avoid the addition of too many unnecessary scenarios, but on the cost of running time.

4.1 Product mix problem

The first example is a twostage model with simple recourse and stochastic technology matrix from [16, page 554]. It describes the production of four classes of products each consuming a certain number of man-hours in two types of factories. Each product earns a certain fixed profit per item, and it is possible to purchase casual labor from outside. The available man-hours (η_1, η_2) are normally distributed random variables with mean 6000 and standard deviation 100 in the first factory, and mean 4000 and standard deviation 50 in the second factory. The man-hours $\xi_{i,j}$ needed to produce a product of type j in factory i are uniformly distributed random variables.

$$
\begin{array}{ll}
\max\limits_{x,y} & \mathbb{E}\left[12x_1 + 20x_2 + 18x_3 + 40x_4 - 5y_1^- - 10y_2^-\right] \\
\text{s.t.} & \xi_{1,1}x_1 + \xi_{1,2}x_2 + \xi_{1,3}x_3 + \xi_{1,4}x_4 - \eta_1 = y_1^+ - y_1^- \\
 & \xi_{2,1}x_1 + \xi_{2,2}x_2 + \xi_{2,3}x_3 + \xi_{2,4}x_4 - \eta_2 = y_2^+ - y_2^- \\
 & x, y^+, y^- \geq 0 \qquad x_1, \ldots, x_4 \text{ deterministic}
\end{array}
$$

An SMPS-formulation of the model is available in the COIN-OR-library [2], including a scenario tree on 300 scenarios. We run the algorithm with several combinations of σ_P and σ_D, see Table 1. It can be observed that the optimal value depends highly on the chosen set of scenarios. Hence, a too rough choice for σ_D can result in a refinement $\mathcal{N}'$ that gives no meaningful approximation of the original scenario tree. Thus, if by coincidence the optimal values of (MSP $[\mathcal{N}]$) and (MSP $[\mathcal{N}']$) are close, the algorithm stops with a still very inaccurate approximation of the original problem, see, e.g., the entries for $\sigma_D = 0.5$ and $\sigma_P = 0.005$ in Table 1 and see Figure 4.1. The accuracy improves for smaller values of σ_D, see, e.g., Figure 4.2.

$\sigma_P \backslash \sigma_D$	0.5	0.1	0.05	0.01
0.1	269	70	120	133
0.01	25	93	231	260
0.005	61	177	177	257
0.001	239	239	239	300
0.0005	239	239	239	300
0.0001	269	269	269	269

$\sigma_P \backslash \sigma_D$	0.5	0.1	0.05	0.01
0.1	1.66 %	11.87 %	5.70 %	3.40 %
0.01	6.17 %	.47 %	4.55 %	.16 %
0.005	13.37 %	5.58 %	5.58 %	2.19 %
0.001	3.14 %	3.14 %	3.14 %	0 %
0.0005	3.14 %	3.14 %	3.14 %	0 %
0.0001	1.66 %	1.66 %	1.66 %	1.66 %

Table 1: Product mix problem: Number of scenarios of the final reduced tree $\bar{N}|_{\mathcal{N}}$ (left) and accuracy of the optimal value (right) for several combinations of σ_P and σ_D.

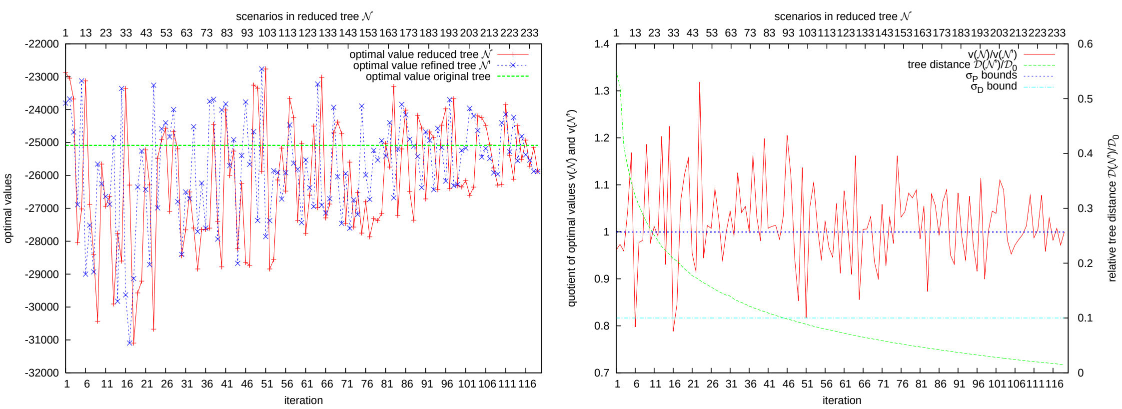

Figure 4.1: Product mix problem: Iteration progress for $\sigma_p = 0.001$ and $\sigma_D = 0.1$.

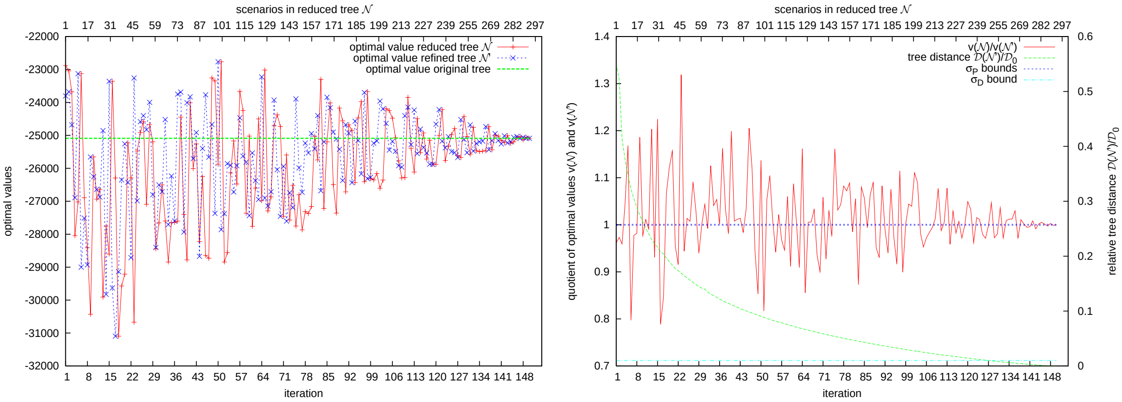

Figure 4.2: Product mix problem: Iteration progress for $\sigma_p = 0.001$ and $\sigma_D = 0.01$.

4.2 Portfolio management

The next example deals with the problem of financing business activities by borrowing and lending bonds with different maturities. The aim is to maximize the profit which is determined by the interest rate and price changes of bonds, and the earnings of the underlying business activities [8].

Let $\mathcal{M} := \{1, \ldots, 36\}$ be a set of maturities in months, and $\mathcal{M}^S := \{1, 2, 3, 6, 12, 24, 36\} \subseteq \mathcal{M}$ be a subset of standard maturities for borrowing and lending. The amount of borrowing and lending introduced to the market at time t with maturity m is denoted by $v_t^{m,+}$ and $v_t^{m,-}$, respectively; v_t^m is the total volume of borrowing and lending at time t with maturity m, i.e., $v_t^m = v_{t-1}^{m+1} + v_t^{m,+} - v_t^{m,-}$, $m \in \mathcal{M}^S$, and $v_t^m = v_{t-1}^{m+1}$, $m \in \mathcal{M} \setminus \mathcal{M}^S$. The total volume at time t is given by $x_t = \sum_{m\in\mathcal{M}} v_t^m$. The stochastic change in volume is denoted by ξ_t, i.e., $x_t = x_{t-1} + \xi_t$, and unlimited funding is prohibited by the inequality $\sum_{m\in\mathcal{M}^S} v_t^{m,+} - v_{t-1}^1 \leq \xi_t$. A starting portfolio v_0^m, $m \in \mathcal{M}$, has been given as input. Finally, the stochastic processes of returns for borrowing and lending with maturity $m \in \mathcal{M}^S$ per period are denoted by $\eta_t^{m,+}$ and $\eta_t^{m,-}$, and the process of return for the associated underlying business activity by η_t^x. The complete model has the form

$$
\begin{array}{lll}
\max\limits_{v,x} & \mathbb{E}\left[\sum_{t=0}^T \left\{\sum_{m\in\mathcal{M}^S} \eta_t^{m,-} v_t^{m,-} - \eta_t^{m,+} v_t^{m,+}\right\} + \eta_t^x x_t\right] & \\
\text{s.t.} & v_t^m - v_{t-1}^{m+1} - v_t^{m,+} + v_t^{m,-} = 0 & t = 1, \ldots, T,\ m \in \mathcal{M}^S \\
& v_t^m - v_{t-1}^{m+1} = 0 & t = 1, \ldots, T,\ m \in \mathcal{M} \setminus \mathcal{M}^S \\
& x_t - \sum_{m\in\mathcal{M}} v_t^m = 0 & t = 1, \ldots, T \\
& x_t - x_{t-1} = \xi_t & t = 1, \ldots, T \\
& \sum_{m\in\mathcal{M}^S} v_t^{m,+} - v_{t-1}^1 \leq \xi_t & t = 1, \ldots, T \\
& v_t^{m,+}, v_t^{m,-}, v_t^m, x_t \geq 0 \text{ and } \mathcal{F}_t\text{-measurable} & t = 1, \ldots, T,\ m \in \mathcal{M}^S
\end{array}
$$

For the numerical experiments we have chosen two trees from [8]. The first tree has $T = 5$ stages, 625 scenarios, and 781 nodes. The second tree has $T = 6$ stages, 3125 scenarios, and 3906 nodes.

For the tree on 625 scenarios, it can be observed that with only 89 scenarios an approximation with only 1% perturbation in the optimal value can be achieved ($\sigma_P = 0.005$, $\sigma_D = 0.01$), cf. Table 2. Similarly, for the tree on 3125 scenarios, a reduction of approx. 90% (in number of scenarios) still allows an approximation with at most 1% perturbation in the optimal value, cf. Table 3.

$\sigma_P \backslash \sigma_D$	0.5	0.1	0.05	0.01
0.1	1	5	5	5
0.01	7	34	50	63
0.005	10	45	58	89
0.001	9	48	83	136
0.0005	9	90	211	242
0.0001	9	82	134	289

$\sigma_P \backslash \sigma_D$	0.5	0.1	0.05	0.01
0.1	5	19	19	19
0.01	24	91	126	149
0.005	30	110	134	188
0.001	28	112	174	263
0.0005	28	176	345	392
0.0001	28	159	235	439

$\sigma_P \backslash \sigma_D$	0.5	0.1	0.05	0.01
0.1	14.39%	7.95%	7.95%	7.95%
0.01	6.05%	2.18%	1.67%	1.43%
0.005	6.19%	2.18%	1.71%	1.00%
0.001	5.74%	1.87%	1.10%	0.68%
0.0005	5.74%	1.28%	0.73%	0.62%
0.0001	5.74%	1.48%	0.97%	0.53%

Table 2: Portfolio management (625 scenarios): Number of scenarios (upper left) and nodes (upper right) of the final reduced tree $\bar{N}|_{\mathcal{N}}$ and accuracy of the optimal value (down) for several values of σ_P and σ_D.

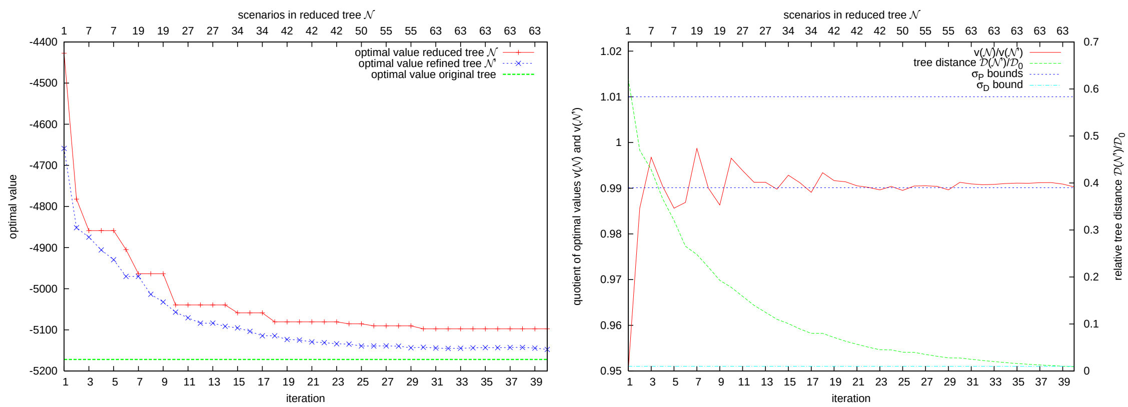

Figure 4.3: Portfolio management (625 scenarios): Iteration progress for $\sigma_P = 0.01$ and $\sigma_D = 0.01$.

4.3 Power management in a hydro-thermal system under uncertain load

In the final example we consider a power generation system consisting of a collection of coal fired thermal units I and pumped hydro units J. The aim is to find cost-optimal operation levels of the thermal and hydro units for one week under uncertain electric load. This model is a simplified version of the one described in [18].

Fixed parameters for the thermal units are fuel prices $b \geq 0$ and operation ranges $0 \leq \underline{p} \leq \bar{p}$. For the hydro units the parameters are the efficiency of the pumps $0 \leq \eta \leq 1$, the maximal operation levels of the pumps and turbines $\bar{w}, \bar{v} \geq 0$ and the capacity of the reservoirs $\bar{l} \geq 0$. For the beginning and end of the week, the fill levels l_{in} and l_{end} of the reservoirs are given. The stochastic data consists of the load d and the required reserve r for one week. We search for a sequence of operation levels for the thermal units p, pumps w, and turbines v, and of fill levels l in the pump-reservoirs.

Further on, the decision process underlies the following constraints: the fill levels at the beginning and end of the week, (4.1a) and (4.1b), time-coupling of the fill levels, (4.1c), meet of the load, (4.1d), and reserve-requirements, (4.1e). The objective is to minimize the expected fuel

$\sigma_P \backslash \sigma_D$	0.5	0.1	0.05	0.01
0.1	1	13	13	13
0.01	11	64	167	270
0.005	7	91	223	327
0.001	16	113	367	647
0.0005	16	143	335	671
0.0001	26	384	465	856

$\sigma_P \backslash \sigma_D$	0.5	0.1	0.05	0.01
0.1	6	53	53	53
0.01	40	203	463	666
0.005	28	275	576	754
0.001	58	315	808	1226
0.0005	58	380	746	1247
0.0001	72	732	886	1478

$\sigma_P \backslash \sigma_D$	0.5	0.1	0.05	0.01
0.1	13.52%	9.67%	9.67%	9.67%
0.01	6.41%	2.15%	1.50%	1.16%
0.005	6.62%	1.50%	1.08%	0.64%
0.001	6.31%	1.45%	0.61%	0.24%
0.0005	6.31%	1.38%	0.77%	0.20%
0.0001	6.44%	1.54%	0.75%	0.15%

Table 3: Portfolio management (3125 scenarios): Number of scenarios (upper left) and nodes (upper right) of the final reduced tree $\bar{N}|_{\mathcal{N}}$ and accuracy of the optimal value (down) for several values of σ_P and σ_D.

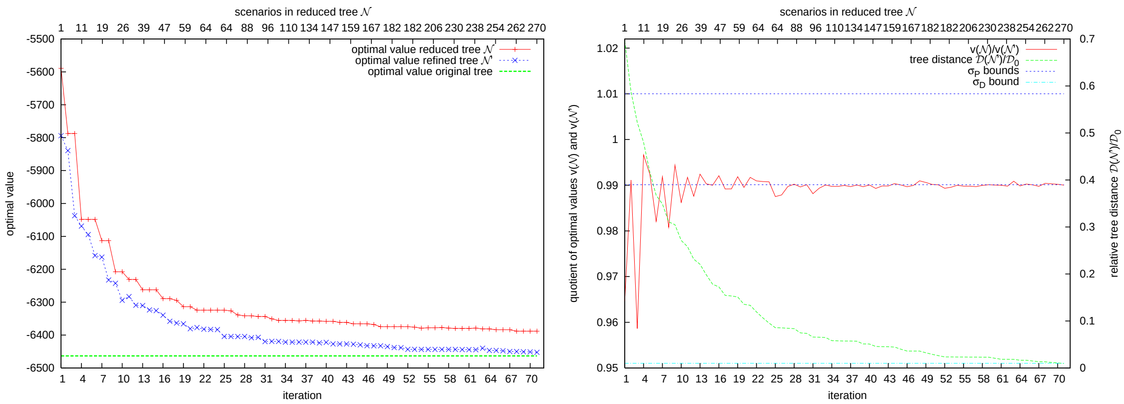

Figure 4.4: Portfolio management (3125 scenarios): Iteration progress for $\sigma_P = 0.01$, $\sigma_D = 0.01$.

costs. Thus, the complete model has the form

$$\min_{p,l,v,w} \mathbb{E}\left[\sum_{i\in I}\sum_{t=1}^{T} b_i p_{i,t}\right]$$

$$\text{s.t. } l_{j,1} + (v_{j,1} - \eta_j w_{j,1}) = l_{\text{in},j} \qquad j \in J \qquad (4.1a)$$

$$l_{j,T} = l_{\text{end},j} \qquad j \in J \qquad (4.1b)$$

$$l_{j,t} - l_{j,t-1} + (v_{j,t} - \eta_j w_{j,t}) = 0 \qquad t = 2,\dots,T,\ j \in J \qquad (4.1c)$$

$$\textstyle\sum_{i\in I} p_{i,t} + \sum_{j\in J}(v_{j,t} - w_{j,t}) \geq d_t \qquad t = 1,\dots,T \qquad (4.1d)$$

$$\textstyle\sum_{i\in I}(\bar{p}_i - p_{i,t}) \geq r_t \qquad t = 1,\dots,T \qquad (4.1e)$$

$$\underline{p}_t \leq p_{i,t} \leq \bar{p}_t \qquad t = 1,\dots,T,\ i \in I$$

$$0 \leq v_{j,t} \leq \bar{v}_j,\ 0 \leq w_{j,t} \leq \bar{w}_j,\ 0 \leq l_{j,t} \leq \bar{l}_j \qquad t = 1,\dots,T,\ j \in J$$

$$p_{i,t}, v_{j,t}, w_{j,t}, l_{j,t} \in \mathcal{F}_t \qquad t = 1,\dots,T,\ i \in I,\ j \in J$$

Two scenario trees modeling the stochasticity in the electric load over a time horizon of one week with hourly discretization points ($T = 168$) were available. The first tree consists of 64 scenarios and 3048 nodes, the second one of 729 scenarios and 22623 nodes (Figures 4.5 and 4.6).

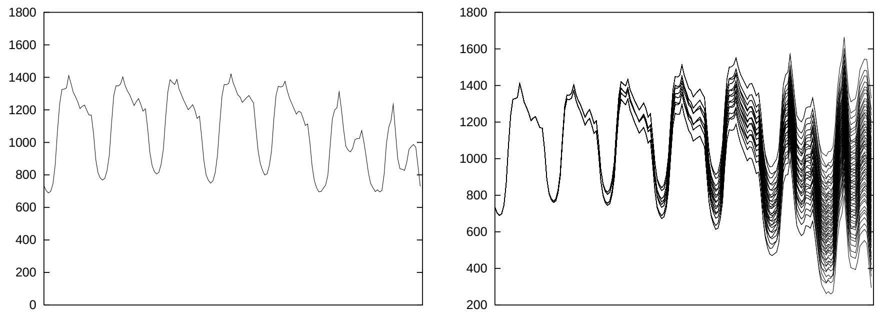

Figure 4.5: On the left, one scenario of the electric load for one week. The seven days are good recognizable. On the right, 64 scenarios of the electric load for one week.

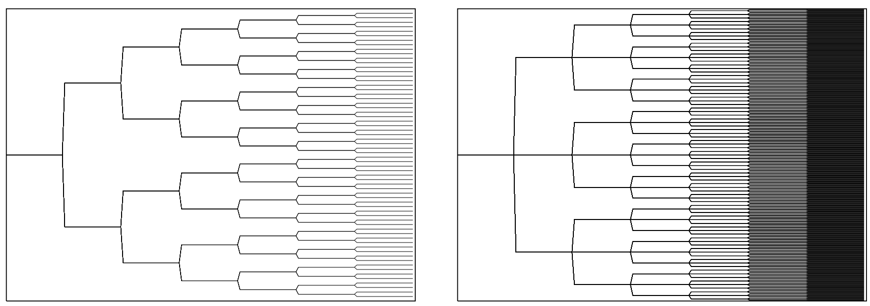

Figure 4.6: The scenario trees for the electric load of one week. Left: 64 scenarios, 3048 nodes. Right: 729 scenarios, 22623 nodes.

When running the algorithm for the tree on 64 scenarios and parameters $\sigma_P = 0.001$ and $\sigma_D = 0.01$, it results in a tree on 46 scenarios, see Figure 4.9 (left). The influence of the parameters σ_P and σ_D is visualized by Table 4. While σ_P has obviously a high impact on the final number of scenarios, the distance $\mathcal{D}(\mathcal{N}')$, controlled by σ_D, seems to be of minor importance.

For the tree on 729 scenarios, $\sigma_P = 0.001$, and $\sigma_D = 0.01$ the algorithm finishes with a tree on 106 scenarios, see Figure 4.9 (right). Figure 4.8 shows the progress of the optimal values $v(\mathcal{N})$ and $v(\mathcal{N}')$ and how the first error and the term $\mathcal{D}(\mathcal{N}')$ develops during the iteration progress. One observes that after the 14th iteration, the remaining iterations were used to push $\mathcal{D}(\mathcal{N}')/\mathcal{D}_0$ below the tolerance σ_D. When increasing σ_D to 0.1, one obtains the same tree, but the number of iterations decreases from 40 to 20.

The numerical results show that scenario trees for stochastic programs in power management environments allow a reduction of 70 to 90% (in number of scenarios) while loosing at most 0.1% accuracy in the optimal value.

5 Conclusions

We have proposed a new algorithm for the discretization of multistage stochastic programs, which integrates the reduction of a scenario tree into the solution of the discretized problem. We estimated the approximation error by a sum of two terms, which we aim to diminish separately, see

$\sigma_P \backslash \sigma_D$	0.5	0.1	0.05	0.01
0.1	1	1	1	1
0.01	5	7	7	7
0.005	9	54	54	57
0.001	34	34	46	46
0.0005	41	41	41	41
0.0001	48	48	48	48

$\sigma_P \backslash \sigma_D$	0.5	0.1	0.05	0.01
0.1	168	168	168	168
0.01	624	720	720	720
0.005	840	2640	2640	2856
0.001	2112	2112	2592	2592
0.0005	2448	2448	2448	2448
0.0001	2664	2664	2664	2664

$\sigma_P \backslash \sigma_D$	0.5	0.1	0.05	0.01
0.1	0.56%	0.56%	0.56%	0.56%
0.01	0.47%	0.22%	0.22%	0.22%
0.005	0.47%	0.72%	0.72%	0.15%
0.001	0.11%	0.11%	0.02%	0.02%
0.0005	0.04%	0.04%	0.04%	0.04%
0.0001	0%	0%	0%	0%

Table 4: Power management model (64 scenarios): Number of scenarios (upper left) and nodes (upper right) of the final reduced tree $\bar{N}|_{\mathcal{N}}$ and accuracy of the optimal value (down) for several values of σ_P and σ_D.

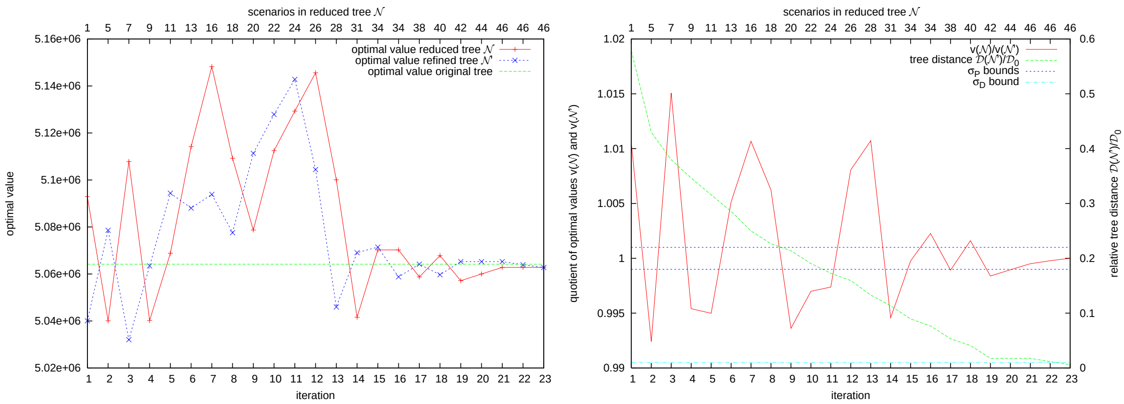

Figure 4.7: Power management model (64 scenarios): Iteration progress for $\sigma_p = 0.001$ and $\sigma_D = 0.01$.

$\sigma_P \backslash \sigma_D$	0.1	0.01
0.1	1	1
0.01	1	1
0.005	91	91
0.001	106	106
0.0001	267	478

$\sigma_P \backslash \sigma_D$	0.1	0.01
0.1	168	168
0.01	168	168
0.005	6384	6384
0.001	7224	7224
0.0001	13944	20160

$\sigma_P \backslash \sigma_D$	0.1	0.01
0.1	.18 %	.18 %
0.01	.18 %	.18 %
0.005	.41 %	.41 %
0.001	.03 %	.03 %
0.0001	.04 %	.01 %

Table 5: Power management model (729 scenarios): Number of scenarios (left) and nodes (middle) of the final reduced tree $\bar{N}|_{\mathcal{N}}$ and accuracy of the optimal value (right) for several values of σ_P and σ_D.

(3.1). The first term is oriented on the optimization problem itself and determines directly the size of the final approximation, while the latter is to ensure reliability of the first term, thus it

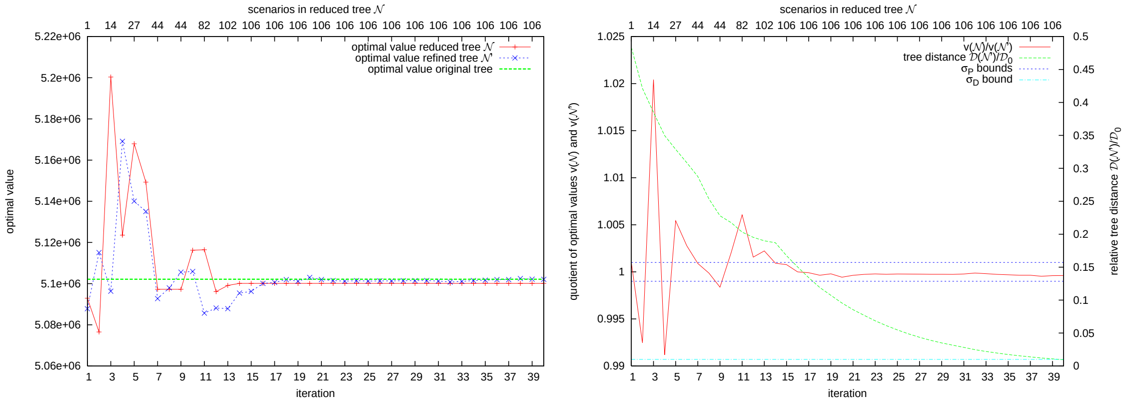

Figure 4.8: Power management model (729 scenarios): Iteration progress for $\sigma_p = 0.001$ and $\sigma_D = 0.01$.

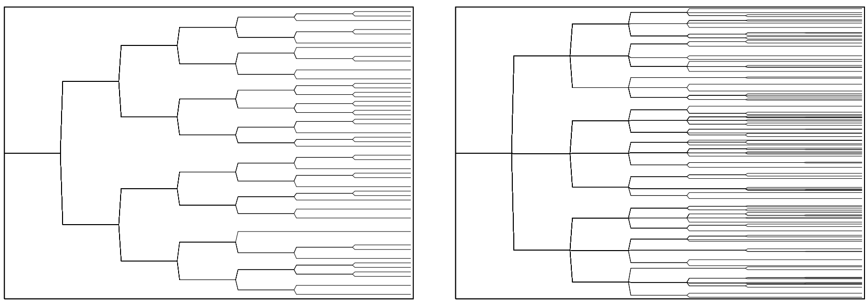

Figure 4.9: Power management model: Reduced scenario trees on 46 and 77 scenarios. The original scenario trees had 64 and 729 scenarios, respectively.

influences the size of the final approximation only indirectly. This division allows the construction of reduced scenario trees that are highly adapted to the optimization problem, and allow therefore much smaller trees than an approach that concentrates on the stochastic process only.

The efficiency of the algorithm depends mostly on the effort spend to solve the problems (MSP $[\mathcal{N}]$) and (MSP $[\mathcal{N}']$). In the current implementation, these problems were solved up to optimality, which is quite expensive, so that further developments might estimate the first error by use of lower and upper bounds on the optimal values. Also an approximate solution of (P) could give information still reliable enough to determine the set $\mathcal{M}$.

An advantage of the proposed approach is its potential extension to nonconvex problems, as discussed in the introduction and proposed in [17, 23]. Using a convex relaxation of (MSP $[\mathcal{N}]$), the gap between the aggregated problem and its refinement, $\Delta(\mathcal{N}, \mathcal{N}')$, can be estimated by the gap between convex relaxations of (MSP $[\mathcal{N}]$) and (MSP $[\mathcal{N}']$). Hence, we can apply the algorithms from Section 3.1 and 3.2. For the reduction of the relaxation gaps we propose a Lagrangian Decomposition approach where cuts for the master problem are constructed from solutions of the subproblems [17, 23]. The dual parameter for the Lagrangian subproblems may be computed by a Column Generation method, i.e., a convex inner approximation of the convexified feasible set. This approach has the advantage that the generated inner approximation points can easily be updated after a refinement of the aggregations $\mathcal{N}$ and $\mathcal{N}'$.

Acknowledgements:

This work was supported by the BMBF under the grant 03SF0312E. The authors wish to thank Holger Heitsch and Werner Römisch (Humboldt-University Berlin) for helpful discussions and providing the scenario trees for the power management problems.

References

[1] M. S. Casey, S. Sen, The scenario generation algorithm for multistage stochastic linear programming, Mathematics of Operations Research, 30 (2005), 615–631

[2] Computational Infrastructure for Operations Research, `http://www.coin-or.org`

[3] CPLEX, ILOG Inc., `http://www.ilog.com/products/cplex`

[4] M.A.H. Dempster, Sequential importance sampling algorithms for dynamic stochastic programming, Zapiski Nauchnykh Seminarov POMI, 312 (2004), 94–129

[5] J. Dupačová, G. Consigli, W. Römisch, Scenarios for multistage stochastic programs, Annals of Operations Research, 100 (2000), 25–53

[6] J. Dupačová, N. Gröwe-Kuska, W. Römisch, Scenario reduction in stochastic programming - an approach using probability metrics, Mathematical Programming, Series A, 95 (2003), 493–511

[7] N. C. P. Edirisinghe, Bound-based approximations in multistage stochastic programming: Nonanticipativity aggregation, Annals of Operations Research 85, 103–127 (1999)

[8] K. Frauendorfer, G. Haarbrücker, C. Marohn, M. Schürle, SGPF - portfolio test problems for stochastic multistage linear programming, Institute of Operations Research, University of St. Gallen, `http://www.ifu.unisg.ch/sgpf/`

[9] H. I. Gassmann, MSLiP: A computer code for the multistage stochastic linear programming problem, Mathematical Programming 47 (1990), 407–423

[10] General Algebraic Modeling System, `http://www.gams.com`

[11] H. Heitsch, W. Römisch, Scenario reduction algorithms in stochastic programming, Computational Optimization and Applications 24 (2003), 187–206

[12] H. Heitsch, W. Römisch, Scenario tree modelling for multistage stochastic programs, Preprint 296 (2005), DFG Research Center Matheon "Mathematics for key technologies"

[13] H. Heitsch, W. Römisch, C. Strugarek, Stability of multistage stochastic programs, Preprint 255 (2005), DFG Research Center Matheon "Mathematics for key technologies", and to appear in SIAM Journal on Optimization

[14] J. L. Higle, B. Rayco, S. Sen, Stochastic scenario decomposition for multi-stage stochastic programs, submitted to Annals of Operations Research

[15] K. Høyland, M. Kaut, S.W. Wallace, A heuristic for moment-matching scenario generation, Computational Optimization and Applications 24 (2003), 169–185

[16] A. J. King, Stochastic programming problems: examples from the literature, in: Numerical techniques for stochastic optimization, Springer Ser. Comput. Math. 10 (1988), 543–567

[17] I. Nowak, Relaxation and decomposition methods for mixed integer nonlinear programming, Springer International Series of Numerical Mathematics Vol. 152 (2005), Birkhäuser

[18] M.P. Nowak, W. Römisch, Optimal power dispatch via multistage stochastic programming, Progress in Industrial Mathematics at ECMI 96 (M. Brøns, M.P. Bendsøe and M.P. Sørensen Eds.) (1997), Teubner, 324–331

[19] G.Ch. Pflug, Scenario tree generation for multiperiod financial optimization by optimal discretization, Mathematical Programming 89 (2001), 251-271

[20] W. Römisch, Stability of stochastic programming problems, Chapter 8 of [21], 483–554

[21] A. Ruszczyński, A. Shapiro, Stochastic Programming, Handbooks in operations research and management science, Volume 10, Elsevier Science, 2003

[22] A. Shapiro, Stochastic programming approach to optimization under uncertainty, submitted (2006)

[23] S. Vigerske, Adaptive discretization of stochastic programs, Diploma thesis, Humboldt-Universität Berlin, 2005