Scenario Reduction Algorithms in Stochastic Programming

Holger Heitsch and Werner Römisch

Humboldt-University Berlin, Institute of Mathematics, 10099 Berlin, Germany

Abstract. We consider convex stochastic programs with an (approximate) initial probability distribution P having finite support $\operatorname{supp} P$, i.e., finitely many *scenarios*. Such stochastic programs behave stable with respect to perturbations of P measured in terms of a Fortet-Mourier probability metric. The problem of optimal *scenario reduction* consists in determining a probability measure which is supported by a subset of $\operatorname{supp} P$ of prescribed cardinality and is closest to P in terms of such a probability metric. Two new versions of forward and backward type algorithms are presented for computing such optimally reduced probability measures approximately. Compared to earlier versions, the computational performance (accuracy, running time) of the new algorithms is considerably improved. Numerical experience is reported for different instances of scenario trees with computable optimal lower bounds. The test examples also include a ternary scenario tree representing the weekly electrical load process in a power management model.

Key Words: stochastic programming, probability metric, scenario reduction, scenario tree, electrical load
2000 MSC: 90C15

1 Introduction

Many stochastic decision problems may be formulated as convex stochastic programs of the form

$$\min\{\int_{\Omega} f_0(\omega,x)P(d\omega) : x \in X\}\,, \tag{1}$$

where $X \subset \mathbb{R}^m$ is a given nonempty closed convex set, Ω a closed subset of $\mathbb{R}^s$, the function f_0 from $\Omega \times \mathbb{R}^m$ to $\mathbb{R}$ is continuous with respect to ω and convex with respect to x, and P is a fixed Borel probability measure on Ω, i.e., $P \in \mathcal{P}(\Omega)$. For instance, this formulation covers (convex) two- and multi-stage stochastic programs with recourse.

Typical integrands $f_0(\cdot,x)$, $x \in X$, in convex stochastic programming problems are nondifferentiable, but locally Lipschitz continuous on Ω. In the following, we assume that there exist a continuous and nondecreasing function $h : \mathbb{R}_+ \to \mathbb{R}_+$ with $h(0) = 0$, a nondecreasing function $g : \mathbb{R}_+ \to \mathbb{R}_+ \setminus \{0\}$ and some fixed element $\omega_0 \in \mathbb{R}^s$ such that

$$|f_0(\omega,x) - f_0(\tilde{\omega},x)| \le g(\|x\|)c(\omega,\tilde{\omega}) \tag{2}$$

for each $x \in X$, where the function $c : \Omega \times \Omega \to \mathbb{R}$ is given by

$$c(\omega, \tilde{\omega}) := \max\{1, h(\|\omega - \omega_0\|), h(\|\tilde{\omega} - \omega_0\|)\}\|\omega - \tilde{\omega}\|, \forall \omega, \tilde{\omega} \in \Omega. \quad (3)$$

This means that the function $h(\| \cdot -\omega_0\|)$ describes the growth of the local Lipschitz constants of $f_0(\cdot, x)$ in balls around ω_0 with respect to some norm $\| \cdot \|$ on $\mathbb{R}^s$. An important particular case is that the function h grows polynomially, i.e., $h(r) = r^{p-1}$ for $r \in \mathbb{R}_+$ and some $p \geq 1$. For instance, it is shown in [11] that the choice $p = 2$ is appropriate for two-stage models with stochasticity entering prices and right-hand sides.
It is shown in [4,11] that the model (1) behaves stable with respect to small perturbations in terms of the probability metric

$$\zeta_c(P, Q) := \sup_{f \in \mathcal{F}_c} | \int_\Omega f(\omega) P(d\omega) - \int_\Omega f(\omega) Q(d\omega) | , \quad (4)$$

where $\mathcal{F}_c$ is the class of continuous functions defined by

$$\mathcal{F}_c = \{f : \Omega \to \mathbb{R} : f(\omega) - f(\tilde{\omega}) \leq c(\omega, \tilde{\omega}) \text{ for all } \omega, \tilde{\omega} \in \Omega\} \quad (5)$$

and probability measures P and Q in the set

$$\mathcal{P}_c(\Omega) := \{Q \in \mathcal{P}(\Omega) : \int_\Omega c(\omega, \omega_0) Q(d\omega) < \infty\} \quad (6)$$

(see also the earlier work in [13]). The distance ζ_c is a probability metric on $\mathcal{P}_c(\Omega)$ with ζ-structure and also called a Fortet-Mourier (type) metric. In this generality, it is introduced in [14] and further studied in [10,12]. In particular, the metric ζ_c has dual representations in terms of the Kantorovich-Rubinstein functional (cf. Section 5.3 in [10] and [12]).
An important instance is that the initial probability measure P is itself discrete with finitely many atoms (or *scenarios*) or that a good discrete approximation of P is available. Its support may be very large so that, for reasons of computational complexity and time limitation, this probability measure is further approximated by a probability measure Q carried by a (much) smaller subset of scenarios. In this case the distance $\zeta_c(P, Q)$ represents the optimal value of a finite-dimensional linear program. More precisely, from (4) we obtain for $P = \sum_{i=1}^N p_i \delta_{\omega_i}$ and $Q = \sum_{j=1, j \notin J}^N q_j \delta_{\omega_j}$ that

$$\zeta_c(P, Q) = \max\{\sum_{i=1}^N p_i u_i - \sum_{\substack{j=1 \\ j \notin J}}^N q_j u_j : u_i - u_j \leq c(\omega_i, \omega_j), \forall i, j \in \{1, \ldots, N\}\},$$

where $J \subset \{1, \ldots, N\}$ and $\delta_\omega \in \mathcal{P}(\Omega)$ denotes the Dirac measure placing unit mass at ω. In particular, the metric ζ_c can be used to evaluate distances of specific probability measures obtained during a scenario-reduction process. Various reduction rules appear in the literature in the context of recent large-scale real-life applications. We refer to the corresponding discussion in [4], to the recent work [3] on scenario generation and reduction, and to the paper [9], in which an approach to scenario generation based on Fortet-Mourier distances is given.

In the present paper, we follow the approach for reducing scenarios of a given discrete probability measure $P = \sum_{i=1}^{N} p_i \delta_{\omega_i}$ developed in [4]. It consists in determining an index set $J_* \subset \{1, \ldots, N\}$ of given cardinality $\# J_* = N - n$ and a probability measure $Q_* = \sum_{j=1, j \notin J_*}^{N} q_j^* \delta_{\omega_j}$ such that

$$\zeta_c(P, Q_*) = \min\{\zeta_c(P, \sum_{\substack{j=1 \\ j \notin J}}^{N} q_j \delta_{\omega_j}) : J \subset \{1, \ldots, N\}, \# J = N - n, \qquad (7)$$
$$\sum_{j \notin J} q_j = 1, q_j \geq 0, j \notin J\}.$$

Problem (7) may be reformulated as a mixed-integer program. In Section 2 we derive bounds for (7) and develop two new algorithms (*fast forward selection* and *simultaneous backward reduction*), which constitute heuristics for solving (7). We study their complexity and their relations to the algorithms in [4]. Indeed, the fast forward selection algorithm turns out to be an efficient implementation of the forward selection procedure of [4], producing the same reduced probability measures. In order to compare the performance of the algorithms we provide, in Section 3, explicit formulas for the minimal distances (7) in case that P is a *regular* (binary or ternary) scenario tree (i.e., a tree having a specific structure) and Q_* is a reduced tree with fixed cardinality n. In Section 4 we report on numerical experience for the reduction of regular binary and ternary scenario trees. The test trees also include the ternary scenario tree representing the weekly electrical load process in a power management model which was considered in [4]. It turns out that the new implementation of the fast forward selection algorithm is about 10-100 times faster than the earlier version. Furthermore, fast forward selection is the best algorithm when comparing accuracy. The results of the simultaneous backward reduction algorithm are more accurate than the backward reduction variant of [4] in most cases, but at the expense of higher running times. When comparing running times, fast forward selection (simultaneous backward reduction) is preferable in case that $n < \frac{N}{4}$ $(n > \frac{N}{4})$(approximately).

2 Scenario Reduction

We consider the stochastic program (1) and select the function c of form (3) such that the Lipschitz condition (2) is satisfied. Let the initial probability distribution P be discrete and carried by finitely many scenarios $\omega_i \in \Omega$ with weights $p_i > 0$, $i = 1, \ldots, N$, and $\sum_{i=1}^{N} p_i = 1$, i.e., $P = \sum_{i=1}^{N} p_i \delta_{\omega_i}$. Let $n \in \mathbb{N}$, $n < N$, $J \subset \{1, \ldots, N\}$ with $\# J = N - n$ and consider the probability measure Q having scenarios ω_j with probabilities q_j, $j \in \{1, \ldots, N\} \backslash J$, i.e., compared to P, the measure $Q = \sum_{j \notin J} q_j \delta \omega_j$ is reduced by deleting all scenarios ω_j, $j \in J$, and by assigning new probabilistic weights q_j to each scenario ω_j, $j \notin J$. The optimal reduction concept described above recommends to consider the probability distance

$$D(J;q) := \zeta_c(\sum_{i=1}^{N} p_i \delta_{\omega_i}, \sum_{j \notin J} q_j \delta_{\omega_j}) \,. \tag{8}$$

depending on the index set J and q. The optimal reduction concept (7) says that the index set J_* and the optimal weight q_* are selected such that $D(J_*;q_*) = \min\{D(J;q) : J \subset \{1,\ldots,N\}, \#J = N-n, \sum_{j \notin J} q_j = 1, q_j \geq 0, j \notin J\}$. First we recall the following bounds for $\min_q D(J;q) = \min\{D(J;q) : \sum_{j \notin J} q_j = 1, q_j \geq 0, j \notin J\}$ when the index set $J \subset \{1,\ldots,N\}$ is fixed ([4], Theorem 3.1).

Theorem 2.1 (redistribution)
For any index set $J \subset \{1,\ldots,N\}$ it holds that

$$\frac{1}{C} \sum_{i \in J} p_i \min_{j \notin J} c(\omega_i, \omega_j) \leq \min_q D(J;q) \leq D(J;\bar{q}) \leq \sum_{i \in J} p_i \min_{j \notin J} c(\omega_i, \omega_j), \tag{9}$$

where $$C := \max_{i \in \{1,\ldots,N\}} \{1, h(\|\omega_i - \omega_0\|)\} \quad \textit{and}$$

$$\bar{q}_j := p_j + \sum_{\substack{i \in J \\ j(i)=j}} p_i \,,\, j \notin J \,,\, j(i) \in \arg\min_{j \notin J} c(\omega_i, \omega_j) \,,\, i \in J. \tag{10}$$

Furthermore, we have equality in (9) and, hence, optimality of $\bar{q}$ if $h \equiv 1$.

For convenience of the reader the proof of Theorem 2.1 is displayed in the Appendix. The interpretation of the formula (10) is that the new probability of a preserved scenario is equal to the sum of its former probability and of all probabilities of deleted scenarios that are closest to it with respect to c. If $h \equiv 1$, we call (10) the *optimal redistribution rule.*
Next we discuss the optimal choice of an index set J for scenario reduction with fixed cardinality $\#J$. Theorem 2.1 motivates to consider the following formulation of the *optimal reduction* problem for given $n \in \mathbb{N}$, $n < N$:

$$\min\{D_J := \sum_{i \in J} p_i \min_{j \notin J} c(\omega_i, \omega_j) : J \subset \{1,\ldots,N\}, \#J = N-n\} \tag{11}$$

Problem (11) means that the set $\{1,\ldots,N\}$ has to be covered by two sets $J \subset \{1,\ldots,N\}$ and $\{1,\ldots,N\} \setminus J$ such that J has fixed cardinality $N-n$ and the cover has minimal cost D_J. Thus, (11) represents a *set-covering problem.* It may be formulated as a 0-1 integer program (cf. [7]) and is $\mathcal{NP}$-hard. Since efficient solution algorithms are hardly available in general, we are looking

for (fast) heuristic algorithms exploiting the structure of the costs D_J. In the specific cases of $n = 1$ and $n = N - 1$, solving (11) becomes quite easy.
In case that $\#J = 1$, the problem (11) takes the form

$$\min_{l\in\{1,\dots,N\}} p_l \min_{j\neq l} c(\omega_l,\omega_j) \tag{12}$$

If the minimum is attained at $l_* \in \{1,\dots,N\}$, i.e., the scenario ω_{l_*} is deleted, the redistribution rule (10) yields the probability distribution of the reduced measure $\bar{Q}$. If $j_* \in \arg\min_{j\neq l_*} c(\omega_{l_*},\omega_j)$, then it holds that $\bar{q}_{j_*} = p_{j_*} + p_{l_*}$ and $\bar{q}_l = p_l$ for all $l \notin \{l_*, j_*\}$. Of course, the optimal deletion of a single scenario may be repeated recursively until a prescribed number $N - n$ of scenarios is deleted. This strategy recommends a conceptual algorithm called *backward reduction.*
In case that $\#J = N - 1$, the problem (11) is of the form

$$\min_{u\in\{1,\dots,N\}} \sum_{i=1}^{N} p_i c(\omega_i,\omega_u)\,. \tag{13}$$

If the minimum is attained at $u_* \in \{1,\dots,N\}$, only the scenario ω_{u_*} is kept and the redistribution rule (10) provides $\bar{q}_{u_*} = p_{u_*} + \sum_{i\neq u_*} p_i = 1$. This strategy provides the basic concept of a second conceptual algorithm called *forward selection.*
First, we take a closer look at the backward reduction strategy. A backward type algorithm was already suggested in [4,6]. It determines a reduced scenario set by reducing $N - n$ scenarios from the original set of scenarios as follows. Let the indices l_i be selected such that

$$l_i \in \arg\min_{l\in\{1,\dots,N\}\setminus\{l_1,\dots,l_{i-1}\}} p_l \min_{j\neq l} c(\omega_l,\omega_j)\,,\ i = 1,\dots,N-n. \tag{14}$$

It can be shown that

$$lb := \sum_{i=1}^{N-n} p_{l_i} \min_{j\neq l_i} c(\omega_{l_i},\omega_j) \tag{15}$$

is a lower bound of the optimal value of (11). Furthermore, it holds that the index set $\{l_1,\dots,l_{N-n}\}$ is a solution of (11) if for all $i = 1,\dots,N-n$, the set $\arg\min_{j\neq l_i} c(\omega_{l_i},\omega_j)\backslash\{l_1,\dots,l_{i-1},l_{i+1},\dots,l_{N-n}\}$ is nonempty ([4,6]). This property motivates the following algorithm. In the first step, an index n_1 with $n \le n_1 < N$ is determined using formula (14) such that $J_1 = \{l_1,\dots,l_{N-n_1}\}$ is a solution of (11) for $n = n_1$. Next, the redistribution rule of Theorem 2.1 is used. This leads to the reduced probability measure P_1 containing all scenarios indexed by $\{1,\dots,N\} \setminus J_1$. If $n < n_1$, the measure P_1 is further reduced by deleting all scenarios belonging to some index set J_2 with $\#J_2 = n_1 - n_2$ and $n \le n_2 < n_1$, which is obtained in the same way using formula (14). This procedure is continued until, in step r, we have $n_r = n$ and $J = \cup_{i=1}^{r} J_i$.

Finally, the redistribution rule (10) is used again for the index set J. This algorithm is called *backward reduction of scenario sets.* There are still many degrees of freedom to choose the next scenario in each step. Often there exist several candidates for deletion. In Section 4 we use one particular implementation of backward reduction of scenario sets.

Another particular variant consists in the case that $\#J_i = 1$ for each $i = 1, \ldots, N-n$. This variant (without the final redistribution) was already announced in [2,5]. However, numerical tests have shown that the backward reduction of scenario sets provides slightly more accurate results compared to backward reduction of single scenarios.

Next we are going to present a new modification of the backward reduction principle. The major difference is to include all deleted scenarios into each backward step simultaneously. Namely, the next index l_i is determined as a solution of the optimization problem

$$l_i \in \arg \min_{l \notin J^{[i-1]}} \sum_{k \in J^{[i-1]} \cup \{l\}} p_k \min_{j \notin J^{[i-1]} \cup \{l\}} c(\omega_k, \omega_j). \tag{16}$$

A more detailed description of the whole algorithm, which will be called *simultaneous backward reduction*, is given in

Algorithm 2.2 (simultaneous backward reduction)

Step 1:

$$\begin{aligned}
& c_{kj} := c(\omega_k, \omega_j)\,,\ k, j = 1, \ldots, N\,, \\
& \textit{Sorting of } \{c_{kj} : j = 1, \ldots, N\}\,,\ k = 1, \ldots, N\,, \\
& c_{ll}^{[1]} := \min_{j \neq l} c_{lj}\,,\ l = 1, \ldots, N\,, \\
& z_l^{[1]} := p_l c_{ll}^{[1]}\,,\ l = 1, \ldots, N\,, \\
& l_1 \in \arg \min_{l \in \{1, \ldots, N\}} z_l^{[1]}\,,\ J^{[1]} := \{l_1\}\,.
\end{aligned}$$

Step i:

$$\begin{aligned}
& c_{kl}^{[i]} := \min_{j \notin J^{[i-1]} \cup \{l\}} c_{kj}\,,\ l \notin J^{[i-1]}\,, k \in J^{[i-1]} \cup \{l\}\,, \\
& z_l^{[i]} := \sum_{k \in J^{[i-1]} \cup \{l\}} p_k c_{kl}^{[i]},\ l \notin J^{[i-1]}\,, \\
& l_i \in \arg \min_{l \notin J^{[i-1]}} z_l^{[i]}\,,\ J^{[i]} := J^{[i-1]} \cup \{l_i\}\,.
\end{aligned}$$

Step N-n+1: *Redistribution by (10).*

Algorithm 2.2 allows the following interpretation. Its first step corresponds to the optimal deletion of only one scenario. For $i > 1$, l_i is chosen such that

$$D_{J^{[i-1]} \cup \{l_i\}} = \min_{l \notin J^{[i-1]}} D_{J^{[i-1]} \cup \{l\}}\,, \tag{17}$$

where $D_{J^{[i-1]} \cup \{l\}}$ is defined in (11). Hence, the index l_i is defined recursively such that the index set $\{l_1, \ldots, l_{i-1}, l_i\}$ is optimal subject to the constraint

that the previous indices $\{l_1, \ldots, l_{i-1}\}$ are fixed.
Since running times are important characteristics of scenario reduction algorithms, we study the computational complexity, i.e., the number of elementary arithmetic operations, of Algorithm 2.2. It is shown in [6] that a proper implementation (without sorting) of backward reduction of scenario sets requires a complexity of $\mathcal{O}(N^2)$ operations which holds uniformly with respect to n. When comparing formulas (14) and (16), one notices an increase of complexity in the cost structure of (16) for determining l_i. More precisely, step i requires the computation of $N-i+1$ sums each consisting of i summands and $N-i+1$ comparisons. Each summand represents a product of two numbers. One of these factors requires about 2 operations for determining the minimum. The sorting process in step 1 requires $\mathcal{O}(N^2 \log N)$ operations ([1], Chapter 1). When excluding the complexity of evaluating the function c and of the redistribution rule, altogether we obtain

$$\begin{aligned} b_N(n) &:= \mathcal{O}(N^2 \log N) + \sum_{i=1}^{N-n} (3i+1)(N-i+1) \qquad (18) \\ &= n^3 - n^2(\frac{3}{2}N + \frac{1}{2}) - n\frac{3}{2}(N+1) + a(N) \\ \text{where} \quad a(N) &:= \frac{N^3}{2} + \mathcal{O}(N^2 \log N) + 2N^2 + \frac{3}{2}N \end{aligned}$$

operations for selecting a subset of n scenarios. Hence, we have

Proposition 2.3 *The computational complexity for reducing a set of $N \in \mathbb{N}$ scenarios to a subset containing $n \in \{1, \ldots, N\}$ scenarios consists of $b_N(n)$ (see (18)) operations when using simultaneous backward reduction.*

Hence, the complexity of simultaneous backward reduction is increasing with decreasing n and is, of course, minimal at $n = N$. This result corresponds to the running time observations of our numerical tests reported in Section 4.
Next, we describe a strategy that is just the opposite of backward reduction. Its conceptual idea is based on formula (13) and consists in the recursive selection of scenarios that will *not* be deleted. The basic concept of such an algorithm is given in [4] and called *forward selection*. Forward selection determines an index set $\{u_1, \ldots, u_n\}$ such that

$$u_i \in \arg \min_{u \in J^{[i-1]}} \sum_{k \in J^{[i-1]} \setminus \{u\}} p_k \min_{j \notin J^{[i-1]} \setminus \{u\}} c(\omega_k, \omega_j), \qquad (19)$$

for $i = 1, \ldots, n$, where $J^{[i-1]} := \{1, \ldots, N\} \setminus \{u_1, \ldots, u_{i-1}\}$. The first step of this procedure coincides with solving problem (13). After the last step, the optimal redistribution rule has to be used to determine the reduced probability measure. Formula (19) allows the same interpretation as in case of simultaneous backward reduction. It is again closely related to the structure of D_J in (11). Now, let us consider the following algorithm, which is easy to implement and is called *fast forward selection*.

Algorithm 2.4 (fast forward selection)

Step 1: $c_{ku}^{[1]} := c(\omega_k, \omega_u)\,,\ k, u = 1, \ldots, N\,,$

$$z_u^{[1]} := \sum_{\substack{k=1 \\ k \neq u}} p_k c_{ku}^{[1]}\,,\ u = 1, \ldots, N\,,$$

$$u_1 \in \arg \min_{u \in \{1,\ldots,N\}} z_u^{[1]}\,,\ J^{[1]} := \{1, \ldots, N\} \backslash \{u_1\}\,.$$

Step i: $c_{ku}^{[i]} := \min\{c_{ku}^{[i-1]}, c_{ku_{i-1}}^{[i-1]}\}\,,\ k, u \in J^{[i-1]}\,,$

$$z_u^{[i]} := \sum_{k \in J^{[i-1]} \backslash \{u\}} p_k c_{ku}^{[i]}\,,\ u \in J^{[i-1]}\,,$$

$$u_i \in \arg \min_{u \in J^{[i-1]}} z_u^{[i]}\,,\ J^{[i]} := J^{[i-1]} \backslash \{u_i\}\,.$$

Step n+1: *Redistribution by (10).*

Theorem 2.5 *The index set $\{u_1, \ldots, u_n\}$ determined by Algorithm 2.4 is a solution of the forward selection principle, i.e., u_i satisfies condition (19) for each $i = 1, \ldots, n$. Furthermore, $z_{u_i}^{[i]} = D_{J^{[i]}}$ holds for each $i = 1, \ldots, n$, where $D_{J^{[i]}}$ is defined in (11).*

Proof: For $i = 1$ the result is immediate. For $i = 2, \ldots, N$, it holds that

$$\begin{aligned}
u_i \in \arg \min_{u \in J^{[i-1]}} z_u^{[i]} &= \arg \min_{u \in J^{[i-1]}} \sum_{k \in J^{[i-1]} \backslash \{u\}} p_k c_{ku}^{[i]} \\
&= \arg \min_{u \in J^{[i-1]}} \sum_{k \in J^{[i-1]} \backslash \{u\}} p_k \min\{c_{ku}^{[i-1]}, c_{ku_{i-1}}^{[i-1]}\} \\
&= \arg \min_{u \in J^{[i-1]}} \sum_{k \in J^{[i-1]} \backslash \{u\}} p_k \min\{c_{ku}^{[i-2]}, c_{ku_{i-1}}^{[i-2]}, c_{ku_{i-2}}^{[i-2]}\} \\
&\ \vdots \\
&= \arg \min_{u \in J^{[i-1]}} \sum_{k \in J^{[i-1]} \backslash \{u\}} p_k \min\{c_{ku}^{[1]}, c_{ku_{i-1}}^{[1]}, \ldots, c_{ku_1}^{[1]}\} \\
&= \arg \min_{u \in J^{[i-1]}} \sum_{k \in J^{[i-1]} \backslash \{u\}} p_k \min_{j \notin J^{[i-1]} \backslash \{u\}} c(\omega_k, \omega_j) \\
&= \arg \min_{u \in J^{[i-1]}} D_{J^{[i-1]} \backslash \{u\}} \qquad (20)
\end{aligned}$$

Hence, the index u_i satisfies condition (19) and it holds that

$$z_{u_i}^{[i]} = \sum_{k \in J^{[i]}} p_k \min_{j \notin J^{[i]}} c(\omega_k, \omega_j) = D_{J^{[i]}} \quad (i = 1, \ldots, n). \qquad \square$$

The conditions (17) and (20) show that both algorithms are based on the same basic idea for selecting the next (scenario) index. The only difference

consists in the use of backward and forward strategies, respectively, i.e., in determining the sets of deleted and remaining scenarios, respectively.
As in the case of backward reduction, the computational complexity of Algorithm 2.4 is of interest. Step i requires $(N-i+1)^2$ operations for computing $c_{ku}^{[i]}$ $(k, u \in J^{[i-1]})$, $(N-i+1)(N-i)$ operations for $z_u^{[i]}$ $(u \in J^{[i-1]})$ and $N-i+1$ operations for determining u_i. Altogether, we obtain

$$f_N(n) := \sum_{i=1}^{n} 2(N-i+1)^2 = \frac{2}{3}n^3 - n^2(2N+1) + n(2N^2+2N+\frac{1}{3}) \quad (21)$$

operations for selecting a subset of n scenarios. Hence, we have

Proposition 2.6 *The computational complexity of fast forward selection for reducing a set of $N \in \mathbb{N}$ scenarios to a subset containing $n \in \{1, \ldots, N\}$ scenarios consists of $f_N(n)$ (see (21)) operations.*

Hence, the complexity of fast forward selection is increasing with increasing n and is maximal if $n = N$. Thus, the use of fast forward selection is recommendable if the number n of remaining scenarios satisfies the condition $f_N(n) \leq b_N(n)$. The number $n_* = n_*(N)$ such that $f_N(n_*) = b_N(n_*)$ holds, is a zero of a polynomial of degree 3 which depends nonlinearly on N. It turns out that $n_* \approx \frac{N}{4}$ for large N.

3 Minimal distances of scenario trees

All algorithms discussed in the previous section provide only approximate solutions of (11) in general. Since error estimates for these algorithms are not available, we need test examples of discrete original and reduced measures of different scale with known (optimal) ζ_c-distances. Because of their practical importance, we consider probability measures with scenarios exhibiting a tree structure. In particular, we derive optimal distances of certain regularly structured original scenario trees and of their reduced trees containing different numbers of scenarios.
We consider a scenario tree that represents a discrete parameter stochastic process with a parameter set $\{0, 1, \ldots, K\}$ for some $K \in \mathbb{N}$ and with scenarios (or paths) branching at each parameter $k \in \{0, 1, \ldots, K\}$ with branching degree d (i.e., each node of the tree has d successors). In case of $d = 2$ and $d = 3$, the tree is called *binary* and *ternary*, respectively. Hence, the tree consists of $N := d^K$ scenarios $\omega_i = (\omega_i^0, \ldots, \omega_i^K)$, $i = 1, \ldots, d^K$, and has $\omega_1^0 = \ldots = \omega_{d^K}^0$ as its root node. Furthermore, we let all scenarios have equal probabilities $p_i = \frac{1}{d^K}, i = 1, \ldots, d^K$. Such a scenario tree is called *regular* if, for each $k \in \{0, \ldots, K\}$, there exist symmetric sets $V_k := \{\delta_1^k, \ldots, \delta_d^k\} \subset \mathbb{R}$ such that

$$\omega_i^k = \sum_{j=0}^{k} \delta_{i_j}^j \quad (k \in \{0, \ldots, K\}), \quad (22)$$

where a $(K+1)$-tuple of indices $(i_0,\ldots,i_K) \in \{1,\ldots,d\}^{K+1}$ corresponds to each index $i = 1,\ldots,d^K$. We say that V_k is symmetric if $\delta \in V_k$ implies $-\delta \in V_k$. In case of $d = 2$ and $d = 3$, this means that the sets V_k are of the form $V_k = \{-\delta^k, \delta^k\}$ and $V_k = \{-\delta^k, 0, \delta^k\}$, respectively, for some $\delta^k \in \mathbb{R}_+$. Clearly, it holds that $\delta_1^0 = \ldots = \delta_d^0 = 0$ for regular trees. Figure 1 shows an example of a regular binary scenario tree with $K = 3$ and $N = 2^3$ scenarios. We specify the function c in (3) by setting $h \equiv 1$ and by choosing the maximum norm $\|\cdot\|_\infty$ on $\mathbb{R}^{K+1}$, i.e.,

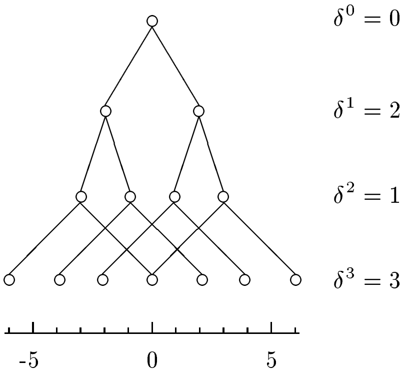

Figure 1: Binary scenario tree

$$c(\omega,\tilde{\omega}) := \|\omega - \tilde{\omega}\|_\infty = \max_{k=0,\ldots,K} |\omega^k - \tilde{\omega}^k| \quad (\omega,\tilde{\omega} \in \Omega).$$

Our first result provides an explicit formula for the minimal distance between a regular binary tree and reduced subtrees with at least $n = \frac{N}{4}$ scenarios.

Proposition 3.1 (3/4-solution)
Let a regular binary scenario tree with $N = 2^K$ scenarios and $K \geq 3$ be given. Let $k_0 \in \arg\min_{1\leq k\leq K} \delta^k$, $k_0 \leq K-2$ and $\max\{\delta^{k_0+1}, \delta^{k_0+2}\} \leq 2\delta^{k_0}$. Then it holds for each $n \in \mathbb{N}$ with $\frac{N}{4} \leq n < N$:

$$D_n^{min} := \min\{D_J : \#J = N-n\} = \frac{N-n}{N} 2\delta^{k_0}. \tag{23}$$

Proof: We use the representation (22) of each scenario ω_i for $i = 1,\ldots,N$. Let $i,j \in \{1,\ldots,N\}$, $i \neq j$, and let $(i_0,\ldots,i_K)$ and $(j_0,\ldots,j_K)$ denote the corresponding $(K+1)$-tuples of indices. Let $l \in \{1,\ldots,K\}$ be such that $i_l \neq j_l$ and $i_r = j_r$ for $r = 0,\ldots,l-1$. Then we obtain

$$\|\omega_i - \omega_j\|_\infty = \max_{k=0,\ldots,K} |\omega_i^k - \omega_j^k| = \max_{k=0,\ldots,K} |\sum_{r=0}^{k} (\delta_{i_r}^r - \delta_{j_r}^r)|$$
$$\geq |\sum_{r=0}^{l} (\delta_{i_r}^r - \delta_{j_r}^r)| = 2\delta^l \geq 2\delta^{k_0}\,.$$

Hence, it holds for each $J \subset \{1,\dots,N\}$ with $\#J = N-n$ that

$$D_J = \sum_{i\in J} p_i \min_{j\notin J} \|\omega_i - \omega_j\|_\infty \geq \sum_{i\in J} \frac{1}{N} 2\delta^{k_0} = \frac{N-n}{N} 2\delta^{k_0} \,.$$

It remains to show that there exists an index set J_* such that $\#J_* = N-n$ and that the lower bound is attained, i.e., $D_{J_*} = \frac{N-n}{N}2\delta^{k_0}$. To this end, we consider the index set

$$I_* := \{i \in \{1,\dots,N\} : \operatorname{sign}(\delta^{k_0}_{i_{k_0}}) = -\operatorname{sign}(\delta^{k_0+1}_{i_{k_0+1}}) = -\operatorname{sign}(\delta^{k_0+2}_{i_{k_0+2}})\}$$

and define $J_* := \{1,\dots,N\} \setminus I_*$. Let Tr_* denote the tree consisting of all scenarios ω_i for $i \in I_*$. Figure 2 illustrates a detail of Tr_* starting at a node at level $k_0 - 1$ and ending at level $k_0 + 2$. Hence, for the cardinality of I_* and J_* we obtain that

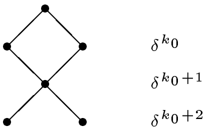

Figure 2: Detail of the subtree Tr_*

$$\#I_* = 2^{k_0-1}\cdot 2 \cdot 2^{K-k_0-2} = \frac{1}{4}2^K = \frac{N}{4} \quad \text{and} \quad \#J_* = N - \#I_* = \frac{3}{4}N \,.$$

Now we want to show that for each $j \in J_*$, there exists an index $i \in I_*$ such that $\|\omega_i - \omega_j\|_\infty = 2\delta^{k_0}$ holds. Let $j \in J_*$ and ω_j be the related scenario. Let us consider the behaviour of ω_j on the branching levels k_0, k_0+1 and k_0+2. Since $j \notin I_*$, we have to distinguish three cases each for $\delta^{k_0}_{j_{k_0}} = \delta^{k_0}$ (resp. $\delta^{k_0}_{j_{k_0}} = -\delta^{k_0}$):

$$\begin{array}{ll}
\text{Case (1):} & \delta^{k_0+1}_{j_{k_0+1}} = {}^{(-)}_{\;+}\delta^{k_0+1} \;\wedge\; \delta^{k_0+2}_{j_{k_0+2}} = {}^{(-)}_{\;+}\delta^{k_0+2} \\
\text{Case (2):} & \delta^{k_0+1}_{j_{k_0+1}} = {}^{(-)}_{\;+}\delta^{k_0+1} \;\wedge\; \delta^{k_0+2}_{j_{k_0+2}} = {}^{(+)}_{\;-}\delta^{k_0+2} \\
\text{Case (3):} & \delta^{k_0+1}_{j_{k_0+1}} = {}^{(+)}_{\;-}\delta^{k_0+1} \;\wedge\; \delta^{k_0+2}_{j_{k_0+2}} = {}^{(-)}_{\;+}\delta^{k_0+2}
\end{array}$$

Now, we consider the following $(K+1)$-tuple $(i_0,\dots,i_K)$ where $i_k = j_k$ for all $k \notin \{k_0, k_0+1, k_0+2\}$ and

$$\delta^{k_0}_{i_{k_0}} = {}^{(+)}_{\;-}\delta^{k_0} \;\wedge\; \delta^{k_0+1}_{i_{k_0+1}} = {}^{(-)}_{\;+}\delta^{k_0+1} \;\wedge\; \delta^{k_0+2}_{i_{k_0+2}} = {}^{(-)}_{\;+}\delta^{k_0+2} \,.$$

Let $i \in \{1,\ldots,d^K\}$ denote the corresponding index. Clearly, $i \in I_*$ and, consequently, it holds for the distance between ω_i and ω_j that

$$
\begin{aligned}
\|\omega_i - \omega_j\|_\infty &= \max_{k=0,\ldots,K} |\sum_{r=0}^{k} (\delta^r_{i_r} - \delta^r_{j_r})| \\
&= \max_{k\in\{k_0,k_0+1,k_0+2\}} |\sum_{r=k_0}^{k} (\delta^r_{i_r} - \delta^r_{j_r})| \\
&= \begin{cases} |2\delta^{k_0}| & \text{, in case (1)} \\ \max\{|2\delta^{k_0}|, |2\delta^{k_0} - 2\delta^{k_0+2}|\} & \text{, in case (2)} \\ \max\{|2\delta^{k_0}|, |2\delta^{k_0} - 2\delta^{k_0+1}|\} & \text{, in case (3)} \end{cases} \\
&= 2\delta^{k_0}\,.
\end{aligned}
$$

The latter equation holds due to the assumption that $\delta^{k_0} \le \max\{\delta^{k_0+1}, \delta^{k_0+2}\} \le 2\delta^{k_0}$. Hence, $D_{J_*} = \frac{\#J_*}{N} 2\delta^{k_0} = \frac{3}{2}\delta^{k_0}$. By considering subsets of J_* having cardinality in $[1, \frac{3}{4}N]$, the result follows for the general case, too. □

The second result provides a similar formula for the minimal distance between a regular ternary tree and reduced subtrees containing $n \ge \frac{2}{9}N$ scenarios.

Proposition 3.2 (7/9-solution)
Let a regular ternary scenario tree with $N = 3^K$ scenarios and $K \ge 3$ be given. Let $k_0 \in \arg\min_{1\le k\le K} \delta^k$ with $k_0 \le K-2$, $\max\{\delta^{k_0+1}, \delta^{k_0+2}\} \le 2\delta^{k_0}$. Then it holds for each $n \in \mathbb{N}$ with $\frac{2}{9}N \le n < N$:

$$D_n^{min} = \min\{D_J : \#J = N-n\} = \frac{N-n}{N}\delta^{k_0}. \tag{24}$$

Proof: Similarly as in Proposition 3.1 we obtain

$$\|\omega_i - \omega_j\|_\infty \ge \delta^{k_0}$$

for all $i, j \in \{1,\ldots,N\}$, $i \ne j$, and, hence,

$$D_J = \sum_{i\in J} p_i \min_{j\notin J} \|\omega_i - \omega_j\|_\infty \ge \frac{N-n}{N}\delta^{k_0}$$

for each subset J of $\{1,\ldots,N\}$ with $\#J = N-n$. Again we have to show that there exists an index set J_{**} such that $\#J_{**} = N-n$ and that the lower bound $\frac{N-n}{N}\delta^{k_0}$ is attained with $D_{J_{**}}$. We consider the index set

$$
\begin{aligned}
I_{**} := \{i \in \{1,\ldots,N\} \,:\, &(\delta^{k_0}_{i_{k_0}} = 0 \,\wedge\, \delta^{k_0+1}_{i_{k_0+1}} \ne 0 \,\wedge\, \delta^{k_0+2}_{i_{k_0+2}} \ne 0) \\
&\vee (\delta^{k_0}_{i_{k_0}} \ne 0 \,\wedge\, \delta^{k_0+1}_{i_{k_0+1}} = 0 \,\wedge\, \delta^{k_0+2}_{i_{k_0+2}} = 0)\}
\end{aligned}
$$

and define $J_{**} := \{1,\ldots,N\}\backslash I_{**}$. Let Tr_{**} denote the tree consisting of all scenarios ω_i for $i \in I_{**}$. Figure 3 illustrates a detail of Tr_{**} starting at a

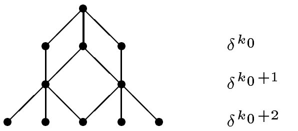

Figure 3: Detail of the subtree Tr_{**}

node at level $k_0 - 1$ and ending at level $k_0 + 2$. We obtain for the cardinality of I_{**} and J_{**} that

$$\#I_{**} = 3^{k_0-1} \cdot 6 \cdot 3^{K-k_0-2} = \frac{2}{9}3^K = \frac{2}{9}N \text{ and } \#J_{**} = N - \#I_{**} = \frac{7}{9}N .$$

Similarly as in Proposition 3.1 it can be shown that for each $j \in J_{**}$, there exists an index $i \in I_{**}$ such that it holds that $\|\omega_i - \omega_j\|_\infty = \delta^{k_0}$. Hence, $D_{J_{**}} = \frac{\#J_{**}}{N}\delta^{k_0} = \frac{7}{9}\delta^{k_0}$. By considering subsets of J_{**} having cardinality in $[1, \frac{7}{9}N]$, the result follows for the general case, too. □

Similar results are available under additional assumptions in case of the Euclidean norm instead of the maximum norm (see also [6]).

4 Numerical Results

The aim of this section is to report on numerical experience of testing and comparing the algorithms described in Section 2, namely, *backward reduction of scenario sets*, *simultaneous backward reduction*, *fast forward selection*. All algorithms were implemented in *C*. The test runs were performed on an HP 9000 (780/J280) Compute-Server with 180 MHz frequency and 768 MByte main memory under HP-UX 10.20, i.e., the same configuration as for the numerical tests in [4]. We consider the situation where the function c is defined by $c(\omega, \tilde{\omega}) := \|\omega - \tilde{\omega}\|_\infty$ $(\forall \omega, \tilde{\omega} \in \Omega)$ and the original discrete probability measure P is given in scenario tree form. More precisely, we use a test battery of three binary and ternary scenario trees, respectively. All test trees are regular and, thus, the results of Section 3 apply. They provide minimal (Fortet-Mourier) distances of P to reduced measures supported by n scenarios when n is not too small.

Example 4.1 (binary scenario tree)
Let $K = 10$, $d = 2$, $N = 2^{10} = 1024$, $p_i = \frac{1}{N}$, $i = 1, \ldots, N$, *and* $(\delta^1, \ldots, \delta^{10}) = (0.5, 0.6, 0.7, 0.9, 1.1, 1.3, 1.6, 1.9, 2.3, 2.7)$. *Figure 4 illustrates the original scenario tree. Proposition 3.1 applies with* $k_0 = 1$ *and* $D_n^{min} = \frac{N-n}{N}$ *holds for each* $\frac{N}{4} = 256 \le n < N$.

Example 4.2 (ternary scenario tree)
Let $K = 6$, $d = 3$, $N = 3^6 = 729$, $p_i = \frac{1}{N}$, $i = 1, \ldots, N$, *and* $(\delta^1, \ldots, \delta^6) = (0.7, 0.9, 1.2, 1.5, 2.6, 3.3)$. *The tree is shown in Figure 5. Proposition 3.2 applies with* $k_0 = 1$ *and* $D_n^{min} = 0.7\frac{N-n}{N}$ *holds for each* $\frac{2N}{9} = 162 \le n < N$.

Example 4.3 (ternary load scenario tree)
We consider the scenario tree construction in Section 4 of [4] for the weekly electrical load process of a German power utility (see also [5,8] for a description of a stochastic power management model and its solution by Lagrangian relaxation). The original construction is based on an hourly discretization of the weekly time horizon with branching points at $t_k = 24k$ for $k = 1, \ldots, 6$, and on a piecewise linear interpolation between the t_k. The corresponding mean shifted tree is illustrated in Figure 6. For a moment, we disregard all non-branching points of the time discretization and consider the corresponding mean shifted tree. The latter tree is a regular ternary scenario tree with $K = 6$, $N = 3^6 = 729$, $p_i = \frac{1}{N}$ for $i = 1, \ldots, N$ and $\delta^k = \sigma_{t_k}\sqrt{\frac{3}{2^{8-k+1}}}$ for $k = 1, \ldots, 6$, where σ_t denotes the standard deviation of the stochastic load process at time t. Since in this case σ_t increases with increasing t, Proposition 3.2 applies with $k_0 = 1$ and it holds that $D_n^{min} = \delta^1\frac{N-n}{N}$ for $\frac{2N}{9} = 162 \leq n < N$. Finally, it remains to remark that, due to the piecewise linear structure of the scenarios and the choice of the maximum norm for defining c, the minimal distance D_n^{min} does not change when including all non-branching points.

The original scenario trees of the Examples 4.1–4.3 were reduced to trees containing n scenarios by using all 3 reduction algorithms. The corresponding tables contain the relative accuracy and the running time of each algorithm needed to produce a reduced tree with n scenarios. In addition, the tables provide the (relative) lower bound (15) and the (relative) minimal distance D_n^{min} in percent if available. Here, "relative" always means that the corresponding quantity is divided by the minimal ζ_c-distance of P and one of its scenarios endowed with unit mass. In particular, the relative accuracy is defined as the quotient of the ζ_c-distance of the original measure P and the reduced measure Q_n (having n scenarios) and of the ζ_c-distance of P and the measure $\delta_{\omega_{i^*}}$, i.e.,

$$\zeta_c^{rel}(P, Q_n) := \frac{\zeta_c(P, Q_n)}{\zeta_c(P, \delta_{\omega_{i^*}})}, \tag{25}$$

where $\{\omega_i\}_{i=1,\ldots,N}$ denotes the set of scenarios of P and ω_{i^*} is defined by

$$\zeta_c(P, \delta_{\omega_{i^*}}) = \min\{D_J : \#J = N-1\} = \min_{i\in\{1,\ldots,N\}} \zeta_c(P, \delta_{\omega_i}). \tag{26}$$

Our numerical experience shows that all algorithms work reasonably well. All algorithms reduce 50% of the scenarios of P in an optimal way. As expected, simultaneous backward reduction and fast forward selection produce more accurate trees than backward reduction of scenario sets at the expense of higher running times. Our results also indicate that fast forward selection is slightly more accurate than simultaneous backward reduction, although both backward reduction variants are sometimes competitive. Fast forward selection works much faster than the implementation of forward selection in

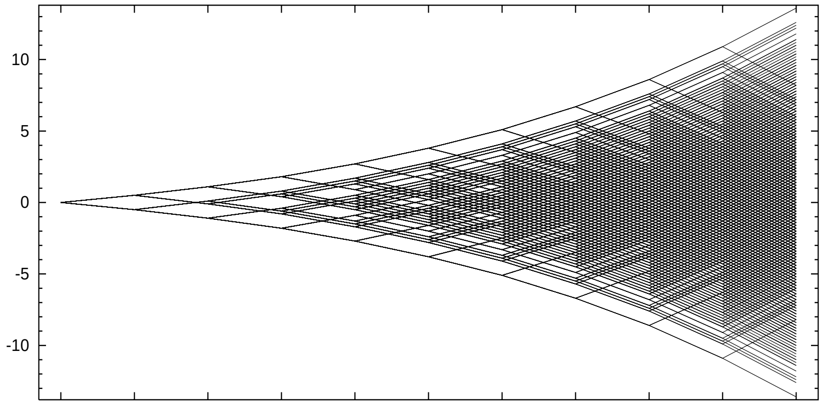

Figure 4: Original binary scenario tree

Number	**Backward of**		**Simultaneous**		**Fast**		Lower	Minimal
n of	**Scenario Sets**		**Backward**		**Forward**		Bound	Distance
Scenarios	ζ_c^{rel}	Time	ζ_c^{rel}	Time	ζ_c^{rel}	Time		
1	116.01 %	2 s	111.93 %	96 s	100.00 %	2 s	19.01 %	100.00 %
2	102.86 %	2 s	75.45 %	96 s	79.16 %	2 s	18.99 %	*
3	78.54 %	2 s	66.54 %	96 s	63.96 %	2 s	18.97 %	*
4	66.35 %	2 s	61.69 %	96 s	59.04 %	3 s	18.95 %	*
5	64.81 %	2 s	57.95 %	96 s	54.51 %	3 s	18.92 %	*
10	53.68 %	2 s	48.21 %	95 s	44.39 %	4 s	18.81 %	*
20	39.16 %	2 s	40.15 %	95 s	35.84 %	7 s	18.59 %	*
30	35.61 %	2 s	34.70 %	94 s	31.56 %	10 s	18.37 %	*
50	31.55 %	2 s	29.11 %	93 s	26.75 %	15 s	17.93 %	*
100	22.68 %	2 s	21.73 %	89 s	20.97 %	27 s	16.98 %	*
150	18.48 %	2 s	18.16 %	85 s	18.02 %	38 s	16.06 %	*
200	16.70 %	2 s	16.50 %	81 s	16.11 %	48 s	15.14 %	*
250	15.23 %	2 s	15.21 %	76 s	14.55 %	56 s	14.22 %	*
260	14.97 %	2 s	14.97 %	75 s	14.26 %	58 s	14.04 %	14.04 %
270	14.75 %	2 s	14.75 %	74 s	14.00 %	60 s	13.86 %	13.86 %
280	14.53 %	2 s	14.53 %	72 s	13.76 %	61 s	13.67 %	13.67 %
290	14.30 %	2 s	14.30 %	71 s	13.54 %	63 s	13.49 %	13.49 %
300	14.08 %	2 s	14.08 %	70 s	13.32 %	64 s	13.30 %	13.30 %
350	12.98 %	2 s	12.98 %	64 s	12.39 %	71 s	12.39 %	12.39 %
400	11.88 %	2 s	11.88 %	57 s	11.47 %	76 s	11.47 %	11.47 %
450	10.78 %	2 s	10.78 %	51 s	10.55 %	81 s	10.55 %	10.55 %
500	9.67 %	2 s	9.67 %	45 s	9.63 %	85 s	9.63 %	9.63 %
600	7.79 %	2 s	7.79 %	33 s	7.79 %	91 s	7.79 %	7.79 %
700	5.95 %	2 s	5.95 %	22 s	5.95 %	95 s	5.95 %	5.95 %
800	4.12 %	2 s	4.12 %	12 s	4.12 %	97 s	4.12 %	4.12 %

Table 1: Results of binary scenario tree reduction

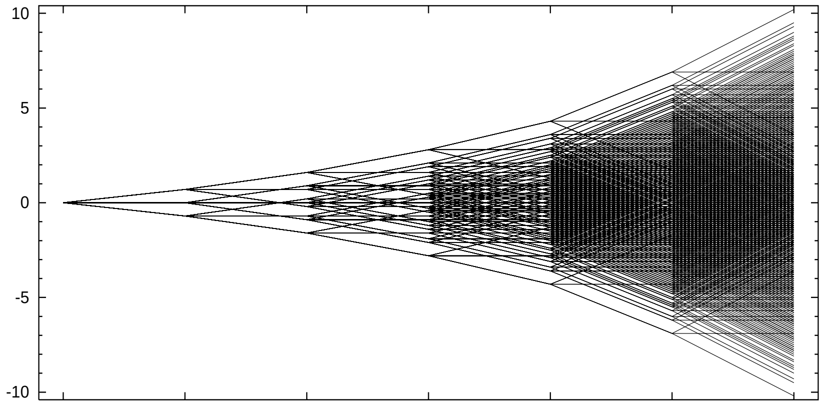

Figure 5: Original ternary scenario tree

Number	**Backward of**		**Simultaneous**		**Fast**		Lower	Minimal
n of	**Scenario Sets**		**Backward**		**Forward**		Bound	Distance
Scenarios	ζ_c^{rel}	Time	ζ_c^{rel}	Time	ζ_c^{rel}	Time		
1	164.68 %	1 s	164.68 %	32 s	100.00 %	1 s	18.66 %	100.00 %
2	93.02 %	1 s	89.29 %	32 s	80.70 %	1 s	18.63 %	*
3	72.84 %	1 s	69.77 %	32 s	61.40 %	1 s	18.60 %	*
4	56.27 %	1 s	56.27 %	32 s	56.59 %	1 s	18.56 %	*
5	53.56 %	1 s	53.56 %	31 s	51.78 %	1 s	18.53 %	*
6	50.85 %	1 s	50.85 %	31 s	49.26 %	1 s	18.50 %	*
10	45.27 %	1 s	44.69 %	31 s	41.78 %	2 s	18.37 %	*
15	39.72 %	1 s	38.83 %	31 s	36.09 %	3 s	18.20 %	*
20	33.92 %	1 s	34.74 %	31 s	32.67 %	3 s	18.06 %	*
30	30.22 %	1 s	30.74 %	31 s	28.41 %	5 s	17.77 %	*
40	27.20 %	1 s	27.56 %	31 s	25.63 %	6 s	17.50 %	*
50	25.05 %	1 s	25.04 %	30 s	23.44 %	7 s	17.25 %	*
100	18.48 %	1 s	17.58 %	29 s	17.88 %	13 s	15.98 %	*
150	15.38 %	1 s	15.33 %	26 s	15.25 %	18 s	14.71 %	*
162	14.99 %	1 s	14.89 %	26 s	14.74 %	19 s	14.40 %	14.40 %
200	13.75 %	1 s	13.62 %	24 s	13.52 %	22 s	13.44 %	13.44 %
220	13.10 %	1 s	13.01 %	23 s	12.94 %	24 s	12.93 %	12.93 %
230	12.77 %	1 s	12.72 %	22 s	12.68 %	24 s	12.68 %	12.68 %
240	12.44 %	1 s	12.43 %	22 s	12.42 %	25 s	12.42 %	12.42 %
250	12.17 %	1 s	12.17 %	21 s	12.17 %	26 s	12.17 %	12.17 %
300	10.90 %	1 s	10.90 %	18 s	10.90 %	28 s	10.90 %	10.90 %
350	9.63 %	1 s	9.63 %	15 s	9.63 %	31 s	9.63 %	9.63 %
400	8.36 %	1 s	8.36 %	12 s	8.36 %	32 s	8.36 %	8.36 %
500	5.82 %	1 s	5.82 %	7 s	5.82 %	34 s	5.82 %	5.82 %
600	3.28 %	1 s	3.28 %	3 s	3.28 %	35 s	3.28 %	3.28 %

Table 2: Results of ternary scenario tree reduction

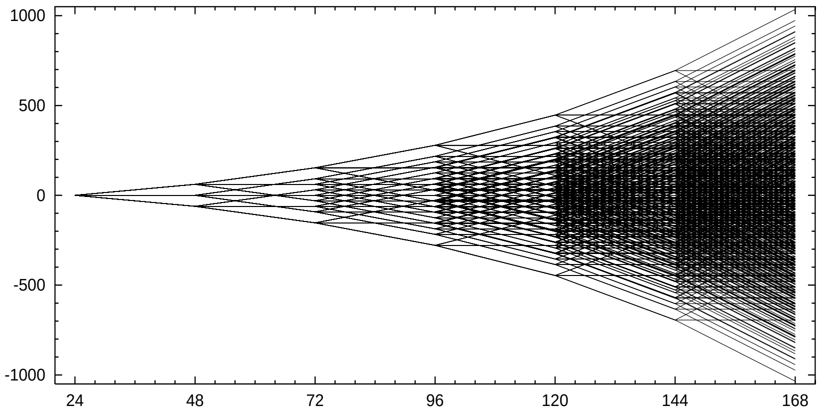

Figure 6: Original load scenario tree

Number	**Backward of**		**Simultaneous**		**Fast**		Lower	Minimal
n of	**Scenario Sets**		**Backward**		**Forward**		Bound	Distance
Scenarios	ζ_c^{rel}	Time	ζ_c^{rel}	Time	ζ_c^{rel}	Time		
1	121.09 %	1 s	117.85 %	31 s	100.00 %	1 s	16.31 %	100.00 %
2	98.80 %	1 s	90.19 %	31 s	80.83 %	1 s	16.28 %	*
3	75.88 %	1 s	72.25 %	31 s	61.65 %	1 s	16.24 %	*
4	73.75 %	1 s	59.71 %	31 s	56.94 %	1 s	16.21 %	*
5	62.04 %	1 s	55.45 %	31 s	52.22 %	1 s	16.18 %	*
6	56.57 %	1 s	52.24 %	31 s	49.57 %	1 s	16.14 %	*
10	46.86 %	1 s	45.20 %	31 s	41.93 %	2 s	16.01 %	*
15	39.69 %	1 s	40.22 %	30 s	35.76 %	3 s	15.85 %	*
20	35.16 %	1 s	36.75 %	30 s	32.32 %	3 s	15.69 %	*
30	30.08 %	1 s	31.20 %	30 s	28.11 %	5 s	15.36 %	*
40	27.77 %	1 s	27.74 %	30 s	25.25 %	6 s	15.13 %	*
50	25.58 %	1 s	25.13 %	29 s	23.02 %	7 s	14.90 %	*
100	19.52 %	1 s	17.31 %	28 s	16.86 %	13 s	13.76 %	*
150	14.52 %	1 s	13.96 %	25 s	13.67 %	18 s	12.67 %	*
162	13.29 %	1 s	13.26 %	25 s	13.15 %	19 s	12.40 %	12.40 %
200	12.04 %	1 s	11.77 %	23 s	11.74 %	22 s	11.57 %	11.57 %
220	11.39 %	1 s	11.16 %	22 s	11.18 %	24 s	11.13 %	11.13 %
230	11.06 %	1 s	10.93 %	22 s	10.95 %	24 s	10.91 %	10.91 %
240	10.73 %	1 s	10.70 %	21 s	10.72 %	25 s	10.70 %	10.70 %
250	10.48 %	1 s	10.48 %	21 s	10.49 %	26 s	10.48 %	10.48 %
300	9.38 %	1 s	9.38 %	18 s	9.38 %	28 s	9.38 %	9.38 %
350	8.29 %	1 s	8.29 %	15 s	8.29 %	31 s	8.29 %	8.29 %
400	7.20 %	1 s	7.20 %	12 s	7.20 %	32 s	7.20 %	7.20 %
500	5.01 %	1 s	5.01 %	7 s	5.01 %	34 s	5.01 %	5.01 %
600	2.82 %	1 s	2.82 %	3 s	2.82 %	35 s	2.82 %	2.82 %

Table 3: Results of load scenario tree reduction

[4]. For instance, fast forward selection required 35 seconds to determine a load scenario subtree (Example 4.3) containing 600 scenarios instead of 8149 seconds reported in [4]. Especially, in the case of deeply reduced trees, fast forward selection works very fast and accurately.
Furthermore, it turned out that the lower bound is very good (even optimal) for large n, but extremely pessimistic for small n. Another observation is that the reduction of half of the scenarios implies only a loss of about 10% of the relative accuracy. For instance, in case of Example 4.2 it is possible to determine a subtree containing just 6 out of the originally 729 scenarios that still carries about 50% of the relative accuracy.
Finally, we take a closer look at the numerical results of the load scenario tree reduction. In particular, we compare the running times of simultaneous backward reduction and fast forward selection in this case. Figure 7 displays the running times of both algorithms and shows clearly their opposite algorithmic strategies. It reflects the corresponding theoretical complexity results (Propositions 2.3 and 2.6) and shows that the running time of fast forward selection is smaller if $n \leq \frac{N}{4}$ (approximately). This confirms again that the forward selection concept is favourable if n is small. Figures 8, 9 and 10 show the reduced load trees with 15 scenarios obtained by all algorithms. The figures display the scenarios with line width proportional to scenario probabilities.

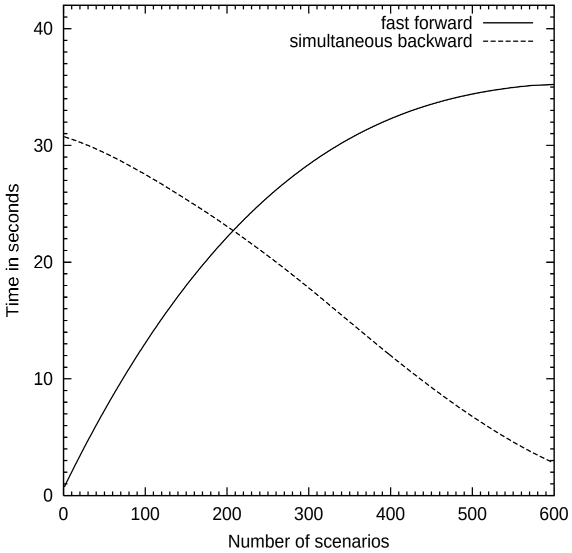

Figure 7: Running time for reducing the load scenario tree

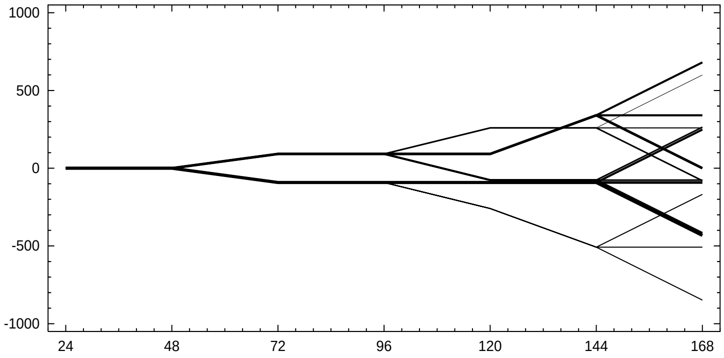

Figure 8: Backward reduction / load tree

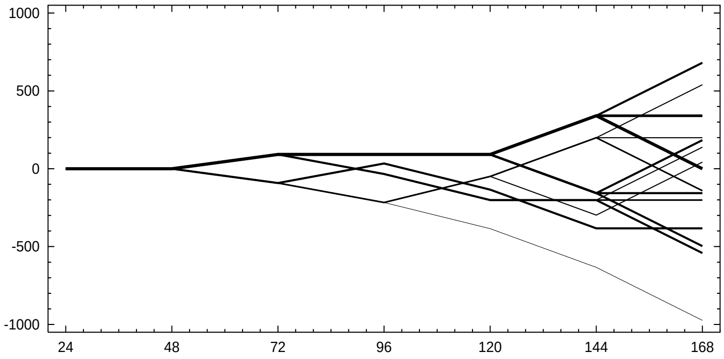

Figure 9: Simultaneous backward reduction / load tree

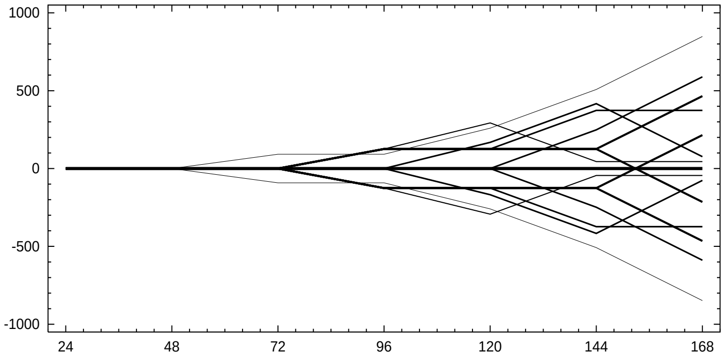

Figure 10: Fast forward selection / load tree

Acknowledgement
This research was partially supported by the BMBF project 03-ROM5B3.

Appendix

Proof (Theorem 2.1): Let $J \subset I := \{1, \ldots, N\}$ be an arbitrary index set. We set $c_{ij} := c(\omega_i, \omega_j)$ for $i, j \in \{1, \ldots, N\}$ and $\bar{q}_j := 0$ for $j \in J$. Linear programming duality implies for any feasible q

$$\begin{aligned} D(J;q) &= \max\{\sum_{i=1}^{N} p_i u_i - \sum_{j=1}^{N} q_j u_j : u_i - u_j \le c_{ij}, \forall i,j \in I\} \\ &= \min\{\sum_{i,j=1}^{N} c_{ij}\eta_{ij} : \eta_{ij} \ge 0\,, \sum_{j=1}^{N} \eta_{ij} - \sum_{k=1}^{N} \eta_{ki} = p_i - q_i\,, \forall i \in I\}\,. \end{aligned}$$

In particular, we consider $\bar{\eta}_{ij} := \begin{cases} p_i & , j = j(i)\,, \, i \in J, \\ p_i & , i = j \notin J, \\ 0 & , \text{otherwise.} \end{cases}$

It holds $\bar{q}_i = \sum\limits_{k=1}^{N} \bar{\eta}_{ki}$ and $p_i = \sum\limits_{j=1}^{N} \bar{\eta}_{ij}$ for each $i \in I$. Hence, we obtain

$$D(J;\bar{q}) \le \sum_{i,j} c_{ij}\bar{\eta}_{ij} = \sum_{i \in J} p_i \min_{j \notin J} c_{ij}\,.$$

Next, we set $u_i := \min\limits_{k \notin J} \|\omega_i - \omega_k\|$ for each $i \in I$. Noting that $u_i = 0$, for any $i \notin J$, we obtain $u_i - u_j = u_i \le c_{ij}$ for all $i \in J$ and $j \notin J$, $u_i - u_j \le 0 \le c_{ij}$ for all $i \notin J$, $j \in I$, and $u_i - u_j \le \min\limits_{k \notin J} \|\omega_i - \omega_k\| - \min\limits_{k \notin J} \|\omega_j - \omega_k\| \le \|\omega_i - \omega_j\| \le c_{ij}$ for all $i, j \in J$. Hence, we obtain for any feasible q that

$$\begin{aligned} D(J;q) &\ge \sum_{i \in J} p_i u_i \ge \sum_{i \in J} p_i \min_{j \notin J} \|\omega_i - \omega_j\| \\ &\ge \frac{1}{C} \sum_{i \in J} p_i \min_{j \notin J} \max\{1, h(\|\omega_i - \omega_0\|), h(\|\omega_j - \omega_0\|)\} \|\omega_i - \omega_j\| \\ &= \frac{1}{C} \sum_{i \in J} p_i \min_{j \notin J} c(\omega_i, \omega_j), \end{aligned}$$

and the proof is complete. □

References

1. T.H. Cormen, C.E. Leiserson and R.L. Rivest, Introduction to Algorithms, Massachusetts Institute of Technology, 1990.
2. J. Dupačová, Portfolio optimization via stochastic programming: Methods of output analysis, *Mathematical Methods of Operations Research* 5 (1999) 245-270.

3. J. Dupačová, G. Consigli an S.W. Wallace, Scenarios for multistage stochastic programs, *Annals of Operations Research* 100 (2001), to appear.
4. J. Dupačová, N. Gröwe-Kuska and W. Römisch, Scenario reduction in stochastic programming: An approach using probability metrics, Preprint 00-9, Institut für Mathematik, Humboldt-Universität Berlin, 2000, revised 2001 and submitted to *Mathematical Programming.*
5. N. Gröwe-Kuska, K.C. Kiwiel, M.P. Nowak, W. Römisch and I. Wegner, Power management in a hydro-thermal system under uncertainty by Lagrangian relaxation, Preprint 99-19, Institut für Mathematik, Humboldt-Universität Berlin, 1999 and to appear in the IMA Volume *Decision Making under Uncertainty: Energy and Environmental Models*, Springer-Verlag.
6. H. Heitsch, Reduktion von Szenariobäumen in der stochastischen Optimierung, Diplomarbeit, Institut für Mathematik, Humboldt-Universität Berlin, 2001.
7. G.P. Nemhauser and L.A. Wolsey, Integer Programming, in: *Handbooks in Operations Research and Management Science*, Vol. 1, Optimization (G.L. Nemhauser, A.H.G. Rinnooy Kan, M.J. Todd eds.), North-Holland, Amsterdam 1989, 447–527.
8. M.P. Nowak and W. Römisch, Stochastic Lagrangian relaxation applied to power scheduling in a hydro-thermal system under uncertainty, *Annals of Operations Research* 100 (2001), to appear.
9. G. Ch. Pflug, Scenario tree generation for multiperiod financial optimization by optimal discretization, *Mathematical Programming*, Series B, 89 (2001), 251–271.
10. S.T. Rachev, *Probability Metrics and the Stability of Stochastic Models*, Wiley, Chichester 1991.
11. S.T. Rachev and W. Römisch, Quantitative stability in stochastic programming: The method of probability metrics, Preprint 00-22, Humboldt-Universität Berlin, Institut für Mathematik, 2000.
12. S.T. Rachev and L. Rüschendorf, *Mass Transportation Problems*, Vol. I and II, Springer, Berlin 1998.
13. W. Römisch and R. Schultz, Stability analysis for stochastic programs, *Annals of Operations Research* 30 (1991), 241-266.
14. V.M. Zolotarev, Probability metrics, *Theory of Probability and its Applications* 28 (1983), 278–302.