A Branch-and-Bound Method for Multistage Stochastic Integer Programs with Risk Objectives

Thomas Heinze* and Rüdiger Schultz†
Department of Mathematics
University of Duisburg-Essen, Campus Duisburg
Lotharstr. 65, D-47048 Duisburg, Germany

Abstract

We identify multistage stochastic integer programs with risk objectives where the related wait-and-see problems enjoy similar separability as in the risk neutral case. For models belonging to this class we present a solution method combining branch-and-bound with relaxation of nonanticipativity and constraint branching along nonanticipativity subspaces.

Key Words. Stochastic integer programming, multistage models, mean-risk models, mixed-integer optimization.

AMS subject classifications. 90C15, 90C11, 90C06.

1 Introduction

Multistage stochastic programs arise as deterministic equivalents to random optimization problems where uncertainty is unveiled stepwise with intermediate decisions that must not anticipate future information. In the traditional setting ([4, 11, 18, 22]), the objective function of a multistage stochastic program is based on taking the expected value of a suitable term, reflecting costs in a minimization setting, for instance.

More recently, risk aversion has gained considerable interest in stochastic programming, see for instance [7, 8, 17, 23, 24, 25]. The objective function then no longer is based on the expectation alone. For instance, a weighted sum of the expectation and a suitable measure expressing risk is taken. For the latter, a wide variety of statistical parameters comes into play such that passing to mean-risk stochastic programs raises many research questions, from principal model setup to induced structures and algorithmic possibilities.

Compared with two-stage models, multistage stochastic programs involve more complex dynamics and require more elaborate treatment of nonanticipativity. In general, this makes them more demanding computationally. It is well known that both two- and multistage risk neutral stochastic programs turn into large-scale block structured optimization problems if the underlying probability distributions are discrete. When including risk aversion it is observed that the selection of the risk measure has a considerable impact on the decomposability of the block structure arising. For two-stage linear stochastic programs this was studied systematically in [14, 23, 24, 25].

The desirable property of the large-scale mixed-integer linear program (MILP) that represents the stochastic program then is that passing to the wait-and-see model, i.e., relaxing the nonanticipativity constraints, fully decouples the model into subproblems for each of the individual realizations (scenarios) of the discrete probability distribution. This decoupling occurs with risk measures such as the Expected Excess of a given target, the Excess Probability, and the Conditional Value-at-Risk. For other risk measures, such as the Semideviation or the Value-at-Risk, internal coupling is tighter such that relaxing nonanticipativity is insufficient to arrive at single-scenario subproblems.

*heinze@math.uni-duisburg.de
†schultz@math.uni-duisburg.de, corresponding author

The scenario decomposition method, proposed in [5] for two-stage models with risk neutral objective and extended to certain mean-risk two-stage models in [14, 24, 25], handles the above principal setup in the framework of Lagrangian relaxation. For multistage models this approach is valid only theoretically. The quality of lower bounding in Lagrangian relaxation (in a minimization context) critically depends on the ability to solve the nonsmooth concave Lagrangian dual. The dimension of the latter, however, grows with the number of scenarios, the number of stages, and the dimension of the nonanticipative variables. It becomes critical already for complex two-stage models such that it makes sense to look for alternatives in the multistage situation.

Furthermore, the quality of upper bounding in Lagrangian relaxation critically depends on obtaining "promising" feasible points for the original problem, mostly on the basis of the results of the dual optimization. For two-stage models nonanticipativity is just a single identity among the first-stage variables, across all individual scenarios. This immediately gives rise to heuristics for "promising" points. In the multistage case there is a whole system of identities corresponding to time stages and connecting subsets of scenarios. This makes respective heuristics a far more complex matter.

In the present paper we develop a solution method for multistage linear mixed-integer stochastic programs involving risk aversion. We will identify multiperiod risk measures whose incorporation into multistage stochastic programs leads to separability of the corresponding wait-and-see model. This provides the basis for a branch-and-bound algorithm involving relaxation of nonanticipativity. Regaining the latter is accomplished in the course of the branching, and can be seen as constraint branching along nonanticipativity subspaces.

Our method bears similarity with the approach of [1, 2]. The authors consider risk neutral two- and multistage pure-integer stochastic programs. Beside nonanticipativity they relax integrality and arrive at single-scenario *linear* programs where we are to tackle single-scenario *mixed-integer linear* programs.

In [13], see also the survey [26], a branch-and-price approach to solving mixed-integer (risk neutral) multistage stochastic linear programs is taken. Lower bounding is achieved by solving associated LP relaxations by means of column generation.

Another principal possibility to solve multistage mixed-integer stochastic programs with risk aversion is to exploit inherent problem separability: In many practical applications the initial random mixed-integer linear program involves physical or abstract entities coupled only mildly by respective constraints. The resulting stochastic program then often can be tackled efficiently by Lagrangian relaxation of the coupling constraints, also called geographical or component decomposition [6, 8, 16, 21].

Our paper is organized as follows. In Section 2 we collect prerequisites about risk neutral multistage stochastic integer programs and about single- as well as multiperiod risk measures. In Section 3 we analyze separability of wait-and-see problems. Section 4 is devoted to the presentation of our algorithm, and first computational results are reported in Section 5.

2 Multistage Stochastic Integer Programs

2.1 Basics and Expectation Model

Consider a finite sequential decision process under uncertainty where the decision $x_t \in \mathbb{R}^{n_t}$ at stage $t \in \{1, \ldots, T\}$ is made according to information available up to time t only. Information occurs as a discrete time stochastic data process $\xi = (\xi_t)_{t=1}^T$ on some probability space $(\Omega, \mathcal{F}, \mathbb{P})$ with values in $\times_{t=1}^T \mathbb{R}^{m_t}$. By $\mathcal{F}_t = \sigma(\xi_1, \ldots, \xi_t), t = 1, \ldots, T$, we denote the sigma algebra generated by the random vector $(\xi_1, \ldots, \xi_t)$. We assume that ξ_1 is deterministic, i.e., $\mathcal{F}_1 = \{\emptyset, \Omega\}$, and that $\mathcal{F}_T = \mathcal{F}$. Clearly, $\mathcal{F}_t \subseteq \mathcal{F}_{t+1}$ for all $t = 1, \ldots, T-1$. Nonanticipativity, i.e., the requirement that x_t is independent on future information, then coincides with the measurability of x_t with respect to $\mathcal{F}_t$. Using conditional expectations, the latter can be expressed as $H_t(x_t) := x_t - \mathbb{E}[x_t|\mathcal{F}_t] = 0$.

We require that $x = (x_t)_{t=1}^T$ has to fulfill the constraints

$$\begin{aligned}
&x_t \in X_t, \\
&H_t(x_t) = 0, \\
&B_t(\xi_t)x_t \leq d_t(\xi_t), \\
&\sum_{\tau=0}^{t-1} A_{t,\tau}(\xi_t)x_{t-\tau} = h_t(\xi_t), \quad t = 1, \ldots, T,
\end{aligned}$$

and that there are costs $b_t(\xi_t)^\top x_t$ for $t = 1, \ldots, T$.

With closed sets X_t whose convex hulls are polyhedra, the first group above models simple fixed constraints. In particular, this covers integrality requirements to components of x. The second group models nonanticipativity. The third and fourth groups are further coupling constraints within and dynamic constraints interlinking time steps, respectively. Cost coeffcients $b_t(.)$, right-hand sides $d_t(.), h_t(.)$, and matrices $A_{t,\tau}(.), B_t(.)$ are supposed to have suitable dimensions and to depend affinely linearly on ξ_t, for all relevant t and τ.

The decisions x_t are understood as members of function spaces $L_p(\Omega, \mathcal{F}, I\!\!P; I\!\!R^{n_t}), p \in [1, \infty], t = 1, \ldots, T$. Constraints from the first, third, and fourth groups then are to be understood pointwise and to hold $I\!\!P$-almost surely. The constraints from the second group impose functional conditions, in fact, membership in a suitable linear subspace of $\times_{t=1}^{T} L_p(\Omega, \mathcal{F}, I\!\!P; I\!\!R^{n_t})$. By $\mathcal{X}(\xi)$ we denote the set of all feasible $x \in \times_{t=1}^{T} L_p(\Omega, \mathcal{F}, I\!\!P; I\!\!R^{n_t})$. Later on, the accent will be on algorithmic issues in conjunction with finite discrete ξ. Hence, $p = \infty$ can tacitly be assumed right away.

Any feasible $x \in \mathcal{X}(\xi)$ thus induces random cost variables

$$z_t := \sum_{\tau=1}^{t} b_\tau(\xi_\tau)^\top x_\tau, \quad t = 1, \ldots, T \tag{1}$$

of which z_1 is deterministic since ξ_1 is so. (For convenience, the dependence of z_t on x will be dropped in the notation throughout the text.) These random variables, which represent accumulated costs over time, can be inspected from different viewpoints: In case only the final costs matter, z_T is the essential object. If intermediate cost monitoring is an issue, then the whole vector $(z_1, \ldots, z_T)$ must be taken into account. Another possibility is to judge by incremental rather than accumulated costs, in which case the differences $z_{\tau+1} - z_\tau, \tau = 1, \ldots, T-1$ become relevant, see also the discussion in [8]. These different views reflect a major conceptual difference between multistage and two-stage problems. For the latter, the only random cost object of interest is $z_T = z_2$.

Which view ever taken, solving the above multistage stochastic program, i.e., finding a “best” $x \in \mathcal{X}(\xi)$ amounts to finding a “best” member in the family of random vectors $\{(z_1, \ldots, z_T) : x \in \mathcal{X}(\xi)\}$ or a related family derived from one of the above views. The specification of “best” then is accomplished by comparing random vectors with the help of statistical parameters. In risk neutral multistage stochastic programs the relevant parameter is the expectation $I\!\!E$ that is applied to the total accumulated costs z_T, leading to the expectation model

$$\min\{I\!\!E[z_T] \ : \ x \in \mathcal{X}(\xi)\}. \tag{2}$$

2.2 Risk Measures

For incorporating risk into multistage stochastic programs statistical parameters, called risk measures, are applied to the random variables $z_1, \ldots, z_T$ from Subsection 2.1. The design of risk measures itself is a field of active research whose review is beyond the scope of the present paper. For more detailed coverage see [3, 8, 9, 15] and the references therein.

Above we had seen that in a multistage stochastic program a vector of random cost outcomes governs the objective while in the two-stage counterpart a real-valued random variable takes this place. Accordingly, the subsequent account on risk measures distinguishes one-period and multiperiod risk measures. The concept of polyhedral risk measures introduced in [8] proves very flexible in modelling important multiperiod aspects and in deriving structural results. Later on we will see that stochastic programs with polyhedral risk objectives form a substantial class amenable to the algorithmic treatment we propose.

A risk measure ρ is understood as a functional from some set of real valued random variables to the real numbers (one-period risk measure) or as a functional from some set of random vectors to the real numbers (multiperiod risk measure).

Definition 2.1 *(One-Period Polyhedral Risk Measure, [8])*
A risk measure ρ on $L_p(\Omega, \mathcal{F}, I\!\!P)$ with some $p \in [1, \infty]$ is called polyhedral if there exist $k_1, k_2 \in I\!\!N$, $c_1, w_1 \in I\!\!R^{k_1}, c_2, w_2 \in I\!\!R^{k_2}$, a nonempty polyhedron $Y_1 \subseteq I\!\!R^{k_1}$, and a polyhedral cone $Y_2 \subseteq I\!\!R^{k_2}$ such that for every $z \in L_p(\Omega, \mathcal{F}, I\!\!P)$

$$\begin{aligned} \rho(z) = \inf\{c_1^\top y_1 + I\!\!E[c_2^\top y_2] \quad : \quad & y_1 \in Y_1, \\ & y_2 \in L_p(\Omega, \mathcal{F}, I\!\!P),\ y_2 \in Y_2, \\ & w_1^\top y_1 + w_2^\top y_2 = -z\}. \end{aligned} \tag{3}$$

This definition slightly differs from the one in [8] by having $-z$ instead of z in the right-hand side of the constraints. The reason is that in the relevant chapters of [8] bigger outcomes of z are preferred to smaller ones. In the present paper this is just the other way around, due to the minimization setting we impose.

As pointed out in [8], Definition 2.1 can easily accommodate the mean-risk setup where the risk of z is measured by a weighted sum of $\mathbb{E}[z]$ and some $\rho(z)$ fulfilling (3). Indeed, with $\mu \in [0,1]$ the weighted sum $\hat{\rho} := \mu\rho + (1-\mu)\mathbb{E}$ conforms with (3) when setting $\hat{c}_1 := \mu c_1 - (1-\mu)w_1$, $\hat{c}_2 := \mu c_2 - (1-\mu)w_2$, $\hat{w}_1 := w_1$, $\hat{w}_2 := w_2$.

A polyhedral risk measure can be seen as the optimal value of a two-stage stochastic program with random right-hand side. The term "polyhedral" is motivated by the fact that for finite cardinality $\#\Omega$ the above ρ is a polyhedral function on $\mathbb{R}^{\#\Omega}$. Definition 2.1 covers well-known risk measures as the following examples show. Again there are slight differences to formulae in [8] due to our preference of smaller outcomes over bigger ones.

Example 2.2 *(Expected Excess)*
The Expected Excess of a random variable z of a given target $\eta \in \mathbb{R}$ is given by $EE_\eta(z) := \mathbb{E}[(z-\eta)^+]$, where $a^+ := \max\{a, 0\}$. This notion fits into (3) by setting $k_1 := 1$, $k_2 := 2$, $c_1 := 0$, $c_2 := (1,0)$, $w_1 := -1$, $w_2 := (-1,1)$, $Y_1 := \{\eta\}$, $Y_2 := \mathbb{R}^2_+$.

Example 2.3 *(Conditional Value-at-Risk)*
With some probability $\alpha \in]0,1[$ the α-Conditional Value-at-Risk (α-CVaR) $CVaR_\alpha(z)$ of a random variable z is given by

$$CVaR_\alpha(z) := \inf_{\eta \in \mathbb{R}} \Big\{\eta + \frac{1}{1-\alpha}\mathbb{E}[(z-\eta)^+]\Big\}.$$

This fits into (3) by setting $k_1 := 1$, $k_2 := 2$, $c_1 := 1$, $c_2 := (\frac{1}{1-\alpha}, 0)$, $w_1 := -1$, $w_2 := (-1,1)$, $Y_1 := \mathbb{R}$, $Y_2 := \mathbb{R}^2_+$.

For two-stage stochastic integer programs structural and algorithmic consequences of including the Expected Excess or the α-CVaR into the objective are studied in [25]. In particular, scenario decomposition can be accommodated with little effort to solve the resulting mean-risk problems. Another risk measure that has been studied in the context of two-stage stochastic integer programs is the probability $\mathbb{P}[z > \eta]$ of exceeding a given target $\eta \in \mathbb{R}$, see [24]. Although no representation as a polyhedral risk measure is known for the Excess Probability, scenario decomposition again can be accommodated. The Semideviation $SD(z) := \mathbb{E}\big[(z - \mathbb{E}[z])^+\big]$ is an example for a risk measure where scenario decomposition cannot be accommodated directly, since after relaxation of nonanticipativity some coupling over scenarios still persists in the two-stage model. Lower bounding by Expected Excess functionals with properly selected targets, however, then enables utilization of scenario decomposition in an approximative manner, see [14].

The transition to multiperiod polyhedral risk measures is accomplished by passing in (3) from a two-stage to a multistage stochastic program:

Definition 2.4 *(Multiperiod Polyhedral Risk Measure, [8])*
A multiperiod risk measure ρ on $\times_{t=1}^T L_p(\Omega, \mathcal{F}_t, \mathbb{P})$ with $p \in [1,\infty]$ is called multiperiod polyhedral if there exist $k_t \in \mathbb{N}, c_t \in \mathbb{R}^{k_t}, t = 1,\dots,T, w_{t,\tau} \in \mathbb{R}^{k_{t-\tau}}, t = 1,\dots,T, \tau = 0,\dots,t-1$, a polyhedron $Y_1 \subseteq \mathbb{R}^{k_1}$, and polyhedral cones $Y_t \subseteq \mathbb{R}^{k_t}, t = 2,\dots,T$, such that

$$\begin{aligned} \rho(z) = \inf \Big\{ \mathbb{E}\Big[\sum_{t=1}^T c_t^\top y_t\Big] \quad : \quad & y_t \in L_p(\Omega, \mathcal{F}, \mathbb{P}; \mathbb{R}^{k_t}), \\ & H_t(y_t) = 0,\ y_t \in Y_t, \\ & \sum_{\tau=0}^{t-1} w_{t,\tau}^\top y_{t-\tau} = -z_t,\ \ t = 1,\dots,T \Big\}. \end{aligned} \tag{4}$$

In [8] several specifications of (4) are discussed that lead to multiperiod risk measures of different complexity. Among others, it is shown that the setting is rich enough for including the filtration $\{\mathcal{F}_t\}_{t=1}^T$, i.e., the information flow over time, into the definition of the risk measure. Here we quote a relatively simple example that later on will be picked up in our computational experiments.

Example 2.5 *(Multiperiod CVaR, [8])*
With nonnegative weights $\gamma_2, \ldots, \gamma_T$ and confidence levels $\alpha_2, \ldots, \alpha_T \in]0,1[$ we sum up the CVaR in the individual time stages:

$$\rho(z) := \sum_{t=2}^{T} \gamma_t CVaR_{\alpha_t}(z_t). \tag{5}$$

The following holds

$$\begin{aligned}
\rho(z) &= \sum_{t=2}^{T} \gamma_t \inf_{\eta_t \in I\!R} \Big\{ \eta_t + \frac{1}{1-\alpha_t} I\!E[(z_t - \eta_t)^+] \Big\} \\
&= \inf_{(\eta_2,\ldots,\eta_T) \in I\!R^{T-1}} \left\{ \sum_{t=2}^{T} \gamma_t \Big(\eta_t + \frac{1}{1-\alpha_t} I\!E[(z_t - \eta_t)^+] \Big) \right\} \\
&= \inf \left\{ \sum_{t=2}^{T} \gamma_t \Big(\eta_t + \frac{1}{1-\alpha_t} I\!E\big[y_t^{(1)}\big] \Big) \; : \; -\eta_t - y_t^{(1)} + y_t^{(2)} = -z_t, \, y_t^{(1)} \geq 0, \, y_t^{(2)} \geq 0, \right. \\
&\qquad\qquad \left. H_t(y_t) = 0, \, t = 2, \ldots, T, \, (\eta_2, \ldots, \eta_T) \in I\!R^{T-1} \right\} \\
&= \inf \Big\{ I\!E\big[\sum_{t=1}^{T} c_t^\top y_t \big] \; : \; y_t \in Y_t, \, H_t(y_t) = 0, \, \sum_{\tau=0}^{t-1} w_{t,\tau}^\top y_{t-\tau} = -z_t, \, t = 1, \ldots, T \Big\}
\end{aligned} \tag{6}$$

with $k_1 = T$, $k_t = 2, t = 2, \ldots, T$, $c_1 = (0, \gamma_2, \ldots, \gamma_T)$, $c_t = (\frac{\gamma_t}{1-\alpha_t}, 0), t = 2, \ldots, T$, $w_{1,0} = e_1$, $w_{t,0} = (-1,1), t = 2, \ldots, T$, $w_{t,t-1} = -e_t, t = 2, \ldots, T$, $w_{t,\tau} = 0, \tau = 1, \ldots, t-2, t = 3, \ldots, T$, $Y_1 = I\!R^T$, $Y_t = I\!R_+^2, t = 2, \ldots, T$, with e_t denoting the t-th canonical basis vector in $I\!R^T$.

Now we are in the position to formulate the following multistage stochastic integer program with multiperiod polyhedral risk objective. The random cost objects are given by (1), and the multiperiod risk measure is as in Definition 2.4:

$$\min \Big\{ \rho(z_1, \ldots, z_T) = \rho\Big(b_1(\xi_1)^\top x_1, b_1(\xi_1)^\top x_1 + b_2(\xi_2)^\top x_2, \ldots, \sum_{\tau=1}^{T} b_\tau(\xi_\tau)^\top x_\tau \Big) \; : \; x \in \mathcal{X}(\xi) \Big\}. \tag{7}$$

Since the polyhedral risk measure is given by a multistage stochastic program with expectation objective this suggests to consider the following expectation-based multistage stochastic program:

$$\begin{aligned}
\min \Big\{ I\!E\Big[\sum_{t=1}^{T} c_t^\top y_t \Big] \quad : \quad & x \in \mathcal{X}(\xi), \\
& H_t(y_t) = 0, \; y_t \in Y_t, \\
& \sum_{\tau=0}^{t-1} w_{t,\tau}^\top y_{t-\tau} + \sum_{\tau=1}^{t} b_\tau(\xi_\tau)^\top x_\tau = 0, \quad t = 1, \ldots, T \Big\}.
\end{aligned} \tag{8}$$

In [8] the following equivalence between these models is established:

Proposition 2.6 *([8], Proposition 4.1)*
Minimizing (7) with respect to x is equivalent to minimizing (8) with respect to all pairs (x, y) in the following sense: The optimal values of both problems coincide, and (x^*, y^*) is a solution to (8) iff x^* solves (7) and y^* is a solution to the minimization problem defining $\rho\Big(\big(\sum_{\tau=1}^{t} b_\tau(\xi_\tau)^\top x_\tau^* \big)_{t=1}^{T} \Big)$.

3 Separable Wait-and-See Problems

The optimization problem that arises when dropping nonanticipativity from a stochastic program usually is called the wait-and-see problem [4, 11, 18, 22].

Lemma 3.1
Relaxation of nonanticipativity in the multistage stochastic program (8) leads to a wait-and-see problem that is separable in $\omega \in \Omega$.

Proof. Without nonanticipativity there are no constraints in (8) containing components of $\xi(.)$ with distinct arguments ω, ω'. This allows an interchange of integration and minimization, see for instance Theorem 14.60 in [19], and the wait-and-see problem reads

$$\begin{aligned} \mathbb{E}\Big[\min\Big\{ \sum_{t=1}^{T} c_t^\top y_t \quad : \quad & x_t \in X_t,\ y_t \in Y_t, \\ & B_t(\xi_t) x_t \le d_t(\xi_t), \\ & \sum_{\tau=0}^{t-1} A_{t,\tau}(\xi_t) x_{t-\tau} = h_t(\xi_t), \\ & \sum_{\tau=0}^{t-1} w_{t,\tau}^\top y_{t-\tau} + \sum_{\tau=1}^{t} b_\tau(\xi_\tau)^\top x_\tau = 0, \quad t = 1, \ldots, T \Big\} \Big]. \end{aligned}$$

The minimization, hence, can be carried out separately for each $\omega \in \Omega$. □

The above result is not surprising since the separability is well known from traditional expectation-based multistage stochastic programs such that the very fact that problems with multiperiod polyhedral risk objective are representable as expectation problems (Proposition 2.6) already settles the issue. But how about this separability with other risk measures ? Consider the following multiperiod extension of the semideviation

$$SD^{mp}(z) = \sum_{t=2}^{T} \gamma_t SD(z_t), \quad \text{with} \quad \gamma_t \ge 0,\ t = 2, \ldots, T. \tag{9}$$

To see that here the desired separability fails to hold, we put $\gamma_2 = \ldots = \gamma_{T-1} = 0, \gamma_T = 1$ and consider the problem

$$\min\big\{ SD(z_T)\ :\ x \in \mathcal{X}(\xi) \big\}.$$

With a new variable $\theta_T \in \mathbb{R}$ (for which in view of $\mathcal{F}_T = \mathcal{F}$ nonanticipativity trivially holds) the above problem is equivalent to

$$\min\big\{ \mathbb{E}[\theta_T] \quad : \quad x \in \mathcal{X}(\xi),\ \theta_T \ge 0,\ \theta_T \ge z_T - \mathbb{E}[z_T] \big\}.$$

The constraint $\theta_T \ge z_T - \mathbb{E}[z_T]$ then generates coupling among distinct $\omega \in \Omega$, and this coupling persists under relaxation of nonanticipativity.
The mentioned importance of problem separability in ω later in the paper will concern lower bounding. So instead of a separable risk objective a separable (and hopefully not too loose) lower bound might do. The following lemma identifies such a bound for (9).

Lemma 3.2
If $\eta_t \le \mathbb{E}[z_t]$, $t = 2, \ldots, T$, then $0 \ \le\ EE_{\eta_t}[z_t] - \mathbb{E}[z_t] + \eta_t \ \le\ SD(z_t)$, $t = 2, \ldots, T$. The first inequality holds strictly,if and only if $\mathbb{P}[\eta_t > z_t] > 0$.

Proof. Without restriction on η_t it holds that $\max\{z_t, \eta_t\} \ge z_t$, and hence $\max\{z_t - \eta_t, 0\} + \eta_t \ge z_t$. Taking expectations yields the first inequality. With $\eta_t \le \mathbb{E}[z_t]$ one obtains $\max\{z_t, \eta_t\} \le \max\{z_t, \mathbb{E}[z_t]\}$. Hence $\max\{z_t - \eta_t, 0\} + \eta_t \le \max\{z_t - \mathbb{E}[z_t], 0\} + \mathbb{E}[z_t]$, and the second inequality is verified by again taking expectations. The first inequality holds strictly if and only if $0 < \mathbb{E}\big[\max\{\eta_t - z_t, 0\}\big]$, i.e., if and only if $\mathbb{P}[\eta_t > z_t] > 0$.. □

Recall that Expected Excess is a polyhedral risk measure. So the lower bound above, indeed, is separable in ω. The lemma also says that the bound is never worse the trivial bound 0, and strictly better as soon as outcomes of z_t fall below the target η_t with positive probability. Recall further that z_t also depends on x, cf. (1). So the lemma must be understood pointwise in x. Since passing to a wait-and-see problem can only enlarge the feasible set, the following provides a feasible choice of η_t for all relevant x

$$\eta_t^{ws} := \mathbb{E}\big[\min\{z_t \quad : \quad x \in \mathcal{X}(\xi),\ \text{with nonanticipativity dropped}\}\big], \quad t = 2, \ldots, T.$$

We conclude this section with a multiperiod extension of the one-period Excess Probability studied in [24]. No representation as a polyhedral risk measure is known in this case, but still the desired separability of the wait-and-see relaxation holds.

Example 3.3
Consider the following multiperiod extension of $EP_\eta(z) := I\!P[z > \eta]$

$$EP^{mp}(z) := \sum_{t=2}^{T} \gamma_t EP_{\eta_t}(z_t), \tag{10}$$

with weight factors $\gamma_t \geq 0$ and target levels $\eta_t \in I\!R$, $t = 2, \ldots, T$. For the problem

$$\min\{EP^{mp}(z) \ : \ x \in \mathcal{X}(\xi)\}$$

we assume that all sets $\{z_t \ : \ x \in \mathcal{X}(\xi)\}$, $t = 2, \ldots, T$ are bounded above $I\!P$-almost surely. Then there exists a constant $M > 0$ such that the problem is equivalent with

$$\begin{aligned} \min\Big\{ I\!E\Big[\sum_{t=2}^{T} \gamma_t u_t\Big] \quad : \quad & x \in \mathcal{X}(\xi), \\ & z_t - \eta_t \ \leq \ M u_t, \ u_t \in \{0,1\}, \ H(u_t) = 0, \ t = 2, \ldots, T\Big\}. \end{aligned} \tag{11}$$

Dropping nonanticipativity in the above model obviously creates separability in ω.

4 Algorithm

Numerical treatment of multistage stochastic programs usually requires discrete probability distributions, since otherwise the multivariate integrals involved become intractable. Resorting to discrete distributions can be justified by stability results for stochastic programs with perturbed distributions, see for instance [8, 20, 21].

So let $\Omega = \{\omega_1, \ldots, \omega_S\}$, $\pi_s := I\!P(\{\omega_s\})$, $\xi^s := \xi(\omega_s)$, $s = 1, \ldots, S$. The multistage stochastic programs from Sections 2 and 3 formulated there as infinite dimensional optimization problems then become finite dimensional. The sigma algebra $\mathcal{F}$ now is the power set 2^Ω of Ω. To each subalgebra $\mathcal{F}_t$, $t = 1, \ldots, T$, there exists a finite family $\mathcal{E}_t \subseteq 2^\Omega$ that forms a partition of Ω and generates $\mathcal{F}_t$. Since $\mathcal{F}_t \subseteq \mathcal{F}_{t+1}$, every element of $\mathcal{E}_t$ is the union of elements in $\mathcal{E}_{t+1}$. The number of elements in $\mathcal{E}_t$ coincides with the number of different elements among $(\xi^1_\tau)^t_{\tau=1}, \ldots, (\xi^S_\tau)^t_{\tau=1}$, the realizations of ξ up to time t. The relations between the elements of $\mathcal{E}_t$ and $\mathcal{E}_{t+1}$ for $t = 1, \ldots, T-1$ can be represented by a tree, called scenario tree. The nodes of this tree occur in layers for $t = 1, \ldots, T$, with each node corresponding to an element of $\mathcal{E}_t$ for some $t \in \{1, \ldots, T\}$. Arcs only exist between nodes in adjacent layers. Each node (element) in $\mathcal{E}_t$ is connected with all those nodes (elements) in $\mathcal{E}_{t+1}$ whose union makes up $\mathcal{E}_t$. The scenarios $\xi^s = (\xi^s_\tau)^T_{\tau=1}$, $s = 1, \ldots, S$, then correspond to the maximal paths in the scenario tree.

In terms of the scenario tree, nonanticipativity of the decision $\mathsf{x} = \big(\mathsf{x}^s\big)^S_{s=1} = \big(\mathsf{x}(\omega_s)\big)^S_{s=1}$ says that components of x^s and $\mathsf{x}^{s'}$ must attain the same values as long as the paths corresponding to ξ^s and $\xi^{s'}$ coincide. More precisely, behind the notation $H_t(\mathsf{x}_t) = 0$, $t = 1, \ldots, T$, that was introduced in Section 2 now there is the following system of linear equations

$$\text{for all } t = 1, \ldots, T: \quad \mathsf{x}^s_t = \mathsf{x}^{s'}_t \ \text{ for all } s, s' \in \{1, \ldots, S\} \text{ for which } \xi^s_\tau = \xi^{s'}_\tau, \ \tau = 1, \ldots, t. \tag{12}$$

Now we are in the position to formulate the multistage stochastic integer programs our solution method will apply to. Recall that in addition to the traditional model (2) we had seen the (expectation) models (8), (11) as equivalents to models with risk objectives. All these models can be obtained as specifations of the following multistage stochastic program:

$$\begin{aligned} \min\Big\{ \sum_{s=1}^{S} \pi_s \sum_{t=1}^{T} \mathsf{b}_t(\xi^s_t)^\top \mathsf{x}^s_t \quad : \quad & H_t(\mathsf{x}_t) = 0, \\ & \mathsf{x}^s_t \in \mathsf{X}_t, \end{aligned}$$

$$\left.\begin{aligned}
&\mathsf{B}_t(\xi_t^s)\mathsf{x}_t^s \ \le\ \mathsf{d}_t(\xi_t^s),\\
&\sum_{\tau=1}^{t-1}\mathsf{A}_{t,\tau}(\xi_t^s)\mathsf{x}_{t-\tau}^s \ =\ \mathsf{h}_t(\xi_t^s),\\
&\qquad s=1,\dots,S,\ \ t=1,\dots,T
\end{aligned}\right\} \tag{13}$$

Relaxing nonanticipativity yields

$$\min\left\{\sum_{s=1}^{S}\pi_s\sum_{t=1}^{T}\mathsf{b}_t(\xi_t^s)^\top\mathsf{x}_t^s \ :\ \begin{aligned}
&\mathsf{x}_t^s\in\mathsf{X}_t,\\
&\mathsf{B}_t(\xi_t^s)\mathsf{x}_t^s \ \le\ \mathsf{d}_t(\xi_t^s),\\
&\sum_{\tau=1}^{t-1}\mathsf{A}_{t,\tau}(\xi_t^s)\mathsf{x}_{t-\tau}^s \ =\ \mathsf{h}_t(\xi_t^s),\\
&\qquad s=1,\dots,S,\ \ t=1,\dots,T
\end{aligned}\right\}, \tag{14}$$

and, by the separability discussed in Section 3, this model is equivalent to

$$\sum_{s=1}^{S}\pi_s\cdot\min\left\{\sum_{t=1}^{T}\mathsf{b}_t(\xi_t^s)^\top\mathsf{x}_t^s \ :\ \begin{aligned}
&\mathsf{x}_t^s\in\mathsf{X}_t,\\
&\mathsf{B}_t(\xi_t^s)\mathsf{x}_t^s \ \le\ \mathsf{d}_t(\xi_t^s),\\
&\sum_{\tau=1}^{t-1}\mathsf{A}_{t,\tau}(\xi_t^s)\mathsf{x}_{t-\tau}^s \ =\ \mathsf{h}_t(\xi_t^s),\ \ t=1,\dots,T
\end{aligned}\right\}. \tag{15}$$

Our algorithm rests on understanding (13) as a nonconvex global minimization problem to be tackled by branch-and-bound. The feasible set will be partitioned using nonanticipativity subspaces. On elements of the partition, relaxations of type (14) will provide lower bounds for the optimal objective values. Upper bounds are obtained from solutions to (14) meeting the relaxed nonanticipativity, or by suitable heuristics.

We assume that (13) is bounded below. The basic scheme of the algorithm then is as follows.

Algorithm

Step 0: (Initialization)
$\mathbf{P} := \{(14)\}$ - a list of problems; $GUB := +\infty$ - a global upper bound for the optimal value; $INC := void$ - a currently best solution; $\varepsilon > 0$ - a tolerance parameter for termination; $\delta > 0$ - a tolerance parameter for branching; $LB((14)) := void$ - a lower bound for a member of $\mathbf{P}$.

Step 1: (Termination)
If $\mathbf{P} := \emptyset$ or $\frac{GUB-\min_{P\in\mathbf{P}} LB(P)}{1+|\min_{P\in\mathbf{P}} LB(P)|} < \varepsilon$, then stop; INC is optimal.

Step 2: (Problem Selection and Solution)
Choose $P \in \mathbf{P}$ and delete it from $\mathbf{P}$. Solve P.

Step 3: (Pruning by Infeasibility)
If the feasible region of P is empty, then go to Step 1. Otherwise, set $\varphi(P)$ to the optimal value and $\mathsf{x}(P)$ to an optimal solution of P.

Step 4: (Pruning by Inferiority)
If $\varphi(P) \ge GUB$, then go to Step 1.

Step 5: (Pruning by Optimality)
If $\mathsf{x}(P)$ fulfils the relaxed nonanticipativity, then delete from $\mathbf{P}$ all problems P with $LB(P) \ge \varphi(P)$. If

$\varphi(P) < GUB$, then set $GUB := \varphi(P)$ and $INC := \mathsf{x}(P)$. Go to Step 1.

Step 6: (Feasibility Heuristic)
Apply a heuristic starting out from $\mathsf{x}(P)$ to obtain a feasible point $\bar{\mathsf{x}}(P)$ for (13). If the objective function value $\bar{\varphi}(P)$ of the latter fulfils $\bar{\varphi}(P) < GUB$, then set $GUB := \bar{\varphi}(P)$ and $INC := \bar{\mathsf{x}}(P)$.

Step 7: (Branching)
Select a component $\mathsf{x}(P)^s_{t,(i)}$ of $\mathsf{x}(P)$ violating nonanticipativity. Let $\mathcal{G}_t$ denote the (unique) element of $\mathcal{E}_t$ (to be identified with a node in layer t of the scenario tree) such that the path corresponding to s passes through $\mathcal{G}_t$.

(i) If $\mathsf{x}^s_{t,(i)}$ is an integer variable, then create two new subproblems P_1 and P_2 by adding to P, for all s' whose path in the scenario tree passes through $\mathcal{G}_t$, the constraints

$$\mathsf{x}^{s'}_{t,(i)} \;\leq\; \mathsf{x}(P)^s_{t,(i)}, \qquad \text{and} \qquad \mathsf{x}^{s'}_{t,(i)} \;\geq\; \mathsf{x}(P)^s_{t,(i)} + 1,$$

respectively.

(ii) If $\mathsf{x}^s_{t,(i)}$ is a continuous variable, then create two new subproblems P_1 and P_2 by adding to P, for all s' whose path in the scenario tree passes through $\mathcal{G}_t$, the constraints

$$\mathsf{x}^{s'}_{t,(i)} \;\leq\; \mathsf{x}(P)^s_{t,(i)}, \qquad \text{and} \qquad \mathsf{x}^{s'}_{t,(i)} \;\geq\; \mathsf{x}(P)^s_{t,(i)} + \delta,$$

respectively.

Set $LB(P_1) := LB(P_2) := \varphi(P)$ and add P_1, P_2 to $\mathbf{P}$. Go to Step 1.

The algorithm's two main features are that, in Step 2, problem solution is accomplished scenario-wise and that, in Step 7, branching is carried out on groups of variables rather than on individual variables.
Initially (Step 0), nonanticipativity is relaxed completely, leading to (14) which, by (15), amounts to a collection of single-scenario problems for $s = 1, \ldots, S$. The algorithm works with single-scenario problems, only. These are updated in Step 7 where nonanticipativity is re-established step by step. The separability of (15) then is maintained since creation of new problems in Step 7 does not lead to coupling across scenarios.
Our implementation involves a book-keeping of single-scenario problems solved in the course of the algorithm. Problem modification in Step 7 typically involves only subsets of scenarios. Therefore, single-scenario problems not modified in Step 7 bear the potential to be re-used in Step 2 when solving subsequent problems from $\mathbf{P}$. Our implementation employs ILOG CPLEX 8.1 [10] for solving the single-scenario mixed-integer linear programs.
Step 7 can be seen as constraint branching on nonanticipativity conditions. When branching on continuous variables, the tolerance δ is used to avoid endless refinements. The nonanticipativity test at the beginning of Step 7 has to be understood accordingly: components with distance less than δ are considered coinciding.
With the tolerance δ, finite termination of the algorithm is granted (in theory) if, for instance, the feasible set of (13) is bounded. In practical computations, however, the sheer dimension of problem instances usually will require to enforce termination of the algorithm by limiting execution time or another relevant quantity.
The above formulation of the algorithm leaves room for various specifications. In our implementation problem selection in Step 2 follows the least-lower-bound rule: a problem P with minimum $LB(P)$ is chosen.
For the heuristic in Step 6 the user first can specify a priority order of the variables, for instance according to inner hierarchies in the model, or following the stages $t = 1, \ldots, T$. It turned out useful in this respect to rank model variables induced by the risk measure, such as η_t in Example 2.5, with lowest priority. The heuristic then proceeeds with fixing parts of the variables according to the priority order and solve reduced problems in the remaining non-fixed variables. Step 6 itself is optional, and can optionally be repeated several times in one loop of the full algorithm.
The priority order from Step 6 is also employed for the selection of the component $\mathsf{x}^s_{t,(i)}$ in Step 7. Conceptually, the selection of this component bears some similarity with the selection of the branching

variable in traditional LP based branch-and-bound for integer programs. Namely, in the latter a variable violating integrality is selected. Here, a variable violating nonanticipativity takes the place. Our selection then follows a priority rule. An alternative could be to mimic the "most fractional variable rule" from integer programming, and pick a "most anticipative variable", i.e., one with maximum dispersion across scenarios (with some prescribed dispersion measure).
In the above formulation of Step 7 the variable range is split at $\mathsf{x}(P)^s_{t,(i)}$. In our implementation other specifications of this "split point", such as the arithmetic mean of $\mathsf{x}(P)^s_{t,(i)}, s = 1, \ldots, S$ (rounded to integer) are used as well.

5 Computations

We have tested our algorithm at multistage multiknapsack problems and at multistage extensions of two-stage models from power optimization introduced in [12].
Two different book-keeping strategies for single-scenario problems were employed in our computations, one based on a tree structure (bb-tree) and the other involving a hash table (bb-hash). In the following we outline these procedures.
Tree Structure: We associate a binary tree where each node corresponds to a complete collection of single-scenario problems (for $s = 1, \ldots, S$). After each loop through Step 7 of the algorithm a new layer is added to the tree. Edges connect nodes in adjacent layers only. Adding a layer means to create two children nodes for each node in the layer that is currently deepest. The children nodes contain the information about the modification of single-scenario problems in Step 7. More precisely, for scenarios s' whose path in the scenario tree passes through $\mathcal{G}_t$ the children nodes point to the single-scenario problems modified according to the branching rules in Step 7. Since branching is binary and uniform for all such s', exactly two children nodes are created for each parent node.
To each layer of this book-keeping tree there corresponds some $\mathcal{G}_t$, i.e., a node in the scenario tree. From the construction it is clear that, for all scenarios whose paths in the scenario tree do not pass through this node, the single-scenario problems in the nodes of the respective layer of the book-keeping tree must be identical. So when arriving at a problem P in Step 2 of the algorithm, the corresponding node in the book-keeping tree can be detected, together with the mentioned identical single-scenario problems of the same layer. Those already solved before then do not have to be tackled again.
For scenarios $\bar{s}$ whose path in the scenario tree passes through the node behind $\mathcal{G}_t$ we distinguish two cases. After having identified the node in the book-keeping tree corresponding to P the search "climbs up" to the parent node in the book-keeping tree. Either the path in the scenario tree belonging to $\bar{s}$ passes through the node behind the $\mathcal{G}_t$ of the parent layer, or not.

- If not, then the parent node contains an "ancestor" of the $\bar{s}$-problem in P, and the parent layer contains identical "mates" of this "ancestor" (unless at root level). The search then proceeds by "climbing down" from these "mates" to the next layer until the $\bar{s}$-problem is found.
- If yes, then the search is continued in another structure. This structure consists of individual trees for the scenarios $s = 1, \ldots, S$. The nodes of tree $s \in \{1, \ldots, S\}$ correspond to the modifications of the single-scenario problem s in the course of the branching in Step 7 of the algorithm. Each branching in a single-scenario problem (parent) creates two children nodes in the tree. Moreover, single-scenario problems already solved are marked as such, with pointers to their optimal solutions. The $\bar{s}$-problem then is looked up in the tree for $\bar{s}$, and tackled only if necessary.

This completes the conceptual description of our tree-based book-keeping strategy. In our implementation, the book-keeping tree does not occur explicitly. Rather, only those parts of the tree which are relevant in a specific search are made explicit. This is done on the basis of the following (actually stored) information: the $\mathcal{G}_t$ corresponding to the loops through Step 7 of the algorithm, and the above mentioned individual trees for $s = 1, \ldots, S$ reflecting the modifications of the single-scenario problems.

Hash Table: The idea is to store in a hash table all single-scenario problems that were solved in the course of the algorithm (together with their optimal solutions), and, when in Step 2 of the algorithm, to look up in the table whether solutions of relevant single-scenario problems can be retrieved directly. As hash function we used a weighted sum of the scenario number s and the number of previous branchings of single-scenario problem s, and this sum taken modulo M, where M is the number of addresses of the hash table. For tie breaking in case of identical addresses a lexicography is employed. This lexicograpphy involves the scenario number, the number of previous branchings of the single-scenario problem, the variable-indices, encodings of the branching directions ($\leq$ vs. $\geq$) in Step 7, and the values of the "split points" in Step 7.

Table 1 reports our computational results for the multistage multiknapsack models. The intention behind these models is to create academic test instances whose internal coupling, drives our algorithm to its limits. The models can be summarized as follows: Given a stochastic data process $d_t(\xi_t)$, $t = 2, \ldots, T$, of budget vectors, multiknapsack decisions $x_t \in \mathbb{Z}_+^{n_t^1}$, $t = 1, \ldots, T-1$, and compensations $y_t \in \{0,1\}^{n_t^2}$, $t = 2, \ldots, T$, must be taken, such that the following constraints are met

$$A_{t-1}^1 x_{t-1} - \mathbb{1} M y_t \ \leq \ d_t(\xi_t), \ \ t = 2, \ldots, T.$$

Here, A_{t-1}^1 is a matrix with nonnegative entries, $\mathbb{1}$ a vector of all ones, and M a sufficiently big positive number. So the multiknapsack decision x_{t-1} at time $t-1$ is followed by budget realization at time t and compensation y_t, whose respective component is 1 if the corresponding budget was exceeded. Moreover, we have constraints coupling the multiknapsack decisions in time

$$\sum_{\tau=0}^{t-1} A_{t,\tau}^2 x_{t-\tau} \ \leq \ h_t, \ \ t = 1, \ldots, T-1.$$

The constraints are completed by nonanticipativity

$$H_1(x_1) \ = \ 0, \ \ H_t(x_t, y_t) \ = \ 0, \ t = 2, \ldots, T-1, \ \ H_T(y_T) \ = \ 0$$

where the last claim is vacuous, of course.
The objective is utility maximization with contributions ${b_t^1}^\top x_t$, $t = 1, \ldots, T-1$, by the multiknapsack and $-{b_t^2}^\top y_t$, $t = 2, \ldots, T$, by the compensation decisions (with positive b_t^1, b_t^2).
As multiperiod risk measures we used weighted sums of the expectation and the multiperiod CVaR in (5). The expectation always enters with weight 1, and the multiperiod CVaR with a nonnegative weight β.
Computations for these models were done on a Sun Ultra Sparc III with 1.2 GHz processor and 32 GB RAM.

Columns 1 to 5 of the table show problem dimensions and the weight factors, respectively. Column 6 shows the gap $\frac{GUB - \min_{P \in \mathbf{P}} LB(P)}{1 + |\min_{P \in \mathbf{P}} LB(P)|}$ (in percent) reached with our algorithm (version bb-tree) after 2 hours of computation time. The entries "0.1%" indicate that the gap dropped below this value before the 2 hours expired. Column 7 displays the gaps obtained after 2 hours of computation time with ILOG CPLEX 8.1 [10]. The entries "-" mark instances where the gap did not drop below 1000% within 2 hours. The eighth column reports the savings by the book-keeping (version bb-tree) of single-scenario problems by avoiding repeated solution of identical MILPs. The percentage reflects the quotient of the number of single-scenario problems actually solved and the maximum theoretically possible such number, i.e., the product of the number of scenarios and the number of loops through Step 2 of our algorithm. Column 9 shows the effect of the savings in terms of solution quality (again with bb-tree). It displays the gaps (after the allotted computation time of 2 hours) when solving in each loop through Step 2 every single-scenario problem anew from scratch. Entries "-" again mark that the gap did not drop below 1000%. The tenth column is the counterpart to Column 6 (bb-tree) when doing the book-keeping with the hash table (and leaving the other specifications of the algorithm unchanged).
It becomes apparent that book-keeping with the tree structure might be advantageous, although there are instances where the hash table performes better, and although our hash-table approach is rather straightforward, without the intention of maximal performance improvement.

Table 2 shows some preliminary results for multistage extensions of two-stage stochastic programs arising in the optimization of the operation of energy systems with dispersed generation of power and heat, see [12] for model details. In contrast with our multiknapsack instances, the intention behind these test examples is to study the performance of our algorithm at instances with practical relevance. At this preliminary stage of investigation, we report results for 10-stage risk-neutral models only.

Stages	Scenarios	Variables	Const.	Weight	bb-tree	CPLEX	Savings	Savings	bb-hash
		Cont/Int/Bin		β			Problems	Gap	
3	48	0/21/1080	1083	0	0.1%	468,7%	1,142%	0.1%	0.1%
		56/21/1080	1137	0.5	0.1%	–	0,025%	122.9%	0.1%
		56/21/1080	1137	1	0.1%	–	0,005%	962.1%	0.1%
		56/21/1080	1137	2	0.1%	–	0,002%	–	0.1%
3	96	0/9/2000	2003	0	0.1%	271,1%	68,604%	0.1%	0.1%
		102/9/2000	2103	0.5	0.1%	520,1%	21,433%	0.1%	0.1%
		102/9/2000	2103	1	0.1%	359,7%	21,602%	55.4%	0.1%
		102/9/2000	2103	2	0.1%	–	10,073%	–	0.1%
4	48	0/33/1160	1163	0	0.1%	344,4%	1,210%	–	0.1%
		61/33/1160	1221	0.5	0.1%	430,5%	0,502%	–	0.1%
		61/33/1160	1221	1	0.1%	377,1%	0,144%	–	0.1%
		61/33/1160	1221	2	1,3%	–	0,036%	–	21.7%
4	96	0/33/2120	2123	0	0.1%	567,6%	2,876%	–	0.1%
		109/33/2120	2229	0.5	0.1%	–	2,434%	50.3%	0.1%
		109/33/2120	2229	1	0.1%	–	0,439%	53.8%	0.1%
		109/33/2120	2229	2	2,3%	–	0,029%	60.8%	55.3%
5	48	0/45/2200	1243	0	0.1%	300,5%	0,093%	511.1%	0.1%
		66/45/1240	1305	0.5	0.1%	443,2%	0,060%	617.5%	0.1%
		66/45/1240	1305	1	0.1%	521,8%	0,033%	716.1%	1.7%
		66/45/1240	1305	2	99,1%	–	0,008%	818.3%	71.1%
5	96	0/45/2200	2203	0	0.1%	589,3%	0,145%	656.8%	0.1%
		114/45/2200	2313	0.5	0,2%	–	0,263%	750.6%	0.2%
		114/45/2200	2313	1	2,1%	–	0,091%	823.2%	4.9%
		114/45/2200	2313	2	15,1%	904,5%	0,024%	–	23.1%
6	48	0/93/1560	1563	0	19,8%	570,3%	0,006%	394.4%	77.3%
		83/93/1560	1641	0.5	29,6%	–	0,002%	454.1%	33.9%
		83/93/1560	1641	1	35,3%	–	0,009%	462.2%	34.6%
		83/93/1560	1641	2	33,2%	–	0,024%	620.8%	51.7%
6	96	0/93/2520	2523	0	10.1%	406,9%	0,005%	918.2%	41.9%
		131/93/2520	2649	0.5	19,5%	561,4%	0,004%	–	35.6%
		131/93/2520	2649	1	35,4%	476,7%	0,006%	615.4%	42.3%
		131/93/2520	2649	2	46,1%	–	0,002%	–	60.5%
7	48	0/189/2200	2203	0	0,1%	211,8%	0,167%	896.6%	0.1%
		116/189/2200	2313	0.5	34,9%	341,1%	0,883%	279.8%	35.3%
		116/189/2200	2313	1	36,4%	453,2%	0,376%	279.0%	37.7%
		116/189/2200	2313	2	35,5%	167,9%	0,062%	336.6%	42.2%
7	96	0/189/3160	3163	0	83,9%	480,3%	0,579%	–	31.2%
		164/189/3160	3321	0.5	61,8%	–	0,329%	–	34.1%
		164/189/3160	3321	1	3,9%	902,1%	0,043%	–	52.9%
		164/189/3160	3321	2	6,2%	–	0,184%	–	54.0%

Table 1: Computational results for multiknapsack problems

Stages	Scenarios	Variables Cont/Bin	Const.	bb-tree	CPLEX	Savings Problems	Savings Gap	bb-hash
10	100	34125/7875	68542	16.9%	–	0.0008%	17.1%	15.4%
	250	77090/17790	160726	20.3%	–	0.0007%	21.1%	21.1%
	500	79495/18345	223872	25.9%	40.9%	0.0007%	26.4%	26.0%

Table 2: Computational results for power problems

Computations for these models were done on a Linux PC with 3.0 GHz processor and 4 GB RAM. The columns of Table 2 have the same interpretations as those of Table 1, with the weights column missing, since no risk measure is applied yet. Here, the entries "-" mark instances where CPLEX was unable to find a feasible solution within 4 hours.

Again it becomes apparent that the decomposition approach provides feasible solutions with (rough though) optimality estimates where standard solvers fail. In addition, the importance of book-keeping to avoid repeated solution of identical single-scenario problems is confirmed again.

Altogether, Tables 1 and 2 show the benefit of the principal decomposition setup of our algorithm. Moreover, they indicate the tendency that solution of single-scenario subproblems is often limited to a fairly low percentage of theoretically possible cases. Book-keeping schemes exploiting this observation have critical impact on solution quality.

Acknowledgement. We would like to thank an anonymous referee for suggestions that have improved the final version of this paper.

References

[1] Alonso-Ayuso, A.; Escudero, L.F.; Garín, A.; Ortuño, M.T.; Pérez, G.: An approach for strategic supply chain planning under uncertainty based on stochastic 0-1 programming, Journal of Global Optimization 26 (2003), 97-124.

[2] Alonso-Ayuso, A.; Escudero, L.F.; Ortuño, M.T.: BFC, A branch-and-fix coordination algorithmic framework for solving some types of stochastic pure and mixed 0-1 programs, European Journal of Operational Research 151 (2003), 503-519.

[3] Artzner, P.; Delbaen, F.; Eber, J.-M.; Heath, D.: Coherent measures of risk, Mathematical Finance 9 (1999), 203-228.

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

[5] Carøe, C.C.; Schultz, R.: Dual decomposition in stochastic integer programming, Operations Research Letters 24 (1999), 37-45.

[6] Dentcheva, D.; Römisch, W.: Duality gaps in nonconvex stochastic optimization, Mathematical Programming 101 (2004), 515-535.

[7] Dentcheva, D.; Ruszczyński, A.: Optimization with stochastic dominance constraints, SIAM Journal on Optimization 14 (2003), 548-566.

[8] Eichhorn, A.; Römisch, W.: Polyhedral risk measures in stochastic programming, SIAM Journal on Optimization 16 (2005), 69-95.

[9] Föllmer, H.; Schied, A.: Stochastic Finance: An Introduction in Discrete Time, Walter de Gruyter, Berlin, 2002.

[10] ILOG CPLEX, User's Manual, ILOG Inc., Mountain View, CA, 2002, www.cplex.com.

[11] Kall, P.; Wallace, S.W.: Stochastic Programming, Wiley, Chichester, 1994.

[12] Kuhn, S.; Schultz, R.: Risk neutral and risk averse power optimization in electricity networks with dispersed generation, Preprint 631-2006, Department of Mathematics, University of Duisburg-Essen, 2006. www.uni-due.de/mathematik/d_preprints06.shtml

[13] Lulli, G.; Sen, S.: A branch-and-price algorithm for multistage stochastic integer programming with application to stochastic batch-sizing problems, Management Science 50 (2004), 786-796.

[14] Märkert, A.; Schultz, R.: On deviation measures in stochastic integer programming, Operations Research Letters 33 (2005), 441-449.

[15] Müller, A.; Stoyan, D.: Comparison Methods for Stochastic Models and Risks, Wiley, Chichester, 2002.

[16] Nowak, M.P.; Römisch, W.: Stochastic Lagrangian relaxation applied to power scheduling in a hydro-thermal system under uncertainty, Annals of Operations Research 100 (2000), 251-272.

[17] Ogryczak, W.; Ruszczyński, A.: Dual stochastic dominance and related mean-risk models, SIAM Journal on Optimization 13 (2002), 60-78.

[18] Prékopa, A.: Stochastic Programming, Kluwer, Dordrecht, 1995.

[19] Rockafellar, R.T.; Wets, R. J-B.: Variational Analysis, Springer, Berlin, 1998.

[20] Römisch, W.: Stability of Stochastic Programming Problems, in: [22], 483-554.

[21] Römisch, W.; Schultz, R.: Multistage stochastic integer programs: an introduction, in: M. Grötschel, S.O. Krumke, J. Rambau (Eds.) Online Optimization of Large Scale Systems, Springer, Berlin, 2001, 581-600.

[22] Ruszczyński, A.; Shapiro, A. (Eds.): Handbooks in Operations Research and Management Science, 10: Stochastic Programming, Elsevier, Amsterdam, 2003.

[23] Schultz, R.: Stochastic programming with integer variables, Mathematical Programming 97 (2003), 285-309.

[24] Schultz, R.; Tiedemann, S.: Risk aversion via excess probabilities in stochastic programs with mixed-integer recourse, SIAM Journal on Optimization 14 (2003), 115-138.

[25] Schultz, R.; Tiedemann, S.: Conditional value-at-risk in stochastic programs with mixed-integer recourse, Mathematical Programming 105 (2006), 365-386.

[26] Sen, S.: Algorithms for stochastic mixed-integer programming problems, in: K. Aardal, G.L. Nemhauser, R. Weismantel (Eds.) Handbooks in Operations Research and Management Science, 12: Discrete Optimization, Elsevier, Amsterdam, 2005, 515-558.